



ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO

**Análisis de los protocolos de transmisión OPENUDS en infraestructuras
VDI y su mejoramiento en la gestión de los recursos informáticos**

BYRON PAUL HUERA PALTAN

**Trabajo de Titulación modalidad Proyectos de Investigación y Desarrollo,
presentado ante el Instituto de Posgrado y Educación Continua de la ESPOCH,
como requisito parcial para la obtención del grado de:**

MAGÍSTER EN SISTEMAS DE TELECOMUNICACIONES

Riobamba – Ecuador

ENERO DE 2024

DECLARACIÓN DE AUTENTICIDAD Y CESIÓN DE DERECHOS DE AUTOR

Yo, Byron Paul Huera Paltán, declaro que el presente **Trabajo de Titulación modalidad Proyectos de Investigación y Desarrollo**, es de mi autoría y que los resultados del mismo son auténticos y originales. Los textos constantes en el documento que provienen de otras fuentes están debidamente citados y referenciados.

Como autor, asumo la responsabilidad legal y académica de los contenidos de este proyecto de investigación de maestría, el patrimonio intelectual pertenece a la Escuela Superior Politécnica de Chimborazo.

Riobamba, enero de 2024

Byron Paul Huera Paltán

0603597824

©2018, Byron Paul Huera Paltán

Se autoriza la reproducción total o parcial, con fines académicos, por cualquier medio o procedimiento, incluyendo la cita bibliográfica del documento, siempre y cuando se reconozca el Derecho del Autor.



ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO

EL TRIBUNAL DEL TRABAJO DE TITULACIÓN CERTIFICA QUE:

El **Trabajo de Titulación modalidad Proyecto de Investigación y desarrollo**, titulado Análisis de los protocolos de transmisión OPENUDS en infraestructuras VDI y su mejoramiento en la gestión de los recursos informáticos, de responsabilidad del señor Byron Paul Huera Paltán, ha sido minuciosamente revisado por los Miembros del Tribunal del Trabajo de Titulación, el mismo que cumple con los requisitos científicos, técnicos, legales, en tal virtud el Tribunal Autoriza su presentación..

Ing. Oswaldo Geovanny Martínez Guashima; M. Sc.

PRESIDENTE

Ing. Fabricio Javier Santacruz Sulca; Mgtr.

DIRECTOR

Ing. Diego Marcelo Reina Haro; Mgtr.

MIEMBRO

Ing. Washington Gilberto Luna Encalada; Mgtr.

MIEMBRO

Enero de 2024

DEDICATORIA

A Dios que siempre ha estado presente protegiéndome y ayudándome a cumplir todos los objetivos personales y profesionales propuestos en mi vida.

A mi madre Luisa quien con su amor incondicional me supo encaminar para lograr ser una excelente persona. A mi padre Hugo quien con su ejemplo de trabajo me ha enseñado a sobrellevar cualquier adversidad con humildad y esfuerzo.

A mis hermanos por ser parte de mi vida, por todo su respaldo y palabras de afecto y cariño en los momentos que más lo he necesitado.

Byron

AGRADECIMIENTO

A Dios por su amor incondicional haciendo que cada día de mi vida sea un motivo para reconocer todo lo extraordinario que ha hecho por mí y los míos.

A mis padres que han sido un ejemplo de vida para mí, agradeciéndoles por lograr mis objetivos. A mis hermanos quienes brindaron su apoyo, comprensión y confianza en todas las etapas de mi vida.

Un agradecimiento especial a mi tutor el Ing. Fabricio Santacruz por su gran apoyo en el desarrollo de este trabajo de investigación, al Msc. Washington Luna y al Msc. Diego Reina en calidad de miembros del tribunal de investigación, por su respaldo y confianza para poder culminar el trabajo de titulación.

Byron

TABLA DE CONTENIDO

RESUMEN.....	xviii
SUMMARY	xix

CAPÍTULO I

1.	INTRODUCCIÓN.....	1
1.1	Problema de la Investigación.....	2
<i>1.1.1</i>	<i>Planteamiento del problema.....</i>	<i>2</i>
<i>1.1.2</i>	<i>Formulación del problema.....</i>	<i>2</i>
<i>1.1.3</i>	<i>Sistematización del problema:.....</i>	<i>3</i>
1.2	Justificación de la investigación.....	3
1.3	Objetivos.....	4
<i>1.3.1</i>	<i>Objetivo General.....</i>	<i>4</i>
<i>1.3.2</i>	<i>Objetivos Específicos:.....</i>	<i>4</i>
1.4	Hipótesis.....	4

CAPÍTULO II

2	MARCO TEÓRICO.....	5
2.1	Computación en la nube.....	5
<i>2.1.1</i>	<i>Reseña histórica.....</i>	<i>5</i>
2.2	La computación virtual en la nube.....	6
<i>2.2.1</i>	<i>Que es la educación virtual en la nube.....</i>	<i>6</i>
<i>2.2.1.1</i>	<i>Escritorio como servicio (DaaS).....</i>	<i>8</i>
<i>2.2.2</i>	<i>Herramientas de desarrollo y despliegue.....</i>	<i>9</i>
<i>2.2.2.1</i>	<i>Lenguaje de programación Python.....</i>	<i>9</i>
<i>2.2.2.2</i>	<i>Django como framework.....</i>	<i>9</i>
<i>2.2.2.3</i>	<i>Google App Engine (GAE).....</i>	<i>10</i>
<i>2.2.2.4</i>	<i>Lightweigth Directory Access Protocol (LDAP).....</i>	<i>10</i>
<i>2.2.2.5</i>	<i>MySQL.....</i>	<i>11</i>
<i>2.2.3</i>	<i>Kernel virtual Machine.....</i>	<i>12</i>
<i>2.2.4</i>	<i>Broker de Conexiones.....</i>	<i>13</i>
2.3	OPEN UDS.....	13

2.3.1	Arquitecturas openuds	14
2.3.2	Arquitecturas restantes	15
2.3.2.1	Arquitectura del Bróker de Conexiones:	15
2.3.2.2	Arquitectura LDAP.....	17
2.3.2.3	Arquitectura MySQL.....	17
2.3.2.4	Arquitectura del Tunelizador.....	18
2.3.2.5	Arquitectura de oVirt.....	19
2.3.2.6	Arquitectura para la implementación de una plataforma en la nube para Infraestructuras de Escritorio Virtual.....	19
2.3.3	Base de Datos utilizados	20
2.3.3.1	Base de Datos de Google Course Builder	20
2.3.3.2	Base de Datos de OpenUDS.....	21
2.3.4	Protocolo SPICE	21
2.3.5	Características SPICE	24
2.3.5.1	Aceleración de hardware.....	24
2.3.5.2	Hardware las ventajas de aceleración son.....	25
2.3.5.3	Gestión de la Cache en Spice	25
2.3.5.4	Monitores múltiples	26
2.3.6	Comunicación Cliente servidor Canales	26
2.3.7	Protocolo HTML5	28
2.3.7.1	Por qué HTML5.....	29
2.3.7.2	Tecnologías HTML5	32
2.3.8	Apache Guacamole	35
2.3.8.1	El protocolo de Guacamole	36
2.3.8.2	Guacd.....	37
2.3.8.3	La aplicación web.....	37
2.4	Operacionalización de variable	38
2.5	Matriz de Consistencia	39

CAPÍTULO III

3	METODOLOGÍA DE INVESTIGACIÓN	40
3.1	Implantación del bróker de conexiones	40
3.2	Servidor Django	40
3.2.1.	<i>Modulo South</i>	41
3.2.2	<i>Base de datos Mysql</i>	41

3.2.3	<i>Librerías Mysql para Python server</i>	41
3.2.3.1	<i>Servidor de charting para Python</i>	42
3.2.4	<i>Implantación de OvirtNode.</i>	45
3.2.5	<i>Implantación de OVirt Engine</i>	48
3.2.6	<i>Configuración e implantación de FreeNAS</i>	50
3.2.7	<i>Implementación del Tunnerler</i>	51
3.2.8	<i>Implantación de la plataforma Ovirt y utilización de los protocolos</i>	52
3.2.9	<i>Configuración de los servicios de protocolos HTML5 Y SPICE</i>	54

CAPÍTULO IV

4.	RESULTADOS Y DISCUSIÓN	56
4.1	Escenarios del Prototipo VDI	57
4.1.1	<i>Escenario Uno Protocolo HTML5</i>	57
4.1.2	<i>Escenario Dos Protocolo SPICE</i>	68
4.2	Análisis de los resultados para protocolo HTML5	77
4.3	Análisis de los resultados para el protocolo Spice	79
4.4	Análisis comparativo entre HTML5 Y SPICE	80
4.5	Análisis y validación de resultados	84
4.5.1	<i>Tabla De Análisis De Kmo Y Pruebas De Bartlett</i>	85
4.5.1.1	<i>Planteamiento de hipótesis.</i>	85
4.5.1.2	<i>Análisis factorial:</i>	86
4.5.2	<i>Método de extracción: Análisis de componentes principales</i>	87
	CONCLUSIONES	88
	RECOMENDACIONES	89

GLOSARIO

BIBLIOGRAFÍA

ANEXOS

ÍNDICE DE TABLAS

Tabla 1-2: Operacionalización de variables (Variable independiente)	38
Tabla 2-2: Operacionalización de variables (Variable dependiente)	38
Tabla 3-2: Matriz de consistencia.....	39
Tabla 1-4: Resumen general de protocolo HTML5	77
Tabla 2-4: Resumen general de protocolo SPICE.....	79
Tabla 3-4: Resumen general de protocolo HTML5 - 1	80
Tabla 4-4: Resumen general de protocolo HTML5 - 2.....	81
Tabla 5-4: Resumen general de protocolo HTML5 - 3.....	82
Tabla 6-4: Resumen general de protocolo HTML5 - 4	83
Tabla 7-4: Promedio de ancho de banda HTML5 Y SPICE.....	84
Tabla 8-4: Tabla medida Kaiser.....	85
Tabla 9-4: Matriz de correlaciones y prueba de Bartlett.....	86
Tabla 10-4: Comunalidades	86
Tabla 11-4: Varianza total explicada	87
Tabla 12-4: Matriz de componentes.....	87

ÍNDICE DE GRÁFICOS

Gráfico 1-4:	Resumen general de uso del procesador con HTML5.....	77
Gráfico 2-4:	Resumen general de uso de la memoria RAM con HTML5	78
Gráfico 3-4:	Resumen general de uso de la memoria RAM con HTML5	78
Gráfico 4-4:	Resumen general de uso de la memoria RAM con HTML5	78
Gráfico 5-4:	Resumen general de uso del procesador con SPICE	79
Gráfico 6-4:	Resumen general de uso de la memoria RAM con SPICE.....	79
Gráfico 7-4:	Resumen general de uso de la memoria RAM con SPICE.....	80
Gráfico 8-4:	Resumen general de uso de la memoria RAM con SPICE.....	80
Gráfico 9-4:	Análisis del ancho de banca	81
Gráfico 10-4:	Análisis del ancho de banda utilizando en la plataforma VDI	82
Gráfico 11-4:	Análisis de ancho de banda 3	83
Gráfico 12-4:	Segmentación.....	87

ÍNDICE DE FIGURAS

Figura 2-1:	Representación de la computación en la nube	5
Figura 2-2:	Entorno de aprendizaje personalizado basado en computación en la nube.....	8
Figura 3-2:	Arquitectura Principal.....	15
Figura 4-2:	Arquitectura Bróker de Conexiones.....	16
Figura 5-2:	Arquitectura Ldap	17
Figura 6-2:	Arquitectura Base de Datos.....	17
Figura 7-2:	Arquitectura del tunelizador.....	18
Figura 8-2:	Arquitectura de oVirt	19
Figura 9-2:	Arquitectura de OpenUDS plataforma.....	20
Figura 10-2:	Arquitectura VID SPICE	21
Figura 11-2:	Autenticación de OpenUDS.....	22
Figura 12-9:	Arquitectura Spice	23
Figura 13-2:	Protocolo Spice	24
Figura 14-2:	servidor Spice	27
Figura 15-2:	Cliente spice.....	28
Figura 16-2:	VDI Puertos	28
Figura 17-2:	Arquitectura de HTML5	29
Figura 18-2:	Logo HTML5.....	30
Figura 19-2:	browser html5	33
Figura 20-2:	Dispositivos vinculados	34
Figura 21-2:	Navegadores utilizados	34
Figura 22-2:	Guacamole	36
Figura 1-3:	Configuración MySQL	41
Figura 2-3:	Modificación MySQL.....	42
Figura 3-3:	Configuración UDS	43
Figura 4-3:	Configuración Dashboard	43
Figura 5-3:	Configuración autenticada	44
Figura 6-2:	Configuración servicios	44
Figura 7-3:	Configuración servicio levantado	45
Figura 8-3:	Configuración Máquina virtual.....	45
Figura 9-3:	OvirtNode	46
Figura 10-3:	OvirtNode red	46
Figura 11-3:	Administración NIC.....	47

Figura 12-3: Ovirt ENGINE	47
Figura 13-3: OVirt-Engine pantalla principal.....	49
Figura 14-3: OVirt-Engine pantalla principal maquinas vituales	50
Figura 15-3: FREENAS.....	50
Figura 16-3: FREENAS Configuración.....	51
Figura 17-3: FREENAS.....	52
Figura 18-3: Consola USB.....	52
Figura 19-3: Consola máquina virtual	53
Figura 20-3: Consola interfaz	53
Figura 21-3: Consola cd.....	53
Figura 22-3: Consola levantas protocolos	54
Figura 23-3: Conectividad	54
Figura 24-3: Activación de html5 o Spice	55
Figura 1-4: Laboratorio de pruebas Informática 4 FIE -ESPOCH	56
Figura 2-4: Laboratorio de pruebas con estudiantes, Informática 4 FIE -ESPOCH.....	56
Figura 3-4: Utilización de la máquina Virtual, Informática 4 FIE -ESPOCH.....	57
Figura 4-4: Utilización de la herramienta OVIRT NODE 4.1.6, en el servidor de la plataforma	57
Figura 5-4: Resumen general, en el servidor de la plataforma	58
Figura 6-4: Utilización del CPU, en el servidor de la plataforma.....	58
Figura 7-4: Utilización de la memoria RAM, en el servidor de la plataforma	59
Figura 8-4: Utilización de la memoria RAM, en el servidor de la plataforma	59
Figura 9-4: Utilización del Disco Duro, en el servidor de la plataforma.....	60
Figura 10-4: Utilización del ancho de banda, en el servidor de la plataforma VDI	60
Figura 11-4: Utilización del ancho de banda, en el servidor de la plataforma VDI.	61
Figura 12-4: Utilización del ancho de banda 8 usuarios activos, en el servidor de la plataforma VDI	61
Figura 13-4: Utilización de la memoria RAM por 8 usuarios, en el servidor de la plataforma. 62	
Figura 14-4: Utilización del disco Duro por 8 usuarios, en el servidor de la plataforma	62
Figura 15-4: Utilización del ancho de banda por 8 usuarios, en el servidor de la plataforma... 63	
Figura 16-4: Resumen de la utilización de los recursos, en el servidor de la plataforma.	63
Figura 17-4: Utilización del CPU por 16 usuarios, en el servidor de la plataforma	64
Figura 18-4: Utilización de la memoria RAM por 16 usuarios, en el servidor de la plataforma	64
Figura 19-4: Utilización del disco Duro por 16 usuarios, en el servidor de la plataforma VDI 65	

Figura 20-4: Utilización del ancho de banda por 16 usuarios, en el servidor de la plataforma VDI	65
Figura 21-4: Resumen de la utilización de los recursos 32 usuarios, en el servidor de la plataforma	66
Figura 22-4: Utilización del CPU por 32 usuarios, en el servidor de la plataforma	66
Figura 23-4: Utilización de la memoria RAM por 32 usuarios, en el servidor de la plataforma.....	67
Figura 24-4: Utilización de tasa de utilización del disco duro por 32 usuarios, en el servidor de la plataforma	67
Figura 25-4: Utilización del ancho de banda 32 usuarios, en el servidor de la plataforma	68
Figura 26-4: Resumen general, en el servidor de la plataforma SPICE	68
Figura 27-4: Utilización del CPU, en el servidor de la plataforma SPICE.....	69
Figura 28-4: Utilización de la memoria RAM, en el servidor de la plataforma SPICE	69
Figura 29-4: Utilización de la memoria RAM, en el servidor de la plataforma SPICE	70
Figura 30-4: Utilización del Disco Duro, en el servidor de la plataforma SPICE.....	70
Figura 31-4: Utilización del ancho de banda, en el servidor de la plataforma VDI SPICE.....	70
Figura 32-4: Utilización del ancho de banda, en el servidor de la plataforma VDI SPICE.....	71
Figura 33-4: Utilización del ancho de banda 8 usuarios activos, en el servidor de la plataforma VDI SPICE.....	71
Figura 34-4: Utilización de la memoria RAM por 8 usuarios, en el servidor de la plataforma SPICE.....	72
Figura 35-4: Utilización del disco Duro por 8 usuarios, en el servidor de la plataforma SPICE	72
Figura 36-4: Utilización del ancho de banda por 8 usuarios, en el servidor de la plataforma SPICE	72
Figura 37-4: Resumen de la utilización de los recursos, en el servidor de la plataforma SPICE	73
Figura 38-4: Utilización del CPU por 16 usuarios, en el servidor de la plataforma	73
Figura 39-4: Utilización de la memoria RAM por 16 usuarios, en el servidor de la plataforma	74
Figura 40-4: Utilización del disco Duro por 16 usuarios, en el servidor de la plataforma VDI	74
Figura 41-4: Utilización del ancho de banda por 16 usuarios, en el servidor de la plataforma VDI	74
Figura 42-4: Resumen de la utilización de los recursos 32 usuarios, en el servidor de la plataforma.	75
Figura 43-4: Utilización del CPU por 32 usuarios, en el servidor de la plataforma	75
Figura 44-4: Utilización de la memoria RAM por 32 usuarios, en el servidor de la plataforma.....	76

Figura 45-4: Utilización de tasa de utilización del disco duro por 32 usuarios, en el servidor de la plataforma	76
Figura 46-4: Utilización del ancho de <i>banda</i> 32 usuarios, en el servidor de la plataforma.....	77

ÍNDICE DE ANEXOS

ANEXO A: INGRESO AL SISTEMA OPENUDS

ANEXO B: UTILIZACION DEL OVIRT NODE MONITOREO

ANEXO C: ACCESO POR EL ESTUDIANTE AL SISTEMA

ANEXO D: INGRESO AL SISTEMA POR MEDIO DE CELULAR

ANEXO E: CREACION DE UNA MAQUILNA VIRTUAL EN LA PLATAFORMA
MONITOREO

ANEXO F: CODIGO PYTHON

ANEXO G: PRESUPUESTO DEL PROTOTIPO VDI

ANEXO H: GRAFICA DE LOS COSTOS QUE SE PUEDE IMPLEMENTAR EN EL
PROYECTO

ANEXO I: CERTIFICACIÓN DE REVISIÓN DE NORMAS TÉCNICAS

ÍNDICE DE ABREVIATURAS

EVA	Entorno Virtual de Aprendizaje
ESPOCH	Escuela Superior Politécnica de Chimborazo
EIS	Escuela de Ingeniería en Sistemas
FIE	Facultad de Informática y Electrónica
FREENAS	Free Network Attached Storage
GAE	Google App Engine
HTML5	Hyper Text Markup Language vs 5
IAAS	Infraestructura como Servicio
LAN	Red de Área Local
LDAP	Lightweight Directory Access Protocol
PAAS	Plataforma como Servicio
OPENUDS	Open Universal Desktop Services
KVM	Kernel-based Virtual Machine
RDP	Protocolo de Escritorio Remoto
SAAS	Software como Servicio
SSH	Intérprete de Órdenes Seguro
SPICE	Simple Protocol for Independent Computing Environments
SQL	Lenguaje de Consulta Estructurada
SSL	Capa de Puertos Seguros
VDI	Infraestructura de Escritorio Virtual
VNC	Computación Virtual en Red
VPN	Red Privada Virtual
WAN	Red de Área Amplia

RESUMEN

El objetivo de este estudio fue comparar los protocolos de transmisión OPENUDS como alternativa para la comunicación virtual en la nube mediante el uso de los escritorios virtuales VDI que pueda coexistir con la tecnología actual debido al continuo crecimiento en la adopción de dispositivos móviles como teléfonos inteligentes, tabletas, computadoras portátiles, etc., que está impulsando una insaciable demanda de acceso a los servicios basados en Nube. La identificación de los protocolos más utilizados se lo realizó mediante el estudio de diferente software que utiliza la arquitectura OPENUDS para el manejo de la información en el internet. Además, como desafío adicional, el sistema fue diseñado, utilizando un método comparativo para la selección de un protocolo de transmisión apropiado que conforma el sistema, que esencialmente permite un manejo adecuado de un sistema cliente-servidor entre los protocolos HTML5 y SPICE realizando las pruebas de usos en los laboratorios de computación de la Facultad de Informática y Electrónica, el protocolo HTML5 utilizó (1680 Mbps) de ancho de banda con los máximos usuarios probados (32 usuarios), mientras que el protocolo SPICE utilizó (2000 Mbps) en el ancho de banda. Se concluye que los resultados obtenidos en los dos protocolos, HTML5 usa menos recursos hardware, lo que nos ayudó a descubrir cuáles son sus debilidades y fortalezas del sistema VDI en comparación con un sistema tradicional de máquinas virtuales en las diferentes plataformas informáticas. Recomendamos continuar con el estudio y análisis de estos sistemas descritos en el presente trabajo, ya que en el futuro podrían ser la solución combinada que aborda los principales problemas de las redes en la nube.

Palabras claves: OPENUSD (SOFTWARE), HTML5 (SOFTWARE), SPICE (SOFTWARE), LDAP (SOFTWARE), APACHE GUACAMOLE (SOFTWARE), ESCRITORIO VIRTUAL (SOFTWARE), PROTOTIPO, VDI (SOFTWARE).

SUMMARY

The objective of this study was to compare OPENUDS transmission protocols as an alternative for virtual communication in the cloud using VDI virtual desktops that can coexist with current technology due to the continuous growth in the adoption of mobile devices such as smartphones, tablets, laptops, etc., which is driving an insatiable demand for access to cloud-based services. The identification of the most used protocols was carried out through the study of different software that uses the OPENUDS architecture in the management of information on the Internet. Furthermore, as an additional challenge, the system was designed, using a comparative method for the selection of an appropriate transmission protocol that conforms the system, which essentially allows proper management of a client-server system between the HTML5 and SPICE protocols by testing the uses in the computer laboratories of the Faculty of Informatics and Electronics, the HTML5 protocol used (1680 Mbps) of bandwidth with the maximum users tested (32 users), while the SPICE protocol used (2000 Mbps) in bandwidth. It is concluded that the results obtained in the two protocols, HTML5 uses less hardware resources, which helped us to discover what are the weaknesses and strengths of the VDI system compared to a traditional virtual machine system on different computing platforms. We suggest further study and analysis of these systems described in this paper, as in the future, they could be the combined solution that addresses the main problems of cloud networks.

Keywords: OPENUSD (SOFTWARE), HTML5 (SOFTWARE), SPICE (SOFTWARE), LDAP (SOFTWARE), APACHE GUACAMOLE (SOFTWARE), ESCRITORIO VIRTUAL (SOFTWARE), PROTOTIPO, VDI (SOFTWARE).

CAPÍTULO I

1. INTRODUCCIÓN

El número de dispositivos con capacidades multimedia y conectadas a internet está aumentando rápidamente, ver videos en alta definición y acceder a servicios basados en la nube son las principales actividades de los usuarios que consumen gran capacidad de datos, en la actualidad y en un futuro próximo. La mayor parte de este consumo de datos se produce en acceso a los medios tales como escritorios virtuales.

Se espera que esta alta demanda de datos basados en la nube crezca y sea un fuerte motivador para la adopción de nueva tecnología, incluyendo el uso de medios guiados y no guiados. En términos de topología de red, las redes heterogéneas jugarán un papel importante en la integración de un espectro diverso para proporcionar una alta calidad de servicio (QoS), especialmente en ambientes lejanos donde existe infraestructura localizada que soporta acceso de corto alcance.

El desarrollo del presente trabajo de titulación por lo tanto está orientado al estudio y aplicación de los protocolos utilizados por OpenUDS para acceder a las plataformas VDI, analizando la posibilidad de su coexistencia entre protocolos. Al implementar un prototipo de virtualización, los datos que involucren las dos tecnologías mejoran tanto el rendimiento del sistema como la calidad de servicio del usuario.

Este trabajo de investigación agrega nuevos y modernos conocimientos relacionados con la comunicación entre clientes bajos y servidores altos, que cada vez más es escaso, pudiendo ser un elemento clave para el notable mejoramiento de velocidad, seguridad y confiabilidad en la transferencia de información, además de ser eficiente a la hora de ahorrar dinero.

En la actualidad las telecomunicaciones en nube en nuestro país carecen de una renovada transformación tecnológica que hace necesario involucrarse y promover el desarrollo de nuevas tecnologías que permitan soluciones efectivas, procurando estar a la par con los avances de las telecomunicaciones, satisfaciendo en lo posible las crecientes demandas de usuarios para que tengan un mejor estilo de vida día a día

1.1 Problema de la Investigación

1.1.1 Planteamiento del problema

Realizar esta investigación es de suma conveniencia para quienes están inmersos en el campo de las tecnologías de enseñanza de nueva generación, por cuanto estamos seguros de que servirá como marco referencial para proyectar el grado de desarrollo en el ámbito de gestión de recursos informáticos mediante nube.

Por otra parte, la formulación de este sistema tiene también consecuencias de carácter informático ya que viene a constituir una nueva alternativa que dará cabida a profesionales jóvenes con alto sentido de innovación, contribuyendo de esta manera al desarrollo de muchos sectores los que obtendrán algún beneficio, como la Educación Superior del país, y nosotros como futuros profesionales, igual estaríamos en la capacidad de emprender procesos similares en otras instituciones y empresas.

Se pretende llenar también un gran vacío de conocimientos, formular técnicas y procedimientos que reflejen lo establecido por la calidad de los recursos utilizados por los estudiantes en cuya población es de 1000 estudiantes por periodo académico, ya que la universidad brinda enseñanza a 25 estudiantes por cada curso mediante la ayuda de las plataformas implantadas.

Planteamos un punto de partida para nuevas investigaciones, ampliaciones, nuevas ideas o hipótesis de futuros estudios. Pretendemos que a través del presente documento se puedan generar nuevas alternativas de discusión sobre la posibilidad de adaptar el proceso de administración de los VDI en el ámbito informático.

1.1.2 Formulación del problema

El problema de buscar nuevas alternativas para la comunicación que puedan convivir y combinarse con la actual tecnología incurre en el crecimiento en la adopción de dispositivos como teléfonos inteligentes, tabletas, ordenadores portátiles, y ahora los dispositivos de la "internet de las cosas" que está impulsando una demanda insaciable de acceso a datos a través de las redes mundiales del internet en el caso especial de la nube y los escritorios por medio de VDI.

Esta abrumadora demanda rara vez se cumple hoy, ya que los usuarios nunca se han quejado de que tienen "demasiado ancho de banda" o "servicio muy rápido" a internet, especialmente cuando utilizan redes de alto nivel.

Esto se manifiesta como la contención y la interferencia resultando en un aumento de latencia, inseguridad y disminución de rendimiento del internet, estas desventajas se han convertido en un alcance tremendo.

Aquí es donde se hace mención a una nueva tecnología de acceso a la nube especialmente VDI. Justamente las empresas de servicios que proveen dicha infraestructura han respondido a este reto teniendo en cuenta los requerimientos que esto conlleva.

Teniendo en consideración la coexistencia de los protocolos de transmisión Openuds, el cual mejorara la gestión de los recursos en la infraestructura. Como consecuencia, estos sistemas deben ser diseñados para mejorar la probabilidad de acceso al medio, mediante el uso de técnicas para que el usuario pueda realizar el acceso con gran facilidad desde cualquier dispositivo.

1.1.3 Sistematización del problema

- A. ¿Dónde Analizar de los sistemas VDI y la utilización de protocolos de transmisión Open UDS?
- B. ¿Porque Desarrollar de la teoría de los protocolos utilizados en Open UDS para VDI?
- C. ¿Para qué realizar un análisis de mecanismos de transmisión por medio de ETHERNET?
- D. ¿Por qué el Prototipo de los mecanismos VDI mediante la utilización de Open UDS?

1.2 Justificación de la investigación

Las infraestructuras informáticas de nube para la utilización tanto de docentes y alumnos tienen un alto índice de utilización por lo que, se debe partir desde la aplicación a nivel interno de aquellos conocimientos, métodos y técnicas que a diario se transmiten en las aulas virtuales.

A través de un sondeo y conversación con los docentes y empleados, notamos que a pesar de existir una infraestructura adecuada no se llega a optimizar los recursos. La ausencia de un análisis de estos protocolos de transmisión en la infraestructura VDI, imposibilita el ansiado deseo de llegar a más estudiantes rumbo a la excelencia académica.

Entre los problemas más evidentes dentro del proceso de optimización de los VDI, están los casos referentes a: procesos, recursos hardware, recursos web, y equipos que conllevan pérdida de tiempo, desperdicio de recursos por parte de los estudiantes.

Por lo mismo, se cree que el problema radica en la inexistencia de un sistema de gestión de procesos de optimización de los recursos, que no implique de cierto modo conveniencias entre usuarios, sino más bien sea algo sostenible a largo plazo, de ahí la propuesta de aprovechar los protocolos de transmisión utilizados por UDS para los contenidos en VDI establecidos en Cloud computing.

1.3 Objetivos

1.3.1 Objetivo General

Analizar los diferentes protocolos utilizados por OPENUDS en VDI, para el mejoramiento de la gestión de los recursos informáticos.

1.3.2 Objetivos Específicos:

- Realizar el diagnóstico de la situación actual en relación a la conectividad de VDI en una infraestructura.
- Construir marco teórico que permita adaptar tecnología OPEN UDS en la administración de las plataformas VDI.
- Utilización de un prototipo de enlace OPEN UDS en VDI, que permita el acceso a la plataforma informática.
- Analizar los diferentes protocolos utilizados por OPENUDS en la plataforma VDI.

1.4 Hipótesis

La coexistencia de los protocolos de transmisión utilizados por OPENUDS, en infraestructuras VDI, mejorara la gestión de los recursos.

CAPÍTULO II

2 MARCO TEÓRICO

2.1 Computación en la nube

2.1.1 *Reseña histórica*

El Instituto Nacional de Estándares y Tecnología de EE.UU. define la computación en la nube como un “modelo para permitir la ubicuidad de información de manera conveniente y de acceso bajo demanda a la red y todo el conjunto de recursos informáticos compartidos y configurables, que pueden ser rápidamente abastecidos y liberados con un mínimo trabajo de gestión e interacción de servicios” (Mell & Grance, 2011).

Gartner define la computación en la nube como un “estilo de computación escalable y flexible habilitado para el trabajo en TI con capacidad de mostrarse como un servicio basado o que utiliza la tecnología de Internet” (Gartner IT Glossary,” n.d.).

Para los usuarios habituales de Internet y computación, el cloud computing es cualquier actividad que se realice en línea, como el acceso a los datos o el uso de un programa de software, desde diferentes dispositivos independientemente de la manera en que se efectúa el acceso a Internet, como en la Figura 1-1. En ésta se observa que las aplicaciones, datos o software no se almacenan en el computador del usuario, sino que se accede a través de la web desde cualquier dispositivo en cualquier lugar en el que se tenga una conexión a Internet (Horrigan, 2008).

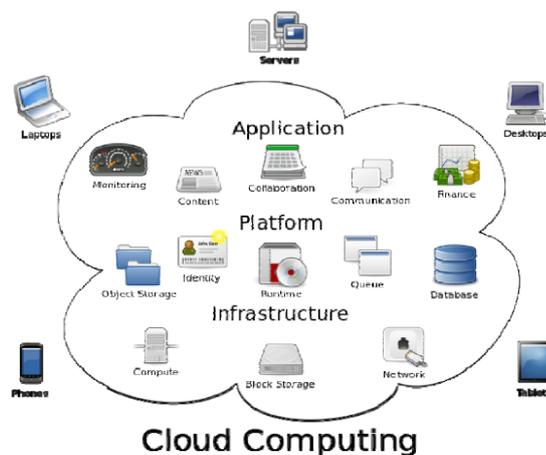


Figura 2-1: Representación de la computación en la nube

Fuente: Johnston, S. 2014.

Se puede decir que la computación en la nube es el nuevo conductor de la revolución en TI, en la cual se están desarrollando nuevos servicios, cambiando la forma en que se accede a la información, la manera en que se usa esta, el mantenimiento y la financiación de los servicios. La computación en nube se caracteriza por la escalabilidad (extensión y flexibilidad de los recursos utilizados de acuerdo con las necesidades de aplicación), la movilidad y la independencia de la plataforma (la capacidad de acceder en cualquier momento, desde cualquier ubicación y dispositivo) (Hewitt, 2008).

Según Bronzin, T., (2009) existen tres tipos de computación en la nube: Infraestructura como Servicio, es el componente de hardware y sus diferentes formas de almacenamiento y tecnología virtual; Plataforma como servicio, implica el uso del sistema operativo y herramientas de desarrollo en la nube; y software como servicio, que se refiere al uso de varias aplicaciones basadas y soportadas en la web, que se ejecutan en el servidor y no requieren de instalación (Žganec, 2011) y (Bronzin, 2009).

Estos modelos difieren en el tipo y el alcance de los recursos accedidos y utilizados por los usuarios. La conexión a la nube hace uso de recursos “ocultos” que permiten el compartir e intercambiar información, la gran escalabilidad de las aplicaciones y herramientas en la nube permite el uso del servicio en cualquier momento y en cualquier lugar, la seguridad de datos, almacenamiento, copias de seguridad, entre otras (Bronzin & Kaldi, 2011).

“Los beneficios potenciales de la apropiación de la computación en la nube se pueden evidenciar financieramente en el ahorro y desde el punto de vista de la administración de recursos” (Lin & Chen, 2012).

2.2 La computación virtual en la nube

2.2.1 *Que es la educación virtual en la nube.*

“El potencial de la computación en nube permitirá mejorar la eficacia, el costo y el interés para el sector educativo, por esto dicha tecnología está siendo reconocida por un gran número de establecimientos educativos sobre todo de los Estados Unidos donde para algunas universidades, la disponibilidad de una impresionante tecnología informática a través de la computación en nube con fines de investigación fue bien recibida”. “Muchas instituciones educativas han comenzado su movimiento hacia la computación en la nube por la trascendencia que tiene en los estudiantes, además también se está comenzando a utilizar los servicios de la nube en un nivel básico, con fines de almacenamiento de datos. Otro uso de la computación en nube que está comenzando a

surgir en la educación es el hospedaje institucional de sistemas gestores de aprendizaje (LMS) en la nube, Según Sultan, (2010, p.109-116).

La externalización de la prestación de LMS como Blackboard o Moodle a un tercero tiene sentido para las instituciones que no pueden justificar los costos de la compra, el mantenimiento y el respaldo al hardware y software de sí mismo” (UNESCO, 2010). La computación en la nube a menudo se asocia con el e-learning y m-learning (Rao, Sasidhar, & Kumar, 2012).

Esto refiriéndose al entorno de e-learning distribuido principalmente en Internet, en la que los recursos educativos abiertos son producidos, investigados y compartidos por participantes de todo el mundo (Kop & Carroll, 2011). Con aplicaciones en la nube, estudiantes y profesores pueden acceder a sus datos de forma flexible a través de un navegador web desde un ordenador en casa, institución educativa, biblioteca, habitación o cualquier lugar, y lograr una rápida y eficiente comunicación, colaboración, intercambio y difusión de documentos, contactos, notas, audio/vídeo entre otros tipos de datos. Teniendo en cuenta esto, los estudiantes pueden crear y usar un “entorno de aprendizaje personalizado basado en computación en la nube” en (Al-Zoube, 2009).

Además de las aplicaciones y datos de uso personal en la nube y correo electrónico gratuito, empresas como Google y Microsoft ofrecen paquetes de aplicaciones (por ejemplo, Google Apps y docs para Educación o Microsoft Live@edu y Office 365, entre otras aplicaciones Educativas) que combinan las herramientas de comunicación y colaboración, con herramientas de oficina para trabajar con documentos, y espacios para almacenar y sincronizar los datos a petición; mostrando que la disponibilidad de las herramientas y aplicaciones antes nombradas es cada vez mayor (Policy Brief UNESCO, 2010).

El uso de servicios y aplicaciones en la nube entre estudiantes y docentes logra romper las barreras espacio temporales o de movilidad, ya que sus recursos educativos y aplicaciones utilizadas están disponibles a través de los dispositivos portables interconectados por medio de la web. Por ejemplo, las clases pueden ser implementadas fuera del campus institucional o facultad y los estudiantes pueden realizar y entregar trabajos o tareas desde diferentes sitios (Al-Zoube, 2009).

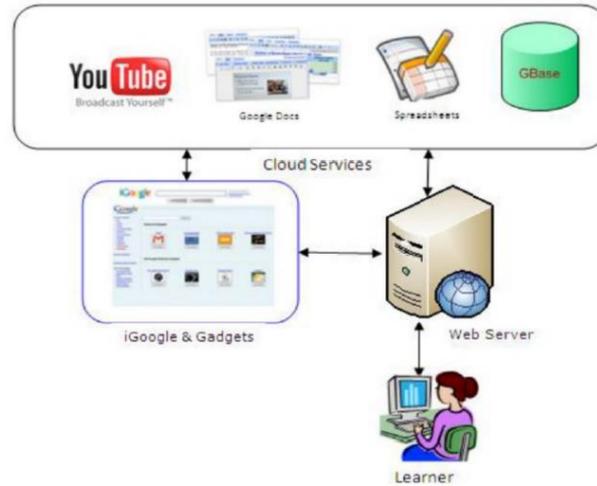


Figura 2-2: Entorno de aprendizaje personalizado basado en computación en la nube

Fuente: Al-Zoube, 2009.

Según Kop y Carroll (2011), “La computación en la nube tiene el potencial para crear nuevas interacciones, metáforas y formas de pensar sobre el aprendizaje, construyéndolo y aprendiendo de las experiencias”. Sin embargo, las posibilidades de la computación en la nube deben ser reconocidas por todos los participantes del proceso educativo con el fin de encontrar sus usos y aplicación en educación.

2.2.1.1 Escritorio como servicio (DaaS)

Con DaaS, los sistemas operativos de escritorio se ejecutan dentro de las máquinas virtuales en servidores en el centro de datos de un proveedor de servicios en la nube. Toda la infraestructura de soporte necesaria, incluidos el almacenamiento y los recursos de red, también vive en la nube. Al igual que con el VDI local, un proveedor de DaaS transmite los escritorios virtuales a través de una red a los dispositivos de punto final de un cliente, donde los usuarios finales pueden acceder a ellos a través del software del cliente o un navegador web. ¿Cómo funciona el escritorio como servicio?

La arquitectura DaaS es multi-tenant, y las organizaciones compran el servicio a través de un modelo de suscripción, generalmente basado en la cantidad de instancias de escritorio virtual que se utilizan por mes.

En el modelo de entrega de escritorio como servicio, el proveedor de computación en la nube maneja las responsabilidades de back-end de almacenamiento de datos, respaldo, seguridad y

actualizaciones. Mientras el proveedor maneja todos los costos y el mantenimiento de la infraestructura de back-end, los clientes generalmente administran sus propias imágenes de escritorio virtual, aplicaciones y seguridad, a menos que esos servicios de administración de escritorio sean parte de la suscripción.

Normalmente, los datos personales de un usuario final se copian ay desde su escritorio virtual durante el inicio de sesión y el cierre de sesión, y el acceso al escritorio es independiente del dispositivo, la ubicación y la red.

2.2.2 Herramientas de desarrollo y despliegue

2.2.2.1 Lenguaje de programación Python

Es un lenguaje de programación de alto nivel, multipropósito, multiparadigma, multiplataforma, interpretado, interactivo y de tipado dinámico, desarrollado bajo una licencia de código abierto administrado por la Python Software Foundation.

Son muchas las características que lo argumentan como un lenguaje de programación productivo, entre las principales tenemos:

- **Multipropósito:** Por su fácil adaptación en el desarrollo de
- **Multiplataforma:** Puede ser utilizado en diversas plataformas y sistemas operativos,
- **Multiparadigma:** Soporta múltiples paradigmas de programación, conteniendo programación orientada a objetos.
- **Tipado dinámico:** el tipo de dato que va a contener una determinada variable, el tipo será asignado a la variable de acuerdo con el valor establecido
- **Interpretado:** No se realiza la transformación del programa a código máquina del sistema
- **Interactivo:** Su facilidad para aprender es excelente para principiantes, como para expertos.

2.2.2.2 Django como framework

Es un framework de código abierto para el desarrollo de aplicaciones web de alto-nivel utilizando Python y regida por el patrón de diseño Modelo-Vista-Template (MTV).

Django se ocupa de un conjunto de componentes necesarias para el desarrollo de una aplicación web.

El framework está diseñado para ayudar a los desarrolladores a crear aplicaciones desde su concepto hasta su culminación, lo más rápido posible.

- Soporte de internacionalización” (Méndez, 2014, p. 27-28).

2.2.2.3 *Google App Engine (GAE)*

Li-fi o también llamada comunicación por luz visible (VLC), utiliza pulsos rápidos de la luz para transmitir información sabiamente. Ya se ha utilizado tecnologías parecidas pero que

GAE es una plataforma de alojamiento web soportada por la infraestructura Google, concebida para crear aplicaciones web y backends móviles escalables y fiables. El paradigma cloud computing ofrecido por la compañía es del tipo Plataforma como Servicio o Plataform as a Service (PaaS) facilitando el construir, mantener y escalar las aplicaciones en la medida que sea necesario, y de forma gratuita hasta determinadas cuotas.

Entre las principales características del porque usar App Engine tenemos:

- **Desarrollo a alta velocidad:** Por el conjunto de servicios que integran la plataforma de Gooogle (Google Cloud Platform), como el registrar aplicaciones.
- **Escalamiento automático:** Cuenta con la elección de escalado automático, permitiendo a las aplicaciones que se ajusten de forma instintiva y rápida
- **Análisis automático seguridad:** El servicio Security Scanner (cloud.google.com/security-scanner) de la plataforma Google Cloud detecta y analiza de manera automática los problemas más comunes de seguridad en las aplicaciones.
- **Utiliza herramientas comunes:** Para el desarrollo de las aplicaciones se pueden utilizar leguajes populares como Java, Python y Go.

2.2.2.4 *Lightweight Directory Access Protocol (LDAP)*

Lightweight Directory Access Protocol (LDAP) es un protocolo cliente-servidor el cual trabaja sobre TCP/IP para acceder y manejar datos sobre directorios. Los directorios son un conjunto de objetos con atributos organizados de una manera lógica y jerárquica.

Según Espinoza las principales características de LDAP son las siguientes:

- **Operaciones de lectura muy rápidas:** Debido a la naturaleza de los datos almacenados en los directorios las lecturas son más comunes que las escrituras.
- **Datos relativamente estáticos:** Los datos almacenados en los directorios no suelen actualizarse con mucha frecuencia. Entorno distribuido, fácil replicación
- **Estructura jerárquica:** Los directorios almacenan la información de forma jerárquica de forma nativa.
- **Orientadas a objetos:** El directorio representa a elementos y a objetos. Los objetos son creados como entradas, que representan a una colección de atributos.
- **Esquema Standard:** Los directorios utilizan un sistema standard que pueden usar fácilmente diversas aplicaciones.
- **Atributos multi-valor:** Los atributos pueden almacenar un valor único o varios.
- **Replicación multi-master:** Muchos de los servidores LDAP permiten que se realicen escrituras o actualizaciones en múltiples servidores” (Espinoza, 2007, p.52).

2.2.2.5 MySQL

MySQL es un sistema de gestión de base de datos (Database Management System, DBMS) de código abierto, regido bajo un lenguaje de consulta estructurado (SQL). Este motor de base de datos brinda los beneficios de almacenamiento y acceso a la información.

“MySQL incluye todos los elementos necesarios para instalar el programa, preparar diferentes niveles de acceso de usuario, administrar el sistema y proteger y hacer volcados de datos. Puede desarrollar sus propias aplicaciones de base de datos en la mayor parte de los lenguajes de programación utilizados en la actualidad y ejecutarlos en casi todos los sistemas operativos, MySQL utiliza el lenguaje de consulta estructurado (SQL). Se trata del lenguaje utilizado por todas las bases relacionales, este lenguaje permite crear bases de datos, así como agregar, manipular y recuperar datos en función de criterios específicos” (López, 2016, p. 5).

“Se ofrece bajo la GNU GPL para cualquier uso compatible con esta licencia, pero para aquellas empresas que quieran incorporarlo en productos privativos deben comprar a la empresa una licencia específica que les permita este uso” (López, 2016, p. 35).

Se presenta las razones para utilizar MySQL según López, P., entre ellas:

- **Coste:** El coste de MySQL es gratuito para la mayor parte de los usos y su servicio de asistencia resulta económico.

- **Asistencia:** MySQL AB ofrece contratos de asistencia a precios razonables y existe una nutrida y activa comunidad MySQL.
- **Velocidad:** MySQL es muchas más rápido que la mayor parte de su competencia.
- **Funcionalidad:** MySQL dispone de muchas de las funciones que exigen los desarrolladores profesionales, como compatibilidad completa, volcados online, duplication, funciones SSL, e integración con la mayor parte de los entornos de programación.
- **Portabilidad:** MySQL se ejecuta en la inmensa mayoría de sistema operativos y la mayor parte de los casos, los datos se pueden transferir de un sistema a otro sin dificultad. Facilidad de uso: MySQL resulta fácil de utilizar de administrar. Gran parte de las viejas bases de datos presentan problemas por utilizar sistemas obsoletos, lo que complica innecesariamente las tareas de administración. Las herramientas de MySQL son potentes y flexibles, sin sacrificar su capacidad de uso” (López, 2016, p. 36).

2.2.3 *Kernel virtual Machine*

Usando KVM, se puede correr múltiples imágenes de máquinas virtuales sin modificar un sistema operativo, ya sea Linux o Windows

.

“KVM es una solución de virtualización completa en la que se utiliza el núcleo de Linux como hypervisor, de manera que tanto el control de los dispositivos reales como la planificación de tareas y la gestión de memoria del sistema anfitrión los hace el núcleo de Linux” (Talens-Oliag, 2010).

La instalación es muy sencilla y tiene un muy buen rendimiento en operaciones de CPU.

Una máquina virtual tendrá 3 modos de ejecución:

- **Modo invitado:** será el modo de ejecución normal para el código del sistema invitado siempre que no tenga operaciones de entrada-salida.
- **Modo usuario:** se usa para ejecutar operaciones de entrada-salida del sistema invitado, gestiona dispositivos virtuales a nivel de usuario.
- **Modo núcleo:** se usa para trabajar en modo invitado y para gestionar las salidas desde modo usuario, causado por operaciones especiales de entrada-salida” (Talens-Oliag, 2010).

2.2.4 Broker de Conexiones

Un broker de conexiones es un elemento software encargado de administrar que servicios o recursos en concreto que se asignará en ese momento al usuario. “Generalmente estos recursos están alojados en un data center y pueden ser virtuales o físicos.

Desde la administración del broker puede definir qué recursos virtuales o físicos ponemos a disposición de cada uno de nuestros usuarios.

Algunos brokers permiten configurar sobre qué hipervisor vamos a alojar nuestros escritorios virtuales o a qué servidor de aplicaciones vamos a acceder.

2.3 OPEN UDS

OpenUDS es un proyecto Open Source, en sus inicios desarrollado por la Compañía Española VirtualCable y liberado como Open Source gracias a Universidades Españolas tales como: Jaen, Malaga, Murcia y Sevilla.

OpenUDS “es un broker de conexiones multiplataforma para:

- VDI: Administración y despliegue escritorios virtuales Windows y Linux.
- Virtualización de aplicaciones.
- Consolidación servicios de escritorio” (UDS Enterprise Team, 2016).

Según el equipo UDS Enterprise “es un software que administra y gestiona el ciclo de vida de escritorios virtuales y el acceso de usuarios remotos y locales, a plataformas de virtualización de escritorios y a recursos físicos dentro del Data Center o Cloud” (UDS Enterprise Team, 2016).

Gestiona y regula el acceso de usuarios desde LAN o WAN a recursos asignados en la Cloud o Data Center, recursos (escritorios, aplicaciones) gestionados automáticamente dentro de un ciclo de vida.

Se presenta un conjunto de caracterices técnicas descritas por el equipo de UDS Enterprise, como:

- “Acceso de usuario desde los principales navegadores web y SO.
- Soporte nativo de escritorios Windows y Linux.
- Sistema avanzado de caché de escritorios virtuales de 2 niveles.

- Acceso WAN Securizado SSL.
- Seguridad incrementada gracias a un sistema de autenticación desdoblada.
- Control de accesos de usuarios a servicios de escritorios y aplicaciones virtuales.
- Programación avanzada de tareas administrativas.
- Multi-hipervisor. Permite elegir el hipervisor más adecuado o habilitar varios de manera simultánea.
- Multi-autenticador. Permite asignar escritorios virtuales a usuarios desde diversas fuentes de manera simultánea y con configuraciones ilimitadas.
- Multi-protocolo. Permite la conexión a escritorios virtuales o recursos físicos utilizando múltiples protocolos de manera simultánea.” (UDS Enterprise Team, 2016).

2.3.1 Arquitecturas openuds

La arquitectura del sistema utilizada para OpenUDS es la óptima en condición a los recursos a disposición usados, separando el servidor de oVirt-node como un procesador independiente debido a la gran cantidad de uso de memoria, procesador y red que este mismo utiliza.

Ya que este es el encargado de dar los recursos de las maquinas virtualizadas bajo oVirt. Y utilizando el segundo procesador como un servidor de virtualización para el resto de los servicios de OpenUDS.

OpenUDS está compuesto por distintos componentes para su correcto desempeño, de los cuales se ha utilizado para el desarrollo de este trabajo de titulación los siguientes:

Bróker de conexiones actualizado,
 Servicio LDAP, Tunelizador,
 Base de datos MySQL,
 Infraestructura de escritorio virtual oVirt.

Todos estos servicios se consumen desde la red de área local mediante la plataforma utilizando código abierto para el acceso alojado bajo la tecnología de computación.

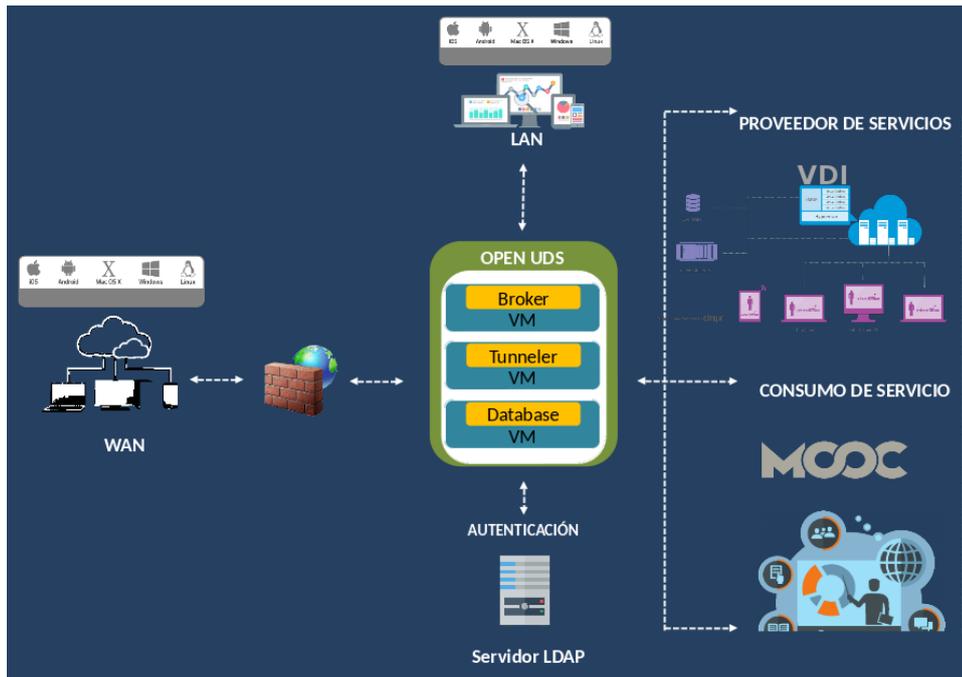


Figura 3-2: Arquitectura Principal

Fuente: Moina W., Borja E., 2017.

En esta figura mostrada se observa los componentes de OpenUDS, los servicios que provee, y los que consumen estos servicios.

2.3.2 Arquitecturas restantes

Mediante el uso de las diferentes arquitecturas se realiza la implantación de la infraestructura de VDI.

2.3.2.1 Arquitectura del Bróker de Conexiones:

El bróker de conexiones de OpenUDS está compuesto esencialmente por los siguientes elementos:

- Aplicación web de administración
- Demonio de OpenUDS taskManager
- Actor UDS engine
- Base de datos de administración interna

Aplicación web de administración: Es una interfaz administrativa escrita en Python 2.7 como núcleo y desarrollada utilizando el framework Django corriendo con el servidor web de Apache.

Para una correcta compilación de la aplicación web son necesarias las siguientes librerías de Python en el sistema:

- Servidor Django 1.8
- Modulo South para Django.
- Librerías MySQL para Python.
- Base de datos MySQL.
- Librerías Ldap para Python.
- Paquete criptográfico para Python.
- Servidor de charting para Python.

Demonio de OpenUDS: Usado para el bróker de conexiones el cual inicializa una serie de demonios utilizados para la gestión de la interfaz gráfica con los servicios de OpenUDS.

Actor UDS: Escrito en Python 2.7 para sistemas operativos basados en Linux y en C# para Windows. El actor es el que proporciona la comunicación y realiza la función de interfaz para la transmisión de datos.

Base de datos de administración interna: La base de datos es necesaria para poner en funcionamiento al Broker, esta base de datos debe ser externa al Broker, proporcionando al Broker derechos administrativos sobre la misma para empezar con el funcionamiento de OpenUDS.

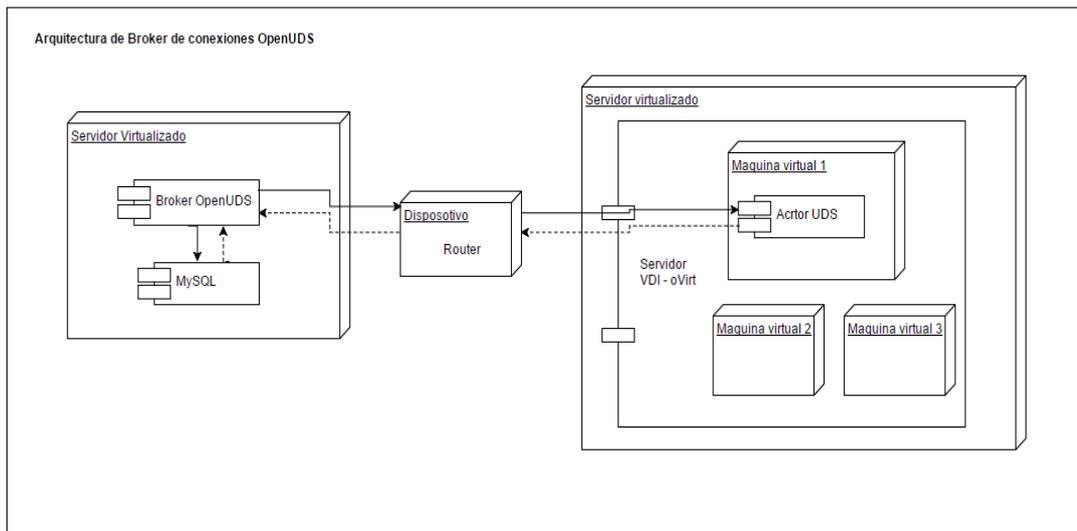


Figura 4-2: Arquitectura Bróker de Conexiones

Fuente: Moina W., Borja E., 2016.

2.3.2.2 Arquitectura LDAP

Ldap está compuesto por dos componentes esenciales: el servicio de Ldap y el directorio de Ldap, en el cual se guarda toda de información de logue de los usuarios de OpenUDS.

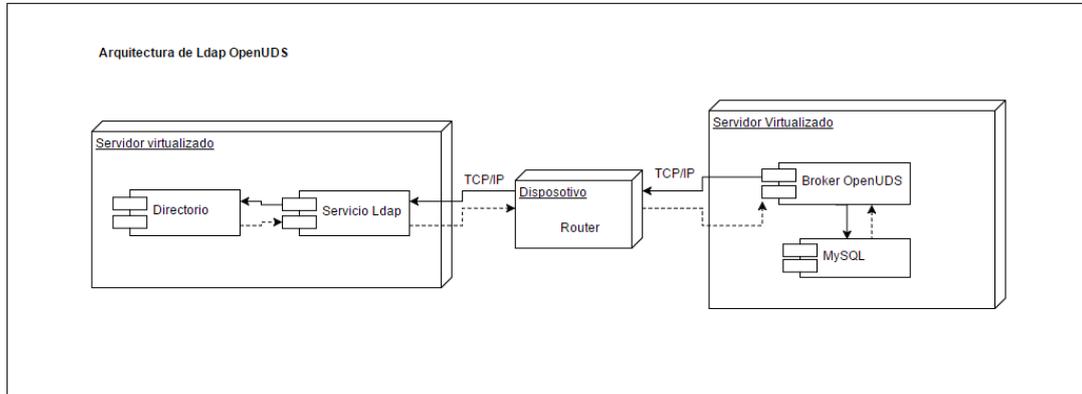


Figura 5-2: Arquitectura Ldap

Fuente: Moina W., Borja E., 2017.

2.3.2.3 Arquitectura MySQL

La base de datos eMySQL es usado para la administración interna del bróker de conexiones de OpenUDS.

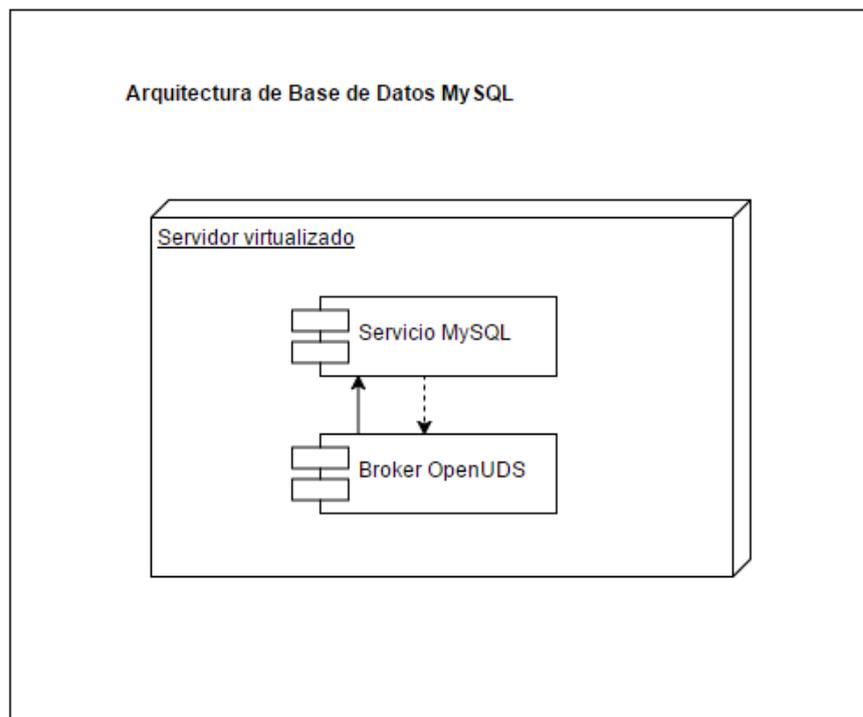


Figura 6-2: Arquitectura Base de Datos

Fuente: Moina W., Borja E., 2017.

2.3.2.4 Arquitectura del Tunelizador

El tunelizador utilizado por el Broker de conexiones es Guacamole Tunnel el cual es utilizado por el Transporte HTML5 RDP Y SPICE, por medio del cual cualquier navegador compatible con HTML5 y SPICE, tendrá acceso a las máquinas virtuales mediante una conexión segura por medio de un túnel SSH.

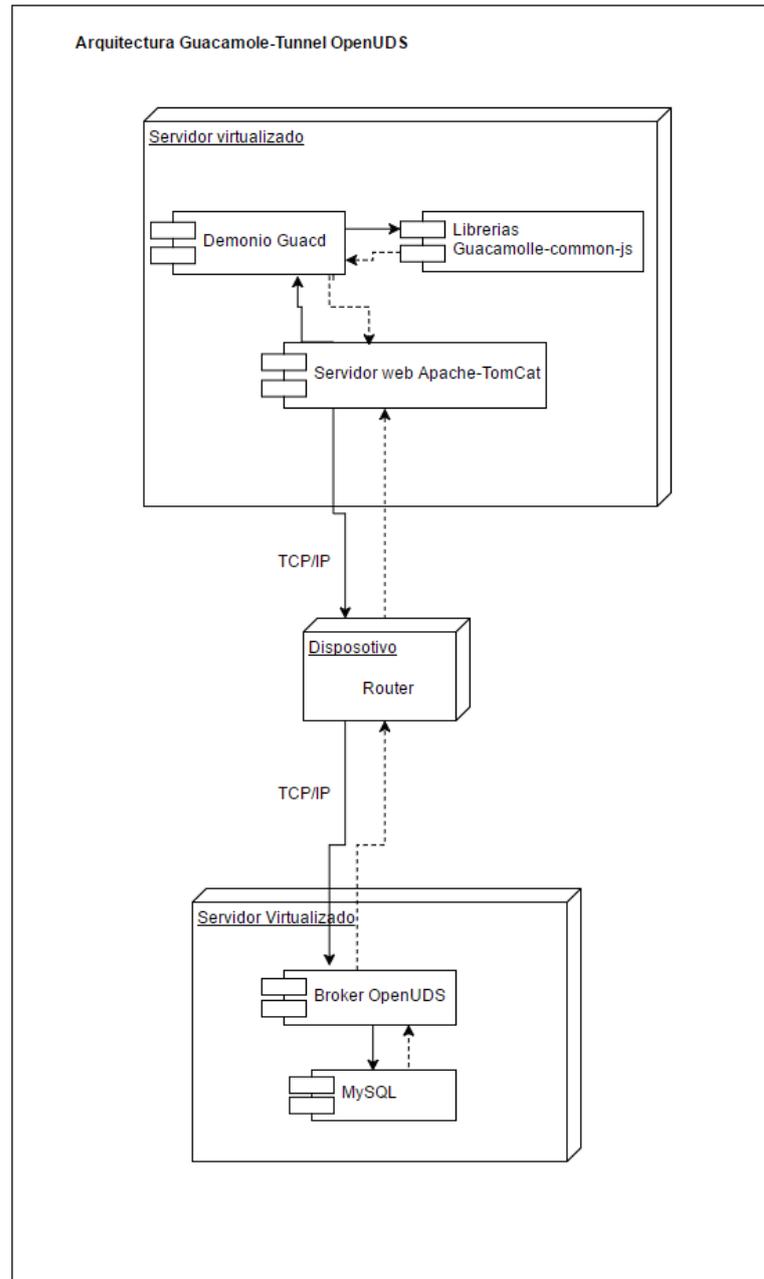


Figura 7-2: Arquitectura del tunelizador

Fuente: Moína W., Borja E., 2016.

2.3.2.5 Arquitectura de oVirt

OVirt es una infraestructura de escritorio virtual administrada por interfaz web, utilizada como un servicio de la plataforma de OpenUDS.

OVirt está constituido por tres elementos:

- oVirt-engine
- oVirt-node
- Almacenamiento

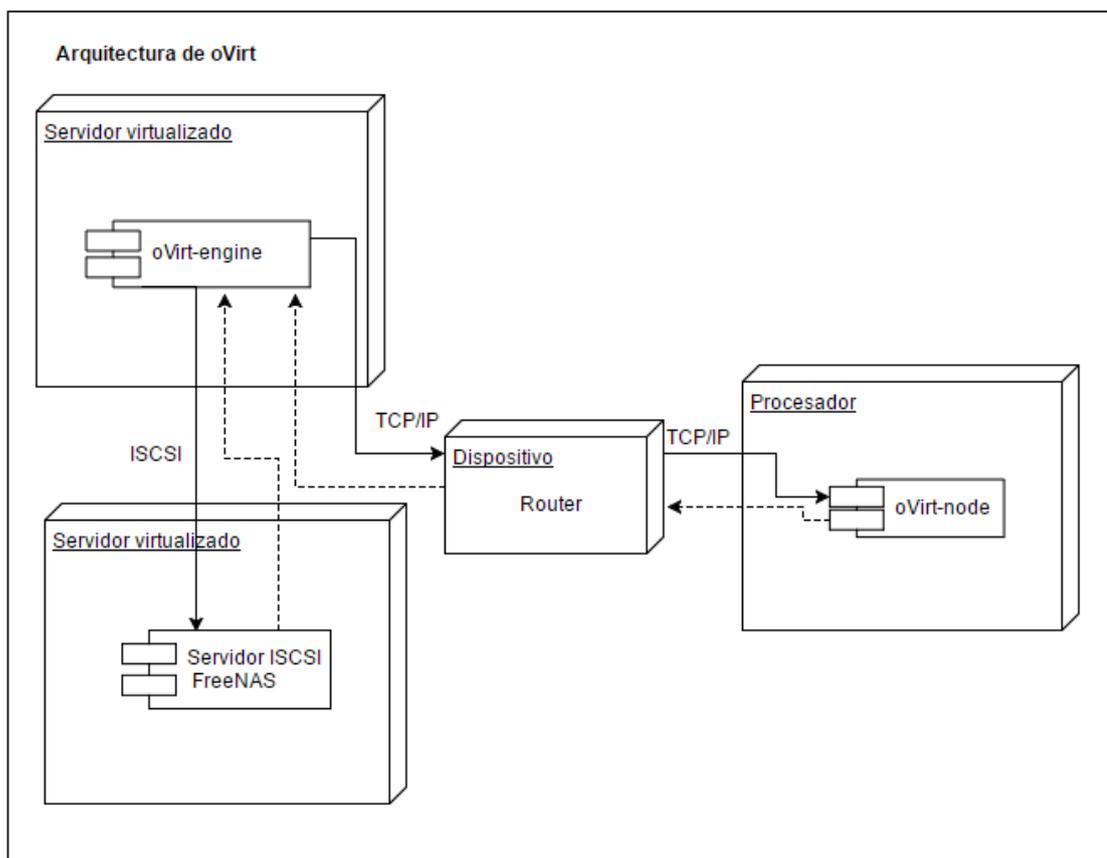


Figura 8-2: Arquitectura de oVirt

Fuente: Moina W., Borja E., 2017.

2.3.2.6 Arquitectura para la implementación de una plataforma en la nube para Infraestructuras de Escritorio Virtual

El prototipo está compuesto por tres servicios que brinda la computación en la nube, se utiliza la IaaS, para conseguir una solución de gestión y administración de escritorios virtuales completamente Open Source se implementa la arquitectura OpenUDS que brinda capacidades de

cómputo, almacenamiento y redes a través de un bróker de conexiones útiles para el suministro de máquinas virtuales a estudiantes registrados en la plataforma, método para fortalecer la parte práctica de una clase tradicional en línea al estar disponibles desde una interfaz web o SO.

Para la integración de este conjunto de servicios se planteó la siguiente arquitectura descrita en la figura 9-2, como una infraestructura convergente que permite simplificar los problemas de compatibilidad.

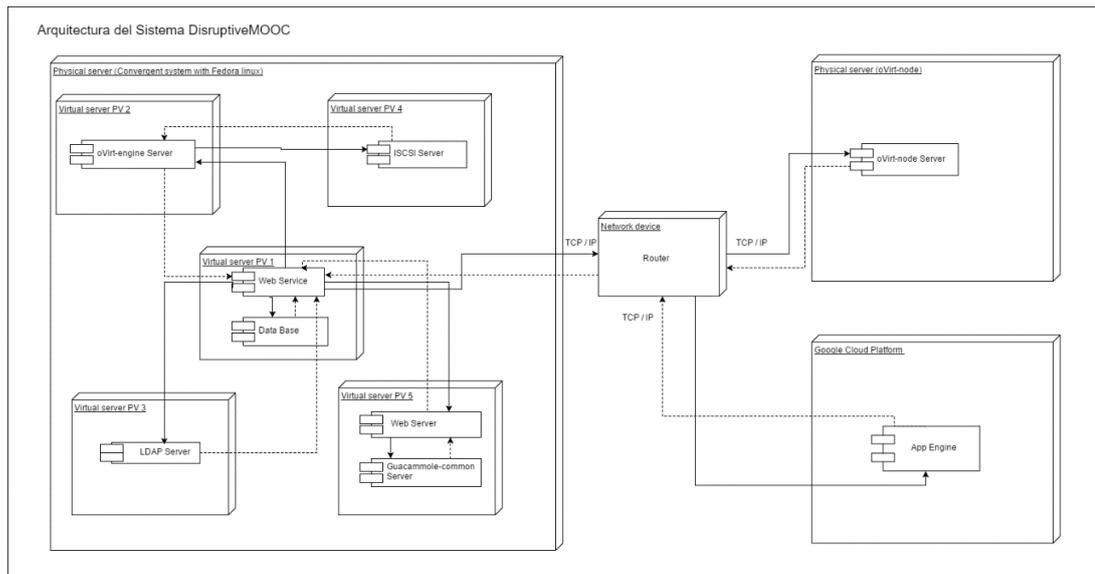


Figura 9-2: Arquitectura de OpenUDS plataforma

Fuente: Moina W., Borja E., 2017.

2.3.3 Base de Datos utilizados

Una base de datos almacena información organizada en un mismo contexto, proporcionando el almacenamiento, recuperación, actualización y análisis de los datos de forma eficaz, utilizado por los sistemas de información de un negocio o empresa en particular.

2.3.3.1 Base de Datos de Google Course Builder

La plataforma Course Builder utiliza el tipo de base de datos NoSQL, al estar desplegada bajo los servicios de Google los datos generados por los contenidos y la estructura de los cursos se guardarán y se gestionarán por Google App Engine Datastore.

2.3.3.2 Base de Datos de OpenUDS

La plataforma OpenUDS utiliza una base de datos migrada MySQL, utilizada para la administración interna de OpenUDS. Fue migrada utilizando el framework Django en su versión para Python.

2.3.4 Protocolo SPICE

SPICE (Protocolo simple para entornos informáticos independientes) es un protocolo de comunicación para entornos virtuales. Permite a los usuarios ver la consola de máquinas virtuales (VM) desde cualquier lugar por Internet. Es un modelo de cliente-servidor que concibe a una Estación virtual como host y los usuarios pueden conectarse a las VM a través del cliente SPICE. Con respecto a Virtualization Station, se tiene instalado un servidor SPICE que se utiliza solo en QVM, y se ejecuta en HybridDesk Station.

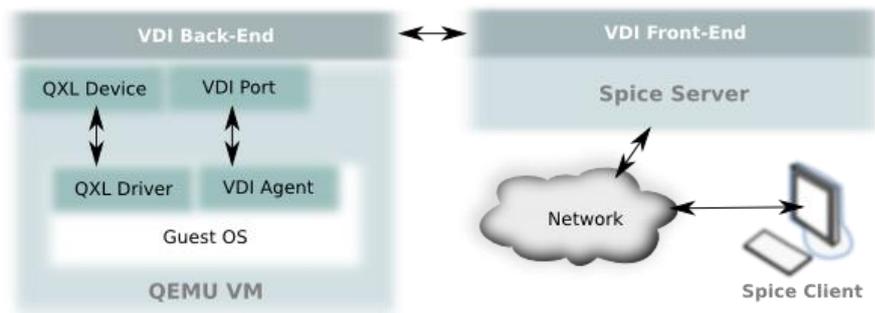


Figura 10-2: Arquitectura VID SPICE

Fuente: SPICE, 2009.

En comparación con la utilización de HTML5, permite la salida de audio y un mejor desempeño a través de una salida HDMI directa. No obstante, SPICE admite solo una conexión por VM por vez, debido al desarrollo incompleto de múltiples conexiones. Por lo tanto, al conectarse a la VM que se utiliza activamente en QVM a través del cliente SPICE, tiene el control total y se fuerza el cierre de la sesión del usuario de QVM original.

Como se mencionó antes, SPICE solo admite una única conexión. Si nadie utiliza la VM como QVM, puede conectarse directamente a la consola VM. Sin embargo, si se utiliza QVM:

Si se utiliza QVM sin contraseña de VM, debe escribir la contraseña predeterminada "@admin" para asumir el control de la consola VM.

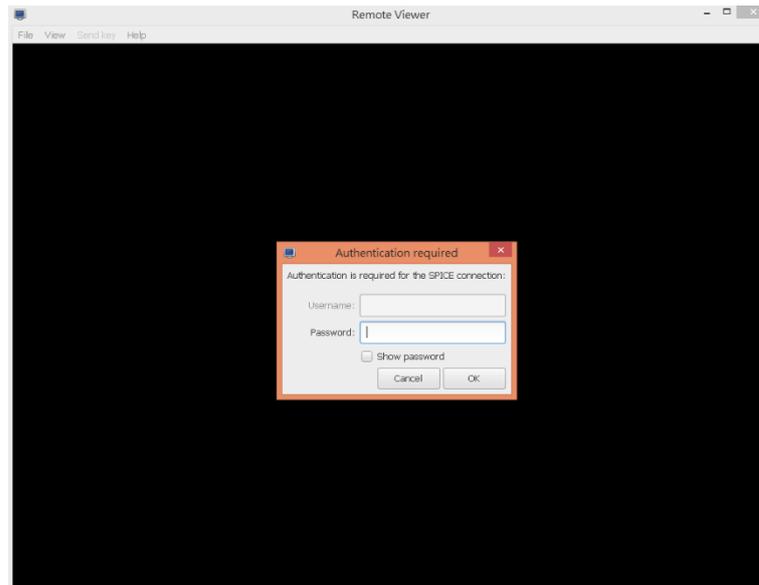


Figura 11-2: Autenticación de OpenUDS

Fuente: SPICE, 2009.

El protocolo Spice define un conjunto de mensajes de protocolo para acceder, controlar y recibir entradas desde dispositivos informáticos remotos (por ejemplo, teclado, video, mouse) a través de redes y envío salida a ellos. Un dispositivo controlado puede residir en cualquier lado, cliente y / o servidor.

En adición, el protocolo define un conjunto de llamadas para admitir la migración de un servidor remoto desde una red dirección a otro. El cifrado de los datos transportados, con una excepción, se mantuvo fuera del protocolo para una flexibilidad máxima en la elección de un método de encriptación. Spice usa simple mensajes y no depende de ningún estándar RPC o una capa de transporte específica.

La sesión de comunicación se divide en múltiples canales de comunicación (por ejemplo, cada canal es un dispositivo remoto) para poder controlar la comunicación y la ejecución de mensajes según el tipo de canal (por ejemplo, cifrado QoS) y para agregar y eliminar canales de comunicación durante el tiempo de ejecución (que es compatible con la definición del protocolo spice).

Los siguientes canales de comunicación se definen en la definición de protocolo actual:

- a) El principal el canal sirve como la conexión principal de la sesión de especias
- b) Canal de visualización para recibir control remoto mostrar actualizaciones
- c) Canal de entradas para enviar eventos de mouse y teclado

- d) Cursor para recepción de la forma y posición del puntero
- e) Canal de reproducción para recibir la transmisión de audio, y
- f) Grabar el canal para enviar la captura de audio. Se agregarán más tipos de canales a medida que el protocolo evoluciona Spice también define un conjunto de definiciones de protocolo para sincronizar la ejecución de canales en el sitio remoto.

SPICE podría dividirse en 4 componentes diferentes: Protocolo, Cliente, Servidor e Invitado. El protocolo es la especificación en la comunicación de los otros tres componentes; Un cliente como Remote-Viewer es responsable de enviar datos y traducir los datos de la Máquina Virtual (VM) para que pueda interactuar con ella; El servidor SPICE es la biblioteca utilizada por el hipervisor para compartir la VM bajo el protocolo SPICE; Y, por último, el lado Invitado es todo el software que debe ejecutarse en la máquina virtual para que SPICE sea completamente funcional, como el controlador QXL y SPICE VDAgent.

Spice Architecture

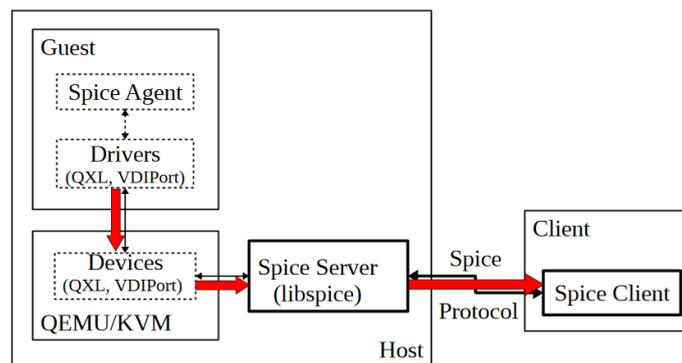


Figura 12-9: Arquitectura Spice

Fuente: SPICE, 2009.

El protocolo Spice admite un canal de comunicación entre el cliente y un agente en el lado del servidor. Al usar QEMU, el agente de Spice reside en el invitado. El puerto VDI es un dispositivo QEMU PCI utilizado para la comunicación con el agente. Se emplea un protocolo de agente específico para la comunicación.

Spice Protocol

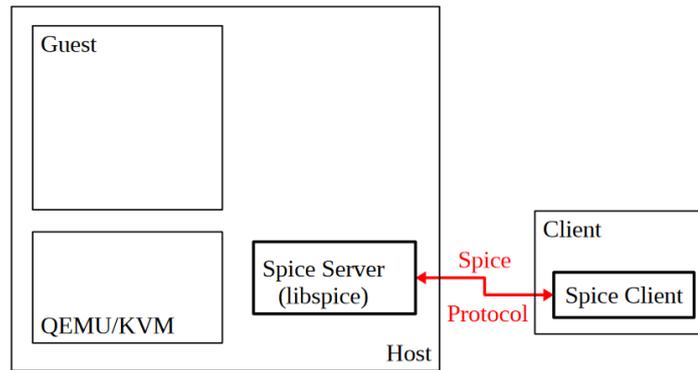


Figura 13-2: Protocolo Spice

Fuente: SPICE, 2009.

Un controlador de invitado de Windows ya está implementado con anterioridad.

2.3.5 Características SPICE

Los comandos gráficos Spice, admite la transmisión y el manejo de comandos gráficos 2D (el soporte 3D está por venir), a diferencia de las actualizaciones de frame buffer, que se utilizan en muchas otras soluciones de escritorio remoto.

Los comandos de los dispositivos QXL son genéricos e independientes de la plataforma, por lo que los controladores de Windows y X usan ellos de forma nativa.

2.3.5.1 Aceleración de hardware

La representación básica del cliente Spice se realiza con Cairo, que es un dispositivo multiplataforma con una biblioteca independiente. Cairo proporciona primitivos gráficos vectoriales para el dibujo bidimensional.

La aceleración de hardware es un modo de renderizado adicional en el que se realiza el renderizado hardware por la GPU del cliente y no por software, usando la CPU del cliente. La aceleración de hardware es implementada usando OpenGL (experimental) en Linux y GDI en Windows.

2.3.5.2 *Hardware las ventajas de aceleración son*

- **Renderización de alto rendimiento:** Al utilizar OpenGL, el cliente Spice puede renderizar mucho más rápido que antes. Las operaciones de software pesado como el estiramiento (utilizado por la transmisión de video) son mucho más rápido cuando está fabricado por hardware que por software. Por lo tanto, Spice logra una mucha experiencia de usuario más suave.
- **Reducción del uso de la CPU del cliente:** el cliente disfruta de más tiempo de CPU, que puede usarse para otras tareas como audio. A diferencia de Cairo, que es una biblioteca de software independiente, OpenGL es una biblioteca de hardware que depende en la implementación del controlador y el hardware. Como resultado, Spice podría sufrir errores procesamiento, uso intensivo de CPU o bloqueo de cliente o host en el peor de los casos. Además, aunque OpenGL es un estándar global, la implementación de hardware y controladores cambia drásticamente entre vendedores. Por lo tanto, en diferentes GPU, Spice podría mostrar diferentes resultados de renderizado, y Se puede detectar un rendimiento diferente. Además, hay dispositivos que no son compatibles OpenGL en absoluto.

El servidor también usa OpenGL para aceleración de hardware, compartiendo el mismo código que el cliente de Linux.

2.3.5.3 *Gestión de la Cache en Spice*

Spice implementa el almacenamiento en caché de imagen del cliente para evitar transmisiones redundantes al cliente.

El almacenamiento en caché se aplica a cualquier tipo de datos de imágenes enviados al cliente, incluidos mapas de píxeles, paletas y cursores. Cada imagen llega desde el controlador con una identificación única y una pista de caché. Imágenes no idénticas tener identificadores diferentes, mientras que las imágenes idénticas comparten la misma identificación. La sugerencia de caché recomienda el servidor para almacenar en caché la imagen.

La memoria caché se define por conexión y sincronizado entre el servidor y el cliente, es decir, en cada momento el servidor sabe exactamente qué imágenes están en la memoria caché del cliente. Además, el servidor es el que decide si un elemento debe agregarse o eliminarse de la memoria caché. El cliente establece el tamaño del caché del cliente y lo transfiere al servidor a través del mensaje de inicialización del canal de visualización.

El servidor monitorea la corriente capacidad de caché, gestiona la misma y cuando carece de espacio, elimina los elementos de caché utilizados menos recientemente hasta que haya suficiente espacio de caché disponible El servidor envía un comando de invalidación con estos elementos y el cliente los elimina.

2.3.5.4 Monitores múltiples

Spice admite cualquier número de monitores, restringido solo por las limitaciones de invitados, clientes y servidores. La cantidad de monitores y su tamaño de RAM se establece al iniciar la máquina virtual. Spice apoya configuración automática de los monitores invitados de resolución y configuración de visualización, de acuerdo con el cliente ajustes de la máquina. Esto se implementa mediante un comando de cliente para el agente invitado.

Compresión de video

Servidor identifica heurísticamente secuencias de video

- Usando el árbol de comandos.
- Áreas constantemente actualizadas por un período de tiempo.
- Envíelos como flujo de video codificado usando M-JPEG.

Ahorra mucho tráfico, mejorando el rendimiento de Spice especialmente en WAN.

Reduce el consumo de CPU del cliente y del servidor.

No requiere ningún códec.

2.3.6 Comunicación Cliente servidor Canales

Cliente y servidor se comunican a través de canales:

- Conexiones TCP
- Cada canal dedicado a un tipo específico de datos.
- El canal se puede proteger utilizando SSL o no seguro.

En el cliente, cada canal tiene un hilo dedicado Diferentes QoS pueden darse por canal.

Spice Servidor: un plugin de conexión

Interfaz de dispositivo virtual (VDI):

- Define un conjunto de interfaces
- Publica dispositivos virtuales (por ejemplo, pantalla, teclado, mouse)
- Permite la interacción con dispositivos

Libspice - el servidor, una biblioteca VDI-conectable:

- Interactúa con la aplicación de host VDI (p. Ej., QEMU)
- Se comunica con el cliente utilizando el protocolo Spice

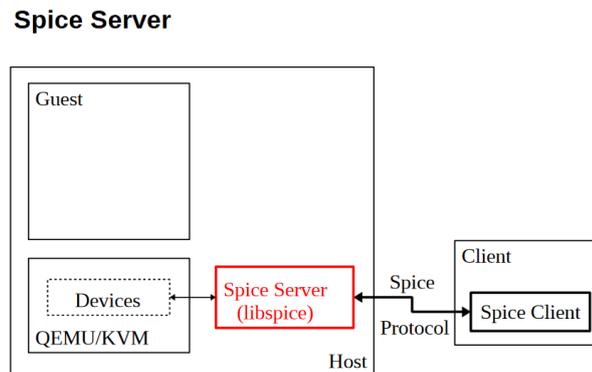


Figura 14-2: servidor Spice

Fuente: SPICE, 2009.

Spice Cliente:

Interfaz de usuario final

- Multiplataforma
- Linux y Windows
- Clases e interfaces genéricas

Implementación paralela de áreas específicas de la plataforma

No requiere ningún códec

Spice Client

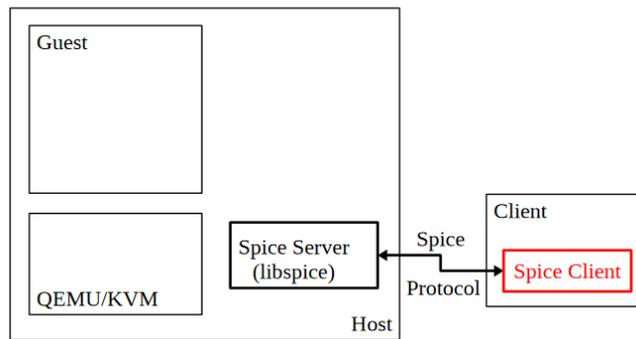


Figura 15-2: Cliente spice

Fuente: SPICE, 2009.

Dispositivo VDIPort

Puerto VDI: dispositivo QEMU PCI utilizado para la comunicación con agente invitado. El agente se comunica con el cliente y el servidor usando agente- protocolo dedicado, controlador invitado de Windows ya implementado.

VDIPort Device & Driver

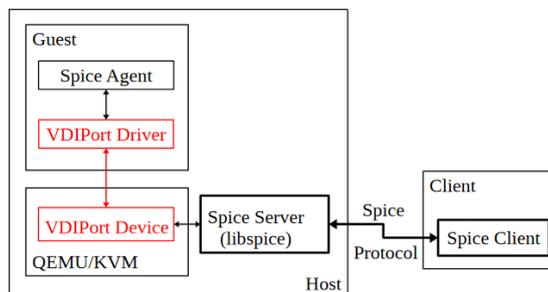


Figura 16-2: VDI Puertos

Fuente: SPICE, 2009.

2.3.7 Protocolo HTML5

Un cliente de escritorio HTML 5 es un tipo de cliente de escritorio remoto que proporciona a un usuario final acceso a un escritorio o aplicación a través de un navegador web.

Otros clientes requieren que el software se instale y ejecute localmente para recibir y representar los datos del protocolo remoto. Sin embargo, un cliente HTML 5 está instalado en un servidor y el usuario final simplemente visita una página web para acceder al escritorio o aplicación remota.

Apache Guacamole Software, es un ejemplo de un cliente de Protocolo de escritorio remoto (RDP) basado en HTML 5 para acceder a las aplicaciones de Windows.

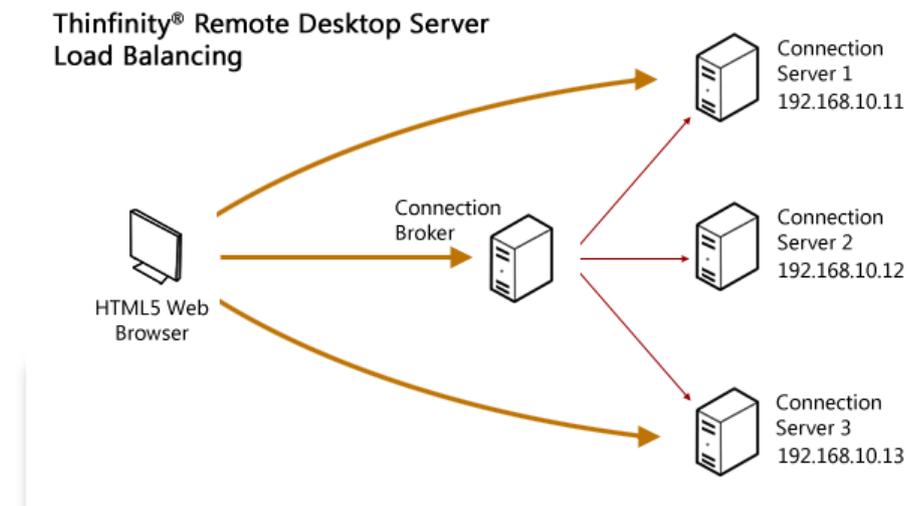


Figura 17-2: Arquitectura de HTML5

Fuente: SPICE, 2009.

2.3.7.1 Por qué HTML5

La comunicación por luz visible no tiene licencia, tiene un amplio ancho de banda, soporta nuevos niveles de seguridad debido a la opacidad de las paredes y puede combinarse para proporcionar tanto iluminación como comunicación de datos.

Acceda a sus computadoras desde cualquier lugar

Debido a que el cliente de Guacamole es una aplicación web HTML5, el uso de sus computadoras no está vinculado a ningún dispositivo o ubicación. Siempre que tenga acceso a un navegador web, tendrá acceso a sus máquinas.

Mantenga su escritorio en la nube

Los equipos de escritorio a los que se accede a través de Guacamole no necesitan existir físicamente. Con Guacamole y un sistema operativo de escritorio alojado en la nube, puede combinar la comodidad de Guacamole con la flexibilidad de la computación en la nube.

Fuente libre y abierta

Apache Guacamole es y siempre será software libre y de código abierto. Está licenciado bajo la Licencia Apache, Versión 2.0, y es mantenido activamente por una comunidad de desarrolladores que usan Guacamole para acceder a sus propios entornos de desarrollo.

Se cree que esto se distingue de otras soluciones de escritorio remoto, y nos da una ventaja distintiva.

Construido sobre una API bien documentada

Apache Guacamole se basa en su propia pila de API centrales que están completamente documentadas, incluidos tutoriales básicos y descripciones conceptuales en el manual en línea. Estas API permiten que Guacamole se integre estrechamente en otras aplicaciones, ya sean de código abierto o de propiedad.

Con apoyo comercial

Para las empresas, el soporte comercial dedicado también está disponible a través de compañías de terceros.



Figura 18-2: Logo HTML5

Fuente: Cooke, A., 2015

WebAssembly para el cliente VDI HTML5

En los últimos años, la infraestructura de escritorio virtual (VDI) han creado clientes HTML5 personalizados para acceder a escritorios remotos en sus stacks, y la cantidad de Gateway RDP HTML5 en el mercado es bastante notable.

Por qué HTML5

En 2011, el protocolo WebSocket fue estandarizado para su uso en aplicaciones cliente / servidor. A diferencia de HTTP, que abre y cierra un socket para cada solicitud, WebSockets utiliza un único socket TCP que se deja abierto, permitiendo que los datos se envíen continuamente entre el navegador y el servidor en tiempo real.

Los clientes VDI HTML5 usan WebSockets para transferir la comunicación entre un servidor y su navegador web, convirtiendo a WebSockets en la primera clave para obtener un rendimiento aceptable en su cliente HTML5 VDI.

Pero, WebSockets solo obtiene los datos en su navegador. Algo más necesita interpretar y representar esos datos en su navegador. Para los clientes VDI HTML5 en el mercado, en este momento, la última parte es un cliente con JavaScript que carga automáticamente en su navegador.

Los clientes VDI HTML5 verifican muchas de las cajas que necesita para satisfacer a los usuarios finales. Son una solución sin cliente, simplificando el despliegue. Los usuarios finales pueden conectarse a su escritorio desde cualquier dispositivo de su elección, incluidos los dispositivos móviles de todo tipo, permitiendo iniciativas BYOD. Y, por naturaleza de ser una puerta de enlace, que proporcionan acceso remoto.

Dicho esto, la aceptación del usuario final a menudo se reduce a las dos "P" de rendimiento y periféricos. Los clientes VDI HTML5 pueden evitar la redirección de impresoras imprimiendo en una impresora PDF local y haciendo que ese archivo sea accesible para su cliente local, pero otros dispositivos USB siguen siendo un desafío.

Varios factores influyen en el rendimiento, como el método utilizado para codificar el video y el audio del escritorio remoto, el ancho de banda entre el servidor y el navegador, y cómo se procesa la sesión remota en el navegador.

WebAssembly

WebAssembly le brinda acceso a un conjunto de elementos básicos de bajo nivel que, potencialmente, puede usar para construir su cliente HTML5 VDI. Es un reemplazo para el componente JavaScript que representa la sesión remota.

WebAssembly está surgiendo como estándar y cuenta con el respaldo de Google, Microsoft y Mozilla. Las versiones experimentales de sus navegadores web ahora son compatibles con WebAssembly, y el rendimiento de la demostración es impresionante.

Mientras tanto, los clientes VDI HTML5 tienen su lugar en una serie de casos de uso: para acceso remoto, para usuarios con dispositivos móviles, como acceso para trabajadores temporales o de tareas, por nombrar algunos.

Al proporcionar a los usuarios finales acceso a los recursos que necesitan, desde los dispositivos que desean, garantizan el éxito de su implementación de VDI.

2.3.7.2 Tecnologías HTML5

Hay dos tecnologías clave que permiten a los clientes de escritorio remoto dentro de un navegador, WebSockets y Canvas. WebSockets es la forma en que los datos del escritorio remoto se envían desde su entorno al navegador, y Canvas es la tecnología que le permite volver a dibujarse en la pantalla.

WebSockets es un protocolo / API que está integrado en todos los navegadores recientes que permite la transmisión continua de datos a través de un socket TCP, en oposición a HTTP, que requiere que cada solicitud tenga una respuesta. Las solicitudes múltiples, entonces, requieren conexiones múltiples, lo cual es bastante complejo e ineficiente para cualquier cosa que necesite tener una sensación en tiempo real. WebSockets cambia esto esencialmente abriendo un canal entre el cliente y el servidor que permanece abierto entre las solicitudes. El principal inconveniente de WebSockets es que solo admite datos de texto, no datos binarios.

Canvas fue creado por Apple en 2004, y se ha convertido en un elemento HTML5 nativo. Canvas habilita la capacidad de controlar cada píxel de manera discreta mediante el uso de javascript, que permite al navegador representar 2D gráficos dinámicamente. Cuando ve animaciones o juegos que se reproducen en el navegador y no usa Flash (es decir, juegos HTML5 como Angry Birds para Chrome), se verá Canvas en acción. Para las conexiones de escritorio remoto, el cliente (en este caso, principalmente un programa de JavaScript) consume los datos que llegan a través de WebSockets y dibuja el escritorio en la pantalla a través de Canvas.

El secreto es con una especie de puerta de entrada. Ericom llama a este servidor AccessNow y Spark View lo llama Spark Gateway. En ambos casos, estas puertas de enlace establecen una sesión de RDP con el host remoto y traducen esos datos binarios en datos de texto para usar con

WebSockets. Los datos de texto se envían al navegador donde el cliente interpreta esos datos y los dibuja en la pantalla con Canvas.

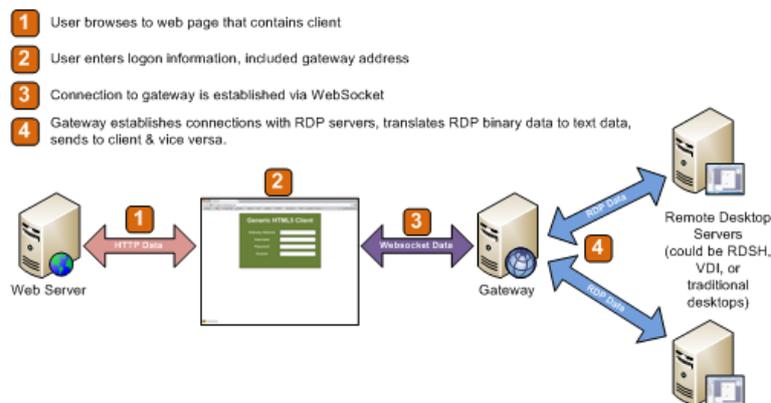


Figura 19-2: browser html5

Fuente: Mullan, B., 2019.

Razones por las que los navegadores con html5 se convertirán en el cliente VDI

Cada vez más clientes basados en navegador se están convirtiendo en el estándar ya que muchas aplicaciones basadas en la nube están usando la interfaz del navegador. Los clientes HTML5 permiten a los usuarios acceder a escritorios y aplicaciones remotos a través de un navegador web que les permite trabajar desde cualquier dispositivo, en cualquier lugar.

Cuando HTML5 apareció por primera vez, la experiencia del usuario necesitaba algunas mejoras, pero ahora los navegadores usan protocolos que lo hacen simple y fácil de usar.

Trabaja desde cualquier dispositivo

Se ha convertido en una práctica cotidiana para las empresas utilizar una combinación de aplicaciones basadas en Windows y Linux combinadas con SaaS. Del mismo modo, los usuarios finales utilizan una combinación de dispositivos alternativos, como PC con Windows, Mac, Android e iOS, PC con Linux, Chromebook, teléfonos inteligentes y más. El navegador es algo que vincula todos estos diferentes dispositivos juntos.

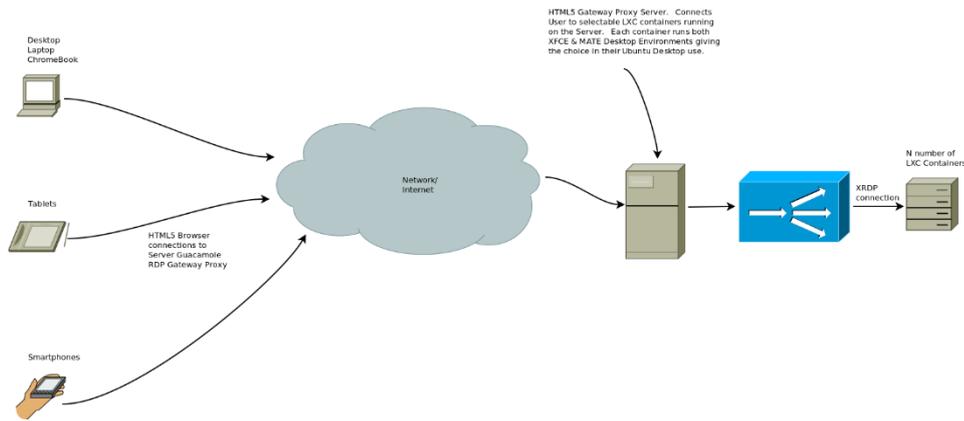


Figura 20-2: Dispositivos vinculados

Fuente: Mullan, B., 2019.

Lo único que todos estos dispositivos tienen en común es el navegador. Aunque no todos los fabricantes de navegadores se adhieren perfectamente al estándar HTML5, casi cualquier dispositivo actual debería ser capaz de renderizar HTML5.

Debido a que las aplicaciones HTML5 no necesitan complementos, los usuarios pueden acceder a sus escritorios desde cualquier dispositivo con un navegador compatible a través de cualquier conexión a Internet. En el mundo de la movilidad de hoy, esa es una gran ventaja.

Las ventajas de navegadores html5 como cliente VDI universal

Hay muchas comparaciones dibujadas entre los navegadores VDI, RDP y HTML5. Para alejarse de las comparaciones y centrarse solo en los beneficios HTML5, aquí hay 3 puntos principales por los que los navegadores HTML5 se están convirtiendo en un cliente universal.

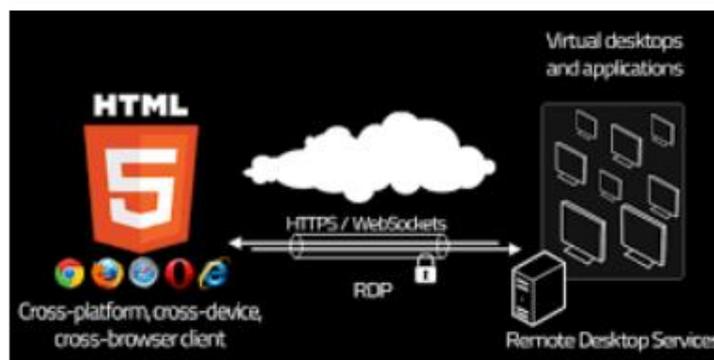


Figura 21-2: Navegadores utilizados

Fuente: Mullan, B., 2019.

- **SENCILLO DE USAR: VDI** generalmente requiere que un cliente se instale localmente en un dispositivo para enviar, recibir y representar datos de protocolo remotos. HTML5 siempre se ejecuta en el navegador. Sin administración, sin instalación local y sin configuración, HTML5 es un medio simple y fácil de usar.
- **NO HAY DATOS LOCALES:** todos los datos residen en la infraestructura de la empresa y no quedan restos en el dispositivo del usuario final. Como resultado, hay más seguridad ya que no hay datos locales guardados en ningún lado.
- **GREAT UI / UX: HTML5** se está convirtiendo en una tecnología completamente madura para desarrollar aplicaciones integrales. Los desarrolladores pueden usar HTML5 para crear aplicaciones web completas que hubieran sido imposibles de construir con versiones anteriores de HTML. Otra ventaja de utilizar navegadores HTML5 para acceder a escritorios remotos es que no se ven ni se sienten significativamente diferentes a los equipos de escritorio a los que se accede desde un cliente local.

Dados los beneficios en movilidad, soporte y flexibilidad, el futuro de VDI puede ser clientes HTML5.

Awingu

Awingu unifica todos los archivos y aplicaciones de la empresa en uno, un espacio de trabajo en línea seguro al que se accede desde cualquier dispositivo que utilice un navegador basado en HTML5.

La solución es brindar la flexibilidad y movilidad de HTML5 también para aplicaciones heredadas que no están escritas en HTML5. Como tal, Awingu puede permitir una transformación digital completa y un movimiento gradual en la nube para empresas de todos los tamaños.

2.3.8 Apache Guacamole

Apache Guacamole es un Gateway de escritorio remoto sin cliente. Admite protocolos estándar como VNC, RDP y SSH. Lo llamamos sin cliente porque no se requieren complementos o software de cliente.

Gracias a HTML5, una vez que Guacamole está instalado en un servidor, todo lo que necesita para acceder a sus escritorios es un navegador web.

Guacamole no es una aplicación web autónoma y está compuesta de muchas partes. La aplicación web en realidad está diseñada para ser simple y mínima, con la mayoría de los gruntwork realizados por componentes de nivel inferior.

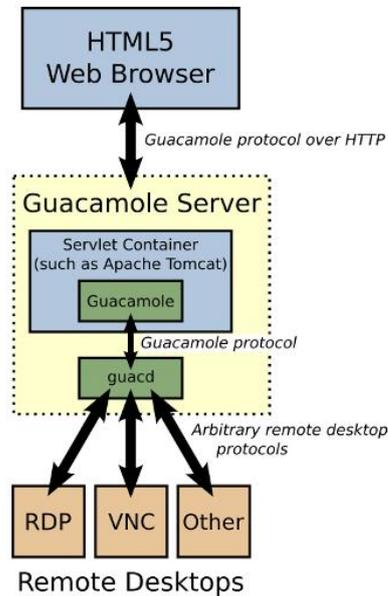


Figura 22-2: Guacamole

Fuente: Guacamole Apache, 2023

2.3.8.1 El protocolo de Guacamole

La aplicación web no comprende ningún protocolo de escritorio remoto. No contiene soporte para VNC o RDP ni ningún otro protocolo soportado por la pila de Guacamole. En realidad, solo comprende el protocolo de Guacamole, que es un protocolo para la visualización remota y el transporte de eventos.

Si bien un protocolo con esas propiedades tendría naturalmente las mismas capacidades que un protocolo de escritorio remoto, los principios de diseño detrás de un protocolo de escritorio remoto y el protocolo Guacamole son diferentes: el protocolo Guacamole no pretende implementar las características de un entorno de escritorio específico.

Como una pantalla remota y protocolo de interacción, Guacamole implementa un superconjunto de protocolos de escritorio remoto existentes. Agregar soporte para un protocolo de escritorio remoto en particular (como RDP) a Guacamole implica escribir una capa intermedia que "traduzca" entre el protocolo de escritorio remoto y el protocolo de Guacamole.

La implementación de una traducción de este tipo no es diferente a la implementación de cualquier cliente nativo, excepto que esta implementación particular se representa en una pantalla remota en lugar de local.

La capa intermedia que maneja esta traducción está garantizada.

2.3.8.2 Guacd

Guacd es el corazón de Guacamole que carga dinámicamente el soporte para protocolos de escritorio remotos (llamados "complementos de cliente") y los conecta a escritorios remotos según las instrucciones recibidas de la aplicación web.

Guacd es un proceso de daemon que se instala junto con Guacamole y se ejecuta en segundo plano, escuchando conexiones TCP desde la aplicación web. Guacd tampoco entiende ningún protocolo de escritorio remoto específico, sino que implementa lo suficiente del protocolo de Guacamole para determinar qué soporte de protocolo necesita cargarse y qué argumentos se le deben pasar. Una vez que se carga un plugin de cliente, se ejecuta independientemente de Guacd y tiene un control total de la comunicación entre él y la aplicación web hasta que finalice el complemento del cliente.

Guacd y todos los complementos de cliente dependen de una biblioteca común, libguac, que hace que la comunicación a través del protocolo de Guacamole sea más fácil y un poco más abstracta.

2.3.8.3 La aplicación web

La parte de Guacamole con la que un usuario realmente interactúa es la aplicación web.

La aplicación web, como se mencionó anteriormente, no implementa ningún protocolo de escritorio remoto. Confía en guacd e implementa nada más que una interfaz web y una capa de autenticación.

Elegimos implementar el lado del servidor de la aplicación web en Java, pero no hay ninguna razón para que no se pueda escribir en un idioma diferente. De hecho, como Guacamole pretende ser una API.

2.4. Operacionalización de variable

Tabla 1-2: Operacionalización de variables (Variable independiente)

VARIABLE INDEPENDIENTE	CONCEPTUALIZACIÓN	DIMENSIONES	INDICADORES	DEFINICION DE LOS INDICADORES	CRITERIOS DE MEDICION	TECNICA	INSTRUMENTO	ESCALA
Protocolos de transmisión OpenUDS en infraestructuras VDI	Conjunto de reglas y procedimientos que permiten la comunicación entre dispositivos en una red VDI	Ancho de banda, en mbps	-Número de conexiones establecidas, -Tasa de transferencia de datos -Latencia, fluctuación del retardo.	Se establece el número de conexiones, tasa de transferencia de datos, latencia, fluctuación del retardo	Numero enteros con decimales	Observación	Software	Numérica

Realizado por: Byron Huera 2018

Tabla 2-2: Operacionalización de variables (Variable dependiente)

VARIABLE DEPENDIENTE	CONCEPTUALIZACIÓN	DIMENSIONES	INDICADORES	DEFINICION DE LOS INDICADORES	CRITERIOS DE MEDICION	TECNICA	INSTRUMENTO	ESCALA
Mejoramiento de la gestión de los recursos informáticos.	Aumento de la eficiencia y la eficacia de la gestión de los recursos informáticos	Productividad de recursos	-Reducción del tiempo de inactividad -Reducción de los costos -Aumento de la productividad	Reducción del tiempo de inactividad, reducción de los costos, aumento de la productividad	Numero enteros	Observación	Laboratorio	Numérica

Realizado por: Byron Huera 2018

2.5. Matriz de Consistencia

Tabla 3-2: Matriz de consistencia

FORMULACIÓN DEL PROBLEMA	OBJETIVO GENERAL	HIPOTESIS	VARIABLES	INDICADORES	TECNICAS	INSTRUMENTOS
El problema de buscar nuevas alternativas para la comunicación que puedan convivir y combinarse con la actual tecnología incurre en el crecimiento en la adopción de dispositivos como teléfonos inteligentes, tabletas, ordenadores portátiles, y ahora los dispositivos de la "internet de las cosas" que está impulsando una demanda insaciable de acceso a datos a través de las redes mundiales del internet en el caso especial de la nube y los escritorios por medio de VDI.	Analizar los diferentes protocolos utilizados por OPENUDS en VDI, para el mejoramiento de la gestión de los recursos informáticos.	La coexistencia de los protocolos de transmisión utilizados por OPENUDS, en infraestructuras VDI, mejorara la gestión de los recursos.	V. Independiente Protocolos de transmisión OpenUDS en infraestructuras VDI	-Número de conexiones establecidas, -Tasa de transferencia de datos -Latencia, fluctuación del retardo.	Observación	Software de conexión.
			V. Dependiente Mejoramiento de la gestión de los recursos informáticos	-Reducción del tiempo de inactividad -Reducción de los costos -Aumento de la productividad	Observación	Ficha de uso de Laboratorio

Realizado por: Byron Huera 2018

CAPÍTULO III

3 METODOLOGÍA DE INVESTIGACIÓN

3.1 Implantación del bróker de conexiones

Para implementar el Bróker de conexiones se debe instalar previamente los siguientes elementos:

- Servidor Django 1.8
- Modulo South para Django.
- Base de datos MySQL.
- Librerías MySQL para Python.
- Librerías Ldap para Python.
- Paquete criptográfico para Python.
- Servidor de charting para Python.

Los siguientes pasos se los realizó en un servidor debían 8 de 64 bits.

3.2 Servidor Django

Instalamos el paquete virtual lenv y pip de python. Actualizado

```
root@debian:~# apt-get install python-pip virtualenv
```

Creamos la carpeta en la cual se ejecutará un ambiente virtual de Python con el nombre.

```
root@debian:~# virtualenv venv
```

Cambiamos de directorio al cual acabamos de crear.

```
root@debian:~# cd venv/
```

Activamos el ambiente virtual en orden para trabajar con Python de una manera controlada.

```
root@debian:~/venv# source bin/activate
```

```
(venv) root@debian:~/venv#
```

En este ambiente virtual se instalarán todas las librerías necesarias para el correcto funcionamiento del bróker de conexiones OpenUDS.

Se instala Django en el ambiente virtual de una manera muy sencilla con el comando pip.

3.2.1. Modulo South

Instalar south por medio del comando pip.

```
(venv) root@debian:~/venv# pip install South
```

3.2.2 Base de datos Mysql

Instalar el servidor MySQL en el servidor Centos.

```
(venv) root@debian:~/venv# apt-get install mysql-server
```

Asignar una contraseña para el administrador.

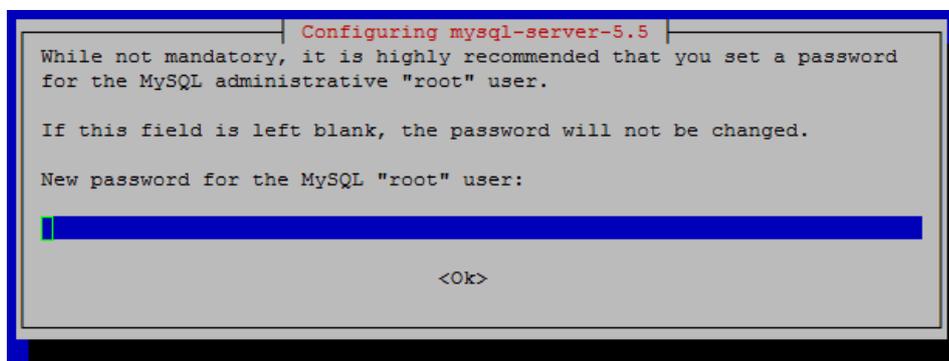


Figura 1-3: configuración MySQL

Realizado por: Huera, B., 2017.

Crear la base de datos de administración interna que será usada por el bróker de conexiones.

```
(venv) root@debian:~/venv/server/src# mysql -u root -p  
Enter password: █
```

Se crea la base de datos.

```
mysql> create database.djangodb;
```

3.2.3 Librerias Mysql para Python server

Instalar el software de desarrollo de Python y de MySQL en el servidor.

```
(venv) root@debian:~/venv# apt-get install python-dev libmysqlclient-dev
```

Configurar MySQL para Python con el comando pip.

```
(venv) root@debian:~/venv# pip install MySQL-python
```

3.2.3.1 Servidor de charting para Python

El servidor de charting a configurar por Python es Pycairo.

```
(venv)root@debian:~/venv# git clone git://git.cairographics.org/git/py2cairo
```

Cambiar al directorio de descarga de Centos

```
(venv)root@debian:~/venv# cd py2cairo/
```

Se debe ejecutar el proceso normal de construcción.

```
(venv)root@debian:~/venv/py2cairo# ./waf configure
```

```
(venv)root@debian:~/venv/py2cairo# ./waf build
```

```
(venv)root@debian:~/venv/py2cairo# ./waf install
```

Instalar los campos se puede proceder a descargar el proyecto OpenUDS

```
(venv)root@debian:~/venv# git clone https://github.com/dkmstr/openuds.git
```

Cambiar de directorio en donde se encuentra el proyecto:

```
(venv)root@debian:~/venv# cd server/src/
```

Cambiar al siguiente directorio:

Copiar los settings de Python para el proyecto va a usar:

```
(venv)root@debian:~/venv/server/src/server# cp settings.py.sample settings.py
```

Modificar los settings de la base de datos para que utilice la base de datos de MySQL.

```
DATABASES = {
    'default': {
        'ENGINE': 'django.db.backends.mysql',
        'NAME': 'djantobd',
        'USER': 'root',
        'PASSWORD': '123456',
        'HOST': '',
        'PORT': '3306',
    }
}
```

Figura 2-3: modificación MySQL

Realizado por: Huera, B., 2017.

Se retrocede un directorio y se procede a realizar la migración de la base de datos:

```
(venv)root@debian:~/venv/server/src/server# cd ..
```

```
(venv)root@debian:~/venv/server/src# python manage.py migrate
```

Con estos pasos se puede configurar la base de datos interna que va a usar para el bróker de conexiones.

Se puede desplegar el proyecto en el servidor Django:

```
(venv) root@debian:~/venv# python manage.py runserver 0.0.0.0:80
```

Funcionamiento del bróker es necesario activar manualmente los demonios del proyecto:

```
(venv) root@debian:~/venv/server/src# python manage.py taskManager --stop
```

El resultado de todas las configuraciones realizadas nos muestra la siguiente pantalla.



Figura 3-3: configuración UDS

Realizado por: Huera, B., 2017.

Se entra en el Dashboard del administrador:

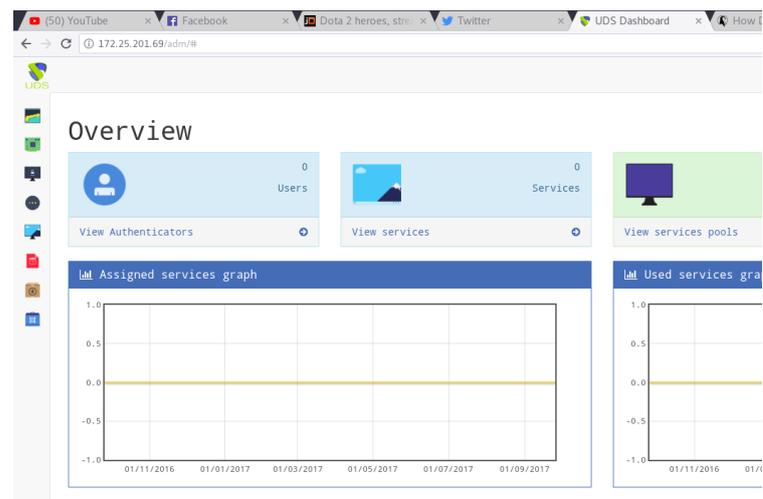


Figura 4-3: configuración Dashboard

Realizado por: Huera, B., 2017.

Entramos en el submenú de Autenticadores de la herramienta utilizada:

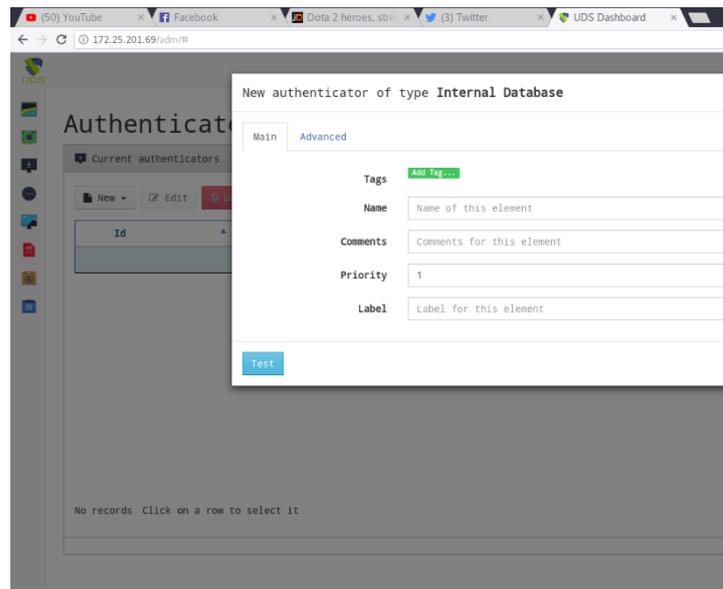


Figura 5-3: configuración autenticada

Realizado por: Huera, B., 2017.

En esta herramienta se puede preparar el resto de los servicios y crear las máquinas virtuales necesarias

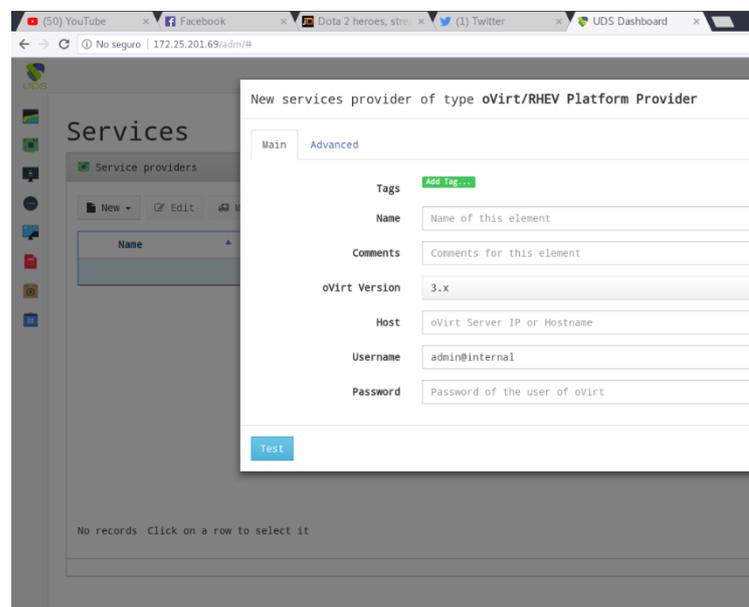


Figura 6-2: configuración servicios

Realizado por: Huera, B., 2017.

Se puede observar que la implementación de un servicio se ha levantado

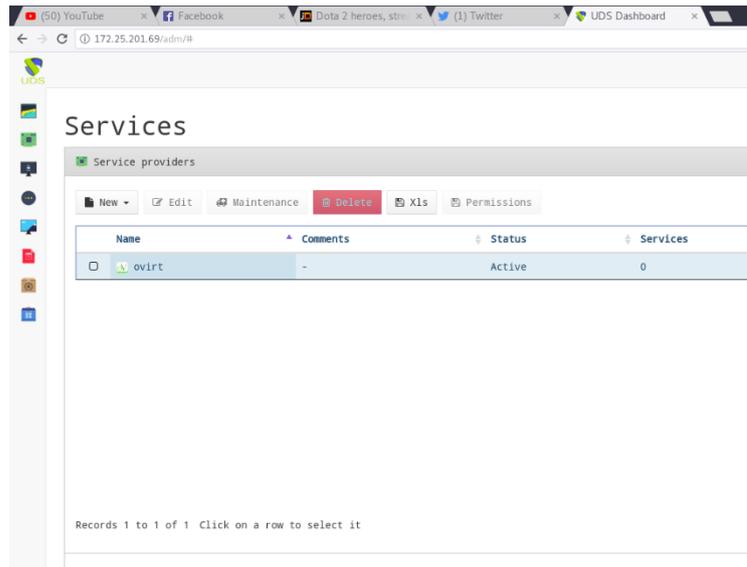


Figura 7-3: Configuración servicio levantado

Realizado por: Huera, B., 2017.

Por último, se puede asignar un determinado espacio de la máquina virtual utilizada en VDI mediante OpenUDS.

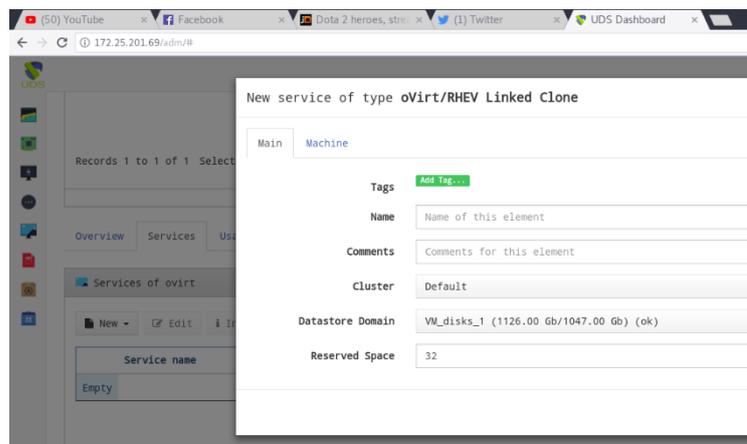


Figura 8-3: Configuración Máquina virtual

Realizado por: Huera, B., 2017.

3.2.4 Implantación de OvirtNode

El sistema operativo ovirt-node se debe instalar en un servidor físico. EL sistema cuenta con un usuario root y la contraseña para el proceso de instalación.

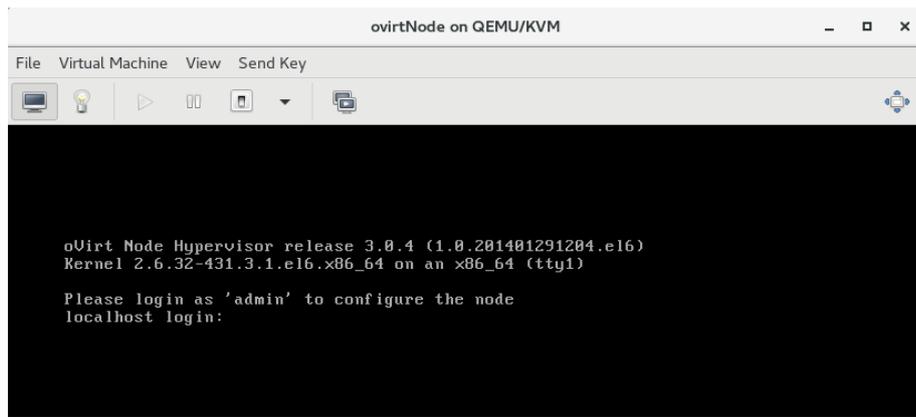


Figura 9-3: OvirtNode

Realizado por: Huera, B., 2017.

La administración de OvirtNode se lo realizará de forma correcta ya que la configuración correcta dará un resultado de conexión.

Configurar la red interna en el dispositivo Ethernet:

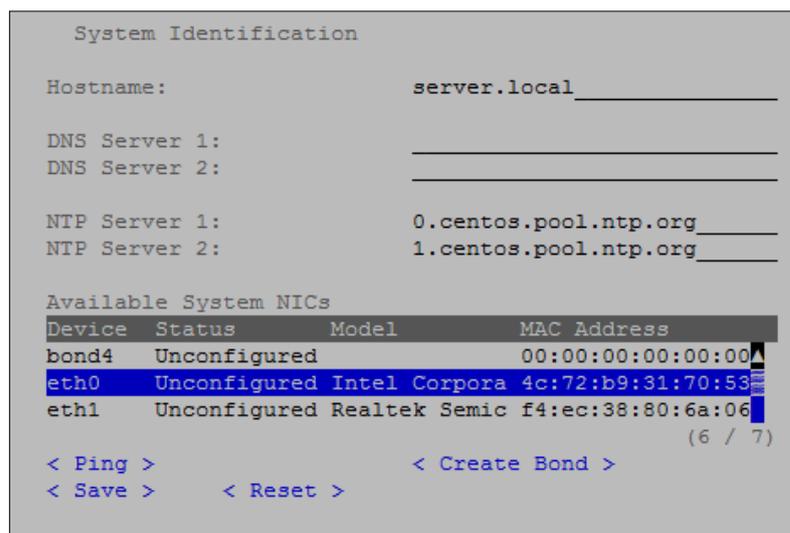


Figura 10-3: OvirtNode red

Realizado por: Huera, B., 2017.

Se debe configurar el nombre del host y la configuración de la red. Utilizar la NIC de la tarjeta física a configurar, procediendo a guardar los cambios correspondientes

```
NIC Details: eth0
Driver:      e1000e          Vendor:      Intel Corporation
Link Status: Connected     MAC Address: 4c:72:b9:31:70:53

IPv4 Settings
Bootprotocol: ( ) Disabled    DHCP      (X) Static
IP Address:   192.168.0.3     Netmask:    255.255.255.0
Gateway:     192.168.0.1

IPv6 Settings
Bootprotocol: (X) Disabled  ( ) Auto    ( ) DHCP    ( ) Static
IP Address:   _____ Prefix Length: _____
Gateway:     _____

VLAN ID:     _____

Use Bridge:  _____ [ ]

< Flash Lights to Identify >
< Save >    < Close >
```

Figura 11-3: Administración NIC

Realizado por: Huera, B., 2017.

Para permitir la conexión con oVirt Engine, se habilita el acceso vía SSH, para lo cual se debe acceder a oVirt Engine y asignamos una contraseña para conexión.

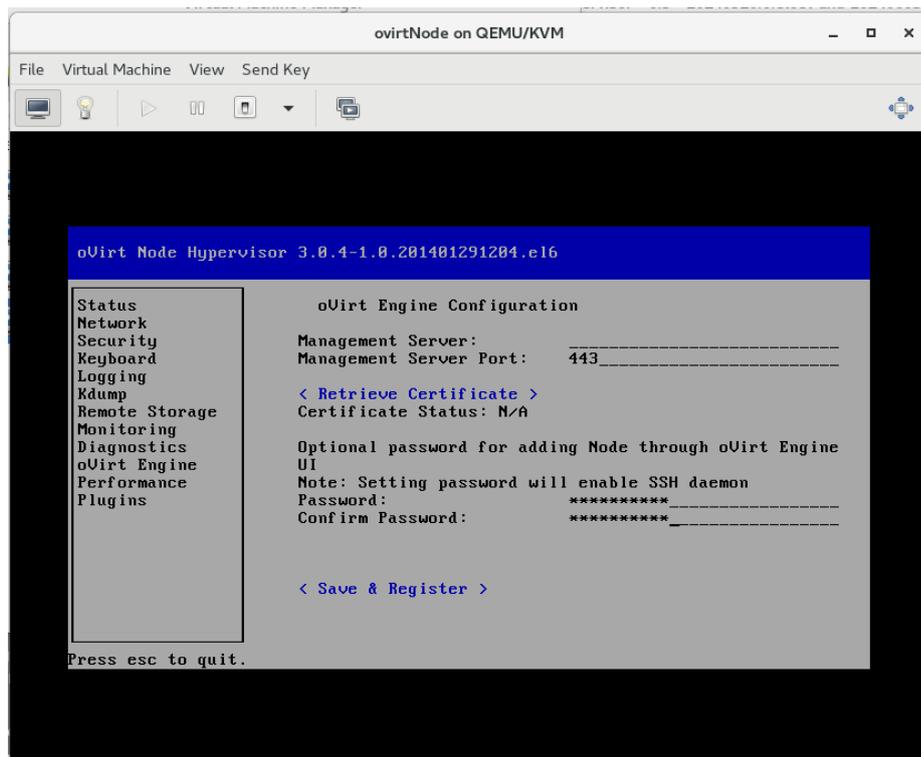


Figura 12-3: Ovirt ENGINE

Realizado por: Huera, B., 2017.

3.2.5 *Implantación de OVirt Engine*

Los siguientes procesos de debe realizar en el sistema operativo CentOS

Configuración de la red:

Configurar de manera estática la CNIC y agregar el nombre de dominio al archivo hosts.

```
[root@localhost ~]# vi /etc/hosts
```

```
DEVICE=eth0
HWADDR=52:54:00:08:EA:19
TYPE=Ethernet
UUID=58e765fb-2527-496f-93ab-061fd0d552c0
ONBOOT=yes
NM_CONTROLLED=no
BOOTPROTO=no
IPADDR=192.168.0.2
NETMASK=255.255.255.0
GATEWAY=192.168.0.1
DNS=192.168.0.1

127.0.0.1 localhost localhost.localdomain localhost4 localhost4.localdomain4
::1 localhost localhost.localdomain localhost6 localhost6.localdomain6
192.168.0.2 eg34.local
```

Descargar el repositorio de oVirt en su versión actual.

```
[root@localhost ~]# yum install http://resources.ovirt.org/pub/yum-repo/ovirt-release35.rpm
```

Se Procede a instalar los paquetes necesarios de oVirt-Engine actualizados:

```
[root@localhost ~]# yum install ovirt-engine
```

Completar la instalación proceder a realizar la configuración del oVirt-Engine:

```
[root@localhost ~]# engine-setup
```

Configurar el Engine en este host:

```
Configure an NFS share on this server to be used as an ISO Domain? (Yes, No) [Yes]: yes
Local ISO domain path [/var/lib/exports/iso]:
Local ISO domain ACL - note that the default will restrict access to eg34.local only, f
or security reasons [eg34.local(rw)]:
Local ISO domain name [ISO_DOMAIN]:
```

Al terminar envía una pantalla de configuración

```
Application mode : both
Firewall manager : iptables
Update Firewall : True
Host FQDN : eg34.local
Engine database name : engine
Engine database secured connection : False
Engine database host : localhost
Engine database user name : engine
Engine database host name validation : False
Engine database port : 5432
Engine installation : True
NFS setup : True
PKI organization : Espoch
NFS mount point : /var/lib/exports/iso
NFS export ACL : eg34.local(rw)
Configure local Engine database : True
Set application as default page : True
Configure Apache SSL : True
Configure WebSocket Proxy : True
Engine Host FQDN : eg34.local

Please confirm installation settings (OK, Cancel) [OK]: ok

[ INFO ] Execution of setup completed successfully
```

Hecho todos estos pasos se debe ingresar al servidor web por medio de un navegador actualizado ya sea Mozilla o google chrome:

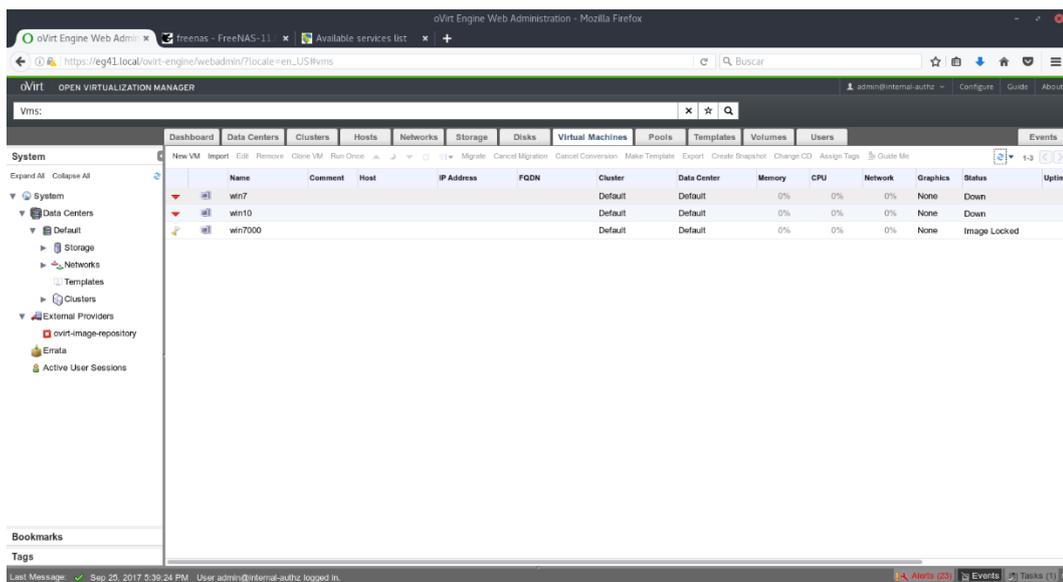


Figura 13-3: OVirt-Engine pantalla principal

Realizado por: Huera, B., 2017.

Aquí se puede observar la configuración de las maquinas disponibles por el servidor

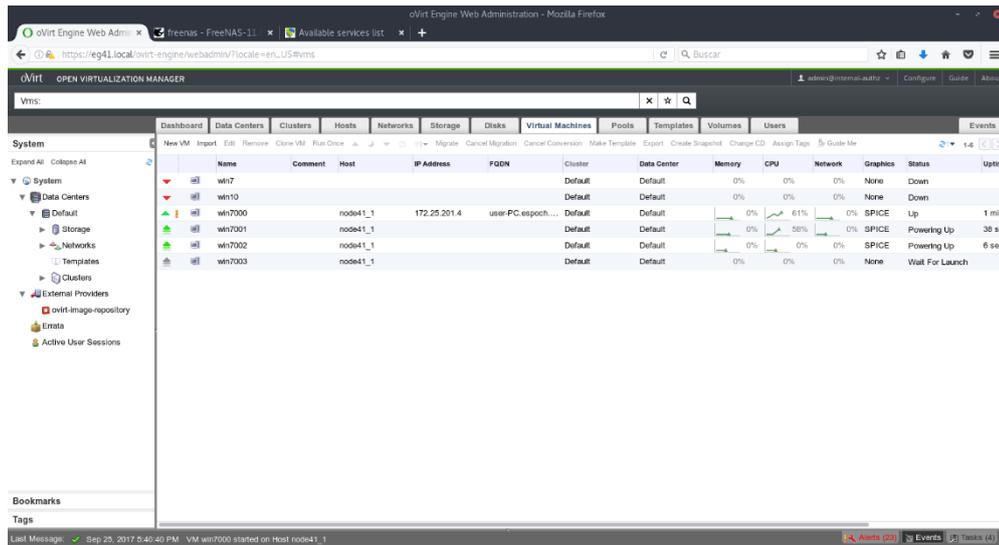


Figura 14-3: OVirt-Engine pantalla principal maquinas virtuales

Realizado por: Huera, B., 2017.

3.2.6 Configuración e implantación de FreeNAS

El servidor FreeNAS de 64 bits se instaló de manera física en el servidor de VDI.

Se debe abrir la interfaz de administración web y loguear con la contraseña de administración.

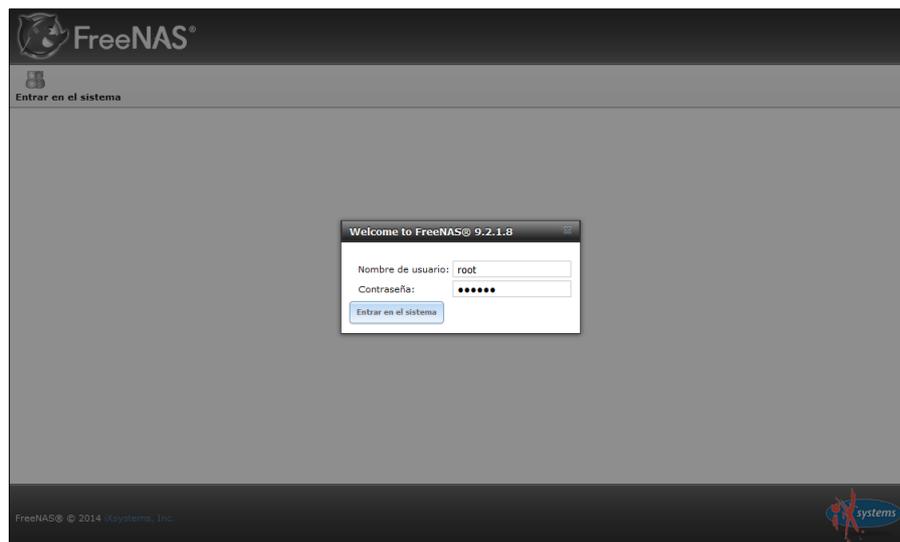


Figura 15-3: FREENAS

Realizado por: Huera, B., 2017.

El tamaño de FreeNAS que se desea configurar

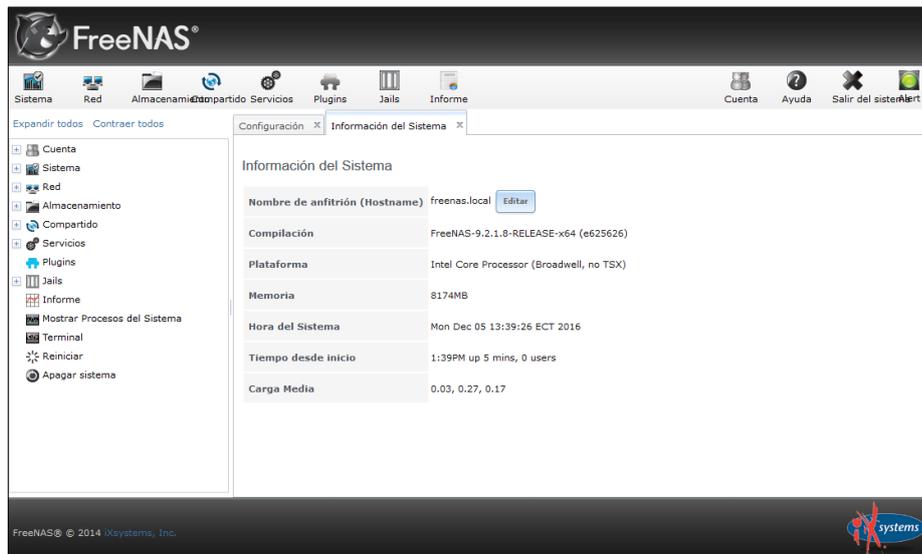


Figura 16-3: FREENAS Configuración

Realizado por: Huera, B., 2017.

3.2.7 Implementación del Tunnerler

Se debe realizar los siguientes pasos en el servidor CentOS de 64 bits.

Instalación de los prerequisites para guacamole en Centos:

```
wget -O /etc/yum.repos.d/home:felfert.repo http://download.opensuse.org/repositories/home:/felfert/Fedora_19/home:felfert.repo
```

```
yum -y install cairo-devel freerdp-devel gcc java-1.8.0-openjdk.x86_64 libguac libguac-client-rdp libguac-client-ssh libguac-client-vnc libjpeg-turbo-devel libpng-devel libssh2-devel libtelnet-devel libvncserver-devel libvorbis-devel libwebp-devel openssl-devel pango-devel pulseaudio-libs-devel terminus-fonts tomcat tomcat-admin-webapps tomcat-webapps uuid-devel
```

Crear el directorio guacamole

```
mkdir ~/guacamole && cd ~/
```

Descargar el proyecto guacamole de su página web oficial:

```
wget http://sourceforge.net/projects/guacamole/files/current/source/guacamole-server-0.9.9.tar.gz
```

Instalar los paquetes de configuración

```
tar -xzf guacamole-server-0.9.9.tar.gz && cd guacamole-server-0.9.9
./configure --with-init-dir=/etc/init.d
make
make install
```

```
ldconfig
```

Configurar el guacamole

Creación en el directorio que albergará el archivo .tar de OpenUDS, cambiando de directorio

```
mkdir -p /var/lib/guacamole && cd /var/lib/guacamole/
```

3.2.8 *Implantación de la plataforma Ovirt y utilización de los protocolos*

Se procede a entrar al sistema de administración web en la pestaña de Máquinas virtuales y agregamos una nueva.

Escoger el sistema operativo, tipo de instancia media, optimizado para escritorio, el protocolo utilizado y dar un nombre a la máquina virtual:

Ciúster	Default/Default
Basado en la plantilla	Blank
Subversión de la plantilla	Plantilla base (1)
Sistema operativo	Windows 7 x64
Tipo de instancia	Medium
Optimizado para	Escritorio
Nombre	Windows 7
Descripción	
Comentario	

Figura 17-3: FREENAS

Realizado por: Huera, B., 2017.

En la pestaña Consola se procede a habilitar el soporte USB y la tarjeta de sonido:

Protocolo	SPICE
Soporte USB	Nativo
Monitores	1 <input type="checkbox"/> PCI sólo

Figura 18-3: consola USB.

Realizado por: Huera, B., 2017.

El almacenamiento de la máquina virtual se procede a configurar:

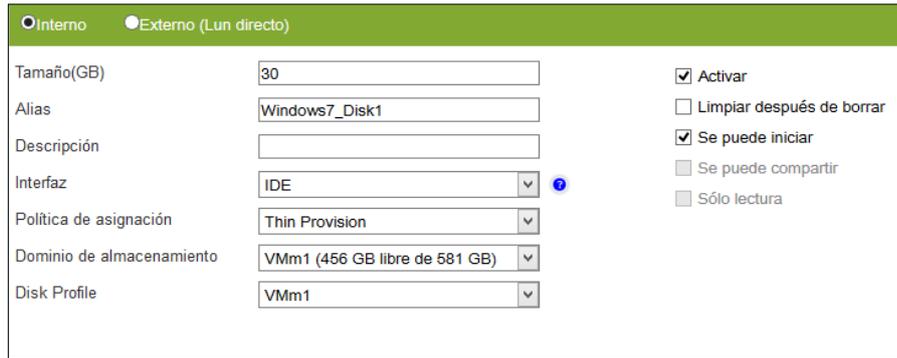


Figura 19-3: consola máquina virtual

Realizado por: Huera, B., 2017.

La máquina virtual que acaba de crear se selecciona y en la pestaña de Interfaces de red se realiza la inserción de una nueva:



Figura 20-3: consola interfaz

Realizado por: Huera, B., 2017.

Se procede a Ejecutar una vez ingresado el cd de instalación del sistema operativo y realizar la instalación.

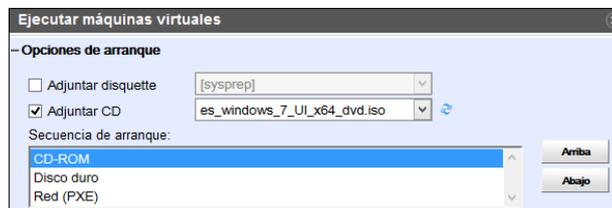


Figura 21-3: consola cd

Realizado por: Huera, B., 2017.

3.2.9 Configuración de los servicios de protocolos HTML5 Y SPICE

Se va a habilitar los servicios de escritorios virtuales que ofrece OpenUDS mediante los protocolos designados. Se ingresa a la interfaz administrativa como usuario root de OpenUDS y la respectiva contraseña agregamos un nuevo oVirt/RHEV Platform Provider.

Agregar una IP de oVirt-Engine:

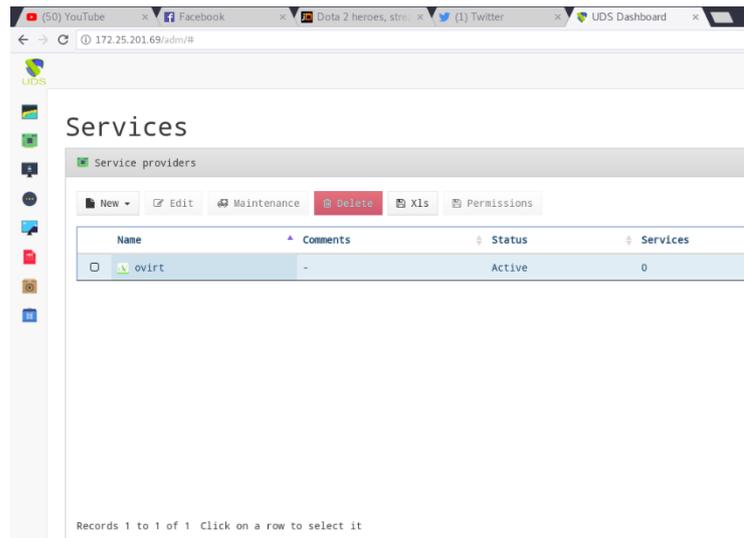


Figura 22-3: consola levantas protocolos

Realizado por: Huera, B., 2017.

Configuración del Transporte:

Hacer clic en el submenú Conectivity, se agrega una nueva red y se escribe el rango de IPS válidas para ese proyecto:

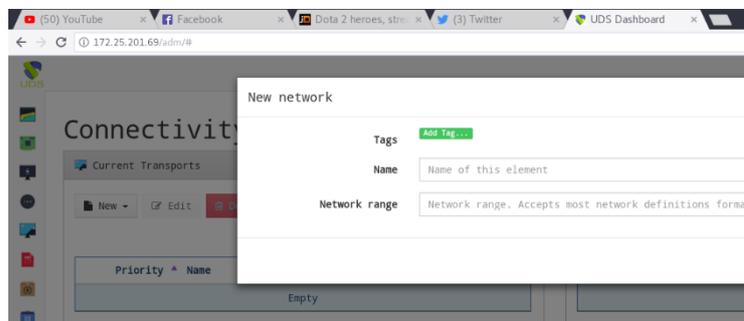


Figura 23-3: conectividad

Realizado por: Huera, B., 2017.

Agregar un nuevo protocolo de utilización ya sea HTML5 RDP Transporte o SPICE, colocando la IP del servidor guacamole, habilitando el audio, archivo de impresión, acceso web.

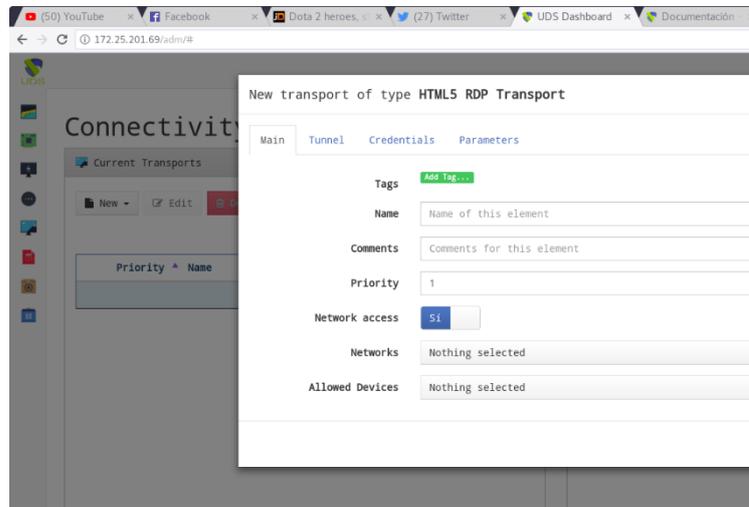


Figura 24-3: Activación de html5 o Spice

Realizado por: Huera, B., 2017.

CAPÍTULO IV

4. RESULTADOS Y DISCUSIÓN

La utilización del prototipo se lo realiza en los laboratorios en la Escuela Superior Politécnica de Chimborazo, Facultad de Informática y Electrónica, con la ayuda de un laboratorio, con estudiantes de la escuela de ingeniería en sistemas, dando como resultado los datos obtenidos para posteriormente ser analizados, como se observa en la figura 1-4.



Figura 1-4: Laboratorio de pruebas Informática 4 FIE -ESPOCH

Realizado por: Huera, B., 2017.

En la figura 2-4 .se observa la utilización de la plataforma VDI, con sus diferentes cuentas de usuario asignadas.



Figura 2-4: Laboratorio de pruebas con estudiantes, Informática 4 FIE -ESPOCH

Realizado por: Huera, B., 2017.

En la figura 3-4 Se observa la utilización de la Máquina virtual Asignada.



Figura 3-4: Utilización de la máquina Virtual, Informática 4 FIE -ESPOCH

Realizado por: Huera, B., 2017.

Para analizar los datos de transmisión de la información mediante los protocolos se plantearon los siguientes escenarios:

4.1 Escenarios del Prototipo VDI

4.1.1 Escenario Uno Protocolo HTML5

En este escenario como primer punto se analizó la utilización de los recursos del servidor mediante un usuario utilizando la herramienta Ovirt node para visualizar y monitorear los recursos consumidos.

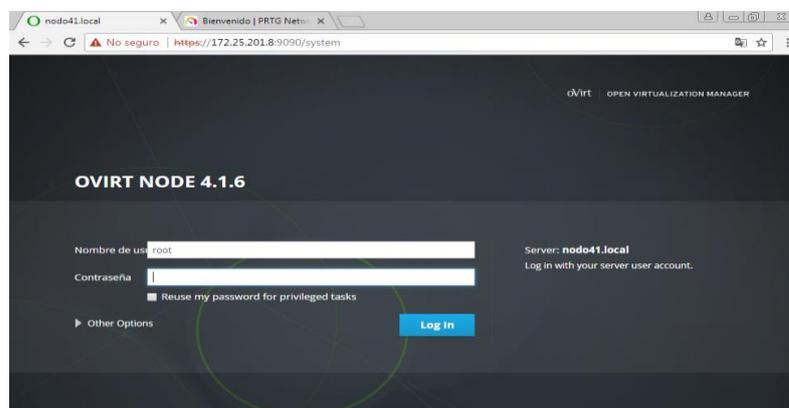


Figura 4-4: Utilización de la herramienta OVIRT NODE 4.1.6, en el servidor de la plataforma

Realizado por: Huera, B., 2017.

La figura 5-4 muestra la gráfica de un resumen general de la utilización de los recursos en la plataforma VDI.

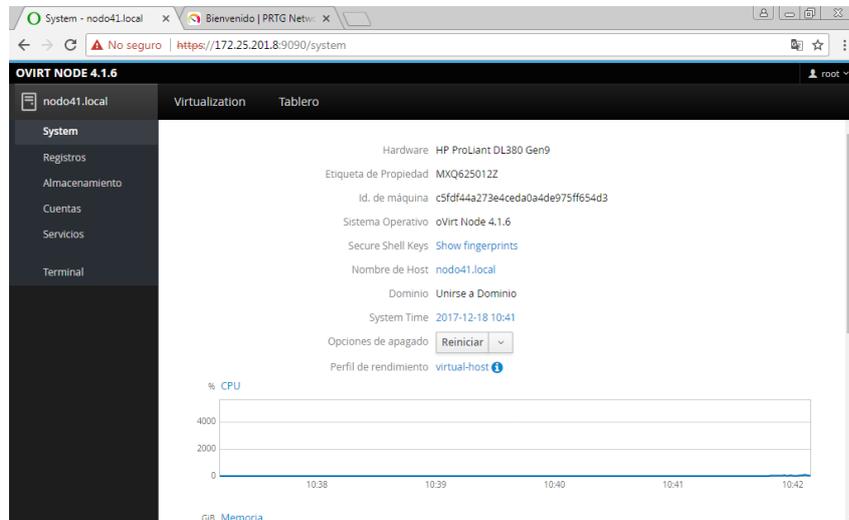


Figura 5-4: Resumen general, en el servidor de la plataforma
Realizado por: Huera, B., 2017.

Dentro del primer punto se verifica el uso de procesador, en el servidor de VDI, donde el dato mínimo en porcentaje es 50 % y el máximo es 250 %, nota esta herramienta analiza el procesador en porcentajes grandes hasta de 5000 % y con la cual se procede al análisis correspondiente de esta manera.

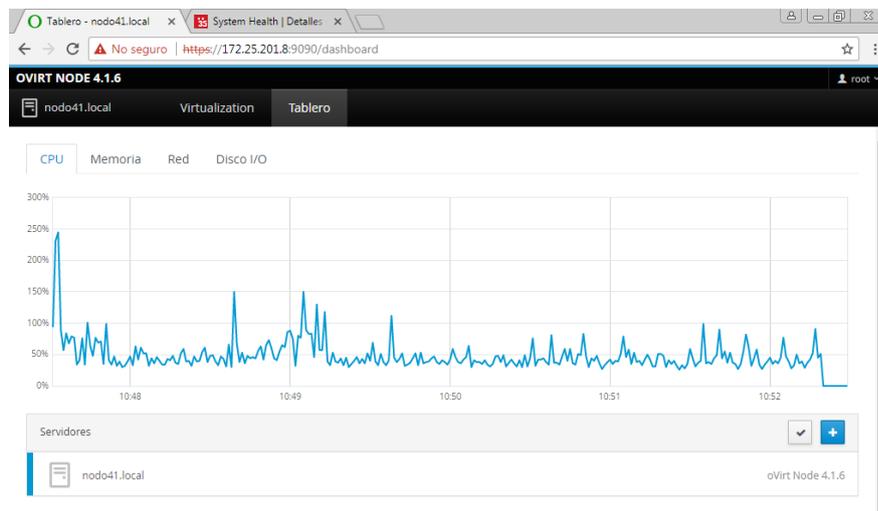


Figura 6-4: Utilización del CPU, en el servidor de la plataforma
Realizado por: Huera, B., 2017.

En el segundo punto se verifica el uso de la memoria RAM, en el servidor de VDI, donde el dato mínimo en uso es 0 GB y cuando el máximo es de 12.5 GB ya que la primera máquina Virtual asignada es de 8 GB, el servidor posee 64 GB de RAM física.

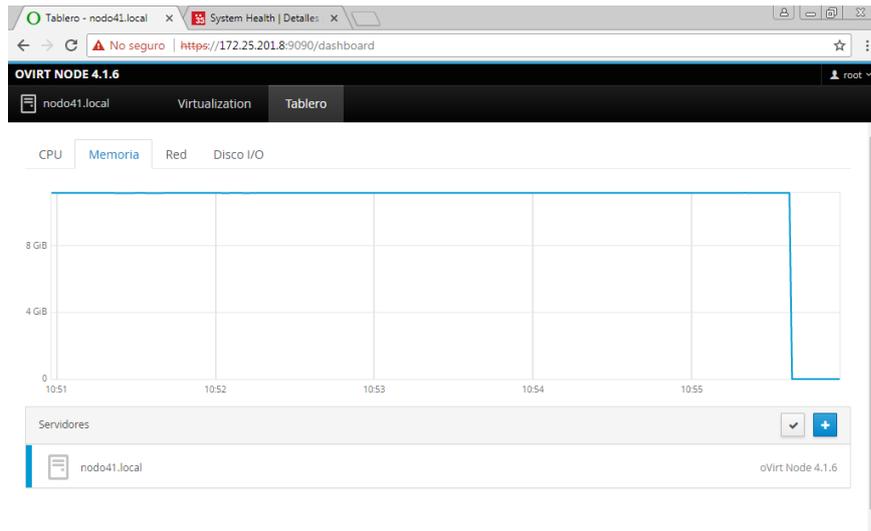


Figura 7-4: Utilización de la memoria RAM, en el servidor de la plataforma

Realizado por: Huera, B., 2017.

En la figura 8-4 se observa la utilización detallada de los recursos de la RAM, con estado libre estado usado, cache del sistema.



Figura 8-4: Utilización de la memoria RAM, en el servidor de la plataforma

Realizado por: Huera, B., 2017.

En el tercer punto la utilización del disco Duro en el servidor de VDI, donde el dato mínimo en uso es 5 Mb/s y cuando el máximo es de 24 Mb/s, ya que la primera máquina Virtual asignada el espacio en disco de 50 GB es espacio de almacenamiento.

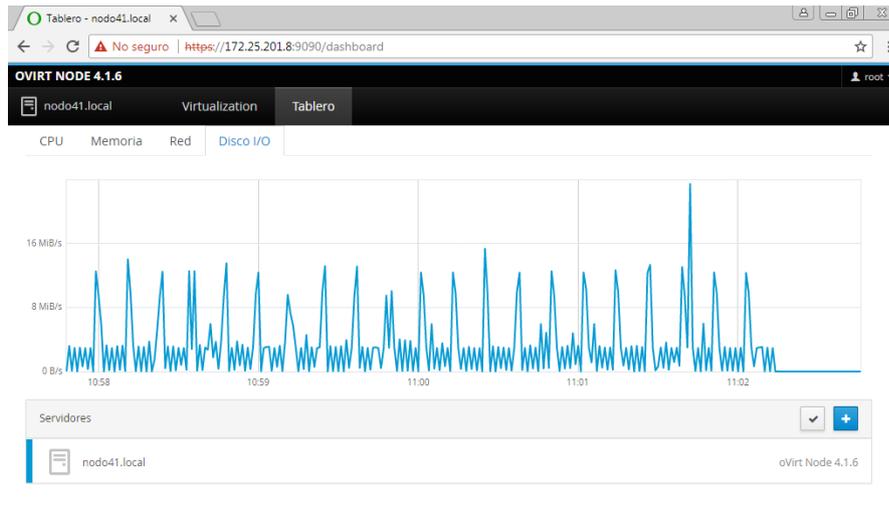


Figura 9-4: Utilización del Disco Duro, en el servidor de la plataforma
Realizado por: Huera, B., 2017.

Como cuarto punto la utilización del ancho de banda en el servidor de VDI, donde el dato mínimo en uso es 20 Mbps y cuando el máximo es de 92 Mbps, por la utilización de una sola máquina virtual.

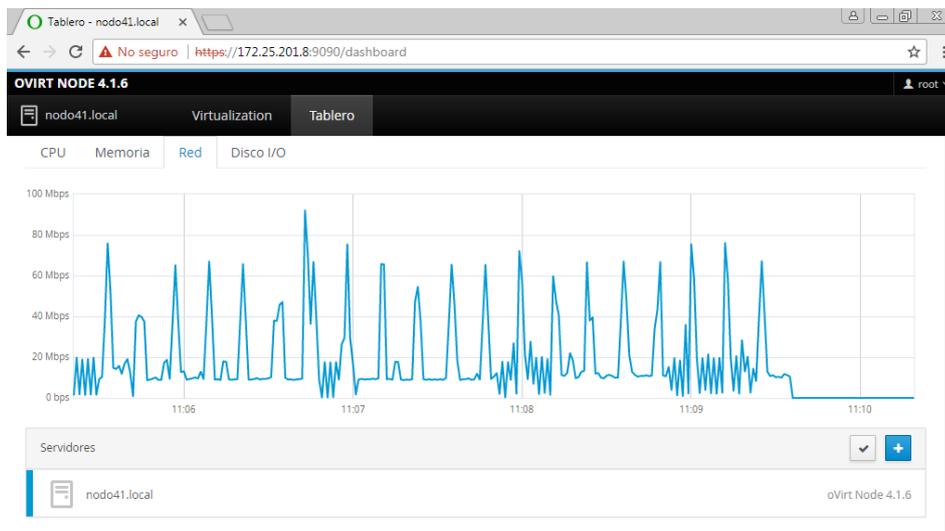


Figura 10-4: Utilización del ancho de banda, en el servidor de la plataforma VDI
Realizado por: Huera, B., 2017.

En este escenario se analiza la utilización de 8 usuarios con las diferentes máquinas virtuales asignadas por el mediante un usuario utilizando la herramienta Ovirt node para visualizar y monitorear los recursos consumidos.

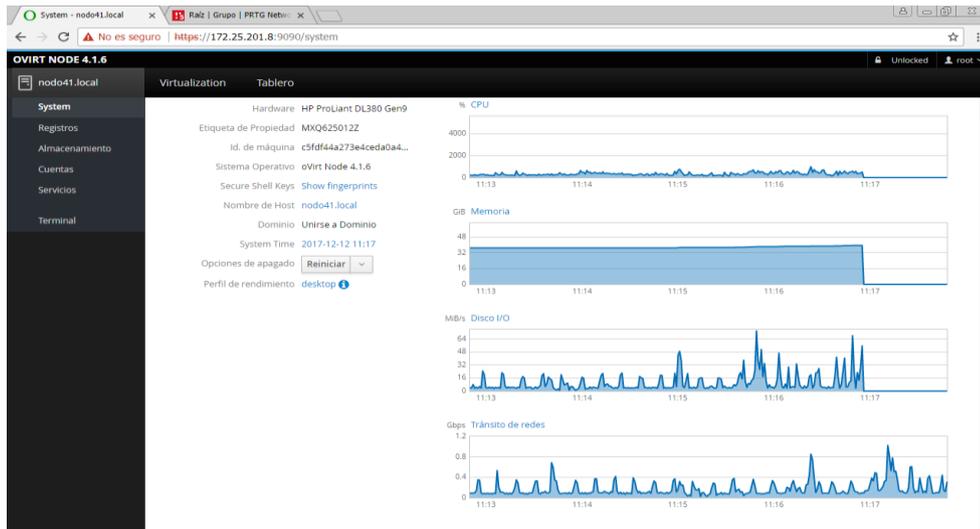


Figura 11-4: Utilización del ancho de banda, en el servidor de la plataforma VDI.

Realizado por: Huera, B., 2017.

En este punto verificamos el uso de procesador, en el servidor de VDI, donde el dato mínimo en porcentaje es 200 % y el máximo es 850 %.

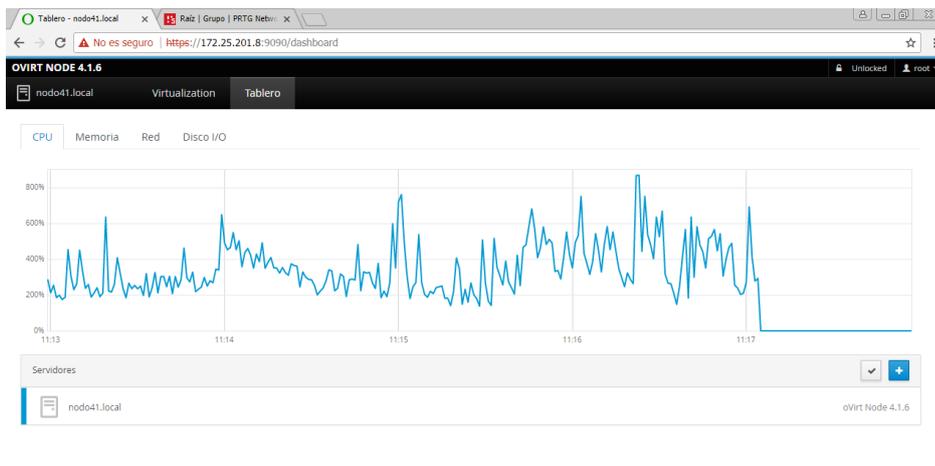


Figura 12-4: Utilización del ancho de banda 8 usuarios activos, en el servidor de la plataforma VDI

Realizado por: Huera, B., 2017.

Segundo punto se analiza el uso de la memoria RAM, en el servidor de VDI, donde el dato mínimo en uso es 33 GB y cuando el máximo es de 36 GB ya que la primera máquina Virtual asignada es de 8 GB, el resto de las máquinas 2 GB física y 1 GB de RAM garantizada.

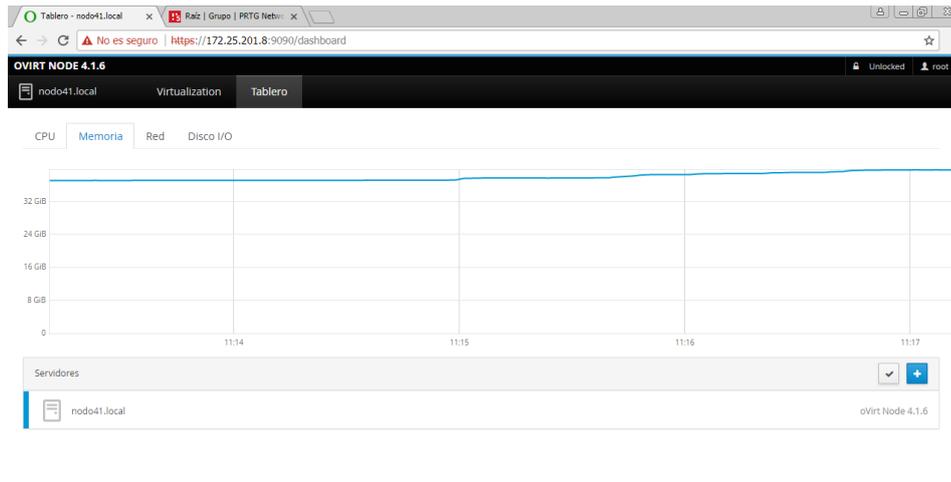


Figura 13-4: Utilización de la memoria RAM por 8 usuarios, en el servidor de la plataforma

Realizado por: Huera, B., 2017.

Como tercer punto la utilización del disco Duro en el servidor de VDI con medio de 8GB, donde el dato mínimo en uso es 64 Mb/s y cuando el máximo es de 180 Mb/s.

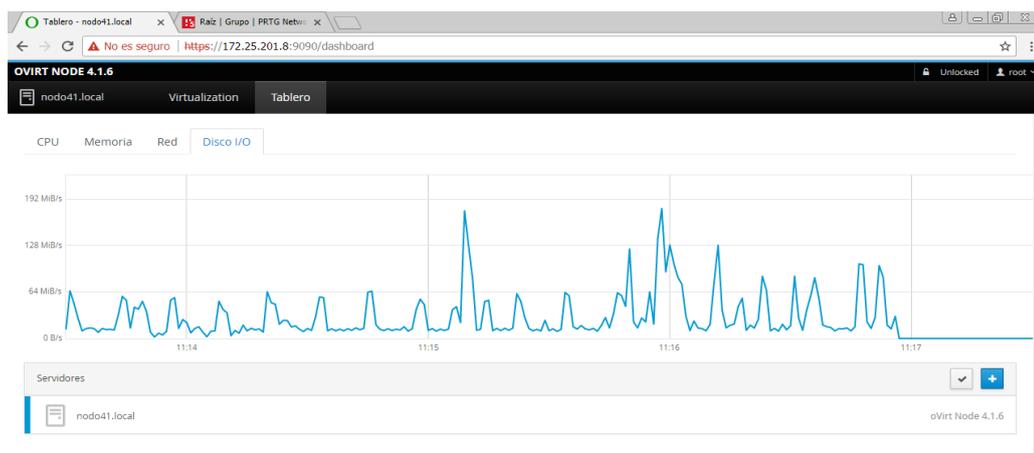


Figura 14-4: Utilización del disco Duro por 8 usuarios, en el servidor de la plataforma

Realizado por: Huera, B., 2017.

En cuarto punto se verifica la utilización del ancho de banda en el servidor de VDI con 8 usuarios, donde el dato mínimo en uso es 200 Mbps y cuando el máximo es de 1010 Mbps, por la utilización de una sola máquina virtual.

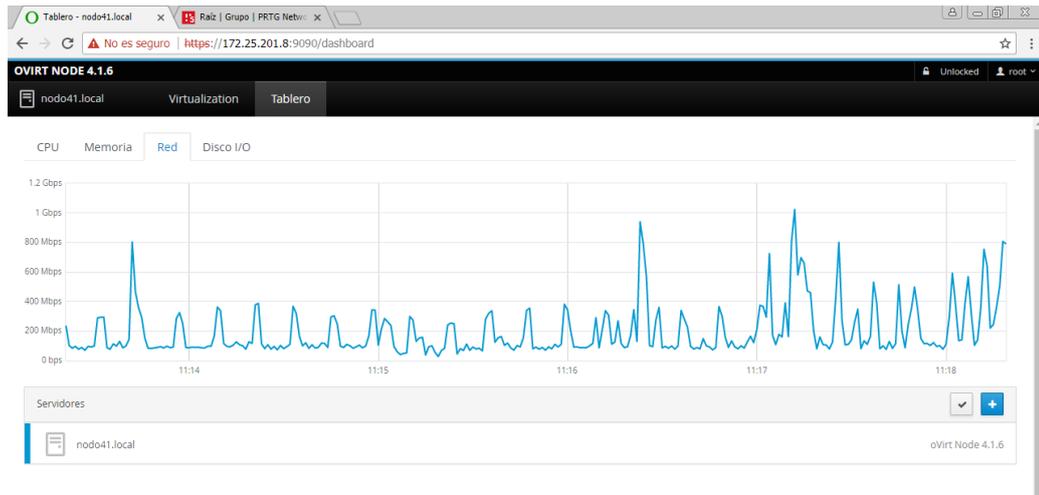


Figura 15-4: Utilización del ancho de banda por 8 usuarios, en el servidor de la plataforma

Realizado por: Huera, B., 2017.

En este escenario se analiza la utilización de 16 usuarios con las diferentes máquinas virtuales asignadas por el mediante un usuario utilizando la herramienta Ovirt node para visualizar y monitorear los recursos consumidos.

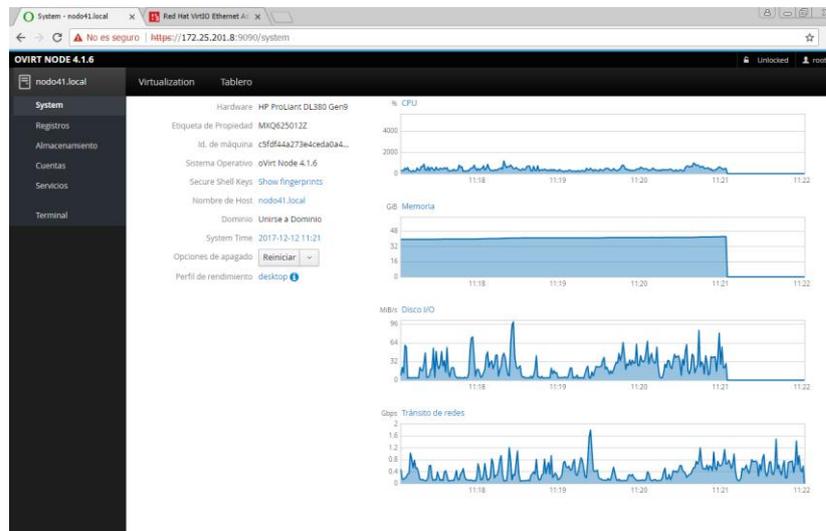


Figura 16-4: Resumen de la utilización de los recursos, en el servidor de la plataforma.

Realizado por: Huera, B., 2017.

En este punto verificamos el uso de procesador, en el servidor de VDI, donde el dato mínimo en porcentaje es 450 % y el máximo es 1000 %.

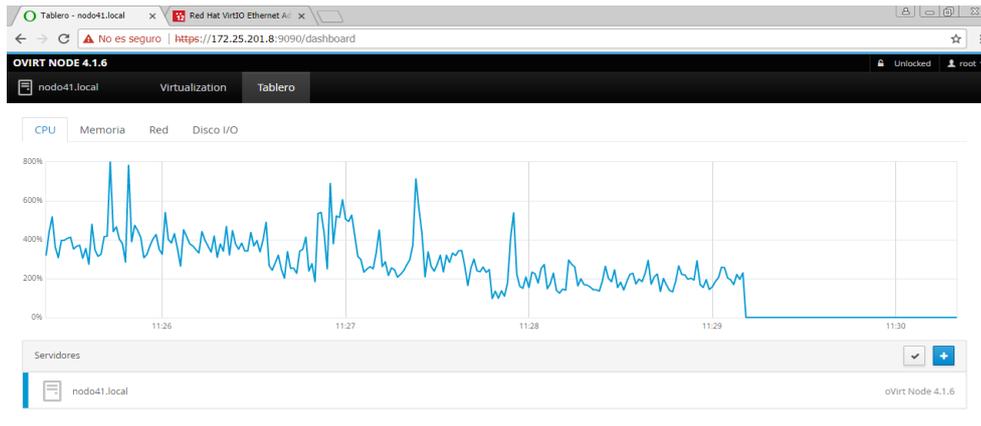


Figura 17-4: Utilización del CPU por 16 usuarios, en el servidor de la plataforma
Realizado por: Huera, B., 2017.

Se verifica en el segundo punto el uso de la memoria RAM, en el servidor de VDI, donde el dato mínimo en uso es 36 GB y cuando el máximo es de 44 GB.

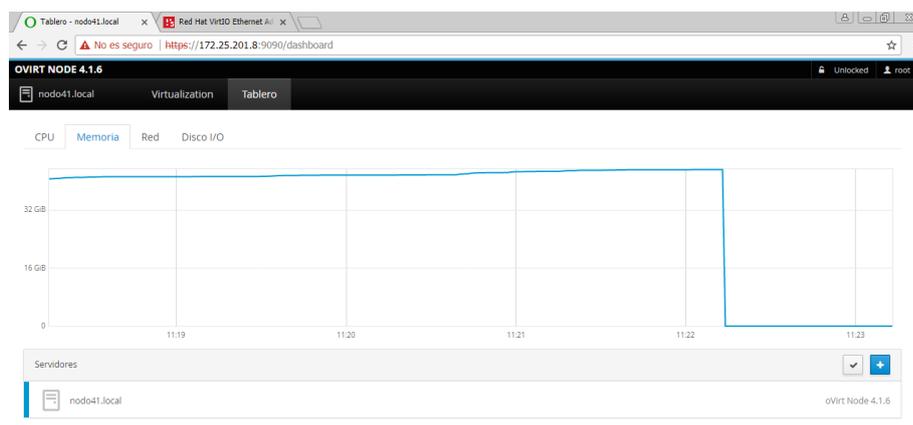


Figura 18-4: Utilización de la memoria RAM por 16 usuarios, en el
 servidor de la plataforma
Realizado por: Huera, B., 2017.

Como tercer punto, se analiza la utilización del disco Duro por parte del servidor de VDI, dando como resultado la obtención de un dato mínimo en uso que es 128 Mb/s y cuando el máximo es de 270 Mb/s.

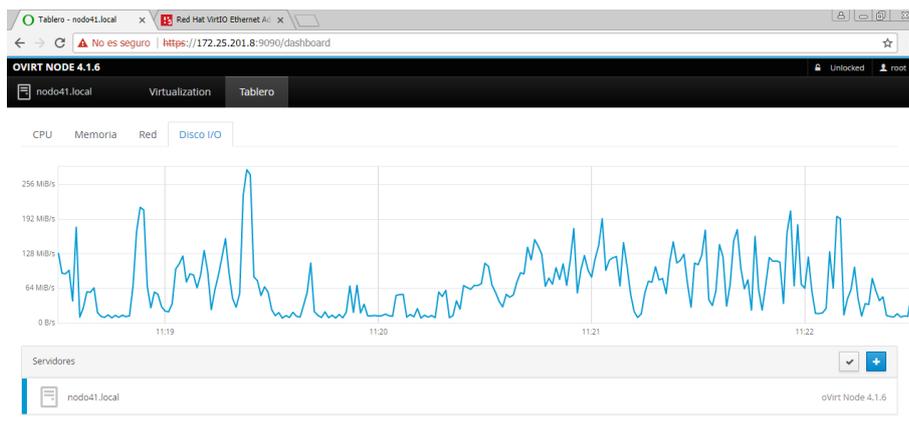


Figura 19-4: Utilización del disco Duro por 16 usuarios, en el servidor de la plataforma VDI

Realizado por: Huera, B., 2017.

En cuarto punto se verifica la utilización del ancho de banda en el servidor de VDI con 16 usuarios, donde el dato mínimo en uso es 1200 Mbps y el máximo es de 1780 Mbps, por la utilización de una sola máquina virtual.

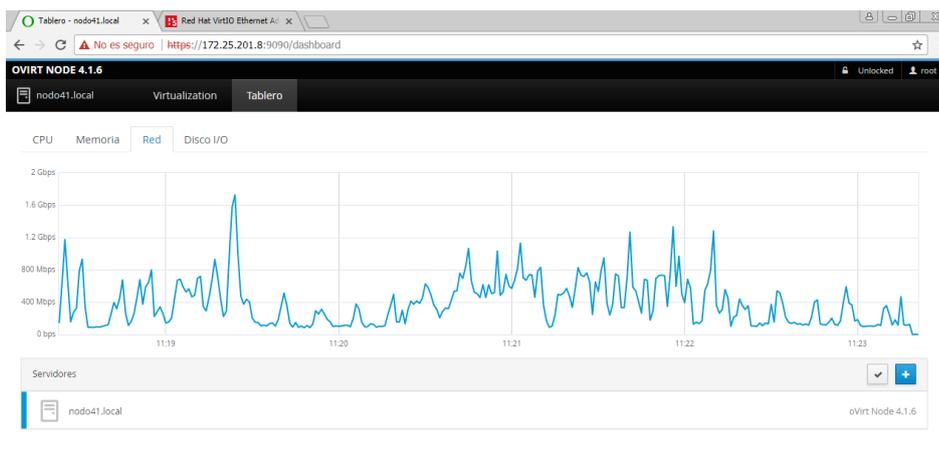


Figura 20-4: Utilización del ancho de banda por 16 usuarios, en el servidor de la plataforma VDI

Realizado por: Huera, B., 2017.

En este escenario se analiza la utilización de 32 usuarios, con las diferentes máquinas virtuales asignadas por el mediante un usuario utilizando la herramienta Ovirt node para visualizar y monitorear los recursos consumidos.

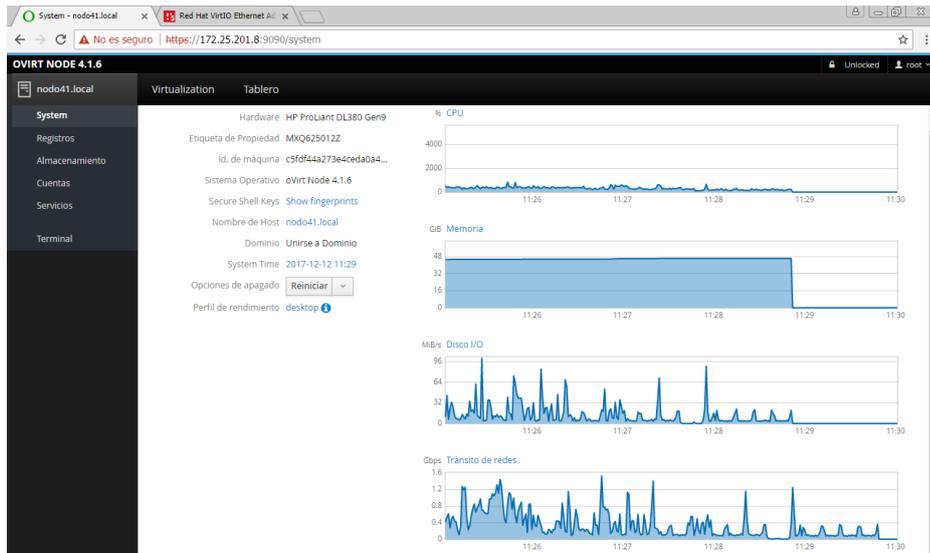


Figura 21-4: Resumen de la utilización de los recursos 32 usuarios, en el servidor de la plataforma

Realizado por: Huera, B., 2017.

En este punto verificamos el uso de procesador, en el servidor de VDI, donde el dato mínimo en porcentaje es 500 % y el máximo es 1250 %.

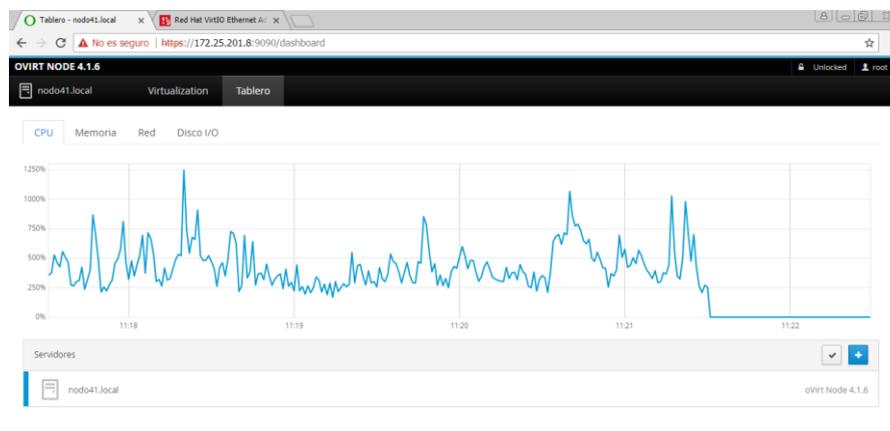


Figura 22-4: Utilización del CPU por 32 usuarios, en el servidor de la plataforma

Realizado por: Huera, B., 2017.

En el segundo punto se verifica el uso de la memoria RAM, en el servidor de VDI, donde el dato mínimo en uso es 40 GB y como máximo es de 48 GB.

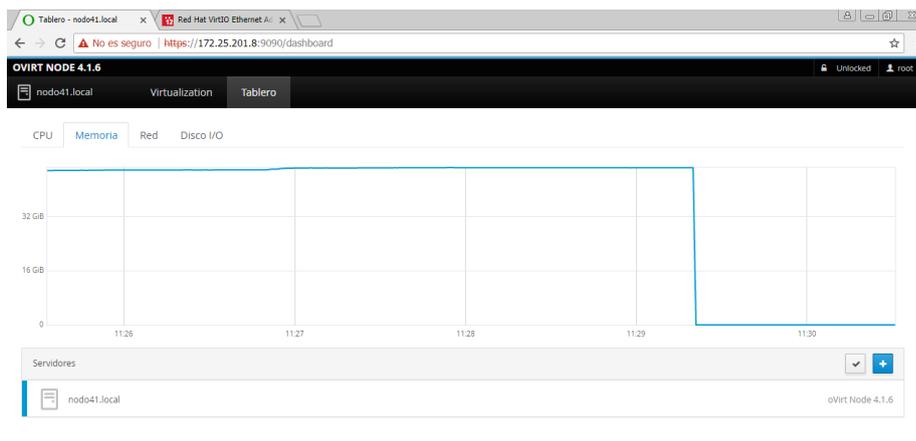


Figura 23-4: Utilización de la memoria RAM por 32 usuarios, en el servidor de la plataforma

Realizado por: Huera, B., 2017.

Se analiza en el tercer punto la utilización del disco Duro en el servidor de VDI, donde el dato mínimo en uso es 192 Mb/s y máximo es de 280 Mb/s.

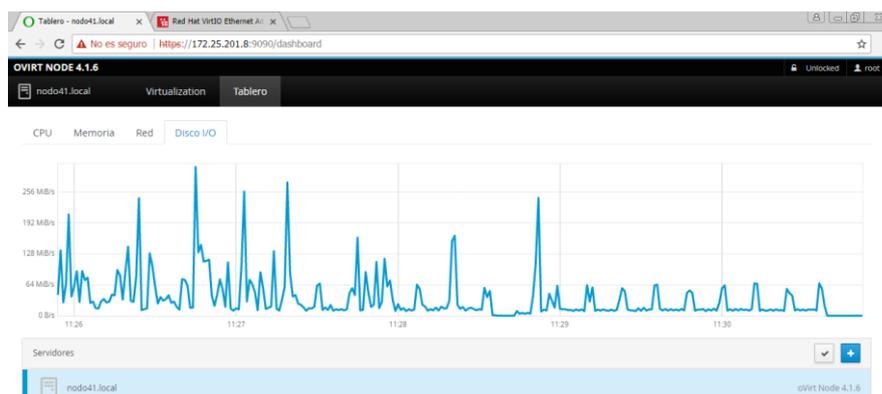


Figura 24-4: Utilización de tasa de utilización del disco duro por 32 usuarios, en el servidor de la plataforma

Realizado por: Huera, B., 2017.

Como cuarto punto se verifico la utilización del ancho de banda en el servidor de VDI con 16 usuarios, donde el dato mínimo en uso es 1300 Mbps y cuando el máximo es de 1680 Mbps, por la utilización de una sola máquina virtual.

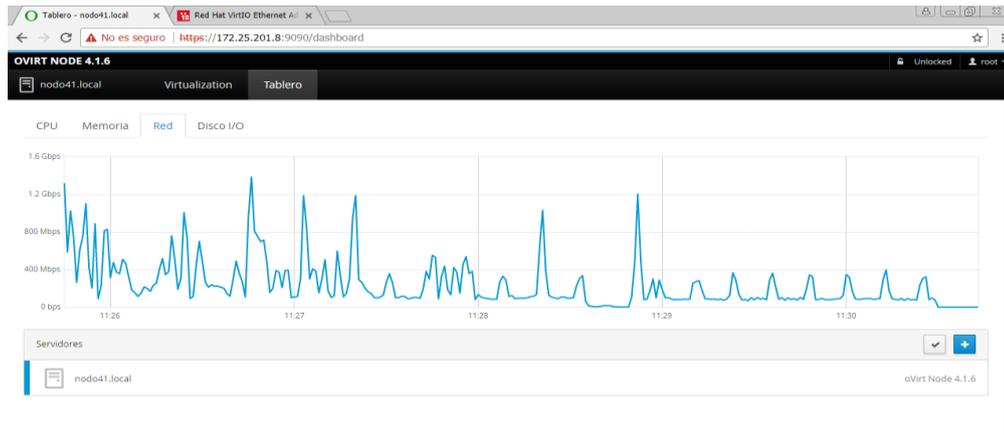


Figura 25-4: Utilización del ancho de banda 32 usuarios, en el servidor de la plataforma

Realizado por: Huera, B., 2017.

4.1.2 Escenario Dos Protocolo SPICE

En este escenario se analiza los diferentes componentes tales como CPU, memoria, procesador y el punto principal el ancho de banda utilizado en el servidor mediante un usuario utilizando la herramienta Ovirt node para visualizar y monitorear los recursos consumidos mediante el uso del protocolo SPICE.

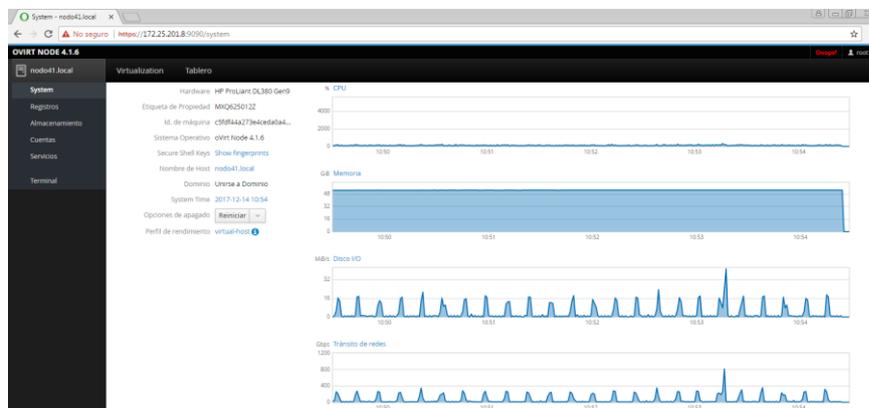


Figura 26-4: Resumen general, en el servidor de la plataforma SPICE

Realizado por: Huera, B., 2017.

En el primer punto se analiza el uso de procesador, en el servidor de VDI, donde el dato mínimo en porcentaje es 150 % y el máximo es 300 %, nota esta herramienta analiza el procesador en porcentajes.

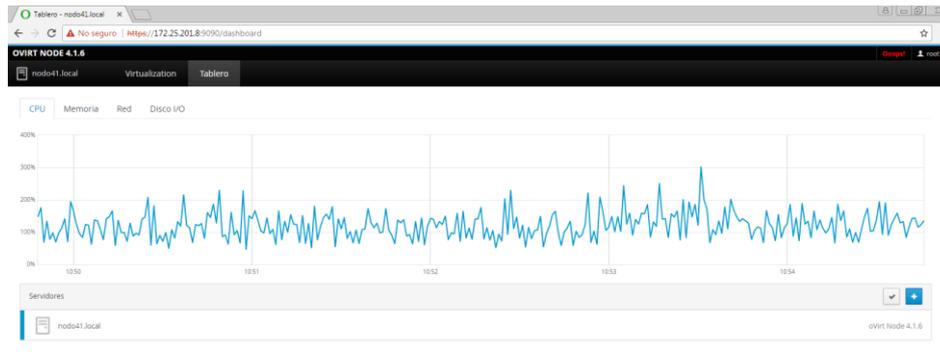


Figura 27-4: Utilización del CPU, en el servidor de la plataforma SPICE

Realizado por: Huera, B., 2017.

Como segundo punto verificamos el uso de la memoria RAM, en el servidor de VDI, donde el dato mínimo en uso es 1GB y máximo es de 11.5 GB ya que la primera máquina Virtual asignada es de 8GB, el servidor posee 64gb de RAM física.

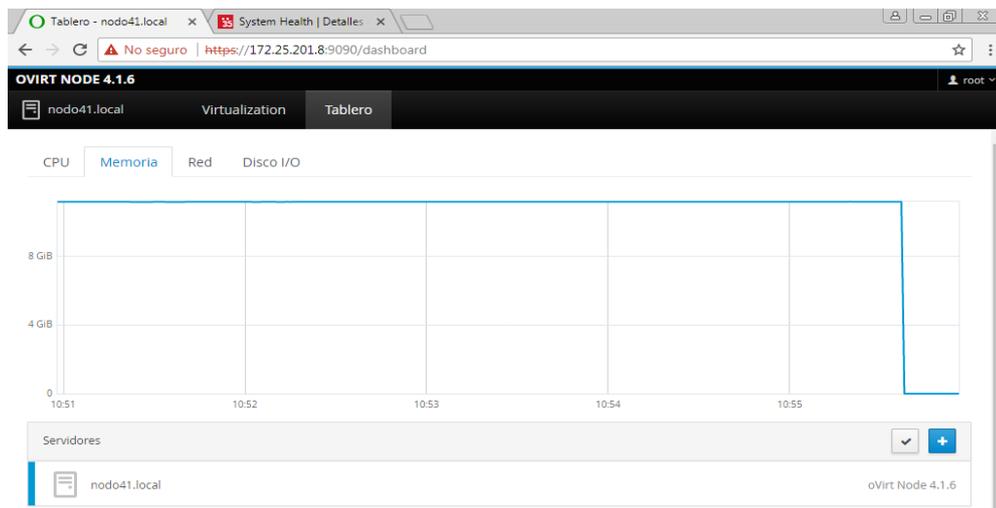


Figura 28-4: Utilización de la memoria RAM, en el servidor de la plataforma SPICE

Realizado por: Huera, B., 2017.

En la figura 28-4 se observa la utilización detallada de los recursos de la RAM, con estado libre estado usado, cache del sistema.

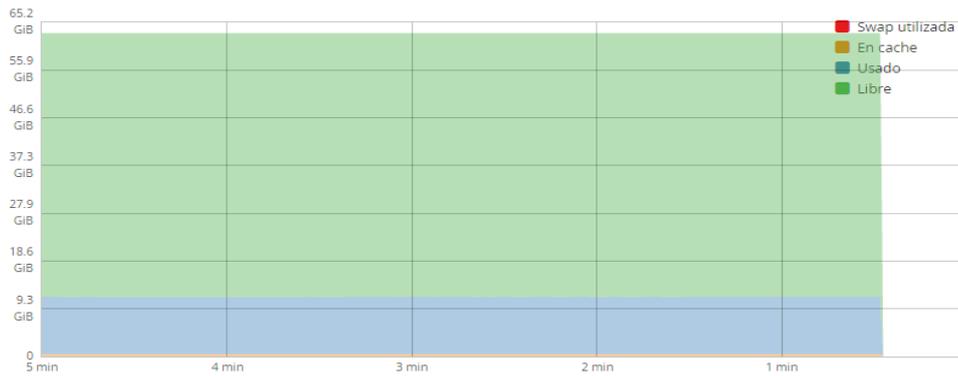


Figura 29-4: Utilización de la memoria RAM, en el servidor de la plataforma SPICE

Realizado por: Huera, B., 2017.

El análisis de la utilización del disco Duro en el servidor de VDI en el tercer punto, da como resultado un dato mínimo de uso que es 3 Mb/s y máximo uso es de 50 Mb/s, ya que la primera máquina Virtual asignada el espacio en disco de 50GB es espacio de almacenamiento.

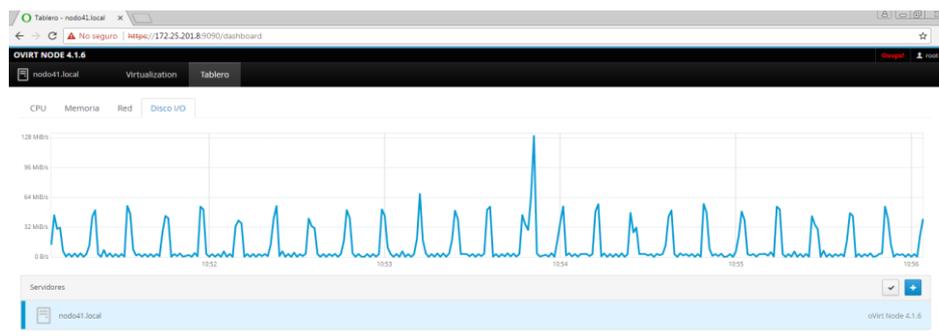


Figura 30-4: Utilización del Disco Duro, en el servidor de la plataforma SPICE

Realizado por: Huera, B., 2017.

En el cuarto ítem se verifica la utilización del ancho de banda en el servidor de VDI, donde el dato mínimo en uso es 10 Mbps y el máximo es de 170 Mbps.

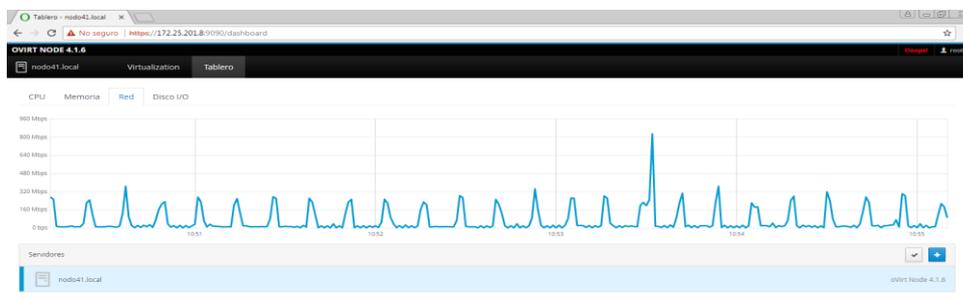


Figura 31-4: Utilización del ancho de banda, en el servidor de la plataforma VDI
SPICE

Realizado por: Huera, B., 2017.

En este escenario se analiza la utilización de 8 usuarios con las diferentes máquinas virtuales asignadas utilizando la herramienta Ovirt node para visualizar y monitorear los recursos consumidos.

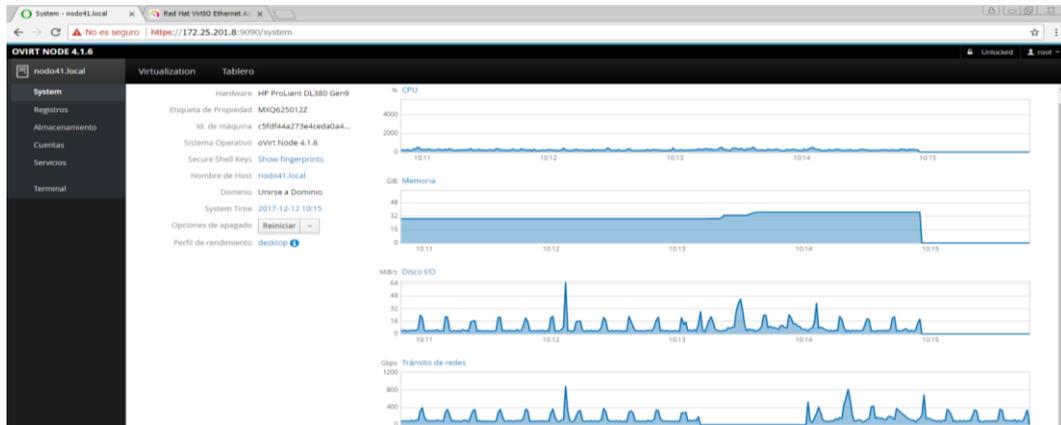


Figura 32-4: Utilización del ancho de banda, en el servidor de la plataforma VDI SPICE

Realizado por: Huera, B., 2017.

En este punto se verifica el uso de procesador, en el servidor de VDI, donde el dato mínimo en porcentaje es 210% y el máximo es 490%,

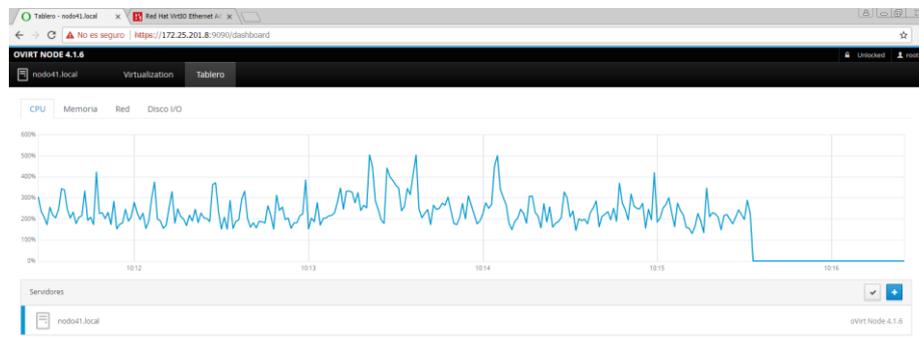


Figura 33-4: Utilización del ancho de banda 8 usuarios activos, en el servidor de la plataforma VDI SPICE

Realizado por: Huera, B., 2017.

Como segundo ítem se analiza el uso de la memoria RAM, en el servidor de VDI, donde el dato mínimo en uso es 28GB y cuando el máximo es de 35 GB ya que la primera máquina Virtual asignada es de 8 GB, el resto de las máquinas 2 GB física y 1 GB de RAM garantizada.

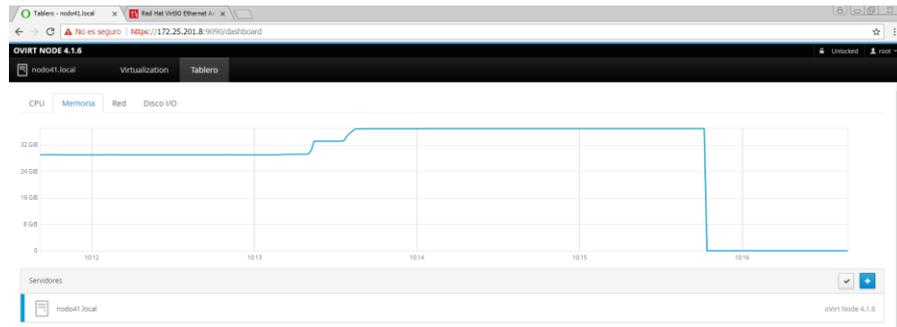


Figura 34-4: Utilización de la memoria RAM por 8 usuarios, en el servidor de la plataforma SPICE

Realizado por: Huera, B., 2017.

El tercer punto se verifica el rendimiento del disco Duro en el servidor de VDI, donde el dato mínimo en uso es 96 Mb/s y cuando el máximo es de 150 Mb/s.

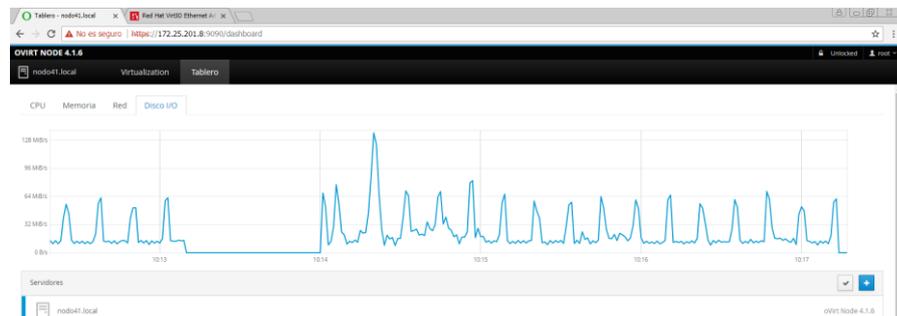


Figura 35-4: Utilización del disco Duro por 8 usuarios, en el servidor de la plataforma SPICE

Realizado por: Huera, B., 2017.

En la cuarta sección se analiza el rendimiento del ancho de banda en el servidor de VDI con 8 usuarios, donde el dato mínimo en uso es 300 Mbps y cuando el máximo es de 960 Mbps

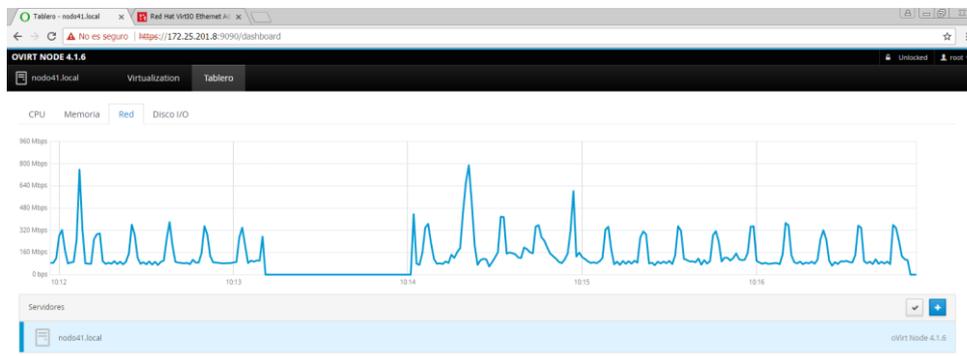


Figura 36-4: Utilización del ancho de banda por 8 usuarios, en el servidor de la plataforma SPICE

Realizado por: Huera, B., 2017.

En este escenario se analiza la utilización de 16 usuarios con las diferentes máquinas virtuales asignadas, utilizando la herramienta Ovirt node para visualizar y monitorear los recursos consumidos.

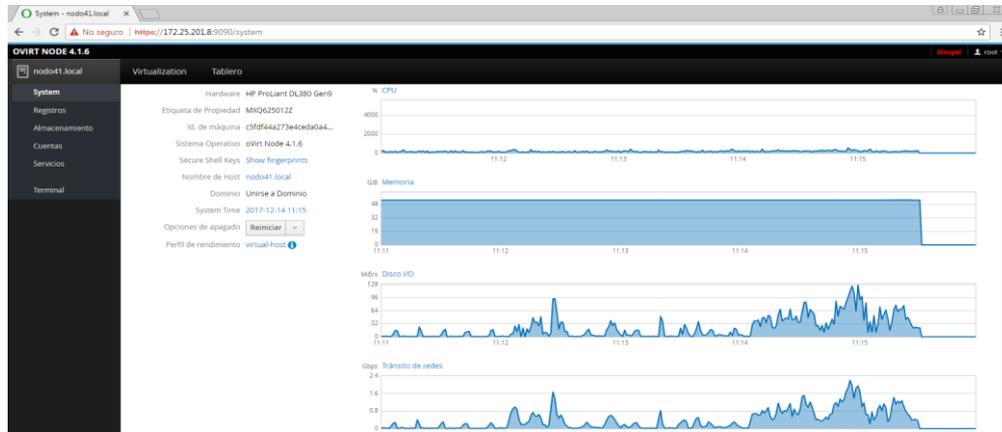


Figura 37-4: Resumen de la utilización de los recursos, en el servidor de la plataforma SPICE

Realizado por: Huera, B., 2017.

En este punto se analiza el uso de procesador, en el servidor de VDI, donde el dato mínimo en porcentaje es 200% y el máximo es 560%.



Figura 38-4: Utilización del CPU por 16 usuarios, en el servidor de la plataforma

Realizado por: Huera, B., 2017.

Para el segundo ítem se verifica el uso de la memoria RAM, en el servidor de VDI, donde el dato mínimo en uso es 36 GB y cuando el máximo es de 50 GB.

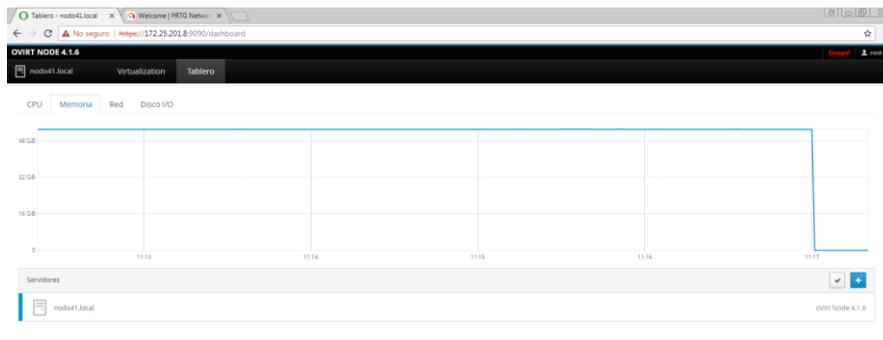


Figura 39-4: Utilización de la memoria RAM por 16 usuarios, en el servidor de la plataforma

Realizado por: Huera, B., 2017.

También en la tercera sección se analiza la utilización del disco Duro en el servidor de VDI, donde el dato mínimo en uso es 128 Mb/s y el máximo es de 350 Mb/s.

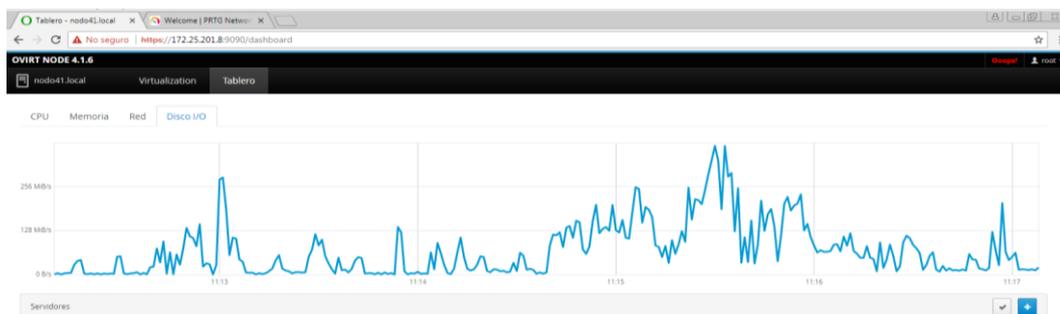


Figura 40-4: Utilización del disco Duro por 16 usuarios, en el servidor de la plataforma VDI

Realizado por: Huera, B., 2017.

En el cuarto punto se verifica la utilización del ancho de banda en el servidor de VDI con 32 usuarios, donde el dato mínimo en uso es 1600 Mbps y cuando el máximo es de 1900 Mbps.

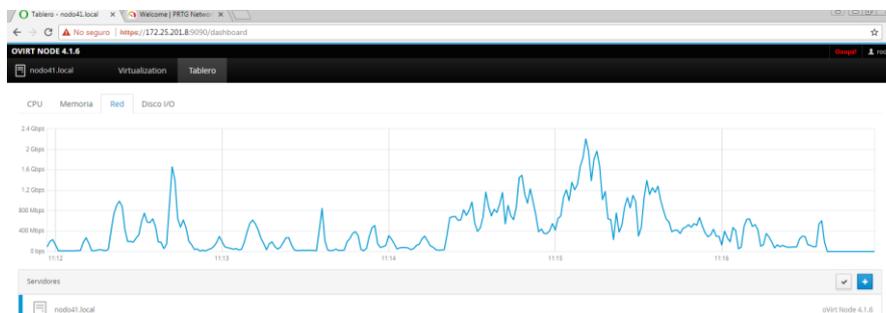


Figura 41-4: Utilización del ancho de banda por 16 usuarios, en el servidor de la plataforma VDI

Realizado por: Huera, B., 2017.

En este escenario se analiza la utilización de 32 usuarios, con las diferentes máquinas virtuales asignadas utilizando la herramienta Ovirt node para visualizar y monitorear los recursos consumidos.

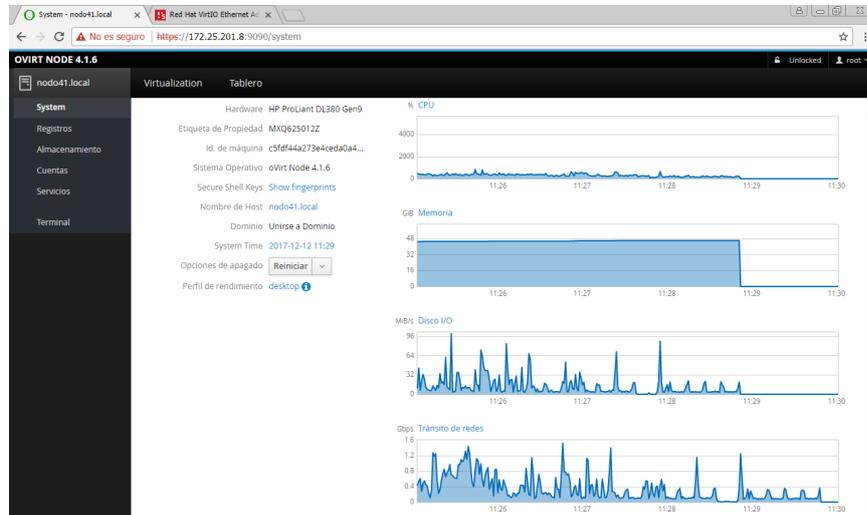


Figura 42-4: Resumen de la utilización de los recursos 32 usuarios, en el servidor de la plataforma.

Realizado por: Huera, B., 2017.

En este punto se obtiene el uso de procesador, en el servidor de VDI, donde el dato mínimo en porcentaje es 500% y el máximo es 1230%.

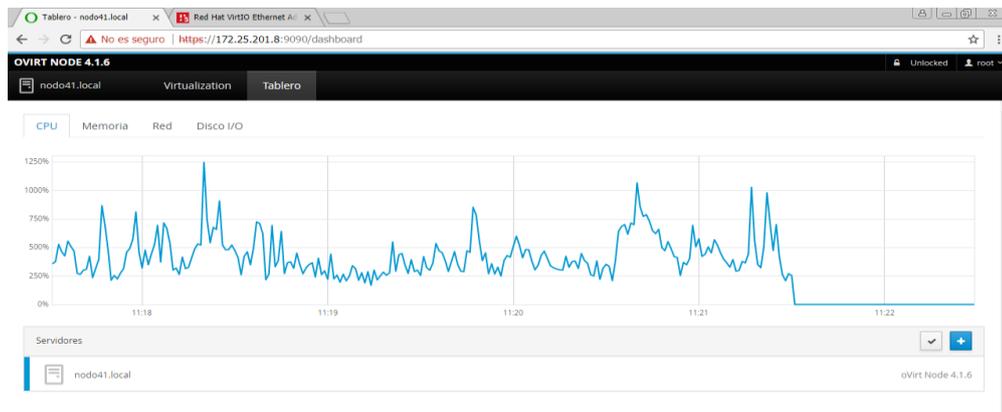


Figura 43-4: Utilización del CPU por 32 usuarios, en el servidor de la plataforma
Realizado por: Huera, B., 2017.

Para el segundo ítem se verifica el uso de la memoria RAM, en el servidor de VDI, donde el dato mínimo en uso es 40 GB y cuando el máximo es de 55 GB.



Figura 44-4: Utilización de la memoria RAM por 32 usuarios, en el servidor de la plataforma

Realizado por: Huera, B., 2017.

También el tercer punto arroja como resultado de la utilización del disco Duro en el servidor de VDI, donde los datos mínimos son 192 Mb/s y el máximo es de 340 Mb/s.

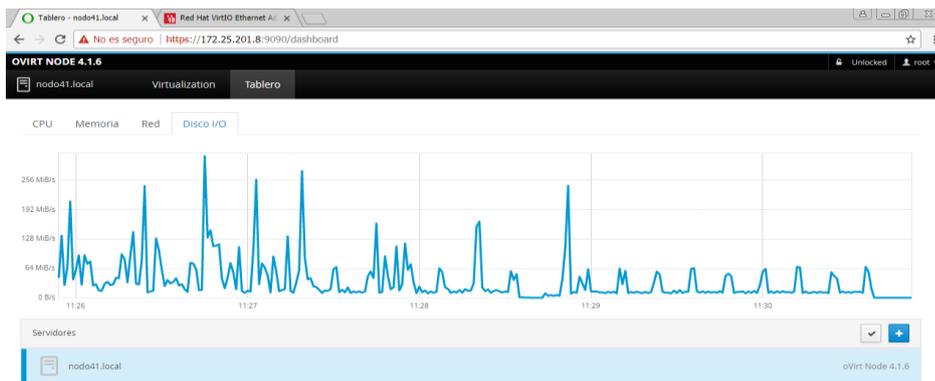


Figura 45-4: Utilización de tasa de utilización del disco duro por 32 usuarios, en el servidor de la plataforma

Realizado por: Huera, B., 2017.

En el cuarto punto la utilización del ancho de banda en el servidor de VDI con 16 usuarios, donde el dato mínimo en uso es 1300 Mbps y cuando el máximo es de 2000 Mbps.

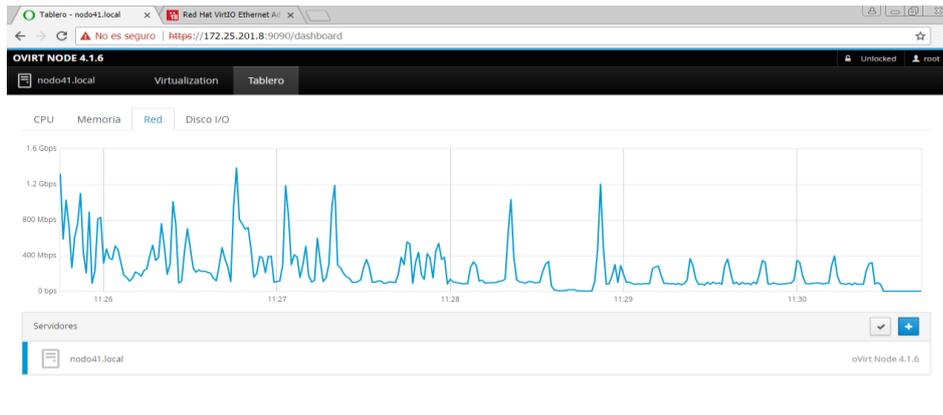


Figura 46-4: Utilización del ancho de *banda* 32 usuarios, en el servidor de la plataforma

Realizado por: Huera, B., 2017.

4.2 Análisis de los resultados para protocolo HTML5

En el siguiente punto se realiza un análisis de cada elemento tomando en cuenta los diferentes componentes utilizados.

Tabla 1-4: Resumen general de protocolo HTML5

	Procesador (%)	Memoria (GB)	Disco duro (Mb/s)	Red /ancho de banda (Mbps)
1 usuario	250	12,5	24	92
8 usuarios	850	36	180	1010
16 usuarios	1000	44	270	1780
32 usuarios	1250	48	280	1680

Realizado por: Huera, B., 2017.

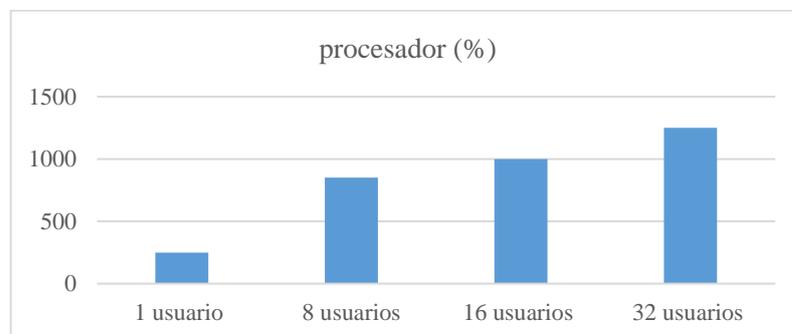


Gráfico 1-4: Resumen general de uso del procesador con HTML5

Realizado por: Huera, B., 2017.

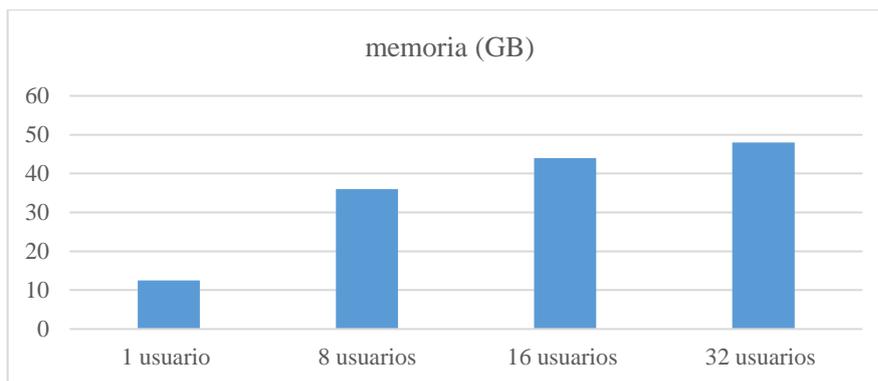


Gráfico 2-4: Resumen general de uso de la memoria RAM con HTML5

Realizado por: Huera, B., 2017.

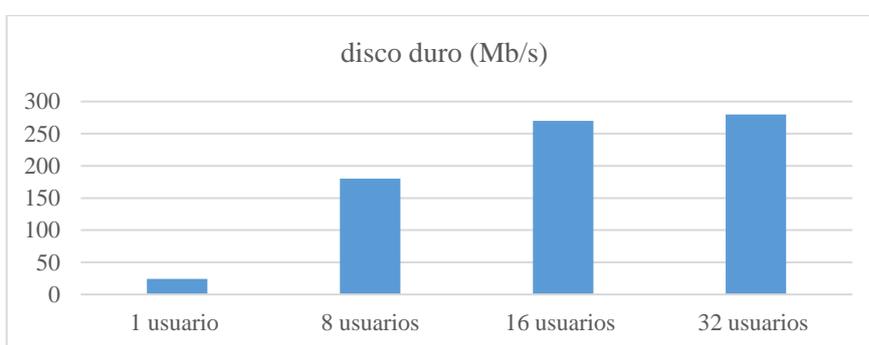


Gráfico 3-4: Resumen general de uso de la memoria RAM con HTML5

Realizado por: Huera, B., 2017.

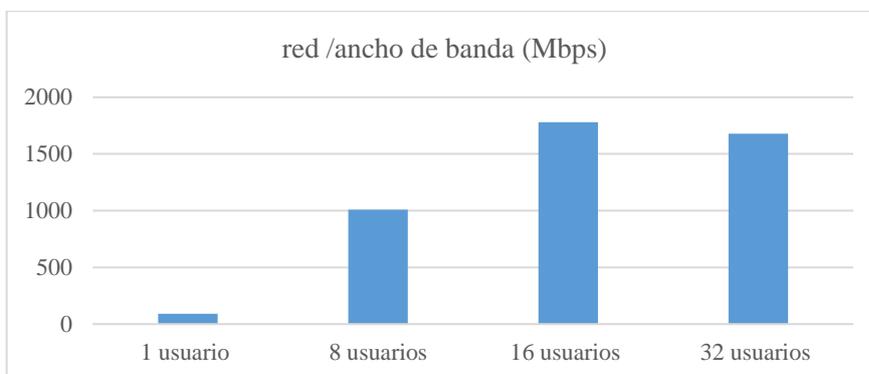


Gráfico 4-4: Resumen general de uso de la memoria RAM con HTML5

Realizado por: Huera, B., 2017.

La grafica general contempla todo análisis detallado del ancho de banda utilizado en la plataforma VDI mediante el protocolo utilizado por OPENUDS con el protocolo HTML5, cual podemos observar un crecimiento progresivo en la implementación de dicha plataforma, pero muy significativa del ancho de banda.

4.3 Análisis de los resultados para el protocolo Spice

En el siguiente punto se realiza un análisis de cada elemento tomando en cuenta los diferentes componentes utilizados.

Tabla 2-4: Resumen general de protocolo SPICE

	Procesador (%)	memoria (GB)	disco duro (Mb/s)	red /ancho de banda (Mbps)
1 usuario	30	11,5	50	170
8 usuarios	490	35	150	960
16 usuarios	560	50	320	1900
32 usuarios	1230	55	340	2000

Realizado por: Huera, B., 2017.

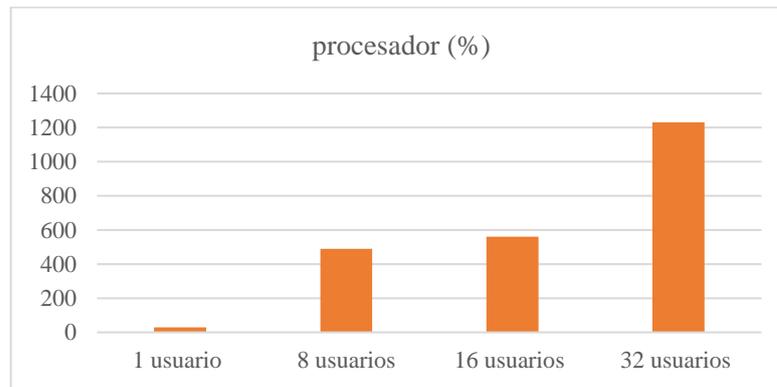


Gráfico 5-4: Resumen general de uso del procesador con SPICE

Realizado por: Huera, B., 2017.

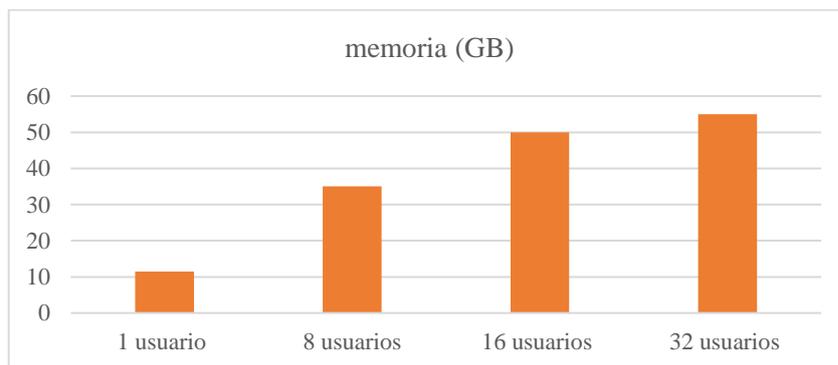


Gráfico 6-4: Resumen general de uso de la memoria RAM con SPICE

Realizado por: Huera, B., 2017.

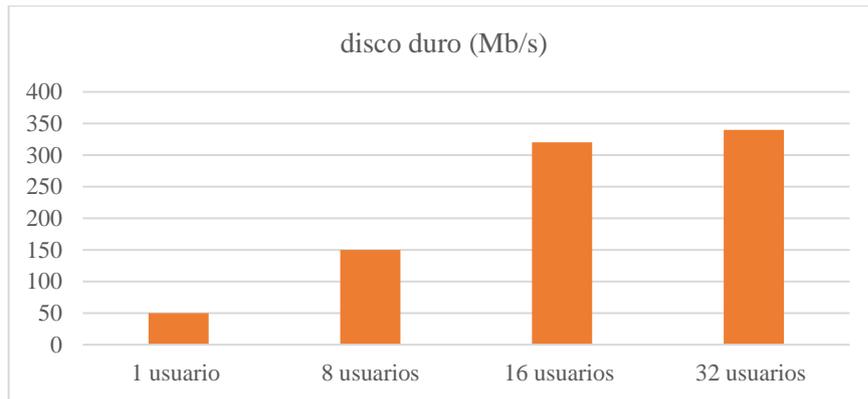


Gráfico 7-4: Resumen general de uso de la memoria RAM con SPICE

Realizado por: Huera, B., 2017.

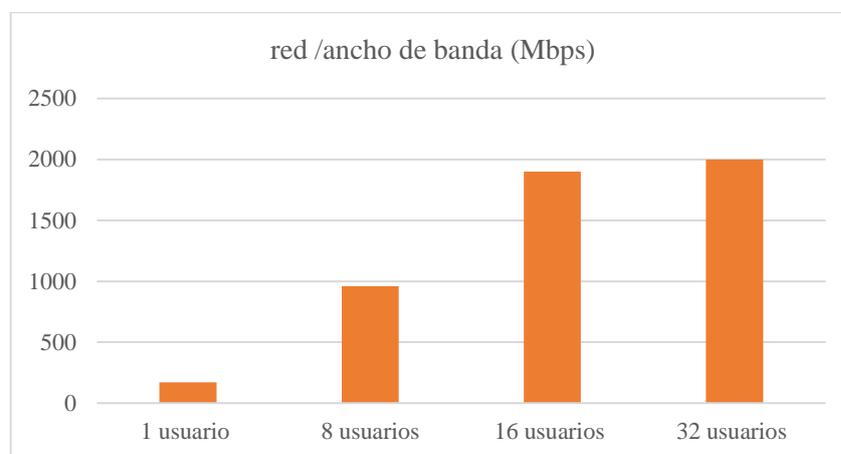


Gráfico 8-4: Resumen general de uso de la memoria RAM con SPICE

Realizado por: Huera, B., 2017.

La grafica general contempla todo análisis detallado del ancho de banda utilizado en la plataforma VDI mediante el protocolo utilizado por OPENUDS con el protocolo SPICE, cual podemos observar un crecimiento progresivo en la implementación de dicha plataforma, pero muy significativa del ancho de banda.

4.4 Análisis comparativo entre HTML5 Y SPICE

Tabla 3-4: Resumen general de protocolo HTML5 - 1

	Procesador (%) SPICE	Procesador (%) HTML5
1 usuario	30	250
8 usuarios	490	850
16 usuarios	560	1000
32 usuarios	1230	1250

Realizado por: Huera, B., 2017.

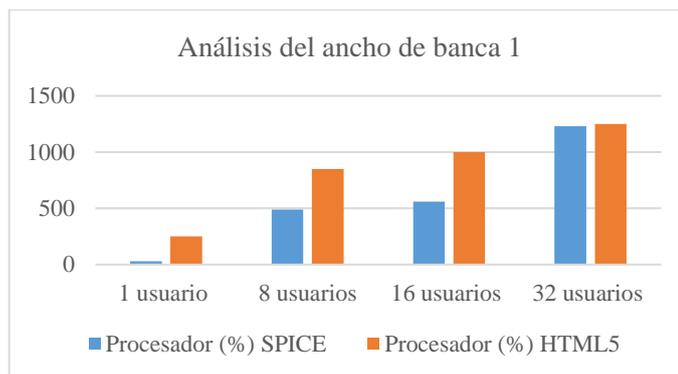


Gráfico 9-4: Análisis del ancho de banca

Realizado por: Huera, B., 2017.

La grafica final se detalla el análisis del ancho de banda utilizado en la plataforma VDI mediante los protocolos OPENUDS con es SPICE y HTML5, ya que es uno de los principales componentes se puede observan una gran similitud en la utilización del internet, pero el protocolo SPICE utiliza un poco más de banda ya que este puede configurar para mejor resolución visual, de audio, de descarga, se pude analizar que es un rango menor que con el número de usuarios incrementados no va haber variación alguna.

Es importante destacar que el protocolo html5 utiliza menos recurso ya que solo utiliza el navegador como parte principal y SPICE requiere una herramienta de software extra para su manipulación.

Tabla 4-4: Resumen general de protocolo HTML5 - 2

	memoria (GB) SPICE	memoria (GB) HTML5
1 usuario	11,5	12,5
8 usuarios	35	36
16 usuarios	50	44
32 usuarios	55	48

Realizado por: Huera, B., 2017.

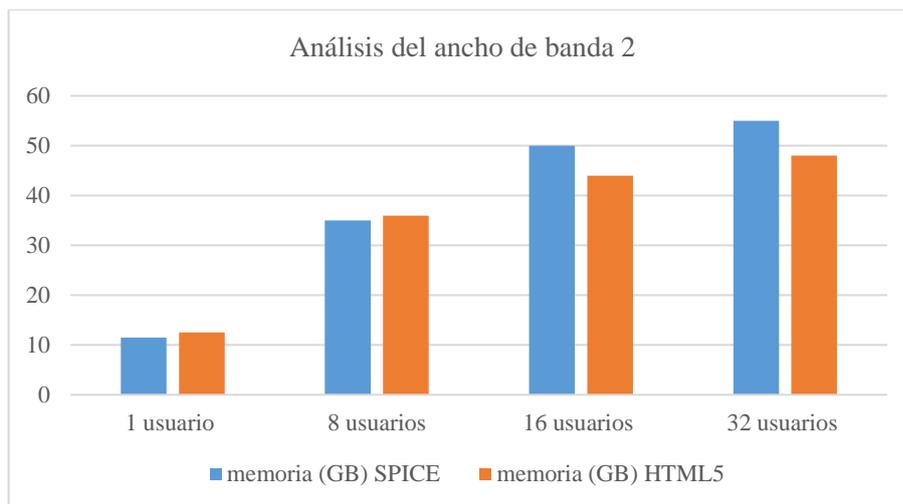


Gráfico 10-4: Análisis del ancho de banda utilizando en la plataforma VDI

Realizado por: Huera, B., 2017.

La grafica final se detalla el análisis del ancho de banda utilizado en la plataforma VDI mediante los protocolos OPENUDS con es SPICE y HTML5, ya que es uno de los principales componentes se puede observar una gran similitud en la utilización del internet, pero el protocolo SPICE utiliza un poco más de banda ya que este puede configurar para mejor resolución visual, de audio, de descarga, se puede analizar que es un rango menor que con el número de usuarios incrementados no va haber variación alguna.

Es importante destacar que el protocolo html5 utiliza menos recursos ya que solo utiliza el navegador como parte principal y SPICE requiere una herramienta de software extra para su manipulación.

Tabla 5-4: Resumen general de protocolo HTML5 - 3

	disco duro (Mb/s) SPICE	disco duro (Mb/s) HTML5
1 usuario	50	24
8 usuarios	150	180
16 usuarios	320	270
32 usuarios	340	280

Realizado por: Huera, B., 2017.

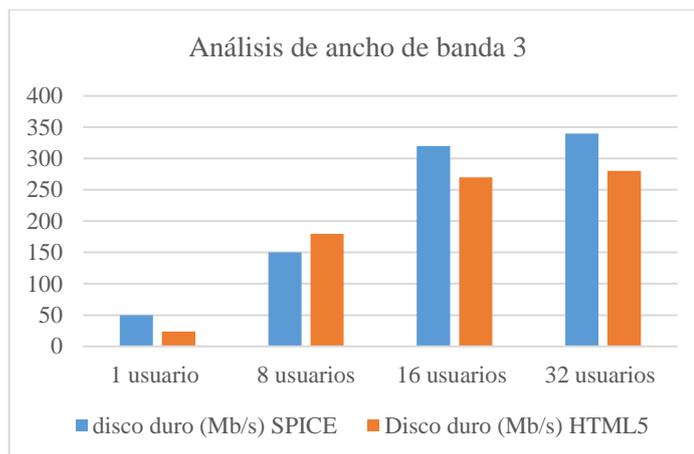


Gráfico 11-4: Análisis de ancho de banda 3

Realizado por: Huera, B., 2017.

La grafica final se detalla el análisis del ancho de banda utilizado en la plataforma VDI mediante los protocolos OPENUDS con es SPICE y HTML5, ya que es uno de los principales componentes se puede observar una gran similitud en la utilización del internet, pero el protocolo SPICE utiliza un poco más de banda ya que este puede configurar para mejor resolución visual, de audio, de descarga, se puede analizar que es un rango menor que con el número de usuarios incrementados no va haber variación alguna.

Es importante destacar que el protocolo html5 utiliza menos recurso ya que solo utiliza el navegador como parte principal y SPICE requiere una herramienta de software extra para su manipulación.

Tabla 6-4: Resumen general de protocolo HTML5 - 4

	red /ancho de banda (Mbps) HTML5	red /ancho de banda (Mbps) SPICE
1 usuario	92	170
8 usuarios	1010	960
16 usuarios	1780	1900
32 usuarios	1680	2000

Realizado por: Huera, B., 2017.

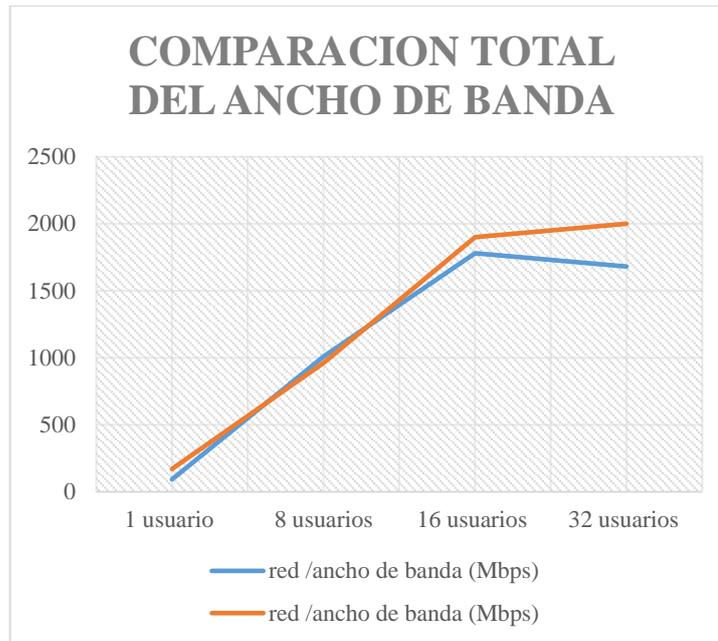


Tabla 7-4: Promedio de ancho de banda HTML5 Y SPICE

Realizado por: Huera, B., 2017.

La grafica final se detalla el análisis del ancho de banda utilizado en la plataforma VDI mediante los protocolos OPENUDS con es SPICE y HTML5, ya que es uno de los principales componentes se puede observar una gran similitud en la utilización del internet, pero el protocolo SPICE utiliza un poco más de banda ya que este puede configurar para mejor resolución visual, de audio, de descarga, se puede analizar que es un rango menor que con el número de usuarios incrementados no va haber variación alguna.

Es importante destacar que el protocolo html5 utiliza menos recurso ya que solo utiliza el navegador como parte principal y SPICE requiere una herramienta de software extra para su manipulación.

4.5 Análisis y validación de resultados

El Método Estadístico utilizado para realizar el análisis de cada escenario se utilizó Pruebas De Bartlett conocido también como modelo de análisis de la varianza para clasificación simple de las observaciones. Es el diseño más sencillo y simple de implementar, ocupado cuando las unidades experimentales son homogéneas o sus variaciones entre ellas son mínimas.

4.5.1 Tabla De Análisis De Kmo Y Pruebas De Bartlett

En el grafico está compuesta por las fuentes de variación, grados de libertad, suma de Cuadrados, cuadrados medidos de cada componente, el factor de f y la probabilidad de significancia, presentan los componentes del análisis de la varianza.

KMO y prueba de Bartlett

Tabla 8-4: Tabla medida Kaiser

Medida de adecuación muestral de Kaiser-Meyer-Olkin.		,826
Prueba de esfericidad de Bartlett	Chi-cuadrado aproximado	44,904
	gl	6
	Sig.	,000

Realizado por: Huera, B., 2017.

El resultado que se obtuvo del análisis de la tabla con respecto a la medida de adecuación muestral de Kaiser-Meyer-Olkin fue de 0,826 por lo que se asegura que la relación que existe en este test es apropiada para aplicar el analisis factorial, debido a que su valor está dentro de 0,5 y factor 1.

4.5.1.1 Planteamiento de hipótesis

El Análisis de los protocolos de transmisión OpenUDS en infraestructuras VDI, mejorara la gestión de los recursos informáticos.

Ho: No existe al menos una componente que caracterice el crecimiento que describa el problema estudiado. Si p valor es mayor o igual a 0,05 se acepta la hipótesis nula.

Ho: No existe al menos una componente en el análisis de los protocolos de transmisión Openuds en infraestructuras VDI que caracterice la mejora de la gestión de los recursos informáticos que describa el problema estudiado. Si p valor es mayor o igual a 0,05 se acepta la hipótesis nula.

Hi: Si existe al menos una componente en el análisis de los protocolos de transmisión Openuds en infraestructuras VDI que caracterice la mejora de la gestión de los recursos informáticos que describa el problema estudiado. Si p valor es mayor o igual a 0,05 se acepta la hipótesis alternativa.

Decisión

Analizando la prueba de esfericidad de Bartlett de la tabla el valor de sig es 0,000 por lo que se rechaza la hipótesis nula aceptando la hipótesis alternativa, mencionando que si es posible realizar el análisis factorial.

4.5.1.2 Análisis factorial:

Tabla 9-4: Matriz de correlaciones y prueba de Bartlett

		red_ancho_banda_Mbps_	procesador	memoria	disco_duro
Correlación	red_ancho_banda_Mbps_	1,000	,823	,983	,994
	procesador	,823	1,000	,852	,817
	memoria	,983	,852	1,000	,983
	disco_duro	,994	,817	,983	1,000
Sig. (Unilateral)	red_ancho_banda_Mbps_		,006	,000	,000
	procesador	,006		,004	,007
	memoria	,000	,004		,000
	disco_duro	,000	,007	,000	

KMO y prueba de Bartlett

Medida de adecuación muestral de Kaiser-Meyer-Olkin.	,826
Prueba de esfericidad de Bartlett	Chi-cuadrado aproximado
	44,904
	gl
	6
	Sig.
	,000

Realizado por: Huera, B., 2017.

Tabla 10-4: Comunalidades

	Inicial	Extracción
red_ancho_banda_Mbps_	1,000	,972
procesador	1,000	,811
memoria	1,000	,980
disco_duro	1,000	,969

Realizado por: Huera, B., 2017.

Tabla 11-4: Varianza total explicada

Componente	Autovalores iniciales			Sumas de las saturaciones al cuadrado de la extracción	
	Total	% de la varianza	% acumulado	Total	% de la varianza
1	3,731	93,284	93,284	3,731	93,284
2	,245	6,122	99,406		
3	,018	,456	99,861		
4	,006	,139	100,000		

4.5.2 Método de extracción: Análisis de componentes principales

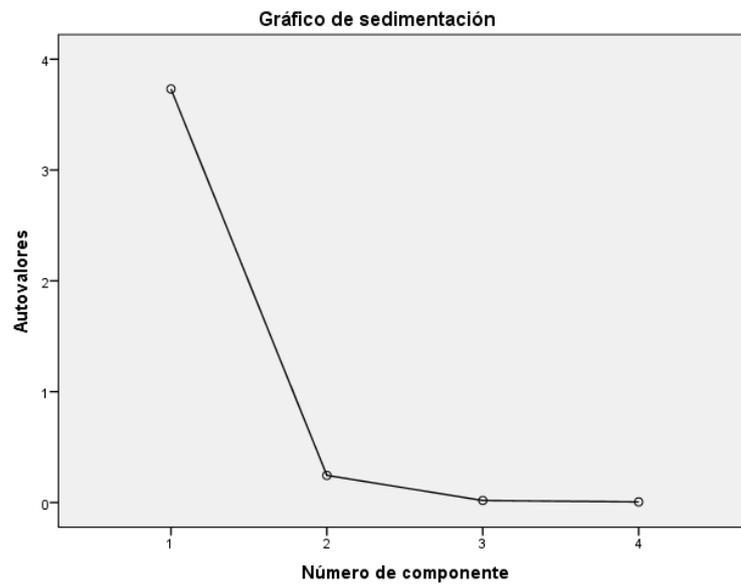


Gráfico 12-4: Segmentación

Realizado por: Huera, B., 2017.

Tabla 12-4: Matriz de componentes

	Componente
	1
red_ancho_banda_Mbps_	,986
procesador	,900
memoria	,990
disco_duro	,984

Realizado por: Huera, B., 2017.

CONCLUSIONES

Mediante el análisis del marco teórico se puede decir que la convivencia de las tecnologías HTML5 y SPICE es posible, ya que trabajan en diferentes ámbitos y que además utilizan diferentes recursos para mejorar el desempeño óptimo en la utilización de la plataforma, pero con similares características de envío y recepción de la información en cloud computing.

Con la utilización del prototipo se pudo cumplir con el objetivo propuesto de que es posible la coexistencia de la tecnología VDI en los dos protocolos estudiados. OpenUDS al ser utilizado como un enlace multidireccional ha sido configurado para un mejor desempeño cliente –servidor en un 90 % ya que no requiere hardware especial para su utilización, sino más bien una buena recepción de ancho de banda y requerimientos básicos de sistemas operativos y navegadores actualizados.

De la comparación de eficiencia entre el sistema tradicional de virtualización, concluimos que las infraestructuras informáticas VDI, funcionan en un 50% mejor en un entorno virtualizado, la congestión en la red es mucho más alta pero el uso de los recursos por parte del usuario es insignificante ya que el hardware usado es de un 10% en el cliente.

De las pruebas realizadas se tiene que debido al gran uso del recurso principal que la memoria RAM y el ancho de banda, pueden fácilmente bloquear el acceso a la plataforma, si no se ha configurado correctamente los recursos como dar un límite de ancho de banda y un tamaño determinado de espacio en la memoria RAM a cada usuario. Es por esto que la correcta configuración del servidor se convierte en un factor muy considerable.

La velocidad de transmisión y recepción en nuestro proyecto se vio bajamente limitada debido a que las configuraciones de la red interna están limitadas y esto no están en la capacidad de procesar y soportar tantas velocidades altas en peticiones de los usuarios. Pero es seguro que conforme las compañías virtualizadas vayan desarrollando esta tecnología se contará con componentes sofisticados capaces de transmitir datos a velocidades inmensa estará presente en un futuro no muy lejano.

Sea ha demostrado que una estrecha integración de ambas tecnologías HTML5 Y SPICE permite que las oportunidades de uso de los recursos informáticos para más usuarios tanto fijos como móviles, ya que ambas tecnologías pueden colaborar eficazmente.

RECOMENDACIONES

Dados los resultados descritos en este trabajo de investigación, la comunicación entre el cliente y el servidor se ve prometedora y evolutiva que ofrece valiosas contribuciones como parte de redes haciendo necesario el seguimiento de su desarrollo en nuestra Politécnica.

Buscar alternativas de dispositivos electrónicos con gran capacidad y velocidad para mejorar el diseño del prototipo tales como memorias RAM de alta velocidad, discos duros de gran capacidad, servidores con mayores desempeños, que me permitan alcanzar mayores velocidades de transmisión.

Se recomienda que cada una de las etapas que componen el sistema VDI, sea realizada por separado para facilitar la configuración y el funcionamiento de cada módulo.

Continuar con los estudios y análisis de los sistemas descritos en el presente trabajo ya que pudieran ser la solución adecuada las necesidades de mejorar la infraestructura actual con las mayores velocidades de datos necesarias para las generaciones actuales y futuras de las redes virtuales.

Abrir las puertas a nuevas tecnologías como lo es OPENUDS Y VDI puede resolver problemas como la escasez de ancho de banda, limitación en los dispositivos con baja capacidad y también permitir el acceso desde cualquier parte del mundo. En el futuro, no solo tendremos cientos de usuarios si no miles de usuarios haciendo uso de esta tecnología.

GLOSARIO

Máquina virtual: es aquella que emula a un ordenador completo. En palabras llanas, es un software que puede hacerse pasar por otro dispositivo -como un PC- de tal modo que puedes ejecutar otro sistema operativo en su interior. Tiene su propio disco duro, memoria, tarjeta gráfica y demás componentes de hardware, aunque todos ellos son virtuales.

Escritorio Remoto: es una manera de que un usuario puede controlar virtualmente un ordenador o dispositivo desde una ubicación remota. Le permite acceder a todos los recursos y funciones del ordenador remoto como si estuviera sentado delante.

Entorno Virtual de Aprendizaje: es un espacio educativo alojado en la web, un conjunto de herramientas informáticas que posibilitan la interacción didáctica de manera que el alumno pueda llevar a cabo las labores propias de la docencia como son conversar, leer documentos, realizar ejercicios, formular preguntas al docente, trabajar en equipo... etc. Todo ello de forma simulada sin que medie una interacción física entre docentes y alumnos.

Hyper Text Markup Language 5: HTML5 es la última versión de HyperText Markup Language (Lenguaje de Marcado de Hipertexto). Combina tres tecnologías web: HTML, para construir la estructura de la página; CSS, para mejorar la presentación; y JavaScript, para agregar funcionalidad a los elementos HTML.

Simple Protocol for Independent Computing Environments: SPICE (Protocolo Simple para Entornos de Computación Independientes) es un protocolo de comunicación para entornos virtuales. Les permite a los usuarios ver la consola de máquinas virtuales (VM) desde cualquier parte a través de Internet. Utiliza un modelo cliente-servidor que utiliza a Virtualization Station como un host y los usuarios se pueden conectar a las VM a través del cliente SPICE.

Infraestructura de Escritorio Virtual: (VDI) es una herramienta de software para crear y administrar escritorios virtuales de forma centralizada. Los escritorios virtuales son aplicaciones de software que imitan a las computadoras de escritorio reales. La tecnología VDI crea un escritorio virtual en un servidor central, al que los usuarios remotos pueden acceder desde cualquier máquina física a través de Internet.

BIBLIOGRAFÍA

Coates, H., James, R., & Baldwin, G. “A critical examination of the effects of learning management systems on university teaching and learning”. *Tertiary Education And Management* [en línea], 2005, (United State of America), vol. 11(1), pág. 19-36. [consulta: 20 agosto 2022]. Disponible en: doi:10.1080/13583883.2005.9967137

Mell, P. & Grance, T. *The NIST Definition of Cloud Computing Recommendations of the National Institute of Standards and Technology*. Gaithersburg: NIST, 2011. Págs. 1-3.

Moina, W. & Borja, E. Implementación de una plataforma en la nube para los cursos masivos abiertos en línea (mooc) utilizando google course builder e infraestructuras de escritorio virtual. [En línea]. (Trabajo de Titulación) (Pregrado). Escuela Superior Politécnica de Chimborazo. Riobamba – Ecuador. 2017. Págs. 1- 63. [Consulta: 20020-05-23]. Disponible en: <http://dspace.esPOCH.edu.ec/handle/123456789/6787>

Cloud Computing - Gartner IT Glossary. *Gartner IT Glossary*. [blog]. EE. UU., 2017. [Consulta: 8 enero 2017]. Disponible en: <http://www.gartner.com/it-glossary/cloud-computing>.

Horrigan, J. *Use of Cloud Computing Applications and Services*. *Pew Research Center: Internet, Science & Tech.* [blog]. EE. UU., 2008. [Consulta: 3 enero 2017]. Disponible en: <http://www.pewinternet.org/2008/09/12/use-of-cloud-computing-applications-and-services/>

Hewitt, C. “ORGs for Scalable, Robust, Privacy-Friendly Client Cloud Computing”. *IEEE Internet Computing*, vol. 12(5), (2008), (Unite State of America). Págs. 96-99. Disponible en: doi:10.1109/mic.2008.107

Bronzin, T., & Kaldi, V. *Mobility in the Cloud*. 2ª ed. Barcelona-España: Infotren, 2011. Pág. 1-186.

Bronzin, T. *Cloud Computing. Pogled kroz prozor*. [blog]. Unite State of America: WordPress, 2009. [Consulta: 24 enero 2013]. Disponible en: <https://pogledkrozprozor.wordpress.com/2009/08/29/%E2%80%9Ecloud-computing%E2%80%9C-ili-programska-rjesenja-u-oblacima/>

Žganec, T. *The Cloud is the earth*. UU. EE.: Infotren, 2011, pág. 184.

Lin, A., & Chen, N.-C. “Cloud computing as an innovation: Perception, attitude, and adoption”. *International Journal of Information Management* [en línea], 2012, (EE. UU.), vol. 127(2), pág. 200. [Consulta: 26 enero 2013]. Disponible en: <http://dx.doi.org/10.1016/j.ijinfomgt.2012.04.001>.

Sultan, N. “Cloud computing for education: A new dawn”. *International Journal of Information Management* [en línea], 2010, (EE. UU.), vol. 30(2), págs. 109–116. [Consulta: 27 enero 2013]. Disponible en: doi:10.1016/j.ijinfomgt.2009.09.004

Rao, N. M., Sasidhar, C., & Kumar, V. S. “Cloud computing through mobile-learning. International”. *Journal of Advanced Computer Science and Applications* [en línea], 2012, (EE. UU.), vol. 1(6). [Consulta: 26 enero 2013]. Disponible en: <http://arxiv.org/abs/1204.1594>

Kop, R., & Carroll, F. (2011). “Cloud Computing and Creativity: Learning on a Massive Open Online Course”. *European Journal of Open, Distance and E-Learning*, Kop, [en línea], 2011, (EE. UU.), vol. 14(2). [Consulta: 26 enero 2013]. Disponible en: <http://arxiv.org/abs/1788.1594>

ADDIN Mendeley Bibliography CSL_BIBLIOGRAPHY Al-Zoube, M. (2009). “E-learning on the Cloud”. *International Arab Journal of e-Technology* [en línea], 2009, (Saudi Arabia), vol. 1(3) [Consulta: 28 febrero 2017]. Disponible en: http://www.iajet.org/iajet/iajet_files/vol.1/no.2/E-Learning%20on%20the%20Cloud.pdf

UNESCO. *Cloud Computing in Education. IITE Policy Brief*, [blog]. Unite State of America: IITE, 2010. [Consulta: 29 enero 2017]. Disponible en: <http://iite.unesco.org/pics/publications/en/files/3214674.pdf>

Bittman, T. *Cloud Computing Inquiries at Gartner. Gartner*, [blog]. Unite State of America: Gartner, 2009. [Consulta: 24 enero 2013]. Disponible en: http://blogs.gartner.com/thomas_bittman/2009/10/29/cloud-computing-inquiries-at-gartner/

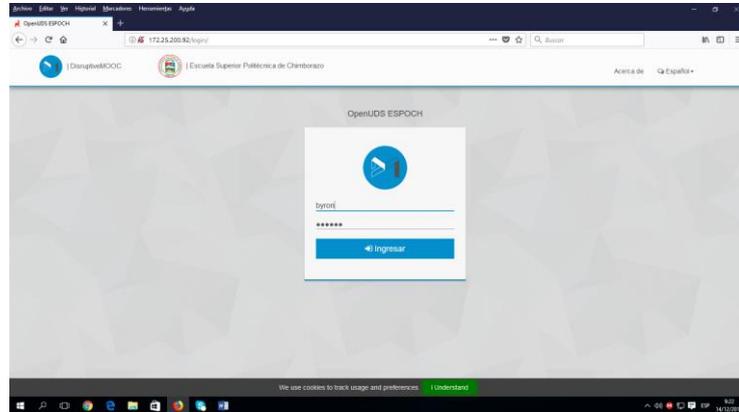
López, P. (2016). Comparación del desempeño de los Sistemas Gestores de Bases de Datos MySQL y PostgreSQL. [en línea]. (Trabajo de Titulación) (pregrado). Universidad Autónoma del Estado de México, Mexico. 2016. págs. 20-23. 3. [Consulta: 24 enero 2017]. Disponible en: <http://dspace.espace.edu.ec/bitstream/123456789/220/1/236T0001.pdf>

UDS Enterprise Team. *¿Qué es un broker de conexiones Retrieved?* [blog]. España: Udsenterprise, 2016. [Consulta: 2 febrero 2017]. Disponible en: <https://www.udsenderprise.com/es/blog/2016/04/25/que-es-un-broker-de-conexiones/>

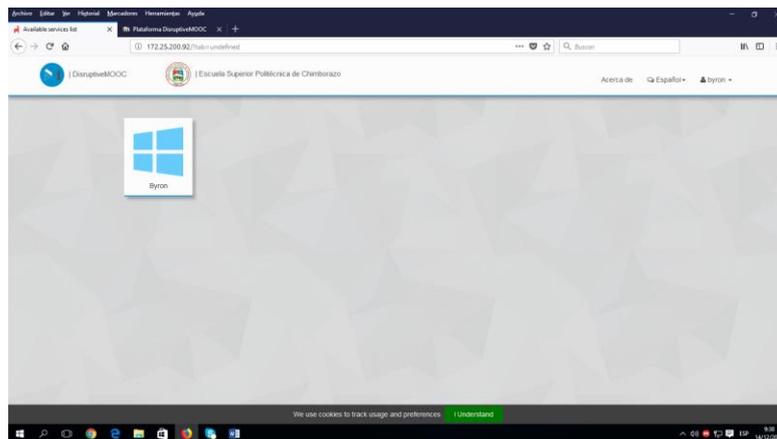
UDS Enterprise Team. (2016). *UDS Enterprise Características generales* [blog]. España: Udsenterprise, 2016. [Consulta: 2 febrero 2017]. Disponible en: https://www.udsenderprise.com/media/filer_public/76/b1/76b1b745-32c2-45c2-8ef9-b0449654a976/uds-enterprise-caracteristicas-generales.pdf

ANEXOS

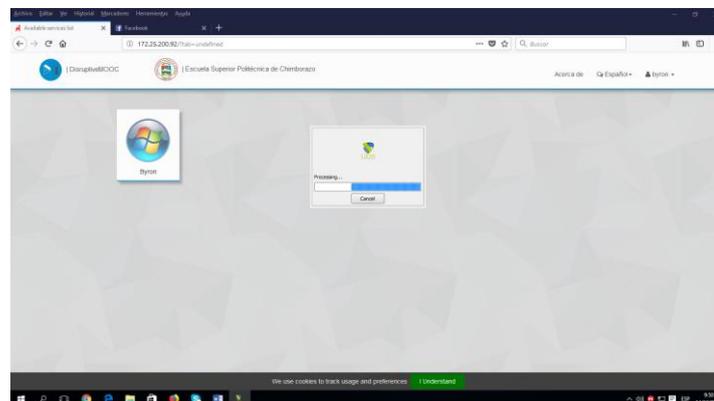
ANEXO A: INGRESO AL SISTEMA OPENUDS



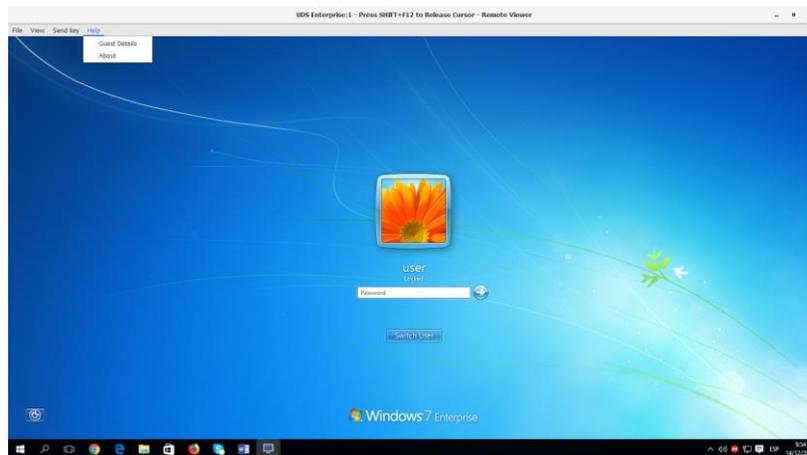
Colocamos nuestro usuario y contraseña para ingresar al sistema por medio de un navegador



Observamos la máquina virtual designada:



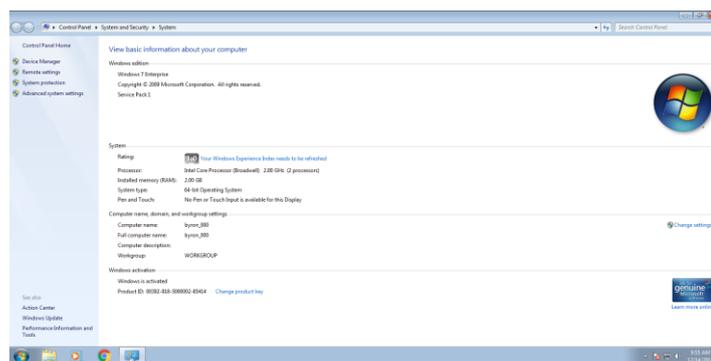
Damos clic y cargamos es gestor Uds



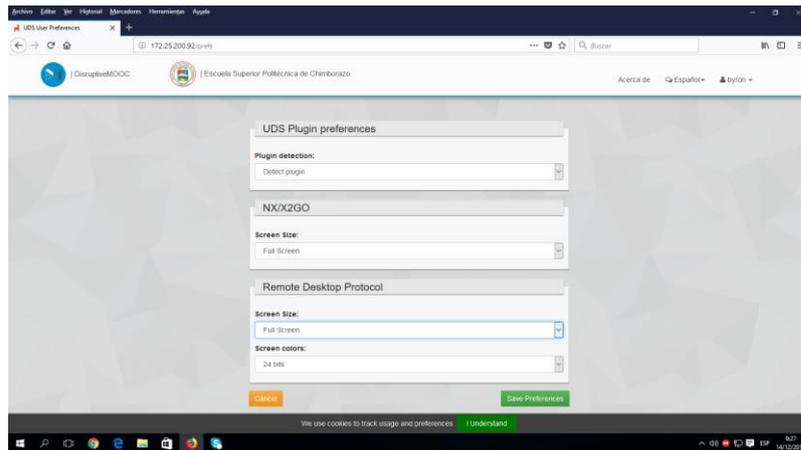
Colocamos nuestro usuario y clave de login en la máquina virtual



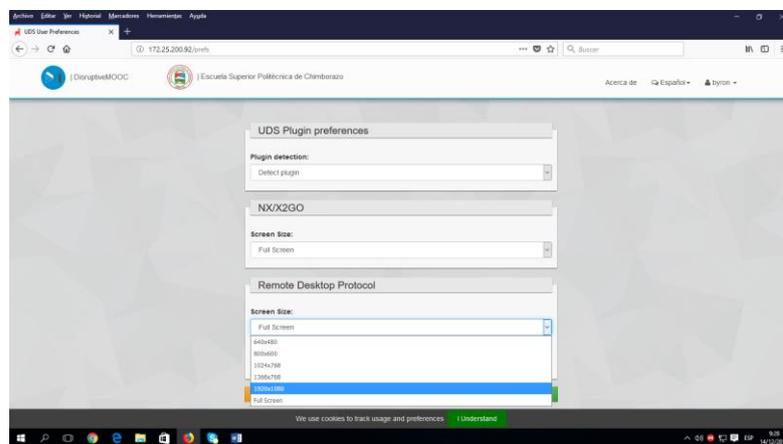
Ingresamos a la máquina virtual



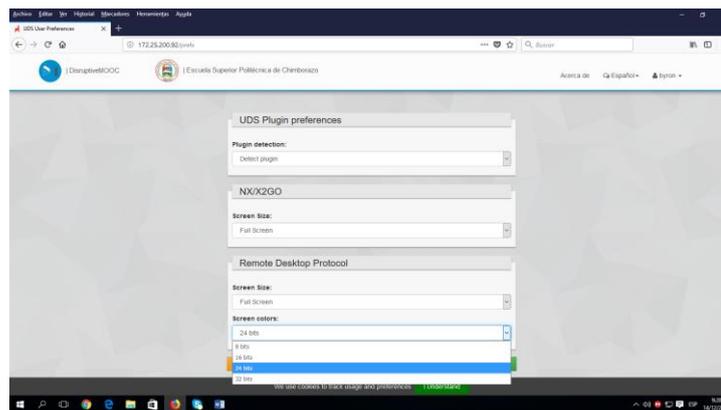
Observamos las propiedades de la máquina virtual previamente asignado RAM y disco duro



Las preferencias de open uds



Podemos darles la resolución de la pantalla a nuestros usuarios



Así como el color en bits

Archivo Editar Ver Historial Marcadores Herramientas Ayuda

172.25.200.92/about

172.25.200.92/about

DisruptiveMOOC | Escuela Superior Politécnica de Chimborazo

Acerca de Español byron

Universal Desktop Services 2.1.0-DEVEL build 20161001-DEVEL

© 2012-2016 Virtual Cable S.L.U.

You can access UDS Open Source code at [OpenUDS](#)

UDS has been developed using these components:

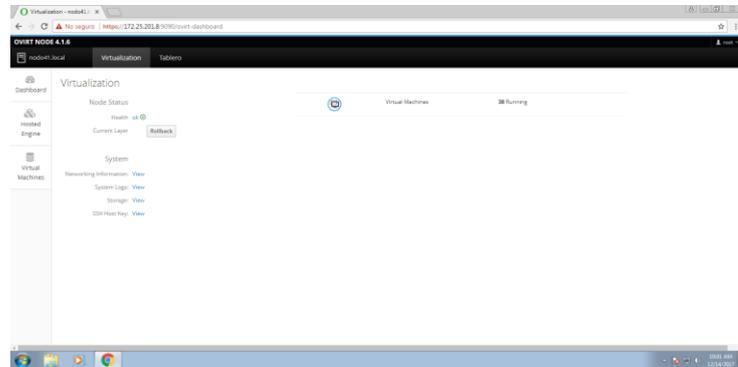
- Eclipse as IDE with pyDev
- Django
- Jactl
- Guacamole
- Django compressor
- Bootstrap
- Bootstrap switch
- Bootstrap select
- Bootstrap touchspin
- Font Awesome by Dave Gandy
- CoffeeScript
- SASS / Compass
- JQuery
- Handlebars
- JQuery cookie

We use cookies to track usage and preferences [I Understand](#)

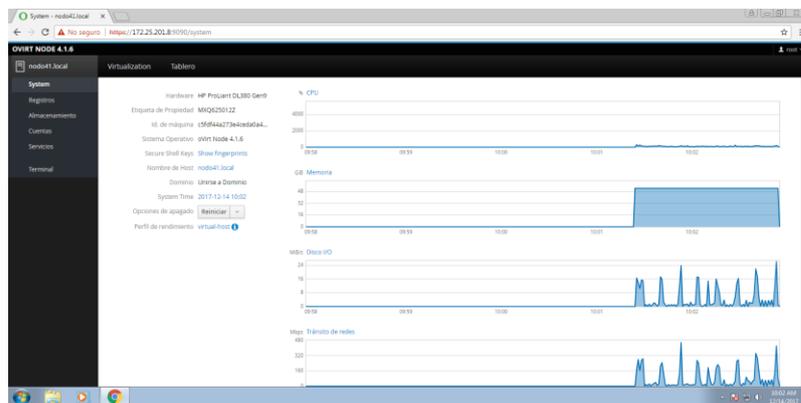
928 14/12/2015

ANEXO B: UTILIZACION DEL OVIRT NODE MONITOREO

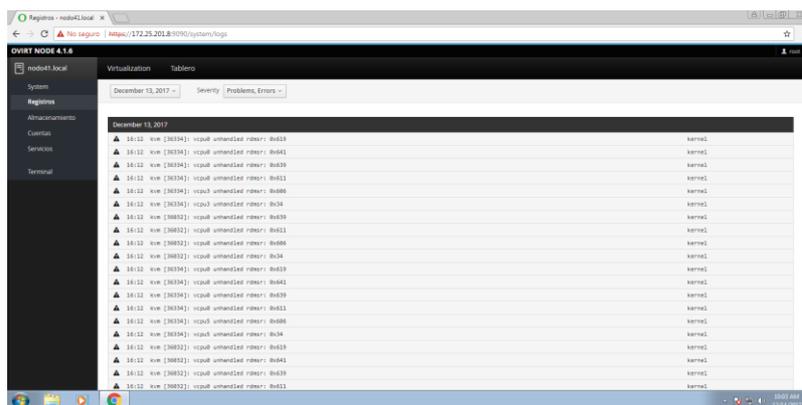
<https://172.25.201.8:9090/system/terminal>



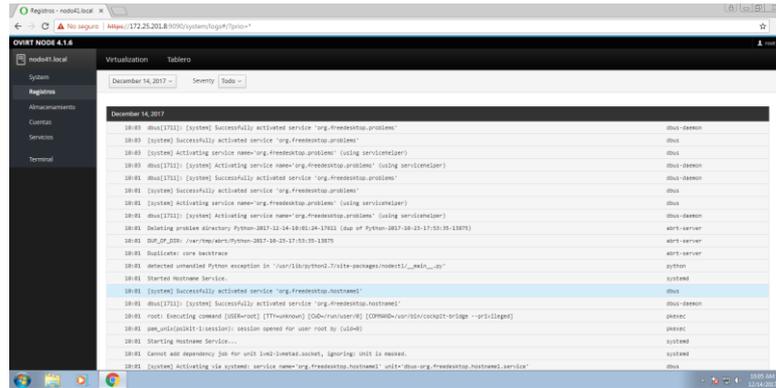
Esta herramienta nos muestra cómo se puede monitorear nuestra plataforma



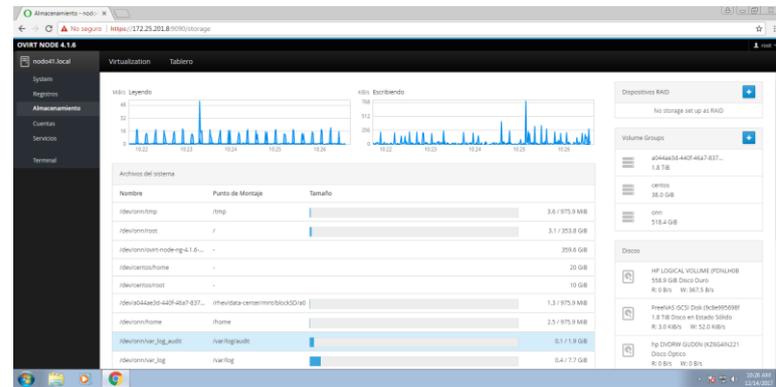
Una visualización general de nuestra plataforma implementada



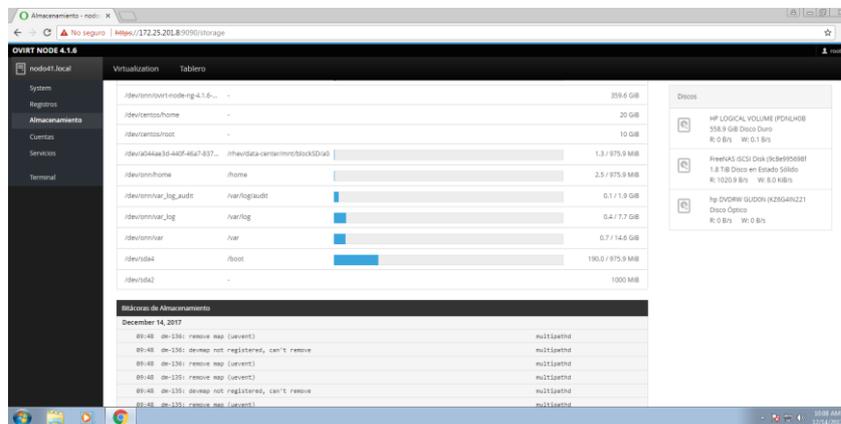
Llevar un registro de cada una de las actividades realizadas en el servidor de VDI



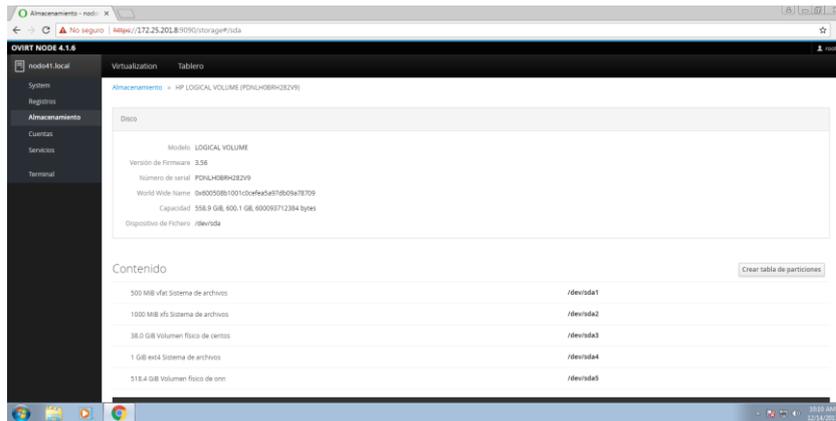
Toda las activada realizada



En el almacenamiento

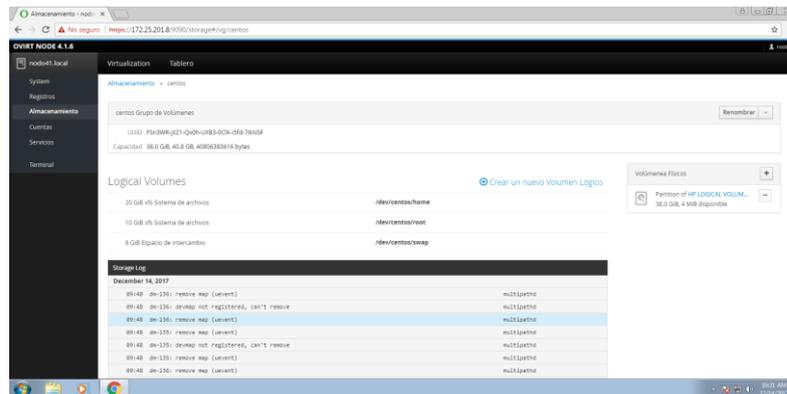


Bitácoras

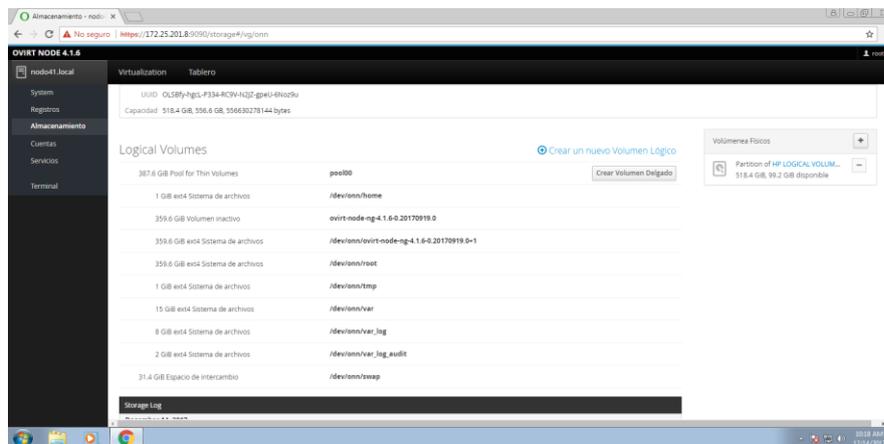


El volumen lógico hp de 600 GB

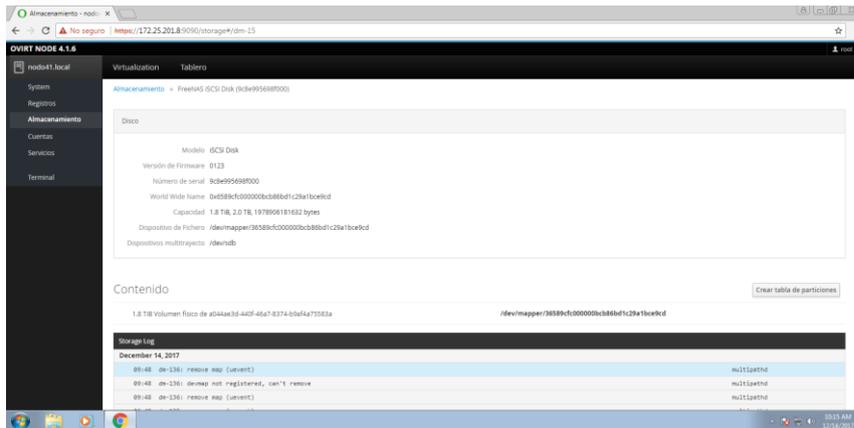
Dentro de este tenemos el sistema operativo Centos instalado, todas las configuraciones necesarias para el correcto funcionamiento tales como Openuds, ovirt node, guacamole, html5, Spice:



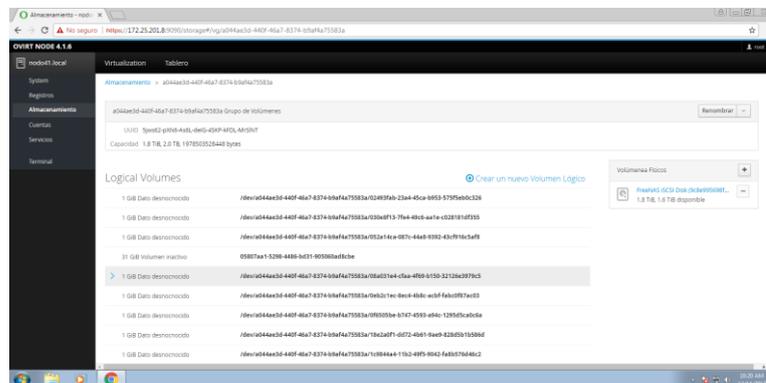
El tamaño de disco utilizado en Centos



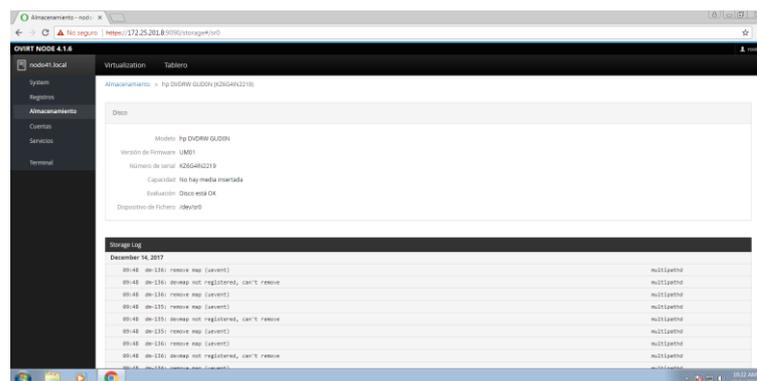
Ovirt-node



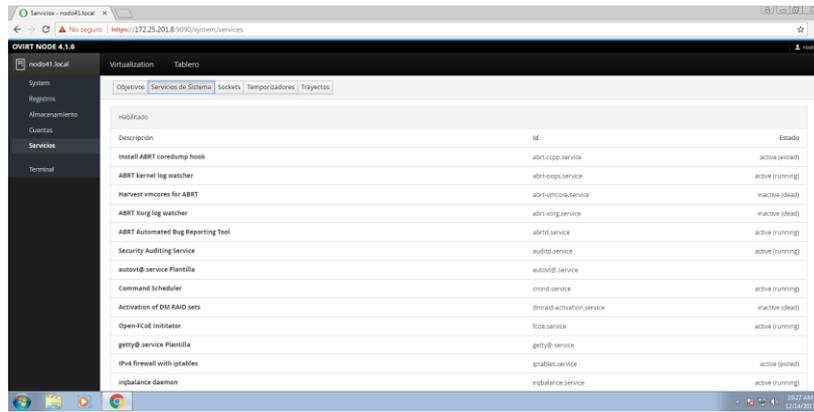
En este disco tenemos configurado las máquinas virtuales que van a hacer utilizado por la plataforma VDI, FreeNAS iscsi con una capacidad de 2TB.



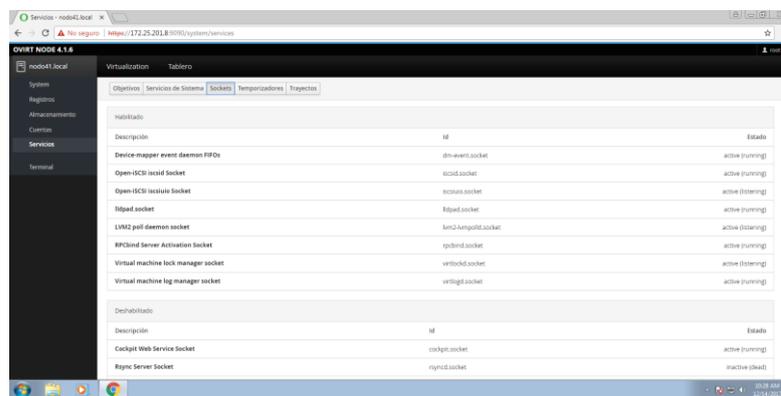
Se puede considerar que cada ítem es una máquina virtual asignada a un usuario



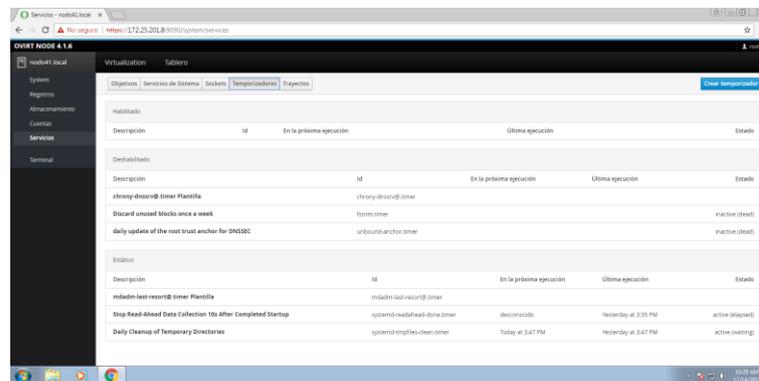
La unidad óptica en el servidor HP



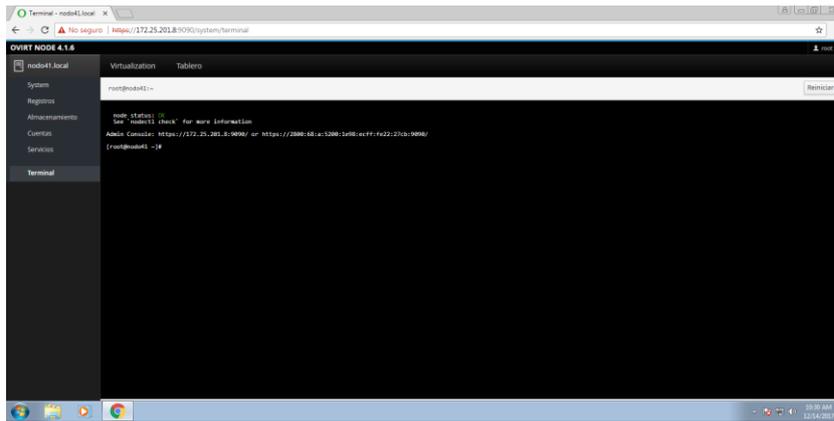
Servicios del sistema que se está ejecutando



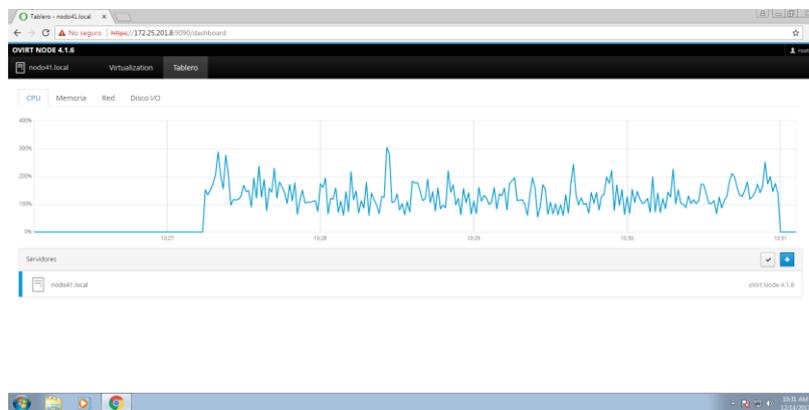
Los sockets en estado de ejecución



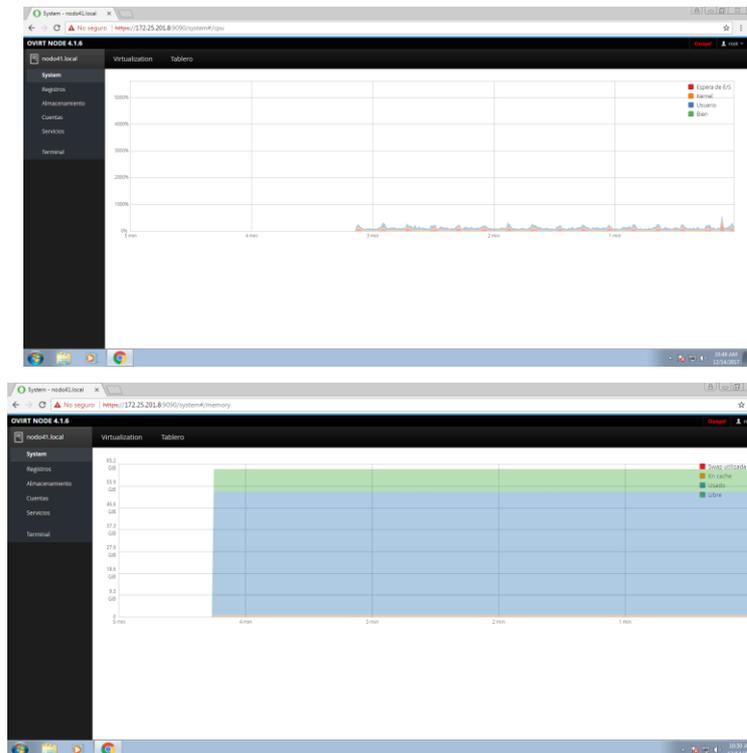
Temporales



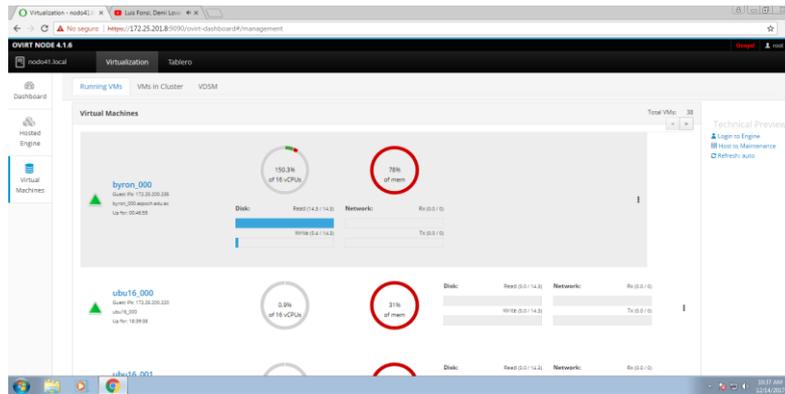
Una consola de terminal capaz de acceder al sistema directamente



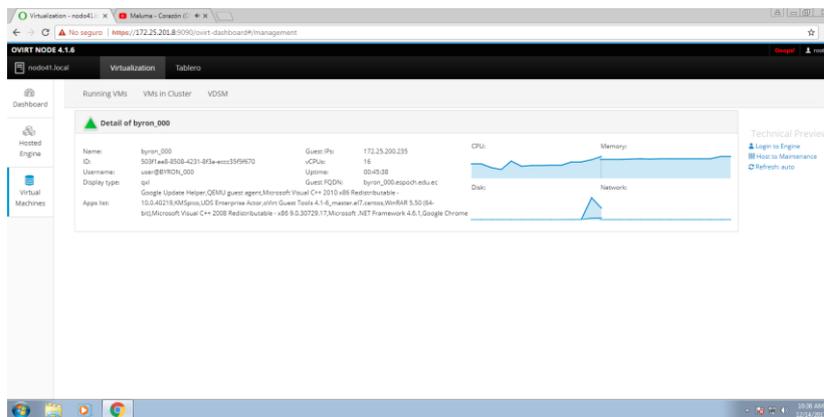
Visualización de nuestro sistema como tablero



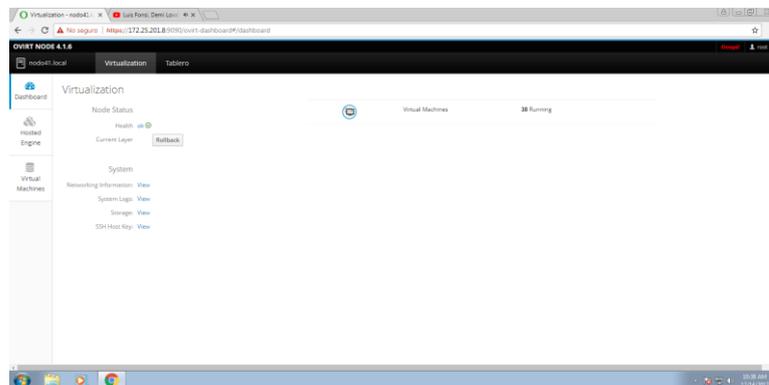
En detalle los componentes ejecutándose



En visualización se ve las máquinas virtuales en ejecución

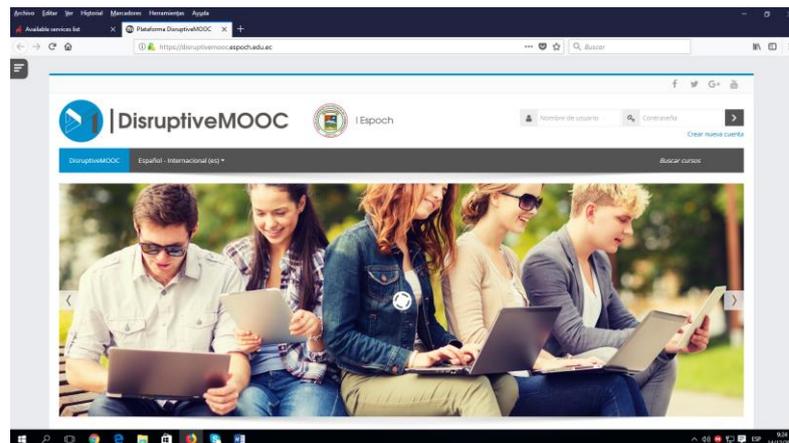


Revisamos una máquina virtual en ejecución

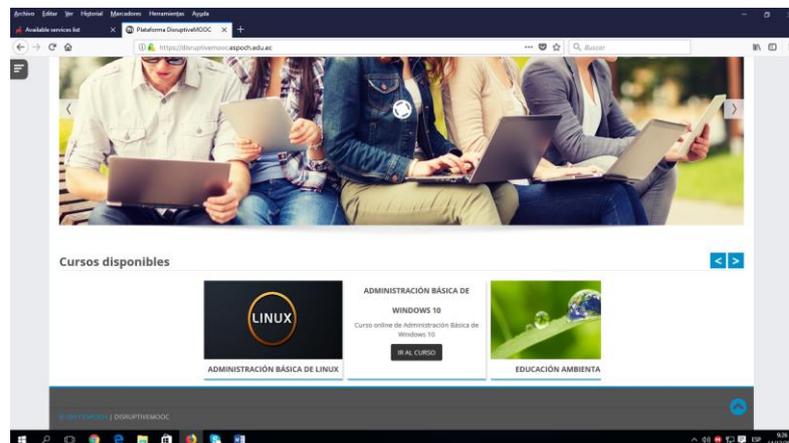


Máquinas actualmente en ejecución

ANEXO C: ACCESO POR EL ESTUDIANTE AL SISTEMA



Colocamos nuestro usuario y contraseña



ANEXO D: INGRESO AL SISTEMA POR MEDIO DE CELULAR

Utilización por medio de un dispositivo móvil Y sistema Operativo Android

Loguin de usuario en dispositivo móvil:



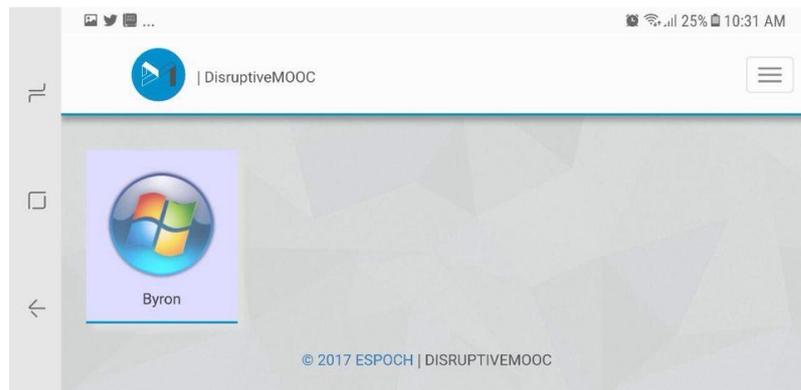
Una descripción de la plataforma



Bienvenido a

La plataforma
DisruptiveMOOC

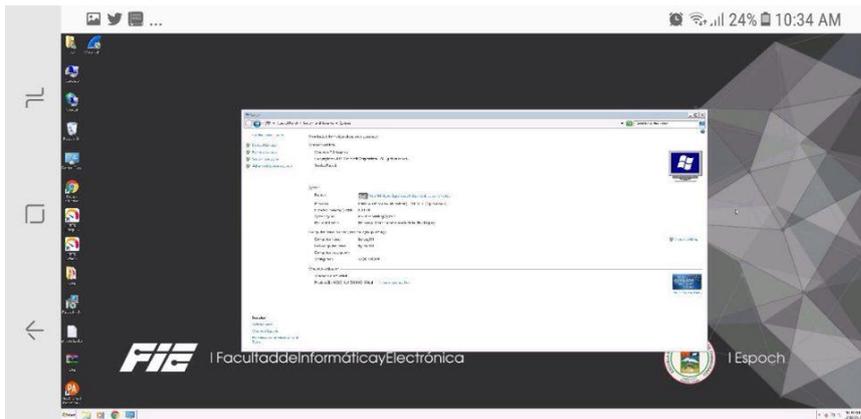
Como siguiente paso observamos la apertura de la cuenta:



Luego escogemos la máquina virtual asignada:

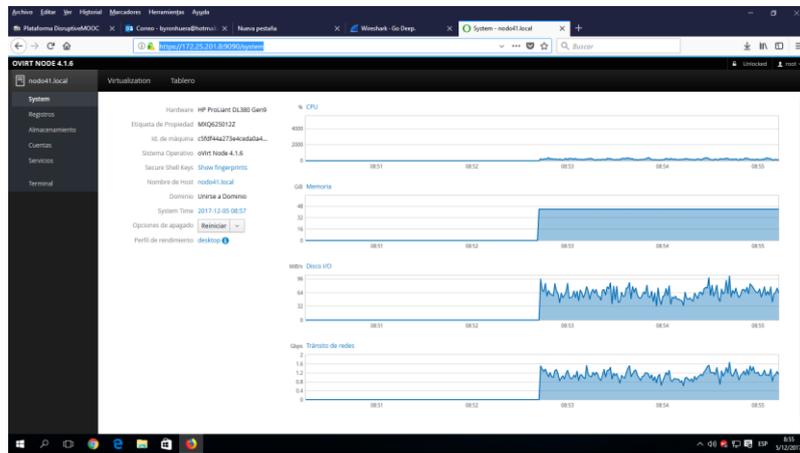


En esta etapa podemos observar la utilización de la máquina virtual desde el dispositivo móvil

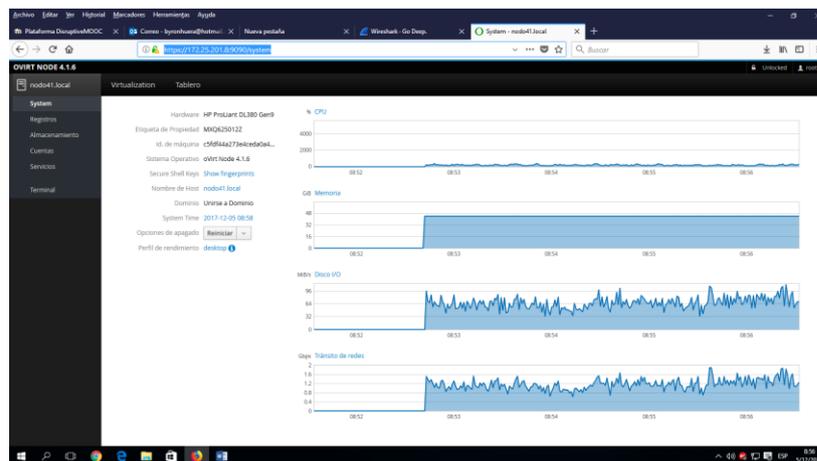


Podemos utilizar como cualquier pc

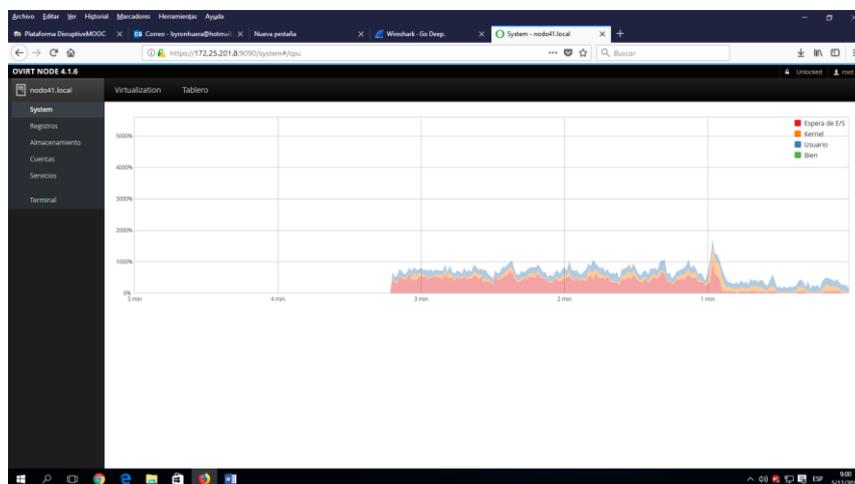
ANEXO E: CREACIÓN DE UNA MAQUILNA VIRTUAL EN LA PLATAFORMA MONITOREO



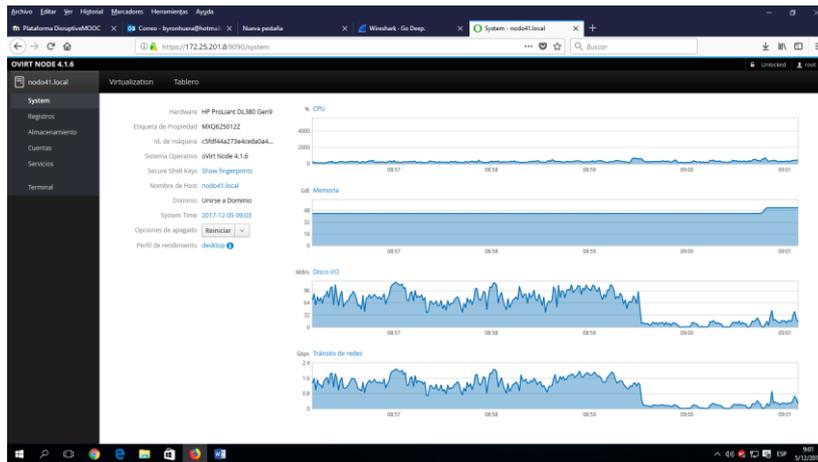
Mas tiempo



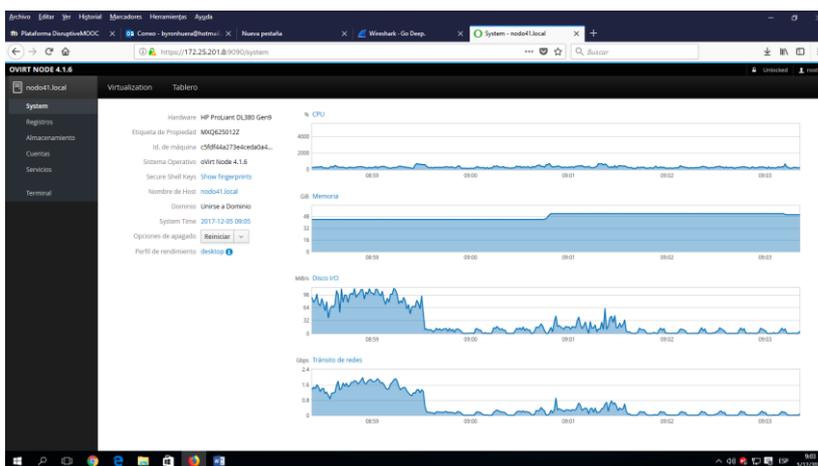
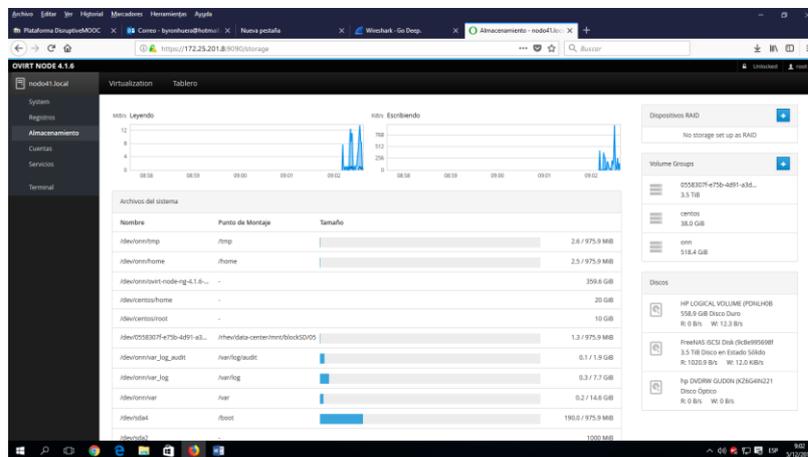
Uso detallado

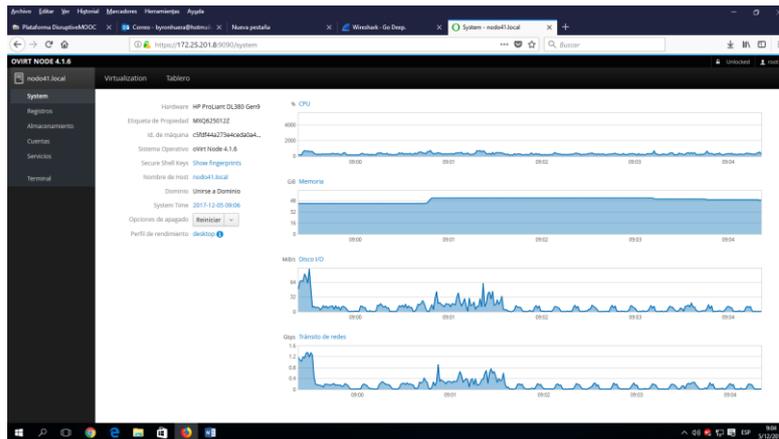


Después de la creación

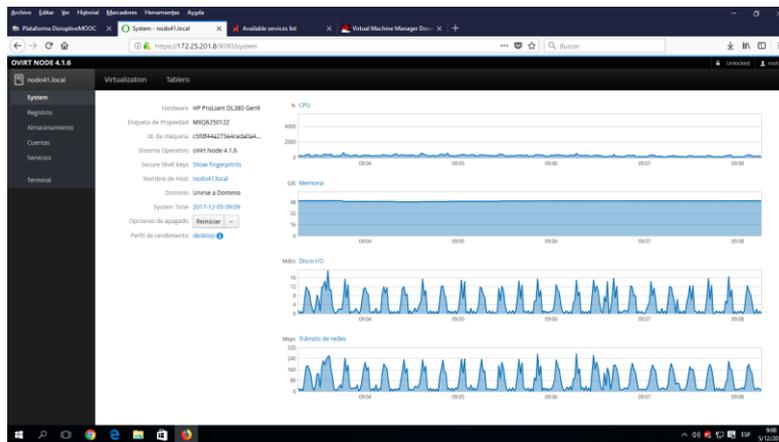
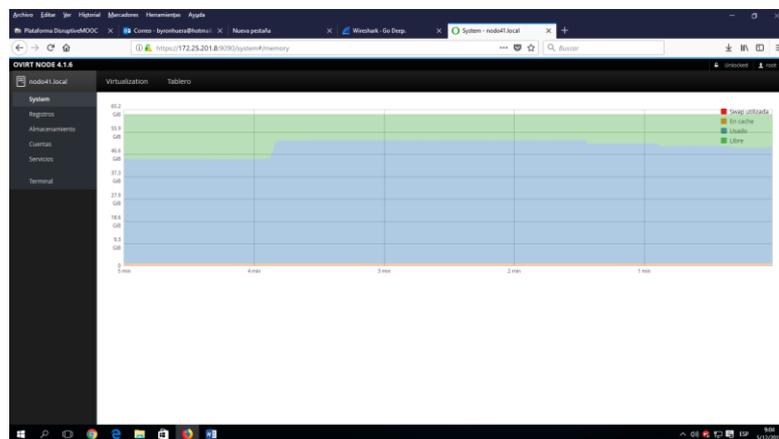


Disco duro

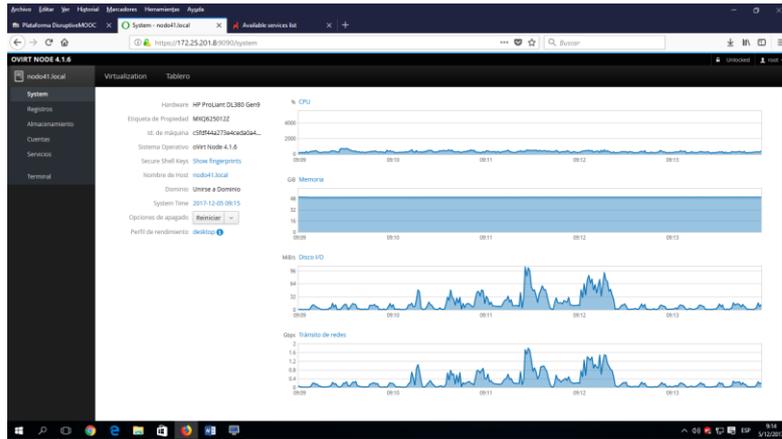




Memoria



La utilización y apagado de la maquina observamos que baja la utilización



ANEXO F: CODIGO PYTHON

```
# -*- coding: utf-8 -*-
#
# Copyright (c) 2014 Virtual Cable S.L.
# All rights reserved.
#
# Redistribution and use in source and binary forms, with or
without modification,
# are permitted provided that the following conditions are met:
#
#     * Redistributions of source code must retain the above
copyright notice,
#     this list of conditions and the following disclaimer.
#     * Redistributions in binary form must reproduce the above
copyright notice,
#     this list of conditions and the following disclaimer in
the documentation
#     and/or other materials provided with the distribution.
#     * Neither the name of Virtual Cable S.L. nor the names of
its contributors
#     may be used to endorse or promote products derived from
this software
#     without specific prior written permission.
#
# THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND
CONTRIBUTORS "AS IS"
# AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
LIMITED TO, THE
# IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A
PARTICULAR PURPOSE ARE
# DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR
CONTRIBUTORS BE LIABLE
# FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR
CONSEQUENTIAL
# DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF
SUBSTITUTE GOODS OR
```

```
# SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS
INTERRUPTION) HOWEVER
# CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT,
STRICT LIABILITY,
# OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY
OUT OF THE USE
# OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH
DAMAGE.
```

```
'''
```

```
@author: Adolfo G3mez, dkmaster at dkmon dot com
```

```
'''
```

```
from __future__ import unicode_literals
```

```
from django.utils.translation import ugettext as _
```

```
from uds.core.util import Config
```

```
from uds.core.util.State import State
```

```
from uds.core.util.model import processUuid
```

```
from uds.core.util import log
```

```
from uds.core.managers import cryptoManager
```

```
from uds.models import TicketStore
```

```
from uds.REST import Handler
```

```
from uds.REST import RequestError
```

```
from uds.models import UserService
```

```
import datetime
```

```
import six
```

```
import logging
```

```
logger = logging.getLogger(__name__)
```

```
# Actor key, configurable in Security Section of administration
interface
```

```

actorKey =
Config.Config.section(Config.SECURITY_SECTION).value('Master
Key',

cryptoManager().uuid(datetime.datetime.now()).replace('-', ''),

type=Config.Config.TEXT_FIELD)
actorKey.get()

# Error codes:
ERR_INVALID_KEY = 1
ERR_HOST_NOT_MANAGED = 2
ERR_USER_SERVICE_NOT_FOUND = 3
ERR_OSMANAGER_ERROR = 4

# Enclosed methods under /actor path
class Actor(Handler):
    '''
    Processes actor requests
    '''
    authenticated = False # Actor requests are not authenticated

    @staticmethod
    def result(result=None, error=None):
        '''
        Helper method to create a "result" set for actor response
        :param result: Result value to return (can be None, in
which case it is converted to empty string '')
        :param error: If present, This response represents an
error. Result will contain an "Explanation" and error contains
the error code
        :return: A dictionary, suitable for response to Caller
        '''
        result = result if result is not None else ''
        res = {'result': result, 'date': datetime.datetime.now()}
        if error is not None:

```

```

        res['error'] = error
    return res

def test(self):
    """
    Executes and returns the test
    """
    return Actor.result(_('Correct'))

def validateRequestKey(self):
    """
    Validates a request key (in "key" parameter)
    """
    # Ensures that key is first parameter
    # Here, path will be .../actor/ACTION/KEY (probably
    /rest/actor/KEY/...)
    logger.debug('{} == {}'.format(self._params.get('key'),
actorKey.get(True)))
    if self._params.get('key') != actorKey.get(True):
        return Actor.result(_('Invalid key'),
error=ERR_INVALID_KEY)
    return None

def getUserServiceByIds(self):
    """
    This will get the client from the IDs passed from
    parameters
    """
    logger.debug('Getting User services from ids:
{}'.format(self._params.get('id')))

    try:
        clientIds = [i.upper() for i in
self._params.get('id').split(',')[:5]]
    except Exception:
        raise RequestError('Invalid request: (no id found)')

```

```

        services =
UserService.objects.filter(unique_id__in=clientIds,
state__in=[State.USABLE, State.PREPARING])
        if services.count() == 0:
            return None

        return services[0]

def getTicket(self):
    '''
    Processes get requests in order to obtain a ticket content
    GET /rest/actor/ticket/[ticketId]
    '''
    logger.debug("Ticket      args      for      GET:
{0}".format(self._args))

    if len(self._args) != 2:
        raise RequestError('Invalid request')

    try:
        return Actor.result(TicketStore.get(self._args[1],
invalidate=True))
    except Exception:
        return Actor.result({})

def get(self):
    '''
    Processes get requests
    '''
    logger.debug("Actor      args      for      GET:
{0}".format(self._args))

    if len(self._args) < 1:
        raise RequestError('Invalid request')

    if self._args[0] == 'ticket':
        return self.getTicket()

```

```

        if self._args[0] == 'testn': # Test, but without master
key
            return self.test()

        #         if         path         is         .../test
(/rest/actor/[test|init]?key=.....&version=....&id=....)
version & ids are only used on init
        if self._args[0] in ('test', 'init'):
            v = self.validateRequestKey()
            if v is not None:
                return v
            if self._args[0] == 'test':
                return self.test()

        # Returns UID of selected Machine
        actorVersion = self._params.get('version', 'unknown')
        service = self.getUserServiceByIds()
        if service is None:
            return Actor.result(_('Unmanaged host'),
error=ERR_HOST_NOT_MANAGED)
        else:
            # Set last seen actor version
            service.setProperty('actor_version',
actorVersion)
            maxIdle = None
            if service.deployed_service.osmanager is not
None:
                maxIdle =
service.deployed_service.osmanager.getInstance().maxIdle()
                logger.debug('Max idle: {}'.format(maxIdle))
            return Actor.result((service.uuid,
                                service.unique_id,
                                0 if maxIdle is None else
maxIdle)
                                )
        raise RequestError('Invalid request')

```

```

    # Must be invoked as '/rest/actor/UUID/[message]', with message
data in post body
def post(self):
    '''
Processes post requests
    '''
    if len(self._args) != 2:
        raise RequestError('Invalid request')

    uuid, message = self._args[0], self._args[1]
    if self._params.get('data') is not None:
        data = self._params['data']
    else:
        data = None

    # Right now, only "message" posts
    try:
        service =
UserService.objects.get(uuid=processUuid(uuid))
    except Exception:
        return Actor.result(_('User service not found'),
error=ERR_USER_SERVICE_NOT_FOUND)

    if message == 'notifyComms':
        logger.debug('Setting comms url to {}'.format(data))
        service.setCommsUrl(data)
        return Actor.result('ok')
    elif message == 'version':
        version = self._params.get('version', 'unknown')
        logger.debug('Got notified version
{}'.format(version))
        service.setProperty('actor_version', version)

    # "Cook" some messages, common to all clients, such as
"log"
    if message == 'log':

```

```

        logger.debug(self._params)
        data = '\t'.join((self._params.get('message'),
six.text_type(self._params.get('level', 10000)))

    try:
        res =
service.getInstance().osmanager().process(service, message,
data, options={'scramble': False})
    except Exception as e:
        return Actor.result(six.text_type(e),
ERR_OSMANAGER_ERROR)

    return Actor.result(res)

```

CONEXIÓN

```

# -*- coding: utf-8 -*-

#
# Copyright (c) 2014 Virtual Cable S.L.
# All rights reserved.
#
# Redistribution and use in source and binary forms, with or
without modification,
# are permitted provided that the following conditions are met:
#
#     * Redistributions of source code must retain the above
copyright notice,
#     this list of conditions and the following disclaimer.
#     * Redistributions in binary form must reproduce the above
copyright notice,
#     this list of conditions and the following disclaimer in
the documentation
#     and/or other materials provided with the distribution.
#     * Neither the name of Virtual Cable S.L. nor the names of
its contributors

```

```
#      may be used to endorse or promote products derived from
this software
#      without specific prior written permission.
#
# THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND
CONTRIBUTORS "AS IS"
# AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
LIMITED TO, THE
# IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A
PARTICULAR PURPOSE ARE
# DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR
CONTRIBUTORS BE LIABLE
# FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR
CONSEQUENTIAL
# DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF
SUBSTITUTE GOODS OR
# SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS
INTERRUPTION) HOWEVER
# CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT,
STRICT LIABILITY,
# OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY
OUT OF THE USE
# OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH
DAMAGE.
```

```
'''
```

```
@author: Adolfo GÃ³mez, dkmaster at dkmon dot com
```

```
'''
```

```
from __future__ import unicode_literals

from django.utils.translation import ugettext as _

from uds.core.util import Config
from uds.core.util.State import State
from uds.core.util.model import processUuid
from uds.core.util import log
from uds.core.managers import cryptoManager
```

```

from uds.models import TicketStore
from uds.REST import Handler
from uds.REST import RequestError
from uds.models import UserService

import datetime
import six

import logging

logger = logging.getLogger(__name__)

# Actor key, configurable in Security Section of administration
interface
actorKey =
Config.Config.section(Config.SECURITY_SECTION).value('Master
Key',

cryptoManager().uuid(datetime.datetime.now()).replace('-', ''),

type=Config.Config.TEXT_FIELD)
actorKey.get()

# Error codes:
ERR_INVALID_KEY = 1
ERR_HOST_NOT_MANAGED = 2
ERR_USER_SERVICE_NOT_FOUND = 3
ERR_OSMANAGER_ERROR = 4

# Enclosed methods under /actor path
class Actor(Handler):
    '''
    Processes actor requests
    '''

```

```

authenticated = False # Actor requests are not authenticated

@staticmethod
def result(result=None, error=None):
    """
    Helper method to create a "result" set for actor response
    :param result: Result value to return (can be None, in
which case it is converted to empty string '')
    :param error: If present, This response represents an
error. Result will contain an "Explanation" and error contains
the error code
    :return: A dictionary, suitable for response to Caller
    """
    result = result if result is not None else ''
    res = {'result': result, 'date': datetime.datetime.now()}
    if error is not None:
        res['error'] = error
    return res

def test(self):
    """
    Executes and returns the test
    """
    return Actor.result(_('Correct'))

def validateRequestKey(self):
    """
    Validates a request key (in "key" parameter)
    """
    # Ensures that key is first parameter
    # Here, path will be .../actor/ACTION/KEY (probably
/rest/actor/KEY/...)
    logger.debug('{} == {}'.format(self._params.get('key'),
actorKey.get(True)))
    if self._params.get('key') != actorKey.get(True):
        return Actor.result(_('Invalid key'),
error=ERR_INVALID_KEY)

```

```

        return None

    def getUserServiceByIds(self):
        '''
        This will get the client from the IDs passed from
parameters
        '''
        logger.debug('Getting User services from ids:
{}'.format(self._params.get('id')))

        try:
            clientIds = [i.upper() for i in
self._params.get('id').split(',')[:5]]
        except Exception:
            raise RequestError('Invalid request: (no id found)')

        services =
UserService.objects.filter(unique_id__in=clientIds,
state__in=[State.USABLE, State.PREPARING])
        if services.count() == 0:
            return None

        return services[0]

    def getTicket(self):
        '''
        Processes get requests in order to obtain a ticket content
GET /rest/actor/ticket/[ticketId]
        '''
        logger.debug("Ticket args for GET:
{0}".format(self._args))

        if len(self._args) != 2:
            raise RequestError('Invalid request')

        try:

```

```

        return Actor.result(TicketStore.get(self._args[1],
invalidate=True))
    except Exception:
        return Actor.result({})

def get(self):
    '''
    Processes get requests
    '''
    logger.debug("Actor      args      for      GET:
{0}".format(self._args))

    if len(self._args) < 1:
        raise RequestError('Invalid request')

    if self._args[0] == 'ticket':
        return self.getTicket()

    if self._args[0] == 'testn': # Test, but without master
key
        return self.test()

    #      if      path      is      .../test
(/rest/actor/[test|init]?key=.....&version=....&id=....)
version & ids are only used on init
    if self._args[0] in ('test', 'init'):
        v = self.validateRequestKey()
        if v is not None:
            return v
        if self._args[0] == 'test':
            return self.test()

    # Returns UID of selected Machine
    actorVersion = self._params.get('version', 'unknown')
    service = self.getUserServiceByIds()
    if service is None:

```

```

        return Actor.result(_('Unmanaged host'),
error=ERR_HOST_NOT_MANAGED)
    else:
        # Set last seen actor version
        service.setProperty('actor_version',
actorVersion)
        maxIdle = None
        if service.deployed_service.osmanager is not
None:
            maxIdle =
service.deployed_service.osmanager.getInstance().maxIdle()
            logger.debug('Max idle: {}'.format(maxIdle))
            return Actor.result((service.uuid,
                                service.unique_id,
                                0 if maxIdle is None else
maxIdle)
                                )
        raise RequestError('Invalid request')

# Must be invoked as '/rest/actor/UUID/[message], with message
data in post body
def post(self):
    '''
    Processes post requests
    '''
    if len(self._args) != 2:
        raise RequestError('Invalid request')

    uuid, message = self._args[0], self._args[1]
    if self._params.get('data') is not None:
        data = self._params['data']
    else:
        data = None

# Right now, only "message" posts
try:

```

```

        service =
UserService.objects.get(uuid=processUuid(uuid))
        except Exception:
            return Actor.result(_('User service not found'),
error=ERR_USER_SERVICE_NOT_FOUND)

        if message == 'notifyComms':
            logger.debug('Setting comms url to {}'.format(data))
            service.setCommsUrl(data)
            return Actor.result('ok')
        elif message == 'version':
            version = self._params.get('version', 'unknown')
            logger.debug('Got notified version
{}'.format(version))
            service.setProperty('actor_version', version)

        # "Cook" some messages, common to all clients, such as
"log"
        if message == 'log':
            logger.debug(self._params)
            data = '\t'.join((self._params.get('message'),
six.text_type(self._params.get('level', 10000))))

            try:
                res =
service.getInstance().osmanager().process(service, message,
data, options={'scramble': False})
            except Exception as e:
                return Actor.result(six.text_type(e),
ERR_OSMANAGER_ERROR)

        return Actor.result(res)

```

ANEXO G: PRESUPUESTO DEL PROTOTIPO VDI

A continuación, se describe el presupuesto económico que se ocupó para la implementación del sistema con todos los elementos adquiridos en Ecuador. En la tabla se detalla cada uno de los componentes que conforman especificando su respectivo precio.

Presupuestó del prototipo VDI nube para la transmisión de datos básico

MÓDULOS	COMPONENTES	CANTIDAD	PRECIO UNITARIO	PRECIO TOTAL
SERVIDOR VDI	Pc core i7	1	1500	1500
	Servidor FreeNAS	1	1000	1000
	Página web alquiler hosting	1	150	150
PRECIO TOTAL DE PRESUPUIESO BASICO				\$ 2650

Realizado por: Huera, B., 2017.

Los costos de implementación proyectados para un requerimiento masivo donde interviene varios factores como el ancho de banda, hardware servidor, hardware de red, entre otro

Costos en un servidor para soportar estudiante 100 estudiantes al mismo tiempo

Presupuestó del servidor plataforma VDI

CANTIDAD	Descripción	Costo individual	Costo total
1	HP ProLiant DL380 Gen9 E5-2660v4 (2) Intel Xeon E5-2660 v4 (14 núcleos, 2,0 GHz, 35 MB, 105W) / 64GB (4 x 16GB) DDR4 2400 MHz RDIMM / Adaptador Ethernet 331FLR de 1 Gb 4 puertos por controlador y; Adaptador Ethernet 560FLR-SFP+ de 10 Gb y 2 puertos por controlador Smart Array P440ar/2G Controller Module /RAID 0/1/1+0/5/5+0/6/6+0)/(8) SFF SAS/SATA HDD bahias/SATA DVD-RW/ 6 PCIe slots / 2 x 800W Flexible Slot Platinum Hot Plug Power Supply / 6 hot plug fans, redundant/ Rack (2U) /3 años en piezas, mano de obra, on site	\$ 8710,00	\$8710,00

2	HPE 32GB 2Rx4 PC4-2400T-R Kit ML350 / DL360 / DL380 / BL460c v4	\$852,00	\$1705,00
2	Disco duro HPE 1.8TB SAS 10K SFF SC 512e DS HDD	\$1557,00	\$3115,00
		Total	\$ 13530,00

Realizado por: Huera, B., 2017.

Presupuestó de PCS VDI

CANTIDAD	Descripción	Costo individual	Costo total
100	HP ProDesk 280 G2 W5Y53LT#ABM SMALL FORM FACTOR 6a generación Intel® Core™ i3-6100 (3,7 GHz, 3 MB de caché, 2 núcleos) Intel® H110 HDD de 1TB 7200 RPM SATA-6 3.5" 4 GB de SDRAM DDR4-2133 (1 x 4 GB) LAN: Realtek RTL8111HSH -CG 10/100/1000M Gigabit Ethernet Unidad Óptica Delgada DVD +/-RW Teclado delgado empresarial USB HP en Español Mouse HP USB. Integrados Intel HD Graphics 530 "2) USB 3.0 (1) Lector Tarjeta SD (1) Auricular y micrófono (6) USB 2.0 (1) Puerto VGA (1) Puerto HDMI (1) RJ-45 (NIC) (1) Entrada de Audio / (1) Salida de Audio" 1) PCIe x16 (1) PCIe x1 (1) Bahía de 3.5" para disco de almacenamiento (1) Bahía de 9.5mm para unidad óptica delgada (1) Bahía para lector de tarjetas multimedia. "	\$ 513,48	\$51348,00
100	FreeDOS 2.0 ,en español 12 meses, 1 año, (1/1/1)	\$0	\$0
100	HP V194 18.5-Inch LED LCD	\$101.74	\$10174,00
		Total	\$ 61522,00

Realizado por: Huera, B., 2017.

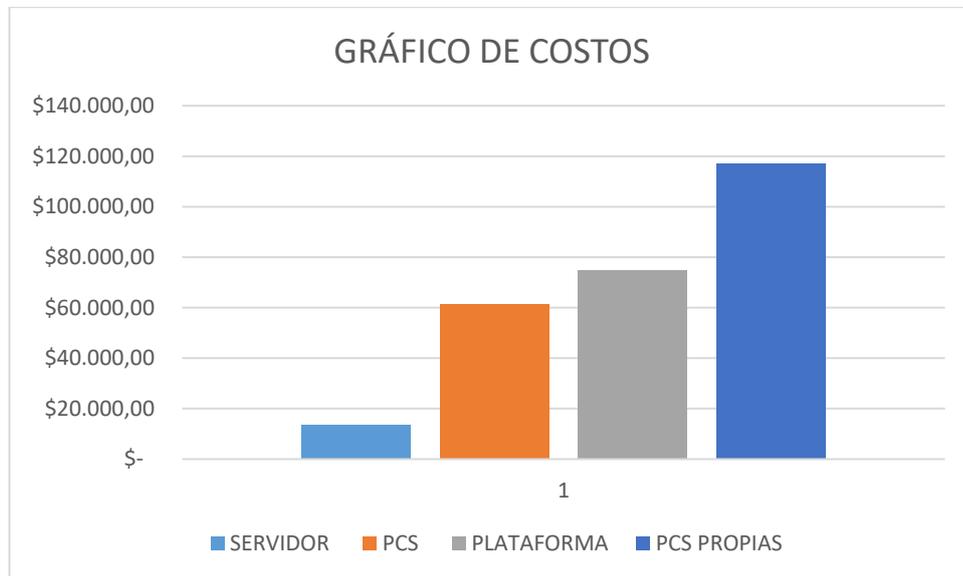
Si se adquiere computadoras individuales a más del sistema operativo costos extras que no serán intervenidos en esta investigación

Presupuestó de PCS VDI

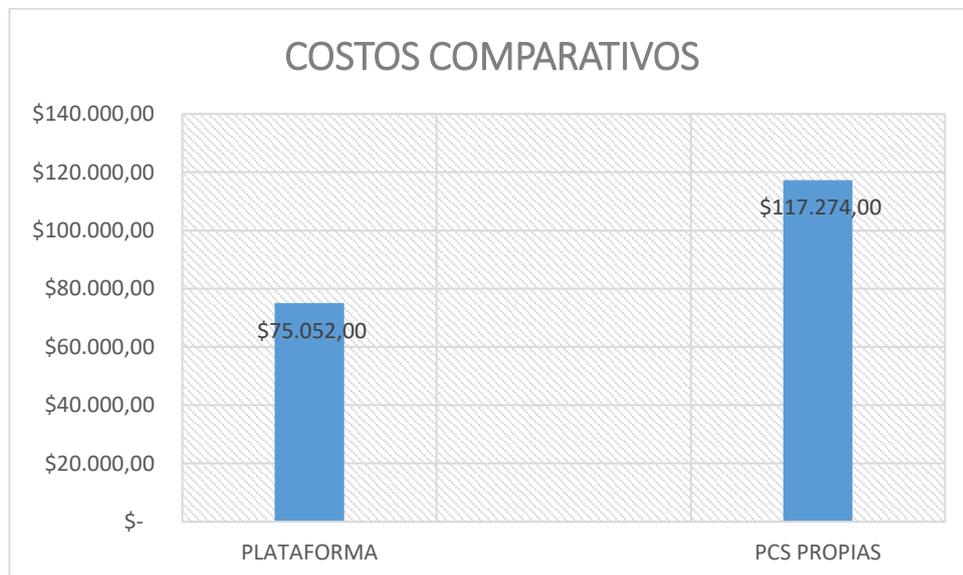
CANTIDAD	Descripción	Costo individual	Costo total
100	<p>HP ProDesk 400 G4 1JF99LA#ABM</p> <p>SMALL FORM FACTOR 7ma Generación Intel Core i5-7500 (3.4 GHz up to 3.8 GHz, 6 MB cache, 4 Núcleos, 4 Hilos) Intel H270 1TB 7200 RPM SATA 6G HDD 4 GB DDR4-2400 nECC (1x4 GB) Realtek RTL8111HSH GbE LOM Network Connection 10/100/1000 Unidad Óptica Delgada DVD +/-RW Teclado delgado empresarial USB HP en Español Mouse HP USB. Intel® HD Graphics 630 (4) USB 3.1 Gen 1 (2 frontales y 2 posteriores); (4) USB 2.0 (posteriores); (1) auricular (frontal); (1) Puerto VGA; (1) Displayport; (1) línea de entrada de audio y (1) línea de salida de audio (posterior); (1) RJ-45 (1) M.2 PCIe x1-2230 (para WLAN); (1) PCIe x16 (v3.0) (cableado como x4); (1) PCIe x16 (v3.0) / (1) Bahía para unidad óptica delgada; (1) Bahía interna de 3.5" /2.5 para disco de almacenamiento</p>	\$ 972	\$97200,00
100	<p>Windows 10 Profesional 64 bits en Español 36 Meses, 3 Años, (3/3/3)</p>	\$99	\$9900
100	HP V194 18.5-Inch LED LCD	\$101.74	\$10174,00
		Total	\$ 117274,00

Realizado por: Huera, B., 2017.

ANEXO H: GRÁFICA DE LOS COSTOS QUE SE PUEDE IMPLEMENTAR EN EL PROYECTO



Realizado por: Huera, B., 2017.



Realizado por: Huera, B., 2017.

Se puede decir que por más que se incremente el número de usuarios, los costos en la plataforma implementada serían mínimos comparados con la utilización de máquinas propias para virtualización, ya que se debe aumentar solo ciertas partes y no toda una infraestructura.

Lo que se plantea es que el estudiante gaste los menos posible para su educación y la universidad provee la máxima comodidad de infraestructura informática, ya sea presencial o a distancia

ANEXO I: CERTIFICACIÓN DE REVISIÓN DE NORMAS TÉCNICAS