

DISEÑO E IMPLEMENTACION DE LA INFRAESTRUCTURA DE UN MODELO PRIVADO DE CLOUD COMPUTING.

Victor Aguirre¹, Joseph Espinoza², Yessenia Sigcha³, Erika Solano⁴

¹ Proyecto de cloud, Universidad Tecnica del Norte, Av. 17 de Julio Ibarra, Imbabura

² Proyecto de cloud, Universidad Tecnica del Norte, Av. 17 de Julio Ibarra, Imbabura

³ Proyecto de cloud, Universidad Tecnica del Norte, Av. 17 de Julio Ibarra, Imbabura

⁴ Proyecto de cloud, Universidad Tecnica del Norte, Av. 17 de Julio Ibarra, Imbabura

vhaguirrec@utn.edu.ec , jsespinozab@utn.edu.ec , yfsigchap@utn.edu.ec , elsolanot@utn.edu.ec

Resumen. *El artículo presenta el proceso realizado para la implementación de un proyecto de Cloud Computing, utilizando gestores de código abierto como OpenStack, Eucalyptus y OpenNebula, los cuales permiten desplegar modelos de Cloud Privado. En la implementación del proyecto de Cloud el proceso se fundamenta en función de características como la arquitectura física de la red, la respectiva infraestructura lógica de cada gestor y el dimensionado de cada una de las plataformas para ofrecer infraestructura como servicio a los diferentes usuarios. Finalmente se da a conocer los resultados obtenidos a través de las diferentes pruebas realizadas a los estudiantes de la Facultad de Ingeniería en Ciencias Aplicadas de la Universidad Técnica del Norte.*

Palabras Claves

Computación en la nube, Software como servicio, virtualización.

Abstract. *The article presents the process realized to implement a cloud computing project, using managers open as OpenStack, Eucalyptus and OpenNebula, which allow private Cloud models deploy code. For the implementation of Cloud Computing project steps core is described, as are the physical architecture of the network and its respective logical architecture of each manager, like how is dimensioned each manager to provide infrastructure as a service to different users. Finally disclosed the results of the various tests on the three platforms to the students of the Faculty of Engineering of Applied Science at the Technical University of the North.*

Keywords

Cloud computing, Software as a service, virtualization.

1. Introducción

En la Universidad Técnica del Norte se lleva a cabo el proyecto de cloud computing orientado a un modelo de cloud privado para ofrecer infraestructura como servicio a los estudiantes y docentes, esta iniciativa nace de los Ingenieros investigadores de la Facultad de Ingeniería en Ciencias Aplicadas y del PhD. José Luis García Dorado.

El avance y desarrollo ha formado una perspectiva en base a una necesidad por parte de los usuarios de manipular la información desde cualquier ubicación basada en el concepto de cloud computing desde el contexto de las IT.

Se planteó realizar como solución el diseño de un Cloud Computing orientado a infraestructura empleando las plataformas libres Openstack, OpenNebula y Eucalyptus. El cual brinde el servicio de máquinas virtuales para los estudiantes de la carrera de Ingeniería en Electrónica y Redes de Comunicación.

En el presente artículo se detallara el diseño e implementación de cada una de las plataformas de software libre al igual que las pruebas de funcionamiento que se realizó para determinar el rendimiento década una de ellas

2. Diseño e Implementación

El modelo de computación denominado Cloud Computing que permite ofrecer escalabilidad de cualquier tipo de recurso virtualizados dentro de la red como un servicio.

2.1 Plataformas

OpenStack

OpenStack es un software de gestión de virtualización de código abierto, que permite a los usuarios conectar distintas tecnologías y componentes de diferentes proveedores y exponen una API unificada, independientemente de la tecnología subyacente. OpenStack, puede administrar diferentes tipos de hipervisores, dispositivos de red y servicios, componentes de almacenamiento, utilizando una única API que crea un tejido de centro de datos unificado, (ORACLE, 2014).



Figura 1. Logo de OpenStack
Referencia: <http://docs.openstack.org/>

OpenStack es la colaboración de desarrolladores y tecnólogos que producen la plataforma de cloud computing estándar abierto tanto para cloud pública como privada. [1](OpenStack, 2015).

Características principales

Este gestor de infraestructura en la nube como lo es OpenStack presenta varias características que se detallan a continuación:

- Su instalación es un tanto complicada, pero a la vez brinda una gran robustez en comparación a los demás gestores y así se obtiene una plataforma óptima y eficiente de cloud.
- OpenStack proporciona una API compatible-EC2, la cual permite interactuar con la API de Amazon EC2; esto es conveniente para entornos multi-cloud donde EC2 es la API común.
- Cuenta con su propio sistema de almacenamiento llamado Swift, el cual está diseñado para proporcionar tolerancia a fallos y escalabilidad.
- Ofrece modelos de redes flexibles para adaptarse a las necesidades de las diferentes aplicaciones o grupos de usuarios. Los modelos estándar incluyen

redes planas o VLAN para la separación de servidores y el tráfico.

- Por medio de su sistema de gestión de la identidad llamado Keystone realiza la autenticación de usuario y del servicio.

Eucalyptus

Eucalyptus es una arquitectura de software basada en Linux que implementa nubes privadas e híbridas escalables dentro de las características de infraestructura de TI.

La plataforma permite utilizar los recursos (hardware, almacenamiento y red) utilizando una interfaz de autoservicio en función de las necesidades, que se requieren dentro de las actividades determinadas. [1]



Figura 2. Logo Oficial Eucalyptus
Referencia: Enterprise Edition 2.0

Eucalyptus es una plataforma de computación en la nube compatible con EC2 y almacenamiento en la nube compatible con S3. [2]

Características principales

- Eucalyptus ofrece compatibilidad con la API de Amazon EC2, S3, IAM, ELB, Auto Scaling y servicios CloudWatch. Que le ofrece la capacidad de una nube híbrida.
- El Control de direcciones IP permite la asignación IPs a las VMs en forma dinámica, elástica y controlada. [3]
- Es implementada como Infraestructura de Servicio (IaaS), donde los usuarios se proveen de capacidad de procesamiento, almacenamiento y recursos computacionales.[4]
- La alta flexibilidad que posee.
- En la implementación permite realizar escalabilidad en su desempeño.
- Las instancias de Máquinas Virtuales se ejecutan a través de un software llamado hypervisor o VMM (Virtual Machine Monitor).

OpenNebula

OpenNebula es una plataforma de Cloud Computing bajo software libre y de código abierto para la gestión distribuida de la infraestructura de un centro de datos capaz de gestionar servidores virtuales para el despliegue de implementaciones privadas, públicas e híbridas de infraestructura como servicio.



Figura 3. Logo de OpenNebula

Referencia:

<https://www.udsenderprise.com/es/blog/2016/02/22/uds-enterprise-soporta-opennebula/>

Características Principales

- OpenNebula es un software open-source que permite construir cualquier tipo de Cloud.
- Está diseñado para lograr una integración con cualquier tipo de red y almacenamiento, para así adaptarse a los centros de datos existentes.
- OpenNebula posee la capacidad de gestionar el almacenamiento, las redes y las diferentes tecnologías de virtualización existentes en la actualidad.
- Proporciona la capacidad de desplegar servicios en infraestructuras distribuidas, combinando recursos de centros de datos así como de Clouds remotos, de acuerdo con las políticas de despliegue.

2.2 Infraestructura Física

Como parte inicial de la implementación de las diferentes plataformas de cloud se explica la infraestructura física que posee la Universidad Técnica del Norte determinando los diferentes elementos que intervendrán en el acceso tanto interno como externo a las diferentes plataformas a instalarse, como se puede observar en la **Figura 4** cada servidor en el cual se implementaran las plataformas esta instalados en el Datacenter de Facultad de Ingeniería en Ciencias Aplicadas (FICA) los cuales se conectan mediante el switch de distribución del Datacenter hacia el edificio central el cual proporciona la salida hacia internet y la conexión con las demás subredes de la universidad

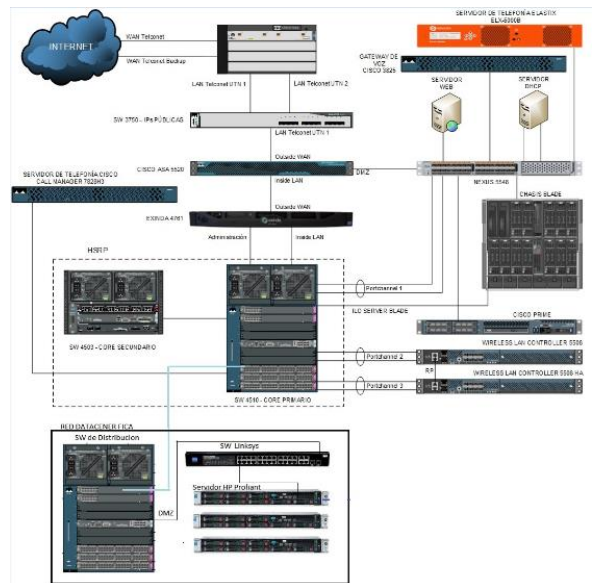


Figura 4. Infraestructura física y sus respectivos componentes

2.3 Arquitectura Lógica

Openstack

En la Figura 3 se presentan los sistemas y componentes con los que cuenta Openstack los cuales cumplen con funcionalidades tan diferentes como: el almacenamiento de volúmenes, la ejecución de máquinas virtuales; estos elementos se comunican entre ellos para suministrar Infraestructura como servicio

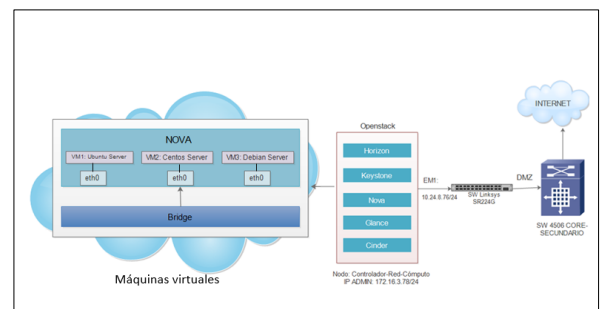


Figura 5. Arquitectura lógica de Openstack

Fuente: Investigación Propia.

Nodo: La arquitectura en la cual se implementó la Infraestructura como Servicio, consta de un solo nodo (“All in One”), en el cual se alojarán los servicios de la plataforma Openstack como: Horizon, Keystone, Nova, Glance y Cinder; manejando a su vez todas las máquinas virtuales. Este nodo hará la función de Controlador, Cómputo y Red.

Nodo Controlador: Instalados los componentes Keystone (Identificación), Glance (Imágenes de SO) y Horizon (Dashboard), Nova que se encarga de la parte de

gestión de los servicios de cómputo, además los servicios asociados a las API y la base de datos MySQL.

Nodo de Cómputo: Se encuentra instalado el componente Nova, el encargado de hacer funcionar las máquinas virtuales, también ejecuta el servicio de dos hipervisores el KVM o Quemu.

Nodo de Red: Conecta todos los elementos de Red, aquí se ejecutan varios agentes como el plugin-in-agent el cual se encarga del enrutamiento, de configurar direcciones DHCP¹, y también provee el servicio de internet.

Máquinas virtuales: Cuentan con una interfaz virtual de red (eth0), la misma que se encuentra conectada a la interfaz física del host anfitrión la em1 a través de una segunda interfaz lógica conocida como puente (Bridge), esta comunicación es gestionada por el componente Nova.

Eucalyptus

Eucalyptus se encuentra desarrollado en forma modular, dando lugar a los cinco principales componentes que serán expuestos con su alto nivel de funcionalidad dentro de la arquitectura de Eucalyptus, como se observa en la figura 6:

- Controlador de la Nube (CLC)
- Controlador de Cluster (CC)
- Node Controller (NC)
- Storage Controller (SC)
- Walrus

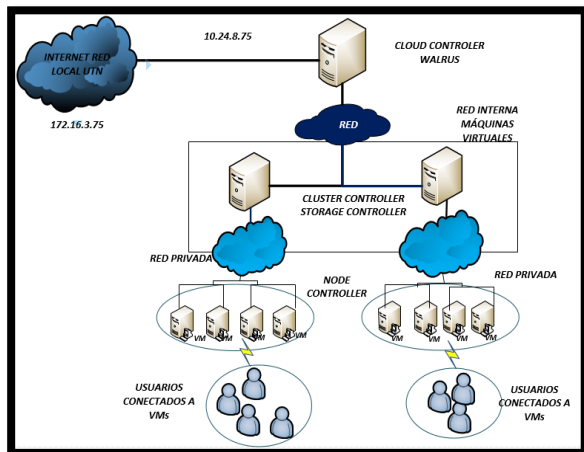


Figura 6. Arquitectura interna de Eucalyptus y sus respectivos componentes
Referencia: (HELION, 2013)

• Controlador de la nube (CLC)

El controlador de la nube cumple su desempeño en la infraestructura efectuando el monitoreo de las instancias y recursos de la nube.

- Monitorear la disponibilidad de recursos en diversos componentes de la infraestructura de nube.
- Realizar el monitoreo de las instancias que se están ejecutando y recursos de la nube.

Controlador de Cluster (CC)

El Controlador de Cluster (CC) recolecta la información del estado de los nodos en la plataforma y efectúa la ejecución de las instancias sobre ellos hasta el CLC.

• Controlador de Nodos (NC)

La funcionalidad del Controlador de Nodos (NC) se fundamenta en la ejecución de las máquinas que aloja las instancias. El NC controla las actividades de las VM, es decir, su ciclo de vida como:

- ✓ Ejecución
- ✓ Inspección
- ✓ Terminación

• Storage Controller (SC)

El Storage Controller en la infraestructura permite gestionar las instancias en Eucalyptus y para el acceso a:

- ✓ Almacenamiento
- ✓ Volúmenes

• Walrus

Proporciona la prioridad para la compatibilidad de la interfaz de S3 y compatibilidad de EBS y almacena las imágenes de las instancias.

Open Nebula

En la siguiente imagen se describen los bloques funcionales que representarán los elementos correspondientes a la estructura física de nuestra plataforma donde se especifica los bloques de:

¹ DHCP: Es el Protocolo de Configuración Dinámica de Host

- Almacenamiento
- Ejecución de máquinas virtuales
- Infraestructura de red
- Frontend

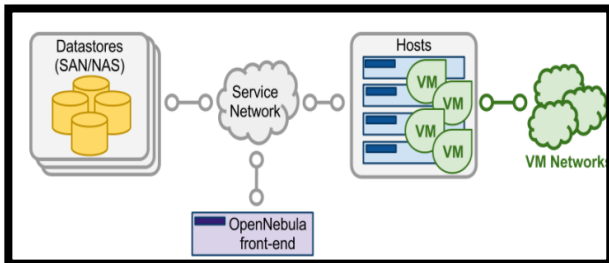


Figura 7. Arquitectura funcional de OpenNebula
Referencia:

http://docs.opennebula.org/4.14/_images/one_high.png

- **El Front-end:** es el servidor donde se ejecutan todos los servicios de gestión, provisión, monitorización, acceso, y administración de la plataforma. Es la parte central del sistema.
- **Los Hosts:** son los servidores que ejecutan las funciones de virtualización donde se instancia y ejecutará las máquinas virtuales (VM) proporcionadas por la plataforma.
- **Los Datastores:** son los servidores donde se almacenan las imágenes correspondientes a las instancias y máquinas virtuales. Los datastores son capaces de almacenar varios formatos de discos, así como ficheros de configuración, kernels, volúmenes de almacenamiento, etc.
- **El Service Network:** es la parte del servicio de red que permite la interconexión de los Hosts con los Datastores y el Frontend, además es la comunicación hacia nuestra red privada o red institucional y nos proporciona una conexión a internet.
- **Virtual Networks:** es la infraestructura virtual de red que permite soportar los servicios de red entre las máquinas virtuales (VM).

Dimensionamiento

El proceso de dimensionamiento, permite analizar los recursos que se posee, para establecer características de la capacidad del equipo y someter a pruebas en base a la ejecución de aplicaciones orientadas a las materias sobre las instancias.

La implementación se la realizo a partir del recurso físico presente, para cada una de las plataformas que se pueden observar en la tabla 1.

NOMBRE SERVIDOR		HP DL360 G9	HP DL360 G9	HP DL360 G9
Características	RAM	32 GB	32 GB	32 GB
	PROCESADOR	1x Intel Xeon E5-2620 V3	1x Intel Xeon E5-2620 V3	1x Intel Xeon E5-2620 V3
	DISCO DURO	3x450 GB	3x450 GB	3x450 GB
	TRAJETA DE RED	4 X 1GBE	4 X 1GBE	4 X 1GBE
	SIST. OPERATIVO	Ubuntu Server 14.04 LTS	CentOS 6.5	Ubuntu Server 14.04 LTS
FUNCIÓN		Cloud OpenStack	Cloud Eucalyptus	Cloud Open Nebula
IMAGEN				

Tabla 1. Características de los servidores físicos
Fuente: Elaboración Propia.

Las plataformas, permiten la compatibilidad con los siguientes sistemas operativos como se muestra en la tabla 2, tales como:

SOFTWARE	LINUX	WINDOWS
Plataforma	Si existe soporte en: -Opensuse	Si existe soporte en : -Windows 7 -Windows 8
Openstack	-CentOS	-Windows 8 server 32 bits
Eucalyptus	-Cirros	-Windows XP
Open nebula	-Ubuntu -Debian -Fedora	

Tabla 2. Soporte de plataforma con los sistemas operativos
Fuente: Elaboración Propia.

2.4 OpenStack

El dimensionamiento para la implementación de la plataforma de OpenStack está basado en requisitos de los diferentes sistemas operativos que se serán instalados para el uso de las aplicaciones que necesitan los estudiantes de la Facultad de Ingeniería en Ciencia Aplicadas de la Universidad; en la tabla 3 se puede observar dichos requisitos.

Sistema Operativo	Memoria RAM	Disco Duro		Procesador
		Mínimo	Máximo	
Ubuntu Server	128 MB	500 MB	1 GB	x86 – 2.4 GHz

CentOs server	512 MB	1 GB	2 GB	x86 - 1 GHz x64 - 2 GHz
Debian server	1 GB	1 GB	2 GB	x86 - 2 GHz

Tabla 3. Requisitos de los Sistemas Operativos a implementar en OpenStack
Fuente: Elaboración propia.

Una vez que se conoce los requisitos que se necesita para la instalación de los diferentes sistemas operativos, se procede a verificar las características principales que tiene el servidor físico donde se implementará la plataforma de OpenStack, como se detallan en la tabla 1.

Para poder dimensionar el número exacto de las máquinas virtuales a implementar en la plataforma de OpenStack, se necesita saber con cuantos CPU cuenta el servidor físico; por lo cual se ejecuta el comando “*lscpu*” para ver los detalles del procesador, como se indica en la figura 8.

```

Arquitectura:          x86_64
CPU op-mode(s):      32-bit, 64-bit
Orden de bytes:      Little Endian
CPU(s):              16
On-line CPU(s) list: 0-15
Hilo(s) por núcleo:  2
Núcleo(s) por socket: 8
Socket(s):           1
Nodo(s) NUMA:       1
ID del vendedor:     GenuineIntel

```

Figura 8. Información del número de cores y sockets en el servidor físico.

Fuente: Terminal del Servidor Ubuntu Server 14.04 LTS

De acuerdo a la figura 8, se puede observar que el servidor donde se alojará la plataforma OpenStack cuenta con un socket, 8 núcleos y 16 CPUs que serán usados para la implementación de las diferentes máquinas virtuales.

Número de máquinas virtuales

Para realizar el cálculo del número de máquinas virtuales que pueden crearse y ejecutarse simultáneamente, se emplea la fórmula del libro “OpenStack Operations Guide” (Tom y otros, 2014).

$$\#VM = \frac{\#CPUs \times Vel \text{ de } VM}{Vel \text{ Procesador}} \quad (1)$$

Donde: **#VM:** Número de máquinas virtuales a trabajar simultáneamente.

#CPUs: Número de CPUs que tiene el servidor físico que alojara a la plataforma OpenStack.

Vel. de VM: Velocidad del procesador en GHz de la máquina virtual.

Vel Procesador: Velocidad del procesador en GHz del servidor físico.

De acuerdo a la ecuación 1, y datos obtenidos en la figura 4, se establece que el número máximo de máquinas virtuales a ejecutarse simultáneamente es de 16.

$$\#VM = \frac{16 \times 2.4 \text{ GHz}}{2.4 \text{ GHz}} = 16VM$$

Memoria RAM

Este cálculo se lo hace de acuerdo a la cantidad de memoria RAM que necesita cada Sistema Operativo para ser instalado en el host virtual y trabajo de manera adecuada.

$$\#VM = \frac{RAM_T}{RAM_SO} \quad (2)$$

Dónde: **#VM:** Número de máquinas virtuales que se puede crear de acuerdo a la cantidad de memoria RAM.

RAM_T: Cantidad total de memoria RAM del servidor físico

RAM_SO: Memoria RAM requerida por el sistema operativo de la máquina virtual.

De acuerdo a los valores detallados en la tabla 3 y de la ecuación 2, se procede a realizar el dimensionamiento del número de máquinas virtuales que pueden ser creadas en relación al consumo de memoria RAM de cada sistema operativo.

RAM TOTAL	RAM S.O.	#VM	Sistema Operativo
	1 GB	32	Ubuntu-Server
32 GB	1 GB	32	CentOs-Server
	1 GB	32	Debian-Server

Tabla 4. Número de VM de acuerdo a la memoria RAM

Fuente: Elaboración propia.

Los resultados obtenidos en la tabla 4, indican que se pueden crear hasta 32 máquinas virtuales en relación al

consumo de memoria RAM que tiene cada sistema operativo.

De acuerdo a la plataforma de OpenStack, el dimensionamiento de las máquinas virtuales se lo realiza en relación a los flavor que nos proporciona al momento de llenar la plantilla de creación de instancias, cuyo valor es 2 GB de memoria RAM. Ahora con este valor se procede a dimensionar el número de VM que se pueden crear, mediante la ecuación 2.

RAM TOTAL	RAM FLAVOR	#VM	Sistema Operativo
32 GB	2 GB	16	Ubuntu-Server
	2 GB	16	CentOs-Server
	2 GB	16	Debian-Server

Tabla 5. Número de VM de acuerdo a los flavors de la OpenStack

Fuente: Elaboración propia.

Los resultados obtenidos en la tabla 5, indican que cada máquina virtual está asignada una memoria RAM de 2 GB, mientras que el equipo físico posee 32 GB total, por lo cual se puede dimensionar un total de 16 instancias que pueden ser de cualquier sistema operativo ya que cada una consume el mismo recurso.

Cálculo de Procesadores

Este cálculo se realiza para determinar el número de CPUs virtuales que se necesita por cada sistema operativo a implementar en las máquinas virtuales.

$$\#VCPU = \frac{Vel_Procesador}{Vel_VM} \quad (3)$$

Donde: **#VCPU**: Número de CPUs por VM.

Vel_Procesador: Frecuencia en GHz del procesador del servidor físico

Vel_VM: Frecuencia en GHz con la que trabajará el sistema operativo en la máquina virtual.

Con los datos obtenidos en la tabla 3 y la ejecución de la ecuación 3, se logra determinar el número de CPUs que necesita cada sistema operativo para un normal funcionamiento.

Velocidad Procesador	Velocidad VM	# VCPUs	Sistema Operativo
2.4 GHz	2.4 GHz	1	Ubuntu-Server
	2.0 GHz	1.2	CentOs-Server
	2.0 GHz	1.2	Debian-Server

Tabla 6. Número de VCPUs por VM

Fuente: Elaboración Propia

En la tabla 6, se puede observar que se necesita por lo menos un VCPU por cada máquina virtual independientemente del sistema operativo que se ejecute en la misma.

Una vez que se tiene calculado el número de VCPUs que se necesita por cada sistema operativo, se procede a realizar el dimensionamiento de cuantas máquinas virtuales se pueden crear.

$$\#VM = \frac{CPUs}{VCPU} \quad (4)$$

Donde: **#VM**: Número de Máquinas virtuales.

#VCPU: CPU virtuales asignados por cada Sistema Operativo

#CPU: CPUs con los que cuenta el servidor físico.

Mediante los resultados obtenidos en la tabla 6 y la ecuación 4, se procede a calcular el número de máquinas virtuales que se pueden crear en función de VCPUs.

#CPUs	#VCPUs	#VM	Sistema Operativo
16	1	16	Ubuntu-Server
	1	16	CentOs-Server
	1	16	Debian-Server

Tabla 7. Número de VM en función al número de VCPUs

Fuente: Elaboración Propia.

Los resultados obtenidos en la tabla 7, indican que se pueden crear un total de 16 máquinas virtuales, ya sea el sistema operativo que necesite en vista que cada uno consume un VCPU por instancia.

A continuación, en la tabla 8 se puede observar un pequeño resumen, donde se detalla el número de máquinas que se pueden crear de acuerdo a los diferentes puntos tratados anteriormente.

RAM TOTAL	CPUs TOTAL	#VM por Fórmula	#VM por RAM	#VM por VCPU
32 GB	16	16	16	16

Tabla 8. Resumen del número de VM a crear en OpenStack
Fuente: Elaboración Propia.

En la tabla 8, se mira que el número máximo de creación de máquinas virtuales es 16, en relación al uso de fórmula de OpenStack, consumo de memoria RAM y el número de VCPUs que se emplea en cada instancia sea cual sea el sistema operativo Linux que se ejecute.

2.5 Eucalyptus

Durante el proceso de dimensionamiento, se requiere realizar mediciones del recurso actual y del recurso consumido por la plataforma.

Análisis de Ancho de Banda a través de EXINDA

Controla y gestiona el acceso Web por usuario o host, el tráfico punto a punto de grupos de usuarios, servidores o individuos.

A través del monitoreo por medio del servidor Exinda que posee la Universidad Técnica del Norte, aportó resultados en base a los parámetros medidos en tiempo real, expuestos en la tabla 9.

Análisis de tráfico	Taza de transferencia (Kbps)	Porcentaje
Ejecución de servidor	1,634	7%
Ejecución de 5 instancias	10,549	43%
Ejecución de 10 instancias	23.969	95%

Tabla 9. Relación de datos Exinda
Fuente: Elaboración propia.

Análisis de memoria RAM

El análisis del parámetro de la memoria RAM es mostrar un resultado de soporte del servidor en base al número de aplicaciones y máquinas virtuales que se ejecuten sobre la plataforma de cloud denominada Eucalyptus.

En base al análisis realizado bajo el consumo de memoria RAM del servidor y fundamentada en la plataforma, se observa el resultado en la tabla 10:

# de Instancias Windows7	Consumo de RAM	Valor Total
Inicial	9206	31985
5	14626	31985
10	31675	31985

Tabla 10. Resultado del consumo de RAM

Los resultados se pueden observar en el figura 10, donde se estableció un cuadro estadístico en función de los resultados.

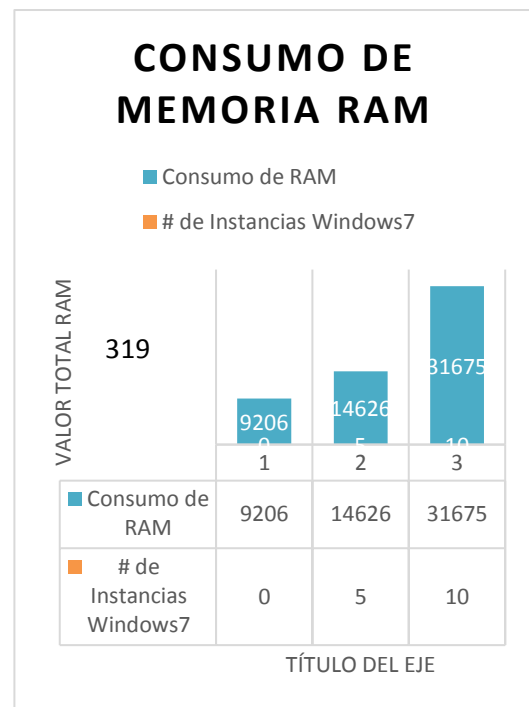


Figura 9. Diagrama estadístico de consumo de RAM

Referencia: Elaboración propia basada en análisis

Nota: El valor expresado en 0, determina el consumo base del sistema operativo en función de la ejecución de la plataforma.

Resultados de consumo de memoria RAM, en las instancias.

El resultado observado en la tabla 11 se obtuvo del análisis en base al consumo de memoria RAM, al ser sometido la plataforma durante la ejecución de un número de instancias.

Nota: GNS3 es una herramienta de alto rendimiento que requiere mayor recurso.

ss

Instancias Windows7	Consumo de RAM	Memoria RAM sin uso	Valor Total
---------------------	----------------	---------------------	-------------

Ejecución SS.OO	1,1GB	0,9GB	2GB
GNS3 y Packet Tracer	1,87GB	0,13GB	2GB

Tabla 11. Resultado del consumo de memoria RAM de la instancia

Consumo de memoria RAM en función del valor pico

RAM aplicaciones= 1,87GB

RAM 1vm= 2 GB

RAM 10vm=2GB*10 =20 GB

RAM total=20 GB

Las instancias se crearon bajo el recurso de 2GB de memoria RAM, datos expuestos en la tabla 11, por lo tanto se establece un diagrama en base al consumo del recurso de las por medio de un diagrama estadístico como se observa en la figura 11.

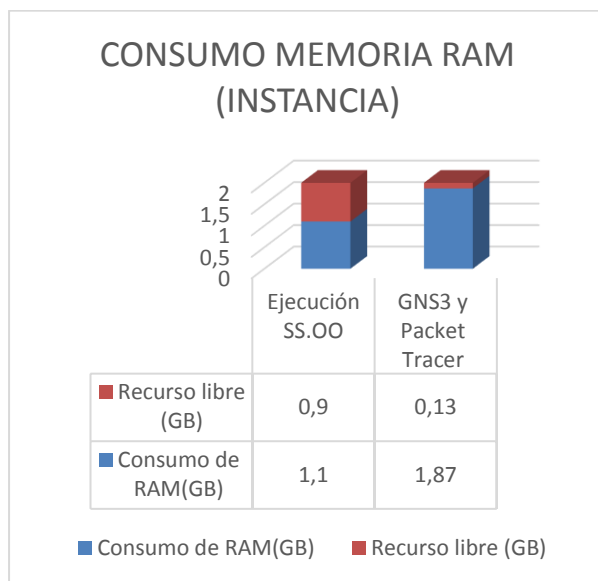


Figura 10. Consumo de memoria RAM en las instancias
Referencia: Elaboración propia basado en análisis.

Consumo de CPU

El consumo de CPU consiste en un análisis del sistema, por medio de herramientas que permitan determinar el uso del CPU y su utilización.

Por medio, de top provee una vista en tiempo real del sistema, a continuación se procede a obtener el dato durante la ejecución de la plataforma.

Load average: 0.14; 0.18; 0.18

Consumo de CPU: 2.3%us+0.6%sy+0.0%wa= 2.9%

Determina el 2.9% del consumo de CPU en el servidor, donde se ejecuta la plataforma.

La ejecución de 5 instancias, utilizadas en las aplicaciones destinadas para networking.

Load average: 1.20; 1.38; 1.04

Consumo de CPU: 25.2%us+6.3%sy+5.8%wa= 37.3%

Determina el 37.3% del consumo de CPU en el servidor, donde se ejecuta la plataforma.

Al ejecutarse 10 instancias, además que se encuentran utilizando las aplicaciones destinadas para networking.

Load average: 6.10; 6.16; 5.08

Consumo de CPU: 54.6%us+7.8%sy+0.4%wa= 62.8%

Determina el 62.8% del consumo de CPU en el servidor, donde se ejecuta la plataforma.

Análisis del resultado del Consumo del CPU

En función de los resultados obtenidos, a través del monitoreo y según los parámetros establecidos, determina que el consumo de CPU como se observa en la tabla 12, se encuentra relacionado en base a la ejecución de aplicaciones y funciones de las instancias, por lo tanto, es proporcional a su uso.

# de instancias	Consumo de CPU (%)	Consumo de CPU sin uso (%)
Valor inicial	2,9	97,1
5	37,8	62,2
10	62,8	37,2

Tabla 12. Análisis del consumo de CPU de acuerdo al número de instancias.

Referencia: Elaboración propia basada en análisis.

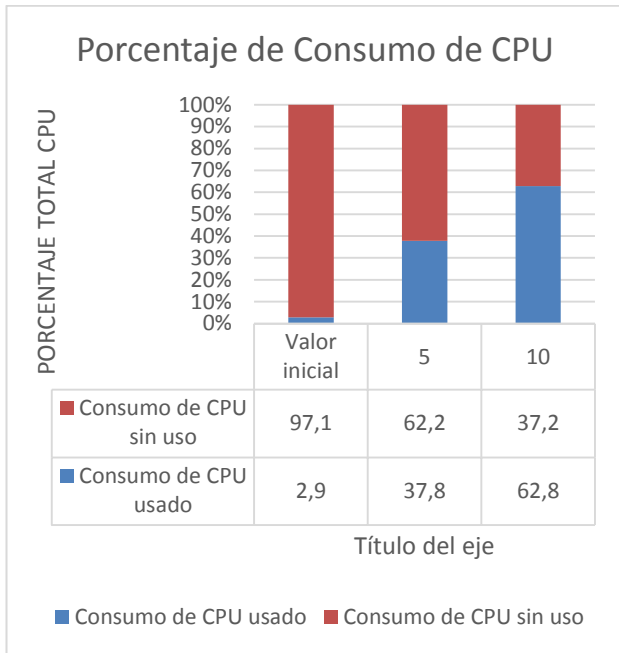


Figura 11. Porcentaje de consumo de CPU
Referencia: Plataforma Eucalyptus SA.

El valor máximo de consumo de CPU alcanzado fue de 62,8% sobre el 100% proporcionado. Este valor varía de acuerdo al uso de las instancias por parte del estudiante o usuario, que se encuentre desempeñando algún tipo de aplicación, como se observa en la figura 12.

Consumo de CPU en las instancias

Se realiza el análisis del parámetro de medición del consumo de CPU en base al recurso de la instancia y durante la ejecución de las aplicaciones.

Las instancias en base al consumo de CPU, varían de acuerdo a las aplicaciones o actividades a la que las instancias se encuentran sometidas, por lo tanto, se establece resultados en base al consumo de CPU de las instancias, como se observa en la tabla 13.

Instancias Windows7	Consumo de CPU (%)
Ejecución SS.OO	0-5
GNS3 y Packet Tracer	52-60

Tabla 13. Consumo de CPU en la instancia

Pruebas de Funcionamiento

Las instancias creadas en base a la Plataforma de Eucalyptus, parte fundamental del estudio de cloud computing que serán empleadas para las diferentes funcionalidades. Aspectos de aplicabilidad como es el caso

de ofrecer determinado número de instancias orientadas a una función específica dentro de la Facultad de Ingeniería en Ciencia Aplicadas de la Universidad Técnica del Norte.

En el caso de la Plataforma de Eucalyptus se orienta a brindar servicios en base al uso de herramientas de Networking, proporcionando al aspecto académico de resolver actividades en función de la satisfacción del estudiante y docente.

2.6 OpenNebula

2.6.1 Dimensionamiento de memoria RAM y CPU

En la siguiente tabla se ha especificado los requerimientos de cada sistema operativo analizado, además se estableció las características del servidor y de acuerdo a esto se logró obtener el número máximo de máquinas virtuales que se puede desplegar en la plataforma, para esto se divide el número de núcleos del servidor para el numero de núcleos necesarios de cada sistema de igual manera se hace con la memoria RAM y el valor menor de los 2 resultados corresponde al máximo de máquinas virtuales que se puede desplegar de cada sistema operativo.

Sistema Operativo	Requerimientos	
	Procesamiento(GHz)	Memoria(GB)
Ubuntu	1	1
Centos	2	1
LinuxMint	1	1
OpenSuse	1	512MB
Fedora	2	2
Windows XP	0.6	512MB
Windows 7	1.2	2
Windows server	2.4	2

Tabla 14. Requerimientos de los sistemas Operativos
Referencia: Elaboración Propia

$$N^{\circ} \text{ maquinas virtuales} = \frac{\text{Vel CPU} * \text{N}^{\circ} \text{ de nucleos del servidor}}{\text{Vel CPU Sistema Operativo}} \quad (1)$$

$$N^{\circ} \text{ maquinas virtuales} = \frac{\text{Memoria RAM del servidor}}{\text{Memoria ram necesario}} \quad (2)$$

De acuerdo a las ecuaciones 1 y 2 se puede determinar el número máximo de máquinas virtuales que se pueden desplegar de acuerdo a los requisitos de cada sistema operativo y a los recursos que posee el servidor.

En la tabla 14 se detalla los recursos que posee el servidor, en este caso se ha tomado en cuenta los recursos básicos para la ejecución de la plataforma OpenNebula por lo que para el dimensionamiento se tomara 15 núcleos de 2,4 Ghz y 30 Gb de memoria RAM, determinando así de cada uno de los diferentes sistemas Operativos el número máximo de máquinas virtuales que se es posible ejecutar al mismo tiempo.

Sistema Operativo	Recursos Actuales		MV Posibles
	Núcleos CPU(2.4GHz)	Memoria(GB)	
Ubuntu	15	30	30
Centos	15	30	18
LinuxMint	15	30	30
OpenSuse	15	30	30
Fedora	15	30	15
Windows XP	15	30	60
Windows 7	15	30	15
Windows server	15	30	15

Tabla 15. Numero de Máquinas virtuales posibles
Referencia Elaboración Propia

2.6.2. Dimensionamiento de Ancho de Banda

Se realiza un análisis del tráfico de entrada y salida para determinar el total de Ancho de Banda (AB) que requiere un usuario para acceder a la plataforma y a la máquina virtual y a partir de esto determinar de acuerdo al número de usuarios cuanto es el AB que se debe asignar a las subredes dentro de la Universidad Técnica del Norte y el AB mínimo que se debe tener en la conexión a internet.

El monitoreo se lo realiza mediante el equipo Exinda que se encuentra instalado en la UTN, en la figura 13 se puede observar que el consumo o de AB de entrada hacia el servidor OpenNebula es de 13 kbps y en la figura 14 se muestra que el tráfico de salida es de 11kbps para un usuario por lo que se realiza un análisis que para el número de estudiantes en un aula de clases que en promedio es de 40 y se determina que el AB mínimo es de 991 kbps para garantizar el acceso a la plataforma OpenNebula

Inbound Conversations				
External IP (User)	Internal IP (User)	Application	Transfer Rate (kbps)	Packet Rate (pps)
Total			13.642	25
FICA-WIRELESS: Auto Catch-all				
172.16.3.74	172.17.42.12	WebSocket[cloudfica.utm.edu.ec:29876]	7.682	14
172.16.3.74	172.17.42.12	Yahoo	5.960	10

Figura 12. Tráfico de entrada en el servidor OpenNebula
Fuente: Elaboración Propia

Outbound Conversations				
External IP (User)	Internal IP (User)	Application	Transfer Rate (kbps)	Packet Rate (pps)
Total			11.143	16
FICA-WIRELESS: Auto Catch-all				
172.16.3.74	172.17.42.12	WebSocket[cloudfica.utm.edu.ec:29876]	7.385	10
172.16.3.74	172.17.42.12	Yahoo	3.758	5

Figura 13. Trafico de Salida del servidor OpenNebula
Fuente: Elaboración Propia

3. Resultados

3.1 Openstack

Para las pruebas de funcionamiento de la plataforma Openstack se realizó la medición de la memoria RAM y el

consumo del CPU en el servidor físico durante la ejecución simultánea de 5, 10 y 15 máquinas virtuales con Sistema Operativo Ubuntu Server.

En la Tabla 14, se muestra el resultado del consumo total de memoria RAM de las máquinas virtuales y el asignado a cada una de acuerdo a los parámetros establecidos de cada Sistemas Operativo, se puede concluir que el de acuerdo a los recursos asignados a cada instancia el consumo de memoria RAM y de procesamiento es bajo, debido a que no se está trabajando con aplicaciones que exijan de mayor cantidad memoria RAM en cada máquina virtual.

Sistema operativo	Número de Máquinas virtuales	Memoria RAM Asignada	Consumo memoria RAM	
			Por VM	Total
Ubuntu Server	5	2 GB	1.4 GB	7 GB
	10	2GB	1.7 GB	17 GB
	15	2GB	1.7 GB	26 GB

Tabla 16. Consumo de memoria RAM en 5, 10 y 15 VM
Fuente: Elaboración propia

La Tabla 15, muestra el consumo total de procesamiento en el host anfitrión utilizado por las máquinas virtuales y el asignado a cada uno de los host virtuales de acuerdo a los parámetros establecidos en cada Sistemas Operativo, se puede concluir que de acuerdo a los recursos asignados a cada instancia el consumo de VCPUs es bajo, debido a que no se está trabajando con aplicaciones que requieran de mayor procesamiento en cada máquina virtual.

Sistema operativo	Número de Máquinas virtuales	VCPU asignados	Consumo de VCPUs	
			Por VM	Total
	5	1 VCPU	1 VCPU	5 VCPUs

Ubuntu Server	10	1 VCPU	1 VCPU	10 VCPUs
	15	1 VCPU	1 VCPU	15 VCPUs

Tabla 17. Consumo de VCPU en 5, 10 y 15 VM
Fuente: Elaboración propia

3.2 Eucalyptus

Análisis de resultados

Resultado de análisis de parámetros de dimensionamiento, como se puede observar en la tabla 16.

Parámetros	Consumo de Equipo (Servidor)	Valor a Implementar por maquina
Ancho de Banda	24Kbps	50 Kbps
Memoria RAM	1,89GB	2GB
Consumo de CPU	62,8%	80%
Estadística de Disco	18% sobre 90%	-----

Tabla 18. Resultado de requerimiento de maquinas
Fuente: Elaboración propia

En base a las pruebas se obtuvo resultados bajo los parámetros de dimensionamiento, donde determina, que Eucalyptus como plataforma de cloud privado, permitió ejecutar aplicaciones en función de la materia de networking como: GNS3, Packet Tracer.

Además de especificar recursos que satisfacen los requisitos de las instancias, mediante las pruebas pertinentes que mejoren las capacidades de ejecución, al equipo físico al ser sometido, nos proporciona la capacidad de crear 8 máquinas en función de windows7 con las características que se muestran en la tabla 17, que permiten cumplir los parámetros de ejecución de aplicaciones dirigidas a networking, en base a los recursos.

Parámetros	SS.OO	# instancias
Numero de maquinas	Windows	10
	Linux	4

Tabla 17. Resultado de requerimiento de maquinas
Fuente: Elaboración propia

3.3 OpenNebula

Para las pruebas de rendimiento se realiza la ejecución de 5, 10, 15 máquinas virtuales tanto del sistema operativo Linux como de Windows determinando el consumo de CPU

y memoria cuando las máquinas están en reposo y cuando se encuentran siendo usadas.

3.3.1. Sistema Linux

En la figura 13 se puede observar el consumo de un sistema operativo Linux el cual en reposo genero un consumo mínimo de 1% del total del servidor, en ejecución el consumo subió pero de igual manera no rebaso los recursos asignados manteniéndose en un funcionamiento normal.

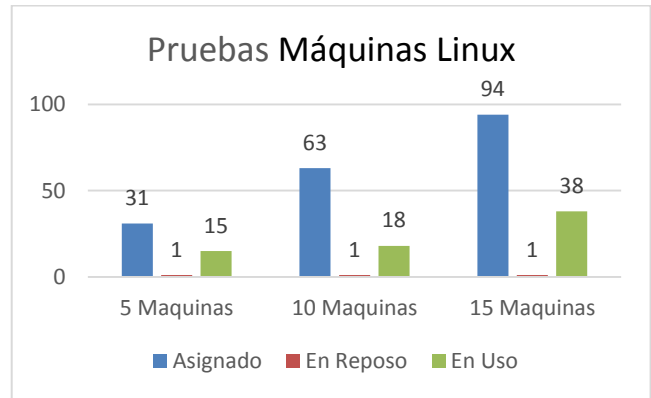


Figura 14. Consumo de CPU en sistemas Linux
Fuente: Elaboración Propia

En la figura 14 se muestra el consumo de memoria RAM del servidor al estar cada máquina virtual en reposo y en ejecución y se determinó el correcto funcionamiento de cada sistema ya que el consumo de memoria RAM no ha excedido los recursos asignados manteniéndose en un nivel normal.

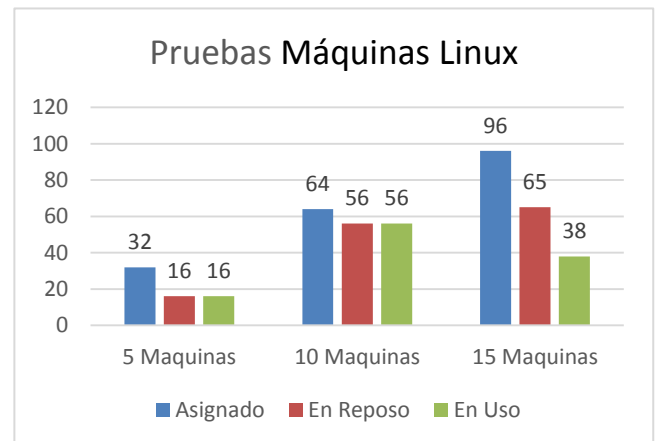


Figura 15. Consumo de Memoria RAM en Sistemas Linux
Referencia: Elaboración propia

3.3.2. Sistema Operativo Windows

En la figura 17 se puede observar el consumo de 5, 10, 15 máquinas virtuales con sistema Windows ejecutándose al mismo tiempo, en donde se observa que mientras se encuentran en reposo el consumo de CPU es mínimo mientras que cuando ya se encuentran en uso su consumo aumenta pero sin rebasar los valores asignados manteniéndose en un correcto funcionamiento.

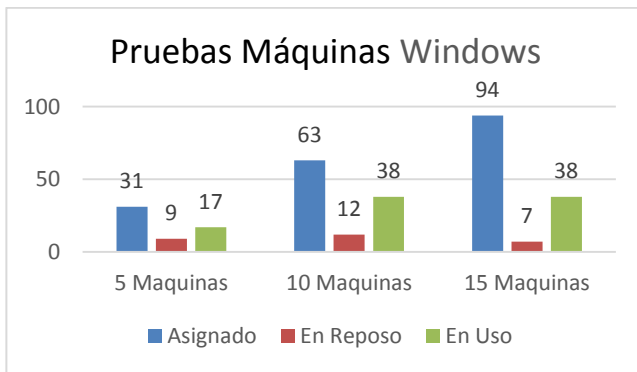
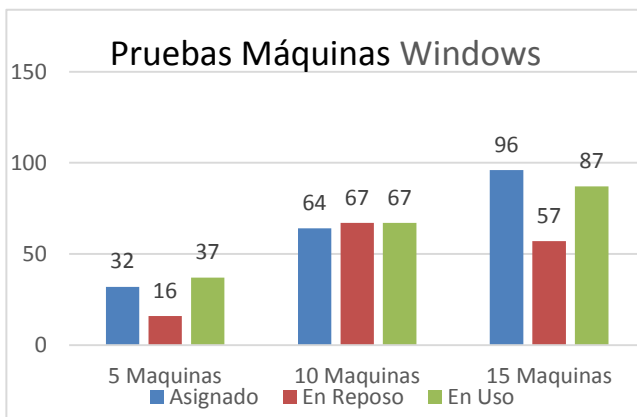


Figura 16. Consumo de CPU en sistemas Windows
Fuente: Elaboración Propia

En la figura 18 se muestra el consumo de memoria RAM que tendrá el servidor al ejecutarse 5, 10, 15 máquinas virtuales con sistema Windows en donde se puede observar que aun estando en reposo el sistema consume una gran cantidad de memoria RAM en algunos casos rebasando los recursos asignados,



4. Conclusiones

- Se diseñó las infraestructuras de cloud privado bajo la plataformas de Openstack, Eucalyptus y Open Nebula dentro de la Facultad de Ingeniería en Ciencias Aplicadas de la Universidad Técnica del Norte para brindar servicios a los estudiantes y docentes en el uso de hardware virtual destinado para diferentes actividades de distintas materias, dentro del marco de cloud computing.
- El dimensionamiento se lo realizó en base a los requerimientos de recursos de cómputo empleados por cada sistema operativo, al trabajar con software libre sus exigencias son menores y por ende el consumo de recursos es menor, a diferencia de

Windows que requiere de mayor disposición de recursos para su implementación.

- La implementación de este proyecto de cloud computing, permite brindar un nuevo servicio a la comunidad universitaria, haciendo énfasis a los estudiantes y docentes, que permita realizar actividades académicas desde cualquier lugar con acceso a internet para la satisfacción a las diferentes necesidades en función de los recursos obtenidos bajo el dimensionamiento.
- Se realizó las pruebas de funcionamiento determinando el rendimiento de las máquinas virtuales cuando se encuentran en reposo y cuando se encuentra en uso y se llegó a la conclusión que cada sistema se ejecuta de una manera correcta sin sobrecargar al servidor en donde se aloja la plataforma.
- La plataforma OpenNebula provee un sistema de monitoreo que viene integrado en la vista de administrador en la interfaz web, con lo cual el administrador podrá tener información acerca del rendimiento de los recursos de CPU y memoria asignados a las máquinas virtuales y los valores reales que se están consumiendo ayudando al administrador al dimensionamiento de la plataforma y al mejor manejo de la misma.

Agradecimientos

Se expresa un agradecimiento a los Ingenieros: Carlos Vásquez y Edgar Maya por su cooperación en el desarrollo del proyecto en que se establecía una guía en base a sus conocimientos y el tiempo en este proceso, además de ser la guía para el cumplimiento del proyecto.

El agradecimiento se extiende a la Facultad de Ingeniería en Ciencias Aplicadas perteneciente a la Universidad Técnica del Norte, entidades que permitieron el desarrollo, siendo parte de los proyectos de investigación de la Universidad que aporta con la formación de los estudiantes.

Referencias Bibliográficas

- [1] Systems.INC, E. (2010). *Cloud Computing Platform, Administrator's Guide*. Estados Unidos: Enterprise Edition 2.0. Recuperado el 07 de 11 de 2015, de Cloud Computing Platform, Administrator's Guide: EucalyptusEE2.0.AdminGuide.1.Master.pdf

- [2] Raju, M. (06 de 2010). *CSS Corp Abra Servicios Fuente*. Recuperado el 01 de 12 de 2015, de CSS Corp Abra Servicios Fuente: <http://www.csscorp.com/enterprise-it-support/open-source-services.php>.
- [3] HELION, H. (09 de 2013). *Eucalyptus Documentation*. Recuperado el 06 de 11 de 2015, de Eucalyptus Documentation: http://docs.hpcloud.com/eucalyptus/4.1.2/#install-guide/euca_components.html
- [4] Rao, M. (2015). *Cloud Computing* . Brooklyn: Editorial Asoke K. Ghosh.

Esc.”Buenos Aires”. Los estudios secundarios fueron en el Colegio Nacional “Ibarra”, donde finalizó en el año 2009, obtuvo el título de Bachiller en Ciencias Especialización Físico Matemático. Actualmente, está realizando su proceso de titulación en Ingeniería en Electrónica y Redes de comunicación, Universidad Técnica del Norte.

Joseph S. ESPINOZA B.

Nació en Ibarra provincia de Imbabura el 17 de agosto de 1992. Realizo sus estudios secundarios en la Unidad educativa Teodoro Gómez de la Torre – Ecuador en la especialidad de Físico Matemático. Cursa sus estudios universitarios en la carrera de Ingeniería en Electrónica y Redes de Comunicación en la Universidad Técnica del Norte. Actualmente es miembro voluntario de la Rama estudiantil IEEE de la universidad Técnica del Norte en donde cumple el rol de Coordinador de Membresías.

Sobre los Autores

Víctor H. AGUIRRE C.

Nació en Ibarra provincia de Imbabura el 23 de junio de 1989. Realizó sus estudios primarios en la Esc. Modelo “Presidente Velasco Ibarra”. Los estudios secundarios fueron en el Colegio “Teodoro Gómez de la Torre”, donde obtuvo el título de Bachiller en Ciencias de Especialización Físico Matemático. Actualmente está realizando el proceso de titulación en Ingeniería en Electrónica y Redes de Comunicación en la Universidad Técnica del Norte.

Yessenia F. SIGCHA P.

Nació en Quito provincia de Pichincha el 06 de febrero de 1992. Realizó sus estudios primarios en la Esc. Nueva Primavera y los finalizó en la Esc. Municipal Alfredo Albuja Galindo. Los estudios secundarios fueron en el Colegio Nacional “Ibarra”, donde finalizó en el año 2010, obtuvo el título de Bachiller en Ciencias Especialización Físico Matemático. Actualmente, está realizando su proceso de titulación en Ingeniería en Electrónica y Redes de comunicación, Universidad Técnica del Norte – Ecuador.

Erika L. SOLANO T

Nació en La Merced de Buenos Aires provincia de Imbabura el 30 de julio de 1991. Realizó sus estudios primarios en la