

MODELO Y PROTOTIPO DE SERVICIOS DE COMPUTACIÓN EN LA NUBE
PARA ESTUDIANTES Y PROFESORES DE LA ESCUELA DE INGENIERÍA DE
SISTEMAS E INFORMÁTICA DE LA UNIVERSIDAD INDUSTRIAL DE
SANTANDER

EMMANUELL DÍAZ CARREÑO

UNIVERSIDAD INDUSTRIAL DE SANTANDER
FACULTAD DE INGENIERÍAS FÍSICO-MECÁNICAS
ESCUELA DE INGENIERÍA DE SISTEMAS E INFORMÁTICA
BUCARAMANGA
2012

MODELO Y PROTOTIPO DE SERVICIOS DE COMPUTACIÓN EN LA NUBE
PARA ESTUDIANTES Y PROFESORES DE LA ESCUELA DE INGENIERÍA DE
SISTEMAS E INFORMÁTICA DE LA UNIVERSIDAD INDUSTRIAL DE
SANTANDER

EMMANUELL DÍAZ CARREÑO

Trabajo de Grado para optar al título de
Ingeniero de Sistemas

Director
MSc. Manuel Guillermo Flórez Becerra

UNIVERSIDAD INDUSTRIAL DE SANTANDER
FACULTAD DE INGENIERÍAS FÍSICO-MECÁNICAS
ESCUELA DE INGENIERÍA DE SISTEMAS E INFORMÁTICA
BUCARAMANGA
2012

*"Ser conscientes de nuestra propia ignorancia
es el primer paso hacia el saber"*
Benjamin Disraeli, 1845

DEDICATORIA

A mi madre Nohema y mi padre Antonio, todo lo que soy y he conseguido ha sido gracias y por ustedes.

A toda mi familia, por apoyarme y seguir creyendo en mí.

AGRADECIMIENTOS

El autor desea expresar sus más sinceros agradecimientos a:

Al Profesor Manuel Guillermo Flórez, por la confianza depositada desde el inicio del proyecto, y por su actitud visionaria.

A la Universidad Industrial de Santander, por permitir espacios de formación y cultura tan necesarios para nuestro país.

A la fundación Raúl D. Ocazonez por creer en el talento humano presente en la UIS, sin su apoyo no hubiese sido posible este proyecto.

Al Grupo de Arquitectura Distribuida de la Universidad Complutense de Madrid, por su aporte a la comunidad del software libre con un producto de tan excelente características como lo es OpenNebula.

A las comunidades de desarrollo de software libre, por aportar sus conocimientos para el beneficio común.

A mis compañeros de la Universidad, Alex, Elkin, John, Lizeth, Sergio y William, por acompañarme en tantos momentos y permitirme compartir con ustedes toda esta experiencia.

Al magister Jorge Armando Ortiz por sus consejos, a mis amigos de Villanueva y a todas aquellas personas que de una u otra manera pusieron su granito de arena para alcanzar esta meta, muchas gracias.

CONTENIDO

	pág
INTRODUCCIÓN	16
1. INFORMACIÓN DEL PROYECTO	18
1.1. Descripción del problema	18
1.2. Justificación	18
1.3. Viabilidad	19
1.4. Metodología	20
2. OBJETIVOS.....	21
2.1. Objetivo General.....	21
2.2. Objetivos Específicos.....	21
3. MARCO TEÓRICO	23
3.1. Virtualización	23
3.1.1 Historia de la virtualización.....	23
3.1.2 Tipos de virtualización	24
3.1.3 Ventajas y desventajas.....	26
3.2. Computación en la Nube	26
3.2.1 Definición	27
3.2.2 Características Esenciales	27
3.2.3 Modelos de Servicio	28
3.2.4 Modelos de despliegue	30
4. ANÁLISIS Y DISEÑO.....	32
4.1. Situación actual de la EISI	32
4.2. Análisis	32
4.3. Clasificación.....	34
4.4. Determinación de los servicios	37
4.5. Diseño del modelo de servicio	39
5. CONSTRUCCIÓN DEL PROTOTIPO.....	43

5.1.	Selección del hipervisor	44
5.2.	Selección herramienta IaaS	45
5.3.	OpenNebula.....	47
5.3.1	¿Qué es OpenNebula?	47
5.3.2	Características	48
5.3.3	Componentes.....	49
5.4.	Integración con el modelo.....	50
5.5.	Implementación de la herramienta.....	52
5.6.	Pruebas de funcionamiento	55
6.	PUESTA EN MARCHA DEL PROTOTIPO.....	57
6.1.	Políticas de seguridad.....	57
6.2.	Acceso de los usuarios	59
6.3.	Carga máxima soportada.....	60
6.3.1	Cantidad de máquinas instanciadas.....	60
6.3.2	Uso de CPU	61
6.3.3	Uso de Memoria RAM	63
6.3.4	Consumo de energía.....	64
6.4.	Pruebas de funcionamiento	67
6.4.1	Pruebas de instanciación	67
6.4.2	Pruebas de migración	69
6.5.	Servicios virtualizados	73
6.5.1	Biblioteca de Imágenes de servidores virtuales	74
6.6.	Balanceo de carga.....	76
6.7.	Cuotas	77
6.8.	Políticas de Uso.....	78
6.9.	Prestación de servicios en la nube	80
6.9.1	Proyecto SIPCE	80
6.9.2	Curso de programación en la web.....	83
6.10.	Documentación administrativa.....	87
7.	CONCLUSIONES	89

8. RECOMENDACIONES.....	91
BIBLIOGRAFÍA.....	93
ANEXOS.....	98

LISTA DE FIGURAS

	pág
Figura 1: Tipos de Hipervisor	25
Figura 2. Modelos de servicio	29
Figura 3: Modelos de despliegue de servicios de computación en la nube	31
Figura 4: Tipos de Proyecto de grado en la EISI 2010-2011	34
Figura 5: Producto final de proyecto de grado en la EISI 2010-2011	36
Figura 6: Pirámide invertida de visibilidad de los modelos de servicio.....	38
Figura 7: Modelo de prestación de servicios para la EISI	40
Figura 8: Logotipo de OpenNebula	48
Figura 9: Componentes de OpenNebula	50
Figura 10: Modelo del prototipo de IaaS para la EISI	52
Figura 11: Interfaz de administración de OpenNebula	53
Figura 12: Interfaz de usuario del portal de Auto-servicio	55
Figura 13: Uso de la CPU durante las pruebas.	62
Figura 14: Número de máquinas virtuales instanciadas durante las pruebas.	63
Figura 15: Uso de memoria RAM durante las pruebas.	64
Figura 16: Potencia consumida por el sistema durante las pruebas.....	64
Figura 17: Costo energético aproximado.	67
Figura 18: Tráfico de red de la instancia de Programación Web.....	85
Figura 19: Uso de CPU por la instancia de Programación Web	86
Figura 20: Uso de la memoria RAM de la instancia de Programación Web	87

LISTA DE TABLAS

	pág
Tabla 1: Escenarios de IaaS para la EISI.....	41
Tabla 2: Hipervisores seleccionados para análisis.....	44
Tabla 3: Herramientas candidatas para administración IaaS.....	46
Tabla 4: Plantillas de auto servicio.....	61
Tabla 5: Tiempo de duración del proceso de instanciación.....	68
Tabla 6: Tiempo de duración del proceso de migración.....	69
Tabla 7: Tiempos de recuperación de la conexión local en una migración.....	70
Tabla 8: Tiempos de recuperación de la disponibilidad del servicio.....	71
Tabla 9: Uso del servidor web de SIPCE durante Marzo-Abril de 2012.....	81
Tabla 10: Uso del servidor de Programación en la web Marzo-Abril de 2012.....	84
Tabla 11: Tráfico de red de la instancia de Programación Web.....	85

LISTA DE ANEXOS

	pág
Anexo A: Términos y Condiciones de Uso.....	98
Anexo B: GID-CONUSS	102

RESUMEN

TÍTULO: MODELO Y PROTOTIPO DE SERVICIOS DE COMPUTACIÓN EN LA NUBE PARA ESTUDIANTES Y PROFESORES DE LA ESCUELA DE INGENIERÍA DE SISTEMAS E INFORMÁTICA DE LA UNIVERSIDAD INDUSTRIAL DE SANTANDER.*

AUTOR: DÍAZ CARREÑO, Emmanuell**

PALABRAS CLAVE: Virtualización, Computación en la Nube, Software Libre, IaaS, Infraestructura como Servicio.

DESCRIPCIÓN: La computación en la nube representa un cambio del paradigma de la computación centrada en la prestación de un único servicio hacia un modelo de prestación de múltiples servicios sobre hardware compartido, así como también un cambio representado en el desplazamiento de las labores de cómputo a dispositivos diferentes a los equipos finales de un usuario, dicho modelo se apoya en el uso de la virtualización para la utilización eficiente de los recursos de hardware con los cuales cuenta un centro de datos.

La infraestructura presente en los entornos académicos y las limitaciones en la cantidad de los recursos con los que cuentan, hace que sea necesario buscar formas que permitan utilizar la capacidad no utilizada del hardware para el beneficio de los miembros de estos entornos, es allí, donde la computación en la nube permite un eficiente aprovechamiento de esta capacidad inactiva.

En el presente trabajo de investigación se presenta la construcción y posterior implementación de un modelo de prestación de servicios que busca apoyar los proyectos desarrollados por los miembros y grupos de investigación de la Escuela de Ingeniería de Sistemas e Informática de la Universidad Industrial de Santander, utilizando el modelo de computación en la nube conocido como IaaS o Infrastructure as a Service.

* Trabajo de Grado en la Modalidad de Investigación.

** Facultad de Ingenierías Físico Mecánicas. Escuela de Ingeniería de Sistemas e Informática. Director MSc. Manuel Guillermo Flórez Becerra

ABSTRACT

TITLE: MODEL AND PROTOTYPE OF A CLOUD COMPUTING SERVICE FOR STUDENTS AND TEACHERS FROM THE SYSTEMS ENGINEERING AND COMPUTER SCIENCE SCHOOL OF THE INDUSTRIAL UNIVERSITY OF SANTANDER*

AUTHOR: DÍAZ CARREÑO, Emmanuell **

KEYWORDS: Virtualization, Cloud Computing, Free Software, IaaS, Infrastructure as a Service.

DESCRIPTION: Cloud computing represents a paradigm shift in computing from a model focused on providing single services to provision multiple services on shared hardware, as well as taking away the computing load from end user devices, this model is based on the use of virtualization for an efficient use of hardware resources that an data center possesses.

The infrastructure present in the academic environments and the limitations on the amount of resources which they have, made it necessary to find ways for using the idle capacity of the hardware for the benefit of members of these environments, it is there, where cloud computing allows efficient utilization of this idle capacity.

This research work presents the construction and implementation of a cloud computing infrastructure as a service model, which seeks to support projects by members and research groups from the Systems Engineering and Computer Science School at the Universidad Industrial de Santander, using the model of cloud computing or IaaS known as Infrastructure as a Service.

* Undergraduate final project, research modality.

** Physico-Mechanical Engineering Faculty. Systems Engineering and Computer Science School. Director MSc. Manuel Guillermo Flórez Becerra.

INTRODUCCIÓN

La computación en la nube se ha convertido en un concepto muy importante en los últimos años [1], este, representa un cambio en la forma en la cual se utilizan los centros de datos y un cambio de paradigma en desarrollo acerca de la computación y la manera en la cual se usa y se accede a ésta [2].

Los modelos de prestación de servicios apoyados en la computación en la nube han generado una industria que mueve miles de millones de dólares al año y a pesar de ser un paradigma con un par de años en el mercado, prácticamente considerado en su infancia, se ha posicionado con sus prestaciones, alta usabilidad y la posibilidad de usar servicios bajo demanda, como una de las tecnologías más prometedoras a corto plazo[3].

El campo de la computación en la nube, ha sido investigado por la academia desde sus inicios, pero debido a la etapa en la cual se encuentra, en la literatura existente es fácil hallar información contradictoria acerca de los conceptos fundamentales que lo rodean [4]; a pesar de eso, hay cada vez un mayor consenso en dichos conceptos gracias al gran potencial futuro de esta tecnología y a los esfuerzos realizados por desarrollar estándares por parte de múltiples entidades.

Las soluciones de código abierto presentan una oportunidad para que organizaciones interesadas en experimentar con la computación en la nube puedan hacerlo, sin necesidad de grandes inversiones iniciales de capital, ni infraestructura, ya que pueden hacer uso de sus equipos existentes y determinar cuáles son sus necesidades respecto a este nuevo paradigma.

Este trabajo presenta inicialmente un panorama de las razones que motivaron su realización y los objetivos perseguidos por el mismo; a partir del capítulo 3 expone una introducción al paradigma de la computación en la nube y una definición de

ésta sobre la cual apoyar la investigación realizada. En el capítulo 4, se realiza un análisis de la situación actual de la EISI respecto a su producción académica, para determinar el modelo de servicio a desarrollar y construir. En el capítulo 5 se describe el proceso de construcción del prototipo de prestación de servicios diseñado, haciendo una evaluación para la selección del hipervisor y de las herramientas de software libre candidatas y su posterior implementación.

En el capítulo 6 se describe la puesta en marcha del prototipo, la implementación de medidas de seguridad y se realiza una serie de pruebas para comprobar la estabilidad, alcance, limitaciones y eficiencia del prototipo. Finalmente, se exponen las políticas de uso del servicio y las experiencias obtenidas con la prestación del servicio de computación en la nube desarrollado en el presente proyecto, implementado sobre dos escenarios de producción.

1. INFORMACIÓN DEL PROYECTO

1.1. Descripción del problema

Actualmente la Escuela de Ingeniería de Sistemas e Informática de la Universidad Industrial de Santander no cuenta con un servicio de Computación en la Nube en ninguna de sus diferentes versiones sobre las cual se pueda realizar desarrollo de aplicaciones o la prestación de servicios.

La infraestructura existente posee ineficiencias debido a la baja tasa de utilización de los recursos como son cómputo y almacenamiento y la dificultad de los miembros de la comunidad para hacer uso de dichos recursos.

La adquisición reciente de nuevo hardware por parte de la Escuela de Ingeniería de Sistemas e Informática requiere el desarrollo de una plataforma sobre la cual se puedan apoyar los diferentes grupos y miembros de la EISI para maximizar la utilización de dichos recursos y sobre la cual puedan desarrollar, implementar, probar y mejorar sus aplicaciones o proyectos antes de ser presentados o publicados ante los interesados.

El surgimiento de nuevas tecnologías y la renovación de otras, requiere mantener la EISI preparada para los nuevos paradigmas de desarrollo y prestación de servicios que se puedan realizar sobre estas en el transcurso del tiempo.

1.2. Justificación

En el desarrollo del proyecto se construirá un modelo de Computación en la Nube que permita el préstamo de algunos servicios con los cuales no cuenta la Escuela

en este momento. Se pretende generar una plataforma sobre la cual los diferentes grupos de investigación, grupos de desarrollo, profesores y asociaciones de estudiantes, puedan aprovechar las ventajas de la computación en la nube en sus diferentes proyectos, respaldados por los diferentes servicios que se configurarán y la seguridad, flexibilidad y disponibilidad a brindar.

Se busca que la Escuela de Ingeniería de Sistemas e Informática cuente con una plataforma en la cual se puedan desarrollar proyectos de diversas áreas sobre un nuevo paradigma de trabajo para estar al día con tecnologías que le permitan cumplir su misión institucional.

1.3. Viabilidad

La Escuela de Ingeniería de Sistemas e Informática cuenta actualmente con nuevos equipos disponibles para la realización de diferentes proyectos, entre los cuales se encuentra el presente trabajo de investigación, con el cual se busca hacer un uso eficiente de los recursos tecnológicos disponibles.

Respecto al software disponible para el desarrollo del presente proyecto, se cuenta con múltiples herramientas de software libre y comunidades de desarrollo en las cuales se puede apoyar la realización del modelo y prototipo a realizar; estas herramientas tienen la ventaja de no originar mayores costos económicos.

Las licencias de software de los diferentes sistemas operativos se implementarán de acuerdo a los acuerdos que maneja actualmente la universidad.

En cuanto al personal, el grupo GID-CONUSS (ver Anexo B) cuenta con un alto grado de experiencia en el manejo de servidores, el cual será fundamental para optimizar muchas de las características de las cuales se espera disponer al final del presente proyecto.

1.4. Metodología

Para la construcción del modelo y desarrollo del prototipo que se pretende realizar en el presente proyecto, se ejecutarán tres fases que permitan llevar a cabo los diferentes objetivos planteados, la división de las diferentes fases se plantean de la siguiente manera:

Una primera fase de investigación y producción del modelo, en la cual se determinarán los servicios que se hacen necesarios para la EISI; estos se conseguirán en base a la documentación disponible, los servicios actualmente disponibles y teniendo en cuenta que se busca que el modelo a construir preste la mayor cantidad de beneficios a la Escuela.

Una segunda fase de implementación del modelo en la cual se harán una serie de comparaciones entre aplicaciones candidatas para el funcionamiento del modelo, la instalación y pruebas de las mismas y la generación u obtención de las diferentes imágenes de servicios virtualizados que se hagan necesarias para satisfacer los requerimientos establecidos en el modelo construido.

Una tercera fase en la cual se realiza la implantación del modelo y la puesta en marcha del prototipo, en dicha fase, se realizará el desarrollo de un sitio web de administración de usuarios, y recursos; la interfaz de interacción entre los usuarios y el prototipo, el establecimiento de políticas de seguridad y copias de respaldo de la información que se pretende almacenar y finalmente la producción de los documentos de confidencialidad, responsabilidad y seguridad sobre la información almacenada por lo usuarios.

2. OBJETIVOS

2.1. Objetivo General

Desarrollar e implementar un modelo de servicios de computación en la nube para la Escuela de Ingeniería de Sistemas e Informática, evaluando la funcionalidad y eficiencia de los diferentes servicios propuestos.

2.2. Objetivos Específicos

Análisis de Servicios.

- Analizar, clasificar y determinar los servicios de computación para la EISI en relación a las ventajas que ofrece “Cloud Computing”.

Construcción del Modelo.

- Construir un modelo de computación en la nube y elaborar un prototipo en el nuevo servidor de la Escuela de Sistemas que permita implementar y probar el funcionamiento de los servicios definidos en el objetivo anterior y que sean de prioridad para la EISI.

Administración del acceso de usuarios a los servicios.

- Implementar un WebSite de registro e información con sus respectivas bases de datos.

Seguridad del modelo.

- Determinar y establecer las políticas de seguridad necesarias para el modelo.

- Implementar medidas de seguridad en el servidor en el que soportará el prototipo
 - Firewall
 - Copias de seguridad
- Generar documento administrativo de la seguridad del modelo.
- Producir un documento tipo contrato para establecer los límites legales de confidencialidad, seguridad de la información de usuarios almacenada y políticas de uso del servicio.

Pruebas de funcionalidad y medidas de eficiencia.

- Simular escenarios de servicios virtuales para probar su funcionalidad.
- Calcular la carga máxima soportada de servicios virtualizados por el hardware disponible, en relación a unas pruebas de rendimiento aceptables.

Otras características del modelo.

- Generar u obtener las imágenes de los servicios virtualizados.
- Integrar software necesario para balanceo de carga y disponibilidad.
- Limitar el uso de recursos mediante el establecimiento de cuotas de espacio en disco por usuario.

3. MARCO TEÓRICO

Para entender de qué se trata el paradigma de la computación en la nube, es necesario comprender los conceptos fundamentales y en qué tipo de tecnologías se apoya, en este capítulo se presentarán dichos conceptos.

3.1. Virtualización

3.1.1 Historia de la virtualización

La virtualización fue desarrollada inicialmente en la década de los 60 como medio de fragmentar el hardware de los mainframes¹ para lograr una mejor utilización de los recursos, estas particiones permitían ejecución “multitareas”: ejecutar múltiples aplicaciones y procesos al mismo tiempo. Los mainframes eran equipos costosos en aquella época, lo que llevó a diseñar el particionamiento como método para reducir en parte el costo de la inversión.

Durante la década de los 80 la virtualización fue abandonada cuando las aplicaciones cliente-servidor, los servidores y equipos de escritorio con arquitectura x86 cambiaron el paradigma hacia la computación distribuida, durante los 90, la masificación de la computación en las empresas llevó a nuevos retos operacionales y de infraestructuras a la industria de las tecnologías de la información y comunicación; algunos de esos retos fueron:

- Baja utilización de la infraestructura
- Costos crecientes en infraestructura física
- Aumento en los costos de administración
- Insuficiente protección ante fallos y desastres
- Alto costo de mantenimiento en los equipos de escritorio.

¹<http://es.wikipedia.org/wiki/Mainframe>

Estos inconvenientes de nuevo trajeron a escena al paradigma de la virtualización como herramienta para reducir la complejidad, conseguir una mejor utilización de los recursos, disminuir los costos y permitir una administración más eficiente; a diferencia de los mainframes los equipos con infraestructuras x86 no estaban diseñadas para soportar la virtualización completa, lo cual supuso un problema de desarrollo que fue superado con la creación de técnicas de interrupción llamadas trampas² para acceder a instrucciones privilegiadas.

3.1.2 Tipos de virtualización

Si bien el concepto de virtualización es abstracto y permite su uso sobre múltiples recursos, para la computación en la nube es clave la virtualización de hardware, esta se ha conseguido por dos métodos, virtualización completa y paravirtualización.

Paravirtualización es una técnica en la cual el hipervisor presenta un API al sistema operativo invitado y este realiza las peticiones a través de él al sistema operativo huésped, esto limita la cantidad de sistemas operativos virtualizables a solo aquellos que tengan soporte desde su núcleo.

Virtualización completa es una técnica que se basa en translación binaria para *atrapar* y virtualizar la ejecución de cierto tipo de instrucciones sensibles que no pueden ser virtualizadas, de esta manera las instrucciones críticas son reemplazadas por el hipervisor y emuladas por software.

Para controlar el tipo de virtualización usada se requiere de un monitor de máquina virtual, conocido como hipervisor, el hipervisor es una de las múltiples técnicas de control de virtualización que permite ejecutar diferentes sistemas operativos en

² [http://en.wikipedia.org/wiki/TRAP_\(processor_instruction\)](http://en.wikipedia.org/wiki/TRAP_(processor_instruction))

un mismo equipo, modificados en el caso de paravirtualización y sin modificaciones en el caso de virtualización completa.

Los hipervisores se clasifican en dos tipos[5]: hipervisor tipo 1 o nativo, el cual se ejecuta directamente sobre el hardware del anfitrión para controlar y administrar el sistema operativo invitado, este tipo de hipervisor representa la clásica implementación usada en arquitectura de máquinas virtuales, ya usada por IBM desde la década de los 60 y en la actualidad por XenServer³ de Citrix, ESX/ESXi⁴ de VMWare y Hyper-V⁵ de Microsoft.

Hipervisor tipo 2 o *hosted* este se ejecuta dentro del entorno de un sistema operativo convencional, de esta manera los sistemas operativos invitados se ejecutan a nivel de software separados del sistema operativo anfitrión por el hipervisor. Ejemplos de este enfoque son Virtualbox⁶ de Oracle y el kernel modificado de Linux KVM⁷. Una comparación gráfica entre los dos tipos puede observarse en la Figura 1.

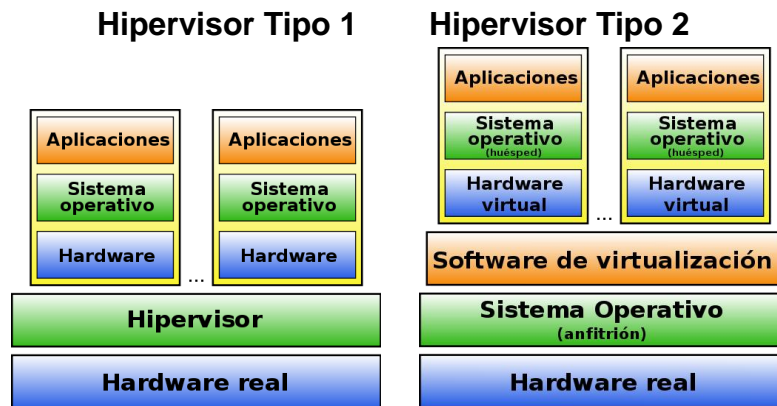


Figura 1: Tipos de Hipervisor

Fuente: sliceoflinux.com

³ www.xensource.com

⁴ www.vmware.com/products/vsphere/esxi-and-esx/index.html

⁵ www.microsoft.com/hyper-v-server/

⁶ www.virtualbox.org

⁷ www.linux-kvm.org/

3.1.3 Ventajas y desventajas

Algunas ventajas importantes de la virtualización incluyen las facilidades de administración y operación, la facilidad de generar instancias virtuales, la clonación de imágenes, la flexibilidad con la cual se pueden asignar una mayor cantidad de recursos computacionales (Memoria, Disco, Procesador e Interfaces de red) en caso de ser necesario, la reducción de procesos administrativos con la homogeneización de máquinas, el aislamiento de cada una de las máquinas virtuales que les permite no afectar a las demás en caso de fallo, la seguridad que presenta cada máquina no compromete al sistema operativo huésped y finalmente la portabilidad, ya que una máquina puede ser migrada fácilmente de un servidor a otro.

La principal desventaja, es la necesidad de equipos con características de hardware robustas, principalmente máquinas dedicadas a esta labor, esto debido a la pérdida de rendimiento a mayor cantidad de máquinas virtuales instanciadas, fundamentalmente por saturación de operaciones de entrada y salida.

3.2. Computación en la Nube

El paradigma del cloud computing ha tenido múltiples definiciones desde la aparición del concepto hasta la realización del presente trabajo [1], [2], [3], [4], [6], [7], algunas plantean enfoques históricos, tecnológicos y comerciales, aún no existe un común acuerdo; sin embargo para efectos de la presentación de este trabajo se presentará la definición, características, modelos de servicio y tipos de despliegue planteadas por el Instituto Nacional de Estándares y Tecnología de los Estados Unidos, NIST por sus siglas en Inglés en su informe final sobre la Definición de Cloud Computing.

3.2.1 Definición

“Es un modelo que permite el acceso por red a un conjunto compartido de recursos computacionales configurables (ej. Redes, servidores, almacenamiento, aplicaciones y servicios) de manera ubicua, conveniente y bajo demanda, que puede ser entregados y proporcionados rápidamente, con un mínimo de esfuerzo administrativo o de interacción del proveedor de servicios.” [6]

3.2.2 Características Esenciales

Auto servicio bajo demanda. Un usuario puede unilateralmente aprovisionarse de capacidades de cómputo, tales como uso de servidor, almacenamiento en red, a medida que lo necesite, de forma automática y sin requerir de la interacción humana del proveedor de servicio.

Amplio acceso por red. Las capacidades de cómputo están disponible en la red y son accesibles a través de mecanismos comunes que permitan el uso de plataformas heterogéneas para los usuarios (ej. teléfonos móviles, tablets, computadores portátiles y de escritorio).

Recursos comunes compartidos. Los recursos de cómputo del proveedor son comunitarios y permiten servir a múltiples usuarios usando un modelo de infraestructura compartida, con diferentes recursos físicos y virtuales asignados dinámicamente y reasignados de acuerdo a la demanda de los usuarios. Existe la sensación de independencia de la ubicación en la cual el usuario no tiene control o conocimiento acerca de la ubicación exacta de los recursos aprovisionados, pero puede existir la posibilidad de especificar la ubicación en un nivel más alto de abstracción (ej. país, estado o centro de cómputo). Ejemplos de recursos incluyen almacenamiento, procesamiento, memoria y ancho de banda.

Elasticidad rápida. Las capacidades de cómputo pueden ser aprovisionadas y liberadas rápidamente y en algunos casos de forma automática para escalar rápidamente aumentando o disminuyendo la cantidad para hacer frente a la demanda. Para el usuario, las capacidades disponibles para aprovisionamiento pueden parecer ilimitadas y puede adquirir cualquier cantidad en cualquier momento.

Servicio medible. Los sistemas de computación en la nube controlan y optimizan automáticamente el uso de recursos suministrando una capacidad de medición bajo algún nivel de abstracción apropiado al tipo de servicio (ej. almacenamiento, procesamiento, ancho de banda y usuarios concurrentes). El uso de recursos puede ser monitoreado y reportado, permitiendo un servicio transparente para el proveedor y para el consumidor del servicio utilizado.

3.2.3 Modelos de Servicio

Software as a Service (SaaS). La capacidad de cómputo proporcionada al usuario, es el uso de las aplicaciones del proveedor que se encuentran en ejecución en la infraestructura cloud. Las aplicaciones son accesibles desde varios dispositivos de usuario a través de interfaces sencillas, como un navegador web (ej. web email), o una aplicación de escritorio. El usuario no administra o controla la infraestructura subyacente sobre la cual se ejecuta, como hardware, servidores, sistemas operativos, almacenamiento o capacidades de control sobre la aplicación; con la posible excepción de preferencias de configuración de la aplicación para específicas para el usuario.

Platform as a Service (PaaS). La capacidad de cómputo proporcionada al usuario es la de desplegar aplicaciones adquiridas o creadas por el usuario, usando lenguajes, librerías, servicios y herramientas soportadas por el proveedor. El

usuario no administra o controla la infraestructura subyacente sobre la cual se ejecuta, incluyendo redes, servidores, sistemas operativos o almacenamiento, pero tiene control sobre las aplicaciones desplegadas y posiblemente sobre las preferencias de configuración del entorno de ejecución de la aplicación.

Infrastructure as a Service (IaaS). La capacidad de cómputo proporcionada al usuario es el aprovisionamiento de redes, procesamiento, almacenamiento y otros recursos computacionales fundamentales sobre los cuales el cliente puede desplegar y ejecutar aplicaciones según su criterio, lo cual incluye sistemas operativos y aplicaciones. El usuario no administra o controla la infraestructura subyacente pero tiene control sobre los sistemas operativos, almacenamiento y aplicaciones desplegadas; y posiblemente un control limitado sobre algunos componentes de red seleccionados (ej. firewall).

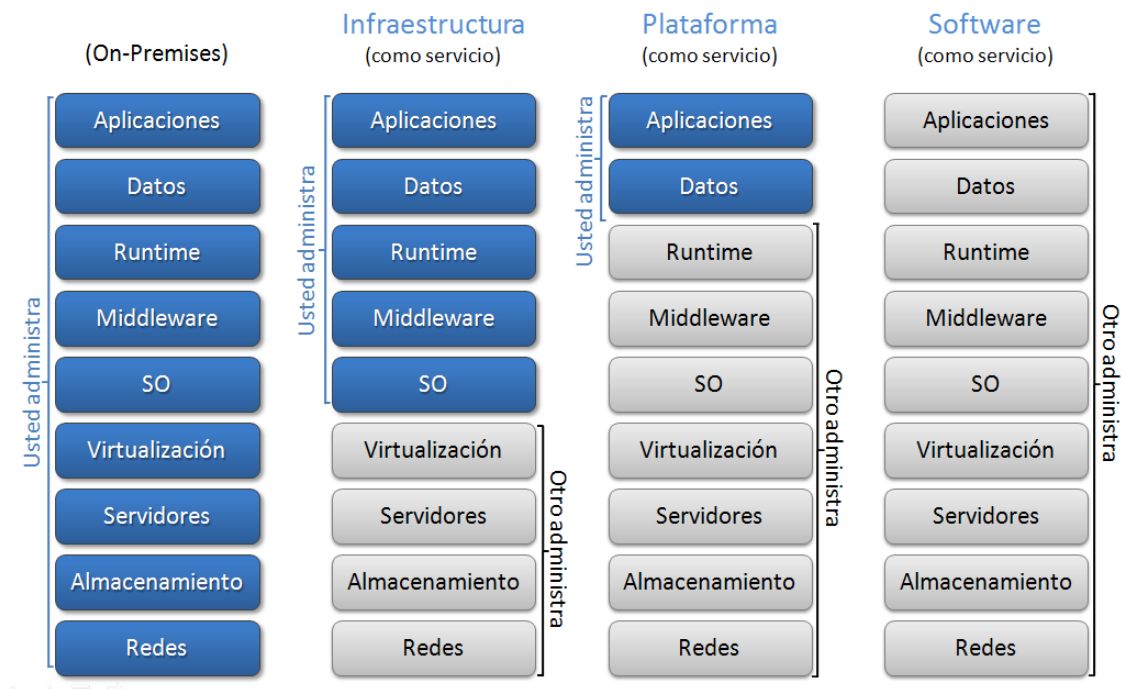


Figura 2. Modelos de servicio
Adaptada de: microsoft.com

En la Figura 2 se puede observar una comparación directa entre los diferentes tipos de modelos de servicios de computación en la nube indicando los niveles de administración permitida al cliente y los que son responsabilidad del proveedor del servicio, junto al modelo tradicional de servicio. *On-premises* quiere decir que el servicio de cómputo es alojado en las instalaciones del usuario, y que este mismo como usuario del servicio administra la totalidad de los componentes, algo que puede ser innecesario según el tipo de empresa, la administración de redes, almacenamiento, servidores, hipervisores, sistemas operativos, *middleware* (software intermediario), *runtimes* (entornos de ejecución), aplicaciones y datos pueden ser manejados por un tercero según el modelo seleccionado. La figura anterior también muestra el rol que desempeña la virtualización dentro de los diferentes modelos de servicio.

3.2.4 Modelos de despliegue

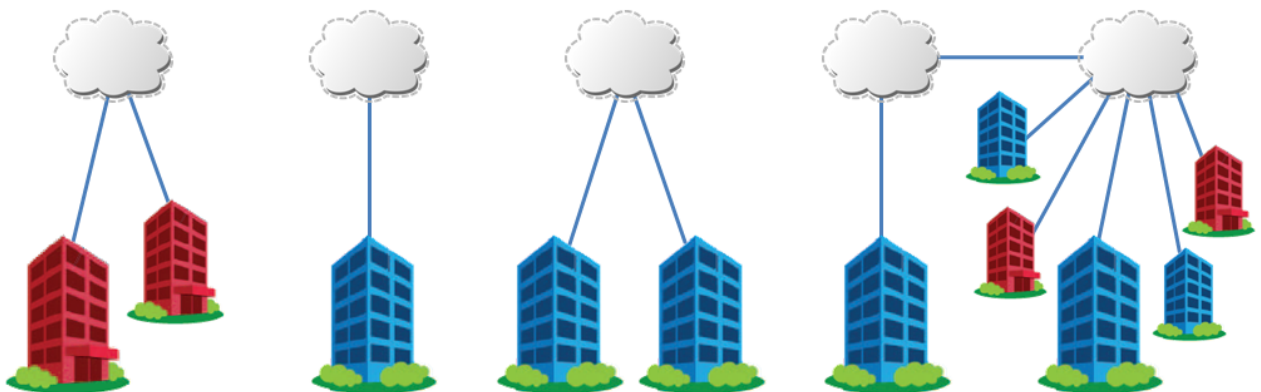
Nube privada. La infraestructura es proporcionada para uso exclusivo de una sola organización, conformada por múltiples usuarios (ej. unidades de negocio). Puede ser administrada, operada o de propiedad de la organización, un tercero o alguna combinación, Puede existir dentro de las instalaciones de la organización o fuera de ella.

Nube Comunitaria. La infraestructura es proporcionada para uso exclusivo de clientes de organizaciones que tengan intereses comunes (ej. misión, políticas, requisitos de seguridad y cumplimiento). Puede ser administrada y operada por uno o más organizaciones de la comunidad, un tercero o alguna combinación, Puede existir dentro de las instalaciones de la organización o fuera de ella.

Nube pública. La infraestructura es proporcionada para uso abierto al público en general. Puede ser administrada, operada o de propiedad por una empresa, una organización académica o del gobierno, o alguna combinación de estas, Existe dentro de las instalaciones del proveedor del servicio.

Nube híbrida. La infraestructura es una construcción de dos o más infraestructuras cloud (privada, comunitaria o pública) que siguen siendo entidades únicas, pero que se encuentran vinculadas por tecnologías estandarizadas o propietarias que permiten portabilidad de datos y aplicaciones (ej. *cloud bursting*⁸ para balanceo de carga entre nubes).

En la Figura 3 se realiza una comparación entre los diferentes modelos de despliegue, en esta, se expone el grado de vinculación que se presenta entre diferentes organizaciones al hacer uso de los diferentes modelos.



Pública: La infraestructura está disponible para el público en general o para un grupo de empresas y es propiedad de un tercero que comercializa los servicios.

Privada: La infraestructura es operada únicamente para una organización. Puede ser administrada por ella misma o por un tercero, puede o no estar dentro de las instalaciones.

Comunitaria: La infraestructura es compartida por varias organizaciones y da soporte a una comunidad que tienen asuntos en común (ej. misión, requerimientos de seguridad, políticas). Puede ser administrada por las organizaciones o por un tercer y puede o no estar dentro de las instalaciones.

Híbrida: Una construcción de dos o más nubes (públicas, privadas o comunitarias) que continúan como entidades únicas pero se encuentran vinculada por tecnologías estandarizadas o propietarias, lo que les permite portabilidad de datos o aplicaciones (ej. *cloud bursting* para balanceo de carga entre nubes).

Figura 3: Modelos de despliegue de servicios de computación en la nube
Adaptada de: thesolutionsarchitect.net

⁸ El cloud bursting está basado en una recolocación del volumen de trabajo entre los espacios privados y la nube pública <http://cloud.ticbeat.com/representa-cloud-bursting-llevarlo-cabo/>

4. ANÁLISIS Y DISEÑO

Este capítulo se encuentra estructurado de la siguiente manera, inicialmente se hace una revisión del estado de los recursos actuales de la EISI y se definen los diferentes grupos objetivos de este proyecto, luego se realiza una clasificación de los productos finales de la producción académica de los trabajos de los estudiantes de pregrado de últimos 2 años; finalmente con el apoyo de esta información se determinan los tipos de servicios de computación en la nube requeridos.

4.1. Situación actual de la EISI

La Escuela de Ingeniería de Sistemas e Informática de la Universidad Industrial de Santander cuenta en su capital humano con cerca de mil estudiantes, 17 profesores planta, 20 profesores cátedra y 7 grupos de Investigación⁹; en cuanto a recursos computacionales, cuenta con 4 salas de cómputo, 76 equipos distribuidos en grupos de investigación, sala de proyectos de grado y laboratorios¹⁰, además 2 servidores principales.

Para la realización del presente proyecto se cuenta con 2 nuevos equipos tipo servidor, donados por la fundación Raúl D. Ocazonez.

4.2. Análisis

Para realizar el análisis de la situación actual de la Escuela, se hace inicialmente una definición formal de los grupos a los cuales que se desea beneficiar con el presente proyecto:

⁹ Información disponible en <http://cormoran.uis.edu.co>

¹⁰ Cuadernillo del Proceso de Renovación de Acreditación de la Escuela de Ingeniería de Sistemas - 2011

Estudiantes de Pregrado: Todos los estudiantes que se encuentran actualmente matriculados en la Universidad Industrial de Santander y que pertenezcan a la Carrera de Ingeniería de Sistemas e Informática.

Estudiantes de Maestría: Todos los estudiantes que se encuentran actualmente matriculados en la Universidad Industrial de Santander y que pertenezcan a la Maestría en Ingeniería con Énfasis en Informática y Ciencias de la Computación.

Profesores: Todos los profesores planta o cátedra vinculados a la Escuela de Ingeniería de Sistemas e Informática.

Grupos de Investigación: Todos los estudiantes, profesores e investigadores que se encuentren vinculados a cualesquiera de los diferentes grupos de Investigación de la Universidad Industrial de Santander y que pertenezcan a la Escuela de Ingeniería de Sistemas e Informática.

Asociaciones de Estudiantes: Todos los estudiantes que se encuentren vinculados a cualesquiera de las diferentes asociaciones creadas y administradas por estudiantes de la Universidad Industrial de Santander; y que se encuentren matriculados en la Carrera de Ingeniería de Sistemas e Informática.

Para la fase inicial de este proyecto se realizó una labor de recopilación del trabajo de los estudiantes pregrado de la EISI, estos han producido en los últimos 2 años (2010-2011) 144 proyectos de grado¹¹. Se realizó un proceso de clasificación del tipo de proyecto realizado y del producto final entregado, con el fin de definir qué modelo de servicio de computación en la nube se seleccionaría inicialmente, buscando producir un beneficio inmediato sobre los proyectos de la comunidad objetivo.

¹¹ Consultado en los recursos electrónicos de la biblioteca UIS, <http://tangara.uis.edu.co>

4.3. Clasificación

Para la clasificación de los proyectos de grado se realizó una revisión de las 144 tesis producidas en el periodo 2010-2011, a partir de este proceso se crearon unas clases sencillas basadas en criterios del autor y encaminadas a la selección del servicio a prestar; estas clases fueron:

Práctica: Proyectos de grado en la modalidad de Práctica empresarial.

Investigación: Proyectos de grado cuya finalidad fuese la producción de un documento base o guía para el desarrollo de proyectos futuros, esta incluye producción literaria, desarrollo de algoritmos y manuales de administración o de implementación.

Desarrollo: Proyectos de grado que produzcan diferentes herramientas computacionales, esta incluye: aplicaciones de escritorio, aplicaciones móviles, frameworks, aplicaciones para HPC y aplicaciones web.

Los totales de cada clase se pueden observar en la Figura 4.

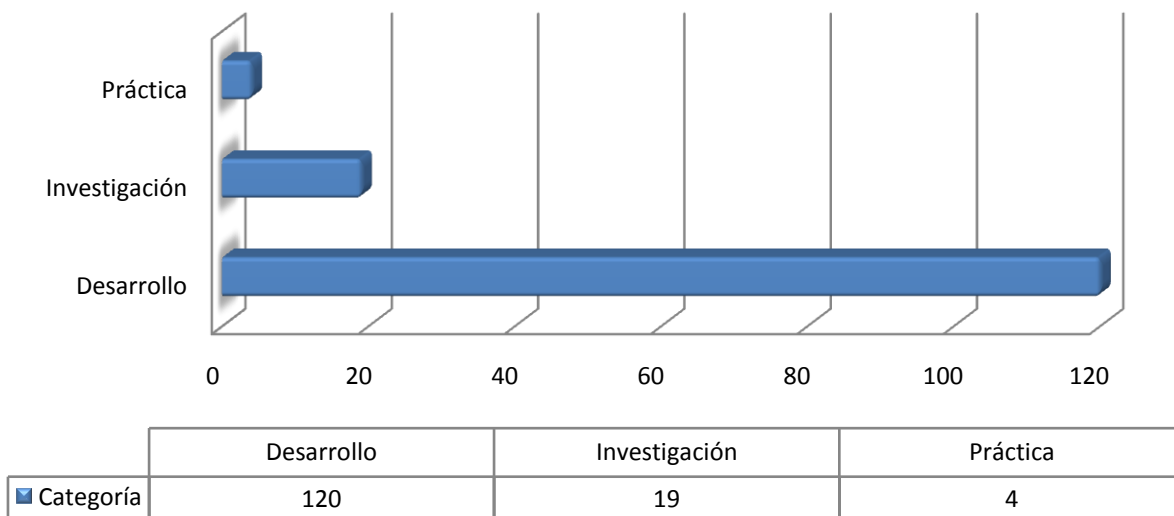


Figura 4: Tipos de Proyecto de grado en la EISI 2010-2011
Fuente: Autor

Un número elevado de proyectos, aproximadamente el 83%, pertenecen a la clase *Desarrollo*, esta tendencia permitió encontrar que este tipo de proyectos pueden ser los más beneficiados inicialmente con la realización del presente proyecto, por lo tanto se da mayor prioridad a la construcción de un modelo de computación en la nube que permita que la clase *Desarrollo* aproveche mejor sus ventajas.

Además de la clasificación de cada proyecto por tipo, se realizó otra clasificación basada en los productos finales de los proyectos de grado, para la cual se determinaron las siguientes clases:

Algoritmo: El desarrollo de un algoritmo o la optimización de uno ya existente.

Aplicación de escritorio: Una aplicación que sea ejecutada en un entorno de escritorio, sobre la cual no importa el lenguaje en que fue desarrollada.

Aplicación móvil: Una aplicación que se ejecute sobre dispositivos móviles; en este caso celulares, emuladores de dispositivos móviles o terminales portátiles.

Framework: Un Framework, sobre la cual no importa el lenguaje en que fue desarrollada ni el sistema operativo sobre el cual se ejecuta.

HPC: Una aplicación que está diseñada para hacer uso de una red de computación distribuida o de programación en paralelo.

Documentación: Un documento guía; este documento puede ser la relatoría del desarrollo de una práctica empresarial, un estudio de caso, el análisis de una herramienta software o un manual administrativo.

Aplicación web: Proyectos de grado cuyo producto final es una aplicación o módulo para aplicación que se ejecuta en un servidor y proporciona acceso para el usuario final por medio de un navegador web.

En la Figura 5 se puede observar que del total de proyectos de grado del periodo analizado un alto porcentaje correspondió a la clase *aplicación web*, aproximadamente el 53%, lo cual permitió inferir que puede existir una alta demanda de servicios de computación en la nube como apoyo para el desarrollo de este tipo de proyectos, por parte de los estudiantes de pregrado de la EISI y permitió también determinar cuáles tipos de servicios tendrían prioridad durante la construcción del modelo de servicios.

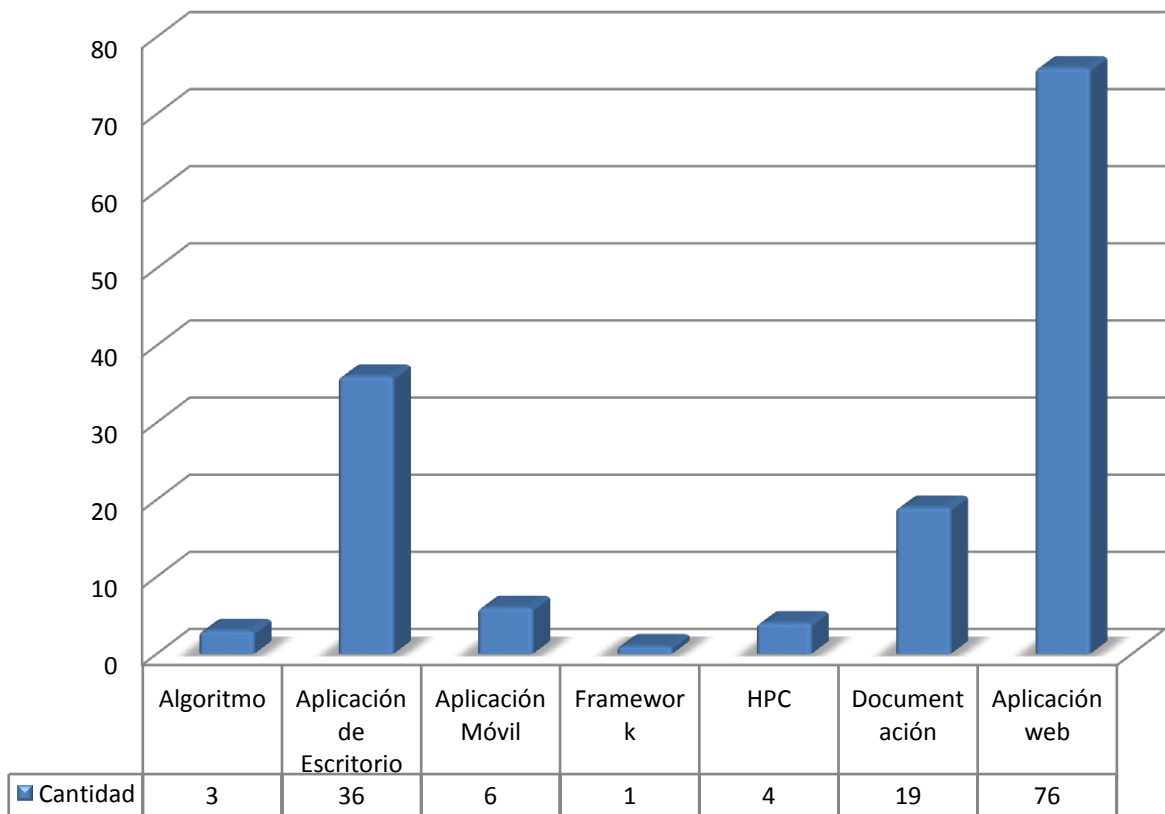


Figura 5: Producto final de proyecto de grado en la EISI 2010-2011

Fuente: Autor.

4.4. Determinación de los servicios

El desarrollo de aplicaciones web representa el más alto porcentaje de proyectos de grado de la EISI según el análisis mostrado anteriormente, esto significó que dentro del prototipo esté sería un servicio prioritario y determinaría en gran parte que tipo de modelo de servicios en la nube se seleccionaría.

Para establecer un marco de desarrollo, se analizó la prestación de un servicio que facilitara el desarrollo de aplicaciones web en un entorno académico, a partir de los diferentes modelos de servicio de la computación en la nube; para el desarrollo de Software como Servicio, SaaS, es necesario que se tenga ya una plataforma de desarrollo de aplicaciones en la nube, es decir desarrollar SaaS implica necesariamente la construcción de aplicaciones o Plataformas como Servicio, PaaS, estas a su vez deben encontrarse apoyadas en el desarrollo de una infraestructura que permita el montaje y administración de dichas plataformas, en este caso, la necesidad de un servicio de Infraestructura como servicio IaaS se hace necesario.

En la Figura 6 se puede observar cual es el apoyo que se puede obtener de parte de cada uno de los modelos de servicio, definiendo cual sería el rol de cada uno de los participantes en el modelo. La figura de pirámide invertida representa el grado de interacción y visibilidad del participante y los diferentes tipos de usuario que pueden verse beneficiados de cada uno de los modelos, teniendo en cuenta las prioridades y limitaciones mencionadas anteriormente.

Para la selección del modelo se hizo énfasis en la construcción de un modelo que permita apoyar a los diferentes posibles proyectos desde un servicio que permita una rápida asignación de recursos y facilidades de administración; con la determinación de estos parámetros se vio la necesidad de seleccionar un modelo de ser-

vicio de IaaS como la base para desarrollar el prototipo de servicio que necesita la EISI inicialmente.

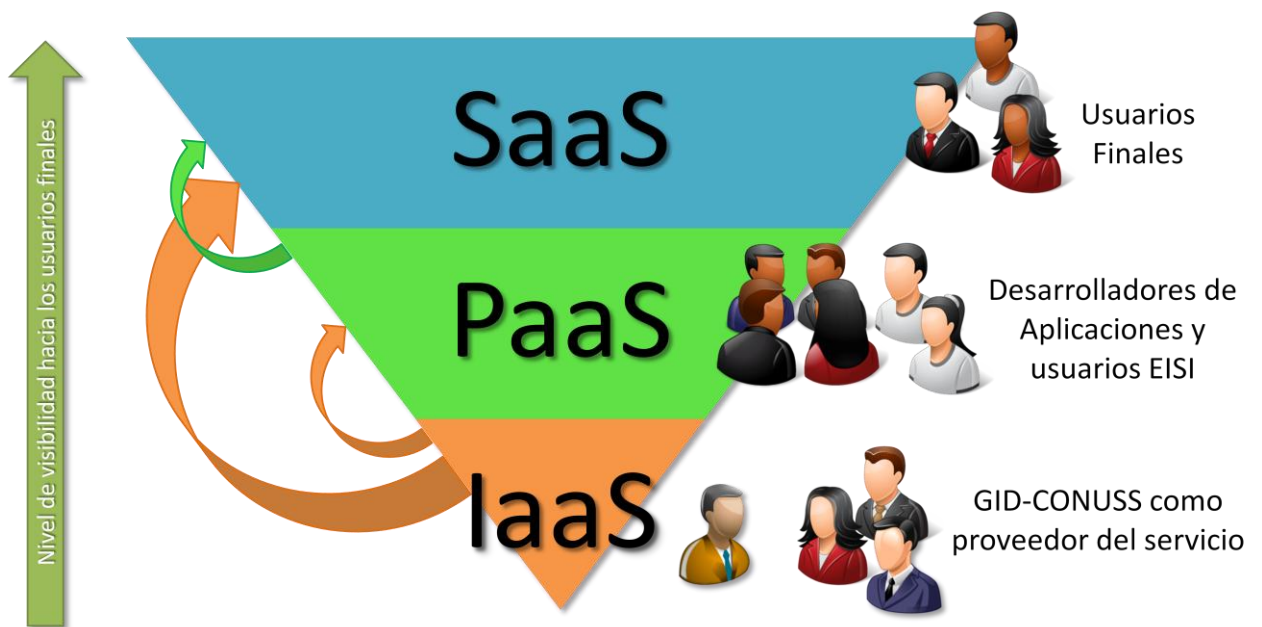


Figura 6: Pirámide invertida de visibilidad de los modelos de servicio

Adaptada de: <http://www.saasblogs.com>

Los servicios que se pueden prestar para la Escuela bajo el modelo de IaaS están limitados a ciertos parámetros definidos por el autor y que corresponden a la organización y estado actual de la Escuela, entre estas limitaciones se encuentra asegurar que las herramientas de desarrollo que se ofrezcan posean un alto grado de seguridad, esto con el fin de preservar la confidencialidad de los proyectos; otra limitación es determinar de qué manera se permite un acceso medurado y responsable a los recursos asignados a cada uno de los beneficiarios de los servicios y cómo permitir que estos puedan disponer a la medida de sus necesidades a las facilidades de migración o aprovisionamiento de recursos sin afectar el trabajo de otros equipos de trabajo, estas limitaciones son aún más marcadas cuando se trata de trabajo de grupos de investigación, en los cuales se pueden presentar necesidades de servicios de mayor capacidad basados en sus diferentes proyectos.

4.5. Diseño del modelo de servicio

El modelo de servicio IaaS, puede permitir ofrecer a los usuarios de la EISI capacidades de cómputo, redes, almacenamiento y cómputo; la distribución de responsabilidades permite que el usuario determine que aplicaciones necesita sobre su aprovisionamiento y a su vez le asigna la responsabilidad sobre la administración de su instancia¹² virtual, por su parte el prestador del servicio es responsable por la disponibilidad de la instancia y del acceso al servicio, la responsabilidad compartida se encuentra en el uso de la red, el cual se comparte entre todos los usuarios del servicio y se limita al ancho de banda asignado para cada instancia.

Para lograr el objetivo de la construcción del prototipo, se determinó la necesidad de incluir en el modelo las siguientes características:

- Un sistema de administración del sistema.
- Un sistema de asignación de recursos.
- Un sistema para el acceso de los usuarios a la administración de sus instancias.
- Un sistema para supervisión de uso de recursos.
- Un acuerdo de uso y limitaciones de prestación del servicio con el usuario final.

La Figura 7 representa la funcionalidad y la forma en la cual los usuarios pueden acceder al servicio; explicando, los usuarios pueden acceder al servicio que han solicitado por medio de cualquier dispositivo con conexión a Internet, estas conexiones se efectúan sobre el servidor principal y este se encarga de redirigir estas peticiones al respectivo servidor virtual sobre el cual se encuentra alojado el

¹² Al igual que en la computación orientada a objetos, una instancia virtual es la ocurrencia de un objeto de alguna clase, en este caso un equipo de cómputo virtual.

servicio, este acceso está limitado al cumplimiento de una serie de acuerdos entre el usuario y el proveedor del servicio.

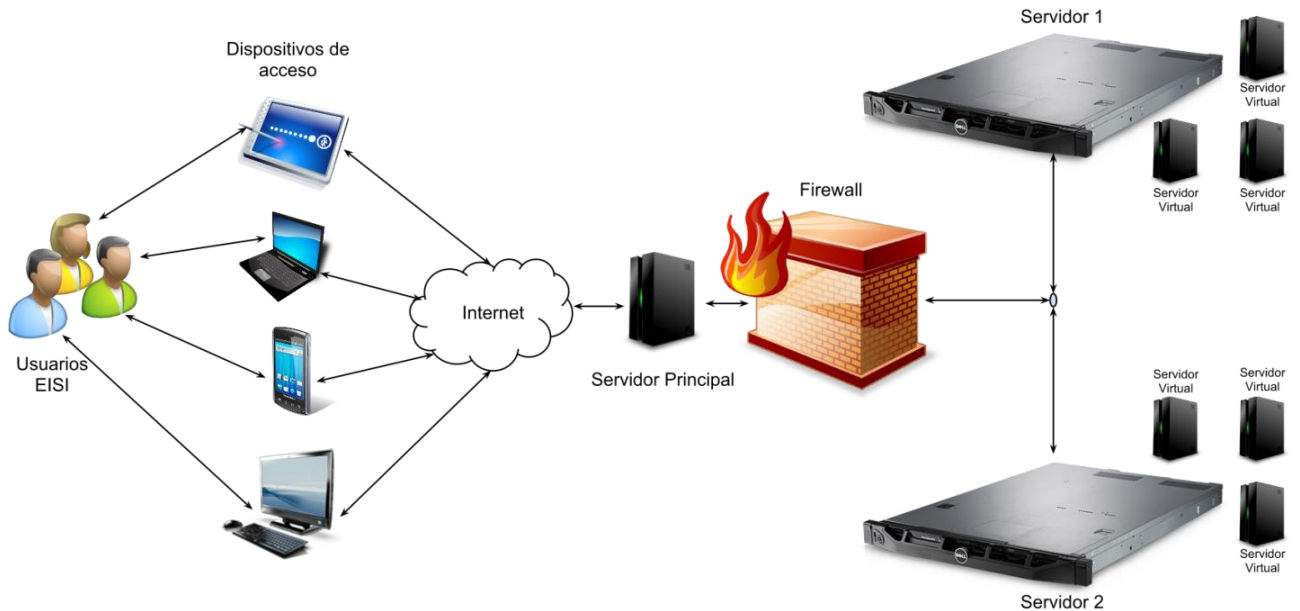


Figura 7: Modelo de prestación de servicios para la EISI
Fuente: Autor.

El servicio debe tener sistemas de seguridad y monitoreo que permitan a los administradores saber en cualquier momento el nivel de disponibilidad, seguridad y uso del servicio.

4.6. Escenarios de prestación de servicio

Con el análisis y diseño anteriormente propuesto, se realizó un proceso construcción de escenarios, que podrían ser prestados una vez estuviese implementada la plataforma. Estos se presentan en la Tabla 1.

Tabla 1: Escenarios de IaaS para la EISI

Beneficiario	Escenario	Descripción
Grupos de investigación	Repositorios de aplicaciones producidas	Permitir que el desarrollo de aplicaciones por parte de los grupos de investigación pueda permanecer disponible como referencia a futuros proyectos, o versiones finales estables que pueden pasar a producción.
	Servidores para desarrollo o producción	Facilitar la adquisición de recursos de cómputo virtuales para poner en marcha proyectos en fase de producción o para continuar el desarrollo de las mismas.
	Repositorios de código fuente	Suministrar espacios que permitan la administración del código producido por los diferentes proyectos de investigación
Proyectos de grado	Servicio de hosting para proyectos web	Facilitar el inicio de proyectos de desarrollo web, con máquinas listas para su uso.
	Control/Integración de versiones	Suministrar espacios para que proyectos que cuenten con más de un desarrollador puedan realizar fácilmente integraciones de código y realizar un control adecuado de las versiones durante el desarrollo.
	Experimentar frameworks	Permitir la fácil experimentación de nuevos frameworks por medio de máquinas instanciables que no signifiquen demoras para el inicio de los proyectos.
	Control de incidencias	Simplificar el proceso de revisión/corrección de incidencias en proyectos en desarrollo, por medio de aplicaciones para manejo de éstas.
Proyectos de curso	Servicio de hosting para proyectos web	Simplificar el proceso de obtención de recursos por parte de proyectos definidos para realización durante un periodo académico
	Control de versiones	Suministrar espacios para que proyectos que cuenten con más de un desarrollador puedan realizar fácilmente integraciones de código y realizar un control adecuado de las versiones durante el desarrollo.
	Repositorio de trabajos de curso	Permitir que los docentes cuenten con repositorios de trabajos de cursos anteriores como medio para motivar a los estudiantes y a su vez demostrar los alcances de sus compañeros.
	Motores bases de datos	Proveer a los estudiantes de asignaturas que soliciten instancias de motores de bases de datos una herramienta para que puedan desarrollar pruebas sin necesidad de realizar instalaciones locales.
Asociaciones de Estudiantes	Servicio de Hosting para portal de servicios	Facilitar el alojamiento del portal de servicios de las asociaciones de estudiantes y a su vez permitir que estos espacios estén disponibles por más de un periodo académico.
	Alojamiento de bases de datos	Suministrar una herramienta para que las aplicaciones desarrolladas para las asociaciones de estudiantes puedan prestar sus servicios de una manera correcta.

Beneficiario	Escenario	Descripción
Estudiantes	Experimentar frameworks para desarrollo web	Permitir a estudiantes hacer pruebas usando la capacidad de cómputo libre para realizar diferentes actividades con servidores de prueba, como apoyo a su formación profesional.
	Pruebas de manejo/administración de servidores	Facilitar un espacio de trabajo sobre un entorno administrativo para que el estudiante desarrolle competencias de control, mantenimiento y configuración de servidores.

Fuente: Autor

Con el hardware que se cuenta y avanzando hacia la construcción de un prototipo que permita inicialmente cumplir con los objetivos planteados, pero dejando abiertas las posibilidades para ampliar los alcances de la plataforma a futuro, se continuó con las bases definidas para la construcción del modelo de computación en la nube para la EISI.

5. CONSTRUCCIÓN DEL PROTOTIPO

Durante el proceso de construcción, se realizó un análisis de las herramientas necesarias para implementar un servicio de computación en la nube que cumpliera con los parámetros determinados en la fase de análisis y que además de esto cumpliera con otros requisitos que se plantearon desde el inicio del proyecto.

- Todas las herramientas usadas deben ser gratuitas.
- No deben existir incompatibilidad de licencias o limitaciones de uso.
- Deben funcionar sobre el sistema operativo Debian.

La gratuidad de las herramientas se requiere al ser este un proyecto netamente académico, el cual se busca que pueda ser replicado sin necesidad de incurrir en costos iniciales, principalmente por licencias de sistemas operativo o de aplicaciones, así mismo las aplicaciones que hagan parte del prototipo no deben estar amparadas en licencias que no permitan el uso por más de un usuario o que estén limitadas a condiciones de hardware o software en el equipo, y finalmente, este proyecto se desarrolló paralelamente a un proyecto de administración de servidores del grupo de investigación CONUSS¹³, el cual definió el sistema operativo de los nodos, este limitó la compatibilidad de las herramientas seleccionadas a éste y a su vez definió una base segura sobre la cual construir el prototipo.

El prototipo se compone de tres partes esenciales, un servidor principal o frontend que permita centralizar los servicios prestados y que actúe como supervisor entre los servicios ofrecidos, una herramienta de administración de la plataforma que permita orquestar los servicios prestados y un hipervisor que permita administrar todo el ciclo de vida de las instancias virtuales.

¹³ Grupo de Investigación y Desarrollo en Computación en la Nube, Seguridad, Servidores y Servicios.

Con estos requisitos presentes en la selección de las herramientas, se realizó un proceso de investigación y comparación de las aplicaciones candidatas.

5.1. Selección del hipervisor

En el proceso inicial fue necesario seleccionar un hipervisor que cumpliera con los requisitos, se seleccionaron para análisis los siguientes hipervisores: Xen, KVM, Virtualbox, Hyper-V y VMware. Estos fueron seleccionados al ser los que poseen mayor penetración en el mercado¹⁴.

Del análisis que se realizó, el cual se encuentra plasmado en la Tabla 2, se descartaron inmediatamente Hyper-V y VMware, ya que no son compatibles con el sistema operativo de los servidores disponibles.

Tabla 2: Hipervisores seleccionados para análisis

Hipervisor	Versión	Licencia	Limitaciones de uso				Compatible con Debian
			RAM	CPU	VMs	Arq	
Xen	6	GPL	2 TB	64	64	Host	Si
KVM	88	GPL v2	32 TB	256	Ilimitado	Host	Si
Virtualbox	4,1	GPL v2	Ilimitado	Ilimitado	Ilimitado	x86-64, x86	Si
Hyper-V	2008 R2	Propietario	1 TB	64	384	x86-64, x86	No, MS Windows
VMware	vSphere 5	Propietario	2 TB	160	512	x86-64, x86	No, VMware_ESXi

Fuente: Autor.

Al comparar los 3 hipervisores restantes se determinó que cualquiera de los 3 podía prestar el servicio sin inconvenientes, sin embargo, debido al análisis de los servicios que se espera prestar, se seleccionó Virtualbox dando mayor prioridad a su facilidad de manejo y de generación de servicios virtualizados de parte de los usuarios objetivo del proyecto; con esto se busca que la curva de adopción inicial del servicio sea menos acentuada y permita a futuro proveer un uso mayor de la plataforma.

¹⁴ http://mediasrc.zenoss.com/documents/cloud_computing_infographic.pdf

Virtualbox permite que un usuario sin conocimientos técnicos elevados transforme su máquina virtual en un servicio virtualizado, la simplicidad de dicha operación permite que los usuarios puedan realizar pruebas técnicas locales y luego de estar conformes con sus resultados puedan alojar estas instancias en el servicio de computación en la nube para continuar su trabajo desde ahí.

Respecto a la licencia de uso de Virtualbox, siendo tipo GPLv2 en su versión OSE y en el caso del presente proyecto se usa la versión directa de los binarios, que posee licencia PUEL¹⁵ la cual permite su uso en entornos académicos para múltiples usuarios y diversos usos sin ningún costo.

Este hipervisor no está limitado de ninguna manera a la cantidad de memoria RAM, el número de CPUs o núcleos que tenga el sistema y tampoco limita el número de instancias virtuales que pueden estar activas en el sistema, como si ocurre en algunos casos con los hipervisores Xen o KVM. La compatibilidad con Debian es completa y permite también emular arquitecturas x86 y x86-64 sin inconvenientes.

5.2. Selección herramienta IaaS

Para el proceso de selección de una herramienta de administración se estableció un requisito extra debido a la selección del hipervisor, la herramienta debe ser compatible con Virtualbox; en la Tabla 3 se pueden observar las herramientas candidatas para administración de IaaS.

¹⁵ Personal Use and Evaluation Licence https://www.virtualbox.org/wiki/VirtualBox_PUEL

Tabla 3: Herramientas candidatas para administración IaaS

Herramienta	Versión	Licencia	Limitaciones de uso					Compa- tible con Debian	Compa- tible con Virtualbox
			Migración/ Suspensión	Nubes Híbridadas	Hipervisores	AWS	WebUI		
OpenStack	Cactus	Apache	Si	No	KVM, Xen, Hyper-V, QEMU	Si	No	Si	No
Eucalyptus	2.0.2	GPL v3	No	No	VMware, Xen, KVM	Si	No	Si	Si
OpenNebula	2,2	Apache	Si	Si	Xen, KVM, VMware	Si	Si	Si	Si, con plugin
Nimbus	2,8	Apache	Si	No	Xen, KVM	No	No	No	No

Fuente: Autor.

Las herramientas OpenStack y Nimbus no cumplieron con el requisito de compatibilidad con Virtualbox por lo tanto fueron descartadas.

Se realizó el proceso de instalación de las herramientas aún no descartadas, se analizó la complejidad para la instalación de cada una, esto pensando en las labores administrativas a futuro al realizar migraciones del servidor o actualizaciones de la plataforma, se determinó también el grado de dificultad para la administración de los recursos, o para la supervisión de estos; este tipo de análisis representa una verdadera dificultad para realizar la selección [8], ya que se pasa de una selección basada en características comparables o medibles al plano de las prestaciones de la herramienta, tanto las que ya están desarrolladas como las que aparecen en los planes de desarrollo; este proceso eleva la complejidad de la selección y hace que se determine una característica a la cual darle mayor prioridad, en este caso, dicha característica fue la posibilidad de migrar las instancias virtuales, esto buscando aprovechar de mejor manera los recursos con que se cuenta, con esta característica prioritaria y debido a que se busca que la plataforma construida cuente con una alta capacidad de expansión a futuro se seleccionó OpenNebula¹⁶.

¹⁶ <http://opennebula.org>

Otras de las características que motivaron la selección de OpenNebula, fueron la modularidad de sus componentes, esta funcionalidad permite que a futuro se puedan usar otros hipervisores sin inconvenientes y facilita el desarrollo de scripts¹⁷ que permitan agregar nuevas funcionalidades a la herramienta sin necesidad de alterar el código fuente y sin quedar limitados al desarrollo que realicen los autores de la herramienta, además de las prestaciones para la construcción de una nube privada [9] y su compatibilidad con OCCI¹⁸. OCCI¹⁹, *Open Cloud Computing Interface*, es un conjunto de especificaciones desarrolladas por comunidades abiertas y reguladas a través del Open Grid Forum, que define un estándar para que los proveedores de servicios puedan prestar sus ofertas de cómputo, datos y redes.

Otra ventaja con la que cuenta OpenNebula es la reciente inclusión de una interfaz gráfica de administración y de usuario final, características que se pueden considerar clave ya que toda la administración del servidor debe realizarse de manera remota y además permite que el proceso de formación a nuevos administradores requiera menos tiempo; la interfaz de usuario final permite que los mismo usuarios administren, instancien y liberen recursos de acuerdo a sus necesidades y acuerdos de servicio.

5.3. OpenNebula

5.3.1 ¿Qué es OpenNebula?

OpenNebula es un gestor Open Source de infraestructuras virtuales distribuido bajo licencia Apache. Esta tecnología permite virtualizar las infraestructuras de almacenamiento y de recursos de cómputo de cualquier centro de datos existente y adaptarla dinámicamente en función de las necesidades de servicio. Combina

¹⁷ <http://opennebula.org/documentation:rel3.4:hooks>

¹⁸ <http://opennebula.org/documentation:archives:rel2.2:occicg>

¹⁹ <http://occi-wg.org/>

tecnologías de virtualización, almacenamiento y redes. Permite la construcción de todo tipo de cloud: públicos, híbridos y privados [10].

Es un proyecto desarrollado por el Grupo de Arquitectura Distribuida de la Universidad Complutense de Madrid, iniciado en el año 2005 como un proyecto de investigación en computación distribuida; tuvo su primer lanzamiento en software en el año 2008, a partir del cual ha evolucionado a una herramienta de administración de máquinas virtuales en infraestructuras distribuidas de gran escala. Muchas de sus características innovadoras le han permitido ser seleccionadas por un gran número de compañías para el montaje de sus soluciones de computación en la nube²⁰, entre estas se encuentran RESERVOIR, StratusLab, BonFIRE, 4CaaS, Telefónica de Alemania, RIM, Akamai, IBM, SAP, RedIris, CESGA, CESCA, SARA, Tokyo Institute of Technology, CERB, Fermilab, Harvard University SEAS y Universidade Federal Do Ceará entre otros²¹.



Figura 8: Logotipo de OpenNebula

Fuente: <http://opennebula.org>

5.3.2 Características

Entre las características principales de OpenNebula se encuentran²² [xx]:

Transparencia:

- Arquitectura abierta.

²⁰ <http://opennebula.org/about:about>

²¹ <http://opennebula.org/community:users>

²² <http://opennebula.org/about:technology>

- Interfaces abiertas.
- Código abierto.

Adaptabilidad

- Arquitectura modular.
- Administración de múltiples combinaciones de hardware y software.

Interoperabilidad

- Portabilidad para evitar dependencia de proveedor.

Estabilidad

- Esta desarrollada para ser usada en producción en entornos de clase empresarial.

Estandarización

- El uso de estándares como OCCI le permite, interactuar con otros proveedores de manera transparente para el usuario final.

5.3.3 Componentes

OpenNebula se conforma de cinco componentes básicos:

Front-end: Es un equipo principal, desde el cual se ejecutan los servicios de OpenNebula, es el encargado de administrar la infraestructura de cómputo.

Hosts: Son los equipos que cuentan con alguno de los hipervisores soportados por OpenNebula y que proveerán los recursos que utilizarán las máquinas virtuales.

Datastores: Dispositivos que almacenan las imágenes²³ base de las máquinas virtuales

Service Network: Es la red física que será usada para el soporte de los servicios básicos: la interconexión entre los servidores de almacenamiento y las operaciones de control de OpenNebula.

VM Networks: Es una red física que soporta las redes virtuales (VLAN) para las máquinas virtuales.

En la Figura 9 se presenta un esquema del tipo de infraestructura básica presentada por el proyecto OpenNebula para el montaje de un servicio IaaS, en este, los componentes se encuentran claramente delimitados según su función final.

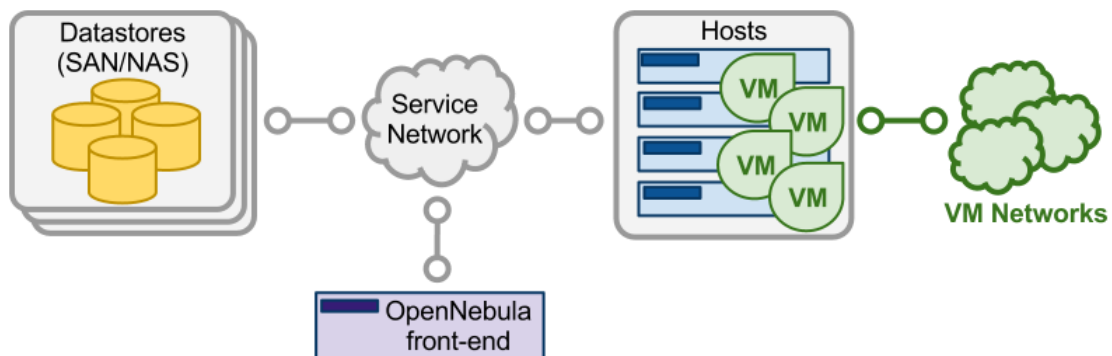


Figura 9: Componentes de OpenNebula

Fuente: <http://openebula.org/documentation:rel3.4:plan>

5.4. Integración con el modelo

Para el proceso de integración, fue necesario realizar cambios sobre la infraestructura que plantea el proyecto OpenNebula, esto, ya que el presente proyecto no

²³ Imágenes, hace referencia a los discos virtuales usados por los hipervisores.

cuenta con algunos de los componentes sugeridos y no existía la posibilidad de adquirirlos.

Los diferentes subsistemas de OpenNebula fueron adaptados de la siguiente manera:

Front-end: El equipo principal para el proyecto representó un inconveniente inicial para el montaje del proyecto, esto a razón de como se mencionó en el capítulo 0, los servidores sobre los cuales se trabajaría ya eran parte de otro proyecto, lo cual limitaba las labores administrativas que se podían realizar sobre este. Como el equipo frontend requiere un nivel de control total por parte del administrador del servicio de IaaS se decidió que la mejor manera era usar un servidor virtualizado que hiciera las veces de frontend, de esta manera se contaría con administración completa y no se crearían inconvenientes con la administración del servidor físico.

Hosts: Para los hosts, fueron acondicionados los dos equipos con los que se contaba, esta labor no presentó cambios sobre el subsistema original.

Datastores: Al no contar con hardware dedicado al almacenamiento de las imágenes virtuales, este componente se integró a la funcionalidad de los hosts.

Service Network: Para la red de soporte de los servicios, se usó la red sobre la cual se encuentran los equipos dentro de la topología usada por la Universidad.

VM Networks: La red sobre la cual se encuentran los servicios virtualizados, se construyó como una red directa sobre una conexión de 1Gbps.

Para describir el modelo final del prototipo, la Figura 10 muestra el modelo adaptado. El servicio de computación en la nube se denominó cloudEISI y corresponde al modelo de despliegue de nube privada.

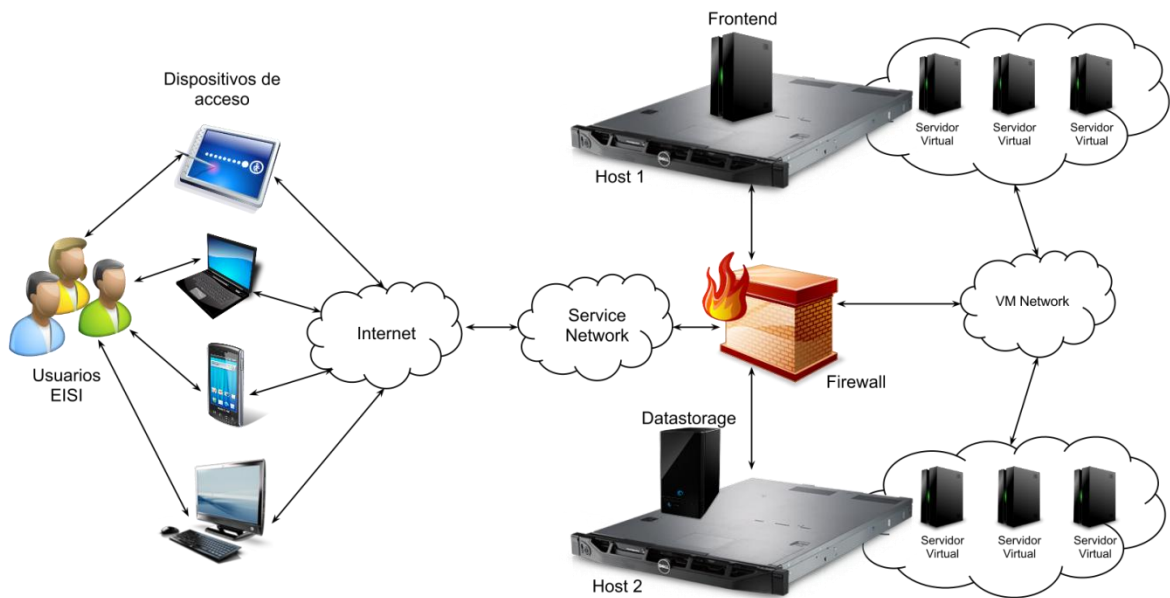


Figura 10: Modelo del prototipo de IaaS para la EISI
Fuente: Autor.

5.5. Implementación de la herramienta

La fase de implementación de OpenNebula como herramienta para la prestación de un servicio representó una tarea de dificultad media, ya que en el sitio web del proyecto la documentación existente es muy puntual en algunos aspectos, pero también ambigua, esto como resultado de que la implementación se pueda realizar en 4 sistemas operativos diferentes, teniendo cada uno una sintaxis diferente que no está cubierta en dicha documentación, esta limitación llevó a la necesidad de fortalecer los conocimientos de administración de servidores para la continuidad del desarrollo del proyecto.

La implementación consistió en el montaje de las bases de datos necesarias por la herramienta, la instalación del hipervisor en los hosts y la adecuación de estos para responder a la administración del frontend; además de lo anterior la configuración de la herramienta es un proceso que requiere tener completamente planea-

da la estructura del modelo de del servicio a prestar y el conocimiento completo del hardware con el cual se cuenta.

Luego de la instalación de OpenNebula y de realizar pruebas que permitieran comprobar la comunicación entre los diferentes host con el frontend se procedió al proceso de aprendizaje del uso de la plataforma.

La interfaz principal de OpenNebula, Figura 11, permite la administración completa de los recursos de una manera clara y precisa, a continuación se presenta una pequeña descripción de su funcionalidad.

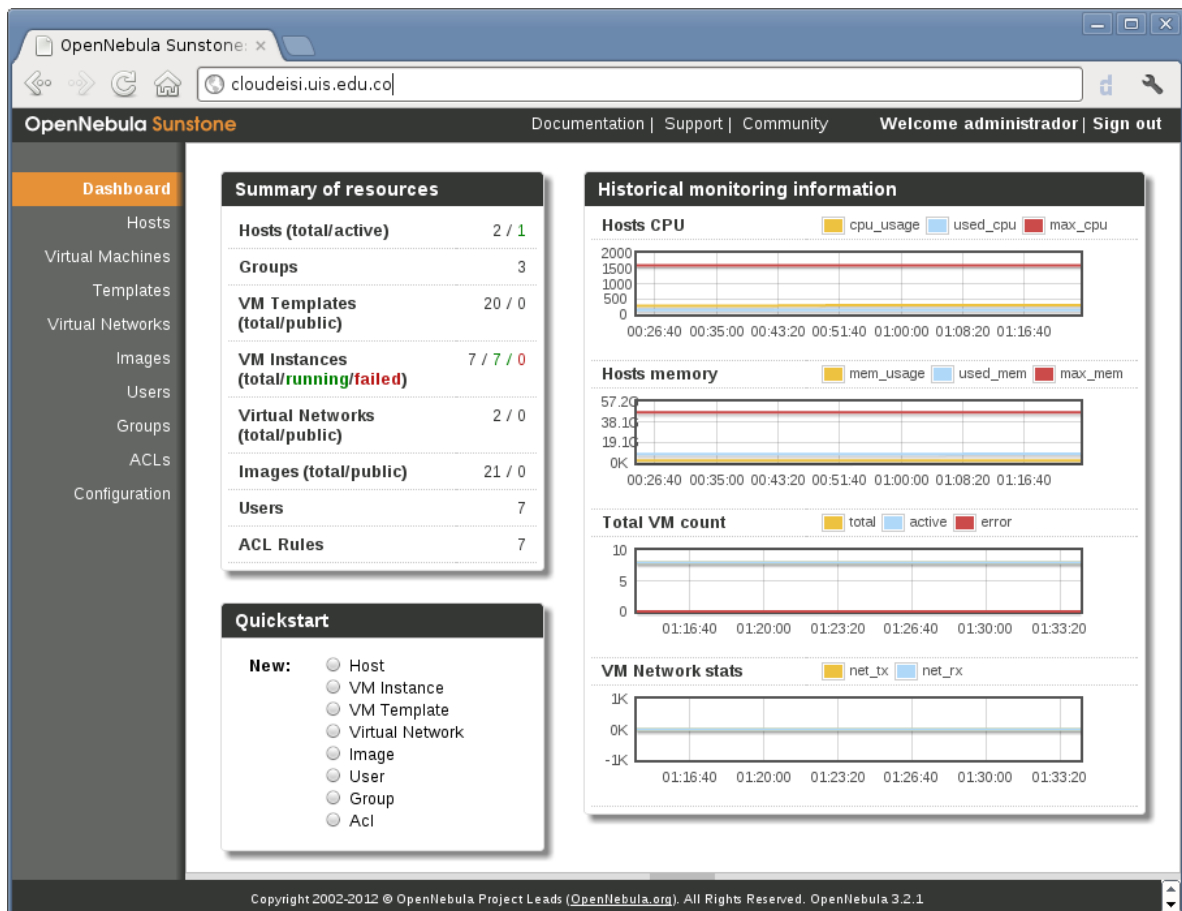


Figura 11: Interfaz de administración de OpenNebula

Fuente: Autor.

OpenNebula cuenta con una interfaz administrativa que permite la creación de la configuración básica de cada uno de los diferentes recursos con los que cuenta la infraestructura.

La interfaz administrativa permite agregar de una manera sencilla nuevos equipos tipo *host*, los cuales inmediatamente pasan al conjunto de recursos compartidos de la plataforma; los subsistemas de imágenes, redes y máquinas virtuales se definen por medio de un formato de texto plano parametrizado llamado *template* (plantilla), estas plantillas, contienen los datos necesarios para que cada recurso pueda ser instanciado las veces que se sea necesario mientras los recursos no estén completamente usados, otra función importante es la administración de *usuarios*.

Usuarios son todos aquellos que puedan hacer uso de los recursos computacionales por medio de la interfaz de administración de usuarios; estos usuarios a su vez pueden pertenecer a *grupos*, esto con varios fines, uno de estos es generar un grupo de recursos administrados por un único usuario que comparta con otros sus recursos, estos grupos y usuarios pueden tener varias limitaciones según la configuración que se establezca en la opción *ACL*, esta opción representa las Listas de Control de Usuario (ACL por sus siglas en inglés), en base a los permisos asignados a cada usuario o grupo estos pueden tener la capacidad de crear nuevas plantillas de máquinas virtuales o dado el caso estar limitados solo a usarlas, también permitirles usar cierto tipo de redes virtuales o limitar la cantidad de recursos de los que pueden disponer.

Para que los usuarios puedan ingresar a la plataforma, se cuenta con otro portal que permite que puedan acceder a los recursos disponibles y administrar aquellos que ya está usando, esta interfaz es llamada interfaz self-service (auto servicio). La interfaz de auto-servicio, Figura 12, está diseñada como una simplificación de la interfaz de administración, en esta el usuario selecciona de una serie de planti-

llas predefinidas los recursos que desea instanciar, sin la necesidad de tener conocimiento del hardware en el cual se va a desplegar la máquina virtual ni la red de la cual va a hacer uso, aunque tiene la opción de hacerlo; es una solución de seleccionar y usar.

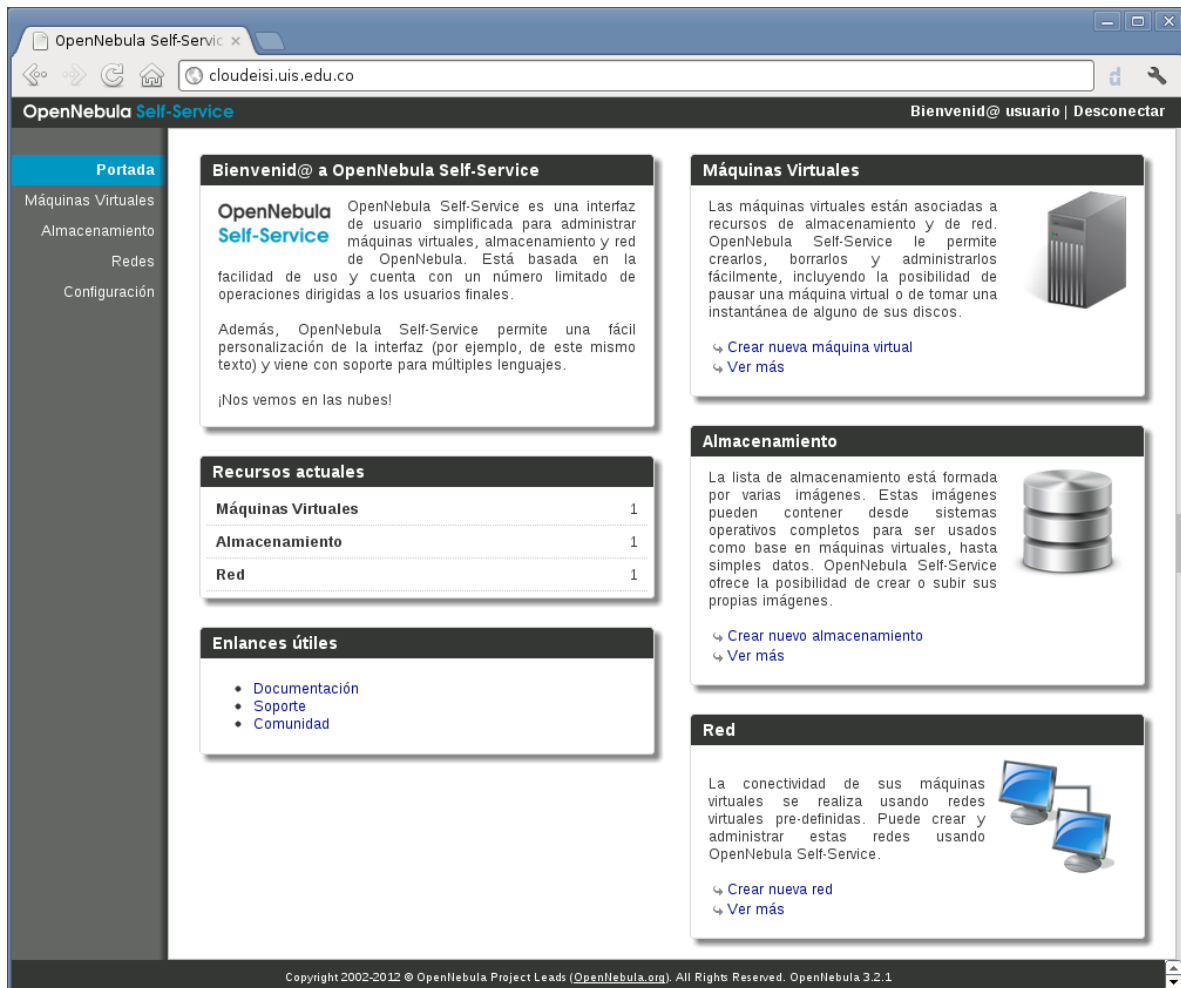


Figura 12: Interfaz de usuario del portal de Auto-servicio
Fuente: Autor.

5.6. Pruebas de funcionamiento

Antes de pasar a la puesta en marcha del prototipo ,se realizaron algunas pruebas básicas del comportamiento de la plataforma, en ella se evidenció que existía un

problema con la forma en que las máquinas virtuales estaban siendo instanciadas, inicialmente se creyó que era un problema de la instalación o del montaje de la herramienta, sin embargo fue necesario hacer un proceso de depuración que permitió determinar la causa del problema, un error en un archivo del código fuente del plugin del hipervisor; con esta información y pensando en los demás usuarios del plugin, se hizo contacto con el desarrollador de este y se realizaron los cambios en los repositorios oficiales, también se generó una bifurcación del proyecto para continuar con el desarrollo producto del presente proyecto.

Superado el inconveniente anteriormente mencionado, se realizaron pruebas de instancias múltiples en los 2 hosts y se realizaron pruebas de suspensión, guardado, apagado y migración de algunas máquinas virtuales; la migración con el hipervisor usado tiene una limitación en la plataforma OpenNebula que se evidenció durante las pruebas, no se permite la migración en “vivo”, esto quiere decir que la máquina guarda el estado actual, es apagada, se migra al nodo destino y finalmente vuelve a iniciar en el estado almacenado.

Las pruebas de funcionamiento demostraron que las diferentes funcionalidades esperadas de la herramienta se estaban realizando correctamente, lo cual significó el paso a la siguiente etapa del proyecto.

6. PUESTA EN MARCHA DEL PROTOTIPO

La puesta en marcha del prototipo está dividida en tres frentes de acción, el primero relativo a la seguridad de los equipos físicos en los cuales se apoya, segundo una serie de métricas y características extras del prototipo que permitan una prestación del servicio de acuerdo adecuada y una parte final relacionada con las condiciones de uso y las políticas que rigen la prestación del servicio.

6.1. Políticas de seguridad

Como se mencionó en el numeral 5.3.3, el prototipo es administrado desde un servidor principal o frontend, este servidor se encarga de la ejecución de los servicios, la monitorización de los hosts y es el encargado de la seguridad del modelo.

Para comprometer al mínimo la seguridad del frontend se determinaron una serie de requerimientos de uso que permitan el mayor grado de control sobre su administración y que a su vez faciliten una labor tan importante como esta.

Los parámetros definidos fueron los siguientes.

- Aislar la red virtualizada de la red de la universidad, es decir, solo se permite acceso a los servicios solicitados por cada una de las instancias y así mismo solo estos servicios tienen acceso a la red externa.
- Automatizar las labores de detección de intrusos y generar alertas administrativas en los casos que sean necesarios.
- Generar automáticamente resúmenes de registros del sistema y de eventos anómalos.
- Permitir la sincronización de la hora de todas las instancias virtuales por medio del frontend para permitir la realización de tareas administrativas programadas.

- Facilitar la administración de redes virtuales con la implementación de un servicio en el frontend.
- Ejecutar tareas de copias de seguridad remotas de la configuración del sistema, las bases de datos y los scripts de mantenimiento.

La tarea de aislar la red virtualizada es una característica fundamental de la seguridad del sistema, durante las pruebas iniciales del sistema, se notó que los intentos de intrusión por parte de equipos externos y algunos de la red interna de la Universidad podrían perjudicar el desempeño del prototipo, por esta razón, la limitación impuesta para las redes responde a las necesidades de cada servicio, y en lugar de convertirse en un inconveniente presento una solución directa a dicho problema.

Los intentos de intrusión se siguieron presentando ya no contra la red virtualizada si no contra el equipo frontend, por esta razón se instalaron herramientas que además de reportar los ataques realizaran una acción preventiva, estas herramientas fueron fail2ban²⁴ y logwatch²⁵; fail2ban es una herramienta para la prevención de intrusiones, que permite por medio de la comparación de una serie de reglas sobre los registros del sistema, realizar tareas de bloqueo y generación de alertas, cubre áreas tan variadas del sistema como fallos de autenticación de usuarios hasta análisis de peticiones al servidor web apache. Para el uso de esta herramienta se optimizaron algunos de los filtros que incluye por defecto la aplicación y se generaron otros para cubrir áreas que se presentaron como requisito. Logwatch es una herramienta de análisis de los archivos de registro del sistema, permite enviar un resumen diario de las diferentes incidencias que se han presentado sobre el servidor y en casos muy puntuales produce advertencias extras sobre dichos resultados.

²⁴ www.fail2ban.org

²⁵ www.logwatch.org

Como parte de la seguridad de la red, se implementó también un servicio de hora (NTP²⁶) que permite que las máquinas instanciadas se encuentren en sincronía con el frontend, se consideró la importancia de este punto, ya que el reloj interno de las máquinas virtuales no es preciso y algunas veces debe ser calibrado, esta diferencia de tiempo puede evitar que ciertos trabajos programados en las máquinas virtuales se ejecuten correctamente.

Otra tarea de seguridad que fue implementada fue la automatización de las copias de seguridad de algunos archivos críticos del sistema, entre estos el estado del firewall del sistema, la base de datos de OpenNebula y otros archivos de configuración, estas copias se almacenan local y remotamente como medida almacenamiento redundante.

6.2. Acceso de los usuarios

Con las medidas de seguridad que fueron implementadas en el servidor, queda aun otra variable: los usuarios del sistema. El acceso permitido para ellos está determinado por el tipo de instancia utilizada, es decir, un usuario que necesite una instancia virtual de un servidor de base de datos solo tendrá acceso a su instancia por medio de los puertos necesarios para uso y administración; mientras que un usuario que tenga una instancia virtual de un servidor web necesitará un puerto para acceso no seguro, otro para acceso seguro, un puerto para administración y un puerto para alojar archivos; sin embargo, el hecho de que una instancia virtual tenga definidos los puertos que usará no significa que si un usuario requiere por algún caso en particular acceso a otro puerto no pueda ser otorgado.

La seguridad de cada servidor puede verse comprometida por muchas causas, pero se puede esperar que una de las mayores causas sea por la inexperiencia en

²⁶ http://es.wikipedia.org/wiki/Network_Time_Protocol

administración de este tipo de equipos (aún los virtuales), es por esto que el grupo CONUSS, pone a disposición de los usuarios de la plataforma sus conocimientos en estas labores, y constituye una parte fundamental del proceso de adopción de este paradigma de cómputo para la EISI.

6.3. Carga máxima soportada

Las pruebas de soporte de la plataforma tuvieron como fin determinar el máximo de máquinas virtuales soportadas por la infraestructura, para este fin se acondicionaron unas instancias de máquinas virtuales que permitieran medir ciertos parámetros funcionales de los servidores, del entorno, del uso de los recursos y de las máquinas en sí.

Para recopilar la información producida durante las pruebas se utilizó la aplicación ganglia²⁷, ganglia es un sistema de monitoreo distribuido escalable para sistemas de cómputo de alto desempeño, permite llevar registros históricos de diferentes parámetros del sistema y representarlos gráficamente, además permite ejecutar análisis del comportamiento y estado de los equipos por parte de los administradores; la configuración de la herramienta fue modificada para que almacenara también los datos generados por la plataforma y otras métricas no incluidas por defecto en la aplicación.

6.3.1 Cantidad de máquinas instanciadas

Como base para calcular el máximo de instancias concurrentes, se establecieron las diferentes configuraciones de hardware virtualizado con las cuales trabajaría la plataforma, esto con el fin de permitir la mayor cantidad de usuarios compartiendo

²⁷ <http://ganglia.sourceforge.net/>

el mismo hardware sin comprometer la estabilidad del equipo; las diferentes plantillas se presentan en la Tabla 4.

Tabla 4: Plantillas de auto servicio

Nombre Plantilla	Tamaño Memoria	Capacidad HDD	Capacidad CPU
small	256 MB	10 GB	0,25 Núcleos
medium	512 MB	20 GB	0,33 Núcleos
large	756 MB	20 GB	0,50 Núcleos

Fuente: Autor.

Realizando una proyección del máximo de máquinas directamente sobre los valores de las plantillas, se obtuvo lo siguiente:

Cantidad Limitada por el uso de CPU:

- En el mejor de los casos $8 \text{ núcleos} / 0,25 \text{ núcleos} \times \text{VM}^{28} = 32 \text{ VM}$
- En el peor de los casos $8 \text{ núcleos} / 0,5 \text{ núcleos} \times \text{VM} = 16 \text{ VM}$

Cantidad Limitada por el uso de memoria RAM

- En el mejor de los casos $24 \text{ GB} / 0,256 \text{ GB} \times \text{VM} = 93 \text{ VM}$
- En el peor de los casos $24 \text{ GB} / 0,768 \text{ GB} \times \text{VM} = 31 \text{ VM}$

Las especificaciones anteriormente mostradas representan unas características básicas de funcionamiento, esto quiere decir que en caso de ser necesarias máquinas con características diferentes, estas pueden ser creadas.

6.3.2 Uso de CPU

Para determinar el número máximo real de máquinas soportado por un host en un escenario particular, se adaptó una instancia de un servidor LAMP para que realizara una serie de operaciones matemáticas consecutivas, y de esta manera se

²⁸ VM se usará en adelante en el presente texto para hacer referencia a una máquina virtual.

obtuvo según el uso de CPU, RAM y potencia, la carga máxima soportada para máquinas tipo *small*. Las pruebas se realizaron durante un periodo de siete horas.

En la Figura 13 se observa, una representación del uso de la CPU durante la prueba, este aumentó a medida que se instanciaron máquinas, el procedimiento realizado fue la instanciación de 5 máquinas virtuales cada 5 minutos hasta llegar a un punto superior al 75% del uso máximo de la CPU, y posteriormente se instanció cada 3 minutos una máquina extra hasta superar el 80%. El eje vertical representa el porcentaje del uso de la CPU y el eje horizontal, un periodo de 10 horas, desde las 00:00 hasta las 10:00, las pruebas inician a partir de las 03:00, se muestra un periodo más largo a las 7 horas de duración de la prueba para contrastar con el uso del equipo en estado ocioso; el eje se utiliza en las próximas 4 figuras.

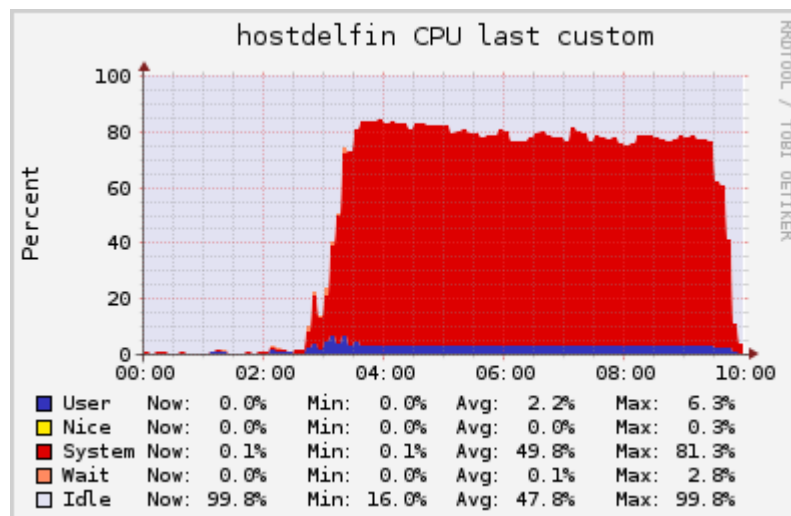


Figura 13: Uso de la CPU durante las pruebas.

Fuente: Autor.

Al llegar a 34 máquinas virtuales como está representado por el eje vertical en la Figura 14, se comprobó que se podía llegar hasta el límite teórico; luego de este punto y observar que la carga de CPU se mantuvo se realizaron pruebas de conexión y respuesta del host, esto con el fin de comprobar que no se había perdido

control sobre el equipo. Los resultados muestran que el host alcanzó un máximo de 81.3% de uso, y según los registros del sistema, un promedio de 64.9% durante las 7 horas de la prueba.

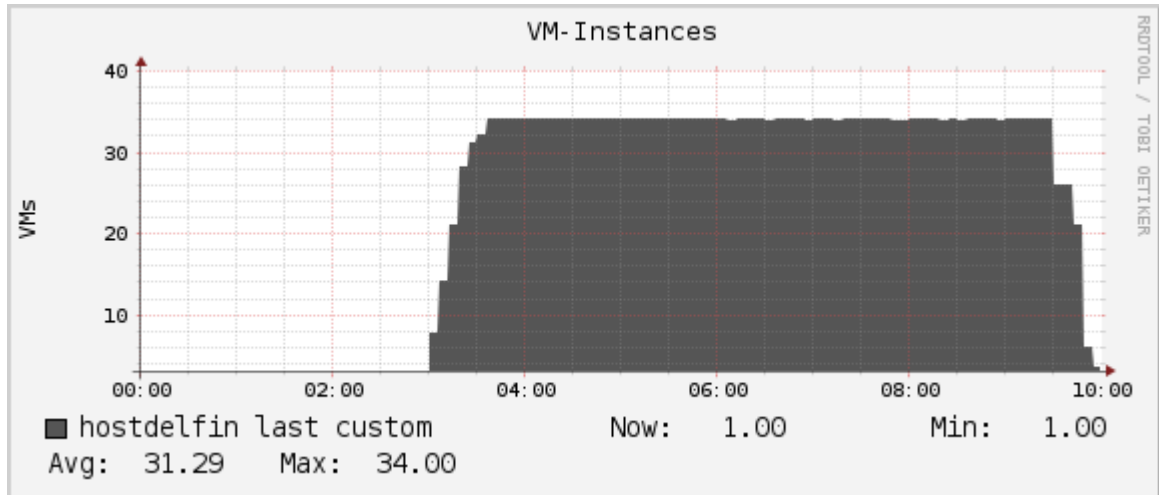


Figura 14: Número de máquinas virtuales instanciadas durante las pruebas.

Fuente: Autor.

6.3.3 Uso de Memoria RAM

El uso esperado de la memoria RAM durante la prueba realizada fue de 8.5 GB, esto al tener 34 instancias en ejecución, es decir $34 \text{ VM} \times 0,25 \text{ GB} / \text{VM} = 8.5 \text{ GB}$. En la Figura 15, en el eje vertical, se encuentra expresado en giga bytes el comportamiento del uso de la RAM durante las pruebas, el uso de la memoria aparece como máximo 8.8 GB, lo cual concuerda con el valor calculado; a este valor es necesario sustraerle el uso de memoria generado por el sistema operativo, en este caso 1.6 GB, valor tomado del host durante el periodo anterior a la prueba, lo cual produce un uso de memoria total por las instancias de 7.2 GB, esto se debe a que estas solo consumen la memoria que van necesitando, no toda la asignada desde el momento de su inicio aunque ésta sea reservada.

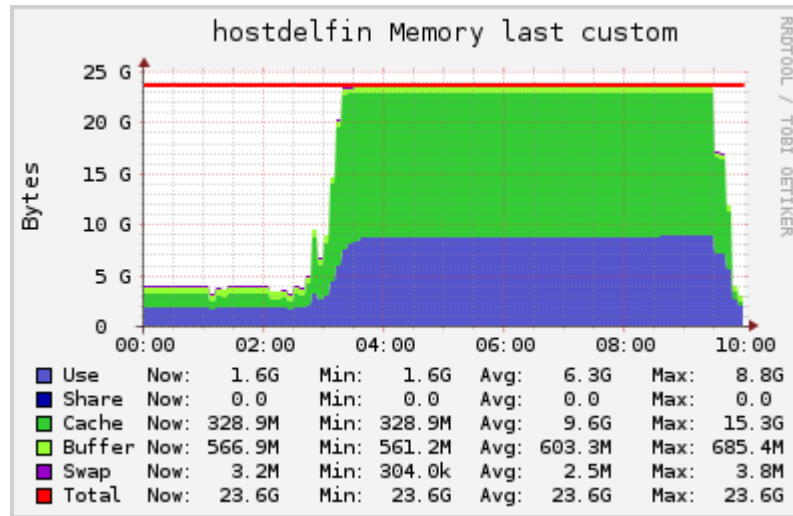


Figura 15: Uso de memoria RAM durante las pruebas.
Fuente: Autor.

6.3.4 Consumo de energía

Otro factor que se consideró importante analizar fue la potencia total consumida por el sistema, se configuraron los servidores host para obtener este valor y así poder hacer estimaciones del grado de eficiencia del modelo, la Figura 16 muestra el comportamiento del consumo de energía durante la prueba, en esta, el eje vertical representa el consumo en vatios-hora del host.

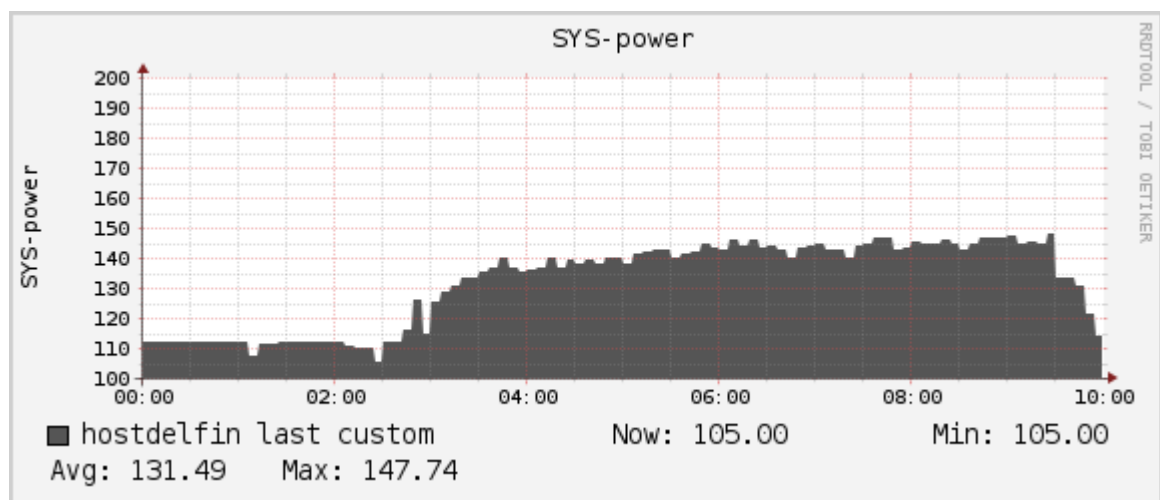


Figura 16: Potencia consumida por el sistema durante las pruebas.
Fuente: Autor.

El host en estado de uso mínimo tiene un consumo eléctrico de 105 Wh²⁹, con las 34 máquinas instanciadas anteriormente el consumo aumenta 147.74 Wh, esto quiere decir que el sistema está usando 42.74 Wh de más para ejecutar las 34 máquinas virtuales, lo cual equivale a 1.25 Wh por cada máquina instanciada.

El consumo generado por tener 34 equipos reales funcionando para prestar los servicios que se simularon se puede aproximar de la siguiente manera, la plantilla utilizada corresponde a una instancia usando 0,25 núcleos de la máquina real, esto es igual a un procesador de 667 MHz, un procesador similar en frecuencia de reloj³⁰ tiene un consumo de 17.5 Wh, esto equivale a un consumo de 595 Wh únicamente en procesador, es decir sin tener en cuenta los demás componentes de un equipo real; simulando el consumo de un equipo real con este tipo de procesador y usando una herramienta para el cálculo aproximado del consumo de energía³¹, se obtiene un valor aproximado de 86 Wh, es decir 2856 Wh para 34 equipos equivalentes a los simulados, aproximadamente 19.36 veces más; ahora si en lugar de ser máquinas con tan limitada capacidad estas máquinas correspondieran a equipos con características similares al host, con un consumo de 105 Wh, el consumo se eleva a 3570 Wh para tener a 34 equipos reales en funcionamiento, un consumo 24.20 veces mayor.

Trasladando estos resultados a costo monetario para un periodo de un mes y usando como precio de KWh el correspondiente al mes de febrero de 2012³² por parte del prestador del servicio para el nivel de tensión que posee el edificio de

²⁹ Wh ó Vatio-hora es la unidad usada para medir el consumo eléctrico.

³⁰ <http://ark.intel.com/products/27548/Intel-Pentium-III-Processor-667-MHz-256K-Cache-133-MHz-FSB>

³¹ <http://extreme.outervision.com/PSUEngine>

³² http://www.essa.com.co/essa/wrm/tariffs_documents/TARIFAS_ESSA_FEBRERO_2012.pdf

tecnologías CENTIC³³, nivel II³⁴, el cual tiene un valor de \$341.32 KWh, se obtienen los siguientes valores:

Un mes de 30 días = 720 horas

- Escenario 1

1 Servidor virtualizando 34 máquinas

$$1 * 0.148 \text{ KWh} * 720 = 106.37 \text{ KWh/mes} * \$341.32/\text{KWh} = \$36.307,45 / \text{mes}$$

- Escenario 2

34 equipos de bajas especificaciones

$$34 * 0.086 \text{ KWh} * 720 = 2105.28 \text{ KWh/mes} * \$341.32/\text{KWh} = \$718.579,85 / \text{mes}$$

- Escenario 3

34 equipos de especificaciones similares al host

$$34 * 0.105 \text{ KWh} * 720 = 2570.40 \text{ KWh/mes} * \$341.32/\text{KWh} = \$ 877.335,87/\text{mes}$$

En la Figura 17 se agruparon los costos calculados para comparar los valores obtenidos, cabe anotar que entre los costos energéticos también se deben incluir costos externos como son los costos de refrigeración del lugar; por otra parte el espacio ocupado por dichos equipos amplía el espacio a climatizar a razón de cantidad de estos, por consiguiente el consumo de energía aumenta aun más en los escenarios en los cuales se contempla un mayor número de equipos reales.

³³ Centro de Tecnologías de Información y Comunicación

³⁴ https://www.uis.edu.co/procesos_contratacion/contrataciones/licitaciones/licitaciones_2005/licitacion_028/nueva/V2_PD_L028.pdf

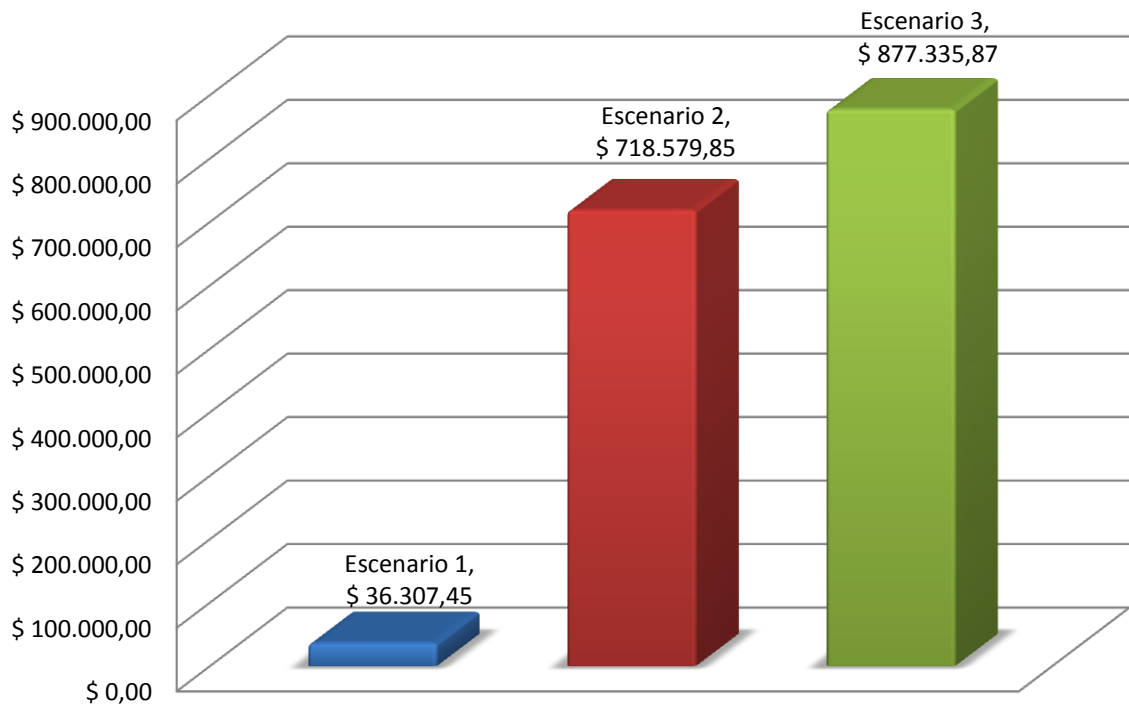


Figura 17: Costo energético aproximado.
Fuente: Autor.

6.4. Pruebas de funcionamiento

Luego de las pruebas para calcular la carga máxima soportada por el sistema y la eficiencia energética, se procedió a medir los tiempos de inicio de las máquinas instanciadas, el tiempo que requieren para migrar a otro host y la cantidad de usuarios simultáneos en un escenario simulado.

6.4.1 Pruebas de instanciación

Para la realización de las pruebas de instanciación se tomó el registro producido por la herramienta durante el proceso de lanzamiento de cada máquina, este registra inicialmente con una marca de tiempo el momento en que la instancia pasa por cuatro estados, ACTIVE, PROLOG, BOOT y RUNNING. Los estados que inte-

resan para este análisis es el tiempo que toma para el sistema pasar del estado ACTIVE, momento en el cual se inicia la instanciación, al estado RUNNING, punto en el cual la máquina ya se encuentra en un estado utilizable por el usuario final, y con estos valores determinar el tiempo promedio de lanzamiento de cada máquina, para este proceso se instanciaron 30 máquinas con plantilla tipo small, los datos obtenidos se presentan en la Tabla 5.

Tabla 5: Tiempo de duración del proceso de instanciación

MV	Fecha	Hora	Estado Inic	Hora	Estado Final	Duración
429.log	Mon Apr 30 2012	3:00:28	ACTIVE.	3:01:03	RUNNING	0:00:35
430.log	Mon Apr 30 2012	3:02:28	ACTIVE.	3:03:06	RUNNING	0:00:38
431.log	Mon Apr 30 2012	3:00:58	ACTIVE.	3:01:36	RUNNING	0:00:38
432.log	Mon Apr 30 2012	3:02:58	ACTIVE.	3:03:41	RUNNING	0:00:43
433.log	Mon Apr 30 2012	3:01:28	ACTIVE.	3:02:08	RUNNING	0:00:40
434.log	Mon Apr 30 2012	3:03:28	ACTIVE.	3:04:09	RUNNING	0:00:41
435.log	Mon Apr 30 2012	3:01:58	ACTIVE.	3:02:41	RUNNING	0:00:43
436.log	Mon Apr 30 2012	3:03:58	ACTIVE.	3:04:41	RUNNING	0:00:43
437.log	Mon Apr 30 2012	3:04:28	ACTIVE.	3:05:07	RUNNING	0:00:39
438.log	Mon Apr 30 2012	3:04:58	ACTIVE.	3:05:39	RUNNING	0:00:41
439.log	Mon Apr 30 2012	3:05:28	ACTIVE.	3:06:11	RUNNING	0:00:43
440.log	Mon Apr 30 2012	3:05:58	ACTIVE.	3:06:39	RUNNING	0:00:41
441.log	Mon Apr 30 2012	3:06:28	ACTIVE.	3:07:10	RUNNING	0:00:42
442.log	Mon Apr 30 2012	3:06:58	ACTIVE.	3:07:41	RUNNING	0:00:43
443.log	Mon Apr 30 2012	3:07:28	ACTIVE.	3:08:12	RUNNING	0:00:44
444.log	Mon Apr 30 2012	3:07:58	ACTIVE.	3:08:42	RUNNING	0:00:44
445.log	Mon Apr 30 2012	3:08:28	ACTIVE.	3:09:10	RUNNING	0:00:42
446.log	Mon Apr 30 2012	3:08:58	ACTIVE.	3:09:40	RUNNING	0:00:42
447.log	Mon Apr 30 2012	3:09:28	ACTIVE.	3:10:07	RUNNING	0:00:39
448.log	Mon Apr 30 2012	3:09:58	ACTIVE.	3:10:42	RUNNING	0:00:44
449.log	Mon Apr 30 2012	3:10:28	ACTIVE.	3:11:11	RUNNING	0:00:43
450.log	Mon Apr 30 2012	3:10:58	ACTIVE.	3:11:36	RUNNING	0:00:38
451.log	Mon Apr 30 2012	3:11:28	ACTIVE.	3:12:12	RUNNING	0:00:44
452.log	Mon Apr 30 2012	3:11:58	ACTIVE.	3:12:40	RUNNING	0:00:42
453.log	Mon Apr 30 2012	3:12:28	ACTIVE.	3:13:10	RUNNING	0:00:42
454.log	Mon Apr 30 2012	3:12:58	ACTIVE.	3:13:43	RUNNING	0:00:45
455.log	Mon Apr 30 2012	3:13:28	ACTIVE.	3:14:13	RUNNING	0:00:45
456.log	Mon Apr 30 2012	3:13:58	ACTIVE.	3:14:34	RUNNING	0:00:36

MV	Fecha	Hora	Estado Inic	Hora	Estado Final	Duración
457.log	Mon Apr 30 2012	3:14:28	ACTIVE.	3:15:10	RUNNING	0:00:42
458.log	Mon Apr 30 2012	3:14:58	ACTIVE.	3:15:45	RUNNING	0:00:47
459.log	Mon Apr 30 2012	3:17:58	ACTIVE.	3:18:34	RUNNING	0:00:36

Promedio: 0:00:41
Desv Estándar: 3,3389E-05

Fuente: Autor.

Los resultados obtenidos muestran que en promedio las 30 máquinas virtuales creadas estaban listas para ser usadas 41 segundos después de ser instanciadas.

6.4.2 Pruebas de migración

Para las pruebas de migración, los registros de cada máquina instanciada agregan dos nuevos estados, SAVE_MIGRATE, en el cual se guarda el estado de la máquina y se apaga y PROLOG_MIGRATE, con el cual los archivos de la instancia virtual son transportados del host origen de la migración al host destino. Los estados que son representativos en la prueba de migración son el estado SAVE_MIGRATE y su cambio al estado RUNNING.

La forma de análisis del proceso de migración, el cual se espera que se ejecute sobre una instancia que ya se encuentra en funcionamiento, se enfocó en el tiempo de respuesta de la instancia, es decir, si la instancia presta un servicio, cuánto tiempo esta no estará disponible. Inicialmente se tomó el tiempo que aparece en los registros para determinar el tiempo promedio de la migración como se muestra en la

Tabla 6.

Tabla 6: Tiempo de duración del proceso de migración

MV	Fecha	Hora	Estado Inicial	Hora	Estado Final	Duración
326.log	Wed Apr 25 2012	01:06:19	SAVE_MIGRATE	01:06:54	RUNNING	00:00:35
326.log	Wed Apr 25 2012	01:12:22	SAVE_MIGRATE	01:12:57	RUNNING	00:00:35
326.log	Wed Apr 25 2012	01:17:41	SAVE_MIGRATE	01:18:15	RUNNING	00:00:34
326.log	Wed Apr 25 2012	01:19:32	SAVE_MIGRATE	01:20:06	RUNNING	00:00:34

MV	Fecha	Hora	Estado Inicial	Hora	Estado Final	Duración
326.log	Wed Apr 25 2012	02:06:16	SAVE_MIGRATE	02:06:51	RUNNING	00:00:35
326.log	Wed Apr 25 2012	02:08:31	SAVE_MIGRATE	02:09:05	RUNNING	00:00:34
326.log	Wed Apr 25 2012	02:10:23	SAVE_MIGRATE	02:10:59	RUNNING	00:00:36
326.log	Wed Apr 25 2012	02:13:39	SAVE_MIGRATE	02:14:11	RUNNING	00:00:32
326.log	Wed Apr 25 2012	02:16:09	SAVE_MIGRATE	02:16:45	RUNNING	00:00:36
326.log	Wed Apr 25 2012	02:18:58	SAVE_MIGRATE	02:19:36	RUNNING	00:00:38
326.log	Wed Apr 25 2012	02:22:27	SAVE_MIGRATE	02:23:02	RUNNING	00:00:35
326.log	Wed Apr 25 2012	02:28:44	SAVE_MIGRATE	02:29:18	RUNNING	00:00:34

Promedio 00:00:35

Desviación estándar 1,698E-05

Fuente: Autor.

Se puede observar que los tiempos obtenidos para la migración son aún más bajos que los del proceso de instanciación, esto se debe a que no realiza el proceso de clonación³⁵ sobre la imagen que se está instanciando. Con este tiempo promedio se procedió a obtener el tiempo que la instancia no presta el servicio, para esto se realizaron 2 pruebas simultáneas, una desde el mismo servidor y otra desde una conexión externa. La prueba consiste en determinar cuantos segundos transcurren desde que la máquina entra al estado SAVE_MIGRATE, en el cual es apagada, hasta que pueda volver a tener conexión con la red de datos, lo cual le permite responder al servicio virtualizado.

Desde el host se uso la herramienta ping³⁶, que permite diagnosticar el estado, velocidad y calidad de una conexión de red. Se almacenaron los resultados de la ejecución de un ping cada segundo junto al cambio del estado de la instancia y de esa manera determinar en qué momento se recuperaba la conexión, los resultados se muestran en la Tabla 7.

Tabla 7: Tiempos de recuperación de la conexión local en una migración.

Fecha	Hora	Resultado ping	Duración
2012-Abr-25	01:17:41	64 bytes from 10.0.0.1 icmp_req=1 ttl=64 time=0.540 ms	00:00:45
		Timeout	
2012-Abr-25	01:18:26	64 bytes from 10.0.0.1 icmp_req=1 ttl=64 time=0.339 ms	

³⁵ La clonación es el proceso de copia de una imagen base para su posterior instanciación.

³⁶ <http://linux.die.net/man/8/ping>

Fecha	Hora	Resultado ping	Duración
2012-Abr-25	01:19:32	64 bytes from 10.0.0.1 icmp_req=1 ttl=64 time=0.336 ms	
		Timeout	
2012-Abr-25	01:20:14	64 bytes from 10.0.0.1 icmp_req=1 ttl=64 time=0.520 ms	00:00:42
2012-Abr-25	02:18:58	64 bytes from 10.0.0.1 icmp_req=1 ttl=64 time=0.665 ms	
		Timeout	
2012-Abr-25	02:19:46	64 bytes from 10.0.0.1 icmp_req=1 ttl=64 time=0.523 ms	00:00:48
2012-Abr-25	02:22:26	64 bytes from 10.0.0.1 icmp_req=1 ttl=64 time=0.581 ms	
		Timeout	
2012-Abr-25	02:23:10	64 bytes from 10.0.0.1 icmp_req=1 ttl=64 time=0.429 ms	00:00:44
2012-Abr-25	02:28:44	64 bytes from 10.0.0.1 icmp_req=1 ttl=64 time=0.360 ms	
		Timeout	
2012-Abr-25	02:29:25	64 bytes from 10.0.0.1 icmp_req=1 ttl=64 time=0.623 ms	00:00:41
Promedio Timeout			00:00:44
Desviación estándar			3,16969E-05

Fuente: Autor.

Los valores marcados como Timeout, representan los mensajes de ping fallidos, estos fueron omitidos de la tabla con el fin de preservar solo los datos importantes de la prueba. Con los anteriores resultados se pudo comprobar que el tiempo de reconexión local es de aproximadamente 9 segundos sobre el tiempo de migración, es decir 44 segundos después de la migración el servicio se encuentra restablecido.

Para las pruebas de disponibilidad remotas se utilizó la herramienta netcat³⁷, con la cual realizan conexiones a diferentes puertos de los protocolos tcp y udp; con esta se hicieron pruebas cada segundo para determinar si la instancia virtual seguía enviando respuestas desde una conexión remota. La Tabla 8 muestra los datos obtenidos.

Tabla 8: Tiempos de recuperación de la disponibilidad del servicio

Fecha	Hora	Resultado netcat	Estado	Tiempo
2012-Abr-25	02:06:14	cloudeisi.uis.edu.co [200.16.117.73] 81	open	
2012-Abr-25	02:06:15	cloudeisi.uis.edu.co [200.16.117.73] 81	Connection timed out	
2012-Abr-25	02:06:16	New VM state is	SAVE_MIGRATE	
2012-Abr-25	02:06:37	cloudeisi.uis.edu.co [200.16.117.73] 81	Connection timed out	

³⁷ <http://netcat.sourceforge.net>

Fecha	Hora	Resultado netcat	Estado	Tiempo
2012-Abr-25	02:06:51	New VM state is	RUNNING	
2012-Abr-25	02:06:59	cloudeisi.uis.edu.co [200.16.117.73] 81	open	00:00:44
2012-Abr-25	02:08:29	cloudeisi.uis.edu.co [200.16.117.73] 81	open	
2012-Abr-25	02:08:30	cloudeisi.uis.edu.co [200.16.117.73] 81	Connection timed out	
2012-Abr-25	02:08:31	New VM state is	SAVE_MIGRATE	
2012-Abr-25	02:08:52	cloudeisi.uis.edu.co [200.16.117.73] 81	Connection timed out	
2012-Abr-25	02:09:05	New VM state is	RUNNING	
2012-Abr-25	02:09:14	cloudeisi.uis.edu.co [200.16.117.73] 81	open	00:00:44
2012-Abr-25	02:10:21	cloudeisi.uis.edu.co [200.16.117.73] 81	open	
2012-Abr-25	02:10:22	cloudeisi.uis.edu.co [200.16.117.73] 81	Connection timed out	
2012-Abr-25	02:10:23	New VM state is	SAVE_MIGRATE	
2012-Abr-25	02:10:44	cloudeisi.uis.edu.co [200.16.117.73] 81	Connection timed out	
2012-Abr-25	02:10:59	New VM state is	RUNNING	
2012-Abr-25	02:11:07	cloudeisi.uis.edu.co [200.16.117.73] 81	open	00:00:45
2012-Abr-25	02:13:38	cloudeisi.uis.edu.co [200.16.117.73] 81	open	
2012-Abr-25	02:13:39	cloudeisi.uis.edu.co [200.16.117.73] 81	Connection timed out	
2012-Abr-25	02:13:39	New VM state is	SAVE_MIGRATE	
2012-Abr-25	02:14:01	cloudeisi.uis.edu.co [200.16.117.73] 81	Connection timed out	
2012-Abr-25	02:14:11	New VM state is	RUNNING	
2012-Abr-25	02:14:23	cloudeisi.uis.edu.co [200.16.117.73] 81	open	00:00:44
2012-Abr-25	02:16:08	cloudeisi.uis.edu.co [200.16.117.73] 81	open	
2012-Abr-25	02:16:09	cloudeisi.uis.edu.co [200.16.117.73] 81	Connection timed out	
2012-Abr-25	02:16:09	New VM state is	SAVE_MIGRATE	
2012-Abr-25	02:16:31	cloudeisi.uis.edu.co [200.16.117.73] 81	Connection timed out	
2012-Abr-25	02:16:45	New VM state is	RUNNING	
2012-Abr-25	02:16:53	cloudeisi.uis.edu.co [200.16.117.73] 81	open	00:00:44
			Promedio timed out	00:00:44
			Desviación estándar	5,18E-06

Fuente: Autor.

Con los resultados obtenidos en las dos pruebas de disponibilidad, se puede establecer que el servicio se restablece en promedio transcurridos 44 segundos desde el inicio de una migración.

Los resultados obtenidos en las pruebas de instanciación y migración demostraron que el servicio ofrecido, para los requerimientos académicos actuales de la EISI, no presenta tiempos de lanzamiento o recuperación elevados.

6.5. Servicios virtualizados

Para agilizar el proceso de adopción de los servicios que puede prestar la plataforma, se pretendía inicialmente crear cada una de las imágenes virtuales para cada uno de los servicios que se quieren prestar, sin embargo esta tarea además de laboriosa no permitiría que el proyecto se enfocara en la prestación del servicio, disminuyendo la capacidad de acción de los administradores; como solución al anterior inconveniente, se determinó que la forma más apropiada de iniciar rápidamente el catálogo de servicios era utilizar imágenes de servicios virtuales que ya existieran en la red, para este fin se utilizó el repositorio de imágenes de TurnKey Linux³⁸, este es un proyecto de código abierto que ha desarrollado una biblioteca de imágenes de servidores basada en las distribuciones de linux debian³⁹ y ubuntu⁴⁰. Cada una de las imágenes puede ser instalada en servidores reales y también ofrecen versiones optimizadas para uso como servidores virtuales, al ser soluciones listas para usar, el tiempo requerido para integrarlas con el proyecto fue mínimo.

Las imágenes de TurnKey Linux están basadas en Ubuntu 10.04 LTS, esto quiere decir que dicha distribución provee soporte de actualizaciones de seguridad hasta el mes de abril del año 2015. Las actualizaciones de seguridad son instaladas automáticamente sobre dichas imágenes, a intervalos de tiempo generados al azar la primera vez que el servicio se ejecuta para evitar saturar la red con una actualización masiva, esta característica combinada con el firewall por defecto que posee cada imagen, provee una fuerte seguridad inicial, y no busca que se descuide la labor administrativa sobre ellas.

³⁸ <http://www.turnkeylinux.org>

³⁹ <http://www.debian.org>

⁴⁰ <http://www.ubuntu.com>

6.5.1 Biblioteca de Imágenes de servidores virtuales

En la biblioteca de TurnKey Linux se encuentran aproximadamente 50 imágenes de diferentes servidores virtualizados, de los cuales para la prestación del servicio para la EISI inicialmente se seleccionaron los siguientes:



Joomla⁴¹: Es un motor de portales dinámicos y gestor de contenidos. Es de código abierto, escrito en PHP.



Moodle⁴²: Es una plataforma de aprendizaje en línea (e-learning), también conocido como sistema de administración de cursos de código abierto.



PostgreSQL⁴³: Es un sistema de administración de bases de datos objeto-relacionales de código abierto que implementa la mayoría del estándar SQL:2008.



Tomcat⁴⁴: Es un servidor web y un contenedor de servlets⁴⁵ y JavaServer Pages⁴⁶ de código abierto, esta imagen contiene solo el servidor Tomcat.



Ruby on Rails⁴⁷: Es un framework para aplicaciones web de código abierto, enfocado en el desarrollo ágil de aplicaciones.

⁴¹ <http://www.joomla.org>

⁴² <http://moodle.org>

⁴³ <http://www.postgresql.org>

⁴⁴ <http://tomcat.apache.org>

⁴⁵ <http://es.wikipedia.org/wiki/Servlet>

⁴⁶ http://es.wikipedia.org/wiki/JavaServer_Pages

⁴⁷ <http://rubyonrails.org>



MySQL: Es un sistema de administración de bases de datos relacionales de código abierto ampliamente usado en desarrollo de aplicaciones web.



Drupal⁴⁸: Es un sistema de administración de contenidos y un framework de administración de código abierto escrito en PHP.



LAPP: Es un acrónimo de la integración de tecnologías, esta son: el sistema operativo linux, el servidor web apache, el motor de base de datos PostgreSQL y el lenguaje de programación interpretado PHP.



Apache Tomcat: Es un servidor web y un contenedor de servlets y JavaServer Pages de código abierto, se diferencia a la versión Tomcat en que integra como servidor web a Apache⁴⁹.



LAMP: Es un acrónimo de la integración de tecnologías, esta son: el sistema operativo linux, el servidor web apache, el motor de base de datos MySQL y el lenguaje de programación interpretado PHP.



Trac⁵⁰: Es un administrador de proyectos y una herramienta de seguimiento de incidencias basada en web. Integra los sistemas de control de versiones Subversion⁵¹, Git⁵², Mercurial⁵³, Bazaar⁵⁴, Perforce⁵⁵ y Darcs⁵⁶.

⁴⁸ <http://drupal.org>

⁴⁹ <http://httpd.apache.org>

⁵⁰ <http://trac.edgewall.org>

⁵¹ <http://subversion.apache.org>

⁵² <http://git-scm.com>

⁵³ <http://mercurial.selenic.com>

⁵⁴ <http://bazaar.canonical.com>

⁵⁵ <http://www.perforce.com>

En el sitio web de TurnKey Linux, se encuentran otras imágenes listas para ser usadas que se espera sean integradas con la plataforma más adelante; Además del uso de las Imágenes de TurnKey linux, se generaron imágenes para pruebas internas y de rendimiento, de éstas, una máquina base sobre la cual se pueden instalar servicios que no se encuentren dentro de la biblioteca anteriormente mencionada, fue dejada a disposición de los usuarios.

6.6. Balanceo de carga

Para una prestación del servicio estable, es necesario que los recursos de los equipos host sean utilizados de la mejor manera, para esto, OpenNebula cuenta con un módulo encargado de balancear la instanciación de las máquinas, que debe ser configurado de acuerdo a las políticas que consideren adecuadas quienes implementan la plataforma.

Las políticas predefinidas en el sistema son:

Packing Policy: Con esta política se busca minimizar el número de hosts en uso; con ello se logra reducir la fragmentación de máquinas virtuales al enviar con mayor prioridad las máquinas recién instanciadas hacia los nodos donde se estén ejecutando la mayor cantidad de éstas.

Striping Policy: Con esta se busca maximizar los recursos disponibles para las máquinas virtuales en los hosts, distribuyendo las máquinas recién instanciadas con mayor prioridad hacia aquellos hosts donde se estén ejecutando la menor cantidad de estas.

⁵⁶ <http://darcs.net>

Load-aware Policy: Esta política busca maximizar los recursos disponibles en los host para las máquinas virtuales enviando las máquinas recién instanciadas hacia los nodos con menor carga de CPU.

Además de las políticas predefinidas es posible establecer nuevas políticas que dependan de otras variables del sistema; esta función de personalización del sistema debe ser tomada en cuenta para la expansión de la plataforma, ya que la política que se define en este punto del proyecto puede no ser adecuada al contar con hosts de diferentes características. Para el presente proyecto se seleccionó como política de balanceo de carga a *Load-aware Policy*, ya que como se observó en el capítulo de pruebas del sistema, los equipos con que se cuenta tienen mayor limitación en número de máquinas instanciables debido a la cantidad de CPU que debido a la cantidad de memoria RAM.

6.7. Cuotas

Otra de las limitaciones de prestación del servicio hacen parte del sistema de cuotas de usuario, con estas se determina qué cantidad de recursos puede utilizar un usuario determinado, es de suma importancia controlar la manera en la cual se asignan estos recursos, ya que sin la implementación de este tipo de restricciones, un usuario puede aprovisionarse de todos los recursos y evitar de esta manera la disposición de estos para todos los usuarios de la EISI.

Los parámetros que pueden ser establecidos para la limitación del servicio por cuotas en OpenNebula son cantidad de CPU, cantidad de RAM, número de máquinas instanciadas y almacenamiento en disco; Esta limitación puede ser global, o por usuario. Las cantidades asignadas en cada límite son acumulativas, es decir, un usuario limitado solo por 1 GB de RAM podría tener 4 instancias de 256 MB, 2 de 512 MB ó 1 de 768 MB y 1 de 256 MB etc.

Para el presente proyecto, se determinó que los usuarios de acceso abierto, pueden hacer uso de las plantillas establecidas, limitados a la ejecución de una sola instancia, esta limitación no aplica a usuarios con proyectos que pueden precisar de servicios en diferentes instancias, por lo tanto no es necesaria una configuración de cuotas global.

6.8. Políticas de Uso

Un tema importante en la prestación de servicios en la nube son los aspectos legales que lo rodean, aunque el presente proyecto sea enfocado a un entorno académico, los aspectos de privacidad, seguridad, confidencialidad y permanencia de los datos siguen siendo un tema que no puede dejar de ser analizado.

En Colombia la ley 1273 de 2009⁵⁷ o ley de delitos informáticos, creó un marco legal sobre el cual existe la protección de la información y de los datos, esta ley penaliza atentados contra la confidencialidad, integridad y disponibilidad de los datos y de los sistemas informáticos. En sus artículos define los tipos de comportamientos sancionables como: acceso abusivo a un sistema informático, obstaculización de redes, interceptación de datos, daños informáticos, uso de software malicioso, violación de datos personales y suplantación de sitios web para capturar datos personales; con penas que van desde los 36 hasta los 96 meses de prisión y con multas de hasta 1500 salarios mínimos legales vigentes.

En un entorno como el que representa el la computación en la nube, en la cual existen usuarios compartiendo recursos de hardware, es una prioridad establecer las normas que regulen el cumplimiento de una serie de políticas de uso de estos recursos, así como también es una obligación del prestador del servicio informar a

⁵⁷ http://www.secretariasenado.gov.co/senado/basedoc/ley/2009/ley_1273_2009.html

los usuarios acerca de la normatividad vigente y la aplicación de dichas leyes sobre el servicio prestado.

La ley 1266 de 2008⁵⁸, o ley de hábeas data, establece el derecho que tiene cualquier ciudadano a conocer, actualizar y rectificar informaciones que se tengan sobre estos en bancos de datos y regula los derechos, libertades y garantías de recolección, tratamiento y circulación restringida de datos personales y temporalidad de los datos.

En un ambiente como el de la computación en la nube, en el cual la migración del servicio es un proceso que es transparente para el usuario, es necesario cumplir con este tipo de reglamentación, para garantizar la confidencialidad de los datos almacenados, y establecer condiciones que permitan a los usuarios conocer el estado y ubicación física de su información.

Otras leyes que establecen marcos para el desarrollo de servicios asociados a la computación en la nube son la Ley 1341 de 2009 o Ley de TICs, la Ley 1221 de 2008 o Ley de Teletrabajo y la recién aprobada Ley 201 de 2012 o Ley Lleras.

Según las regulaciones existentes en nuestro país para el desarrollo de un servicio como el que se busca prestar, es necesario dejar completamente definidos los límites de la prestación del servicio, los deberes como usuario de cada uno de aquellos que se beneficiarán con la plataforma y el marco de protección de los datos almacenados; con este fin se generó un documento, anexo A, en el cual se determina la responsabilidad de GID-CONUSS como prestador del servicio, hasta que punto llega esa responsabilidad y los términos de uso de la plataforma.

⁵⁸ http://www.secretariasenado.gov.co/senado/basedoc/ley/2008/ley_1266_2008.html

6.9. Prestación de servicios en la nube

Como un objetivo de este proyecto se planteó la simulación de alguno de los escenarios de prestación de servicio para comprobar el comportamiento de la plataforma frente a la carga de uso generada por los usuarios; para el beneficio del presente proyecto, se contó con la colaboración de dos proyectos académicos que confiaron en las posibilidades del servicio ofrecido, estos proyectos permitieron obtener información más valiosa que la se podría haber obtenido por medio de una simulación.

6.9.1 Proyecto SIPCE

SIPCE⁵⁹ versión 3, es el Sistema de Información de las Prácticas Clínico Comunitarias de Enfermería, la descripción de SIPCE proporcionada cuando el grupo de investigación interesado solicitó el servicio es la siguiente:

“Es un sistema de información que recoge datos de las prácticas que realizan los estudiantes de la Escuela de Enfermería, aplicando los estándares de enfermería NIC, NOC Y NANDA. Nació como un proyecto de pregrado en 2001 de estudiantes de la Escuela de Ingeniería de Sistemas pertenecientes al Grupo de Investigación de Ingeniería Biomédica (GIIB) de la Universidad Industrial de Santander y con la ayuda conjunta de la Escuela de Enfermería (UIS), con el fin de solucionar problemáticas que se presentaban debido a que los registros de enfermería carecían de Importancia, formatos normalizados, presenta distintos sistemas de clasificación, poca validación y evaluación, ausencia en tecnologías de información”.

El servicio prestado a la plataforma SIPCE, es el de un servidor web con una configuración ajustada a la aplicación web que han desarrollado, se trata de una instancia LAMP tipo small; los administradores de la plataforma SIPCE cuentan con el control absoluto sobre la instancia virtual asignada, este tipo de libertad de ad-

⁵⁹ <http://cloudeisi.uis.edu.co/sipce/>

ministración no era proporcionada por el prestador de servicio de hosting con el que contaban antes de la migración.

Al proyecto se le prestó inicialmente colaboración en el proceso de migración, el ajuste de la configuración del servidor y la supervisión del estado del mismo durante el primer mes, periodo durante el cual se observó el comportamiento de la prestación del servicio. Luego de ese periodo, el control total de la plataforma pasó a manos de los administradores de la plataforma SIPCE.

El servicio fue activado a finales del mes de febrero de 2012 y a partir de ese momento se vio la necesidad de medir el nivel de uso de la instancia y el tráfico de datos generado, esto con la finalidad de establecer políticas de uso para los usuarios futuros de los servicios desarrollados a través de este proyecto.

Luego de 65 días de ejecución sin interrupción de la instancia de SIPCE, se obtuvieron los datos presentes en la Tabla 9.

Tabla 9: Uso del servidor web de SIPCE durante Marzo-Abril de 2012

Estadísticas mensuales de SIPCE	Marzo de 2012		Abril de 2012	
Total Accesos	29856		49026	
Total Archivos	18614		30396	
Total Páginas	9894		15626	
Total Visitas	388		571	
Total kB Files	743871		1731686	
Total Clientes	244		433	
Total URLs	563		231	
Total Enlaces origen	146		97	
Total Navegadores	77		112	
	Media	Max	Media	Max
Accesos por Hora	40	1220	68	4574
Accesos por Día	963	2516	1634	11926
Archivos por Día	600	1730	1013	7834
Páginas por Día	319	1082	520	4394
Visitas por Día	12	52	19	93
kB Files per Day	23996	81246	57723	513833

Fuente: Autor.

La Tabla 9 resume los totales del uso del sitio web de la siguiente manera:

Accesos: Cualquier tipo de petición al servidor, pueden ser páginas web, archivos de imágenes, de audio, scripts etc.

Archivos: Son las solicitudes al servidor que devuelven algún tipo de contenido, la relación con los accesos puede ser tomada como solicitudes entrantes y solicitudes salientes.

Páginas: Cualquier documento HTML o que genere un documento HTML es considerado una página.

Visitas: Cada solicitud hecha al servidor desde un sitio único referenciado por la dirección IP, ocurrida entre un lapso mayor a 30 minutos. Si la solicitud ocurre luego de este periodo es considerado una nueva visita.

kB Files: Es la cantidad de datos en KB que salieron del servidor durante el periodo analizado.

Clientes: Número de diferentes direcciones IP desde las cuales se ha accedido al servidor.

URL: Es el número de diferentes direcciones relativas al sitio generadas por el servidor.

Enlaces Origen: Muestra la cantidad de sitios que redirigieron al servidor web, estos pueden ser resultados de motores de búsquedas, vínculos incrustados en otros sitios web, o cualquier acceso a una URL del servidor no escrita directamente en el navegador.

Navegadores: Número de diferentes combinaciones de navegadores/sistemas operativos usados para acceder al servidor web.

Los accesos por hora y día presentan los promedios y los máximos de cada uno de estos datos.

Los datos presentes en la Tabla 9 presentan el uso del servidor web asignado a la plataforma SIPCE de los meses de marzo a abril, durante todos los días de estos

meses; como se mencionó anteriormente, el servicio se inició desde finales de febrero, pero los datos de dicho mes no contenían datos suficientes para realizar una comparación completa, por esta razón no se tuvieron en cuenta para el presente análisis.

Se puede observar que el tráfico del sitio (Total kB files) durante el mes de marzo fue de 743.871 KB y durante el mes de abril fue de 1.731.686 KB, estos datos contrastan con los suministrados por los administradores de SIPCE cuando solicitaron el servicio, en estos adjuntaron los datos del servicio de hosting con el que contaban en el momento, el cual limitaba el tráfico mensual del servicio a 1 GB, con los datos aquí recolectados se pudo determinar que el tráfico en el mes de abril superó en un 73.1% el servicio que tenían contratado, tráfico que en el presente proyecto no representó ningún tipo de limitación.

El resultado de la anterior experiencia permitió determinar algunas de las fronteras entre la prestación del servicio de computación en la nube y la colaboración como grupo CONUSS en el proceso de administración de servidores, además permitió demostrar que la estabilidad del servicio para una aplicación en producción es aceptable.

6.9.2 Curso de programación en la web

Se realizó otra prueba de producción, en este escenario, se creó una instancia para un curso de programación en la web de la EISI, la motivación de esta experiencia fue establecer un marco para la presentación de los trabajos finales de los estudiantes, en el cual se pudiesen evaluar sus proyectos en igualdad de condiciones y ellos a su vez tuvieran la experiencia de trabajar con un servidor que no se encontrara alojado en sus equipos personales. Los datos obtenidos durante la prestación del servicio, se presentan en la Tabla 10.

Tabla 10: Uso del servidor de Programación en la web Marzo-Abril de 2012

Estadísticas mensuales de 22967	Marzo de 2012		Abril de 2012	
Total Accesos	6629		172279	
Total Archivos	2909		75908	
Total Páginas	2250		55183	
Total Visitas	29		394	
Total kB Files	51678		1205754	
Total Clientes	1		1	
Total URLs	227		1911	
Total Enlaces origen	41		569	
Total Navegadores	16		93	
	Media	Max	Media	Max
Accesos por Hora	39	644	239	6824
Accesos por Día	947	2575	5742	32946
Archivos por Día	415	1100	2530	13650
Páginas por Día	321	866	1839	10245
Visitas por Día	4	6	13	84
kB Files per Day	7383	24181	40192	261267

Fuente: Autor.

La Tabla 10 resume los totales del uso del sitio web, la explicación de los campos de la tabla se puede observar nuevamente en la página 77.

Esta instancia a diferencia de la asignada a SIPCE, no corresponde a un escenario de únicamente uso de una aplicación, se puede considerar como un servidor de desarrollo, esto se puede apreciar en la cantidad de acceso al servidor en el mes de abril, el servicio fue prestado a dos cursos simultáneamente, con 11 subgrupos de 4 estudiantes, es decir, el sistema soportó 44 estudiantes concurrentemente realizando desarrollo web sin ningún contratiempo. Otro tipo de métrica que se recopiló, fue el tráfico de la interfaz de red, se presenta gráficamente en la Figura 18 y en la Tabla 11, esto se debe a que tráfico del servidor web no incluye el generado por las cargas y descargas de archivos que se producen durante el proceso de desarrollo.

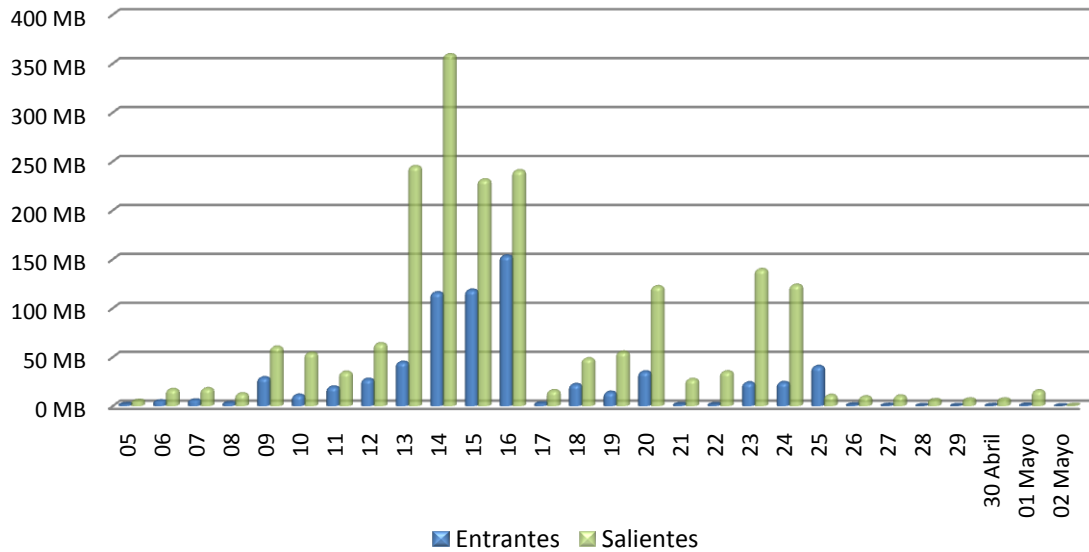


Figura 18: Tráfico de red de la instancia de Programación Web

Fuente: Autor.

Tabla 11: Tráfico de red de la instancia de Programación Web

Fecha	Entrantes	Salientes	Total
05 Abril	2,09 MB	5,87 MB	7,96 MB
06 Abril	5,28 MB	17,19 MB	22,47 MB
07 Abril	6,22 MB	17,99 MB	24,21 MB
08 Abril	3,24 MB	12,73 MB	15,96 MB
09 Abril	29,14 MB	60,38 MB	89,53 MB
10 Abril	11,32 MB	54,29 MB	65,61 MB
11 Abril	19,67 MB	34,81 MB	54,47 MB
12 Abril	27,47 MB	63,88 MB	91,35 MB
13 Abril	44,93 MB	244,93 MB	289,86 MB
14 Abril	115,91 MB	359,14 MB	475,06 MB
15 Abril	118,60 MB	231,41 MB	350,01 MB
16 Abril	153,48 MB	240,71 MB	394,19 MB
17 Abril	2,91 MB	15,90 MB	18,80 MB
18 Abril	22,16 MB	48,55 MB	70,71 MB
19 Abril	14,27 MB	55,35 MB	69,62 MB
20 Abril	35,13 MB	122,25 MB	157,38 MB
21 Abril	1,89 MB	27,46 MB	29,36 MB
22 Abril	1,86 MB	35,19 MB	37,04 MB
23 Abril	23,91 MB	139,72 MB	163,63 MB
24 Abril	24,22 MB	123,73 MB	147,95 MB
25 Abril	40,63 MB	11,14 MB	51,77 MB
26 Abril	1,57 MB	9,60 MB	11,17 MB
27 Abril	0,987 MB	10,53 MB	11,50 MB

Fecha	Entrantes	Salientes	Total
28 Abril	0,438 MB	6,72 MB	7,15 MB
29 Abril	0,435 MB	7,57 MB	8,00 MB
30 Abril	0,667 MB	7,59 MB	8,25 MB
01 Mayo	1,22 MB	15,82 MB	17,04 MB
02 Mayo	0,055 MB	0,934 MB	0,989 MB
Total	709,70 MB	1981,38 MB	2691,04 MB

Fuente: Autor.

Los datos obtenidos por el servidor web mostraron que el uso de red durante el mes de abril fue de 1206 MB, mientras la interfaz de red, en la Tabla 11 muestra que el tráfico saliente fue de 1981 MB; por lo anterior se observa la importancia de el uso de herramientas adecuadas para el seguimiento del tráfico de red.

Los factores de rendimiento del sistema que también se buscaron medir, fueron los relacionados con el consumo de CPU y de memoria RAM, la instancia asignada fue una tipo small, por lo tanto contó con una CPU de 0,25 núcleos y 256 MB de memoria RAM, por esta razón se supervisó que en ningún momento la instancia fuera a fallar por falta de capacidad de cómputo o de la misma memoria RAM. En la Figura 19 y la Figura 20 se observan los datos obtenidos por medio de la herramienta ganglia.

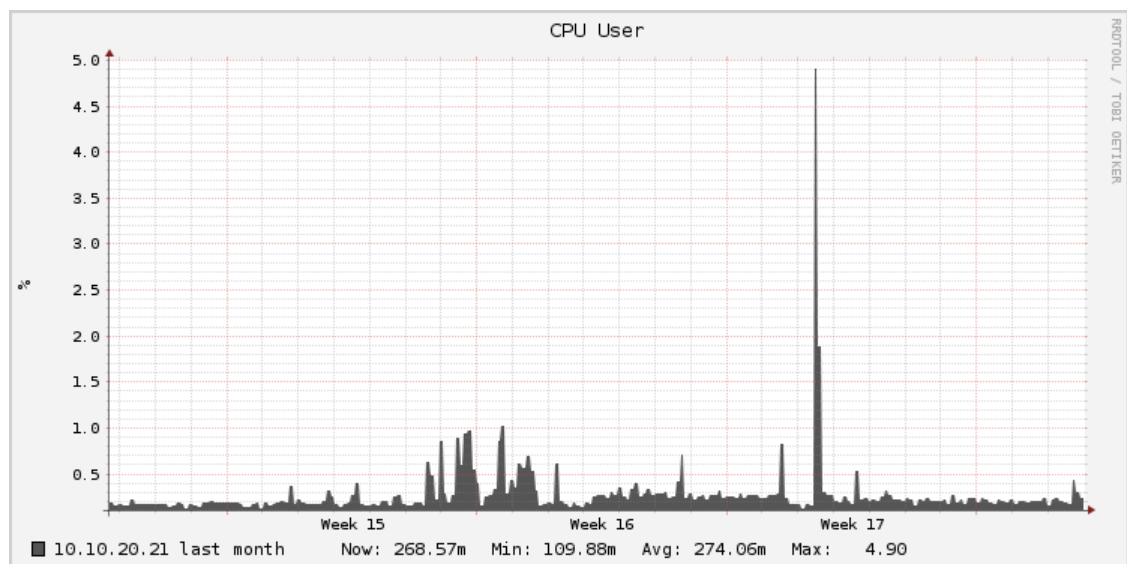


Figura 19: Uso de CPU por la instancia de Programación Web

Fuente: Autor.

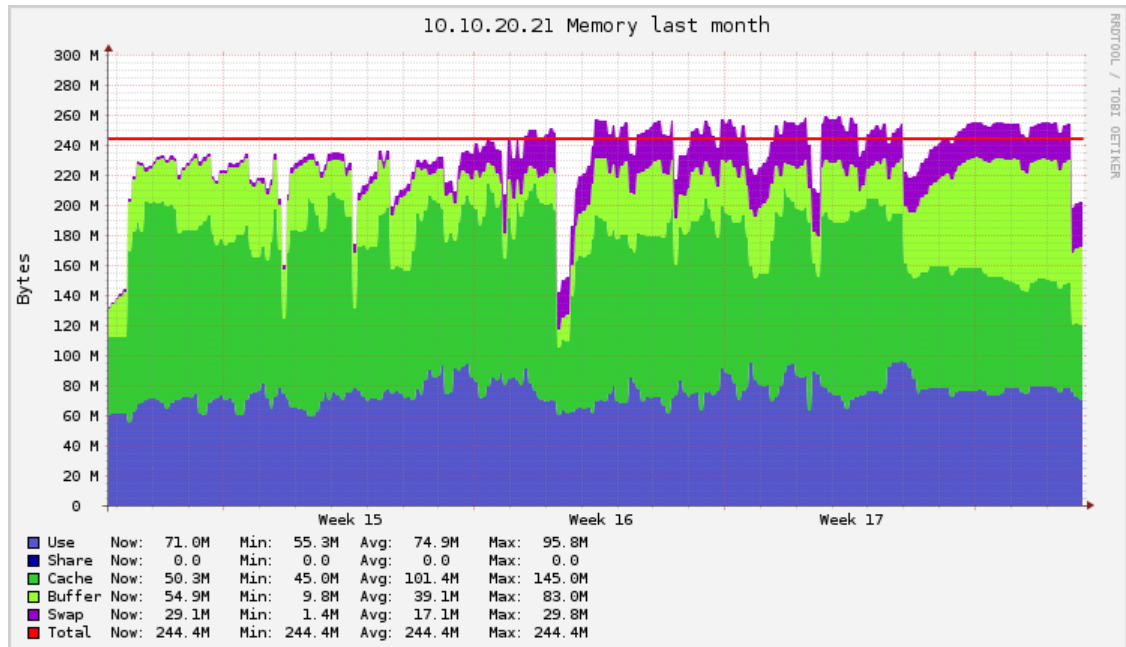


Figura 20: Uso de la memoria RAM de la instancia de Programación Web
Fuente: Autor.

En la Figura 19 se puede observar según el reporte que el uso máximo de la CPU fue de 4.9% en el cual se destaca el uso entre la semana 15 y 17, periodo de entrega de trabajos finales de parte de los estudiantes; así mismo el reporte de la Figura 20 indica que la memoria RAM en uso tuvo un máximo de 95.8 MB, lo cual demuestra que los parámetros de la instancia asignada fueron suficiente para dar soporte a los proyectos de curso.

6.10. Documentación administrativa

Como parte del proceso de planificación, construcción y administración del presente proyecto se elaboró un manual administrativo que describe los procesos de implementación de la plataforma, creación de nuevas máquinas virtuales, imágenes, redes, usuarios y grupos; así como lo relacionado a la implementación de medidas de seguridad, los controles de acceso de usuario e información relacionada al fun-

cionamiento de la plataforma. Por las razones expuestas en el numeral 6.8, Políticas de Uso, y las limitaciones del acuerdo de uso, la mayoría de la documentación es de carácter confidencial, razón por la cual se encuentra a disposición de los miembros del equipo de administración del grupo CONUSS y de la plataforma cloudEISI y bajo la responsabilidad del docente encargado de la dirección del grupo.

7. CONCLUSIONES

El desarrollo de una plataforma de prestación de servicios de computación en la nube para comunidades académicas usando soluciones de software libre o abierto, representa una gran oportunidad para el aprovechamiento eficiente de los recursos subutilizados con los que estas cuentan, sin costos de inversión, o de licenciamiento.

El establecimiento de políticas de uso para un servicio de computación en la nube, basada en la legislación vigente, se hace necesario frente a las diferentes eventualidades de uso del servicio por parte de los usuarios y constituye un punto clave para el desarrollo de futuros servicios como apoyo a proyectos académicos, de una manera segura y dentro del marco legal de la nación y la institución.

La selección de un hipervisor como pieza fundamental del desarrollo de servicios en los cuales se busque un uso eficiente de recursos, establece los límites y el alcance de la funcionalidad de éstos respecto al tipo de sistemas operativos, interfaces de red y características especiales de la prestación del servicio.

El uso de un modelo de Infraestructura como servicio permitió apoyar el desarrollo de proyectos de grupos de investigación y de cursos ofrecidos por la Universidad, utilizando únicamente la capacidad de cómputo que requirieron y sin perjudicar el desempeño del servicio prestado.

A partir de las pruebas realizadas se comprobó que la utilización eficiente de los recursos de cómputo con que cuenta la EISI, permite además de conseguir prestar servicios a un mayor número de miembros de la comunidad, la posibilidad hacerlo con un menor costo energético, causando así una menor huella de carbono en el ambiente.

El cálculo de la carga máxima soportada por el hardware disponible no solo permite determinar la cantidad de servicios que pueden ser prestados, también puede ser tomado como punto de partida para establecer el tipo de algoritmo de balanceo de carga y el valor de las cuotas de usuario que deben ser implementadas en un modelo de infraestructura como servicio.

La generación de documentación administrativa y de apoyo a los administradores futuros de un proyecto académico como este, pone en evidencia la necesidad de establecer este tipo de actividades como una práctica necesaria para asegurar la continuidad y mantenibilidad de este tipo de proyectos a largo plazo.

Este paradigma de computación, representa un cambio en la forma en la cual pueden realizarse las inversiones de hardware de una institución académica; los costos cada vez más altos de centros de cómputo subutilizados, representados en gastos de consumo de energía eléctrica, climatización, adecuación de instalaciones y seguridad física de los equipos, convierten este tipo de proyectos en soluciones de bajo costo, para el uso eficiente de recursos computacionales.

Las labores de gestión tecnológica asociadas a la construcción y administración de modelos de prestación de servicios en la nube, requieren de un alto nivel de dedicación laboral y de revisión constante de literatura reciente, al ser éste un modelo en constante evolución y por encontrarse en su etapa inicial.

Con el trabajo realizado en el presente proyecto se genera una nueva rama de investigación y desarrollo en la EISI en tecnologías de computación en la nube, seguridad y servicios, liderado por el grupo de investigación CONUSS.

8. RECOMENDACIONES

Para la continuación del presente proyecto, se recomienda la implementación y pruebas de otros tipos de hipervisores como son Xen y KVM, preservando la filosofía del presente proyecto en cuanto al uso de tecnologías libres o de código abierto.

Buscar de financiación para la integración de nuevo hardware que permita el apoyo a proyectos de requerimientos más altos, esto puede incentivar la inversión en este modelo de prestación de servicios por parte de otras unidades académicas.

Instalar un nodo de almacenamiento en red que permita un sistema de alta disponibilidad, para facilitar las labores administrativas de la plataforma.

Implementar un sistema de monitoreo que permita a los usuarios conocer en cualquier momento la cantidad de recursos consumidos y la creación de acuerdos de nivel de uso de servicio con éstos.

La creación de un entorno integrado para desarrollo de aplicaciones web, puede disminuir el tiempo de adopción de la plataforma por parte de los grupos especializados en desarrollo de la EISI.

Se recomienda que el desarrollo de la próxima versión de la aplicación web de cursos virtuales Meiweb se realice sobre la plataforma cloudEISI, utilizando las ventajas que ofrece la misma.

La realización de talleres de formación que permitan a los diferentes grupos de investigación y desarrollo construir repositorios de los proyectos realizados por

estudiantes, puede convertir uno de los escenarios planteados por el presente proyecto en una herramienta de apoyo para los proyectos futuros.

Finalmente se recomienda a la EISI generar espacios para la formación de estudiantes en el uso de estas tecnologías, para el beneficio de la comunidad objetivo del presente proyecto.

BIBLIOGRAFÍA

[1] Gartner, Inc.; "Gartner Says Cloud Computing Will Be as Influential as E-business." Consultado el 10 de diciembre de 2011.

<http://www.gartner.com/it/page.jsp?id=707508>.

[2] Maggiani, R.; "Cloud computing is changing how we communicate," Professional Communication Conference, 2009. IPCC 2009. IEEE International , vol., no., pp.1-4, 19-22 July 2009 Consultado el 10 de Diciembre de 2011.

<http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=5208703&isnumber=5208666>

[3] Forrester Research, Inc. Is Cloud Computing Ready For The Enterprise? Consultado el 10 de Diciembre de 2011.

<http://www.forrester.com/Research/Document/Excerpt/0,7211,44229,00.html>.

[4] Jeremy Geelan. Twenty one experts define cloud computing.

Virtualization, August 2008. Consultado el 10 de Diciembre de 2011.

<http://virtualization.sys-con.com/node/612375>.

[5] Goldberg, Robert P. (February 1973) (PDF). Architectural Principles for Virtual Computer Systems. Harvard University. pp. 22–26. Consultado el 10 de diciembre de 2011. <http://www.dtic.mil/cgi-bin/GetTRDoc?AD=AD772809&Location=U2&doc=GetTRDoc.pdf>

[6] Mell, P. & Grance, T., 2011. The NIST Definition of Cloud Computing. NIST Special Publication 800-145. Consultado el 10 de Diciembre de 2011.

<http://csrc.nist.gov/publications/nistpubs/800-145/SP800-145.pdf>

[7] Luis M. Vaquero, Luis Rodero-Merino, Juan Caceres, and Maik Lindner. 2008. A break in the clouds: towards a cloud definition. SIGCOMM Comput. Commun. Rev. 39, 1 (December 2008), 50-55. Consultado el 10 de Diciembre de 2011. <http://doi.acm.org/10.1145/1496091.1496100>

[8] A Comparison and Critique of Eucalyptus, OpenNebula and Nimbus, Peter Sempolinski and Douglas, Thain University of Notre Dame.
<http://www.cse.nd.edu/~ccl/research/papers/psempoli-cloudcom.pdf>

[9] Hybrid Clouds: Comparing Cloud Toolkits. Seminar Paper. Chair in Information Systems III. Prof. Dr. Martin Schader. University of Mannheim. Abril, 2010

[10] González-Madroño Abad, M^a Pilar y López Collado, Diana y Viudez Corroto, Raúl (2010) An Open Cloud Computing Interface (OCCI) management console for the OpenNebula Toolkit. [Trabajo de curso] (No publicado) Disponible en <http://eprints.ucm.es/11432/>

OTRAS REFERENCIAS

Dave Durkee. Why Cloud Computing Will Never Be Free. Queue 8, 4, Página 20, <http://doi.acm.org/10.1145/1755884.1772130>

Faouzi Kamoun. 2009. Virtualizing the Datacenter Without Compromising Server Performance. Ubiquity 2009, Agosto, <http://doi.acm.org/10.1145/1595422.1595424>

Kemal A. Delic and Martin Anthony Walker. 2008. Emergence of the Academic Computing Clouds. Ubiquity 2008, Agosto, <http://doi.acm.org/10.1145/1414663.1414664>

George Reese. 2009. Cloud Application Architectures: Building Applications and Infrastructure in the Cloud. O'Reilly Media, Inc..

Jason Buffington. 2010. Data Protection for Virtual Data Centers (1era ed.). SYBEX Inc., Alameda, CA, USA.

Nelson Ruest and Danielle Ruest. 2009. Virtualization, a Beginner's Guide (1era ed.). McGraw-Hill, Inc., New York, NY, USA.

Nikolaus Huber, Marcel Von Quast, Fabian Brosig, and Samuel Kounev. 2010. Analysis of the performance-influencing factors of virtualization platforms. In Proceedings of the 2010 international conference on On the move to meaningful internet systems: Part II (OTM'10), Robert Meersman, Tharam Dillon, and Pilar Herrero (Eds.). Springer-Verlag, Berlin, Heidelberg, 811-828.

Vassilka Tchifilionova. 2010. Security and privacy implications of cloud computing: lost in the cloud. In Proceedings of the 2010 IFIP WG 11.4 international conference on Open research problems in network security (iNetSec'10), Jan Camenisch, Maria Dubovitskaya, and Valentin Kisimov (Eds.). Springer-Verlag, Berlin, Heidelberg, Páginas 149-158.

U. Gurav and R. Shaikh. 2010. Virtualization: a key feature of cloud computing. In Proceedings of the International Conference and Workshop on Emerging Trends in Technology (ICWET '10). ACM, New York, NY, USA, Páginas 227-229. <http://doi.acm.org/10.1145/1741906.1741957>

RECURSOS WEB

Andrei, T. 2009. Cloud Computing Challenges and Related Security. Consultado el 28 de Abril de 2011, de <http://www1.cse.wustl.edu/~jain/cse571-09/ftp/cloud.pdf>

Bime Analytics. 2010. Analysis: Gartner's Emerging Technology Hype Cycle 2010: What's Hot and What's Not. Consultado el 28 de Abril de 2011, de <http://bimeanalytics.com/blog/analysis-gartners-emerging-technology-hype-cycle-2010-hot/>

Bime Analytcs. 2009. Gartner's Hype Cycle 2009: SAAS Business Intelligence is on the rise. Consultado el 22 de Abril de 2011, de <http://bimeanalytics.com/blog/gartners-hype-cycle-2009-saas-business-intelligence-rise/>

Chandrasekaran, A. 2010. Cloud Computing - FAQ. Consultado el 28 de Abril de 2011, de <http://www.frost.com/prod/servlet/market-insight-top.pag?Src=RSS&docid=207327187>

Cloud Security Alliance. 2009. Security Guidance for Critical Areas of Focus in Cloud Computing. Consultado el 10 de Mayo de 2011, de <https://cloudsecurityalliance.org/csaguide.pdf>

Information Systems Audit and Control Association. 2011. Essential characteristics of Cloud Computing. Consultado el 15 de Mayo de 2011, de <http://www.isaca.org/Groups/Professional-English/cloud-computing/GroupDocuments/Essential%20characteristics%20of%20Cloud%20Computing.pdf>

Laird, P. 2009. Cloud Computing Taxonomy at Interop Las Vegas. Consultado el 28 de Abril de 2011, de <http://peterlaird.blogspot.com/2009/05/cloud-computing-taxonomy-at-interop-las.html>

Laird, P. 2009. Understanding the Cloud Computing/SaaS/PaaS markets: a Map of the Players in the Industry. Consultado el 28 de Abril de 2011, de <http://peterlaird.blogspot.com/2008/05/understanding-cloud-computingsaaspaas.html>

Observatorio Regional de Sociedad de la Información. 2010. La Tecnología como Servicio. Consultado el 21 de Abril de 2011, de Observatorio da Sociedade da Información e da Modernización de Galicia: http://www.osimga.org/gl/documentos/d/2010_12_29_ORSI_estudio_cloud_computing.pdf

Open Security Architecture. 2010. 08.02 Security Architecture Patterns. Consultado el 28 de Abril de 2011, de <http://www.opensecurityarchitecture.org/cms/library/patternlandscape>

Research Councils. 2010. Cloud Computing for Research. Consultado el 28 de Abril de 2011, de <http://www.rcuk.ac.uk/documents/research/esci/CloudWorkshop-July2010.pdf>

Youseff, L., Butrico, M., & Da Silva, D. 2010. Toward a Unified Ontology of Cloud Computing. Consultado el 3 de Abril de 2011, de <http://freedomhui.com/wp-content/uploads/2010/03/CloudOntology.pdf>

ANEXOS

Anexo A: Términos y Condiciones de Uso

[2012/05/01]

Este acuerdo establece los términos y condiciones de uso aplicables para usted (el "**USUARIO**") en su uso de los servicios prestados (los "**SERVICIOS**") por el Grupo de Investigación y Desarrollo en Computación en la Nube, Seguridad, Servidores y Servicios ("**GID-CONUSS**") para la plataforma de infraestructura como servicio (IaaS) de computación en la nube cloudEISI.

Al usar hacer uso de los **SERVICIOS**, el **USUARIO** automáticamente acepta estos Términos y Condiciones de Uso y acuerda registrarse por ellos.

1. Usuarios elegibles. Los servicios están abiertos para el uso de cualquier miembro de la comunidad académica de la Escuela de Ingeniería de Sistemas e Informática de la Universidad Industrial de Santander.

2. Cuenta de usuario, contraseña y seguridad. Como parte de la creación de una cuenta de usuario se otorgará un nombre de cuenta y una contraseña. El **USUARIO** es completamente responsable por mantener la confidencialidad de su cuenta o contraseña. Si el **USUARIO** tiene conocimiento de que la privacidad de su contraseña ha quedado expuesta, debe notificar inmediatamente a **GID-CONUSS** e informar acerca de cualquier fallo de seguridad.

3. Contenidos del usuario. El **USUARIO** acepta usar los **SERVICIOS** sólo para fines legales y queda prohibida la publicación, distribución y alojamiento en los **SERVICIOS** de: (a) cualquier material ilegal, perjudicial, amenazante, abusivo, acosador, difamatorio u obsceno de cualquier tipo, incluyendo, entre otros, cualquier material que fomente conductas que constituirían una ofensa penal, que

produjera responsabilidad civil o que de otro modo infrinja cualquier ley aplicable local, estatal, nacional o internacional. (b) del cual el **USUARIO** no tenga el derecho de poner a disposición conforme a ninguna ley o relación contractual; o (c) infrinja cualquier derecho de patente, marca comercial, secreto de mercado, derecho de autor u otros derechos de propiedad de cualquier parte. El **USUARIO** acepta no transmitir, subir, publicar, enviar por email o poner a disposición de otro modo virus informáticos, publicidad no solicitada o no autorizada, material promocional o de venta, lo que incluye cadenas de correspondencia, correos masivos o cualquier otra forma de "spam". El **USUARIO** acepta NO (i) falsificar encabezados o manipular de otro modo los identificadores de archivos con el objeto de ocultar el origen de cualquier contenido del **USUARIO**; (ii) recopilar o almacenar información de identificación personal acerca de otros usuarios sin la debida autorización y consentimiento de estos; (iii) utilizar herramientas de interceptación de datos; (iv) hacer uso de software malicioso o de obstaculización de redes o, (v) realizar accesos abusivos a los equipos informáticos que permiten la prestación de los **SERVICIOS** o la prestación los **SERVICIOS** de los demás usuarios. El **USUARIO** reconoce que **GID-CONUSS** tiene el derecho (pero no la obligación), de eliminar el contenido del **USUARIO** que viole estos Términos y Condiciones, y nos reservamos el derecho de denegar el servicio y/o cancelar cuentas sin previo aviso para cualquier usuario que viole estos Términos y Condiciones o infrinja los derechos de terceros.

4. Seguridad de los servicios. La seguridad de las instancias asignadas por parte de los **SERVICIOS** prestados recaen bajo la administración del **USUARIO**, así como el manejo de las copias de seguridad de los datos, código fuente, aplicaciones y bases de datos que hagan parte de sus contenidos, **GID-CONUSS** no se responsabiliza en casos de falla total o parcial de los servidores y que imposibiliten la recuperación de los contenidos por los sistemas de almacenamiento implementados.

5. Quejas por derecho de autor. **GID-CONUSS** respeta la propiedad intelectual de terceros. Si usted cree que su trabajo ha sido copiado en este sitio de manera que constituya una infracción a sus derechos de autor, puede escribir a cloudeisi@gmail.com para notificarnos acerca de la situación.

6. Descargo de responsabilidad. El **USUARIO** reconoce y acepta que utiliza este servicio sin cargo alguno y que por lo tanto este uso se ofrece “tal como está”, **GID-CONUSS** expresamente niega toda garantía de cualquier tipo, ya sea expresa o implícita. Como **USUARIO** acepta de forma expresa que todo uso de la información disponible en las los **SERVICIOS** asignados son para su uso personal o académico y que dicha información se utilizará bajo su propio riesgo y responsabilidad. **GID-CONUSS** expresamente niega toda responsabilidad por errores u omisiones en la información, los materiales y las funciones que se incluyen en nuestros servidores o en los sitios de terceros enlazados desde o hacia nuestros servicios por parte del **USUARIO**.

7. Indemnización. El **USUARIO** acepta indemnizar y liberar de toda responsabilidad a **GID-CONUSS** sus directivos, desarrolladores, investigadores, y administradores en cuanto a cualquier reclamación o demanda, lo que incluye los honorarios razonables de representación legal, entablada por terceros debida a, o que resulte de, el uso por parte del **USUARIO** de los **SERVICIOS** prestado.

8. Potestad jurídica. Los **SERVICIOS** son administrados por **GID-CONUSS** desde sus oficinas dentro de la República de Colombia. Al hacer uso de los **SERVICIOS**, tanto el **USUARIO** como **GID-CONUSS** aceptan que todos los asuntos relacionados con su uso de los **SERVICIOS** estarán regidos e interpretados de acuerdo con la legislación de la República de Colombia, sin considerar los principios de conflictos de leyes, como si se realizaran enteramente dentro de la República de Colombia.

9. Modificación de los términos de uso del servicio. **GID-CONUSS** puede cambiar los términos y condiciones de los Términos de Uso del Servicio. **GID-CONUSS** hará las notificaciones de los cambios y publicará la versión más reciente en la sección Términos de Uso del sitio <http://cloudeisi.uis.edu.co>, si el **USUARIO** no está de acuerdo con las modificaciones puede dejar de usar los **SERVICIOS** informando a **GID-CONUSS**.

10. Confidencialidad de la Información. **GID-CONUSS** no compartirá ni revelará la información confidencial con terceros, excepto que tenga expresa autorización de parte del **USUARIO**, o cuando ha sido requerido por orden judicial o legal.

Anexo B: GID-CONUSS

El Grupo de Investigación y Desarrollo en Computación en la Nube, Seguridad, Servidores y Servicios (GID-CONUSS), surgió como respuesta a las diferentes áreas hacia las cuales el grupo de desarrollo MEIWEB enfocó algunos sus proyectos de grado a partir del año 2011.

El grupo MEIWEB, a partir de su creación en el año 2006 por el Magíster Ingeniero Manuel Guillermo Flórez Becerra, ha realizado el desarrollo de un entorno virtual de aprendizaje que en su constante evolución se ha convertido en una excelente herramienta los para docentes y estudiantes que hacen usos de sus servicios.

Este grupo, en el año 2010, gracias a la donación por parte de la fundación Raúl D. Ocazonez de dos servidores para el apoyo a proyectos en la Escuela de Ingeniería de Sistemas e Informática, decidió tomar la responsabilidad de desarrollar proyectos que hicieran usos de los equipos donados y a su vez, pudiesen prestar servicios a toda la comunidad EISI.

Para este fin se desarrollaron dos proyectos paralelos de investigación, un proyecto que desarrollara el modelo administrativo para los equipos e incluyera un prototipo de clúster de alta disponibilidad⁶⁰ y un proyecto de prestación de servicios de computación en la nube para la EISI⁶¹. Con el desarrollo de estos dos proyectos surgió la necesidad de unificar los nuevos campos de investigación de MEIWEB con la rama existente de desarrollo, es ahí donde nace GID-CONUSS.

⁶⁰ Instalación, administración, configuración e implementación de servidores linux con énfasis en el desarrollo de un modelo administrativo y la creación de un prototipo de clúster de alta disponibilidad./ Barbosa A, Alexander. Muñoz D, Elkin. 2012

⁶¹ Modelo y prototipo de servicios de computación en la nube para estudiantes y profesores de la Escuela de Ingeniería de Sistemas e Informática de la Universidad Industrial De Santander / Díaz C, Emmanuell. 2012

GID-CONUSS representa la unificación de las ramas de investigación y acción del grupo, la descripción del acrónimo y de las ramas mencionadas es la siguiente:

GID-CONUSS: Grupo de Investigación y Desarrollo, para indicar que el grupo incluye una rama de investigación, sin abandonar su área de desarrollo.

GID-CONUSS: La rama de investigación y desarrollo en computación en la nube y virtualización.

GID-CONUSS: Seguridad, Servidores y Servicios. Las múltiples 'S' del acrónimo hacen referencia, a los diferentes campos de acción y a que no están limitados a estos. La rama de Seguridad, está encargada de definir estándares y políticas dentro de los servicios ofrecidos por el grupo. La rama de Servidores, es un área de apoyo a los diferentes procesos de administración de éstos, y presta también sus servicios a los usuarios de la plataforma de computación en la nube. La rama de Servicios, está enfocada a la prestación de servicios y desarrollo de aplicaciones y ésta, a su vez, es un área de trabajo de la rama de Computación en la Nube.

Como se puede observar, la vinculación y el trabajo en equipo de las diferentes áreas es lo que conforma a GID-CONUSS.

Como parte del proceso de unificación del grupo, se creó también un logotipo que permita identificar los diferentes productos y servicios a los cuales esté vinculado GID-CONUSS, en este, se abrevia el nombre, quedando el acrónimo CONUSS.



El logotipo representado por un caparazón, hace referencia lo delicada que es la prestación de servicios en los cuales los datos de los usuarios deben ser prioritarios, así como la importancia que tiene la protección y seguridad de los servicios para el grupo. Las líneas que conforman la silueta del caparazón, son múltiples líneas juntas simbolizando el uso de las redes de datos para la interconexión de áreas independientes y también para la prestación de los servicios.

Esta es, a grandes rasgos, la descripción de las diferentes ramas de investigación y acción del grupo y una pequeña reseña del nacimiento del mismo.