

**UNIVERSIDAD POLITÉCNICA SALESIANA
SEDE QUITO**

**CARRERA:
INGENIERÍA ELECTRÓNICA**

**Trabajo de titulación previo a la obtención del título de:
INGENIERO ELECTRÓNICO**

**TEMA:
DESARROLLO DE UN AMBIENTE CLOUD LINUX PARA MOODLE
ECUADOR**

**AUTOR:
CARLOS RONY RECALDE VITONERA**

**TUTOR:
LUIS GERMÁN OÑATE CADENA**

Quito, marzo del 2019

CESIÓN DE DERECHOS DE AUTOR

Yo Carlos Rony Recalde Vitonera, con documento de identificación N° 0705209831, manifiesto mi voluntad y cedo a la Universidad Politécnica Salesiana la titularidad sobre los derechos patrimoniales en virtud de que soy autor del trabajo de titulación intitulado: “DESARROLLO DE UN AMBIENTE CLOUD LINUX PARA MOODLE ECUADOR”, mismo que ha sido desarrollado para optar por el título de: Ingeniero Electrónico, en la Universidad Politécnica Salesiana, quedando la Universidad facultada para ejercer plenamente los derechos cedidos anteriormente.

En aplicación a lo determinado en la Ley de Propiedad Intelectual, en mi condición de autor me reservo los derechos morales de la obra antes citada. En concordancia, suscribo este documento en el momento que hago entrega del trabajo final en formato impreso y digital a la Biblioteca de la Universidad Politécnica Salesiana.

Quito, marzo del 2019.

A handwritten signature in blue ink that reads "Carlos Rony". The signature is stylized and includes a horizontal line underneath the name.

Carlos Rony Recalde Vitonera

C.I.: 0705209831

DECLARATORIA DE COAUTORÍA DEL DOCENTE TUTOR

Yo, declaro que bajo mi dirección y asesoría fue desarrollado el trabajo de titulación: **DESARROLLO DE UN AMBIENTE CLOUD LINUX PARA MOODLE ECUADOR** realizado por Carlos Rony Recalde Vitonera, obteniendo un producto que cumple con todos los requisitos estipulados por la Universidad Politécnica Salesiana, para ser considerados como trabajo final de titulación.

Quito, marzo del 2019



Luis Germán Oñate Cadena

C.I.: 0802820183

DEDICATORIA

Mi tesis la dedico al amor, esfuerzo, esmero y paciencia inagotable de dos seres humanos, mis padres. Ustedes me han permitido soñar grande, sé que voy a conquistar el mundo a mí alrededor.

En honor de quienes partieron antes de hora.

AGRADECIMIENTO

Agradezco al Todopoderoso, sin él no soy. Eu vou viver para crer que o meu Santo é forte e vai me proteger.

Gracias totales a mi Madre, Mercy Vitonera y a mi Padre, Carlos Recalde, por su infinito amor y todo su esfuerzo para con sus hijos.

A mi hermana, Mercy Recalde y a mi perro, Bruck, por brindar paz y felicidad a mi vida.

A mi cuñado, David Altamirano, por permitirme aprender de él. Y a su hermosa familia, por integrarme.

A mi tutor, Luis Oñate, por aceptar guiarme en el último paso previo a la obtención del título académico.

ÍNDICE

CESIÓN DE DERECHOS DE AUTOR.....	ii
DECLARATORIA DE COAUTORIA DEL DOCENTE TUTOR.....	iii
DEDICATORIA	iv
AGRADECIMIENTO.....	v
RESUMEN.....	x
ABSTRACT	xi
INTRODUCCIÓN	1
CAPÍTULO 1.....	2
ANTECEDENTES	2
1.1 Planteamiento del Problema.....	2
1.2 Justificación.....	2
1.3 Objetivos.....	2
1.4 Metodología.....	3
CAPÍTULO 2.....	4
MARCO TEÓRICO	4
2.1 Hosting	4
2.2 Cloud Computing	5
2.3 Diferencia entre sistema Hosting y sistema <i>Cloud</i>	9
2.4 Sistema Operativo Ubuntu server.....	9
2.5 OpenStack.....	10
2.6 Tipos de pruebas a servidores.....	12
2.7 Apache JMeter.....	13
CAPÍTULO 3.....	14
DESARROLLO DE UN AMBIENTE CLOUD LINUX PARA MOODLE ECUADOR.....	14
3.1 Descripción general de Moodle Ecuador	14
3.2 Servidor Hosting de Moodle Ecuador	14

3.3	Servidor de Cloud propuesto para Moodle Ecuador	15
3.4	Pruebas realizadas en Moodle Ecuador.....	19
3.5	Costos implicados para la implementación del sistema Cloud	29
CAPÍTULO 4.....		30
ANÁLISIS DE LOS RESULTADOS		30
4.1	Resultado de las pruebas realizadas sobre el sistema Hosting	30
4.2	Resultado de las pruebas realizadas sobre el sistema <i>Cloud</i>	32
4.3	Comparación entre resultados de pruebas en sistema Hosting y sistema <i>Cloud</i> 34	
CAPÍTULO 5.....		36
CONCLUSIONES Y RECOMENDACIONES		36
5.1	Conclusiones.....	36
5.2	Recomendaciones.....	37
LISTA DE REFERENCIAS		38
ANEXOS.....		2

ÍNDICE DE FIGURAS

Figura 2.1. Modelos de Servicios.....	6
Figura 2.2. Modelos de Despliegue.....	8
Figura 3.1. Servidor Hosting de Moodle Ecuador	15
Figura 3.2. Características de MVS	16
Figura 3.3. Servidor Cloud propuesto para Moodle Ecuador	17
Figura 3.4. Infraestructura Moodle Ecuador	19
Figura 3.5. Peticiones de 20 usuarios de la prueba de carga.....	20
Figura 3.6. Peticiones de 30 usuarios de la prueba de estrés	21
Figura 3.7. Características de la instancia 120	24
Figura 3.8. Peticiones de 50 usuarios de la prueba de carga.....	25
Figura 3.9. Peticiones de 100 usuarios de la prueba de carga.....	26
Figura 3. 10. Peticiones de 200 usuarios de la prueba de carga.....	27

ÍNDICE DE TABLAS

Tabla 3.1. Máquinas Virtuales creadas en la Cloud de Moodle Ecuador	18
Tabla 3.2. Datos de 10 peticiones	20
Tabla 3.3. Datos de 20 peticiones	21
Tabla 3.4. Datos de 30 peticiones	22
Tabla 3.5. Datos de 10 peticiones	22
Tabla 3.6. Datos de 20 peticiones	23
Tabla 3.7. Datos de 30 peticiones	23
Tabla 3.8. Datos de 50 peticiones	25
Tabla 3.9. Datos de 100 peticiones	26
Tabla 3.10. Datos de 200 peticiones	27
Tabla 3.11. Datos de 50 peticiones	28
Tabla 3.12. Datos de 100 peticiones	28
Tabla 3.13. Datos de 200 peticiones	29
Tabla 3.14. Costos de implementación del sistema Cloud.....	29
Tabla 4.1. Diferencias entre tiempos mínimos Hosting	30
Tabla 4.2. Diferencias entre tiempos máximos Hosting	31
Tabla 4.3. Diferencias entre porcentaje de error Hosting	31
Tabla 4.4. Rendimiento Hosting	32
Tabla 4.5. Diferencias entre tiempos mínimos Cloud.....	32
Tabla 4.6. Diferencias entre tiempos máximos Cloud	33
Tabla 4.7. Diferencias entre porcentaje de error Cloud	33
Tabla 4.8. Rendimiento Cloud	33
Tabla 4.9. Comparación de tiempos máximos	35

RESUMEN

El sistema Cloud es un sistema que presenta características deseables para que las empresas migren sus servicios. Las principales características, conocidas como calidad de servicio son: alta disponibilidad, seguridad contra ataques de DDoS (Ataque por denegación de servicios), firewall robusto, iptables, escalabilidad on demand y redundancia de lo alojado en la nube.

Moodle Ecuador utilizaba un sistema Hosting, el cual no poseía los suficientes recursos computacionales para la atención de alta cantidad de acceso de usuarios simultáneos (20 usuarios simultáneos máximo). Por ello, se propuso en el presente proyecto el diseño e implementación de un sistema Cloud para la empresa antes mencionada.

Para mostrar que el sistema Cloud es superior al sistema Hosting se realizó pruebas de tiempos máximos y mínimos de respuesta, para así obtener un porcentaje de error. Este depende de la cantidad de peticiones que el servidor tenga. Con las pruebas se demuestra que el sistema Hosting obtuvo un error de 31,33% con 30 peticiones, en cambio, el sistema Cloud obtuvo en error de 0% con 200 peticiones. La superioridad del sistema Cloud se basa en la escalabilidad en los picos de tráfico de usuarios y en la redundancia.

ABSTRACT

The system Cloud is a system that presents desirable characteristics in order that the companies migrate his services. The principal characteristics known as quality of service are: high availability, safety against DDoS's assaults (Assault for refusal of services), firewall robust, iptables, scalability on demand and redundancy of the lodged at the cloud.

Moodle Ecuador was using a Hosting system, which was not possessing the sufficient computational resources for the attention of high quantity of access of simultaneous users (20 simultaneous users maximum). For it, one proposed in the present project the design and implementation of a Cloud system for the company before mentioned. To show the Cloud system is superior to the Hosting system there were realized tests of maximum and minimums times of response, this way to obtain a percentage of mistake. East depends on the quantity of requests that the servant has. With the tests there is demonstrated that the Hosting system obtained a mistake of 31,33 % with 30 requests, on the other hand, the Cloud system it obtained a mistake of 0 % with 200 requests. The superiority of the Cloud system is based on the scalability on the beaks of users' traffic and on the redundanc

INTRODUCCIÓN

Un ambiente Hosting presenta recursos estáticos (no escalables) al momento de ser usada para plataformas con altos accesos simultáneos como Moodle, el cual, permiten realizar evaluaciones y cursos en línea. Estas funcionalidades necesitan una gran cantidad de recursos computacionales del servidor. Por el uso de ellos, se reduce la cantidad de usuarios que puedan acceder de manera simultánea a esta plataforma.

La Empresa Moodle Ecuador presenta como principal dificultad el acceso de una alta cantidad de usuarios simultáneos a la plataforma Moodle mediante la tecnología de Hosting. Para solucionar el problema de los recursos estáticos se plantea un sistema Cloud. La principal característica es la escalabilidad, los recursos pasan a ser dinámicos. Esta característica viene incluida en los acuerdos de servicio contratados al proveedor de la Cloud.

El contenido de este documento se encuentra dividido en cinco capítulos. El primer capítulo detalla la problemática detectada en la Empresa Moodle Ecuador y la justificación para el diseño e implementación de la solución planteada. Se definen los objetivos y la metodología a usar. El segundo capítulo describe un marco teórico referencial sobre ambientes Hosting, Cloud y herramientas usadas en el desarrollo del proyecto (sistema operativo Ubuntu Server, OpenStack, Havana y JMeter). El tercer capítulo muestra las características del sistema Hosting, el diseño e implementación de la Cloud y el resultado de las pruebas realizadas sobre estos dos sistemas. El cuarto capítulo se analiza los resultados obtenidos de las diferentes pruebas realizadas del tercer capítulo. El quinto y último capítulo se muestra las conclusiones y recomendaciones que se obtuvieron del presente estudio.

CAPÍTULO 1

ANTECEDENTES

En este capítulo se describe el planteamiento del problema, la justificación, los objetivos que deberán ser cumplidos, el alcance del proyecto y se detallará la metodología utilizada.

1.1 Planteamiento del Problema

La empresa Moodle Ecuador brinda diferentes servicios a sus clientes. Los servicios son de aulas virtuales mediante la plataforma Moodle, administración de servidores virtualizados, seguridad de sistemas, capacitaciones para docentes Universitarios, asesoría técnica a entidades públicas y privadas en el área de informática y redes de comunicación. Los servicios de aulas virtuales mediante la plataforma Moodle, administración de servidores virtualizados y seguridad de sistemas están alojados en un servidor Hosting contratado a una tercera empresa. El servidor Hosting implementado posee características insuficientes para la operatividad de los servicios mencionados por el aumento de clientes simultáneos. Por este motivo se propone el cambio de un servidor Hosting a un servidor *Cloud* el cual puede satisfacer las necesidades de los servicios implementados por la escalabilidad.

1.2 Justificación

El Hosting implementado posee características limitadas de almacenamiento, ancho de banda y velocidad. Esto provoca acceso limitado de usuarios y pérdidas de clientes. El *Cloud* posee características de escalabilidad. Esto significa que a mayor cantidad de clientes simultáneos que accedan a los servicios, el ancho de banda, almacenamiento y velocidad aumenten de manera automática. Este aumento depende del plan contratado con los proveedores de la *Cloud*.

1.3 Objetivos

1.3.1 Objetivo general

Implementar un ambiente *Cloud* basado en *Ubuntu Server* en la empresa Moodle Ecuador, para brindar servicios en la nube optimizando los recursos de la red.

1.3.2 Objetivos específicos.

- Analizar el estado actual de la red de la empresa Moodle Ecuador para conocer sus necesidades de negocios y de infraestructura y en base a esto buscar una solución usando un sistema *Cloud*.
- Diseñar un sistema *Cloud* basado en *Ubuntu Server* para satisfacer las necesidades de la empresa Moodle Ecuador.
- Realizar un análisis de costos para determinar el valor de implementar el sistema *Cloud*.
- Implementar el sistema *Cloud* de la red Moodle Ecuador para comprobar el funcionamiento de esta.

1.4 Metodología

- Recolectar información (teoría, implementación, costos, modelos) de un ambiente *Cloud* por medio de artículos científicos, revistas técnicas, documentación web, a través de la investigación teórica.
- Partiendo de los datos obtenidos, se buscará establecer el mejor procedimiento a seguir para la implementación del ambiente *Cloud* y sus respectivos servicios, a través de la investigación experimental.
- Se probará el ambiente *Cloud* con todos sus servicios a través de la investigación de campo por medio de pruebas.
- Finalmente, se analizará los resultados obtenidos en las pruebas de funcionamiento del ambiente *Cloud*.

CAPÍTULO 2

MARCO TEORICO

En este capítulo se describen los conceptos de Hosting y *Cloud computing*. Se detalla sus características más relevantes, modelos de servicio, modelos de despliegue, servicios en la *Cloud*. Adicionalmente se menciona el sistema operativo Ubuntu, herramienta *openstack* para el desarrollo de un ambiente *Cloud*, tipos de pruebas a servidores y JMeter.

2.1 Hosting

2.1.1 Concepto

El Hosting es el alojamiento en un servidor externo accesible por medio de una red, principalmente Internet. Las características del Hosting son dependientes del proveedor del servicio. Dependiendo del costo del paquete contratado al proveedor, las características como velocidad, ancho de banda, redundancia y tiempos de respuesta varían. (Molnar)

2.1.2 Modelos de Servicios

Los modelos de servicio en Hosting pueden ser con un servidor dedicado o compartido.

2.1.2.1 Servidor Dedicado

En este modelo de servicio se paga por los recursos completos de los servidores contratados. Se tiene el control total sobre los recursos del servidor, pero no sobre el ancho de banda. El ancho de banda se establece por el proveedor de servicio dependiendo del paquete contratado. El problema con este modelo de servidor reside en la escalabilidad. Cuando los recursos no abastecen a las peticiones, el servidor se satura y deja de funcionar correctamente.

2.1.2.2 Servidor Compartido

En este modelo de servicio se paga por una cantidad determinada de almacenamiento de un servidor debido a que, en el mismo servidor, es contratado por terceros al mismo tiempo. Los recursos son compartidos y solo se tiene control sobre el almacenamiento y no sobre los otros recursos. El problema con este modelo de servidor reside en los picos de tráfico. Estos pueden variar dependiendo de los terceros que contratan el

servidor. Si posee mayor pico un sitio contratado, este tercero toma la mayoría de los recursos computacionales reduciéndolos a los demás sitios alojados en el mismo servidor (SearchITChannel).

2.2 Cloud Computing

2.2.1 Concepto

Cloud Computing es el término utilizado tecnológicamente para el almacenamiento de datos en servidores externos que puedan ser alcanzados a través de una red. El concepto nació de la idea en la que todo el mundo pueda estar interconectado para poder acceder a los datos desde cualquier lugar. La computación en la nube se ha convertido en una importante tendencia tecnológica, y muchos expertos esperan que la computación en la *Cloud* modifique los procesos de tecnología de la información. (Furht & Escalante, 2010)

2.2.2 Características de Cloud Computing

Según el Instituto Nacional de Estándares y Tecnología de Estados Unidos (NIST), son 5 las principales características para tener en cuenta en el modelo de una *Cloud*:

- **Autoservicio bajo demanda**

Con esta característica un usuario de la *Cloud* tiene la posibilidad de hacer uso de los recursos computacionales cuando éste los necesite sin la necesidad que se dé una intervención por parte del proveedor, de forma automática y unilateral. (Joyanes Aguilar, 2012)

- **Acceso ubicuo a la Red**

Una característica clave para el *Cloud computing* es permitir la disponibilidad de sus servidores tanto para una red privada, pública o compartida, mediante métodos estandarizados que permitan a los usuarios poder acceder a los servicios desde cualquier dispositivo, ya sea un teléfono móvil, tableta, laptop o desktop. (Microsoft, 2014)

- **Distribución de recursos independientes de la posición**

En esta característica se hace referencia a la distribución de los recursos físicos y virtuales. Los recursos computacionales deben ser agrupados por el proveedor con la finalidad de poder ser distribuidos entre muchos usuarios (uso multiusuario),

diferentes recursos físicos y virtuales se asignan dinámicamente conforme a la demanda del consumidor. Entre los recursos que pueden ser distribuidos se encuentran el almacenamiento, procesamiento, memoria, ancho de banda de la red y máquinas virtuales. (Joyanes Aguilar, 2012)

- **Elasticidad rápida**

Con esta característica el usuario puede obtener de manera rápida y elástica los recursos, según como sea su necesidad al momento que los solicite. Sus características de aprovisionamiento dan la sensación de ser ilimitadas y puedan adquirirse en cualquier medida o momento. (Joyanes Aguilar, 2012)

- **Servicio medido**

Los usuarios en la *Cloud* pagan solo por la que consumen y cuando sea necesario para el usuario hacer uso de los recursos, por lo que se puede decir que el uso de sus recursos está medido, además de poder ser monitorizado y reportado al cliente, lo que brinda un acuerdo de transparencia entre el proveedor y el usuario, evitando los malentendidos por posibles cargos económicos indebidos. (Microsoft, 2014)

2.2.3 Modelos de Servicios

Los modelos de servicio hacen referencia a los servicios específicos a los que se puede acceder en una plataforma de *Cloud Computing* (Joyanes Aguilar, 2012)

Figura 2.1. Modelos de Servicios



Escenarios donde pueden desarrollarse cada uno de los modelos de servicios (Azure, 2018)

2.2.3.1 SaaS

Software as a Service (SaaS), es un modelo de servicio que permite al o los usuarios el uso de aplicaciones (mayoría de las veces de pago) basadas en una infraestructura de *Cloud* mediante una red, comúnmente usada Internet con navegadores web. La diferencia con el modelo tradicional es que las aplicaciones no deben ser instaladas en los dispositivos del usuario, van a estar almacenadas en los servidores del vendedor, que se enfoca en captar la atención de clientes empresariales, a los que les ofrece los mismos beneficios de software que en un modelo tradicional, pero mucho más amigable y a bajos precios.

En este modelo el usuario se encuentra limitado al uso de la aplicación, toda la infraestructura subyacente es total responsabilidad del vendedor. (Timothy Grance, 2011)

2.2.3.2 PaaS

Platform as a Service (PaaS), este modelo de servicio permite a los usuarios desarrollar sus propias aplicaciones dentro de una infraestructura *Cloud*, haciendo uso de lenguajes de programación, librerías, servicios y herramientas que son provistas por el proveedor de la *Cloud*. (Jadeja & Modi, 2012)

Los usuarios no podrán administrar o controlar la infraestructura *Cloud*, pero tendrán el control sobre las aplicaciones desarrolladas y las posibles configuraciones con las herramientas del ambiente. (Timothy Grance, 2011)

2.2.3.3 IaaS

Infrastructure as a Service (IaaS), en este modelo los usuarios van a poder acceder a los recursos informáticos de un ambiente *Cloud*, igual que los dos modelos antes mencionados, pero con la caracterización que se base en la virtualización de infraestructura de procesamiento, permitiendo que el usuario haga uso de los componentes de hardware virtualizados para su propia plataforma informática. (Bhardwaj, Jain, & Jain, 2010)

Los usuarios no podrán administrar o controlar la infraestructura *Cloud*, pero tendrán el control sobre sistemas operativos, almacenamiento y desarrollo de aplicaciones. (Timothy Grance, 2011)

2.2.4 Modelos de Despliegue

Los modelos de despliegue hacen referencia a 2 principales aspectos, que son, localización y gestión de la infraestructura de *Cloud computing*.

Figura 2.2. Modelos de Despliegue



Tipos de Cloud informática (Garrido, 2013)

2.2.4.1 *Cloud Pública*

Probablemente sea el tipo de *Cloud* más conocido, este tipo de nube está enfocado para ofertarse al público en general, los usuarios se conectan a una nube por medio de un navegador web dentro de dispositivos como una *laptop*, *smartphone* o *tablet*. Los usuarios no deben pagar por una infraestructura, sino adquirir planes de uso que se ajusten a sus necesidades, los valores de pago van a varias según los requerimientos del usuario, similar a un sistema prepago. La *Cloud* pública es menos seguras a diferencia de otros modelos de nube, por lo que pierden confianza entre los usuarios. (Ramgovind S, 2010)

2.2.4.2 *Nube Privada*

Es un tipo de nube, que, a diferencia con una nube pública, se enfoca en un público específico y no se oferta al público en general, generalmente corporaciones, hace uso de la infraestructura actual de una empresa, permite mejorar los niveles de seguridad, mejorar la escalabilidad de la red por costos menores que la vuelven una buena opción para las empresas. (Ramgovind S, 2010)

2.2.4.3 Nube Híbrida

Este modelo es caracterizado por la utilización de dos o más modelos de implementación, cada nube funciona de forma independiente, pero interactúan entre ellas. Los usuarios van a contar con las características propias de cada nube, lo que hace de este modelo el mejor escenario para los usuarios, ya que van a tener un escenario que cuenta con las mejores prestaciones de cada nube, como flexibilidad y escalabilidad con un menor riesgo. (Pedrosa & Nogueira, 2011)

2.2.4.4 Nube Comunitaria

La característica principal de este modelo es que se trata de la utilización de una misma nube, que se encuentra compartida entre varias organizaciones que comparten mismos objetivos, requisitos y políticas de seguridad. Es administrada de forma local o remota ya sea por las empresas o por algún tercero. (Pedrosa & Nogueira, 2011)

2.3 Diferencia entre sistema Hosting y sistema Cloud

La diferencia entre el servidor Hosting y el sistema *Cloud* se encuentra en la escalabilidad. En el sistema Hosting no posee escalabilidad, los recursos computacionales son fijos, dependientes del contrato de los servidores. En el sistema *Cloud* posee escalabilidad, los recursos computacionales son dinámicos. Si los recursos se están terminando, por medio de algoritmos, estos aumentan automáticamente. Este aumento solo se efectúa durante picos de tráfico en el servidor. (Molnar)

2.4 Sistema Operativo Ubuntu server

Ubuntu server es un sistema operativo de software libre y gratuito de código abierto. Está optimizado para gestión de servidores, escritorios virtuales e instancias en la *Cloud*. Para este último, dispone de un script llamado Cloud-init. Este habilita e inicia instancias en la nube con OpenStack. Se puede configurar a OpenStack desde este script y posteriormente ejecutarlo. (Ubuntu Server)

2.5 OpenStack

OpenStack es una herramienta desarrollada en software libre que permite crear ambientes de nubes tanto públicas como privadas, actualmente cuenta con varias versiones, cada una con sus propios módulos que le permiten al usuario trabajar a voluntad para crear su IaaS. (Jackson & Bunch, 2013)

2.5.1 Arquitectura de OpenStack Havana

Como se mencionó *OpenStack* es una herramienta desarrollada en software libre y se encuentra en constante mejora con nuevas actualizaciones desarrolladas por empresas como *Red Hat* y con la comunidad de software libre. Cada versión cuenta con características propias. A continuación, se detallan las características de componentes para la versión *OpenStack* Havana.

2.5.1.1 Almacenamiento de Objetos OpenStack (Swift)

Contar con un almacenamiento de objetos, permite contar con el almacenamiento y la recuperación de grandes objetos binarios que no son estructurados en datos, por medio de una API HTTP. Básicamente, toma un objeto entero y lo almacena en la red, evitando la partición y el almacenamiento por partes del mismo objeto. (Boucheron, 2018)

2.5.1.2 OpenStack Compute (Nova)

Por medio de esta característica el operario puede crear y administrar servidores virtuales con herramientas del sistema, en conjunto con otros de los servicios para su correcto funcionamiento. (OpenStack, docs.openstack, 2018)

En esta versión, Nova cuenta con una funcionalidad que permite que las instancias separadas por un período extendido se muevan fuera del hipervisor, y de esta forma, liberar recursos. (Bell, 2013)

2.5.1.3 Servicio de Imagen de OpenStack (Glance)

Este servicio permite al operario cargar y descubrir datos en estado activo utilizables con otros servicios, que incluye imágenes de las unidades de disco dura disponibles y metadatos. (OpenStack, docs.openstack, 2018)

En esta versión los usuarios pueden ser autorizados para crear, actualizar y leer diferentes propiedades de entidades arbitrarias. (Bell, 2013)

2.5.1.4 *OpenStack Dashboard (Horizon)*

Este servicio es el que proporciona la interfaz gráfica para la gestión de *OpenStack*, desde aquí el operario podrá ver la situación de su *Cloud* y hacer las gestiones. (OpenStack, wiki.openstack, 2018)

2.5.1.5 *Identidad OpenStack (Keystone)*

Por medio de este servicio se controlará la autenticación de los usuarios que ingresen a la nube, el acceso a los servicios a aplicaciones según, dependiendo la autorización de cada usuario. (OpenStack, docs.openstack, 2018)

2.5.1.6 *Servicio de Red de OpenStack (Neutron)*

Este servicio permite tener conectividad de red, comunica diferentes módulos, es así como *OpenStack*, se relaciona entre dispositivos de interfaz que sean administrados por servicios de *OpenStack*. (OpenStack, docs.openstack, 2018)

2.5.1.7 *Almacenamiento en Bloque de OpenStack (Cinder)*

Esta función brinda un dispositivo de almacenamiento, como un disco de almacenamiento de estado sólido, mediante la red. Con esta funcionalidad, el operario puede hacer uso de su dispositivo de almacenamiento y anejarlo a su máquina virtual. Estos dispositivos de almacenamiento son configurables, y pueden cambiar el tamaño de almacenamiento según la necesidad del cliente. Es una característica que brinda mucha flexibilidad para el consumidor. (Boucheron, 2018)

2.5.1.8 *Medición OpenStack (Ceilometer)*

Este servicio se establece por primera vez en esta versión. Le ofrece al operario contar con monitoreo de los usuarios vinculados a la infraestructura de la nube. Cuenta con alarmas que le van a permitir al usuario y operario actuar ante la comparación de tendencias estadística contra un límite durante un período de tiempo. (Bell, 2013)

2.5.1.9 Orquestación OpenStack (Heat)

Este servicio se establece por primera vez en esta versión y permite organizar aplicaciones basadas en la nube por medio de una REST API. Básicamente, permite establecer requisitos de una infraestructura en función de las aplicaciones del sistema. (Bell, 2013)

2.6 Tipos de pruebas a servidores

Las pruebas a servidores son necesarias para conocer el funcionamiento que estos tendrán bajo diferentes escenarios. Son necesarias durante todas las fases de implementación. Las pruebas utilizadas para conocer el uso de los recursos computacionales de un servidor son: pruebas de carga, de estrés y rendimiento. Se basan en tiempos de respuesta, errores y velocidad de procesamiento. Las cuales se detallan a continuación.

2.6.1 Prueba de carga

Las pruebas de carga se basan en medir el funcionamiento del servidor en condiciones normales y en el límite de los recursos que posea. Con esta prueba se conoce los diferentes tiempos de respuesta y el punto máximo o extremo que el servidor funciona correctamente.

2.6.2 Pruebas de estrés

Las pruebas de estrés se basan en medir el funcionamiento del servidor sobrepasando el límite de los recursos que posea. Con esta prueba se conoce los diferentes tiempos de respuesta, el porcentaje de error que ocurrió por sobrepasar el límite.

2.6.3 Pruebas de rendimiento

Las pruebas de rendimiento se basan en medir la capacidad del servidor en procesar peticiones enviadas. Con esta prueba se conoce la velocidad de peticiones por unidad de tiempo. (Simba)

2.7 Apache JMeter

Es una aplicación de código abierto gratuito útil para conocer el comportamiento funcional de servidores por medio de pruebas y medir el rendimiento. Esta aplicación ayuda a realizar pruebas de carga, estrés y rendimiento. Puede medir recursos dinámicos como estáticos. Trabaja a nivel de protocolos como HTTP, HTTPS, SOAP, REST, FTP, LDAP, MOM, SMTP, POP3 IMAN Y TCP. Es portable y multiplataforma. (The Apache Software Foundation)

CAPÍTULO 3

DESARROLLO DE UN AMBIENTE CLOUD LINUX PARA MOODLE ECUADOR

En este capítulo se describe a la Empresa Moodle Ecuador, el servidor Hosting, la implementación del servidor *Cloud* propuesto, las pruebas realizadas a los servidores y los costos necesarios para la implementación del servidor *Cloud*.

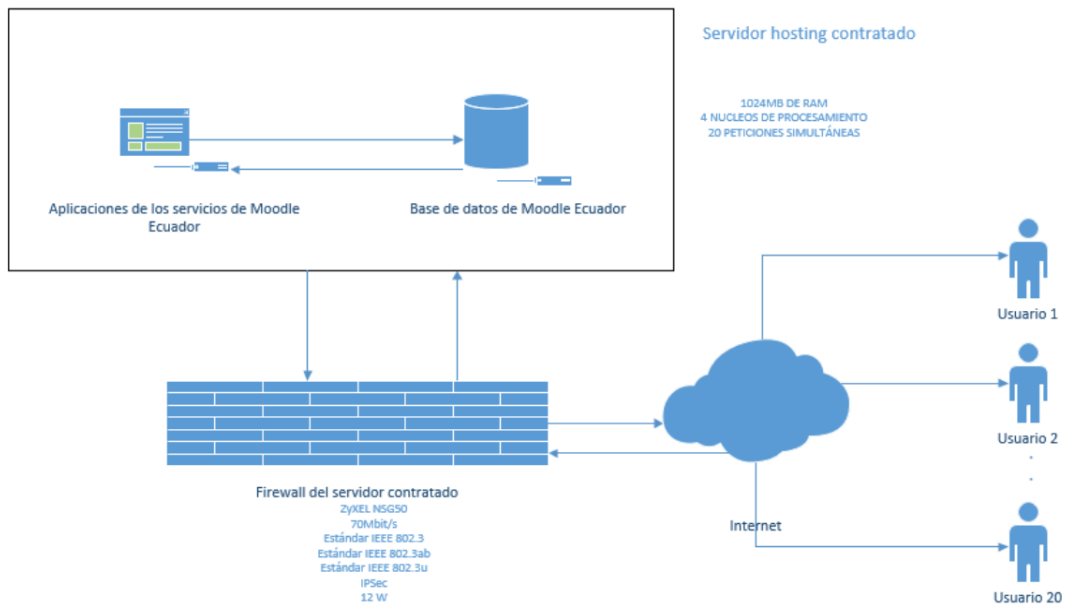
3.1 Descripción general de Moodle Ecuador

Moodle Ecuador es una empresa ecuatoriana con sede en Quito, dedicada a ofrecer una cartera de servicios tecnológicos. Dicha cartera está conformada por aulas virtuales mediante la plataforma Moodle, administración de servidores virtualizados, seguridad de sistemas, capacitaciones para docentes Universitarios, asesoría técnica a entidades públicas y privadas en el área de informática y redes de comunicación. Existen servicios ofrecidos en línea (Internet), los cuales están alojados en un servidor de tipo Hosting.

3.2 Servidor Hosting de Moodle Ecuador

El servidor Hosting en el cual está alojado los servicios de Moodle Ecuador posee las siguientes características: Memoria RAM de 1024 MB de uso máximo, CPU con 4 núcleos y 20 peticiones simultáneas entrantes. Las características de seguridad y redundancia no se las han implementado en este modelo por el contrato establecido con el proveedor. Sin embargo, el servidor posee un firewall estándar. En la Figura 3.1. se puede ver la descripción gráfica del servidor Hosting que posee Moodle Ecuador

Figura 3.1. Servidor Hosting de Moodle Ecuador

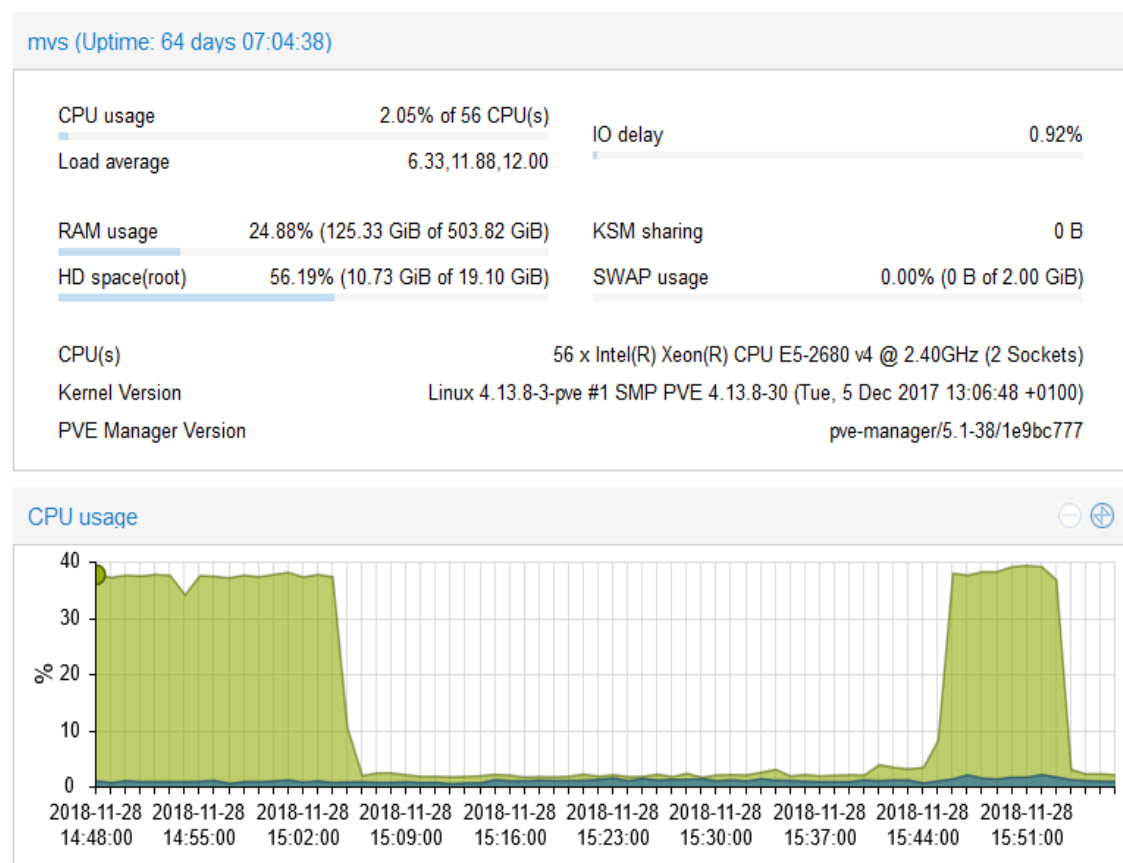


Elaborado en Visio 2013 por: Carlos Recalde

3.3 Servidor de Cloud propuesto para Moodle Ecuador

Moodle Ecuador no cuenta con una infraestructura propia para el montaje de la nube. Se realizó un acuerdo comercial para el alquiler de este servicio con la empresa OVH. OVH es una empresa francesa que se especializa en ofertar servidores dedicados, alojamiento compartido, y servicios *Cloud*. La *Cloud* implementada sigue el modelo de IaaS en una *Cloud* privada. Multiple Virtual Storage (MVS) es la *Cloud* propuesta. Este servidor cuenta con 56 núcleos distribuidos en 2 sockets físicos, una memoria RAM de 503.82 GB, para el hipervisor se cuenta con 19.10 GB de disco, para la memoria de intercambio (SWAP) se cuenta con 2 GB, el kernel que usa el sistema operativo (hipervisor) está basado en Linux 4.13.8-3-pve y un ancho de banda adaptable a los picos de tráfico. En la Figura 3.2 se muestra con detalles las características antes mencionadas.

Figura 3.2. Características de MVS



Elaborado por: Carlos Recalde

Además de lo mencionado, OVH ofrece características de calidad de servicio que son: disponibilidad del 99.9% y escalabilidad on demand.

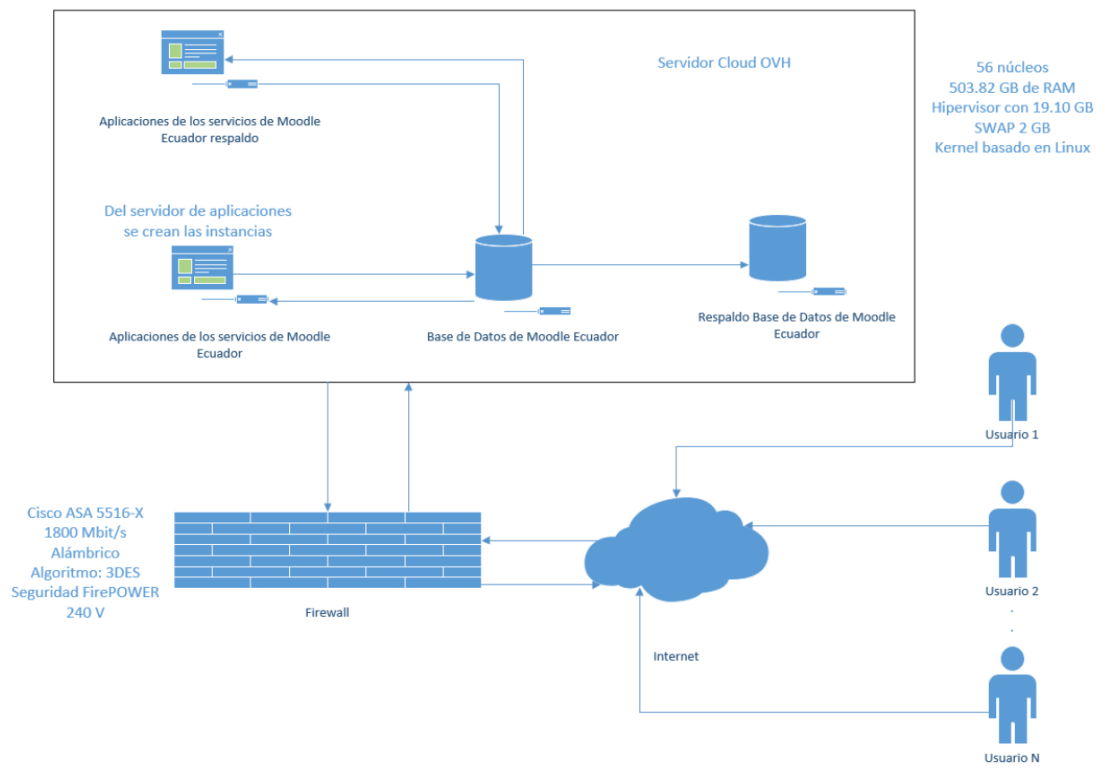
Todos estos servicios y características que ofrece OVH se encuentran en el acuerdo a nivel de servicios que posee la empresa Moodle Ecuador. Crear un modelo de red de esta infraestructura no es posible. Esto se debe a que no se conoce la topología de red que posee OVH.

Para la *Cloud* se hizo un diseño que consta de un servidor de aplicaciones, en el cual, se encuentran toda la cartera de servicios de Moodle Ecuador. Existe una copia de este servidor que, en el caso del que otro falle, esta se pondrá en producción. En este diseño también se posee dos servidores de bases de datos. Uno tendrá toda la información necesaria para el funcionamiento de las aplicaciones. El otro servirá de respaldos generados diariamente. De esta manera se implementa redundancia.

Se configura la seguridad mediante un script bash, en donde, se detallan todas las reglas que posee el firewall y el manejo de las iptables. Esta configuración se la realiza en el sistema operativo Ubuntu server. Estas reglas evitan que se sufra ataques por medio de protocolos o puertos. Por ejemplo, un ataque de DDoS (ataque de denegación de servicios). El contenido del script se lo encuentra en el Anexo 3.

Se implementa la *Cloud* con OpenStack Havana disponible gratuitamente. Se crean instancias (máquinas virtuales), las cuales representan a cada uno de los usuarios. Esto permite mejorar las características para que cada usuario reciba recursos completos exclusivos tales como la memoria y procesador. Las instancias permiten el monitoreo y gestión en tiempo real. El diseño detallado anteriormente se muestra en la Figura 3.3.

Figura 3.3. Servidor Cloud propuesto para Moodle Ecuador



Elaborado por: Carlos Recalde.

La Tabla 3.1. muestra las características de 21 instancias creadas con el fin de satisfacer las necesidades de los usuarios, además se observa a detalle las características del servidor para cada una de las instancias (CPU, MEMORY BOOTDISK), y un identificador de cada usuario.

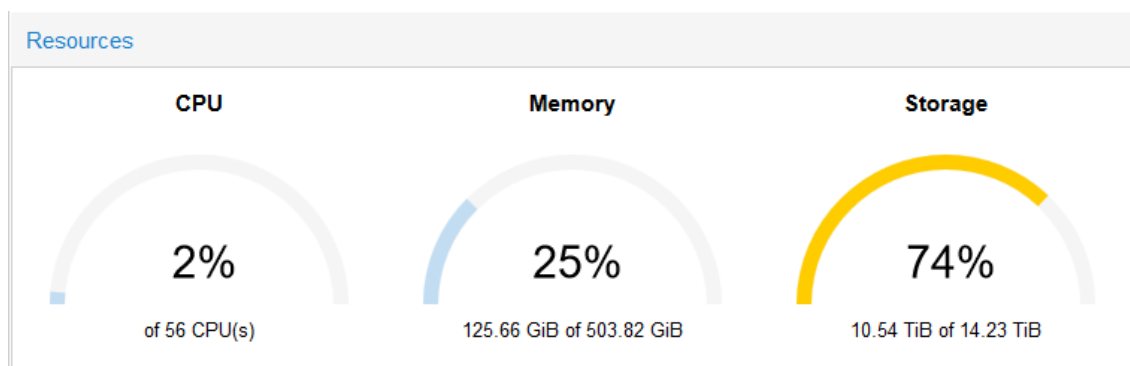
Tabla 3.1. Máquinas Virtuales creadas en la Cloud de Moodle Ecuador

No.	NOMBRE DE MÁQUINA VIRTUAL	CPU		MEMORY		BOOTDISK o ALMACENAMIENTO
		Disponible	Usado	Usado	Disponible	
1	101	8	0.13%	57.06%	10 GB	250 GB
2	102	2	0.29%	84.57%	2.50 GB	5.05 TB
3	103	4	0.95%	50.16%	4 GB	250 GB
4	104	8	3.54%	95.28%	20 GB	500 GB
5	105	4	0%	90.27%	5 GB	50 GB
6	106	1	0%	31.43%	3.02 GB	50 GB
7	108	1	1.15%	75.28%	1.46 GB	10 GB
8	109	1	1.16%	82.06%	2 GB	10 GB
9	110	1	0.54%	85.83%	3.02 GB	50 GB
10	111	1	0%	68.19%	3.02 GB	50 GB
11	112	1	0.59%	66.10%	3.02 GB	50 GB
12	113	4	1.07%	15.07%	32 GB	50 GB
13	114	1	0.69%	72.41%	3.02 GB	50 GB
14	115	1	0%	83.66%	4 GB	50 GB
15	116	1	0.91%	37.65%	3.02 GB	50 GB
16	117	2	2.39%	91.82%	2 GB	32 GB
17	118	2	0.51%	16.41%	8 GB	50 GB
18	119	2	0.49%	60.66%	4 GB	50 GB
19	120	2	3.54%	68.93%	4 GB	100 GB
20	127	20	0.32%	86.22%	2 GB	10 GB
21	200	2	0.26%	18.68%	4 GB	100 GB

Elaborado por: Carlos Recalde

En la Figura 3.4. se observa el porcentaje de los recursos que actualmente son utilizados del servidor contratado, estos valores se tomaron después de la migración de los usuarios descritos en la Tabla 3.1.

Figura 3.4. Infraestructura Moodle Ecuador



Elaborado por: Carlos Recalde

3.4 Pruebas realizadas en Moodle Ecuador

Se mostrará los resultados de pruebas de carga y estrés en los dos sistemas: Hosting y Cloud. Estas pruebas se las realizó con la herramienta JMeter.

Las unidades de las pruebas de rendimiento se han mantenido en segundos debido a la facilidad de comprensión de este valor. Al cambiarlos a milisegundos el análisis no se comprendería adecuadamente.

3.4.1 Sistema Hosting

3.4.1.1 Pruebas de carga y estrés

Para obtener los resultados de estas pruebas se han planteado tres escenarios:

El primer escenario es una prueba de carga. Se ha realizado 10 peticiones desde JMeter para permitir el acceso a los servicios de 10 usuarios en un intervalo de tiempo de 60 segundos. Los resultados obtenidos se encuentran detallados en la Tabla 3.2., en donde, se observa: el porcentaje de error de las peticiones realizadas es 0%. Todas las

10 peticiones enviadas fueron recibidas y estas obtuvieron un tiempo mínimo y máximo de respuesta de 1113 ms - 3811 ms respectivamente.

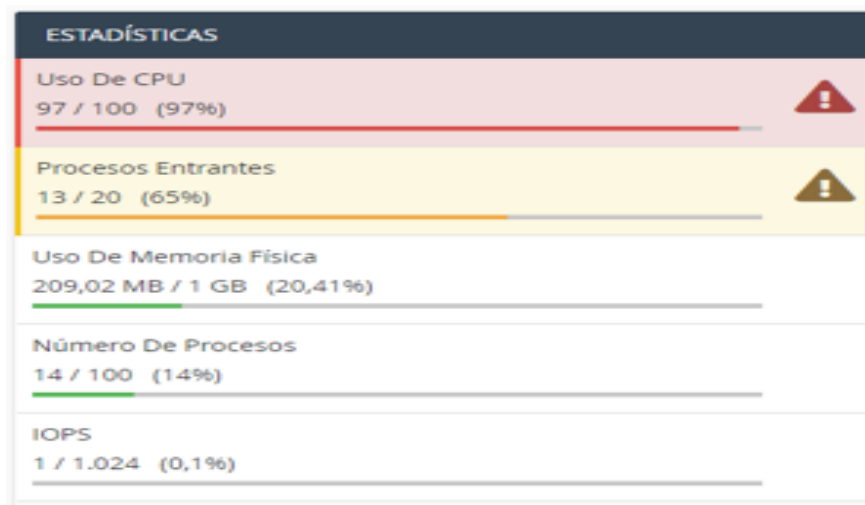
Tabla 3.2. Datos de 10 peticiones de la prueba de carga

	Peticiones	Min	Max	Error%
HTTP Request	10	1113	3811	0,00%

Elaborado por: Carlos Recalde

El segundo escenario es una prueba de carga. Se han realizado 20 peticiones desde JMeter para permitir el acceso a los servicios de 20 usuarios en un intervalo de tiempo de 60 segundos. En la Figura 3.5. se observan los recursos que han sido consumidos. En las 13 peticiones, el uso del CPU fue de 97%, mientras que la memoria física fue de 20,41%. Los resultados obtenidos se encuentran detallados en la Tabla 3.3., en donde, se observa: el porcentaje de error de las peticiones realizadas es 0%. Todas las 20 peticiones enviadas fueron recibidas y estas obtuvieron un tiempo mínimo y máximo de respuesta de 1218 ms – 7786 ms respectivamente.

Figura 3.5. Peticiones de 20 usuarios de la prueba de carga



Elaborado por: Carlos Recalde

Tabla 3.3. Datos de 20 peticiones de la prueba de carga

	Peticiones	Min	Max	Error%
HTTP Request	20	1218	7786	0,00%

Elaborado por: Carlos Recalde

El tercer escenario es una prueba de estrés. Se han realizado 30 peticiones desde JMeter para permitir el acceso a los servicios de 30 usuarios en un intervalo de tiempo de 60 segundos. En la Figura 3.6. se observan los recursos que han sido consumidos. En las 20 peticiones, el uso del CPU fue de 100%, mientras que la memoria física fue de 29,48%. Los resultados obtenidos se encuentran detallados en la Tabla 3.4., en donde, se observa: el porcentaje de error de las peticiones realizadas es 31,33%. De 30 peticiones enviadas, 20 fueron recibidas y estas obtuvieron un tiempo mínimo y máximo de respuesta de 548 ms – 11124 ms respectivamente.

Figura 3.6. Peticiones de 30 usuarios de la prueba de estrés



Elaborado por: Carlos Recalde

Tabla 3.4. Datos de 30 peticiones de la prueba de estrés

	Peticiones	Min	Max	Error%
HTTP Request	30	548	11124	31,33%

Elaborado por: Carlos Recalde

3.4.1.2 Pruebas de rendimiento

La herramienta JMeter llama al rendimiento Throughput. Para obtener los resultados de estas pruebas se han planteado 3 escenarios:

El primer escenario se ha realizado 10 peticiones desde JMeter para permitir realizar una evaluación en un aula virtual de 10 usuarios en un intervalo de tiempo de 60 segundos. Los resultados obtenidos detallan en la Tabla 3.5., se observa la cantidad de peticiones procesadas por segundo cuyo valor es 2 peticiones/segundo.

Tabla 3.5. Datos de 10 peticiones de la prueba de rendimiento

	Peticiones	Throughput (petición/segundo)
HTTP Request	10	2/sec

Elaborado por: Carlos Recalde

El segundo escenario se ha realizado 20 peticiones desde JMeter para permitir realizar una evaluación en un aula virtual de 20 usuarios en un intervalo de tiempo de 60 segundos. Los resultados obtenidos detallan en la Tabla 3.6., se observa la cantidad de peticiones procesadas por segundo cuyo valor es 2,5 peticiones/segundo.

Tabla 3.6. Datos de 20 peticiones de la prueba de rendimiento

	Peticiones	Throughput (petición/segundo)
HTTP Request	20	2,5/sec

Elaborado por: Carlos Recalde

El tercer escenario se ha realizado 30 peticiones desde JMeter para permitir realizar una evaluación en un aula virtual de 30 usuarios en un intervalo de tiempo de 60 segundos. Los resultados obtenidos detallan en la Tabla 3.7., se observa la cantidad de peticiones procesadas por segundo cuyo valor es 3,1 peticiones/segundo.

Tabla 3.7. Datos de 30 peticiones de la prueba de rendimiento

	Peticiones	Throughput (petición/segundo)
HTTP Request	30	3,1/sec

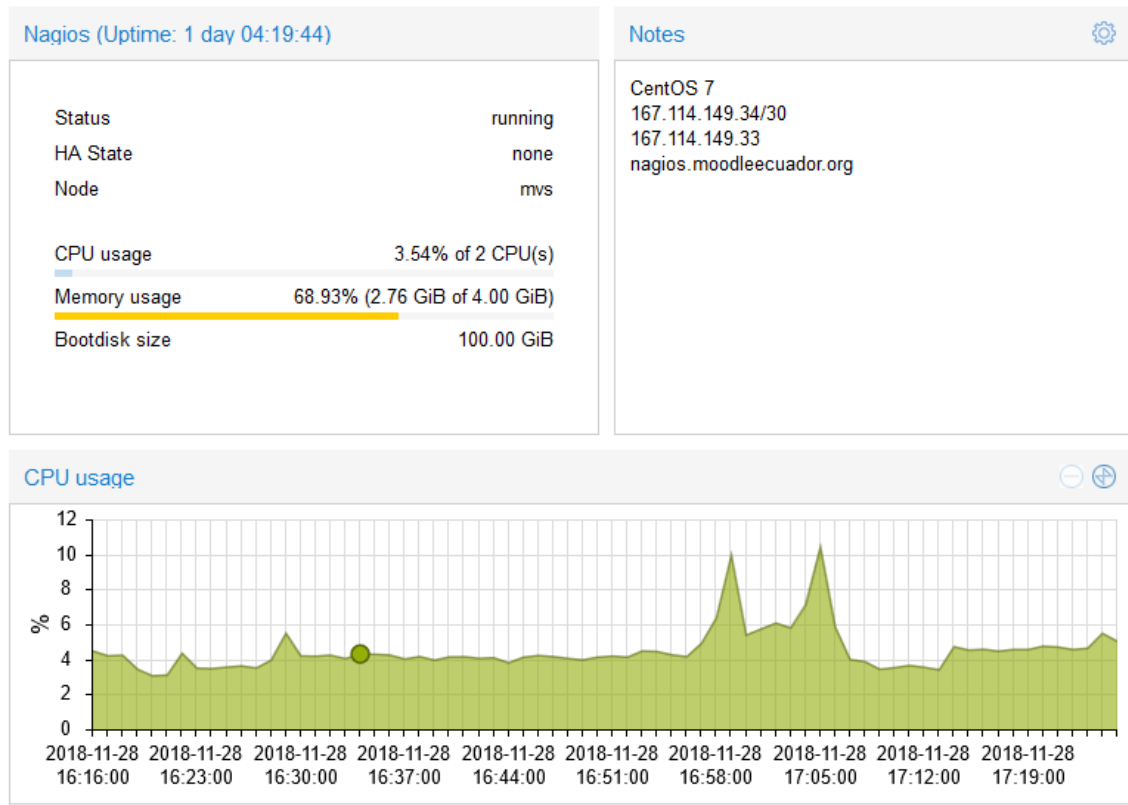
Elaborado por: Carlos Recalde

3.4.2 Sistema Cloud

3.4.2.1 Pruebas de carga y estrés

Las pruebas se realizaron en la instancia 120, esta instancia se encuentra detallada en la Tabla 3.1. La instancia cuenta con 2 núcleos, una memoria RAM de 4 y un almacenamiento 100 GB como se muestra en la Figura 3.7.

Figura 3.7. Características de la instancia 120

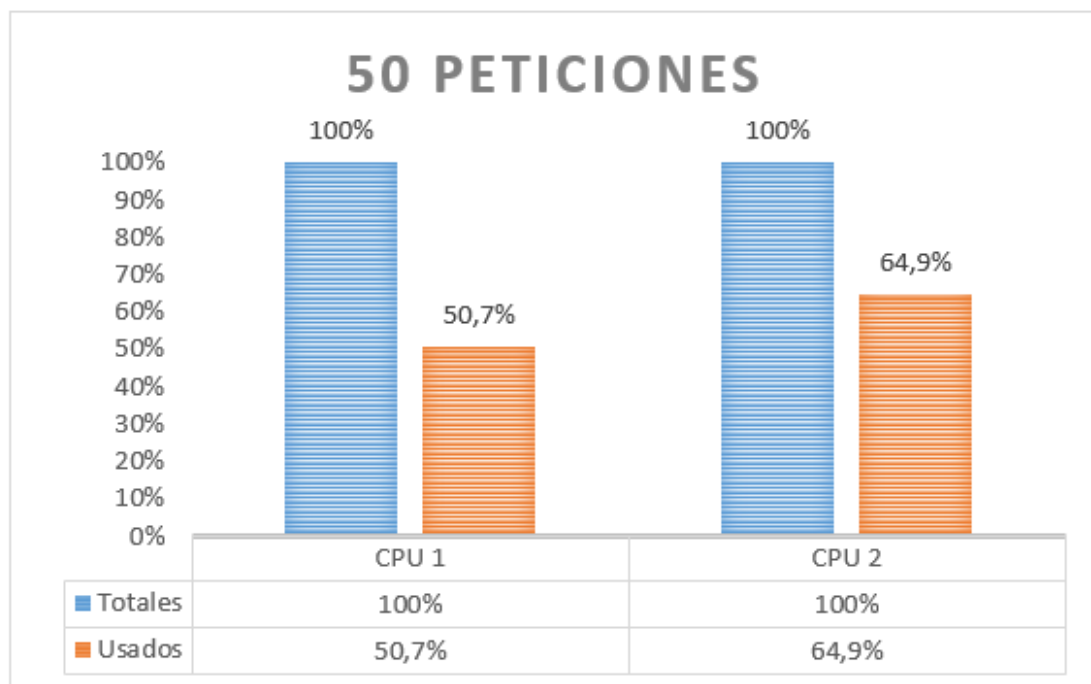


Elaborado por: Carlos Recalde

Para obtener los resultados de esta prueba se han planteado 3 escenarios:

El primer escenario se han realizado 50 peticiones desde JMeter para permitir el acceso a los servicios de 50 usuarios en un intervalo de tiempo de 60 segundos. En la Figura 3.8. se observan los recursos que han sido consumidos, luego de finalizar las peticiones.

Figura 3.8. Peticiones de 50 usuarios de la prueba de carga



Gráfica del resultado de 50 peticiones a la instancia. Elaborado por: Carlos Recalde

Los resultados obtenidos se encuentran detallados en la Tabla 3.8., en donde, se observa: el porcentaje de error de las peticiones realizadas es 0%. Todas las 50 peticiones enviadas fueron recibidas y estas obtuvieron un tiempo mínimo y máximo de respuesta de 1045 ms – 2821 ms respectivamente.

Tabla 3.8. Datos de 50 peticiones de la prueba de carga

	Peticiones	Min.	Max.	Error%
HTTP Request	50	1045	2821	0,00%

Elaborado por: Carlos Recalde

El segundo escenario se han realizado 100 peticiones desde JMeter para permitir el acceso a los servicios de 100 usuarios en un intervalo de tiempo de 60 segundos. En la Figura 3.9. se observan los recursos que han sido consumidos, luego de finalizar las peticiones.

Figura 3.9. Peticiones de 100 usuarios de la prueba de carga



Elaborado por: Carlos Recalde

Los resultados obtenidos se encuentran detallados en la Tabla 3.9., en donde, se observa: el porcentaje de error de las peticiones realizadas es 0%. Todas las 100 peticiones enviadas fueron recibidas y estas obtuvieron un tiempo mínimo y máximo de respuesta de 1053 ms - 3472 ms respectivamente.

Tabla 3.9. Datos de 100 peticiones de la prueba de carga

	Peticiones	Min	Max	Error%
HTTP Request	100	1053	3472	0,00%

Elaborado por: Carlos Recalde

El tercer escenario se ha realizado 200 peticiones desde JMeter para permitir el acceso a los servicios de 200 usuarios en un intervalo de tiempo de 60 segundos. En la Figura 3.10. se observan los recursos que han sido consumidos, luego de finalizar las peticiones.

Figura 3. 10. Peticiones de 200 usuarios de la prueba de carga



Elaborado por: Carlos Recalde

Los resultados obtenidos se encuentran detallados en la Tabla 3.10., en donde, se observa: el porcentaje de error de las peticiones realizadas es 0%. Todas las 200 peticiones enviadas fueron recibidas y estas obtuvieron un tiempo mínimo y máximo de respuesta de 1048 ms - 4098 ms respectivamente.

Tabla 3.10. Datos de 200 peticiones de la prueba de carga

	Peticiones	Min	Max	Error%
HTTP Request	200	1048	4098	0,00%

Elaborado por: Carlos Recalde

En la Cloud no se puede realizar la prueba de estrés debido a la escalabilidad que posee. Al no poseer un límite establecido, no se conoce el comportamiento al sobrepasarlo.

3.4.2.2 Pruebas de rendimiento

La herramienta JMeter llama al rendimiento Throughput. Para obtener los resultados de estas pruebas se han planteado 3 escenarios:

El primer escenario se ha realizado 50 peticiones desde JMeter para permitir realizar una evaluación en un aula virtual de 50 usuarios en un intervalo de tiempo de 60 segundos. Los resultados obtenidos detallan en la Tabla 3.11., se observa la cantidad de peticiones procesadas por segundo cuyo valor es 1,2 peticiones/segundo.

Tabla 3.11. Datos de 50 peticiones de la prueba de rendimiento

	Peticiones	Throughput (petición/segundo)
HTTP Request	50	1,2/sec

Elaborado por: Carlos Recalde

El segundo escenario se ha realizado 100 peticiones desde JMeter para permitir realizar una evaluación en un aula virtual de 100 usuarios en un intervalo de tiempo de 60 segundos. Los resultados obtenidos detallan en la Tabla 3.12., se observa la cantidad de peticiones procesadas por segundo cuyo valor es 1,7 peticiones/segundo.

Tabla 3.12. Datos de 100 peticiones de la prueba de rendimiento

	Peticiones	Throughput (petición/segundo)
HTTP Request	100	1,7/sec

Elaborado por: Carlos Recalde

El tercer escenario se ha realizado 200 peticiones desde JMeter para permitir realizar una evaluación en un aula virtual de 200 usuarios en un intervalo de tiempo de 60 segundos. Los resultados obtenidos detallan en la Tabla 3.13., se observa la cantidad de peticiones procesadas por segundo cuyo valor es 3,3 peticiones/segundo.

Tabla 3.13. Datos de 200 peticiones de la prueba de rendimiento

	Peticiones	Throughput (petición/segundo)
HTTP Request	200	3,3/sec

Elaborado por: Carlos Recalde

3.5 Costos implicados para la implementación del sistema Cloud

Para contratar el alojamiento en la nube, se solicitó a 3 empresas proformas con las mismas características. Y en la Tabla 3.14 se muestra la empresa y el valor del servicio.

Tabla 3.14. Costos de implementación del sistema Cloud

Empresa	Costo en dólares americanos mensuales
OVH	\$800,00
Amazon Web Service	\$1526,00
Windows Azure	\$1732,00

Elaborado por: Carlos Recalde

Por los valores mencionados se decidió contratar los servicios de la empresa OVH por ser la más económica.

Los costos de la implementación de la *Cloud* se especifican en la Tabla 3.15. Los costos descritos solo abarcan la implementación del sistema y no su mantenimiento.

Tabla 3. 15. Costos de implementación del sistema Cloud

Descripción	Costo en dólares americanos
Servicio de OVH	\$800,00
OpenStack Havana	\$0,00
Sistema Operativo Ubuntu server	\$0,00
Implementación	\$250,00
Total	\$1050,00

Elaborado por: Carlos Recalde

CAPÍTULO 4

ANÁLISIS DE LOS RESULTADOS

4.1 Resultado de las pruebas realizadas sobre el sistema Hosting

El tiempo mínimo crece al aumentar las peticiones de 10 a 20. En las 20 peticiones el servidor está a su mayor capacidad y necesita mayor tiempo de procesamiento. Sin embargo, en las 30 peticiones este tiempo mínimo decae. El servidor usa su mayor capacidad y al no poder procesar las peticiones se satura. La saturación del servidor conlleva a que una cantidad de peticiones no sean procesadas y el tiempo mínimo se reduzca. El servidor se satura debido a que el ancho de banda y la cantidad de usuarios permitidos de manera simultánea excede al máximo contratado (20 peticiones simultáneas). En la tabla 4.1. se observa este cambio.

Tabla 4.1. Diferencias entre tiempos mínimos Hosting

Peticiones	Tiempo Min (ms)
10	1113
20	1218
30	548

Elaborado por: Carlos Recalde

El tiempo máximo crece al aumentar las peticiones de 10 a 30. En las 20 peticiones el servidor está a su mayor capacidad y necesita mayor tiempo de procesamiento. En las 30 peticiones, el servidor usa su mayor capacidad y al no poder procesar las peticiones se satura. La saturación del servidor conlleva a que una cantidad de peticiones no sean procesadas y el tiempo máximo aumente debido a que el procesamiento es más lento. El servidor se satura debido a que el ancho de banda y la cantidad de usuarios permitidos de manera simultánea excede al máximo contratado (20 peticiones simultáneas). En la tabla 4.2. se observa este cambio.

Tabla 4.2. Diferencias entre tiempos máximos Hosting

Peticiones	Tiempo Max (ms)
10	3811
20	7786
30	11124

Elaborado por: Carlos Recalde

El porcentaje de error se mantiene constante en las peticiones de 10 a 20. El valor de este porcentaje es de 0. En las 20 peticiones el servidor está a su mayor capacidad, pero puede procesar todas las peticiones. En las 30 peticiones, el servidor usa su mayor capacidad y al no poder procesar todas las peticiones se satura. La saturación del servidor conlleva a que 10 peticiones no sean procesadas, lo que aumenta el porcentaje de error. El valor de este porcentaje en las 30 peticiones es 31,33%. Este valor significa que se perdió el 31,33% de las peticiones. En la tabla 4.3. se observa el porcentaje de error.

Tabla 4.3. Diferencias entre porcentaje de error Hosting

Peticiones	Porcentaje de error (%)
10	0%
20	0%
30	31,33%

Elaborado por: Carlos Recalde

El Throughput o rendimiento del servidor aumenta con la cantidad de peticiones. Por ello, cuando se realiza 30 peticiones este valor es el más alto. Este valor indica que se procesará más rápido cada petición por segundo, siendo 3,1 peticiones/s. Sin embargo, se debe tomar en cuenta los recursos computacionales disponibles. En el caso de las 30 peticiones, los recursos están a su máxima capacidad y no son suficientes para poder procesar todas las peticiones. Esto se ve reflejado en el porcentaje de error que indica que el 31,33% de las peticiones se perdieron. En la Tabla 4.4. se observa el rendimiento y el porcentaje de error.

Tabla 4.4. Rendimiento Hosting

Peticiones	Throughput (petición/segundo)	Porcentaje de error
10	2/sec	0%
20	2,5/sec	0%
30	3,1/sec	31,33%

Elaborado por: Carlos Recalde

4.2 Resultado de las pruebas realizadas sobre el sistema *Cloud*

El tiempo mínimo no sufre variaciones considerables en las peticiones de 50, 100 y 200. En la *Cloud*, los recursos son escalables en tiempo real, lo que significa, que el procesamiento se adapta a la cantidad de peticiones enviadas y mantiene valores cercanos al promedio de tiempos de respuesta mínimos contratado. En la tabla 4.5. se observa esto.

Tabla 4.5. Diferencias entre tiempos mínimos Cloud

Peticiones	Tiempo Min (ms)
50	1045
100	1053
200	1048

Elaborado por: Carlos Recalde

El tiempo máximo crece al aumentar las peticiones. En las 100 peticiones la instancia está a su mayor capacidad y necesita mayor tiempo de procesamiento. En las 200 peticiones, la *Cloud* aumenta los recursos de la instancia y puede procesar todas las peticiones en su mayor capacidad por la escalabilidad. En la tabla 4.6. se observa este cambio.

Tabla 4.6. Diferencias entre tiempos máximos Cloud

Peticiones	Tiempo Max (ms)
50	2821
100	3472
200	4098

Elaborado por: Carlos Recalde

El porcentaje de error se mantiene constante al aumentar las peticiones. El valor de este porcentaje es de 0. En las 100 y 200 peticiones el servidor está a su mayor capacidad, pero puede procesar todas las peticiones. No se produce error con ningún número de peticiones por la escalabilidad de la *Cloud*. En la tabla 4.7. se observa el porcentaje de error.

Tabla 4.7. Diferencias entre porcentaje de error Cloud

Peticiones	Porcentaje de error (%)
50	0%
100	0%
200	0%

Elaborado por: Carlos Recalde

El Throughput o rendimiento del servidor aumenta con la cantidad de peticiones. Este valor indica que se procesará más rápido cada petición por segundo. Para mantener el tiempo mínimo constante la cantidad de peticiones por segundo aumenta, En la Tabla 4.8. se observa el rendimiento.

Tabla 4.8. Rendimiento Cloud

Peticiones	Throughput (petición/segundo)
50	1,2/sec
100	1,7/sec
200	3,3/sec

Elaborado por: Carlos Recalde

4.3 Comparación entre resultados de pruebas en sistema Hosting y sistema *Cloud*

4.3.1 Número de peticiones

El sistema Hosting tiene una limitante en comparación con el sistema *Cloud*. Esta limitante es la cantidad de peticiones que puede aceptar, ya que este sistema solo puede aceptar un máximo de 20 peticiones antes de que el servidor sufra una saturación. En cambio, el sistema *Cloud* acepta una cantidad diez veces más de peticiones sin sufrir ningún problema de procesamiento. En el análisis realizado se observa que el sistema *Cloud* es la mejor opción para implementar por Moodle Ecuador.

4.3.2 Tiempos mínimos

En el sistema Hosting al aumentar el número de peticiones el tiempo mínimo de respuesta aumenta. Esta condición provoca que la respuesta al usuario sea más lenta en comparación al sistema *Cloud*. El sistema *Cloud* no posee esta desventaja. El tiempo mínimo sufre variaciones mínimas al aumentar el número de peticiones y la respuesta se mantiene constante. Por esta razón el sistema *Cloud* es el más competente a ser implementado por Moodle Ecuador.

4.3.3 Tiempos Máximos

En ambos sistemas los tiempos máximos aumentan con el número de peticiones. Pero el sistema con valores más altos es el Hosting. Este tiempo aumenta por la cantidad de recursos computacionales limitados que posee. En el sistema *Cloud* los tiempos máximos aumentan, pero no de manera excesiva como en el sistema Hosting, como se puede observar en la Tabla 4.9.

Tabla 4.9. Comparación de tiempos máximos

Hosting		Cloud	
Peticiones	Tiempo Max (ms)	Peticiones	Tiempo Max (ms)
30	11124	200	4098

Elaborado por: Carlos Recalde

4.3.4 Porcentaje de error

El sistema Hosting posee una desventaja con respecto al sistema *Cloud*. La cual consiste en que mientras el número de peticiones sobrepase el máximo (20), el porcentaje de error aumenta. Esto quiere decir que existen peticiones que no llegan a ser procesadas debido a la saturación que sufre el servidor. Esta situación no ocurre en el sistema *Cloud*. En este último, por más peticiones que se realicen, no existe un porcentaje de error por la escalabilidad.

4.3.5 Rendimiento

Como se observa en las tablas: Tabla 4.4. y Tabla 4.8. los rendimientos son similares en ambos sistemas. Pero tomando en cuenta la cantidad de peticiones que los sistemas reciben y procesan, se llega a la conclusión de que el sistema *Cloud* es el que posee mayor rendimiento, ya que este recibe muchas más peticiones que el sistema Hosting y tiene el mismo rendimiento.

Por los análisis realizados en el capítulo 4, se llega a la conclusión que el sistema *Cloud* es la mejor opción para implementar por la empresa Moodle Ecuador.

CAPÍTULO 5

CONCLUSIONES Y RECOMENDACIONES

5.1 Conclusiones

En el presente trabajo se logró cumplir los objetivos que se plantearon, desde implementar un ambiente *Cloud* en la empresa Moodle Ecuador, analizar el estado actual de la red (Hosting) de la empresa, diseñar el sistema *Cloud* y la realización de un análisis de costos de implementación.

Al analizar el sistema Hosting implementado en la empresa Moodle Ecuador se determinó que dicho sistema no era el adecuado ya que poseía algunas limitantes tales como la cantidad de accesos simultáneos (máximo 20), recursos computacionales insuficientes para el procesamiento y un porcentaje de error considerable de 31,33% al sobrepasar el máximo de peticiones. Y por ello se implementó el sistema *Cloud* en la empresa.

El diseño planteado se logró implementar sin inconvenientes, ya que dicho diseño satisface las necesidades de la empresa. Este sistema, a diferencia del otro, posee servidores de respaldo de aplicaciones y bases de datos. El costo de implementación sin tomar en cuenta la contratación del espacio en la nube es bajo debido a que se utiliza software libre gratuito. El espacio en la nube contratada a la empresa OVH se justifica por todas las características presentes en el paquete contratado. Por ello este sistema diseñado es más competitivo que el anterior sistema Hosting.

La escalabilidad es la principal diferencia entre los dos sistemas. En el caso del Hosting, no existe la escalabilidad (por este motivo al sobrepasar las 20 peticiones, el procesamiento de ellas produce un porcentaje de error). En el caso de la *Cloud*, la escalabilidad es automática dependiente del tráfico de peticiones. La escalabilidad de los recursos computacionales no posee límite, según el contrato de OVH. Esta aparente infinidad de recursos se debe a que a mayor tráfico de peticiones se le asignan nuevos recursos de otros servidores al procesamiento. Al decaer el tráfico, esta asignación se retira y se regresa a los recursos computacionales iniciales.

5.2 Recomendaciones

La Empresa debe tener respaldos del servidor de aplicaciones y de base de datos en otro proveedor de *Cloud*. Si la empresa OVH sufre algún ataque, la información puede verse comprometida y esto ocasionaría pérdidas tanto informáticas como económicas a Moodle Ecuador.

Medir los recursos ocupados por los servicios. Si la cantidad de recursos es excesiva, se debe contratar otro paquete de menores características a OVH para reducir costos.

LISTA DE REFERENCIAS

- Azure, M. (2018). *¿Qué es SaaS?* Obtenido de <https://azure.microsoft.com/es-es/overview/what-is-saas/>
- Bell, T. (Octubre de 2013). *wiki.openstack*. Obtenido de https://wiki.openstack.org/wiki/ReleaseNotes/Havana/pt_BR
- Bhardwaj, S., Jain, L., & Jain, S. (2010). CLOUD COMPUTING: A STUDY OF INFRASTRUCTURE AS A SERVICE (IAAS). *International Journal of Engineering and Information Technology*, pág. 4.
- Boucheron, B. (26 de Julio de 2018). *DigitalOcean*. Obtenido de <https://www.digitalocean.com/community/tutorials/servicios-de-armazenamento-de-objetos-versus-armazenamento-em-blocos-pt>
- Furht, B., & Escalante, A. (05 de 10 de 2010). *Handbook of Cloud Computing*. Springer. Obtenido de file:///C:/Users/Catherine/Downloads/C083_LNCS_CIARP12_postprint.pdf
- Garrido, J. (2013). *ASTIC*. Obtenido de <http://theweddingdrops.com/wp-content/uploads/2016/02/047.pdf>
- Jackson, K., & Bunch, C. (2013). *OpenStack Cloud Computing Cookbook*. Birmingham: Packt Publishing Ltd.
- Jadeja, Y., & Modi, K. (2012). Cloud Computing - Concepts, Architecture and Challenges. *IEEE*, 4.
- Joyanes Aguilar, L. (2012). COMPUTACIÓN EN LA NUBE NOTAS PARA UNA ESTRATEGIA ESPAÑOLA EN CLOUD COMPUTING. *Revista del Instituto Español de Estudios Estratégicos*, 24.
- Microsoft. (27 de Mayo de 2014). *5 características que definen la nube*. Obtenido de <https://news.microsoft.com/es-xl/5-caracteristicas-que-definen-la-nube/>
- OPEN CV. (19 de Septiembre de 2017). *Face Recognition with OpenCV*. Obtenido de OPEN CV: http://docs.opencv.org/2.4/modules/contrib/doc/facerec/facerec_tutorial.html#local-binary-patterns-histograms
- OpenStack. (24 de 10 de 2018). *docs.openstack*. Obtenido de <https://docs.openstack.org/nova/pike/>
- OpenStack. (05 de 11 de 2018). *docs.openstack*. Obtenido de <https://docs.openstack.org/glance/latest/>

- OpenStack. (09 de 10 de 2018). *docs.openstack*. Obtenido de <https://docs.openstack.org/keystone/pike/>
- OpenStack. (06 de 11 de 2018). *docs.openstack*. Obtenido de <https://docs.openstack.org/neutron/pike/>
- OpenStack. (05 de 11 de 2018). *wiki.openstack*. Obtenido de <https://wiki.openstack.org/wiki/Horizon>
- Pedrosa, P., & Nogueira, T. (2011). Computação em Nuvem. *UNICAMP*, 4.
- Ramgovind S, E. M. (2010). The management of Security in Cloud Computing. *School of Computing, University of South Africa*, 7.
- Timothy Grance, P. M. (2011). The NIST Definition of Cloud Computing. *National Institute of Standards and Technology*, 7.
- SearchITChannel. (20 de 01 de 2019). *hosted services*. Obtenido de <https://searchitchannel.techtarget.com/definition/hosted-services>
- Simba, G., Soto, S. & Gómez E. Factores críticos en pruebas de estrés en sistemas de información transaccionales. Escuela Politécnica del Ejército.
- Molnar, D & Schechter S. Self Hosting vs. Cloud Hosting: Accounting for the security impact of Hosting in the cloud. Microsoft Research.
- Ubuntu Server (20 de 01 de 2019). *ubuntu*. Obtenido de: <https://www.ubuntu.com/server>
- The Apache Software Foundation (20 de 01 de 2019). APACHE JMeter. Obtenido de: <https://jmeter.apache.org/>

ANEXOS

Anexo 1: REQUERIMIENTOS PREVIOS PARA LA INSTALACIÓN DE LA NUBE

Para el levantamiento de una nube inicialmente se necesitan de algunos servicios básicos que posteriormente van a permitir la creación y configuración de una nube, a continuación, se muestra la forma de instalar cada uno de ellos sobre Ubuntu.

- APTITUDE

Es un requerimiento para instalar la nube, le brinda la opción al usuario para ver la lista de paquetes y gestionar tareas, ya sean instalar, actualizar o eliminar paquetes. (Burrows & Manuel, 2016)

Figura A. 1: Instalación de Aptitude

```
root@ubuntu-HP-ENVY-TS-17-Notebook-PC:/home/ubuntu# apt-get install aptitude
```

Instalación de la interfaz de texto, aptitude, sobre Ubuntu. Elaborado por: Carlos Recalde

- KVM

Kernel-based Virtual Machine (KVM), permite que linux funcione como un hypervisor, que va a permitir a nuestra máquina servidor o host ejecutar entornos virtuales multiples. (RedHat, 2018)

QEMU, es un emulador y virtualizador de medios físicos. (QEMU, 2017)

Figura A. 2: Instalación de Hipervisor

```
root@ubuntu-HP-ENVY-TS-17-Notebook-PC:/home/ubuntu# aptitude -y install kvm qemu-kvm libvirt-bin virtinst bridge-utils
```

Instalación del KVM sobre Ubuntu. Elaborado por: Carlos Recalde

- VHOST_NET

Es una característica de KVM que permite transferir los paquetes de red entre el invitado y el host. (IBM, s.f.)

Figura A. 3: Habilitación de vhost_net

```
root@ubuntu-HP-ENVY-TS-17-Notebook-PC:/home/ubuntu# modprobe vhost_net
root@ubuntu-HP-ENVY-TS-17-Notebook-PC:/home/ubuntu# lsmod | grep vhost
vhost_net          18390  0
macvtap           18492  1 vhost_net
vhost              29287  1 vhost_net
root@ubuntu-HP-ENVY-TS-17-Notebook-PC:/home/ubuntu# echo vhost_net >> /etc/modules
```

Habilitación de vhost_net sobre Ubuntu. Elaborado por: Carlos Recalde

- PUENTE DE RED

La configuración por defecto de los adaptadores de red es automática (DHCP), para la implementación de la nube, es necesario establecer una ip estática de manera manual además de la creación de un puente que permita la comunicación de las máquinas virtuales y el host anfitrión.

Figura A. 4: Configuración por defecto

```
root@ubuntu-HP-ENVY-TS-17-Notebook-PC:/home/ubuntu# ifconfig
eth0      Link encap:Ethernet  direcciónHW fc:15:b4:ff:5c:63
          Dirección inet6: fe80::fe15:b4ff:feff:5c63/64 Alcance:Enlace
          ACTIVO DIFUSIÓN MULTICAST MTU:1500 Métrica:1
          Paquetes RX:434 errores:0 perdidos:0 overruns:0 frame:0
          Paquetes TX:750 errores:0 perdidos:0 overruns:0 carrier:0
          colisiones:0 long.colatX:1000
          Bytes RX:41887 (41.8 KB)  TX bytes:83730 (83.7 KB)
```

Valores por defecto del adaptador de red. Elaborado por: Carlos Recalde

Figura A. 5: Configuración de Interfaces

```
root@ubuntu-HP-ENVY-TS-17-Notebook-PC:/home/ubuntu# nano /etc/network/interfaces
```

Por medio de este comando se ingresa al archivo de configuración de interfaces. Elaborado por: Carlos Recalde

Figura A. 6: Configuración para el puente de red

```
iface br0 inet static
address 10.0.0.30
network 10.0.0.0
netmask 255.255.255.0
broadcast 10.0.0.255
gateway 10.0.0.1
dns-nameservers 10.0.0.30
bridge_ports eth0
bridge_stp off
auto br0
```

Valores asignados para la configuración del puente de red. Elaborado por: Carlos Recalde

Figura A. 7: Verificación de la configuración

```
root@ubuntu-HP-ENVY-TS-17-Notebook-PC:/home/ubuntu# ifconfig
br0      Link encap:Ethernet  direcciónHW fc:15:b4:ff:5c:63
         Direc. inet:10.0.0.30  Difus.:10.0.0.255  Másc:255.255.255.0
         Dirección inet6: fe80::fe15:b4ff:feff:5c63/64 Alcance:Enlace
         ACTIVO DIFUSIÓN FUNCIONANDO MULTICAST MTU:1500 Métrica:1
         Paquetes RX:8 errores:0 perdidos:0 overruns:0 frame:0
         Paquetes TX:44 errores:0 perdidos:0 overruns:0 carrier:0
         colisiones:0 long.colaTX:0
         Bytes RX:368 (368.0 B)  TX bytes:11786 (11.7 KB)

eth0     Link encap:Ethernet  direcciónHW fc:15:b4:ff:5c:63
         Direc. inet:10.0.0.30  Difus.:10.0.0.255  Másc:255.255.255.0
         Dirección inet6: fe80::fe15:b4ff:feff:5c63/64 Alcance:Enlace
         ACTIVO DIFUSIÓN FUNCIONANDO MULTICAST MTU:1500 Métrica:1
         Paquetes RX:442 errores:0 perdidos:0 overruns:0 frame:0
         Paquetes TX:845 errores:0 perdidos:0 overruns:0 carrier:0
         colisiones:0 long.colaTX:1000
         Bytes RX:42367 (42.3 KB)  TX bytes:107562 (107.5 KB)
```

Verificación de la configuración del puente de red. Elaborado por: Carlos Recalde

- NTP

Network Time Protocol (NTP), es un protocolo que permite la sincronización de los relojes de los sistemas informáticos. (NTP, 2014)

Figura A. 8: Verificación de la configuración

```
root@ubuntu-HP-ENVY-TS-17-Notebook-PC:/home/ubuntu# aptitude -y install ntp
```

Verificación del NTP sobre Ubuntu. Elaborado por: Carlos Recalde

- MySQL

Es un gestor de bases de datos, probablemente el más usado en la actualidad, administra los valores de configuración de cada servicio, necesarios para la implementación de la nube.

Figura A. 9: Verificación de la configuración

```
root@ubuntu-HP-ENVY-TS-17-Notebook-PC:/home/ubuntu# aptitude -y install mysql-server-5.5
```

Verificación del NTP sobre Ubuntu. Elaborado por: Carlos Recalde

Figura A. 10: Prueba de funcionalidad

```
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 42
Server version: 5.5.54-0ubuntu0.12.04.1 (Ubuntu)

Copyright (c) 2000, 2016, Oracle and/or its affiliates. All rights reserved.

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

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

mysql> select user,host,password from mysql.user;
+-----+-----+-----+
| user          | host                | password                                     |
+-----+-----+-----+
| root          | localhost           | *81F5E21E35407D884A6CD4A731AEBFB6AF209E1B |
| root          | ubuntu-hp-envy-ts-17-notebook-pc | *81F5E21E35407D884A6CD4A731AEBFB6AF209E1B |
| root          | 127.0.0.1          | *81F5E21E35407D884A6CD4A731AEBFB6AF209E1B |
| root          | ::1                 | *81F5E21E35407D884A6CD4A731AEBFB6AF209E1B |
| debian-sys-maint | localhost          | *EC730D82948C261B706FEAC4DA1871C917182F60 |
+-----+-----+-----+
5 rows in set (0.00 sec)

mysql> delete from mysql.user where user='';
Query OK, 0 rows affected (0.00 sec)
```

Inicialmente MySQL solicita establecer la clave de administrador. Elaborado por: Carlos Recalde

- PYTHON

Figura A. 11: Instalación de Python

```
root@ubuntu-HP-ENVY-TS-17-Notebook-PC:/home/ubuntu# aptitude -y install python-software-properties
```

Se hace la instalación de Python para que Ubuntu pueda reconocer los comandos que usa Python.
Elaborado por: Carlos Recalde

Anexo 2: OPENSTACK HAVANA

Se añade este repositorio con la finalidad de encontrar los instaladores necesarios y compatibles con la versión a instalar.

Figura A. 12: Repositorio OpenStack Havana

```
root@ubuntu-HP-ENVY-TS-17-Notebook-PC:/home/ubuntu# add-apt-repository cloud-archive:havana
```

Instalación del repositorio de OpenStack Havana sobre Ubuntu. Elaborado por: Carlos Recalde

Rabbitmq y Memcached

Rabbitmq gestiona la cola de mensajes entre el host y el virtual host mientras que el memcached se encarga de administrar la memoria ram, buscando que no se sature.

Figura A. 13: Instalación de rabbitmq y memcached

```
root@ubuntu-HP-ENVY-TS-17-Notebook-PC:/home/ubuntu# aptitude -y install rabbitmq-server memcached
```

Se instalan los servicios de rabbitmq y memcached. Elaborado por: Carlos Recalde

Se edita el archivo de configuración de MySQL para permitir que el rabbitmq enlace los mensajes entre el adaptador de red y el local host. Una vez realizado el cambio, se reinicia el servicio de MySQL.

Figura A. 14: Configuración de MySQL

```
root@ubuntu-HP-ENVY-TS-17-Notebook-PC:/home/ubuntu# nano /etc/mysql/my.cnf
```

Edición del archivo de configuración de MySQL en Ubuntu. Elaborado por: Carlos Recalde

Figura A. 15: Reinicio de MySQL

```
root@ubuntu-HP-ENVY-TS-17-Notebook-PC:/home/ubuntu# /etc/init.d/mysql restart
Rather than invoking init scripts through /etc/init.d, use the service(8)
utility, e.g. service mysql restart

Since the script you are attempting to invoke has been converted to an
Upstart job, you may also use the stop(8) and then start(8) utilities,
e.g. stop mysql ; start mysql. The restart(8) utility is also available.
mysql stop/waiting
mysql start/running, process 11385
```

Luego de la configuración, el servicio de MySQL se reinicia. Elaborado por: Carlos Recalde

Figura A. 16: Reinicio de MySQL

```
root@ubuntu-HP-ENVY-TS-17-Notebook-PC:/home/ubuntu# /etc/init.d/mysql restart
Rather than invoking init scripts through /etc/init.d, use the service(8)
utility, e.g. service mysql restart

Since the script you are attempting to invoke has been converted to an
Upstart job, you may also use the stop(8) and then start(8) utilities,
e.g. stop mysql ; start mysql. The restart(8) utility is also available.
mysql stop/waiting
mysql start/running, process 11385
```

Luego de la configuración, el servicio de MySQL se reinicia. Elaborado por: Carlos Recalde

Figura A. 17: Establecer contraseña al servicio rabbit

```
root@ubuntu-HP-ENVY-TS-17-Notebook-PC:/home/ubuntu# rabbitmqctl change_password guest password
```

Se debe establecer una contraseña para el uso del servicio rabbit. Elaborado por: Carlos Recalde

Figura A. 18: Establecer la interface de escucha

```
root@ubuntu-HP-ENVY-TS-17-Notebook-PC:/home/ubuntu# echo 'net.ipv4.ip_forward=1' >> /etc/sysctl.conf
root@ubuntu-HP-ENVY-TS-17-Notebook-PC:/home/ubuntu# sysctl -p
```

Se debe establecer la interface por la que se establece la conexión con el servicio. Elaborado por: Carlos Recalde

Figura A. 20: Crear base de datos para el servicio Keystone

```
root@ubuntu-HP-ENVY-TS-17-Notebook-PC:/home/ubuntu# mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 36
Server version: 5.5.54-0ubuntu0.12.04.1 (Ubuntu)

Copyright (c) 2000, 2016, Oracle and/or its affiliates. All rights reserved.

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

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

mysql> create database keystone character set utf8;
Query OK, 1 row affected (0.00 sec)

mysql> grant all privileges on keystone.* to keystone@%' identified by 'password';
Query OK, 0 rows affected (0.00 sec)

mysql> flush privileges;
Query OK, 0 rows affected (0.00 sec)

mysql> exit
Bye
```

Es necesario crear una base de datos donde se establecen configuraciones del servicio Keystone. Elaborado por: Carlos Recalde

Figura A. 21: Establecer configuración del servicio Keystone

```
root@ubuntu-HP-ENVY-TS-17-Notebook-PC:/home/ubuntu# keystone-manage db_sync
root@ubuntu-HP-ENVY-TS-17-Notebook-PC:/home/ubuntu# service keystone restart
keystone stop/waiting
keystone start/running, process 12684
```


Se realiza el reinicio del servicio Keystone para aplicar la configuración de base de datos. Elaborado por: Carlos Recalde

Figura A. 22: Carga de variables de entorno

```
root@ubuntu-HP-ENVY-TS-17-Notebook-PC:/home/ubuntu# export SERVICE_TOKEN=admintoken
root@ubuntu-HP-ENVY-TS-17-Notebook-PC:/home/ubuntu# export SERVICE_ENDPOINT=http://10.0.0.30:35357/v2.0/
```

Se cargan las variables de entorno y se establecen los valores del servicio de tokens. Elaborado por: Carlos Recalde

Figura A. 23: Carga de variables de entorno

```
root@ubuntu-HP-ENVY-TS-17-Notebook-PC:/home/ubuntu# export SERVICE_TOKEN=admintoken
root@ubuntu-HP-ENVY-TS-17-Notebook-PC:/home/ubuntu# export SERVICE_ENDPOINT=http://10.0.0.30:35357/v2.0/
```

Se cargan las variables de entorno y se establecen los valores del servicio de tokens. Elaborado por: Carlos Recalde

Figura A. 24: Carga y configuración de servicios

```
root@ubuntu-HP-ENVY-TS-17-Notebook-PC:/home/ubuntu# keystone tenant-create --name admin --description "Admin Tenant" --enabled true
+-----+-----+
| Property | Value |
+-----+-----+
| description | Admin Tenant |
| enabled | True |
| id | fdac3f79ab3c4cc188d7bbfbfa2e367 |
| name | admin |
+-----+-----+
```

Se cargan las configuraciones necesarias para el inicio de los servicios. Elaborado por: Carlos Recalde

Figura A. 25: Carga y configuración de servicios

```
root@ubuntu-HP-ENVY-TS-17-Notebook-PC:/home/ubuntu# keystone tenant-create --name service --description "Service Tenant" --enabled true
+-----+-----+
| Property | Value |
+-----+-----+
| description | Service Tenant |
| enabled | True |
| id | 40ed5a3ef9684894abe1b7ba9ebfd4e8 |
| name | service |
+-----+-----+
```

Se cargan las configuraciones necesarias para el inicio de los servicios. Elaborado por: Carlos Recalde

Figura A. 26: Carga y configuración de servicios

```
+-----+-----+
| Property | Value |
+-----+-----+
| id | aaecbbabc869443dbbc6a9e3ed8c5f88 |
| name | rony |
+-----+-----+
root@ubuntu-HP-ENVY-TS-17-Notebook-PC:/home/ubuntu# keystone role-create --name Member
+-----+-----+
| Property | Value |
+-----+-----+
| id | 558b254b58794fe19bedc9f11bedbf8c |
| name | Member |
+-----+-----+
```

Se cargan las configuraciones necesarias para el inicio de los servicios. Elaborado por: Carlos Recalde

Figura A. 27: Verificación de miembros del servicio

```
root@ubuntu-HP-ENVY-TS-17-Notebook-PC:/home/ubuntu# keystone role-list
```

id	name
558b254b58794fe19bedc9f11bedbf8c	Member
9fe2ff9ee4384b1894a90878d3e92bab	_member_
d5676f58bdfe4d279e8ef15413d6b499	admin
aaecbbabc869443dbbc6a9e3ed8c5f88	rony

Se verifican los miembros que tendrán acceso al servicio. Elaborado por: Carlos Recalde

Figura A. 28: Carga y configuración de servicios

```
root@ubuntu-HP-ENVY-TS-17-Notebook-PC:/home/ubuntu# keystone user-create --tenant admin --name admin --pass adminpassword --enabled true
```

Property	Value
email	
enabled	True
id	fbf21d4c0e154b76b4b5d2373549b712
name	admin
tenantId	fdac3f79ab3c4cc188d7bbfebfa2e367

```
root@ubuntu-HP-ENVY-TS-17-Notebook-PC:/home/ubuntu# keystone user-role-add --user admin --tenant admin --role admin
root@ubuntu-HP-ENVY-TS-17-Notebook-PC:/home/ubuntu# keystone user-role-add --user rony --tenant admin --role admin
No user with a name or ID of 'rony' exists.
root@ubuntu-HP-ENVY-TS-17-Notebook-PC:/home/ubuntu# keystone user-create --tenant service --name glance --pass servicepassword --enabled true
```

Property	Value
email	
enabled	True
id	f3abb81964ac4ffe88754f9cc7203e25
name	glance
tenantId	40ed5a3ef9684894abe1b7ba9ebfd4e8

```
root@ubuntu-HP-ENVY-TS-17-Notebook-PC:/home/ubuntu# keystone user-role-add --user glance --tenant service --role admin
root@ubuntu-HP-ENVY-TS-17-Notebook-PC:/home/ubuntu# keystone user-create --tenant service --name nova --pass servicepassword --enabled true
```

Property	Value
email	
enabled	True
id	99430adac87241b1a675775f5f704f8b
name	nova
tenantId	40ed5a3ef9684894abe1b7ba9ebfd4e8

Se cargan las configuraciones necesarias para el inicio de los servicios. Elaborado por: Carlos Recalde

Figura A. 29: Carga y configuración de servicios

```
root@ubuntu-HP-ENVY-TS-17-Notebook-PC:/home/ubuntu# keystone service-list
```

id	name	type	description
b393d054fddc47bda2467351b3ae5b4f	glance	image	Glance Image Service
55153ff92e794397bd2b2deaaa950394	keystone	identity	Keystone Identity Service
dccafa745b9745f698e2af630ffd54dd	nova	compute	Nova Compute Service

Se cargan las configuraciones necesarias para el inicio de los servicios. Elaborado por: Carlos Recalde

Figura A. 30: Configuración de servicio keystone

```
root@ubuntu-HP-ENVY-TS-17-Notebook-PC:/home/ubuntu# keystone endpoint-create --region RegionOne \
> --service keystone \
> --publicurl "http://$my_host:${public_port}/v2.0" \
> --internalurl "http://$my_host:${public_port}/v2.0" \
> --adminurl "http://$my_host:${admin_port}/v2.0"
+-----+-----+
| Property | Value |
+-----+-----+
| adminurl | http://10.0.0.30:${admin_port}/v2.0 |
| id       | 11847476dafc499fb7eabd0b2224dfde |
| internalurl | http://10.0.0.30:${public_port}/v2.0 |
| publicurl  | http://10.0.0.30:${public_port}/v2.0 |
| region    | RegionOne |
| service_id | 55153ff92e794397bd2b2deeea950394 |
+-----+-----+
```

Se definen las configuraciones para el servicio keystone. Elaborado por: Carlos Recalde

Figura A. 30: Verificación de configuración de servicio

```
root@ubuntu-HP-ENVY-TS-17-Notebook-PC:/home/ubuntu# keystone endpoint-list
+-----+-----+-----+-----+-----+
| id | service_id | region | publicurl | internalurl | adminurl |
+-----+-----+-----+-----+-----+
| 05251404851941bb8a206abd404c44d | RegionOne | http://10.0.0.30:9292/v1 | http://10.0.0.30:9292/v1 | http://10.0.0.30:9292/v1 |
| b393d054fd8dc47bda24673151b3ae5b4f | RegionOne | http://10.0.0.30:${public_port}/v2.0 | http://10.0.0.30:${public_port}/v2.0 | http://10.0.0.30:${admin_port}/v2.0 |
| 11847476dafc499fb7eabd0b2224dfde | RegionOne | http://10.0.0.30:${public_port}/v2.0 | http://10.0.0.30:${public_port}/v2.0 | http://10.0.0.30:${admin_port}/v2.0 |
| 55153ff92e794397bd2b2deeea950394 | RegionOne | http://10.0.0.30:${compute_port}/v1.1/${tenant_id}s | http://10.0.0.30:${compute_port}/v1.1/${tenant_id}s | http://10.0.0.30:${compute_port}/v1.1/${tenant_id}s |
| 7ab420367f27473391038f2c4545f3a1 | RegionOne | http://10.0.0.30:${compute_port}/v1.1/${tenant_id}s | http://10.0.0.30:${compute_port}/v1.1/${tenant_id}s | http://10.0.0.30:${compute_port}/v1.1/${tenant_id}s |
| dcca7a745b9745f698e2af630ffd54dd | RegionOne | http://10.0.0.30:${compute_port}/v1.1/${tenant_id}s | http://10.0.0.30:${compute_port}/v1.1/${tenant_id}s | http://10.0.0.30:${compute_port}/v1.1/${tenant_id}s |
+-----+-----+-----+-----+-----+
```

Se verifican las configuraciones realizadas en el servicio keystone. Elaborado por: Carlos Recalde

Figura A. 31: Configuración del servicio glance en la base de datos

```
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.
mysql> create database glance character set utf8;
Query OK, 1 row affected (0.00 sec)

mysql> grant all privileges on glance.* to glance@%' identified by 'password';
Query OK, 0 rows affected (0.00 sec)

mysql> flush privileges;
Query OK, 0 rows affected (0.00 sec)

mysql> exit
Bye
```

Se configuran las tablas necesarias para el normal funcionamiento del servicio glance. Elaborado por: Carlos Recalde

Figura A. 32: Registro y reinicio del servicio glance

```
root@ubuntu-HP-ENVY-TS-17-Notebook-PC:/home/ubuntu# nano /etc/glance/glance-registry.conf
root@ubuntu-HP-ENVY-TS-17-Notebook-PC:/home/ubuntu# nano /etc/glance/glance-api.conf
root@ubuntu-HP-ENVY-TS-17-Notebook-PC:/home/ubuntu# glance-manage db_sync
root@ubuntu-HP-ENVY-TS-17-Notebook-PC:/home/ubuntu# for service in api registry; do
> service glance-$service restart
> done
glance-api stop/waiting
glance-api start/running, process 13994
glance-registry stop/waiting
glance-registry start/running, process 13999
```

Para el funcionamiento de glance se registra las tablas creadas en la base datos y se reinicia el servicio. Elaborado por: Carlos Recalde

Figura A. 33: Configuración del servicio nova en la base de datos

```
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.
mysql> create database nova character set utf8;
Query OK, 1 row affected (0.00 sec)

mysql> grant all privileges on nova.* to nova@%' identified by 'password';
Query OK, 0 rows affected (0.00 sec)

mysql> flush privileges;
Query OK, 0 rows affected (0.00 sec)
```

Se configuran las tablas necesarias para el normal funcionamiento del servicio nova. Elaborado por: Carlos Recalde

Figura A. 34: Registro y reinicio del servicio nova

```
root@ubuntu-HP-ENVY-TS-17-Notebook-PC:/home/ubuntu# for service in api conductor network scheduler
> service nova-$service restart
> done
nova-api stop/waiting
nova-api start/running, process 15890
nova-conductor stop/waiting
nova-conductor start/running, process 15900
nova-network stop/waiting
nova-network start/running, process 15910
nova-scheduler stop/waiting
nova-scheduler start/running, process 15924
nova-objectstore stop/waiting
nova-objectstore start/running, process 15940
nova-cert stop/waiting
nova-cert start/running, process 15959
nova-consoleauth stop/waiting
nova-consoleauth start/running, process 15969
nova-compute stop/waiting
nova-compute start/running, process 15986
```

Para el funcionamiento de nova se registra las tablas creadas en la base datos y se reinicia el servicio. Elaborado por: Carlos Recalde

Figura A. 34: Verificación de servicios nova

```
root@ubuntu-HP-ENVY-TS-17-Notebook-PC:/home/ubuntu# nova-manage service list
Binary      Host                                Zone      Status      State Updated_At
nova-conductor  ubuntu-HP-ENVY-TS-17-Notebook-PC  internal  enabled     (-)  2018-11-13 05:18:22
nova-scheduler  ubuntu-HP-ENVY-TS-17-Notebook-PC  internal  enabled     (-)  2018-11-13 05:18:22
nova-network    ubuntu-HP-ENVY-TS-17-Notebook-PC  internal  enabled     (-)  2018-11-13 05:18:23
nova-compute    ubuntu-HP-ENVY-TS-17-Notebook-PC  nova      enabled     (-)  2018-11-13 05:18:23
nova-cert       ubuntu-HP-ENVY-TS-17-Notebook-PC  internal  enabled     (-)  2018-11-13 05:18:23
nova-consoleauth ubuntu-HP-ENVY-TS-17-Notebook-PC  internal  enabled     (-)  2018-11-13 05:18:23
```

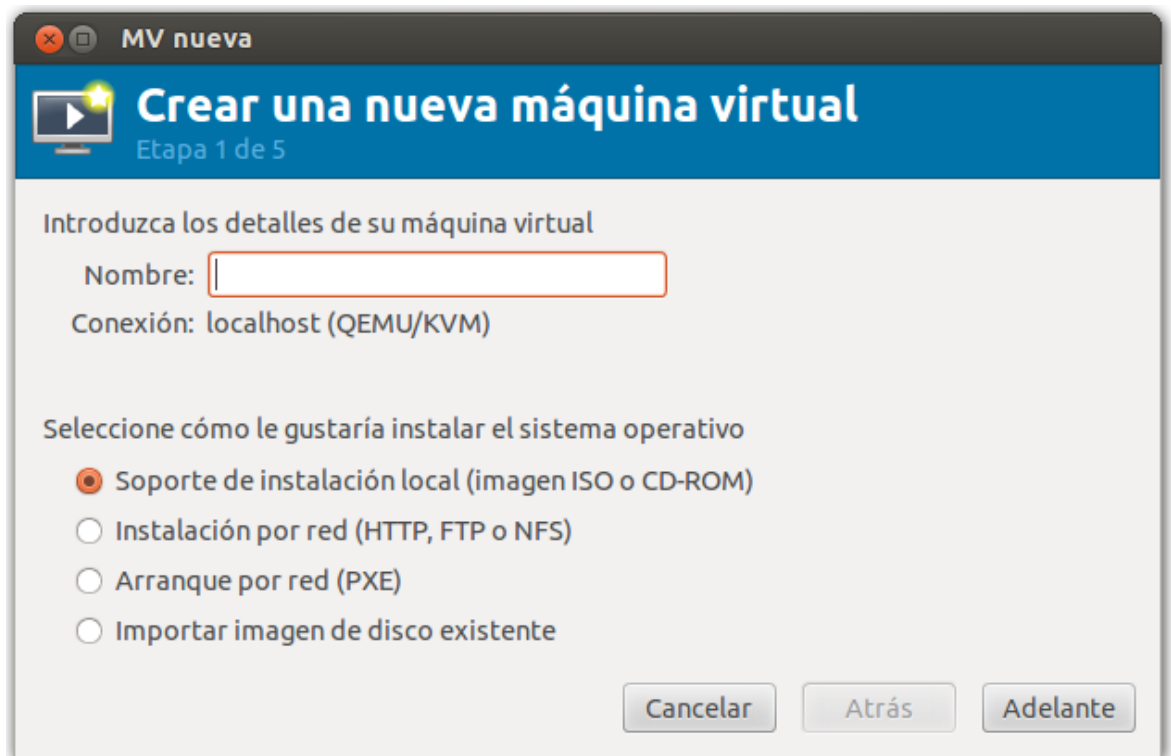
Se verifica que los servicios nova estén activos. Elaborado por: Carlos Recalde

Figura A. 34: Inicio de servicio keystone

```
root@ubuntu-HP-ENVY-TS-17-Notebook-PC:/home/ubuntu# nano ~/keystonerc
root@ubuntu-HP-ENVY-TS-17-Notebook-PC:/home/ubuntu# chmod 600 ~/keystonerc
root@ubuntu-HP-ENVY-TS-17-Notebook-PC:/home/ubuntu# source ~/keystonerc
root@ubuntu-HP-ENVY-TS-17-Notebook-PC ubuntu(keystone)# echo "source ~/keystonerc " >> ~/.bash_profile
```

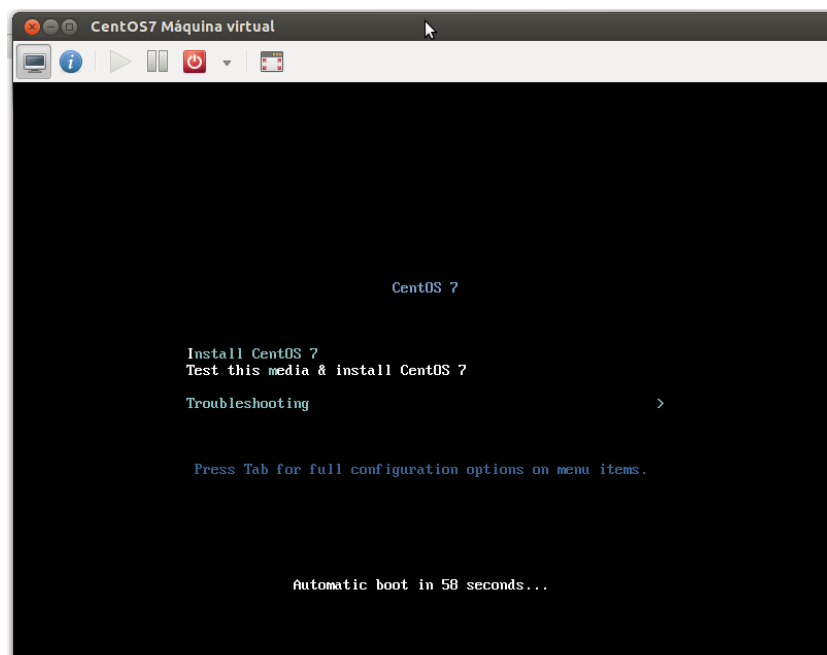
Se inicia sesión en el servicio keystone para continuar con la configuración. Elaborado por: Carlos Recalde

Figura A. 35: Creación de máquina virtual



Se crea máquina virtual que servirá de modelo para crear las instancias en el servicio. Elaborado por: Carlos Recalde

Figura A. 36: Creación de máquina virtual



Se crea máquina virtual que servirá de modelo para crear las instancias en el servicio. Elaborado por: Carlos Recalde

Figura A. 37: Agregar imagen a repositorio del servicio

```
root@ubuntu-HP-ENVY-TS-17-Notebook-PC ubuntu(keystone)# glance image-create --name="CentOS" --is-public=true --disk-format=raw
+-----+-----+
| Property | Value |
+-----+-----+
| checksum | ec41a294c9afe052a78f1c63cab15ed0 |
| container_format | bare |
| created_at | 2018-11-14T07:23:43 |
| deleted | False |
| deleted_at | None |
| disk_format | raw |
| id | bbab3585-5ecf-4465-818d-38e1eccb3a5 |
| is_public | True |
| min_disk | 0 |
| min_ram | 0 |
| name | CentOS |
| owner | fdac3f79ab3c4cc188d7bbfebfa2e367 |
| protected | False |
| size | 32212254720 |
| status | active |
| updated_at | 2018-11-14T07:28:32 |
+-----+-----+
```

Se agrega la imagen al repositorio del servicio glance, la cual será el modelo del que se crearan las instancias. Elaborado por: Carlos Recalde

Figura A. 38: Verificación imagen agregada

```
root@ubuntu-HP-ENVY-TS-17-Notebook-PC ubuntu(keystone)# glance image-list
+-----+-----+-----+-----+-----+-----+
| ID | Name | Disk Format | Container Format | Size | Status |
+-----+-----+-----+-----+-----+-----+
| bbab3585-5ecf-4465-818d-38e1eccb3a5 | CentOS | raw | bare | 32212254720 | active |
+-----+-----+-----+-----+-----+-----+
```

Se verifica que la imagen esté disponible en el servicio glance. Elaborado por: Carlos Recalde

Figura A. 39: Configuración de red del servicio nova

```
root@ubuntu-HP-ENVY-TS-17-Notebook-PC ubuntu(keystone)# nova-manage network create --label network01 --dns1 8.8.8.8 --fixed_range_v4=10.1.0.0/24
2018-11-14 02:30:37.666 28480 INFO nova.network.driver [-] Loading network driver 'nova.network.linux_net'
root@ubuntu-HP-ENVY-TS-17-Notebook-PC ubuntu(keystone)# nova-manage network list
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| id | IPV4 | IPV6 | Dirección Inicial | DNS1 | DNS2 | None | VLANID | None | proyecto | uuid |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| 1 | 10.1.0.0/24 | None | 10.1.0.2 | 8.8.8.8 | None | None | None | None | b1c9796e-fe0a-4acd-86f1-ae01f9c7dd4 |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
```

Se realiza la configuración de red para el funcionamiento del servicio nova. Elaborado por: Carlos Recalde

Figura A. 40: Configuración de grupo de seguridad

```
root@ubuntu-HP-ENVY-TS-17-Notebook-PC ubuntu(keystone)# nova secgroup-add-rule default tcp 22 22 0.0.0.0/0
+-----+-----+-----+-----+-----+
| IP Protocol | From Port | To Port | IP Range | Source Group |
+-----+-----+-----+-----+-----+
| tcp | 22 | 22 | 0.0.0.0/0 | |
+-----+-----+-----+-----+-----+
root@ubuntu-HP-ENVY-TS-17-Notebook-PC ubuntu(keystone)# nova secgroup-add-rule default icmp -1 -1 0.0.0.0/0
+-----+-----+-----+-----+-----+
| IP Protocol | From Port | To Port | IP Range | Source Group |
+-----+-----+-----+-----+-----+
| icmp | -1 | -1 | 0.0.0.0/0 | |
+-----+-----+-----+-----+-----+
root@ubuntu-HP-ENVY-TS-17-Notebook-PC ubuntu(keystone)# nova secgroup-list-rules default
+-----+-----+-----+-----+-----+
| IP Protocol | From Port | To Port | IP Range | Source Group |
+-----+-----+-----+-----+-----+
| tcp | 22 | 22 | 0.0.0.0/0 | |
| icmp | -1 | -1 | 0.0.0.0/0 | |
+-----+-----+-----+-----+-----+
```

Se crea un grupo de seguridad en el cual se añaden servicios que se permitirán su acceso. Elaborado por: Carlos Recalde

Figura A. 41: Verificación de recetas de creación de instancia

```
root@ubuntu-HP-ENVY-TS-17-Notebook-PC ubuntu(keystone)# nova flavor-list
```

ID	Name	Memory_MB	Disk	Ephemeral	Swap	VCPUS	RXTX_Factor	Is_Public
1	m1.tiny	512	1	0		1	1.0	True
2	m1.small	2048	20	0		1	1.0	True
3	m1.medium	4096	40	0		2	1.0	True
4	m1.large	8192	80	0		4	1.0	True
5	m1.xlarge	16384	160	0		8	1.0	True

Se verifica las recetas que servirán para la creación de las instancias con las características basadas en dichas recetas. Elaborado por: Carlos Recalde

Figura A. 42: Creación de instancia

```
root@ubuntu-HP-ENVY-TS-17-Notebook-PC ubuntu(keystone)# nova boot --flavor 3 --image CentOS --security_group default Generic
```

Property	Value
OS-EXT-STS:task_state	scheduling
image	CentOS
OS-EXT-STS:vm_state	building
OS-EXT-SRV-ATTR:instance_name	instance-00000001
OS-SRV-USG:launched_at	None
flavor	m1.medium
id	b06d4374-e9b4-4b1c-bee9-4da4cd771d7a
security_groups	[[{'name': 'default'}]]
user_id	fbf21d4c0e154b76b4b5d2373549b712
OS-DCF:diskConfig	MANUAL
accessIPv4	
accessIPv6	
progress	0
OS-EXT-STS:power_state	0
OS-EXT-AZ:availability_zone	nova
config_drive	
status	BUILD
updated	2018-11-14T07:35:50Z
hostId	
OS-EXT-SRV-ATTR:host	None
OS-SRV-USG:terminated_at	None
key_name	None
OS-EXT-SRV-ATTR:hypervisor_hostname	None
name	Generic
adminPass	aXzJF3HCq2Yq
tenant_id	fdac3f79ab3c4cc188d7bbfefa2e367
created	2018-11-14T07:35:50Z
os-extended-volumes:volumes_attached	[]
metadata	{}

Se crea una instancia basado en las recetas anteriores. Elaborado por: Carlos Recalde

Figura A. 43: Verificación de la creación de la instancia

```
root@ubuntu-HP-ENVY-TS-17-Notebook-PC ubuntu(keystone)# nova list
```

ID	Name	Status	Task State	Power State	Networks
b06d4374-e9b4-4b1c-bee9-4da4cd771d7a	Generic	BUILD	spawning	NOSTATE	network01=10.1.0.2

Se verifica que la instancia se haya creado correctamente. Elaborado por: Carlos Recalde

Figura A. 44: Prueba de conectividad con instancia

```
root@ubuntu-HP-ENVY-TS-17-Notebook-PC ubuntu(keystone)# ping 10.1.0.2
PING 10.1.0.2 (10.1.0.2) 56(84) bytes of data:
64 bytes from 10.1.0.2: icmp_req=1 ttl=64 time=0.216 ms
64 bytes from 10.1.0.2: icmp_req=2 ttl=64 time=0.172 ms
64 bytes from 10.1.0.2: icmp_req=3 ttl=64 time=0.146 ms
64 bytes from 10.1.0.2: icmp_req=4 ttl=64 time=0.188 ms
64 bytes from 10.1.0.2: icmp_req=5 ttl=64 time=0.141 ms
^C
--- 10.1.0.2 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 3997ms
rtt min/avg/max/mdev = 0.141/0.172/0.216/0.031 ms
```

Se realiza la prueba de conectividad con la instancia. Elaborado por: Carlos Recalde

Figura A. 45: Creación de IP flotante

```
root@ubuntu-HP-ENVY-TS-17-Notebook-PC ubuntu(keystone)# nova-manage floating create --ip_range=10.0.0.248/29
root@ubuntu-HP-ENVY-TS-17-Notebook-PC ubuntu(keystone)# nova-manage floating list
None 10.0.0.249 None nova eth0
None 10.0.0.250 None nova eth0
None 10.0.0.251 None nova eth0
None 10.0.0.252 None nova eth0
None 10.0.0.253 None nova eth0
None 10.0.0.254 None nova eth0
```

Se crean IP flotantes para realizar la comunicación con la instancia. Elaborado por: Carlos Recalde

Figura A. 46: Asignación de IP flotante

```
root@ubuntu-HP-ENVY-TS-17-Notebook-PC ubuntu(keystone)# nova list
+-----+-----+-----+-----+-----+-----+
| ID | Name | Status | Task State | Power State | Networks |
+-----+-----+-----+-----+-----+-----+
| aa92eccd-c162-48d9-80ea-2d1906fc4c6d | Generic | SHUTOFF | None | Shutdown | network01=10.1.0.3, 10.0.0.249 |
+-----+-----+-----+-----+-----+-----+

```

Se asigna una IP flotante a la instancia creada anteriormente. Elaborado por: Carlos Recalde

Figura A. 47: Carga y configuración de servicios

```
root@ubuntu-HP-ENVY-TS-17-Notebook-PC ubuntu(keystone)# keystone user-create --tenant service --name cinder --pass servicepassword --enabled true
+-----+-----+
| Property | Value |
+-----+-----+
| email | |
| enabled | True |
| id | ac801208479f4aa1ae96ca7ad1359b9b |
| name | cinder |
| tenantId | 40ed5a3ef9684894abe1b7ba9ebfd4e8 |
+-----+-----+
root@ubuntu-HP-ENVY-TS-17-Notebook-PC ubuntu(keystone)# keystone user-role-add --user cinder --tenant service --role admin
root@ubuntu-HP-ENVY-TS-17-Notebook-PC ubuntu(keystone)# keystone service-create --name=cinder --type=volume --description="Cinder Service"
+-----+-----+
| Property | Value |
+-----+-----+
| description | Cinder Service |
| id | ca0de60b81e94245af0b866c4890ba99 |
| name | cinder |
| type | volume |
+-----+-----+

```

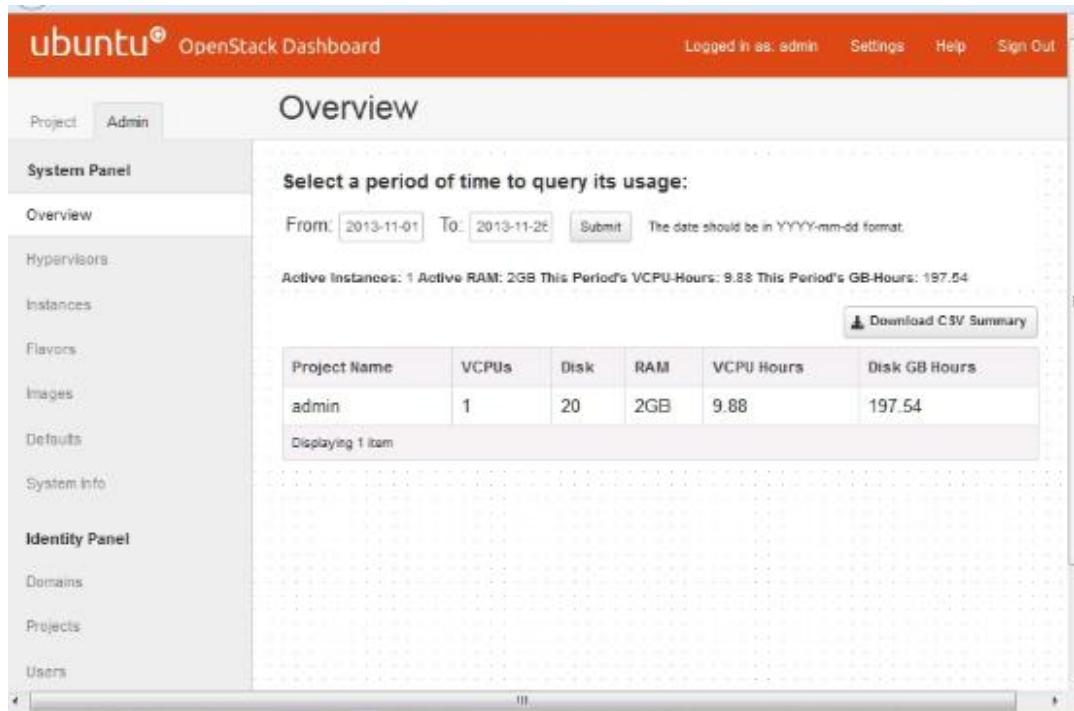
Se cargan las configuraciones necesarias para el inicio de los servicios. Elaborado por: Carlos Recalde

Figura A. 48: Carga y configuración de servicios

```
root@ubuntu-HP-ENVY-TS-17-Notebook-PC ubuntu(keystone)# nano /etc/cinder/cinder.conf
root@ubuntu-HP-ENVY-TS-17-Notebook-PC ubuntu(keystone)# nano /etc/cinder/api-paste.ini
root@ubuntu-HP-ENVY-TS-17-Notebook-PC ubuntu(keystone)# cinder-manage db sync
2018-11-14 03:23:37.900 11152 INFO migrate.versioning.api [-] 0 -> 1...
2018-11-14 03:23:40.357 11152 INFO migrate.versioning.api [-] done
2018-11-14 03:23:40.358 11152 INFO migrate.versioning.api [-] 1 -> 2...
2018-11-14 03:23:40.810 11152 INFO migrate.versioning.api [-] done
2018-11-14 03:23:40.810 11152 INFO migrate.versioning.api [-] 2 -> 3...
2018-11-14 03:23:40.968 11152 INFO migrate.versioning.api [-] done
2018-11-14 03:23:40.968 11152 INFO migrate.versioning.api [-] 3 -> 4...
2018-11-14 03:23:41.680 11152 INFO 004_volume_type_to_uuid [-] Created foreign key volume_type_extra_specs_ibfk_1
2018-11-14 03:23:41.703 11152 INFO migrate.versioning.api [-] done
2018-11-14 03:23:41.703 11152 INFO migrate.versioning.api [-] 4 -> 5...
2018-11-14 03:23:41.850 11152 INFO migrate.versioning.api [-] done
2018-11-14 03:23:41.850 11152 INFO migrate.versioning.api [-] 5 -> 6...
2018-11-14 03:23:42.053 11152 INFO migrate.versioning.api [-] done
2018-11-14 03:23:42.053 11152 INFO migrate.versioning.api [-] 6 -> 7...
```

Se cargan las configuraciones necesarias para el inicio de los servicios. Elaborado por: Carlos Recalde

Figura A. 49: Verificación de funcionamiento final de la nube



Se verifican que todos los servicios funcionen correctamente. Elaborado por: Carlos Recalde

Anexo 3: CONFIGURACIÓN IPTABLES Y FIREWALL

Crear el archivo que contendrá la configuración del firewall e iptables.

Figura A. 50: Creación del archivo que contendrá la configuración del firewall e iptables

```
root@ubuntu-hp-envy-ts-17-notebook-pc:/home/ubuntu# nano script.sh
```

Elaborado por: Carlos Recalde

Se coloca los comandos para la configuración. En el script se encuentra comentado las diferentes configuraciones.

Figura A. 51: Configuraciones del archivo de iptables

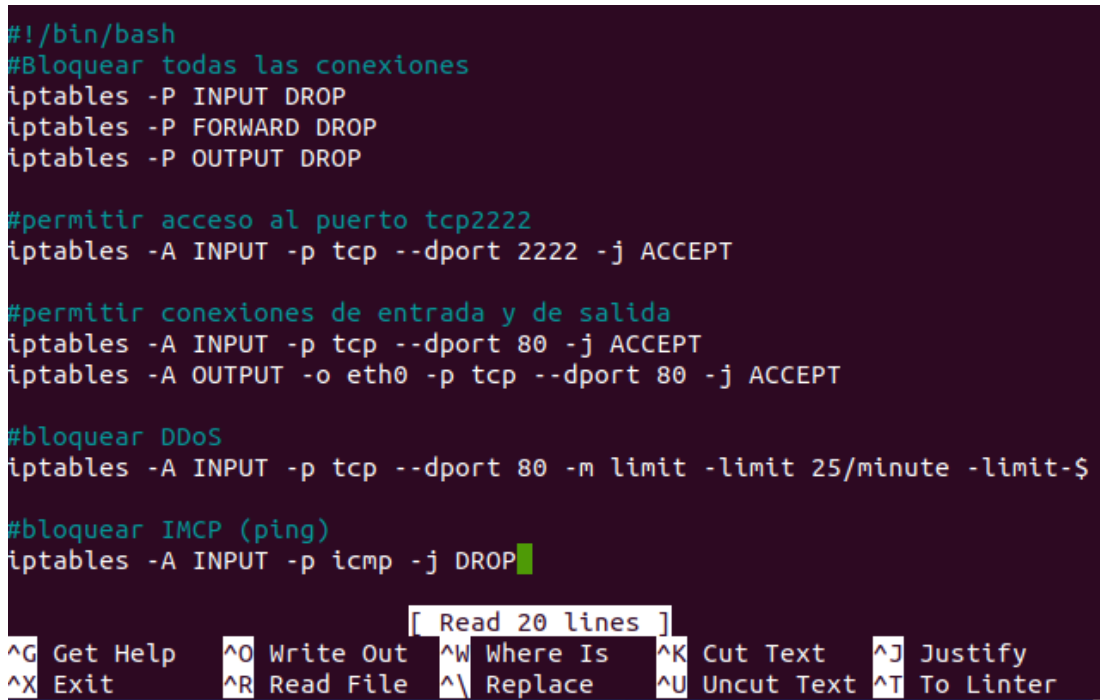
```
#!/bin/bash
#Bloquear todas las conexiones
iptables -P INPUT DROP
iptables -P FORWARD DROP
iptables -P OUTPUT DROP

#permitir acceso al puerto tcp2222
iptables -A INPUT -p tcp --dport 2222 -j ACCEPT

#permitir conexiones de entrada y de salida
iptables -A INPUT -p tcp --dport 80 -j ACCEPT
iptables -A OUTPUT -o eth0 -p tcp --dport 80 -j ACCEPT

#bloquear DDoS
iptables -A INPUT -p tcp --dport 80 -m limit --limit 25/minute --limit-burst 100 -j DROP

#bloquear IMCP (ping)
iptables -A INPUT -p icmp -j DROP
```



Elaborado por: Carlos Recalde

Estatus del Firewall antes de la configuración

Figura A. 52: Estatus del Firewall antes de la configuración

```
root@ubuntu-hp-envy-ts-17-notebook-pc:/home/ubuntu# iptables -L
Chain INPUT (policy ACCEPT)
target     prot opt source                destination

Chain FORWARD (policy ACCEPT)
target     prot opt source                destination

Chain OUTPUT (policy ACCEPT)
target     prot opt source                destination
```

Elaborado por: Carlos Recalde

Dar permisos de ejecución al script.

Figura A. 53: Permisos de ejecución al script

```
root@ubuntu-hp-envy-ts-17-notebook-pc:/home/ubuntu# chmod u+x script.sh
```

Elaborado por: Carlos Recalde

Ejecutar el script

Figura A. 54: Ejecutar el script

```
root@ubuntu-hp-envy-ts-17-notebook-pc:/home/ubuntu# ./script.sh
```

Elaborado por: Carlos Recalde

Estatus del Firewall después de la configuración

Figura A. 55: Estatus del Firewall después de la configuración

```
root@ubuntu-hp-envy-ts-17-notebook-pc:/home/ubuntu# iptables -L
Chain INPUT (policy DROP)
target     prot opt source                destination
ACCEPT     tcp  --  anywhere              anywhere             tcp dpt:2222
ACCEPT     tcp  --  anywhere              anywhere             tcp dpt:http
ACCEPT     tcp  --  anywhere              anywhere             tcp dpt:2222
ACCEPT     tcp  --  anywhere              anywhere             tcp dpt:http
ACCEPT     tcp  --  anywhere              anywhere             tcp dpt:2222
ACCEPT     tcp  --  anywhere              anywhere             tcp dpt:http
DROP       icmp --  anywhere              anywhere
ACCEPT     tcp  --  anywhere              anywhere             tcp dpt:2222
ACCEPT     tcp  --  anywhere              anywhere             tcp dpt:http
DROP       icmp --  anywhere              anywhere
ACCEPT     tcp  --  anywhere              anywhere             tcp dpt:2222
ACCEPT     tcp  --  anywhere              anywhere             tcp dpt:http
DROP       icmp --  anywhere              anywhere

Chain FORWARD (policy DROP)
target     prot opt source                destination

Chain OUTPUT (policy DROP)
target     prot opt source                destination
ACCEPT     tcp  --  anywhere              anywhere             tcp dpt:http
ACCEPT     tcp  --  anywhere              anywhere             tcp dpt:http
ACCEPT     tcp  --  anywhere              anywhere             tcp dpt:http
ACCEPT     tcp  --  anywhere              anywhere             tcp dpt:http
ACCEPT     tcp  --  anywhere              anywhere             tcp dpt:http
```

Elaborado por: Carlos Recalde