

**PONTIFICIA UNIVERSIDAD CATÓLICA DEL ECUADOR**



**FACULTAD DE INGENIERÍA**

**MAESTRÍA EN REDES DE COMUNICACIÓN**

**INFORME FINAL CASO DE ESTUDIO PARA UNIDAD DE TITULACIÓN  
ESPECIAL**

**TEMA:**

**“Análisis e Implementación de una solución para automatizar la configuración de los  
enlaces en un ambiente de Computación en la Nube provistos por una Red De  
Transporte MPLS. Caso de estudio CNT E.P.”**

***Juan Rolando Chicaiza Acosta***

Quito – 2016

## **AUTORÍA**

Yo, ***Juan Rolando Chicaiza Acosta***, portador de la cédula de ciudadanía No. ***1712173705***, declaro bajo juramento que la presente investigación es de total responsabilidad del autor, y que se ha respetado las diferentes fuentes de información realizando las citas correspondientes. Esta investigación no contiene plagio alguno y es resultado de un trabajo serio desarrollado en su totalidad por mi persona.

---

***Juan Rolando Chicaiza Acosta***

## CONTENIDO

1.	Introducción.....	15
2.	Justificación.....	16
3.	Antecedentes .....	19
4.	Objetivos .....	22
	Objetivo General:.....	22
	Objetivos Específicos: .....	22
5.	Análisis de tecnologías utilizadas por el proveedor del caso de estudio para brindar servicios de conectividad a Data Centers Virtuales.....	24
5.1	Modelo Tecnológico de Computación en la Nube .....	24
5.1.1	Características Claves.....	26
5.1.1.1	Auto Servicio bajo demanda .....	26
5.1.1.2	Amplio acceso desde la red.....	26
5.1.1.3	Disponibilidad de Recursos.....	27
5.1.1.4	Multi-Propietario (Multi-Tenancy).....	27
5.1.1.5	Rápida elasticidad.....	27
5.1.1.6	Servicio Medido .....	27
5.1.2	Roles y Actividades.....	28

5.1.2.1	Cliente del servicio en la Nube.....	28
5.1.2.2	Socio del servicio en la Nube .....	28
5.1.2.3	Proveedor de servicios en la Nube .....	28
5.1.3	Tipo de capacidades y Categoría de servicios .....	29
5.1.3.1	Capacidades por Aplicación.....	29
5.1.3.2	Capacidades por Infraestructura .....	29
5.1.3.3	Capacidades por Plataforma.....	29
5.1.3.4	Comunicaciones como servicio (CaaS) .....	30
5.1.3.5	Computación como servicio (CompaaS) .....	30
5.1.3.6	Data Storage como servicio (DSaaS) .....	30
5.1.3.7	Network como servicio (NaaS) .....	30
5.1.3.8	Software como Servicio (SaaS).....	30
5.1.3.9	Plataforma como Servicio (PaaS).....	30
5.1.3.10	Infraestructura como Servicio (IaaS).....	31
5.1.4	Modelos de Despliegue .....	31
5.1.4.1	Nube Privada .....	31
5.1.4.2	Nube Comunitaria.....	31
5.1.4.3	Nube Pública .....	32
5.1.4.4	Nube Híbrida .....	32
5.2	Conmutación Multiprotocolo basado en Etiquetas, MPLS (Multiprotocol Label Switching) .....	32
5.2.1	Componentes de MPLS.....	33

5.2.1.1	Dominio MPLS .....	33
5.2.1.2	LSR (Label Switched Router).....	33
5.2.1.3	LER (Label Edge Router) .....	34
5.2.1.4	FEC (Forwarding Equivalence Class) .....	34
5.2.1.5	LSP (Label Switched Path).....	35
5.2.1.6	Etiquetas MPLS.....	36
5.2.1.7	LDP (Label Distribution Protocol).....	37
5.2.2	Operación de MPLS .....	38
5.2.3	Ventajas de MPLS.....	39
5.3	Centro de Datos Virtual basado en el Modelo de Negocio IaaS .....	40
5.3.1	Arquitectura de Referencia de un Centro de Datos Físico .....	40
5.3.1.1	Data Center Switching .....	42
5.3.1.2	Switches Data Center Sistema Operativo .....	45
5.3.2	Arquitectura de Referencia de un Centro de Datos Virtual.....	48
5.4	Interconexión entre una red MPLS y el Servicio de Computación en la Nube Caso de Estudio Proveedor CNT E.P. ....	50
5.4.1	IGP (Interior Gateway Protocol).....	50
5.4.2	OSPF (Open Shortest Path First) .....	50
5.4.2.1	Áreas OSPF .....	51
6.	Estandarización de la configuración de servicios de conectividad en la Capa de Networking del proveedor de Servicios de Nube del caso de estudio.....	54

7.	Análisis de herramientas OPEN SOURCE que permiten el desarrollo de una solución de automatización prototipo para la configuración de servicios de conectividad de los Data Centers Virtuales.....	61
7.1	Herramientas de Código abierto (Open Source) .....	61
7.1.1	Lenguaje de programación Python Software Foundation .....	63
7.1.2	Base de datos MySQL.....	66
7.1.3	Administración de MySQL.....	68
7.1.4	Sistema Operativo GNU/Linux.....	69
8.	Implementación del programa de automatización.....	72
a.	Arquitectura General de la Solución.....	72
b.	Prototipo para automatizar la configuración de enlaces en el servicio de Computación en la Nube del CSP del caso de estudio .....	73
9.	Conclusiones y Recomendaciones.....	78
10.	Bibliografía.....	84
11.	Acrónimos .....	89
12.	Anexos .....	92

## ÍNDICE DE FIGURAS

Figura 1. Cadena de Valor del Servicio de Computación en la Nube caso CNT E.P. (El Autor, 2016).....	17
Figura 2. Arquitectura Data Center, servicio Cloud Computing (El Autor, 2016) .....	20
Figura 3. Computación en la Nube según la norma ISO/IEC 17788 (El Autor, 2016) .....	26
Figura 4. Modelo de Red MPLS (El Autor, 2016).....	36
Figura 5. Formato de una etiqueta MPLS (El Autor, 2016).....	36
Figura 6. Arquitectura con componentes vPC (Cisco, 2014).....	41
Figura 7. Topología Recomendada Centro de Datos (CISCO SYSTEMS, INC., 2015).....	44
Figura 8. Cuadrante Mágico para Networking Data Center (Gartner, Inc., 2016).....	45
Figura 9. Sistema Operativo NX-OS (Fast Lane Consulting and Education Services, Inc., 2014).....	46
Figura 10. Características de NX-OS (Fast Lane Consulting and Education Services, Inc., 2014).....	48
Figura 11. Arquitectura de Referencia Centro de Datos Virtual, IaaS (El Autor, 2016).....	49
Figura 12. Áreas OSPF (CISCO, 2015).....	52
Figura 13. Redes de Data Center Virtual (CNT E.P., 2016).....	60
Figura 14. Los mejores lenguajes de programación 2016 (IEEE Spectrum, 2016) .....	65
Figura 15. Clasificación de las 10 Bases de Datos más utilizadas (DB-Engines, 2016).....	67
Figura 16. Arquitectura General de la Solución (El Autor, 2016) .....	73
Figura 17. Pantalla de Inicio del Programa de Automatización (El Autor, 2016) .....	74
Figura 18. Creación de Enlaces (El Autor, 2016).....	75
Figura 19. Ingreso de Parámetros para la configuración de enlaces (El Autor, 2016) .....	75
Figura 20. Borrar un enlace (El Autor, 2016) .....	76
Figura 21. Consulta de un enlace (El Autor, 2016) .....	76

Figura 22. Opción Salir (El Autor, 2016) .....	77
Figura 23. Acceso al portal de Gestión del VDC (VMware, Inc.® VMware vCloud Director, 2016) (Corporación Nacional de Telecomunicaciones CNT EP, 2016).....	92
Figura 24. Creación de una vAPP (VMware, Inc.® VMware vCloud Director, 2016) .....	93
Figura 25. Creación de una máquina virtual (VMware, Inc.® VMware vCloud Director, 2016).....	94
Figura 26. Catálogos de medios VDC (VMware, Inc.® VMware vCloud Director, 2016)	95
Figura 27. Arranque del Instalador Sistema Operativo Ubuntu Server (Ubuntu Server, 2016).....	95
Figura 28. Instalación de LAMP en Ubuntu Server (Ubuntu Server, 2016).....	96
Figura 29. Configuración MySQL (Ubuntu Server, 2016).....	97
Figura 30. Instalación de VMware Tools Servidor Ubuntu (VMware, Inc.® VMware vCloud Director, 2016).....	97
Figura 31. Instalación y configuración de phpMyAdmin (phpMyAdmin, 2016).....	98
Figura 32. Portal de phpMyAdmin (phpMyAdmin, 2016).....	99
Figura 33. Instalación de Python (El Autor, 2016).....	99
Figura 34. Consola de programación de Python (El Autor, 2016).....	100
Figura 35. Instalación del módulo Paramiko en Python (El Autor, 2016).....	101
Figura 36. Instalación de la librería MySQLdb (El Autor, 2016) .....	102

## ÍNDICE DE TABLAS

Tabla 1. Tipos de LSA .....	53
Tabla 2. Descripción del rango de VLAN's.....	55
Tabla 3. Versiones de HSRP .....	57
Tabla 4. Configuración recomendada equipos Core para enlaces de Datos e Internet.....	60
Tabla 5. Configuración recomendada en los equipos Agregación o Distribución para enlaces de Datos e Internet. ....	60
Tabla 6. Distribuciones de Linux.....	71
Tabla 7. Componentes para desarrollar la Solución de Automatización de Enlaces.....	72
Tabla 8. Lista de Máquinas Virtuales implementadas en el VDC.....	94

## 1. Introducción

El presente trabajo presenta el análisis e implementación de una solución para automatizar la configuración de enlaces en un ambiente de Computación en la Nube provistos por una red de transporte MPLS (Multi-Protocol Label Switching), tomando como caso de estudio al proveedor de servicios de Computación en la Nube o Cloud Computing Corporación Nacional de Telecomunicaciones CNT E.P. con la finalidad de brindar eficiencia, optimizar la configuración y despliegue de enlaces que brindan servicios de conectividad en los Data Centers Virtuales que se encuentran implementados bajo el modelo de servicio IaaS (Infraestructura como Servicio).

Como producto de este análisis se obtendrá un entregable cuyo prototipo automatizará esta tarea y planteará la estandarización en la interconexión entre el servicio de Computación en la nube y la red de transporte MPLS (Multi-Protocol Label Switching), este entregable es un programa que contiene scripts desarrollados bajo herramientas y software Open Source (Código abierto) como Python Software Foundation (2016)<sup>1</sup>, motor de base de datos MySQL Oracle Corporation (2016)<sup>2</sup> y un administrador de base de datos phpMyAdmin (2016)<sup>3</sup>, bajo el sistema operativo GNU/Linux Ubuntu Server (2016)<sup>4</sup>.

Al final este análisis servirá como base para la adopción futura de SDN (Redes Definidas por Software) en el servicio de Cloud Computing que provee la CNT EP que ayudará a automatizar y simplificar la administración de la capa de red.

---

<sup>1</sup> Lenguaje de programación Python

<sup>2</sup> Base de datos MySQL

<sup>3</sup> Interfaz web para administrar bases de datos con compatibilidad para el motor MySQL

<sup>4</sup> Sistema Operativo en el que se desarrollará el programa es Ubuntu Server 16.04 LTS

## **2. Justificación**

En la actualidad existe una tendencia de mercado a la proliferación de los servicios operados en nube, estos servicios permiten asignación flexible y escalable de recursos en función de las necesidades del cliente o suscriptor, aportando una reducción de costos con respecto al modelo tradicional de adquisición de infraestructuras considerables.

Como caso de estudio se ha tomado como referencia la cadena de valor del servicio de Computación en la Nube del proveedor CNT EP, Figura 1, que al momento brinda este servicio de acuerdo a los siguientes pasos ejecutados por distintas áreas dentro de la empresa:

- Última Milla.- La instalación de enlaces que pueden ser de datos o internet en las premisas de los clientes consumidores del servicio de Cloud Computing.
- MPLS.- Configuración en la red de transporte MPLS que brinda el acceso a los clientes hacia Cloud Computing.
- Interconexión MPLS – Cloud Computing.- Interconexión entre los PE's (Provider Edge) de la Red de Transporte MPLS y los CE's (Customer Edge) Capa Core Cloud Computing
- Cloud Computing.- Servicio brindado por la CNT EP en la capa del modelo de negocio IaaS.

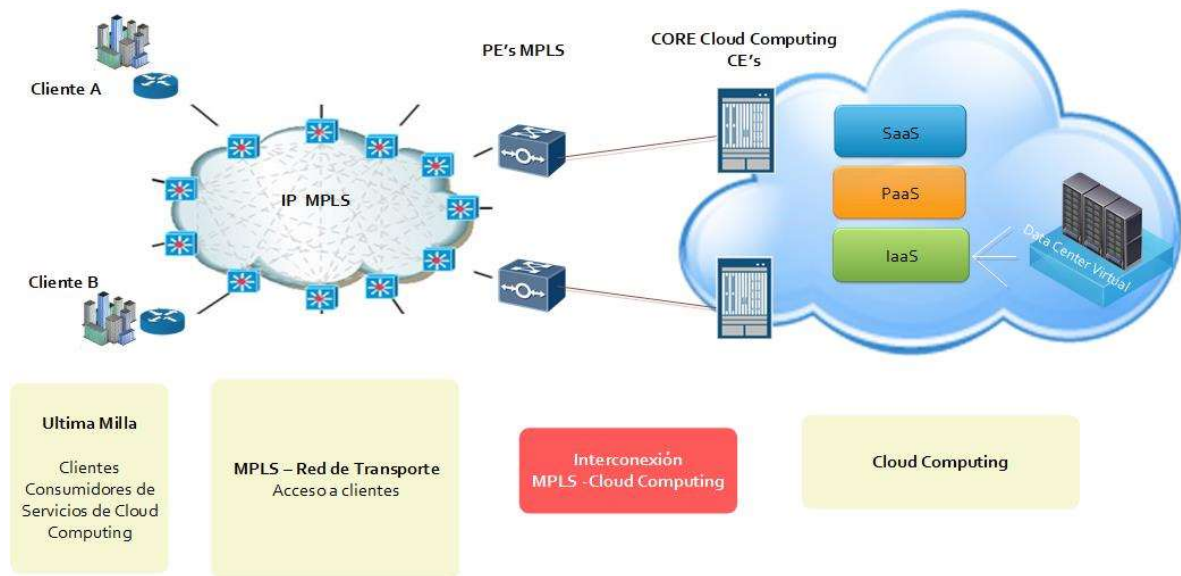


Figura 1. Cadena de Valor del Servicio de Computación en la Nube caso CNT E.P. (El Autor, 2016)

Es importante definir que el servicio de Cloud Computing de la CNT EP brinda servicios de conectividad a los Data Centers Virtuales de clientes finales bajo el modelo de negocio IaaS a través de una red de transporte MPLS, en esta interconexión se realizan configuraciones a nivel de CE's (Customer Edge Router) donde el servicio es configurado en el diseño de tres niveles de Data Center (Core, Agregación y Acceso).

Esta tarea actualmente es manual y se plantea en este proyecto entregar como caso de uso la automatización de dicha tarea, que simplificará el despliegue y la administración de los enlaces de datos e internet, dejando la forma tradicional de implementación brindando estandarización, reducción de tiempo, optimización, control y eficiencia para que el servicio de Cloud Computing se conecte a cualquier red de transporte MPLS propia de CNT o de cualquier proveedor y que sirva de base para la adopción de una evolución del ecosistema Cloud Computing hacia un modelo de auto aprovisionamiento del servicio donde con el proveedor de servicios de nube CNT EP, obtenga todos los beneficios de un nuevo modelo que incremente la competitividad, productividad y mejore la atención al cliente o suscriptor.

Es importante tomar en cuenta que el producto final de este proyecto es la entrega de la implementación de un programa desarrollado con herramientas “Open Source” para reducir los costos de CAPEX (inversión en bienes de capital) que no impliquen un gasto o inversión inicial en licenciamiento o derechos de uso de esta solución, para el proveedor del servicio de Cloud Computing y que sirvan como base para una adopción futura de la tecnología SDN.

### **3. Antecedentes**

Como afirma el Centro de estudios de telecomunicaciones de América Latina<sup>5</sup> (2014) “la Computación en la Nube está revolucionando la implementación de la infraestructura de las Tecnologías de la Información TI,” (...) (pág. 5) este modelo tecnológico brinda diferentes usos y servicios con eficiencia, flexibilidad y ahorro de recursos.

La Computación en la Nube tiene tres modelos de servicio: Software como Servicio (SaaS), Plataforma como Servicio (PaaS) e Infraestructura como Servicio (IaaS) en el cual la infraestructura es gestionada por el proveedor como un servicio bajo demanda y evita la compra de hardware y componentes físicos, ya que ofrece recursos (procesamiento, memoria y almacenamiento) al suscriptor accesible a través de una interfaz de usuario.

Esto le permite al suscriptor o cliente contar con su Data Center Virtual (VDC) donde puede aprovisionar y gestionar de manera fácil sus recursos de computación, redes, seguridades, monitoreo, almacenamiento y otros servicios, según las necesidades concretas de su negocio. Al mencionar redes, se requiere enlaces para tener conectividad hacia la gestión y publicación, de sus aplicativos y servicios.

Basados en estos antecedentes, CNT EP como proveedor de servicios de Computación en la Nube diseñó este servicio para los suscriptores o clientes finales, entendiendo la importancia de la conectividad hacia los Data Centers Virtuales, ya que contar con una red de datos y comunicaciones es fundamental para la expansión del modelo de negocio de Cloud Computing, tomando en cuenta la cadena de valor mencionada en el punto anterior los trabajos para interconectar la nube al mundo requieren de una red de transporte MPLS basados en una

---

<sup>5</sup> El Centro de Estudios de Telecomunicaciones de América Latina tiene por objetivo promover y apoyar la reflexión y el debate sobre las políticas públicas orientadas al desarrollo de la Sociedad de la Información en la región, contribuyendo con elementos de análisis técnicos y económicos, a su diseño, ejecución y evaluación.

arquitectura de diseño de Data Center (Core, Agregación y Acceso) propuesta por el proveedor de servicios Cloud de Computing que se describe a continuación:

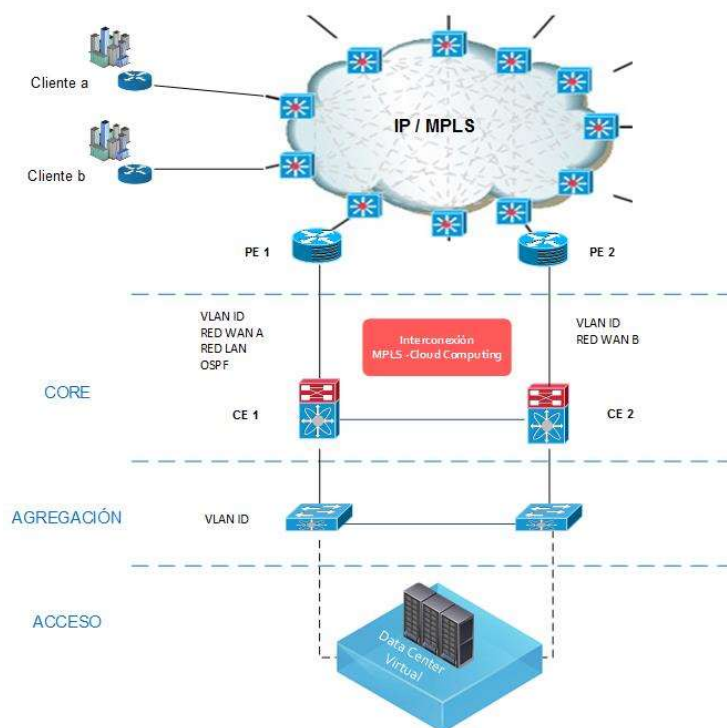


Figura 2. Arquitectura Data Center, servicio Cloud Computing (El Autor, 2016)

De acuerdo con Cisco (2015), MPLS (Multi-Protocol Label Switching) es una tecnología de reenvío de paquete que utiliza etiquetado para tomar decisiones de reenvío de datos, proporcionando una red de transporte que utiliza aplicaciones como: Virtual Private Networking (VPN)<sup>6</sup>, Ingeniería de tráfico (TE)<sup>7</sup>, Calidad del servicio (QoS)<sup>8</sup>, entre otras, de la cual el servicio de Cloud Computing es un cliente de esta red de transporte.

<sup>6</sup> VPN (Virtual Private Networking): Es una Red Privada Virtual establecida sobre un medio que está compartido, esto permite extender una red estableciendo una conexión virtual punto a punto utilizando conexiones dedicadas, cifrado o ambos métodos.

<sup>7</sup> Ingeniería de tráfico (TE): Adapta flujos de tráfico de acuerdo a los recursos físicos de la red, buscando que no exista recursos excesivamente utilizados basados en la demanda y naturaleza de los servicios. Brinda optimización, rendimiento y control del tráfico que circula por una red operativa.

<sup>8</sup> Calidad de Servicio (QoS): Proporciona la capacidad de administrar el tráfico de red determinando el grado de satisfacción de un usuario del servicio.

La asignación de recursos para el despliegue de los enlaces en la capa Core del servicio de Cloud Computing, son provistos por la Red MPLS, para que los mismos sean interconectados hacia los Data Centers Virtuales de los clientes finales, los parámetros que se requieren son:

- Enlace configurado con el protocolo de enrutamiento dinámico OSPF
- VLAN ID
- Direccionamiento para la Red WAN
- Direccionamiento para la Red LAN

Con estos parámetros se procede a configurar en la Capa de Networking del servicio de Cloud Computing de acuerdo a la arquitectura mencionada:

- VRF Lite
- Protocolo de enrutamiento dinámico OSPF
- VLAN y SVI (Switch virtual interface)
- Protocolo de redundancia HSRP
- Direccionamiento WAN y LAN en sus interfaces respectivas

Con este procedimiento se brinda conectividad para el servicio de IaaS de Cloud Computing, al ser una configuración manual, se presenta una solución a este caso, donde se determina una oportunidad de mejora para implementar scripts que automaticen el proceso, incrementando la eficiencia, brindará una administración simple donde se reducen los errores, existirá una mejora en el soporte y resolución de problemas, mejorando la atención y aprovisionamiento al cliente final o suscriptor.

## **4. Objetivos**

### **Objetivo General:**

Analizar e implementar un prototipo de una solución que automatice la configuración de enlaces que brindan conectividad en un ambiente de Computación en la Nube, basados en una arquitectura de configuración estándar, caso de estudio de CNT EP, entregando como producto final un programa que brinda una mayor eficiencia en la administración y gestión de los recursos en la capa de Red de los servicios de Nube.

### **Objetivos Específicos:**

1. Analizar las tecnologías que utiliza el proveedor de servicios de Computación en la Nube CNT EP para brindar servicios de conectividad a los Data Centers Virtuales de clientes finales en el Modelo de Negocio IaaS, basados en su arquitectura actual.
2. Estandarizar la configuración de los servicios de conectividad brindados a los Data Centers Virtuales a nivel de la capa de Networking, con el fin de que el proveedor de Servicios en la Nube CNT EP pueda interconectarse hacia la red de transportes MPLS, basados en su arquitectura actual.
3. Analizar herramientas Open Source que permitan desarrollar una solución de automatización prototipo para la configuración de los servicios de conectividad brindados a los Data Centers Virtuales a nivel de la capa de Networking, este desarrollo permitirá crear un entregable que es un programa que no representará gastos de inversión para el proveedor de solución de Cloud Computing y que estará basado en la arquitectura actual.

4. Implementar un programa de automatización para facilitar: la configuración de enlaces, reduciendo fallas por la ejecución manual en el proceso de configuración, mejorando los tiempos de respuesta a los requerimientos de clientes finales, brindando eficiencia y control a la administración en la capa de Red de la arquitectura del servicio de Computación en la nube que brinda el proveedor de servicios del caso de estudio.

## **5. Análisis de tecnologías utilizadas por el proveedor del caso de estudio para brindar servicios de conectividad a Data Centers Virtuales**

A continuación se realiza el análisis de las tecnologías que utiliza el proveedor de servicios de Computación en la Nube del caso de estudio CNT EP, para brindar servicios de conectividad a los Data Centers virtuales, basados su cadena de valor de servicios y arquitectura actual, en este proceso intervienen: el Modelo Tecnológico de Computación en la Nube, la Red de Transporte MPLS, Centro de Datos Virtual basado en el Modelo de Negocio IaaS, interconexión entre la Red MPLS y el servicio de Computación en la Nube:

### **5.1 Modelo Tecnológico de Computación en la Nube**

Basado en los antecedentes y según ITpreneurs (2015) Computación en la Nube o Cloud Computing es un modelo para organizar recursos y entrega de servicios TI brindando flexibilidad para las empresas y usuarios finales. Esta flexibilidad incluye:

- Aprovisionamiento de los recursos TI
- Tener el control y acceso a los recursos TI
- ¿Cómo, cuándo, dónde y qué dispositivos? pueden acceder a los recursos TI
- Costo y propiedad de los recursos de TI

Existen varias definiciones y perspectivas sobre Computación en la Nube de las cuales se tomó como referencia tres principales:

- La del Instituto Nacional de Estándares y Tecnología (NIST)<sup>9</sup> (2011) que cita:

---

<sup>9</sup> NIST es el Instituto Nacional de Mediciones de Estados Unidos, su misión es desarrollar y promover mediciones, normas y tecnología para incrementar la productividad, facilitar el comercio y mejorar la calidad de vida.

Computación en la Nube es un modelo tecnológico que permite el acceso ubicuo, adaptado y bajo demanda en red a un conjunto compartido de recursos de computación configurables y que pueden ser rápidamente aprovisionados y liberados con un esfuerzo de gestión reducido o interacción mínima con el proveedor del servicio (...). Este modelo de nube se compone de tres modelos de servicio: Software como Servicio (SaaS), Plataforma como Servicio (PaaS) e Infraestructura como Servicio (IaaS). (pág. 6)

- La Organización Internacional para la Estandarización y la Comisión Electrotécnica Internacional ISO/IEC 17788 <sup>10</sup> (2014) que define a la computación en nube como “un paradigma que permite el acceso a través de la red a un conjunto de recursos físicos o virtuales compartidos, escalables y elásticos con autoservicio de aprovisionamiento y administración bajo demanda.”
- Y a Gartner, Inc.<sup>11</sup> (2016) que define a la Computación en la nube como un estilo de computación que habilita de manera escalable y elástica capacidades de TI que se entregan como un servicio utilizando la tecnología de Internet.

Basados en el estándar de norma ISO/IEC 17788, (2014) se define características claves, roles, actividades, tipos de capacidades, categorías de servicios, y modelos de implementación de Computación en la nube o Cloud Computing, que se resumen en la figura 3:

---

<sup>10</sup> ISO/IEC 17788, serie normalizada y publicada por las organizaciones ISO e IEC que es el estándar reconocido internacionalmente para proporcionar una visión general de Computación en la Nube junto con un conjunto de términos y definiciones.

<sup>11</sup> Gartner Inc. empresa que realiza investigación, análisis y consultoría para las industrias de hardware, software, comunicaciones y de TI relacionadas.

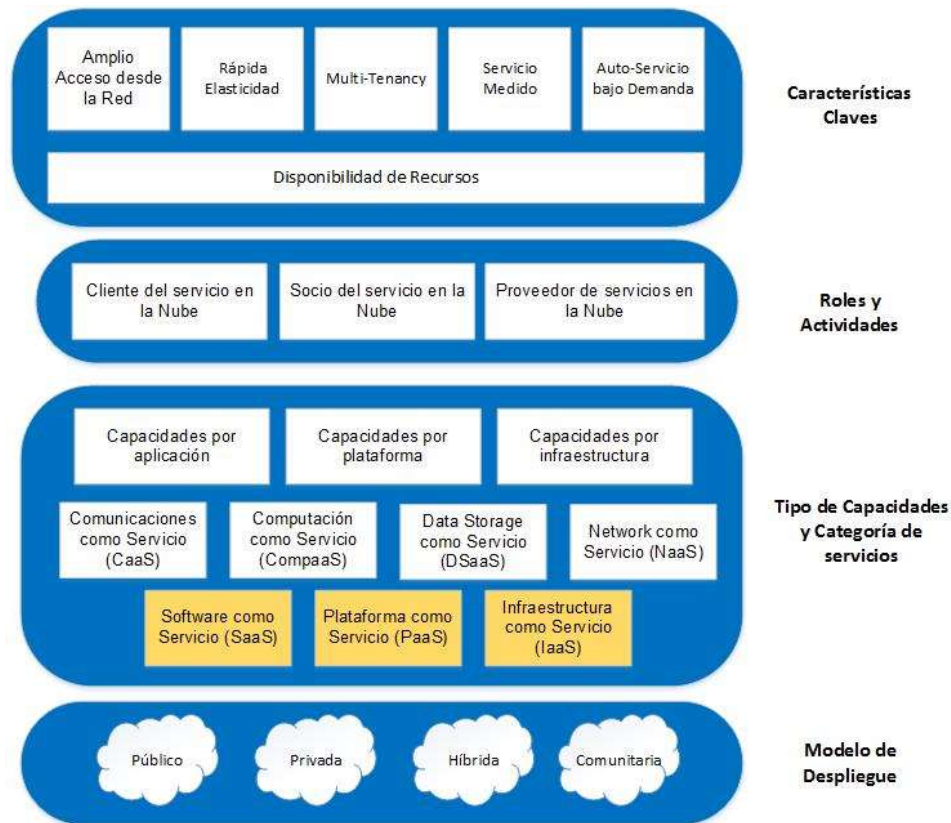


Figura 3. Computación en la Nube según la norma ISO/IEC 17788 (El Autor, 2016)

## 5.1.1 Características Claves

### 5.1.1.1 Auto Servicio bajo demanda

El consumidor está en control, los servicios están disponibles bajo demanda y pueden ser aprovisionados por el usuario final, no se requiere funciones de soporte de TI del proveedor para acceder a los servicios.

### 5.1.1.2 Amplio acceso desde la red

Las capacidades están disponibles en la red y pueden ser accedidas a través de mecanismos estándares como Internet, utilizando diversos dispositivos como por ejemplo, teléfonos móviles, tabletas, portátiles y estaciones de trabajo.

### **5.1.1.3 Disponibilidad de Recursos**

Los recursos informáticos que brinda el proveedor se combinaron para servir a múltiples consumidores utilizando un modelo Multi-propietario o Multi-Tenant<sup>12</sup>, con diferentes recursos físicos y virtuales asignados dinámicamente de acuerdo a la demanda del consumidor, sin importar la ubicación exacta de los recursos proporcionados, la disponibilidad de recursos incluyen el almacenamiento, procesamiento, memoria y ancho de banda de la red.

### **5.1.1.4 Multi-Propietario (Multi-Tenancy)**

Es una característica donde se asignan recursos físicos o virtuales para que múltiples usuarios y sus datos estén aislados e inaccesibles uno del otro en un mismo servicio brindado por un proveedor.

### **5.1.1.5 Rápida elasticidad**

Las capacidades pueden ser rápidamente aprovisionadas o liberadas de forma elástica y automática de acuerdo a la demanda. Para el consumidor las capacidades disponibles parecen ser ilimitadas y pueden ser requeridas en cualquier cantidad y en cualquier momento.

### **5.1.1.6 Servicio Medido**

El uso de recursos puede ser monitorizado, controlado y medido a un nivel de abstracción adecuado al tipo de servicio (cuentas de usuario, almacenamiento, procesamiento, ancho de banda, etc.) proporcionando transparencia, tanto para el proveedor como al consumidor por el servicio utilizado.

---

<sup>12</sup> Multi-tenant: Son arquitecturas multi-propietario en el que todos los clientes consumen un servicio desde la misma plataforma tecnológica.

## **5.1.2 Roles y Actividades**

Todas las actividades relacionadas con Computación en la Nube se pueden clasificar en tres grupos principales: actividades que utilizan los servicios, actividades que proveen el servicio y actividades que brindan soporte a los servicios.

Las principales funciones de Computación en la Nube son:

### **5.1.2.1 Cliente del servicio en la Nube**

Es el cliente final que puede ser una persona natural o jurídica que está relacionada con los propósitos del negocio con el fin de utilizar los servicios de nube, la relación comercial es con un proveedor de servicios en la nube. Las actividades claves para un cliente de servicio en la Nube incluyen pero no se limitan a: uso y administración de servicios en Nube y ejecución de administración del negocio.

### **5.1.2.2 Socio del servicio en la Nube**

Es un tercero que brinda soporte a un proveedor de servicios de Nube o al cliente que consume el servicio de Nube, el trabajo que realiza un socio del servicio de Nube puede ser como auditor de nube<sup>13</sup> o como agente de servicios de Nube<sup>14</sup>.

### **5.1.2.3 Proveedor de servicios en la Nube**

Es un tercero que permite que los servicios de nube estén siempre disponibles, el proveedor de servicios de Nube se centra en realizar todas las actividades necesarias para proporcionar, asegurar y mantener la entrega de servicios de Nube al cliente, estas actividades pueden incluir por ejemplo: proporcionar, implementar, supervisar el servicio, administrar un plan de negocio, brindar datos de auditoría, etc. También puede contar con

---

<sup>13</sup> Auditor de Nube (Cloud auditor), tiene la responsabilidad de llevar a cabo una auditoría de la provisión y uso de servicios de Nube.

<sup>14</sup> Agente de servicios de nube (Cloud service broker) es el que negocia las relaciones entre los clientes del servicio en la Nube y los Proveedores de servicios de Nube.

sub-roles como: Área Comercial, de servicios, proveedores de Red, seguridad y manejo de riesgos, etc.

### **5.1.3 Tipo de capacidades y Categoría de servicios**

Capacidad es una clasificación de funcionalidad proporcionada por un servicio de Nube al cliente, basados en los recursos utilizados. Hay tres tipos de capacidades de nube: de aplicaciones, infraestructura y de plataforma.

Los tipos de capacidades de Nube son:

#### **5.1.3.1 Capacidades por Aplicación**

El cliente del servicio en la Nube puede utilizar el servicio de aplicaciones en la nube del proveedor de servicio.

#### **5.1.3.2 Capacidades por Infraestructura**

El cliente del servicio en la Nube puede usar y aprovisionarse de procesamiento, almacenamiento o recursos de redes.

#### **5.1.3.3 Capacidades por Plataforma**

El cliente del servicio en la Nube puede implementar, gestionar y ejecutar aplicaciones creadas o adquiridas por el mismo, utilizando uno o más lenguajes de programación y uno o más entornos de ejecución soportados por el Proveedor de Servicios en la nube.

Las capacidades son definidas por la ISO/IEC 17788 (2014) y no deben confundirse con otras categorizaciones de servicios en la nube.

Una categoría de servicio en la nube en cambio es un grupo de servicios que poseen un conjunto de cualidades comunes. Una categoría puede incluir capacidades de uno o más tipos.

Las categorías de servicios de nube más representativos son:

#### **5.1.3.4 Comunicaciones como servicio (CaaS)**

Es una categoría de servicio en la Nube que tiene la capacidad de proporcionar al consumidor, colaboración e interacción en tiempo real.

#### **5.1.3.5 Computación como servicio (CompaaS)**

Categoría de servicio en la Nube en la que las capacidades proporcionadas al consumidor son el aprovisionamiento y uso de recursos de procesamiento necesarios para implementar y ejecutar software.

#### **5.1.3.6 Data Storage como servicio (DSaaS)**

En esta categoría de servicio en la Nube la capacidad proporcionada al consumidor es el aprovisionamiento y uso de almacenamiento de datos y capacidades relacionadas.

#### **5.1.3.7 Network como servicio (NaaS)**

Categoría de servicio en la Nube en la que la capacidad proporcionada al consumidor es la conectividad, transporte y capacidades de red.

#### **5.1.3.8 Software como Servicio (SaaS)**

En esta categoría el consumidor utiliza las aplicaciones del proveedor que se ejecutan en una infraestructura de nube. Las aplicaciones pueden ser utilizadas desde varios dispositivos del cliente a través de una interfaz como un navegador web u otro programa. El consumidor no gestiona ni controla la infraestructura de nube subyacente incluyendo la red, servidores, sistemas operativos, almacenamiento, o incluso capacidades de aplicaciones individuales, con excepción de valores de configuración de aplicaciones específicas de usuario limitadas.

#### **5.1.3.9 Plataforma como Servicio (PaaS)**

En PaaS, el consumidor está en capacidad de desplegar en la infraestructura de nube del proveedor aplicaciones creadas por el mismo o adquiridas utilizando lenguajes de

programación, bibliotecas, servicios y herramientas admitidas por el proveedor. El consumidor o cliente no administra o controla la infraestructura de nube subyacente incluyendo la red, servidores, sistemas operativos, almacenamiento, pero tiene control sobre las aplicaciones desplegadas y ajustes de configuración en el entorno de aplicaciones alojadas.

#### **5.1.3.10 Infraestructura como Servicio (IaaS)**

Permite al consumidor o cliente aprovisionar y disponer de recursos informáticos tales como procesamiento, memoria, almacenamiento, redes y otros donde es capaz de desplegar y ejecutar cualquier tipo de software que puede incluir sistemas operativos y aplicaciones. El consumidor no gestiona ni controla la infraestructura de nube subyacente, pero tiene el control de los sistemas operativos, almacenamiento, aplicaciones desplegadas, y componentes de red seleccionados como firewalls, routers, balanceadores entre otros.

#### **5.1.4 Modelos de Despliegue**

Los modelos de despliegue representan cómo la Computación en la Nube puede ser organizada basados en el control y distribución de los recursos físicos o virtuales.

##### **5.1.4.1 Nube Privada**

La infraestructura de la nube es solo para el uso exclusivo de una sola organización, la administración y operación puede ser de propiedad de la organización, un tercero, o alguna combinación de ellos y está implementada dentro o fuera de sus premisas.

##### **5.1.4.2 Nube Comunitaria**

La infraestructura de esta nube está disponible para el uso exclusivo de una comunidad específica, de consumidores que comparten preocupaciones en particular como por ejemplo, la misión, requisitos de seguridad, políticas o cumplimientos. Puede estar bajo

administración y operación de: una o más organizaciones de una comunidad, un tercero, o alguna combinación de ellos, se encuentra implementada dentro o fuera de sus premisas.

#### **5.1.4.3 Nube Pública**

La infraestructura de esta nube está disponible para uso del público en general, puede estar bajo la administración y operación de una empresa, organización gubernamental, académica, o alguna combinación de las mismas. Se encuentra implementada en las instalaciones del proveedor de nube.

#### **5.1.4.4 Nube Híbrida**

La infraestructura de esta nube está compuesta de dos o más infraestructuras de nube que pueden ser privadas, comunitarias o públicas que permanecen como entidades únicas, pero unidas por tecnología estándar o propietaria, permitiendo la portabilidad de aplicaciones y datos.

### **5.2 Conmutación Multiprotocolo basado en Etiquetas, MPLS (Multiprotocol Label Switching)**

Es una tecnología de reenvío de paquetes creada por la IETF<sup>15</sup> y que se encuentra definida en el RFC<sup>16</sup> 3031, que proporciona una designación eficiente de etiquetas a los paquetes, enrutamiento, envío y conmutación de flujos de tráfico a través de la red, que trabaja en las capas 2 y 3 del Modelo de Referencia OSI<sup>17</sup>. La IETF organizó el grupo de trabajo MPLS en 1997, donde actualmente es un estándar de la industria.

---

<sup>15</sup> IETF (Internet Engineering Task Force): Grupo de trabajo que viene de fabricantes, proveedores de servicio, investigadores, profesores, estudiantes y otros, que definen los estándares de Internet.

<sup>16</sup> RFC (Request for Comments): Es un documento formal, específico y detallado de protocolos, procedimientos y eventos de la IETF, estos documentos son actualizados continuamente y tiene un número asignado.

<sup>17</sup> OSI (Open System Interconnection): Modelo de referencia para los protocolos de red basado en una arquitectura de capas.

Las etiquetas de MPLS se anuncian entre los routers para que puedan construir un mapeo de etiqueta a etiqueta, estas etiquetas están asociadas a los paquetes IP, lo que permite a los routers reenviar el tráfico tomando en cuenta la etiqueta y no la dirección IP de destino. Los paquetes se envían por conmutación de etiquetas en lugar de conmutación IP. (Ghein, 2007, p.30).

## **5.2.1 Componentes de MPLS**

### **5.2.1.1 Dominio MPLS**

Es un conjunto de nodos adyacentes que conmutan etiquetas y que pertenecen a un dominio o se encuentran bajo una misma administración.

Tomando como referencia a The Internet Society<sup>18</sup> (2001) e Hidalgo (2013), los routers o nodos de MPLS pueden tener las siguientes características:

### **5.2.1.2 LSR (Label Switched Router)**

Router Conmutador de etiquetas es un equipo de alta velocidad en el core de una Red MPLS especializado en el envío de paquetes etiquetados, también llamados P Routers (Provider Routers) ya que le pertenecen al proveedor de servicios. Cuando los paquetes llegan a los LSR's estos intercambian las etiquetas existentes por otras y envían el paquete al siguiente LSR y así sucesivamente logrando la distribución de etiquetas.

Existen tres clases de LSR's en una red MPLS:

- LSR's de Ingreso (Ingress LSR's).- Recibe un paquete que no está etiquetado e inserta una etiqueta delante del paquete (pila) y lo envía por el enlace de datos.

Esta operación es conocida como Push (encapsula).

---

<sup>18</sup> Internet Society (ISOC): Organización no gubernamental y sin fines de lucro dedicada a asegurar que Internet siga siendo abierta, transparente y definida. Apoya a la IETF para la elaboración de los Request for Comments (RFC editor).

- LSR's de salida (Egress LSR's).- Recibe paquetes etiquetados, elimina la etiqueta y reenvía el paquete por un enlace de datos. Esta operación es conocida como Pop (desencapsula).

Los LSR's de entrada y salida pueden ser LSR's de borde (Edge LSR's) es decir pueden estar entre una Red MPLS y una Red no MPLS.

- LSR's Intermedios (Intermediate LSRs).- Reciben un paquete etiquetado entrante y este realiza una operación de cambio de la etiqueta inicial por otra, para que se realice el reenvío por el enlace de datos correcto. Esta operación es conocida como Swap.

### **5.2.1.3 LER (Label Edge Router)**

Router de Etiquetas de borde, son equipos situados en la frontera o borde de una red MPLS, que desempeñan funciones de enrutamiento y conectividad siendo capaces de conectarse a diversas redes, tales como Ethernet<sup>19</sup>. Se los conoce también como PE Routers (Provider Edge Routers) y se encargan de insertar y extraer las etiquetas al ingreso o la salida de la red MPLS, basándose en una conmutación conocida como FEC (Forwarding Equivalence Class).

### **5.2.1.4 FEC (Forwarding Equivalence Class)**

FEC o Clase Equivalente de Envío, representa un grupo de paquetes que comparten los mismos requerimientos para su transporte que pueden ser: el mismo servicio, el mismo destino, la misma VPN, etc. Este grupo de paquetes entran a la red MPLS por la misma interfaz recibiendo la misma etiqueta y circulando por un mismo trayecto donde

---

<sup>19</sup> Ethernet: Protocolo de red estándar utilizado en redes de área local, según Tanenbaum (2012) existen dos tipos de Ethernet la clásica que resuelve el problema de acceso múltiple al medio, y la conmutada que utiliza switches para conectar diferentes dispositivos operando a 100, 1000 y 10 000 Mbps, actualmente en la práctica sólo se utiliza la Ethernet conmutada.

se asigna a cada paquete un FEC, los paquetes de un mismo flujo de datos pertenecen a una misma FEC que es representada por una etiqueta de cada LSP (Label Switched Path), importante mencionar que una FEC puede agrupar varios flujos, pero un mismo flujo no puede pertenecer a más de una FEC al mismo tiempo.

#### **5.2.1.5 LSP (Label Switched Path)**

LSP o Rutas Conmutadas por Etiquetas es un camino virtual específico unidireccional establecido antes que la transmisión de datos comience a través de la red MPLS, esta transmisión la siguen los paquetes de una misma conexión y se asigna una etiqueta en cada salto que puede ser establecida por protocolos de enrutamiento o de forma manual. Cuando recibe un paquete el LER verifica el FEC al que pertenece y lo encamina por el LSP correspondiente. LSP permite una rápida conmutación debido a que las etiquetas son insertadas al inicio del paquete y son de longitud fija determinando que la conmutación se realice vía hardware.

MPLS cuenta con 2 opciones para seleccionar un LSP:

- Enrutamiento hop by hop.- Cada LSR escoge el próximo salto para una FEC, haciendo uso de un protocolo de enrutamiento como OSPF, ofreciendo una rápida conmutación y tratamiento de paquetes de diferentes FEC.
- Enrutamiento explícito (ER-LSP).- Un LSR de ingreso define la ruta completa para los paquetes de un FEC y elabora una lista de los LSR que esta cruza. A lo largo del camino los recursos pueden ser reservados para asegurar QoS al tráfico de datos, facilitando Ingeniería de Tráfico a través de la red brindando servicios diferenciados con flujos basados en políticas de gestión de red.

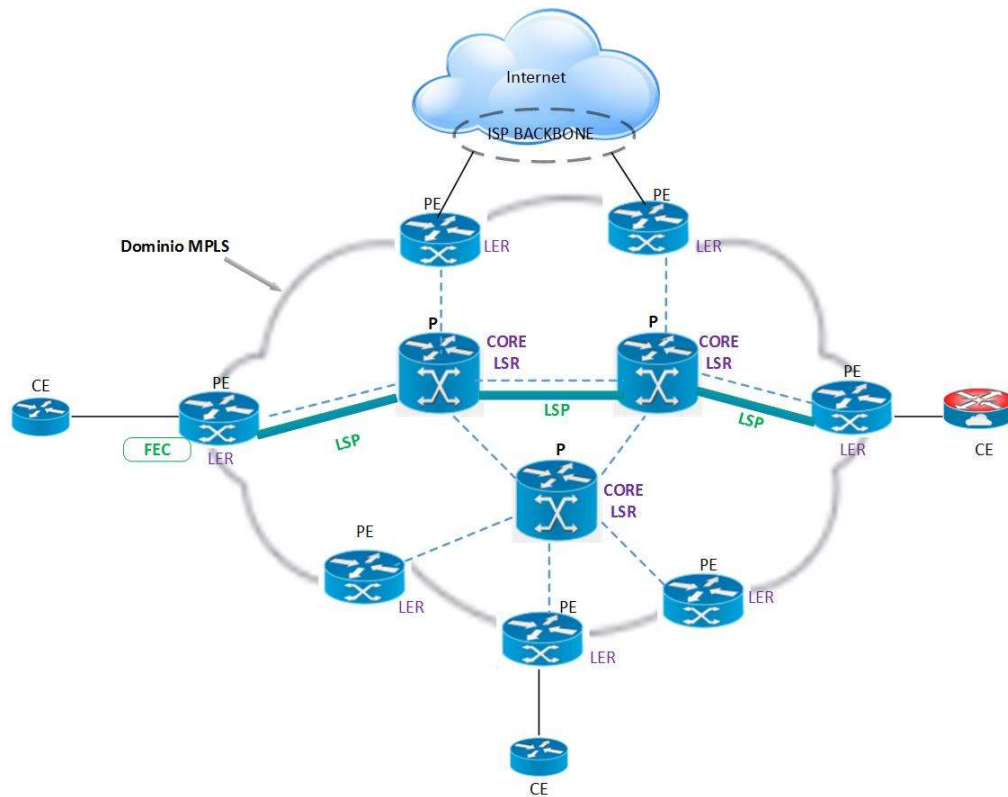


Figura 4. Modelo de Red MPLS (El Autor, 2016)

### 5.2.1.6 Etiquetas MPLS

De acuerdo con Ghein (2007, p.50) y Cisco (2015), una etiqueta MPLS tiene 32 bits, los campos de la cabecera MPLS de 4 bytes son:

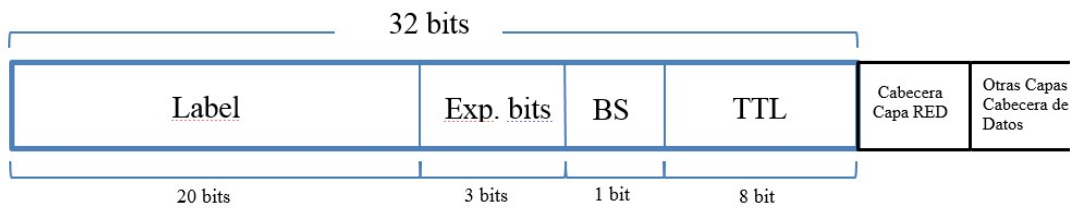


Figura 5. Formato de una etiqueta MPLS (El Autor, 2016)

**Escritura de etiqueta (Label).**- Los primeros 20 bits son para el valor de etiqueta.

**Exp.-** Uso experimental, 3 bits, se utiliza como campo de Clase de Servicio (CoS)<sup>20</sup>.

**BS (Bottom of Stack).**- Parte inferior de la pila, 1 bit, es cero a menos que la etiqueta esté en la parte inferior de la pila, si es así el valor es uno. La pila es una colección de etiquetas que están en la parte superior del paquete, esta puede constar de una o más etiquetas, aunque rara vez existen pilas que tengan cuatro o más etiquetas.

**TTL (Time to Live).**- Son 8 bits utilizados para TTL<sup>21</sup>, tiene la misma función del TTL de una cabecera IP, se reduce en 1 en cada salto y su función principal es evitar que un paquete pueda caer en un bucle de enrutamiento, cuando la etiqueta del TTL llega a 0, el paquete es descartado.

Las etiquetas en MPLS tienen significado local, siendo un identificador corto donde el router que recibe el paquete determina la etiqueta, para que no existan problemas cuando al mismo le lleguen paquetes que pertenecen a distintas FEC con la misma etiqueta.

Las etiquetas no tienen una dirección IP como en el caso de una cabecera de un paquete IP, más bien tienen un valor numérico entre dos nodos que están conectados a través de un LSP. Estas etiquetas están asociadas a un FEC por un flujo de datos o por control de tráfico.

### 5.2.1.7 LDP (Label Distribution Protocol)

El Protocolo de Distribución de Etiquetas es un conjunto de procedimientos y mensajes mediante el cual un LSR informa a su vecino las asociaciones de etiquetas a FEC's realizadas para que se establezca un LSP. Los tipos de mensajes de LDP son:

---

<sup>20</sup> Clase de Servicio (CoS): Es una técnica utilizada para diferenciar y gestionar el tráfico de una red, que consiste en agrupar tipos similares de tráfico y tratar cada tipo como una clase con su nivel de prioridad de servicio. En MPLS este campo permite seleccionar diferentes clases de servicio para paquetes que tengan una misma etiqueta.

<sup>21</sup> Time to Live (TTL): Tiempo de vida, indica por cuantos nodos o dispositivos puede pasar un paquete antes de ser descartado o devuelto a su origen.

- Descubrimiento (Discovery).- Anuncia y mantiene la presencia de un LSR en la red.
- Sesión (Session).- Establece, mantiene y termina sesiones entre LDP pares.
- Anuncio (Advertisement).- Crea, cambia y borra mapeo de etiquetas para FEC.
- Notificación (Notification).- Brinda información de avisos e indicaciones de errores.

### 5.2.2 Operación de MPLS

De acuerdo con González & Teng (2015), los siguientes pasos deben tomar los paquetes de datos para viajar a través de un dominio MPLS:

- Creación y distribución de una etiqueta.- Antes que el tráfico inicie, los routers asocian una etiqueta a un FEC y construyen sus tablas, mediante LDP los routers inician la distribución de etiquetas y asocian etiqueta/FEC así como negocian tráfico y capacidades MPLS gracias a LDP.
- Creación de tablas en cada ruteador.- Una vez que recibe las etiqueta asociadas, cada LSR crea la entradas en una Base de Información de Etiquetas conocida como LIB (Label Information Base) que es una tabla que almacena etiquetas asignadas por el router local y asignaciones de etiquetas de los vecinos. El contenido de la tabla especifica la asociación entre la etiqueta y un FEC (asociación entre la etiqueta de entrada de la tabla y el puerto de entrada a la etiqueta de salida de la tabla y el puerto de salida). Las entradas de la tabla son actualizadas cuando se renegocia la asociación de etiquetas y el FEC.
- Creación de una Ruta de Conmutación de Etiquetas (LSP).- Los LSP's son creados en dirección inversa a la creación de entradas en el LIBs.
- Inserción de Etiquetas y acceso en tablas.- El primer router LER usa la tabla LIB para determinar el siguiente salto y solicita una etiqueta para un FEC específico, los

routers contiguos solo usan la etiqueta para encontrar el próximo salto, una vez que el paquete alcanza el LSR de salida, la etiqueta es removida y el paquete es entregado al destino.

- Reenvío de paquetes.- El primer router LER puede no tener ninguna etiqueta para este paquete por ser la primera solicitud de este requerimiento, en una red IP el primer router LER encontrará la mejor dirección que coincida en sus tablas de ruteo para definir el próximo salto, LSR será el próximo salto para el primer router LER, donde este iniciará el requerimiento de etiquetas hacia LSR, este requerimiento se propagará en el dominio MPLS.

Cada router intermedio en el dominio MPLS recibirá una etiqueta desde el router subsiguiente donde se examinará la etiqueta en el paquete recibido y lo reemplazarán con la etiqueta de salida donde será reenviado, cuando llegue el paquete al router de borde PE, este removerá la etiqueta porque el paquete está dejando el dominio MPLS y será entregado al destino.

### **5.2.3 Ventajas de MPLS**

De acuerdo con Ghein (2007), las ventajas que brinda MPLS son:

- El uso de una infraestructura de red unificada.
- Mejor desempeño de reenvío de paquetes en la red.
- Soporta aplicaciones como: QoS (Calidad de servicio), CoS (Clase de servicio) e Ingeniería de Tráfico (TE) para servicios diferenciados, así como la creación de túneles VPN que permiten aislar el tráfico, como referencia de implementación sobre MPLS, se tiene la RFC 2547 (BGP/MPLS VPN's).
- Escalabilidad de la red.

- Importante mencionar que se debe contar con nodos robustos que puedan procesar y entender MPLS para aprovechar todos los beneficios de esta tecnología.

### **5.3 Centro de Datos Virtual basado en el Modelo de Negocio IaaS**

Un Centro de Datos Virtual o Virtual Data Center (VDC), es una solución basada en el Modelo de Negocio de Infraestructura como Servicio (IaaS), escalable, automatizada que brinda procesamiento, almacenamiento, memoria y recursos de red bajo demanda donde el Proveedor de Servicios ha implementado herramientas de gestión y acceso para que el cliente final pueda acceder a la administración de sus recursos.

Esta solución es rentable y fácil de provisionar está basada en los Centros de Datos físicos evitando la adquisición, gestión y operación de una infraestructura física propietaria.

#### **5.3.1 Arquitectura de Referencia de un Centro de Datos Físico**

Basados en el caso de estudio que está basado en una arquitectura referida por CISCO (2010), importante mencionar que la mayoría de los Centros de Datos son el resultado de un proceso de consolidación gracias a la facilidad y disponibilidad brindada por el ancho de banda para conectar ubicaciones remotas a Centro de Datos centralizados. Durante este proceso de consolidación, las empresas y proveedores de servicios simplificaron el uso del Networking e infraestructura de almacenamiento adoptando topologías que aprovechan la virtualización, VLAN's y tecnología de redes de almacenamiento virtual VSAN<sup>22</sup>, donde se han logrado solventar los siguientes requerimientos en el diseño:

- Mayor nivel de fiabilidad al tener infraestructuras redundantes reduciendo el tiempo de inactividad en actualizaciones y cambios de configuración.
- Optimización del uso de la infraestructura de red en los Centros de Datos hacia una topología que tiene todos sus enlaces activos, esto no ocurría en topologías tradicionales

---

<sup>22</sup> VSAN: Una solución de almacenamiento virtual que es compartido.

basadas en Spanning Tree Protocol (STP)<sup>23</sup>, que son conocidas por ser ineficientes debido al bloqueo de enlaces y alto tiempo de convergencia, para esto se recomienda el uso de tecnologías de trayecto múltiple de Capa 2 conocidas como Virtual PortChannels (vPCs). vPC permite que los enlaces que están físicamente conectados a dos switches diferentes, aparenten ser un simple dispositivo que soporte IEEE 802.3ad como parte de un único PortChannel<sup>24</sup>, estos dos equipos forman lo que se conoce como un dominio vPC. Los Centros de Datos basados en esta tecnología construyen topologías redundantes libres de lazos.

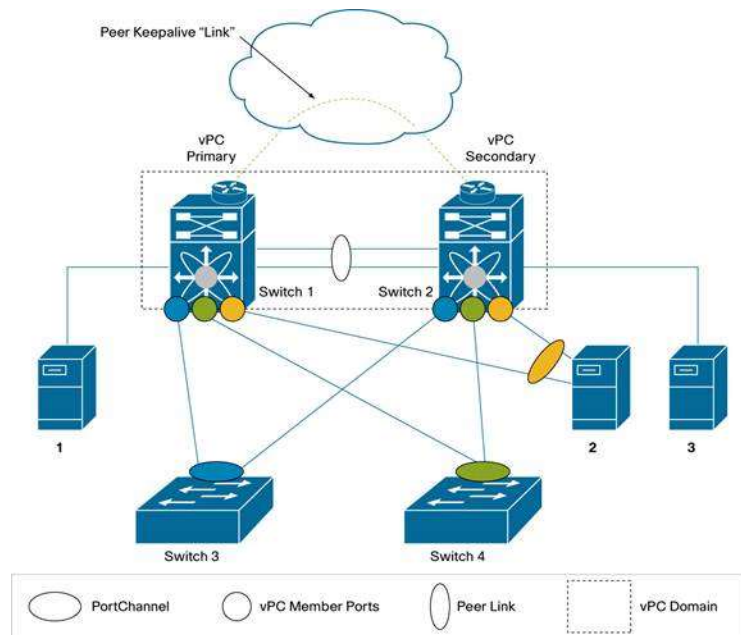


Figura 6. Arquitectura con componentes vPC (Cisco, 2014)

Los enlaces que se mencionan en la Figura 6 son: el vPC Peer Link que se utiliza para emular un único plano de control mediante el reenvío de paquetes desde el switch vPC secundario hacia el switch vPC primario. Para que un vPC envíe una VLAN, esa VLAN debe existir en ambos switches y debe estar permitida en el puerto troncal del switch para el propio vPC. Si no se cumple ninguna de estas condiciones, la VLAN no se muestra ya que no está permitida.

<sup>23</sup> Spanning Tree Protocol (STP): Es un protocolo capa 2 IEEE 802.1D, que impide la creación de bucles en enlaces redundantes en una red.

<sup>24</sup> PortChannel: Tecnología orientada a optimizar los anchos de banda de dos equipos que se interconectan a través de dos o más puertos, utilizando la capacidad de estos puertos en una sola conexión virtual.

Por último tenemos el vPC Peer-Keepalive que resuelve escenarios activo-activo en los que se pierda conectividad el Peer Link, también es conocido como tolerante a fallos (fault-tolerant link).

- Optimizar los recursos informáticos con la virtualización, tomando en cuenta la tasa de crecimiento de los nodos físicos.
- Reducción del tiempo de implementación de nuevos servidores, mediante la configuración de perfiles o plantillas de servidores, que puede aplicarse al hardware.
- Reducción del consumo general de energía de un Centro de Datos con tecnologías que reducen el número de adaptadores de un servidor, virtualización que optimiza el consumo de energía del hardware.
- Aumento de la potencia de los recursos computacionales a un menor costo, adoptando el modelo de negocio de Computación en la Nube.
- Arquitecturas capaces de soportar SAN y LAN en una misma red (ofreciendo reducción de consumo de energía y consolidación de servidores).
- Arquitecturas que ofrecen una latencia menor a redes LAN tradicionales empresariales de modo que el proveedor de Computación en la Nube pueda construir aplicaciones transaccionales regulares.
- Arquitecturas que proveen la habilidad de distribuir tráfico de capa 2 a todos los enlaces disponibles.
- Cableado simplificado que ofrece eficiencia en flujos de aire, bajo consumo de energía y bajos costos de implementación para redes de alto ancho de banda.
- Mejoras en la administración y gestión limitando el impacto de tener equipamiento disperso.

Tomando en cuenta este proceso de consolidación mencionado, el proveedor de servicios de Computación en la Nube del caso de estudio, ha implementado una arquitectura de tres niveles con switches Data Center.

### **5.3.1.1 Data Center Switching**

El Centro de Datos o Data Center del servicio de Computación en la Nube es un punto consolidado que suministra varios servicios orientados al giro de negocio. El entorno del Centro de Datos requiere una sólida metodología de diseño para construir la red.

Según CISCO SYSTEMS, INC. (2015) las capas funcionales diseñadas para satisfacer las necesidades de un Centro de Datos se construyen normalmente alrededor de tres capas: la capa Core, la capa de agregación y la capa de Acceso. La capa Core es la principal en la red del Centro de Datos y proporciona interconexión hacia las capas de Agregación, la capa Core utiliza Switches de baja latencia y alto rendimiento que proporcionan altas velocidades de 10Gbps. Se recomienda el uso de 10GE<sup>25</sup> para realizar la interconexión con los Switches de Capa de Agregación y con enlaces redundantes. Los Switches de la capa Core funcionan exclusivamente como dispositivos de capa 3.

La capa de Agregación trabaja como una capa de servicios para el Centro de Datos, se encuentran en esta capa servicios tales como balanceadores de carga, firewalls, storage, etc. los Switches de capa de múltiple acceso también utilizan la capa de agregación como punto de interconexión, recomendado el uso de enlaces 10GE en esta capa para la interconexión hacia la capa Core (ascendente) y hacia la capa Acceso (descendente), proporcionando un mayor ancho de banda y escalabilidad en la red.

La capa Acceso proporciona conectividad para los servidores que ofrecen servicios de aplicaciones y servicios de Nube del proveedor, así como interconexiones diseñadas para clústers de servidores. Los Switches de esta capa se pueden configurar para implementaciones de capa 2 y capa 3. La flexibilidad para extender las VLAN en la capa de acceso es una característica importante que debe ser atendida en el diseño que se solventa con ambientes virtualizados que utilizan la tecnología VXLAN<sup>26</sup>. El uso de enlaces redundantes hacia la capa de agregación (ascendente) se utiliza para mejorar la resiliencia de la red<sup>27</sup>.

---

<sup>25</sup> 10 GE: Tecnología 10 Gigabit Ethernet.

<sup>26</sup> Virtual eXtensible LAN (VXLAN): Tecnología de tipo Overlay escalable utilizado en entornos virtualizados de tipo multi-tenants como VDC's que permiten extender la cantidad de Vlans a utilizarse (4096) hasta 16 millones de VLAN's ID.

<sup>27</sup> Resiliencia en la red: Se entiende como la capacidad de una red para recuperarse de fallas en condiciones de operación extrema, sin alterar su función y estructura.

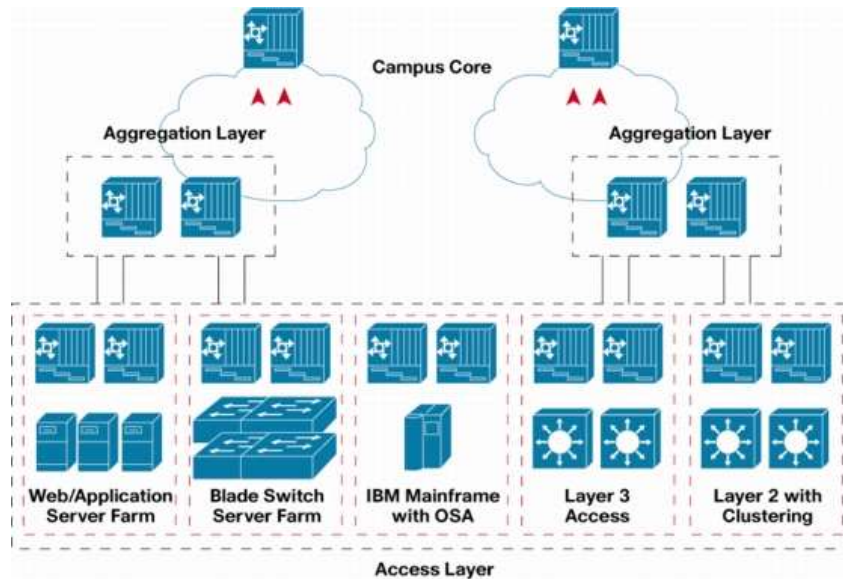


Figura 7. Topología Recomendada Centro de Datos (CISCO SYSTEMS, INC., 2015)

Estos diseños han sido confirmados y validados basados en pruebas internas de fabricantes como CISCO SYSTEMS, INC., que proporcionan diseños de sistemas basados en casos de uso común o en las prioridades actuales de sistemas de ingeniería. Incorporan un amplio conjunto de tecnologías, características y aplicaciones para satisfacer las necesidades de los servicios, cada diseño permite asegurar un despliegue más rápido, más confiable y totalmente predecible.

Basados en esta arquitectura, el proveedor de soluciones de servicios de Computación en la Nube ha optado por el uso de Switches Data Center que proveen soluciones para las demandas de Centro de Datos:

Diseño Flexible: Cuenta con Switches modulares, integrados y optimizados para Gigabit Ethernet, 10 Gigabit Ethernet, 40 Gigabit Ethernet y 100 Gigabit Ethernet.

Líderes en la industria de dispositivos para Networking de Centro de Datos: Cuenta con funciones de capa 2 y 3 ayudando a crear centros de datos escalables, seguros y

altamente seguros. Según Gartner, Inc. (2016) , CISCO SYSTEMS INC., se mantiene como uno de los líderes.



Figura 8. Cuadrante Mágico para Networking Data Center (Gartner, Inc., 2016)

Consistencia operativa: Interfaz y herramientas que simplifiquen la gestión, las operaciones, resolución de problemas y actualizaciones.

Virtualización: Dispositivos que soporten la movilidad, administración y operaciones de la máquina virtual.

Ejemplos de Switches Data Center son: CloudEngine series de Huawei, QFX y EX series de Juniper, Nexus de Cisco, Arista series 7000 R, entre otros.

### 5.3.1.2 Switches Data Center Sistema Operativo

Importante mencionar que los Switches Data Center cuentan con su propio sistema operativo dependiendo del fabricante, como el caso de Juniper con JUNOS basado en FreeBSD, Arista EOS (Extensible Operating System) basado en Linux, por citar algunos,

enfocándonos en el caso de estudio de CNT EP, los Switches Cisco Nexus cuenta con Cisco NX-OS.

NX-OS es un sistema operativo de red (Data Center Class) que permite una continua disponibilidad y estándares de misión crítica para ambientes de Centro de datos de próxima generación. Está basado e incluye características de los sistemas operativos propietarios de Cisco Systems: Storage Area Network Operating System (SAN-OS)<sup>28</sup>, Internetwork Operating System (IOS)<sup>29</sup> y Catalyst Operating System (CatOS)<sup>30</sup>.

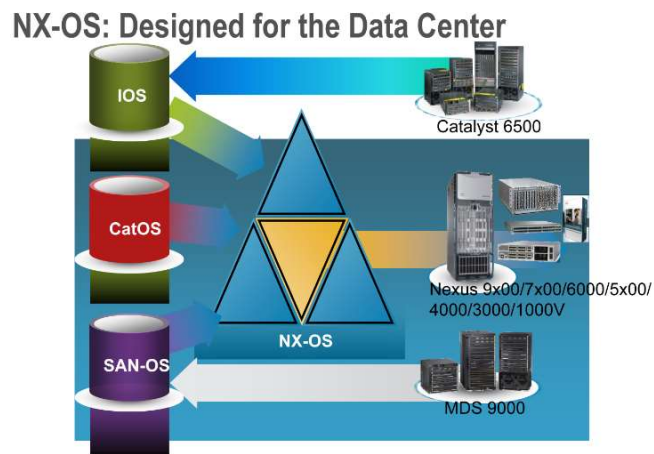


Figura 9. Sistema Operativo NX-OS (Fast Lane Consulting and Education Services, Inc., 2014)

Tomando en cuenta a Fuller, Jansen, & Mc Pherson (2013), las principales características y beneficios de NX-OS incluyen:

- Línea base de Enrutamiento y conmutación.

<sup>28</sup> Storage Area Network Operating System (SAN-OS): Sistema Operativo propietario de CISCO que introduce la tecnología de almacenamiento en la red. La familia de productos CISCO MDS cuentan con este Sistema Operativo.

<sup>29</sup> Internetwork Operating System (IOS): Sistema Operativo propietario de CISCO, utilizado en la mayoría de routers y switches que cuenta con funciones de enrutamiento, conmutación y networking.

<sup>30</sup> Catalyst Operating System (CatOS): Sistema Operativo propietario de CISCO, actualmente discontinuado y que fue diseñado con características para conmutación, es decir para switching de red.

- Virtualización y Multi-Propietario (Multi-Tenancy), utiliza Contextos de dispositivos virtuales (VDCs), los switches Data Center pueden ser segmentados en dispositivos virtuales. Los VDCs ofrecen varias ventajas tales como aislamiento de fallas, plano de administración, separación de tráfico y seguridad mejorada.
- Canales de puerto virtual (vPCs): Tecnologías Fabric, Virtual Port Channel permiten que un servidor o switch utilice un EtherChannel a través de dos switches con tráfico upstream sin un puerto STP-bloqueado permitiendo el uso de todo el ancho de banda del enlace disponible.
- Operación continua del sistema: El mantenimiento y actualizaciones se pueden realizar sin interrupciones de servicio, debido a la naturaleza modular de NX-OS que tiene la capacidad de reiniciar procesos dinámicamente.
- Seguridad: Cisco NX-OS proporciona confidencialidad e integridad de datos, con funciones de seguridad como listas de control de acceso (ACL), seguridad de puertos y accesos.
- Protocolos para Interconexión de Centros de Datos: Protocolos como Overlay Transport Virtualization (OTV) propietario de CISCO que habilita la extensión de capa 2 entre Centros de Datos distribuidos, sobre una capa de transporte de red capa 3.
- Alta Disponibilidad, monitoreo, administración y cuenta con una base de SDN (Software Definido por Red) para automatizar la gestión de la red.

# Comprehensive Data Center Feature Set

Engineered to meet evolving data center landscape



Figura 10. Características de NX-OS (Fast Lane Consulting and Education Services, Inc., 2014)

## 5.3.2 Arquitectura de Referencia de un Centro de Datos Virtual

De acuerdo con uno de los líderes en soluciones de virtualización VMware Inc. (2016) la computación moderna definida por software conocida como virtualización, es el primer paso y la capa base de Computación en la Nube.

Basados en una arquitectura de diseño de Data Center (Core, Agregación y Acceso) se describe la capa acceso de la Figura 2, donde se encuentra el Centro de Datos Virtual provisto por un Proveedor de Servicios:

**Infraestructura Física.-** Provista de hardware, como Granjas de Servidores, almacenamiento (Storage), componentes de Networking y Seguridad como routers, switches, balanceadores de carga, firewalls, etc.

**Virtualización.-** Se encuentran los Hipervisores o monitor de máquinas virtuales bare metal<sup>31</sup> que implementados sobre los servidores permiten una versión virtual de recursos físicos, esta capa de software maneja, gestiona y administra los recursos principales como

<sup>31</sup> Hypervisor Bare Metal: Software de virtualización implementado directamente sobre el hardware.

son el procesamiento, memoria, almacenamiento y conexiones de red, asignando dinámicamente estos recursos para todas las máquinas virtuales definidas en el servidor. Entre los principales proveedores de software de esta tecnología se encuentran por ejemplo: VMware vSphere ESXi, Microsoft Hyper-V, KVM, entre otros.

**Administrador de Hipervisores.-** Este componente permite gestionar y administrar toda la infraestructura virtual de manera centralizada, ejemplos de estos administradores tenemos a: VMware vSphere Client, oVirt KVM, Microsoft System Center Virtual Machine Manager (VMM), VMware VCenter Server que gestiona varios nodos de ESXi, etc.

**Capa de acceso al Centro de Datos Virtual Multi-propietario.-** Es Capa permite el acceso al servicio de Computación en la nube, que ha sido diseñado para agrupar los recursos de la infraestructura virtual de los Centros de Datos Virtuales de modo que cada cliente pueda acceder a servicios basados en catálogos que cuenten con una abstracción de recursos (procesamiento “CPU”, almacenamiento “storage”, memoria y red) agrupados en “pools” donde puedan acceder a un portal de auto-aprovisionamiento basado en web, ejemplos de esta capa multipropietario son: VMware vCloud Director, Manage IQ, entre otros.

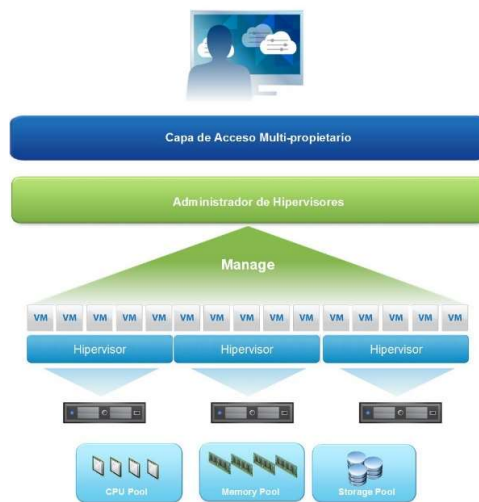


Figura 11. Arquitectura de Referencia Centro de Datos Virtual, IaaS (El Autor, 2016)

## **5.4 Interconexión entre una red MPLS y el Servicio de Computación en la Nube**

### **Caso de Estudio Proveedor CNT E.P.**

La interconexión se da entre PE (Provider Edge) y CE (Customer Edge) donde la infraestructura de Cloud Computing es un cliente de la red MPLS. Entre los dos equipos para establecer comunicación se habla un protocolo de enrutamiento de Gateway Interior IGP (Interior Gateway Protocol) como es el protocolo OSPF (Open Shortest Path First). De acuerdo con Cisco Networking Academy (2014) se define estos conceptos:

#### **5.4.1 IGP (Interior Gateway Protocol)**

Utilizado para el enrutamiento dentro de un Sistema Autónomo<sup>32</sup>. También se conoce como enrutamiento intra-AS. Las empresas, organizaciones y proveedores de servicios utilizan un IGP en sus redes internas. Los IGP incluyen protocolos de enrutamiento como: RIP, EIGRP, OSPF e IS-IS. Dentro de los IGP tenemos los protocolos de enrutamiento de estado de enlace que pueden crear una vista completa de la topología de la red, recopilando información de todos sus routers vecinos como es el caso de OSPF.

#### **5.4.2 OSPF (Open Shortest Path First)**

De acuerdo con CISCO (2013) (2015), el protocolo Open Shortest Path First (OSPF) se define en el RFC 2328 y es un Protocolo de Gateway Interior (IGP), que se utiliza para distribuir información de enrutamiento dentro de un Sistema Autónomo.

Un router envía un mensaje llamado paquete “Hello”, a cada interfaz habilitada para OSPF para descubrir otros routers vecinos. Una vez que se descubre un vecino, los dos routers comparan la información del paquete “Hello” para determinar si los routers tienen configuraciones compatibles, los routers vecinos intentan establecer adyacencia, es decir que los routers sincronizan sus bases de datos de estado de enlace para asegurarse de que tienen la

---

<sup>32</sup> Sistema Autónomo: Un sistema autónomo (AS) también conocido como Dominio de Enrutamiento es una colección de routers que se encuentran bajo una administración común ya sea una empresa u organización.

información de enrutamiento idéntica y actualizada. Los routers adyacentes comparten anuncios de estado de enlace (LSA) que son los que incluyen información sobre el estado operativo de cada enlace, el costo del enlace y cualquier otra información del vecino. Los enrutadores luego inundan estos LSA que reciben en cada interfaz habilitada para OSPF de modo que todos los routers eventualmente tengan bases de datos de estado de enlace idénticas. Cuando todos los routers tienen bases de datos de estado de enlace idénticas, la red es convergente. Cada enrutador entonces utiliza el algoritmo SPF (Shortest Path First) de Dijkstra<sup>33</sup> para construir su tabla de enrutamiento.

#### 5.4.2.1 Áreas OSPF

Un área es una división lógica de routers y enlaces dentro de un dominio OSPF, que crea subdominios separados. La inundación de LSA's está contenida dentro de un área, y la base de datos de estado de enlace está limitada a los enlaces dentro del área. Puede asignar una ID de área a las interfaces dentro del área definida, el ID de área es un valor de 32 bits que puede ser un número o puede tener una notación decimal punteada, como por ejemplo 10.10.10.1.

Si existe más de un área en una red OSPF, siempre se debe tener una área conocida como "Backbone"<sup>34</sup>, que tiene el ID de área reservada "0", al contar con más de un área, uno o más routers se convierten en routers área de frontera (ABR) que es el que se conecta tanto al área 0 o Backbone y al menos a otra área definida.

---

<sup>33</sup> Algoritmo SFP Dijkstra: También llamado algoritmo de caminos mínimos, determina el camino más corto dado un nodo origen al resto de nodos en un grafo con pesos en cada arista. Su nombre se refiere a Edsger Dijkstra (1959).

<sup>34</sup> Área Backbone: En una topología OSPF es el área principal, siempre debe existir y las demás deben estar conectadas a ella, etiquetada como área 0 y es un área estándar.

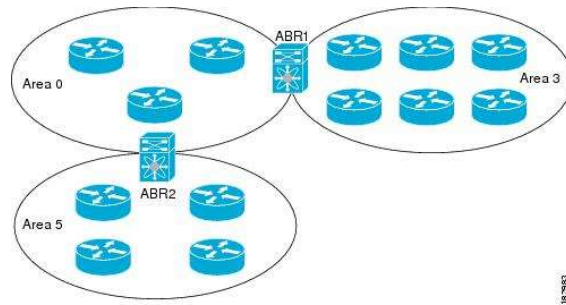


Figura 12. Áreas OSPF (CISCO, 2015)

De acuerdo al Figura 6, el ABR (Area Border Routers) tiene una base de datos de estado de enlace independiente para cada área a la que se conecta. El ABR envía una red sumariada desde un área conectada al área backbone. El área backbone envía la información resumida sobre un área a otra área, en este ejemplo el Área 0 envía información resumida sobre el Área 5 al Área 3.

OSPF define otro tipo de router, Autonomous System Boundary Router (ASBR) que conecta un área OSPF a otro sistema autónomo, puede redistribuir su información de enrutamiento en otro sistema autónomo o recibir rutas redistribuidas desde otro sistema autónomo.

Los LSA's construyen la tabla de ruteo y de acuerdo con CISCO (2015) se tiene los siguientes tipos:

Tipo	Nombre	Descripción
1	Router LSA	LSA son enviados a cada router, incluyen el estado, costo de todos los enlaces y vecinos. Estos LSA solo son inundados en el área local a la cual pertenecen.
2	Network LSA	Estos LSA son enviados por el DR <sup>35</sup> , los LSA nunca salen del área correspondiente, es decir un ABR no lo reenvía a otra área.
3	Network Summary LSA	Los LSA son enviados por el ABR para transferir la información de un área otra.

<sup>35</sup> Designated Router (DR): Si existen varios routers en una red y cada uno de ellos envía sus LSA, la misma información de estado de enlace se tendrá de múltiples fuentes, por lo que OSPF usa un solo router (DR) para controlar estas inundaciones de LSA y representar la red al resto de las áreas. Si el DR falla, OSPF seleccionará un router de respaldo conocido como Backup Designated Router (BDR).

4	ASBR Summary LSA	Los LSA son enviados a un área externa, es el comportamiento de un ASBR.
5	AS External LSA	Los LSA son generados por el ASBR. Este LSA incluye costo del enlace de un destino para un AS externo. Los LSA de los AS externos son inundados a través de los AS.
7	NSSA External LSA	Son LSA generados por el ASBR dentro de un área "not-so-stubby area" (NSSA). Este LSA incluye el coste del enlace a un destino de sistema autónomo externo. Describen redes externas a un área OSPF donde el ASBR envía las redes en formato tipo y el ABR las recibe y las redistribuye como tipo 5.
9-11	Opaque LSAs	Los LSA son usados para extender funciones de OSPF.

Tabla 1. Tipos de LSA

Dentro de la configuración de enlaces de Internet en el servicio de Cloud Computing se crean áreas NSSA (Not - So- Stubby Area) del lado de la infraestructura de Data Center de Cloud Computing, un área NSSA permite importar rutas externas de un sistema autónomo dentro de un NSSA utilizando redistribución, los ASBR redistribuyen estas rutas y generan un NSSA Externo, el LSA tipo 7 inunda a lo largo del área NSSA. Importante definir que se puede configurar el ABR que conecta el NSSA a otras áreas para traducir este LSA tipo 7 a LSA Externo (tipo 5). El área 0 o Backbone no puede ser un NSSA.

## **6. Estandarización de la configuración de servicios de conectividad en la Capa de Networking del proveedor de Servicios de Nube del caso de estudio**

El proveedor de Servicios en la Nube es un cliente de la Red MPLS, ya que esta red de transporte interconecta al Centro de Datos Virtual hacia el Internet o hacia las premisas del suscriptor con enlaces de datos. El Centro de Datos Virtual cuenta con una arquitectura de conectividad redundante basado en el diseño de tres niveles con la topología multicapa: Core, Agregación y Acceso, el proveedor de la Red MPLS proporciona los siguientes parámetros para su configuración, en los CE's que se encuentran en la capa Core que para el caso de estudio son switch Data Center "carrier class"<sup>36</sup> son capa L3 (IP y enrutamiento) y se configura los siguientes parámetros:

1. Redes WAN.- Dos subredes privadas con máscara de 30 bits, con las mismas se interconectan los CE's contra los PE's en un enlace troncal.
2. Red LAN.- Subred pública para el caso de un enlace de Internet y subred privada para el caso de un enlace de datos, con máscara de 29 bits como mínimo, para configurar la redundancia en el enlace.
3. Vlan id.- El rango acordado por el proveedor. En este punto se hace referencia al rango recomendado por el fabricante CISCO (2016) de acuerdo a la siguiente tabla:

---

<sup>36</sup> Carrier Class: Hardware y software extremadamente confiable, con pruebas de alto rendimiento, utilizado en redes grandes y de alta velocidad, los principales proveedores de servicios compran equipamiento de este tipo.

Número de Vlans	Rango	Uso
1	Normal	Viene por defecto, se puede utilizar esta VLAN, pero no se puede borrar o modificar esta VLAN.
2—1005	Normal	Se puede utilizar, modificar y eliminar estas VLANs.
1006—4094	Extendido	Se puede crear, nombrar y utilizar estas VLAN, pero no se puede cambiar los siguientes parámetros: * Estado siempre está activo. * Que la VLAN siempre está habilitada y no se puede deshabilitar estas VLAN.
3968—4047 y 4094	Asignado Internamente	Este rango de 80 VLAN's más la VLAN 4094, están asignadas para uso interno. No se puede crear, eliminar o modificar ninguna VLAN dentro del bloque reservado.

Tabla 2. Descripción del rango de VLAN's

4. Área OSPF.- Por lo general el proveedor de la red MPLS recomienda que se utilice el área 0 para enlaces de datos y para el caso de enlaces de Internet se lo maneje como en áreas NSSA agrupando clientes de 10 en 10.

Del lado del proveedor de Cloud Computing se configuran los siguientes parámetros para habilitar un enlace ya sea de datos o Internet, para el caso de la capa Core:

1. VRF Lite.- Virtual Routing and Forwarding (VRF) tiene como característica que un equipo de enrutamiento se pueda dividir en múltiples instancias de enrutamiento para que pueda manejar varias tabla de rutas trabajando al mismo tiempo, el término “lite” implica que solo se pertenece a un dominio de enrutamiento. La VRF lite se configura por cada enlace y se encuentran aisladas una de la otra.
2. Para el caso de un enlace de datos es importante conocer las redes LAN que se van a configurar en el Centro de Datos Virtual para propagarlas a través de rutas estáticas a través del protocolo OSPF.
3. Configuración de un SVI (Switch Virtual Interface), una SVI es una interfaz virtual configurada en un equipo multicapa, se crea un SVI por VLAN, es virtual porque no existe un puerto físico dedicado en la interfaz donde se proporciona procesamiento de capa 3.

4. Configuración de Protocolos de Redundancia, para el caso de estudio se estandarizó el uso de Hot Standby Router Protocol (HSRP).

Según CISCO (2016), HSRP es un protocolo propietario de su marca, que se encuentra especificado en la IETF RFC 2281, capaz de proporcionar alta disponibilidad de red brindando redundancia en el primer salto para hosts IP en una red con una dirección IP de puerta de enlace predeterminada. Este protocolo encamina el tráfico IP sin depender de que un único equipo de enrutamiento esté disponible, de esta manera un conjunto de interfaces de varios equipos pueden trabajar juntos para presentar la apariencia de que existe un único equipo de enrutamiento virtual o que hay una puerta de enlace predeterminada para todos los hosts en una red, esto permite que dos o más equipos configurados con este protocolo usen la dirección MAC y la dirección de red IP de un equipo de enrutamiento virtual. Uno de los enrutadores se selecciona para ser el enrutador activo y otro para ser el enrutador en espera o “stand by”, que asume el control de la dirección MAC del grupo y la dirección IP en caso de que el enrutador activo designado falle.

Es importante asignar una prioridad a la interfaz del equipo de enrutamiento, que seleccionará al equipo activo y al equipo “stand by”, el valor por defecto es 100, mientras más alta la prioridad el equipo será elegido como activo. Se recomienda el uso del comando “preempt” en la configuración del equipo activo para que se convierta en activo, incluso aunque su prioridad sea superior a la de los equipos restantes.

HSRP viene activo para el caso de estudio en el Switch Data Center por defecto en la versión 1, pero se puede utilizar la versión 2, de acuerdo a la Tabla 2., se tiene las siguientes diferencias:

	Nro. De Grupos Soportado	Soporta IPv6	Autenticación	Observaciones
<b>HSRP Versión 1</b>	0-255	NO	Texto plano	Rango pequeño para los grupos ID
<b>HSRP versión 2</b>	0-4095	SI	MD5	Versión 2 no es compatible con versión 1

Tabla 3. Versiones de HSRP

5. Subinterfaz CE hacia PE, una subinterfaz supera las limitaciones de hardware creando una interfaz lógica dentro de una interfaz física, pueden existir varias subinterfaces en una sola interfaz física. Esto se logra a través del protocolo de encapsulamiento IEEE 802.1Q (dot1Q) que permite a múltiples redes compartir el mismo medio físico para este caso por existir un enlace troncal entre CE y PE. En cada subinterfaz se configura la dirección IP, máscara de subred, VLAN asociada al protocolo dot1Q, asociación de la VRF Lite y protocolo OSPF con su ID y área.
6. Subinterfaz Inter Switch Data Center Core, de igual manera que en el punto anterior en cada subinterfaz se configura la dirección IP, máscara de subred, VLAN asociada al protocolo dot1Q, asociación de la VRF Lite y protocolo OSPF con su ID y área. Tomando en cuenta que aquí existe un enlace troncal entre los equipos CE de cloud para ofrecer redundancia.

Para el caso de la capa de Agregación o Distribución se cuenta con una solución de switch Data Center Carrier Class que cumplen funciones de capa 2, es decir aquí solo se configuran VLAN's.

Una vez realizada estas tareas se procede a configurar en la capa acceso que corresponde a los servidores y la capa de virtualización, la vlan y la red ya sea pública si es el caso de un enlace de Internet o privada si es un enlace de Datos para presentarla al Centro de Datos Virtual.

A continuación se detalla a modo general la configuración recomendada para que sea estandarizada en los equipos de la capa Core (Tabla 4), definido para este caso de estudio basado en los switches Datacenter de su arquitectura:

- CISCO (2016) para la configuración de una VRF lite.
- CISCO (2015) para configurar enlaces capa 3.
- CISCO (2016) para configurar el protocolo HSRP.
- CISCO (2015) para habilitar el protocolo de enrutamiento OSPF en esta configuración.

<b>Comando</b>	<b>Propósito</b>
vrf context <i>nombre_vrf</i>	Configuración VRF Lite, donde “nombre_vrf” es el nombre asignado a la VRF
ip route <i>subred/máscara gateway</i>	Ruta estática, para el caso que se desee propagar una red Lan configurada en el Centro de Datos Virtual en el caso de un enlace de datos, “subred/máscara” representa la subred junto con su máscara y el parámetro “gateway” su puerta de enlace o Gateway. Para el caso de configurar un enlace de Internet no se requiere este comando.
vlan <i>id_vlan</i>	Configuración de la VLAN, “id_vlan” es el ID de la VLAN
<b>Configurando interfaces Capa 3</b>	
interface Vlan <i>vlan_asignada</i>	Configuración del SVI, “vlan_asignada” corresponde al ID de la VLAN
vrf member <i>nombre_vrf</i>	Agrega el SVI a la VRF
ip address <i>subred/máscara</i>	Configuración de una dirección ip con su máscara: pública para el caso de un enlace de Internet ó privada para el caso de un enlace de Datos que a va a ser enrutada. Estas direcciones son proporcionadas por el proveedor de la red MPLS y deben tener como mínimo una máscara de 29 bits.
ip router ospf <i>etiqueta_instancia</i> area <i>area_id</i>	Habilitar protocolo OSPF en la interfaz para que la misma sea propagada. Agregar a la interfaz la instancia (“etiqueta_instancia”) y el área (“área_id”)
hsrp <i>id_grupo_hsrp</i>	Crea un grupo HSRP y configura el protocolo. Rango de la versión 1 de HSRP es desde 0 hasta 255.
preempt	Configura al equipo como activo en un grupo HSRP, si se tiene una prioridad más alta que el equipo activo actual. El intervalo es de 0 a 3600 segundos, viene deshabilitado por defecto.

priority <i>rango_prioridad</i>	Nivel de prioridad que selecciona el equipo activo en un grupo HSRP. Rango de 0 a 255. El valor por defecto es 100.
ip <i>dirección_ipvirtual</i>	Dirección IP virtual ("dirección_ip") del grupo HSRP y activa en el grupo, debe pertenecer a la misma subred que la dirección IPv4 de la interfaz, se la utiliza como puerta de enlace para este caso.
interface Ethernet <i>slot/puerto.número</i>	Habilitar subinterfaz cuando la Interfaz padre este habilitada. La Interfaz Ethernet padre es la interconexión entre el CE de Cloud Computing y el PE de la red MPLS, el número de rango es desde 1 hasta 4094.
encapsulation dot1q <i>id-vlan</i>	Configura la encapsulación IEEE 802.1Q VLAN en la subinterfaz. El rango es de 2 a 4093.
vrf member <i>nombre_vrf</i>	Agrega esta subinterfaz a la VRF.
ip address <i>subred/máscara</i>	Configura una dirección IP para esta interfaz. Se debe realizar este paso después de asignar esta interfaz a un VRF, esta dirección son las direcciones WAN privada de 30 bits proporcionadas por el proveedor de la Red MPLS.
ip router ospf <i>etiqueta_instancia</i> area <i>area_id</i>	Habilitar protocolo OSPF en la interfaz para que la misma sea propagada. Agregar a la interfaz la instancia ("etiqueta_instancia") y el área ("área_id")
interface Ethernet <i>slot/puerto.número</i>	Habilitar subinterfaz cuando la Interfaz padre este habilitada. La Interfaz Ethernet padre es la interconexión entre los switches Data Center que garantizan la redundancia entre ellos, el número de rango es desde 1 hasta 4094.
encapsulation dot1q <i>id-vlan</i>	Configura la encapsulación IEEE 802.1Q VLAN en la subinterfaz. El rango es de 2 a 4093.
vrf member <i>nombre_vrf</i>	Agrega esta subinterfaz a la VRF.
ip address <i>subred/máscara</i>	Se asigna una dirección IP para esta interfaz. Se debe realizar este paso después de asignar esta interfaz a un VRF, este direccionamiento son las direcciones entre los switches Data Center, punto a punto con máscara de 30 bits.
ip router ospf <i>etiqueta_instancia</i> area <i>area_id</i>	Habilitar protocolo OSPF en la interfaz para que la misma sea propagada. Agregar a la interfaz la instancia ("etiqueta_instancia") y el área ("área_id")
<b>Configuración del protocolo de enrutamiento OSPF</b>	
router ospf <i>id_ospf</i>	Configuración del protocolo de enrutamiento dinámico, "id_ospf" es el ID del proceso OSPF
vrf <i>nombre_vrf</i>	Asociación de la VRF al protocolo OSPF
router-id <i>dirección_ip</i>	Se recomienda configurar un ID equipo, esta dirección IP "dirección_ip" identifica esta instancia de OSPF y debe existir en una interfaz configurada en el sistema.

<p>area <i>area_id</i> nssa</p>	<p>Crea esta área como un NSSA recomendada para este caso sólo para enlaces de Internet, para enlaces de Datos no se requiere este comando debido a que se configura en el área 0.</p>
---------------------------------	--

Tabla 4. Configuración recomendada equipos Core para enlaces de Datos e Internet

Para el caso de la capa Agregación se detalla a modo general la configuración estandarizada (Tabla 5) recomendada por CISCO (2016), para los switch Data Center tomando en cuenta el caso de estudio:

Comando	Propósito
vlan <i>id_vlan</i>	Configuración de la VLAN, “id_vlan” es el ID de la VLAN. El ID es el número asignado a la VLAN.

Tabla 5. Configuración recomendada en los equipos Agregación o Distribución para enlaces de Datos e Internet.

Con esto la vlan es configurada en la capa de virtualización y la misma es presentada al Centro de datos virtual para que el cliente final pueda utilizar el enlace.

Nombre	Estado	Dirección de puerta de enlace	Tipo	Conec...	Grupo de direcciones IP (utilizadas/total)	Compartida	Propietario
v250_voip	OK	172.16.1.1/24	Directa	v250_v	3,20 %	Yes	vDC_DEMO_SDN
v676d_voip_internet	OK	201.219.1.241/29	Directa	v676_v	33,33 %	Yes	vDC_DEMO_SDN

Figura 13. Redes de Data Center Virtual (CNT E.P., 2016)

La configuración recomendada tanto para la capa Core, como la capa Agregación de la Tabla 4 y Tabla 5, está basada en el Sistema Operativo NX-OS que utilizan los switches Datacenter.

## **7. Análisis de herramientas OPEN SOURCE que permiten el desarrollo de una solución de automatización prototipo para la configuración de servicios de conectividad de los Data Centers Virtuales**

Una vez realizado el análisis de las tecnologías y recomendar una estandarización en las configuraciones de los servicios de conectividad de los Data Centers Virtuales basados en la arquitectura de Computación en la Nube del proveedor del caso de estudio, se procede a analizar herramientas Open Source o de código abierto, que permitan desarrollar una solución prototipo que estandarizará y automatizará la configuración de estos servicios de conectividad, las cuales mencionamos a continuación para lograr este objetivo: lenguaje de programación Python, Base de Datos MySQL y Sistema Operativo Linux Ubuntu Server.

### **7.1 Herramientas de Código abierto (Open Source)**

De acuerdo con Opensource.com supported by Red Hat (2014) el término "código abierto" se refiere a algo que se puede modificar y compartir porque su diseño es accesible al público, el término tuvo su origen en el contexto del desarrollo de software designando un enfoque específico para crear programas informáticos. Código abierto (Open Source) hoy en día es un conjunto más amplio que incluye proyectos, productos o iniciativas de código abierto que sigan los principios del intercambio abierto, la participación colaborativa, elaboración de rápidos prototipos, transparencia, meritocracia y desarrollo orientado a la comunidad.

Se puede llegar a confundir el término Open Source con el de Software libre o cualquier otro tipo de software, algunos programas informáticos tienen un código fuente que sólo puede ser modificado por una persona, un equipo o la organización que lo creó y mantiene un control exclusivo sobre este, lo que se conoce como software "propietario" o "software de código

cerrado". Sólo los autores originales del software propietario pueden legalmente copiar, inspeccionar y modificar este software, para utilizar un software propietario, los usuarios de deben estar de acuerdo (generalmente con una firma de licencia mostrada la primera vez que se instala o ejecuta el software) que no se hará nada con el software sin el consentimiento expreso de los autores.

En cambio el software de código abierto es diferente ya que sus autores hacen que el código fuente esté disponible para otros que quieran ver, copiar, aprender, modificar y compartirlo. De la misma manera que el software propietario, los usuarios aceptan los términos de una licencia cuando usan software de código abierto, pero en este caso, los términos legales de estas licencias difieren considerablemente de las licencias de software propietario ya que las licencias de código abierto afectan a la forma en que la gente puede usar, estudiar, modificar y distribuir software.

En general, las licencias de código abierto otorgan a los usuarios permisos para usar software de código abierto, con el propósito que deseen, algunas licencias de código abierto, como las licencias "copyleft", estipulan que cualquier persona que publique un programa de código abierto modificado también se debe lanzar en conjunto con el código fuente. Por otra parte, algunas licencias de código abierto estipulan que cualquier persona que altera y comparte un programa con otros también debe compartir el código fuente de ese programa sin cobrar un cargo por el licenciamiento.

Por diseño, las licencias de software de código abierto promueven la colaboración y el intercambio ya que se pueden hacer modificaciones en el código fuente e incorporar esos cambios a proyectos propios, apoyando a los programadores para que puedan acceder, ver y modificar el software de código abierto en cualquier momento, siempre y cuando dejen que otros hagan lo mismo cuando comparten su trabajo.

Gran parte del Internet está basado en tecnologías de código abierto, como el uso de Sistemas Operativos Linux, aplicaciones como servidores web Apache, bases de datos como MySQL, lenguajes de programación como Python que permiten que cualquier persona que utilice Internet tenga todos los beneficios del software de código abierto. Esto ha generado una evolución hacia el consumo de servicios hacia tecnologías y ecosistemas remotos como la Computación en la Nube, porque involucra actividades (como almacenar archivos, compartir fotos o ver vídeos) que incorporan no sólo a dispositivos locales sino también una red global de computadoras remotas que forman una "atmósfera" a su alrededor.

La Computación en la Nube es cada vez más importante de la vida cotidiana con dispositivos conectados a Internet, algunas aplicaciones de Cloud Computing, como Google Apps<sup>37</sup> que son propietarias, otras como ownCloud<sup>38</sup>, SugarCRM<sup>39</sup>, que están basadas en herramientas Open Source, así también se puede validar que las plataformas de Computación en Nube están basadas en código abierto (como el caso de Manage IQ) o de código cerrado (como el caso de VMWare vCloud Director).

El prototipo que automatizará y estandarizará la interconexión entre el servicio de Computación en la nube y la red MPLS está basado en las siguientes herramientas y software Open Source: Python, MySQL, phpMyAdmin, implementados en una máquina virtual de un Data Center Virtual en el servicio de Computación en la Nube del proveedor del caso de estudio CNT EP.

### **7.1.1 Lenguaje de programación Python Software Foundation**

Es una organización sin fines de lucro, dedicada a promover, proteger y avanzar el lenguaje de programación Python, apoyando y facilitando el crecimiento de una gran

---

<sup>37</sup> Google Apps: Suite de herramientas en línea basadas en web para mensajería y colaboración.

<sup>38</sup> ownCloud: Programa multiplataforma que sincroniza y permite el almacenamiento en línea de archivos.

<sup>39</sup> SugarCRM: Es un sistema para administración y relación de clientes (CRM).

comunidad diversa e internacional de programadores que utilizan este lenguaje. Python Software Foundation (2016).

Python soporta programación orientada a objetos y es multiplataforma brindando el uso de extensiones y un intérprete que incluye un modo interactivo para validar expresiones una a una, útil para realizar scripts y programas, desde aplicaciones Windows, Linux, Networking, servidores, portales web, etc.

Es un lenguaje interpretado, es decir que no necesita compilar el código fuente para poder ejecutarlo (en realidad se realiza la compilación pero es de manera transparente para el programador), ofreciendo ventajas como rapidez en el desarrollo, depuración de código y fácil aprendizaje, por esta razón en este último tiempo el lenguaje se ha hecho muy popular, debido a:

- La variedad de librerías, tipos de datos y funciones incorporadas en el propio lenguaje, que ayudan a tareas habituales sin tener que programarlas.
- Sencillez y rapidez para crear programas, un programa hecho en Python puede tener de 3 a 5 líneas de código menos que los lenguajes tradicionales Java o C. Tiene una sintaxis visual, gracias a su notación indentada (con márgenes) de cumplimiento estricto, en muchos lenguajes, para separar partes de un código, se usan símbolos como llaves o palabras claves “begin” y “end”. Para separar las porciones de código en Python se requiere tabular hacia el interno, colocando un margen al código que va a ir dentro de una función o bucle, esto permite que los programadores adopten notaciones y que los programas tengan un aspecto muy similar.
- Variedad de plataformas para desarrollar, como Unix, Windows, OS/2, Mac, Linux y otros.
- Python es gratuito, incluso para uso con fines empresariales.

Python cuenta con varios casos de éxito por su productividad, la calidad del software y facilidad de mantenimiento en muchas empresas e instituciones alrededor del mundo, citamos algunas áreas y tecnologías que lo utilizan: Desarrollo de Redes, Computación en la Nube (con el desarrollo de API's<sup>40</sup>), efectos visuales, aviación, entre otras áreas y tecnologías.

De acuerdo con la IEEE Spectrum (2016) que hace referencia a los mejores lenguajes de programación ubicando a Python dentro de los 10 más populares y preferidos por los desarrolladores:











Language Rank	Types	Spectrum Ranking
1. C		100.0
2. Java		98.1
3. Python		98.0
4. C++		95.9
5. R		87.9
6. C#		86.7
7. PHP		82.8
8. JavaScript		82.2
9. Ruby		74.5
10. Go		71.9

Figura 14. Los mejores lenguajes de programación 2016 (IEEE Spectrum, 2016)

Estas son muchas de las razones por las que se escogió Python para realizar el prototipo que contiene el programa para automatizar la implementación de enlaces que brindan servicio a Cloud Computing del proveedor del caso de estudio.

<sup>40</sup> Application Programming Interface (API): La Interfaz de Programación de Aplicaciones, es un código que cuenta con especificaciones para que las aplicaciones puedan comunicarse entre ellas, creando una interfaz entre diferentes programas para su interacción.

### 7.1.2 Base de datos MySQL

Oracle MySQL (2016) la define como una base de datos de código abierto, embebida<sup>41</sup> y relacional, ampliamente utilizada en Web utilizada por empresas como Facebook, Twitter, Youtube y sitios web principales según la clasificación de Alexa Internet, Inc.<sup>42</sup>.

Es distribuida por miles de ISV<sup>43</sup> y OEM<sup>44</sup>, funciona sobre múltiples plataformas como: GNU/Linux, FreeBSD, Mac OS X, Windows, etc., su dueño actual es Oracle Corporation y cuenta con licencia dual: GPL<sup>45</sup> y licencia comercial de la que se clasifican los siguientes productos:

- Oracle MySQL Cloud Service versión comercial que está basada en la edición MySQL Enterprise y funciona con Oracle Cloud brindando un servicio de base de datos MySQL de nivel empresarial.
- MySQL Enterprise Edition versión comercial que incluye MySQL Enterprise Server, Monitoreo de la red MySQL, servicios de consulta y soporte de producción MySQL.
- MySQL Cluster CGE versión comercial es una base de datos transaccional diseñada para un rápido acceso en tiempo real y permanente a los datos bajo altas condiciones de rendimiento.

---

<sup>41</sup> Base de Datos Embebida: Es aquella que no inicia ni implementa su servicio en una máquina cliente y es independiente de la aplicación, puede enlazar directamente al código fuente o utilizarse como una librería.

<sup>42</sup> Alexa Internet, Inc.: Compañía que provee información y estadísticas acerca de la cantidad de visitas que tiene un sitio web clasificándolo en un ranking.

<sup>43</sup> Independent Software Vendor (ISV): Es un proveedor independiente de software que fabrica y vende software que puede ejecutarse en uno o varias plataformas tanto a nivel de hardware o sistemas operativos.

<sup>44</sup> Original Equipment Manufacturer (OEM): Es un fabricante que manufactura equipamiento o productos originales, que son comprados por otro fabricante y vendidos al por menor bajo la marca de la empresa que los adquirió.

<sup>45</sup> General Public License (GPL): Es una licencia utilizada para software que garantiza al usuario final la libertad de usarlo, estudiarlo, copiarlo y modificarlo. Declara que el software cubierto por esta licencia es libre y lo protege de intereses que intenten adueñarse del mismo.

- MySQL Community Edition con licencia GPL, versión libre de código abierto, que cuenta con el respaldo de una comunidad activa de desarrolladores y colaboradores del “open source”.

MySQL cuenta varios casos de éxito en todas las áreas como: telecomunicaciones, seguridad, Computación en la Nube, educación, Geo-espacial, Big Data<sup>46</sup>, gobierno, salud, entre otras. Según DB-Engines (2016), MySQL según su popularidad se mantiene entre las 10 bases de datos más utilizadas:

Rank			DBMS	Database Model	Score		
Nov 2016	Oct 2016	Nov 2015			Nov 2016	Oct 2016	Nov 2015
1.	1.	1.	Oracle +	Relational DBMS	1413.01	-4.09	-67.94
2.	2.	2.	MySQL +	Relational DBMS	1373.56	+10.91	+86.71
3.	3.	3.	Microsoft SQL Server	Relational DBMS	1213.80	-0.38	+91.48
4.	↑ 5.	↑ 5.	PostgreSQL	Relational DBMS	325.82	+7.12	+40.13
5.	↓ 4.	↓ 4.	MongoDB +	Document store	325.48	+6.67	+20.87
6.	6.	6.	DB2	Relational DBMS	181.46	+0.90	-21.07
7.	7.	↑ 8.	Cassandra +	Wide column store	133.97	-1.09	+1.05
8.	8.	↓ 7.	Microsoft Access	Relational DBMS	125.97	+1.30	-14.99
9.	9.	↑ 10.	Redis	Key-value store	115.54	+6.00	+13.13
10.	10.	↓ 9.	SQLite	Relational DBMS	112.00	+3.43	+8.55

Figura 15. Clasificación de las 10 Bases de Datos más utilizadas (DB-Engines, 2016)

La Figura 15, muestra esta clasificación a nivel general, utilizando los siguientes parámetros: número de búsquedas en sitios web, sistemas de interés general, frecuencia de discusiones técnicas sobre el sistema, número de ofertas de empleo en las que se menciona el sistema, número de perfiles en las redes profesionales, en las que se menciona el sistema y relevancia en redes sociales.

<sup>46</sup> Big Data: Se refiere a grandes cantidades de datos y procedimientos para encontrar y analizar patrones dentro del mismo.

MySQL Es uno de los componentes principales de una solución para desarrollo de aplicaciones web conocida como “LAMP” (Linux, Apache<sup>47</sup>, MySQL y PHP<sup>48</sup>) o WAMP (Windows, Apache, MySQL y PHP) que es la unión de un sistema operativo, servidor web, base de datos y lenguaje de programación.

Basados en estas características se utiliza esta Base de datos para almacenar los parámetros de los enlaces que son configurados en el servicio de Computación en la Nube que brinda el proveedor con el fin de obtener reportes y estado de la configuración.

### **7.1.3 Administración de MySQL**

Una vez implementada la solución LAMP, resulta fácil integrar una herramienta escrita en PHP para la administración y gestión de la Base de Datos MySQL a través de un portal web al que puede acceder el administrador, conocido como “PhpMyAdmin” que puede administrar un servidor completo de MySQL o una base de datos sencilla, por lo que es necesario tener configurado un usuario MySQL que tenga privilegios de lectura o escritura solamente en la base de datos deseada.

El equipo de desarrollo de esta herramienta, The phpMyAdmin devel team (2016) confirma que PhpMyAdmin es capaz de visualizar, borrar bases de datos, tablas, vistas, campos e índices, mostrar varios resultados a través de procedimientos almacenados o consultas, realizar tareas de mantenimiento, ejecutar cualquier expresión SQL, exportar e importar datos, validar cambios en bases de datos, tablas y vistas entre otras tareas administrativas y de gestión para MySQL.

---

<sup>47</sup> Apache: Es un servidor web HTTP de código abierto multiplataforma, desarrollado y mantenido por la comunidad de Apache Software Foundation proyecto HTTP server.

<sup>48</sup> Hipertext Preprocesor (PHP): Es un lenguaje de código abierto para el desarrollo web y que puede ser insertado en código HTML.

### 7.1.4 Sistema Operativo GNU/Linux

De acuerdo con Debian GNU/Linux (2015), Linux es un sistema operativo que cuenta con un conjunto de programas que interactúan con el equipo y ejecutan programas. Estos programas fundamentales son necesarios para que el equipo pueda comunicarse, interactuar y recibir instrucciones del usuario tales como leer y escribir datos al disco duro, cintas e impresoras, contralar la memoria, entre otras tareas. Lo más importante de un sistema operativo es el núcleo o kernel, en el sistema GNU/Linux, Linux es el núcleo y el resto del sistema consiste de otros programas escritos para el proyecto GNU<sup>49</sup>, como el núcleo de Linux no forma un sistema operativo funcional, se lo llama “GNU/Linux”, mas conocido como el sistema operativo “Linux”.

Esta basado y modelado en un sistema operativo tipo Unix, diseñado para que fuera un sistema multi-tarea y multi-usuario, gran parte de su desarrollo lo realizan voluntarios y comunidades con el fin de mejorarlo cada día. Free Software Foundation ha sido quien más ha contribuido creando la mayor parte de herramientas que se utilizan en Linux manteniendo la filosofía del código abierto.

El núcleo de Linux fue desarrollado en 1991 por Linus Torvalds quien actualmente coordina el trabajo de cientos de desarrolladores, experimentando un crecimiento en el mercado de servidores y empezando a ser popular entre usuarios domésticos y en empresas en el sitio Linux Kernel Organization, Inc. (2016) se cuenta con las últimas versiones estables de los nucleos de Linux. Gracias a estas comunidades de desarrolladores, Linux cuenta con varias distribuciones, la diversidad de las distribuciones Linux se da debido a requerimientos técnicos de organización y de puntos de vista diferentes entre usuarios y proveedores. El modo de licenciamiento del software libre permite que cualquier usuario

---

<sup>49</sup> GNU's not Unix (GNU): Es un sistema operativo de software libre, que ha permitido utilizar un equipo respetando la libertad de los usuarios, ejemplo de versiones de GNU que se pueden instalar son las distribuciones de GNU/Linux.

con los conocimientos e interés pueda adaptar o diseñar una distribución de acuerdo a sus necesidades, las distribuciones de Linux pueden ser:

- Comerciales o no comerciales.
- Ser completamente libres o incluir software propietario
- Orientadas a usuarios de hogar y empresas.
- Para servidores, escritorios o dispositivos virtuales.
- Para usuarios regulares o usuarios avanzados con software y herramientas especializadas.
- Para uso general o dispositivos especializados, como firewalls, routers, balanceadores de carga, clústers de servidores, orquestadores de nube, entre otros.
- Recomendadas y certificadas para tipos específicos de hardware o arquitecturas definidas.
- Diseñadas para grupos específicos, por ejemplo su localidad, lenguaje.
- Para aplicaciones específicas como: educación, multimedia, científicas y video juegos.
- Diseñadas para ser portables.

Tomando como referencia a [Linux.com News for the Open Source Professional](#) by The Linux Foundation (2016) y [Unsigned Integer Limited, DistroWatch.com](#) (2016) mencionan una lista de las distribuciones más populares, basadas en el número de visitas y descargas:

<b>Distribución Populares según DistroWatch</b>	<b>Mejores Distribuciones según Linux.Com</b>	<b>Distribuciones para empresas y servidores</b>
Linux Mint	OpenSuse	Red Hat Enterprise Linux
Ubuntu	Arch Linux	SUSE Linux Enterprise
Debian	Elementary OS	Ubuntu Server
CentOS	Solus	Debian
Fedora	Chrome OS	CentOS
OpenSuse	Ubuntu	
Arch Linux	Linux Mint	
Slackware Linux	Tails	

Tabla 6. Distribuciones de Linux

Tomando como referencia a las distribuciones para empresas y servidores, se seleccionó Ubuntu Server tomando en cuenta las siguientes características:

- Soporte de 5 años patrocinado por Canonical Ltd. que es una compañía británica que no vende la distribución con fines de lucro, esto permite acceder a los paquetes, parches y actualizaciones sin tener que pagar por una suscripción. Ubuntu Server esta basado en Debian una distribución que se destaca por ser estable y tener un gran sistema de gestión de paquetes.
- Corre sobre la mayoría de arquitecturas físicas y compatible con la mayoría de ambientes virtuales.
- Kernel linux estable 4.4
- Certificado en la mayoría de ambientes de Cloud Computing como Amazon Web Services<sup>50</sup>, Microsoft Azure<sup>51</sup>, IBM, HP Cloud y es el líder en soluciones de plataformas abiertas de Nube con OpenStack.

---

<sup>50</sup> Amazon Web Services (AWS): Plataforma de servicios de nube de la empresa Amazon, que brinda procesamiento, almacenamiento de base de datos, entrega de contenido y es escalable.

<sup>51</sup> Microsoft Azure: Servicios de Nube ofrecidos por la multinacional Microsoft.

## 8. Implementación del programa de automatización

Para la implementación del programa prototipo, por favor referirse al **ANEXO I: CONFIGURACIÓN DE LOS COMPONENTES DE LA SOLUCIÓN**, que automatizará la configuración de los enlaces que brindan servicios de conectividad en la Nube del proveedor del caso de estudio, ha sido realizada en un Centro de Datos Virtual bajo el modelo de negocio IaaS, utilizando las siguientes tecnologías, componentes y herramientas:

Componentes de la Solución		Versión
Centro de Datos	Virtual (VDC)	Cloud 3.0 CNT EP
Sistema Operativo	Linux Ubuntu Server	16.04.1
Lenguajes de Programación	Python	2.7.12
	PHP	7.0.8
Base de datos	MySQL	5.7.16
Servidor Web	Apache	2.4.18
Administrador de Base de Datos	phpMyAdmin	4.5.4

Tabla 7. Componentes para desarrollar la Solución de Automatización de Enlaces

### a. Arquitectura General de la Solución

A continuación detallamos la Arquitectura General de la Solución implementada en un Centro de Datos Virtual (VDC), que cuenta con un enlace de Internet para realizar las pruebas de automatización y un enlace de Gestión que interconecta al VDC con la Gestión de la capa de Networking del servicio de Computación en la Nube del CSP del caso de estudio, para lograr configurar de forma automática a través del programa esta capa.

Cuenta con máquinas virtuales implementadas una para el servidor que contiene el programa de automatización, otra para gestión del VDC y una para el Firewall Virtual que cumple funciones de enrutamiento y seguridad perimetral.

Para acceder al VDC se lo puede realizar a través de dos formas: por el portal de gestión del CSP o mediante un VPN que está configurada a nivel del Firewall Perimetral.

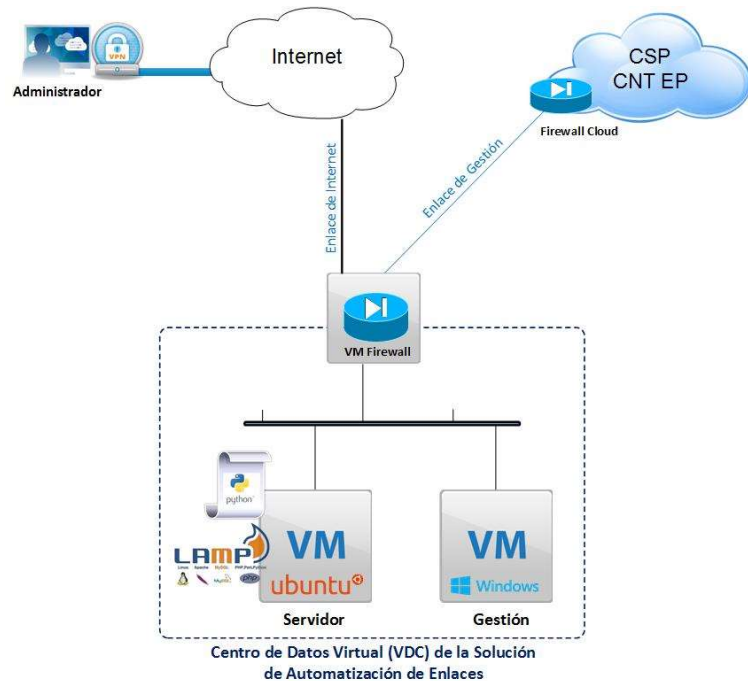


Figura 16. Arquitectura General de la Solución (El Autor, 2016)

### b. Prototipo para automatizar la configuración de enlaces en el servicio de Computación en la Nube del CSP del caso de estudio

El prototipo de automatización Versión 1.0, es un programa desarrollado en Python que permite configurar enlaces con tareas como: crear un nuevo enlace, borrarlo, realizar consultas. El código fuente del programa se encuentra en el **ANEXO II: CODIGO FUENTE DEL PROTOTIPO**.

El programa brinda un aprovisionamiento de enlaces de forma automática, facilitando el diseño, despliegue y gestión, optimizando la disponibilidad y confiabilidad del servicio

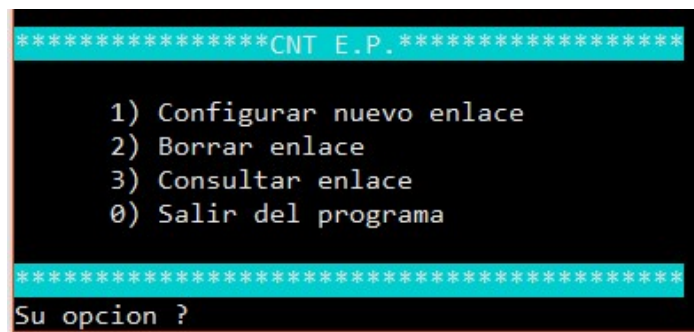
reduciendo el tiempo de gestión global y posibles errores en la implementación tradicional reduciendo fallas operativas y por lo tanto el OpEX<sup>52</sup>.

A continuación se describen los pasos para utilizar el programa de automatización, ejecutar en línea de comandos el prototipo:

```
cloudcnt@sdnserver:~/codigo$ python p_auto_v1.py
```

- Opción 1. Configurar nuevo enlace

Pantalla principal, se ingresa la opción 1



```
*****CNT E.P.*****  
  
1) Configurar nuevo enlace  
2) Borrar enlace  
3) Consultar enlace  
0) Salir del programa  
  
*****  
Su opcion ?
```

Figura 17. Pantalla de Inicio del Programa de Automatización (El Autor, 2016)

Luego proporcionamos las credenciales de acceso al Switch Data Center, como son dirección ip, usuario y contraseña.

---

<sup>52</sup> Operating Expenditure (OpEX): Gasto de funcionamiento operativo, se refiere a los costos asociados con el mantenimiento de equipos, gastos de consumibles y otros de funcionamiento para la producción y el funcionamiento del negocio.

```
*****CNT E.P.*****
1) Configurar nuevo enlace
2) Borrar enlace
3) Consultar enlace
0) Salir del programa
*****
Su opcion ? 1
+ Configurar nuevo enlace
Ip/Hostname: 192.168.1.254
Usuario: user_admin
Password:
Bienvenido a SW_CORE_CLOUD1
+ Configurar Enlace
VLAN ID: █
```

Figura 18. Creación de Enlaces (El Autor, 2016)

Como siguiente paso ingresamos los siguientes parámetros: VLAN ID, VRF, Subred LAN (pública o privada), ID OSPF, ÁREA OSPF, HSRP group, descripción para la Subinterfaz, Subred WAN (interconexión contra MPLS), Subred Inter-Switch, al final se creará el enlace.

```
Bienvenido a SW_CORE_CLOUD1
+ Configurar Enlace
VLAN ID: 888
VRF: demo_sdn
SUBRED LAN: 150.1.1.0/29
ID OSPF: 112
AREA OSPF: 0
ID HSRP: 150
DESCRIPCION: Internet Cliente DEMO
SUBRED WAN: 10.16.1.0/30
SUBRED INT_SW: 192.168.130.1/30

Guardando configuracion ...
Por favor espere
100%
|#####
-El enlace ha sido creado
```

Figura 19. Ingreso de Parámetros para la configuración de enlaces (El Autor, 2016)

- Opción 2. Borrar Enlace:

En el menú principal escogemos la opción 2, ingresamos el Vlan ID del enlace que vamos a borrar. Se despliega la información del enlace que será eliminado, se confirma el borrado “Desea Continuar (s/n):” opción “s” , con esto la configuración del enlace será borrada.

```
- Borrar Enlace
VLAN ID: 888
- El siguiente enlace ser@ eliminado:

VRF:          demo_sdn_i
VLAN:         888
ID OSPF:      112
Desea Continuar (s/n): s
Guardando configuracion ...
Por favor espere
100% |#####|

- El enlace ha sido borrado
```

Figura 20. Borrar un enlace (El Autor, 2016)

- Opción 3. Consultar si existe un enlace configurado:

En esta opción se puede validar si existe un enlace configurado, donde se despliega en pantalla información de los parámetros del enlace.

```
? Consulta de enlace
VLAN ID: 888

-----
VRF:          demo_sdn_i
VLAN:         888
ID OSPF:      112
-----

*****CNT E.P.*****

1) Configurar nuevo enlace
2) Borrar enlace
3) Consultar enlace
0) Salir del programa

*****
Su opcion ?
```

Figura 21. Consulta de un enlace (El Autor, 2016)

Es importante señalar que para obtener un reporte más detallado del enlace se puede utilizar la herramienta de administración “phpMyAdmin” y exportar un reporte.

- Opción 0. Salir del programa.

```
*****CNT E.P.*****  
  
1) Configurar nuevo enlace  
2) Borrar enlace  
3) Consultar enlace  
0) Salir del programa  
  
*****  
Su opcion ? 0  
  
Usted ha salido del programa, hasta pronto!!!  
cloudcnt@sdnsver:~/codigo$
```

Figura 22. Opción Salir (El Autor, 2016)

Estas son las opciones principales de la versión 1.0 del programa de automatización, este entregable contiene scripts desarrollados bajo herramientas y software Open Source.

## 9. Conclusiones y Recomendaciones

Después del análisis del caso de estudio se concluye lo siguiente:

- Basados en el análisis de las tecnologías que utiliza el proveedor de servicios de Computación en la Nube del caso de estudio, para brindar servicios de conectividad a los Data Centers Virtuales, se identifica una arquitectura y uso de tecnologías de referencia que son: Plataforma de servicios de Nube, Red de Transporte MPLS y Centro de Datos Virtual, la interoperabilidad e integración de estas tecnologías brindan soluciones de TI al cliente o suscriptor de acuerdo a la demanda tecnológica actual.
- Se ha planteado la estandarización en la configuración de los servicios de conectividad para los Data Centers Virtuales del proveedor del caso de estudio, basados en su arquitectura actual, se ha tomado en cuenta las recomendaciones de diseños validados y recursos técnicos de uno de los fabricantes líderes en el sector de las telecomunicaciones con el fin de poder interconectarse hacia la red de transporte MPLS para proveer enlaces de Internet y Datos hacia la Nube garantizando así movilidad y la necesidad de estar permanente conectado a la información.
- Se ha realizado el análisis de herramientas Open Source o de Código Abierto que apoyaron al desarrollo una solución de automatización prototipo para la configuración de los servicios de conectividad para los Data Centers virtuales, donde se implementó una máquina virtual en un ambiente de Nube con un Sistema Operativo base Linux Ubuntu Server, lenguaje de programación

Python, motor de base de datos MySQL y la herramienta de gestión y administración para la base de datos phpMyAdmin.

- Basados en la arquitectura del proveedor del caso de estudio, se ha logrado implementar un prototipo de automatización de la configuración de enlaces en la capa de Red del CSP CNT EP, donde se pudo evidenciar una reducción de tiempo en la configuración, administración y gestión de los recursos de Red.
- El prototipo está basado y desarrollado con herramientas de código libre reduciendo los gastos de Capex en temas de licenciamiento.
- El prototipo ha estandarizado las configuraciones de los enlaces reduciendo fallas operativas, ya que ejecuta y almacena la información apoyando a las tareas administrativas y de gestión de la plataforma de Red de los servicios de Computación en la Nube.
- El uso del prototipo ha brindado la base para la adopción de SDN (Redes definidas por Software) en el CSP del caso de estudio, para que pueda programar y administrar sus recursos de red a través de un controlador o software externo, brindando una provisión bajo demanda de servicios de red. Por otro lado se continuará trabajando en mejorar este prototipo que al estar programado bajo Python tiene ventajas de integración con API's de varias soluciones SDN.
- El tiempo que tomaba configurar un enlace era de 1 a 2 horas en la plataforma de Red de Computación en la Nube, tomando en cuenta las pruebas y solución

de problemas entre las diversas áreas que administran todos los componentes de la cadena de servicio. Con la ejecución del prototipo el tiempo promedio de configuración de un nuevo enlace es de 5 mins, esto reduce al CSP gastos de OPEX y mejora el tiempo de respuesta ante requerimientos del cliente final brindando un servicio de calidad.

- La implementación de soluciones y servicios de Computación en la Nube aporta muchas ventajas para varios sectores en nuestro medio, basándose en su modelo de negocio bajo demanda y movilidad.
- Desde el punto de vista del cliente o suscriptor, adquirir un servicio de Computación en la Nube no requiere asumir altos costos de inversión, la solución contratada es flexible y escalable e impacta favorablemente en el medio ambiente debido a la reducción y optimización en el consumo de energía y recursos.
- Desde el punto de vista del proveedor de servicios en la Nube, se aprecian varios retos hacia una evolución, mejorando la disponibilidad del servicio, la búsqueda de estandarizar e integrar soluciones tecnológicas, seguridad y privacidad de datos, búsqueda de socios estratégicos de negocio, reducción de sus costes de OPEX y CAPEX aprovechando las economías de escala, interoperabilidad, gobernanza y una legislación adecuada.
- La importancia que tienen las Telecomunicaciones en el servicio de Computación en la Nube es el principal factor de éxito de esta tecnología. Los proveedores de telecomunicaciones deben garantizar conectividad y

rendimiento, tomando en cuenta el desarrollo de servicios de Nube bajo sus principales modelos de negocio IaaS, PaaS y SaaS los cuales se encuentran condicionados por la capacidad y disponibilidad de las redes de telecomunicaciones fijas y móviles.

- MPLS es una evolución de conmutación multinivel que brinda interconexión a la plataforma de Nube hacia redes como: Internet, enlaces de datos, redes móviles, telefonía fija y otras, permitiendo cumplir uno de los principios de la Nube que está basado en la movilidad y acceso desde cualquier sitio.
- La base de la Computación en la Nube es la virtualización y se identifica una arquitectura de referencia para Centros de Datos Virtuales del proveedor del caso de estudio basada en una capa inferior conformada por los hipervisores, una capa intermedia que cuenta con un administrador y gestor de la capa de hipervisores y una capa superior que permite el acceso multi-usuario, recomendado por líderes de estas soluciones.
- La arquitectura de referencia del Centro de Datos del CSP está basada en el modelo de tres niveles propuesto y validado por fabricantes líderes en soluciones de conectividad para Centro de Datos, utilizando equipo de “clase proveedor de servicios” de alto rendimiento y performance donde uno de los diferenciadores es que son capaces de soportar arquitecturas SAN, LAN, vPC y Multi-tenant en una misma red gracias a sus componentes de hardware y sistemas operativos diseñados para estas arquitecturas.

Como recomendaciones podemos señalar las siguientes:

- De acuerdo al análisis realizado, se puede evidenciar que el uso del protocolo OSPF para la interconexión entre el CSP y la MPLS (CE – PE) presenta ventajas en bajo consumo de recursos en los equipos de borde del servicio de Cloud y una rápida convergencia, pero tiene una limitante en cuanto a las rutas que puede manejar, ya que según fabricante son entre 15000 y 50000 dependiendo de la versión de sistema operativo, esto no brinda escalabilidad para un servicio de Nube, por lo que una solución que se recomienda es que los CE's del CSP se conviertan en PE's formando parte de la MPLS y utilicen como protocolo de enrutamiento BGP, con este protocolo se puede manejar hasta 3 millones de rutas según fabricante, importante tomar en cuenta el procesamiento del equipo ya que el uso de un protocolo especializado como BGP consume altos recursos y además es más lento en la convergencia.
- El Centro de Datos desempeña un papel importante en la expansión de capacidades para el CSP que cuenta con una arquitectura tradicional de tres niveles (Core, Agregación y Acceso) donde el tráfico es apilado en múltiples flujos en un solo enlace y se interconectan todos los dispositivos de nivel inferior a un solo dispositivo de nivel superior, transportando los flujos en serie en lugar de hacerlo en paralelo. Por esta razón se desarrolló un nuevo diseño de Centro de Datos llamado “spine and leaf” donde el tráfico va a lo largo de múltiples rutas proporcionando suficientes conexiones cruzadas permitiendo que cualquier dispositivo alcance a cualquier otro colocando múltiples flujos de

tráfico en paralelo, por lo que se recomienda que para servicios de Nube se opte por adoptar esta nueva arquitectura que ofrece conectividad de servidor a servidor con gran ancho de banda, baja latencia y sin bloque de puertos.

- Se recomienda la implementación de tecnologías SDN que permiten virtualizar funciones de red, ayudan a automatizar y a simplificar la administración de la red del CSP, brindando una amplia red programable que provee auto-aprovisionamiento de recursos de red y servicios de red administrados.

## 10. Bibliografía

- Centro de estudios de telecomunicaciones de América Latina. (Noviembre de 2014). *cet.la*.  
Obtenido de <http://cet.la/blog/course/computacion-en-la-nube-desafio-y-oportunidad-en-la-sociedad-conectada/>
- Cisco Networking Academy. (2014). *Routing Protocols Companion Guide*.
- CISCO SYSTEMS, INC. (2010). *Data Center Design with Cisco Nexus Switches and Virtual PortChannel: Overview*. Obtenido de Design Guide:  
[http://www.cisco.com/c/dam/en/us/products/collateral/switches/nexus-7000-series-switches/C07-572831-00\\_Dsgn\\_Nexus\\_vPC\\_DG.pdf](http://www.cisco.com/c/dam/en/us/products/collateral/switches/nexus-7000-series-switches/C07-572831-00_Dsgn_Nexus_vPC_DG.pdf)
- CISCO SYSTEMS, INC. (2013). *Guía de diseño de OSPF*. Obtenido de  
[http://www.cisco.com/cisco/web/support/LA/7/73/73214\\_1.html#intro](http://www.cisco.com/cisco/web/support/LA/7/73/73214_1.html#intro)
- CISCO SYSTEMS, INC. (2014). *Cisco NX-OS Software Virtual PortChannel: Fundamental Concepts 5.0*. Obtenido de  
[http://www.cisco.com/c/en/us/products/collateral/switches/nexus-5000-series-switches/design\\_guide\\_c07-625857.html#\\_Toc271759437](http://www.cisco.com/c/en/us/products/collateral/switches/nexus-5000-series-switches/design_guide_c07-625857.html#_Toc271759437)
- CISCO SYSTEMS, INC. (14 de Abril de 2015). *Cisco Nexus 7000 Series NX-OS Interfaces Configuration Guide, Release 5.x*. Obtenido de Chapter: Configuring Layer 3 Interfaces: [http://www.cisco.com/c/en/us/td/docs/switches/datacenter/sw/5\\_x/nx-os/interfaces/configuration/guide/if\\_cli/if\\_layer3int.html#48044](http://www.cisco.com/c/en/us/td/docs/switches/datacenter/sw/5_x/nx-os/interfaces/configuration/guide/if_cli/if_layer3int.html#48044)
- CISCO SYSTEMS, INC. (27 de Julio de 2015). *Cisco Nexus 7000 Series NX-OS Unicast Routing Configuration Guide, Release 4.x*. Obtenido de  
[http://www.cisco.com/c/en/us/td/docs/switches/datacenter/sw/4\\_2/nx-os/unicast/configuration/guide/l3\\_cli\\_nxos/l3\\_ospf.html#50266](http://www.cisco.com/c/en/us/td/docs/switches/datacenter/sw/4_2/nx-os/unicast/configuration/guide/l3_cli_nxos/l3_ospf.html#50266)

CISCO SYSTEMS, INC. (27 de Julio de 2015). *Cisco Nexus 7000 Series NX-OS Unicast Routing Configuration Guide, Release 4.x*. Obtenido de Chapter: Configuring

OSPFv2: [http://www.cisco.com/c/en/us/td/docs/switches/datacenter/sw/4\\_2/nx-os/unicast/configuration/guide/l3\\_cli\\_nxos/l3\\_ospf.html#41886](http://www.cisco.com/c/en/us/td/docs/switches/datacenter/sw/4_2/nx-os/unicast/configuration/guide/l3_cli_nxos/l3_ospf.html#41886)

CISCO SYSTEMS, INC. (25 de Agosto de 2015). *cisco.com*. Obtenido de

[http://www.cisco.com/cisco/web/support/LA/102/1023/1023775\\_mpls\\_faq\\_4649.pdf](http://www.cisco.com/cisco/web/support/LA/102/1023/1023775_mpls_faq_4649.pdf)

CISCO SYSTEMS, INC. (2015). *Data Center Switching Solutions White Paper*. Obtenido de

[http://www.cisco.com/c/en/us/solutions/collateral/data-center-virtualization/data-center-switching/net\\_implementation\\_white\\_paper0900aecd8053495a.html](http://www.cisco.com/c/en/us/solutions/collateral/data-center-virtualization/data-center-switching/net_implementation_white_paper0900aecd8053495a.html)

CISCO SYSTEMS, INC. (2015). *Design Zone for Data Center Networking*. Obtenido de

<http://www.cisco.com/c/en/us/solutions/enterprise/data-center-designs-data-center-networking/index.html#~designs>

CISCO SYSTEMS, INC. (26 de Septiembre de 2016). *Cisco Nexus 5000 Series NX-OS*

*Software Configuration Guide*. Obtenido de Chapter: Configuring VLANs:

<http://www.cisco.com/c/en/us/td/docs/switches/datacenter/nexus5000/sw/configuration/guide/cli/CLIConfigurationGuide/VLANs.html>

CISCO SYSTEMS, INC. (29 de Febrero de 2016). *Cisco Nexus 7000 Series NX-OS Unicast Routing Configuration Guide, Release 5.x*. Obtenido de

[http://www.cisco.com/c/en/us/td/docs/switches/datacenter/sw/5\\_x/nx-os/unicast/configuration/guide/l3\\_cli\\_nxos/l3\\_hsrp.html](http://www.cisco.com/c/en/us/td/docs/switches/datacenter/sw/5_x/nx-os/unicast/configuration/guide/l3_cli_nxos/l3_hsrp.html)

CISCO SYSTEMS, INC. (29 de Febrero de 2016). *Cisco Nexus 7000 Series NX-OS Unicast Routing Configuration Guide, Release 5.x*. Obtenido de Chapter: Configuring Layer 3

Virtualization: [http://www.cisco.com/c/en/us/td/docs/switches/datacenter/sw/5\\_x/nx-os/unicast/configuration/guide/l3\\_cli\\_nxos/l3\\_virtual.html](http://www.cisco.com/c/en/us/td/docs/switches/datacenter/sw/5_x/nx-os/unicast/configuration/guide/l3_cli_nxos/l3_virtual.html)

- CISCO SYSTEMS, INC. (29 de Febrero de 2016). *Cisco Nexus 7000 Series NX-OS Unicast Routing Configuration Guide, Release 5.x*. Obtenido de Chapter: Configuring HSRP: [http://www.cisco.com/c/en/us/td/docs/switches/datacenter/sw/5\\_x/nx-os/unicast/configuration/guide/l3\\_cli\\_nxos/l3\\_hsrp.html#76351](http://www.cisco.com/c/en/us/td/docs/switches/datacenter/sw/5_x/nx-os/unicast/configuration/guide/l3_cli_nxos/l3_hsrp.html#76351)
- CNT E.P. (2016). *CNT - CLOUD*. Obtenido de <https://vdc.cntcloud.com>
- Corporación Nacional de Telecomunicaciones CNT EP. (2016). *Data Center Virtual*. Obtenido de <http://cntempresas.com/data-center-virtual-cloud/>
- DB-Engines. (2016). *Knowledge Base of Relational and NoSQL Database Management Systems*. Obtenido de <http://db-engines.com/en/ranking>
- Debian GNU/Linux. (2015). *Guía de instalación de Debian GNU/Linux*. Obtenido de <https://www.debian.org/releases/stable/mips/ch01s02.html.es>
- Fast Lane Consulting and Education Services, Inc. (2014). *Unified Data Center Test Drive Modular: Nexus 2-7K (UDCTD) Module 3- NXOS Software*.
- Fuller, R., Jansen, D., & Mc Pherson, M. (2013). *NX-OS and Cisco Nexus Switching Next-Generation Data Center Architectures Second Edition*. Indianapolis: Cisco Press.
- Gartner, Inc. (2016). *gartner.com*. Obtenido de <http://www.gartner.com/it-glossary/cloud-computing>
- Gartner, Inc. (2016). *Magic Quadrant for Data Center Networking*. Obtenido de <https://www.gartner.com/doc/reprints?ct=160517&id=1-36UFTCZ&st=sb>
- Ghein, L. D. (2007). *MPLS Fundamentals*. Indianapolis: Cisco Press.
- González, A. J., & Teng, Y. (2015). *Multi-Protocol Label Switching*. Obtenido de Repositorio Profesores Departamento de Electrónica, Universidad Técnica Federico Santa María, Chile: <http://profesores.elo.utfsm.cl/~agv/elo323/2s06/lectures/MPLS.ppt>
- Hidalgo, P. (2013). *Multiprotocol Label Switching (MPLS)*. Ecuador.

IEEE Spectrum. (2016). *The 2016 Top Programming Languages*. Obtenido de <http://spectrum.ieee.org/computing/software/the-2016-top-programming-languages>

Instituto Nacional de Estándares y Tecnología (NIST). (Septiembre de 2011). *nist.gov*. Obtenido de <http://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-145.pdf>

ISO/IEC 17788. (2014). *iso.org*. Obtenido de Information technology -- Cloud computing -- Overview and vocabulary: [http://www.iso.org/iso/home/store/catalogue\\_tc/catalogue\\_detail.htm?csnumber=60544](http://www.iso.org/iso/home/store/catalogue_tc/catalogue_detail.htm?csnumber=60544)

ITpreneurs. (2015). *Cloud Essentials Course Book*. Nederland B.V.

Linux Kernel Organization, Inc. (2016). *The Linux Kernel Archives*. Obtenido de <https://www.kernel.org/>

Linux.com News for the Open Source Professional by The Linux Foundation. (2016). *The Best Linux Distros of 2016*. Obtenido de <https://www.linux.com/NEWS/BEST-LINUX-DISTROS-2016>

Opensource.com supported by Red Hat. (2014). *What is open source?* Obtenido de <https://opensource.com/resources/what-open-source>

Oracle Corporation. (2016). *MySQL Community*. Obtenido de <https://dev.mysql.com/downloads/mysql/>

Oracle MySQL. (2016). *Las 10 razones principales para usar MySQL como base de datos integrada*. Obtenido de <https://www.mysql.com/why-mysql/white-papers/las-10-razones-principales-para-usar-mysql-como-base-de-datos-integrada/>

Paramiko. (2016). *Installing*. Obtenido de A Python implementation of SSHv2: <http://www.paramiko.org/installing.html>

phpMyAdmin. (2016). *phpMyAdmin*. Obtenido de <https://www.phpmyadmin.net/>

Python Software Foundation. (2016). *PEP 249 -- Python Database API Specification v2.0*.

Obtenido de <https://www.python.org/dev/peps/pep-0249/>

Python Software Foundation. (2016). *Python*. Obtenido de <https://www.python.org/>

Python Software Foundation. (2016). *Python Software Foundation*. Obtenido de

<https://www.python.org/psf/>

Python Software Foundation. (2016). *The Python Standard Library*. Obtenido de

<https://docs.python.org/2/library/index.html>

Tanenbaum, A. S. (2012). Redes de Computadoras. En A. S. Tanenbaum, & D. J. Wetherrall,

*Redes de Computadoras* (págs. 241-256). Mexico: Pearson Educación.

The Internet Society. (2001). *Multiprotocol Label Switching Architecture, IETF, Request for*

*Comments: 3031*. Obtenido de <https://tools.ietf.org/html/rfc3031>

The phpMyAdmin devel team. (2016). <https://docs.phpmyadmin.net/es/latest/intro.html>.

Obtenido de <https://docs.phpmyadmin.net/es/latest/intro.html>

Ubuntu Server. (2016). *Ubuntu*. Obtenido de <http://www.ubuntu.com/server>

Unsigned Integer Limited, DistroWatch.com. (2016). *Top Ten Distributions*. Obtenido de An

overview of today's top distributions:

<http://distrowatch.com/dwres.php?resource=major>

VMware, Inc. (2016). *Knowledge Base*. Obtenido de

[https://kb.vmware.com/selfservice/microsites/search.do?language=en\\_US&cmd=displayKC&externalId=1022525](https://kb.vmware.com/selfservice/microsites/search.do?language=en_US&cmd=displayKC&externalId=1022525)

VMware, Inc. (2016). *Software-Defined Datacenter – In Depth*.

VMware, Inc.® VMware vCloud Director. (2016). *vCloud Suite*. Obtenido de

<http://www.vmware.com/products/vcloud-suite.html>

## **11. Acrónimos**

Multi-Protocol Label Switching (MPLS)

Infraestructura como Servicio (IaaS)

Redes Definidas por Software (SDN)

Provider Edge (PE)

Customer Edge (CE)

Capital Expenditure (CAPEX)

National Institute of Standards and Technology (NIST)

Virtual Private Networking (VPN)

Ingeniería de tráfico (TE)

Calidad del servicio (QoS)

Open Shortest Path First (OSPF)

Virtual LAN (VLAN)

Wide Area Network (WAN)

Local Area Network (LAN)

Virtual Routing and Forwarding (VRF)

Switch virtual interface (SVI)

Hot Standby Router Protocol (HSRP)

Information Technology (IT)

Internet Engineering Task Force (IETF)

Request for Comments (RFC)

Open Systems Interconnection (OSI)

International Organization for Standardization (ISO)

International Electrotechnical Commission (IEC)

Time to Live (TTL)

Class of Service (CoS)

Label Switched Router (LSR)

Label Edge Router (LER)

Forwarding Equivalence Class (FEC)

Label Switched Path (LSP)

Label Distribution Protocol (LDP)

Virtual Data Center (VDC)

Interior Gateway Protocol (IGP)

Link State Advertisements (LSA)

ABR (Area Border Routers)

Autonomous System Boundary Router (ASBR)

Designated Router (DR)

Backup Designated Router (BDR)

Not-So-Stubby Area (NSSA)

Switch Virtual Interface (SVI)

Virtual Routing and Forwarding (VRF)

Hot Standby Router Protocol (HSRP)

Spanning Tree Protocol (STP)

Virtual PortChannels (vPCs)

Virtual eXtensible Local Area Network (VXLAN)

Storage Area Network Operating System (SAN-OS)

Internetwork Operating System (IOS)

Catalyst Operating System (CatOS)

Application Programming Interface (API)

Independent Software Vendor (ISV)

Original Equipment Manufacturer (OEM)

## 12. Anexos

### ANEXO 1

#### CONFIGURACIÓN DE LOS COMPONENTES DE LA SOLUCIÓN

##### Sección I: Creación de Máquinas Virtuales en el portal de Gestión del CSP.

El acceso al Centro de Datos Virtual se lo realiza a través del portal de gestión con las credenciales proporcionadas por el CSP.

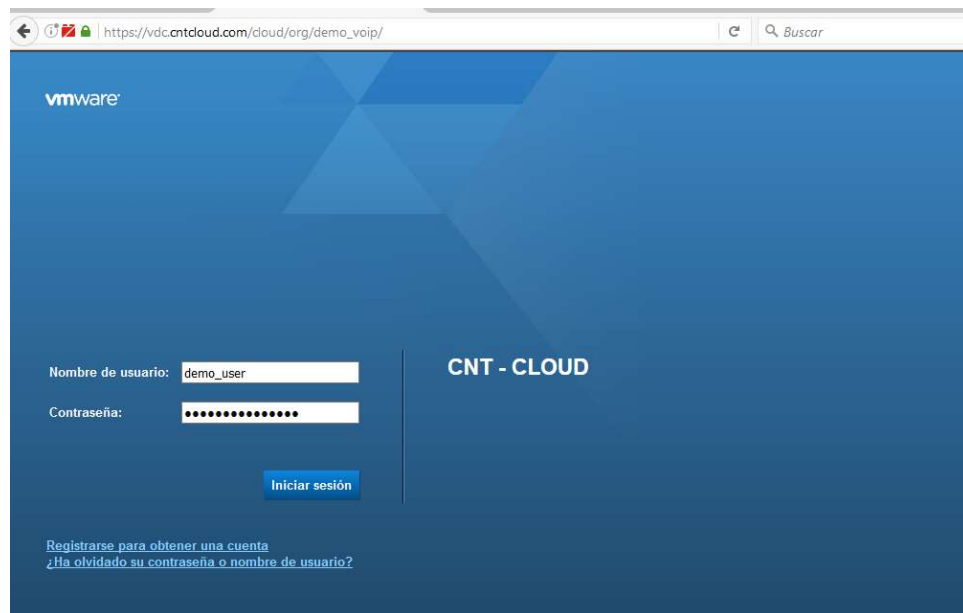


Figura 23. Acceso al portal de Gestión del VDC (VMware, Inc.® VMware vCloud Director, 2016)  
(Corporación Nacional de Telecomunicaciones CNT EP, 2016)

Una vez que se accedió a su VDC, se procede con la creación de “vApps” que son contenedores que agrupan varias máquinas virtuales que forman parte de un servicio, redes y recursos. Una vApp puede manejar tres tipos de esquema para crear una red, descritos a continuación:

Interna: Donde solo existe comunicación entre las máquinas virtuales que pertenecen a la vApp.

Externa Enrutada: Existe comunicación entre las máquinas virtuales que pertenecen a la vApp y a través de un componente de firewall propio de VMware

conocido como vShield puede realizar tareas como NAT desde la red interna hacia una red externa.

Externa Directamente Conectada: Existe comunicación entre las máquinas virtuales que pertenecen a la vApp y una red externa sin utilizar vShield.

Estas redes ya han sido aprovisionadas en el ambiente virtual por el CSP.

Nueva vApp

Seleccionar nombre y ubicación

Una vApp es un sistema informático en nube que contiene una o varias máquinas virtuales. Describa esta vApp y configure su centro de datos virtual y concesiones.

Nombre: sdnserver \*

Descripción:

Centro de datos virtual

Seleccione el centro de datos virtual (VDC) en el que se almacenará esta vApp y donde se ejecutará cuando se inicie.

vDC\_DEMO\_SDN

Concesiones

Concesión de tiempo de ejecución: Nunca caduca Horas \*

Tiempo que se puede ejecutar la vApp antes de detenerse automáticamente.

Concesión de almacenamiento: Nunca caduca Horas \*

Cuando la vApp se detiene, tiempo que está disponible antes de limpiarse de manera automática.

Atrás Siguiente Finalizar Cancelar

Figura 24. Creación de una vAPP (VMware, Inc.® VMware vCloud Director, 2016)

Luego de tener las vApp's, se crean las máquinas virtuales de acuerdo a la Figura 25, aquí se asignan los recursos a cada máquina virtual, en este caso se describen los recursos en la Tabla 8.

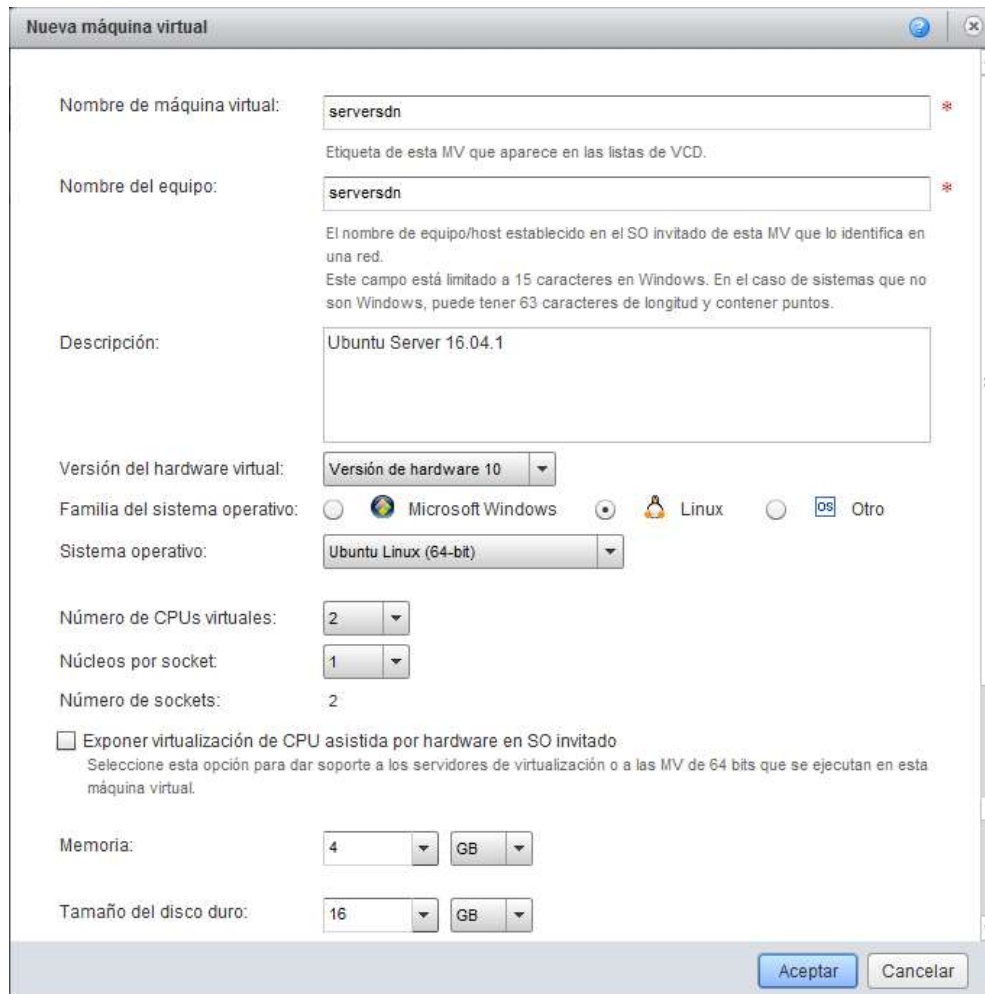


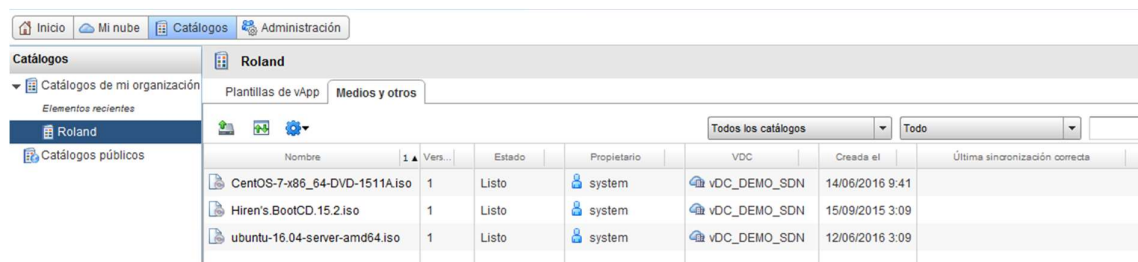
Figura 25. Creación de una máquina virtual (VMware, Inc.® VMware vCloud Director, 2016)

Maquina Virtual	Sistema Operativo	vAPP	Procesamiento	Memoria	Almacenamiento
serversdn	Ubuntu Server	sdnserver	2 vCPU	4 GB	16 GB
gestion_temporal	Lubuntu	vAPP_seg_nube	2 vCPU	2 GB	20 GB
Firewall_demo	Suse	vAPP_seg_nube	2 vCPU	4 GB	40 GB

Tabla 8. Lista de Máquinas Virtuales implementadas en el VDC

## Sección II: Instalación del Sistema Operativo Linux Ubuntu Server en una Máquina Virtual del Centro de datos Virtual sobre el portal de Gestión del CSP.

El proceso de instalación para un Sistema Operativo se lo realiza de forma tradicional que un ambiente físico, se requiere de una imagen que es subida a un catálogo del Data Center Virtual, desde el que se arranca el instalador, el cual debe ser insertado en la máquina virtual para iniciar la instalación.



Nombre	Vers...	Estado	Propietario	VDC	Creada el	Última sincronización correcta
CentOS-7-x86_64-DVD-1511A.iso	1	Listo	system	vDC_DEMO_SDN	14/06/2016 9:41	
Hiren's BootCD 15.2.iso	1	Listo	system	vDC_DEMO_SDN	15/09/2015 3:09	
ubuntu-16.04-server-amd64.iso	1	Listo	system	vDC_DEMO_SDN	12/08/2016 3:09	

Figura 26. Catálogos de medios VDC (VMware, Inc.® VMware vCloud Director, 2016)

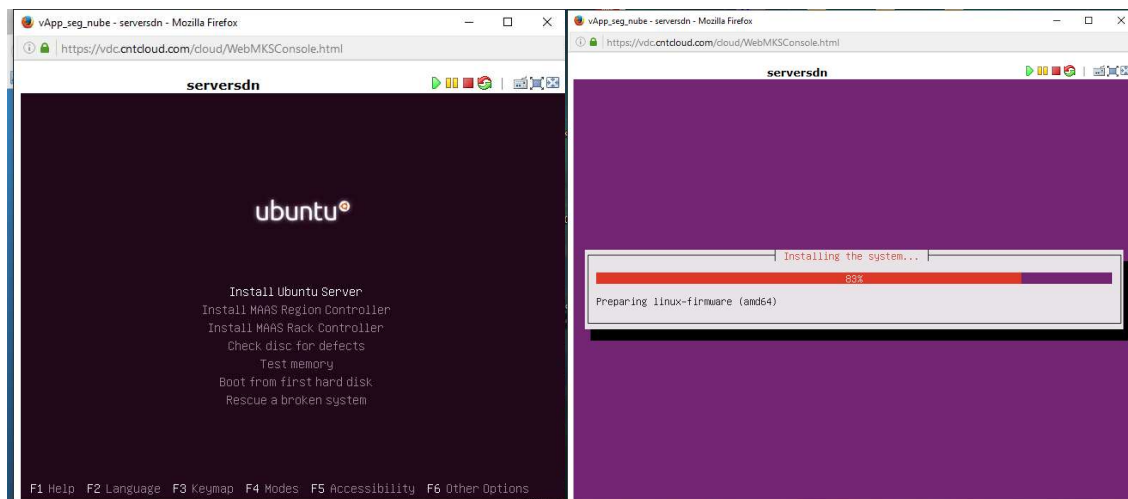


Figura 27. Arranque del Instalador Sistema Operativo Ubuntu Server (Ubuntu Server, 2016)

Luego de realizar la configuración de requisitos para la instalación del sistema operativo tales como idioma, configuración regional, red y particiones se procede a instalar el paquete con la opción LAMP (Linux, Apache, MySQL y PHP). Para finalizar la instalación ejecute una actualización de paquetes sobre el servidor con el comando “sudo apt-get update”<sup>53</sup>, esto permite que el servidor este al día con las últimas versiones de utilitarios y herramientas de Ubuntu Server.

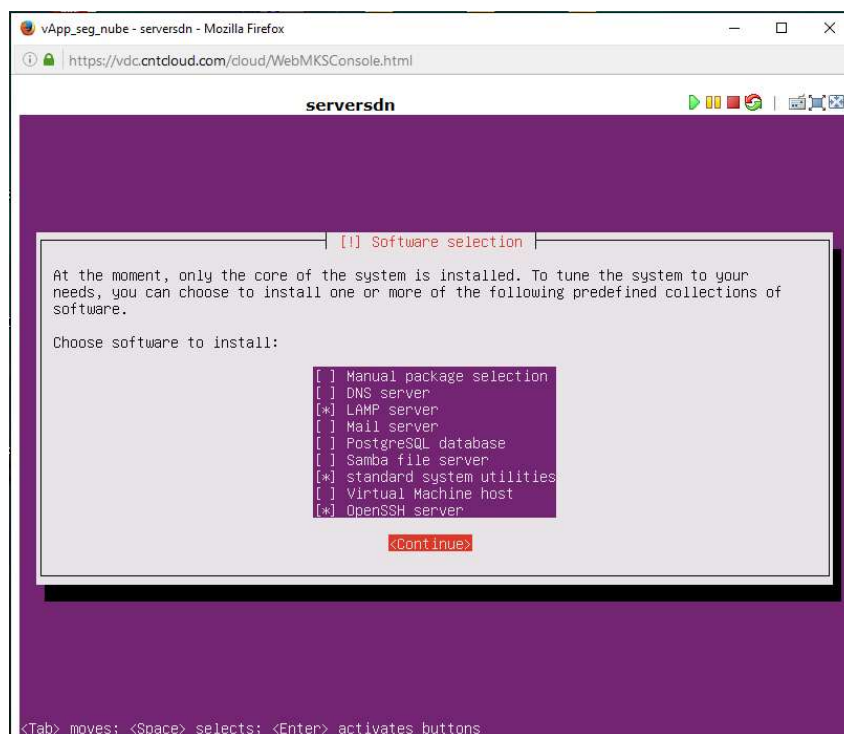


Figura 28. Instalación de LAMP en Ubuntu Server (Ubuntu Server, 2016)

<sup>53</sup> Comando sudo apt-get update: Este comando se compone de “sudo” que viene del término inglés “super user do” permite la ejecución de un programa con privilegios de un usuario administrador “root” para el caso de Linux. El comando “apt-get update” instala y actualiza paquetes desde un repositorio de Linux resolviendo dependencias.

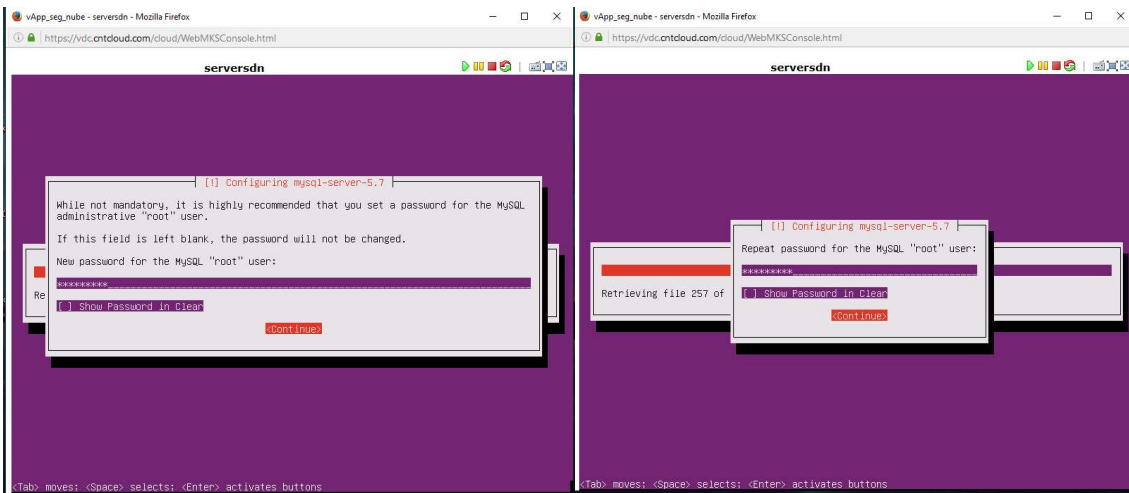


Figura 29. Configuración MySQL (Ubuntu Server, 2016)

Como paso final se recomienda instalar VMware Tools<sup>54</sup> para mejorar el rendimiento de la máquina virtual. En el servidor Ubuntu a través de línea de comandos se ejecuta el comando “sudo ./vmware-install.pl” para instalar este paquete, esta configuración está basada en la base de conocimientos del fabricante, VMware, Inc, (2016).

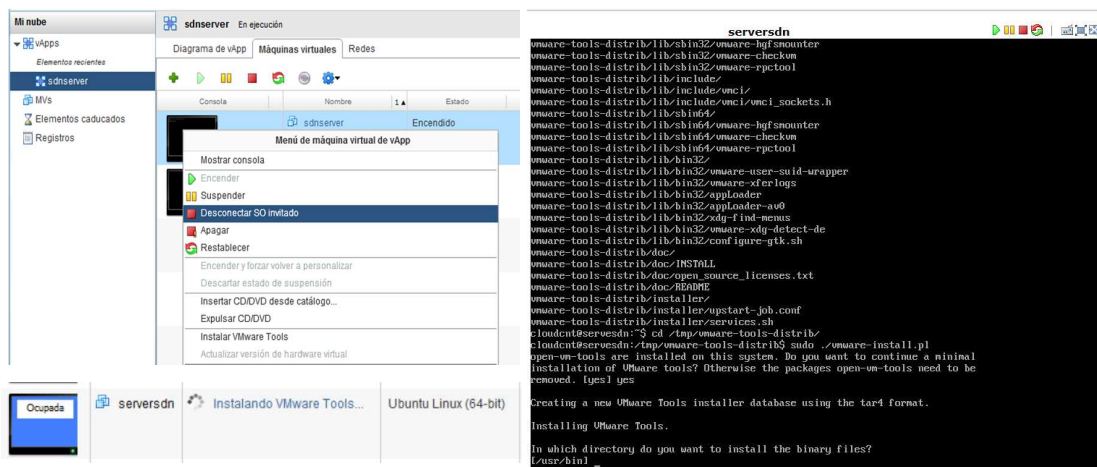


Figura 30. Instalación de VMware Tools Servidor Ubuntu (VMware, Inc.® VMware vCloud Director, 2016)

<sup>54</sup> VMware Tools: Es un conjunto de controladores (drivers) y utilitarios que mejoran el rendimiento de la máquina virtual y su sistema operativo.

### Sección III: Instalación del administrador phpMyAdmin para la de Base de Datos MySQL en Linux Ubuntu Server.

Una vez instalado LAMP, se procede con la instalación del administrador y gestor de MySQL, phpMyAdmin. En línea de comandos se ejecuta “sudo apt-get install phpmyadmin”<sup>55</sup>. De inmediato se despliega el paquete de configuración de phpMyAdmin solicitando se confirme el servidor web a utilizar que para este caso es Apache, como paso final se asigna la contraseña de administrador de MySQL para la conexión de phpMyAdmin.

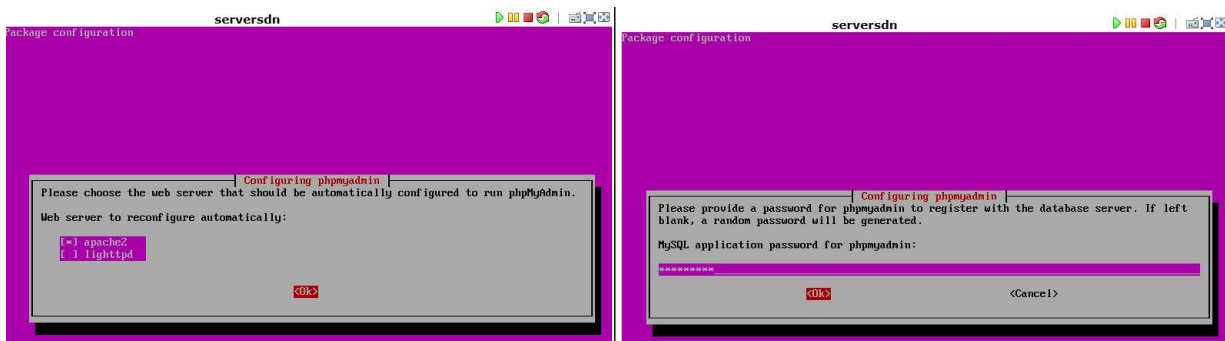


Figura 31. Instalación y configuración de phpMyAdmin (phpMyAdmin, 2016)

Se accede al portal de gestión de phpMyAdmin para validar su instalación de un navegador, con la dirección:

<http://ipserver/phpmyadmin>

Donde “ipserver” es la dirección del servidor Ubuntu, se procede ingresando las credenciales configuradas en la instalación y accedemos al portal.

---

<sup>55</sup> Comando sudo apt-get install phpmyadmin: Instala los paquetes de phpmyadmin con privilegios de administrador.

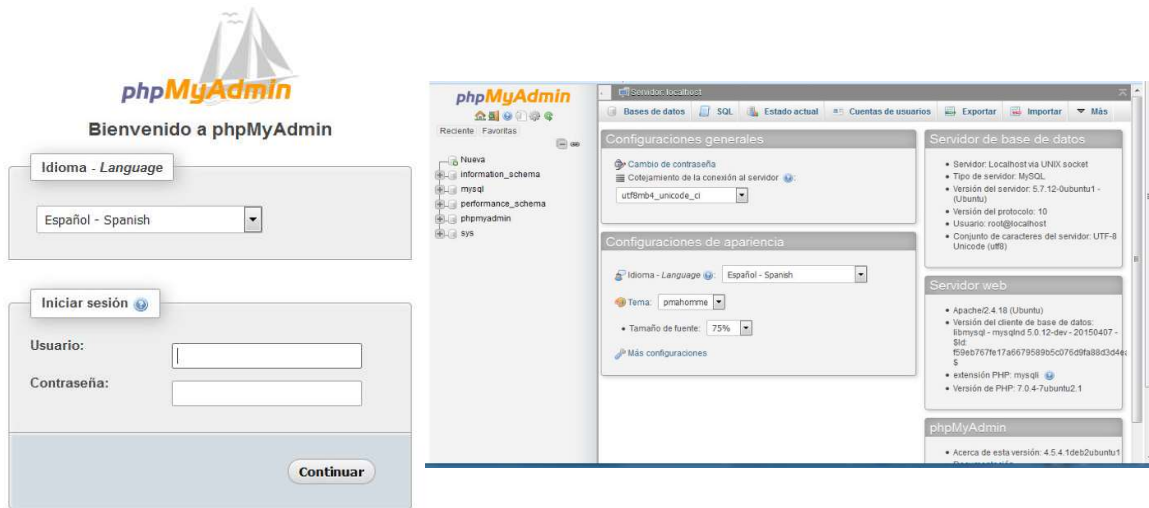


Figura 32. Portal de phpMyAdmin (phpMyAdmin, 2016)

#### Sección IV: Instalación del Lenguaje de Programación Python en Linux Ubuntu Server.

Desde la línea de comandos en el servidor Ubuntu ejecutamos “sudo apt-get install python”<sup>56</sup> y continuamos con el proceso de instalación aceptando la descarga e instalación de dependencias de paquetes.

```
cloudcmt@sdnserver:~$ sudo apt-get install python
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  libpython-stdlib libpython2.7-minimal libpython2.7-stdlib python-minimal python2.7
  python2.7-minimal
Suggested packages:
  python-doc python-tk python2.7-doc binfmt-support
The following NEW packages will be installed:
  libpython-stdlib libpython2.7-minimal libpython2.7-stdlib python python-minimal python2.7
  python2.7-minimal
0 upgraded, 7 newly installed, 0 to remove and 0 not upgraded.
Need to get 3,868 kB of archives.
After this operation, 16.3 MB of additional disk space will be used.
Do you want to continue? [Y/n] y
```

Figura 33. Instalación de Python (El Autor, 2016)

<sup>56</sup> Comando sudo apt-get install python: Instala el lenguaje de programación Python con privilegios de administrador.

Para validar la instalación ejecutamos el comando “python” y podemos validar al versión que es la 2.7.12, para salir de esta consola se puede ejecutar la sentencia “exit()”.

```
cloudcnt@sdnserver:~$ python
Python 2.7.12 (default, Nov 19 2016, 06:48:10)
[GCC 5.4.0 20160609] on linux2
Type "help", "copyright", "credits" or "license" for more information.
>>> _
```

Figura 34. Consola de programación de Python (El Autor, 2016)

Python brinda la facilidad de instalar módulos y librerías que pueden utilizarse para ejecutar programas, el prototipo diseñado ha aprovechado estas características.

Los módulos y librerías principales con los que se realizaron pruebas en el código y fueron utilizados son: Paramiko, sys, os, getpass, time, ipcalc, netaddr, MySQLdb.

De acuerdo con Paramiko (2016), el módulo Paramiko se basa en la implementación del protocolo SSH v257 tanto para cliente como servidor, esta librería aprovecha una extensión de criptografía<sup>58</sup> de Python y brinda una interfaz pura basada en conceptos para redes que utilizan SSH y es la base de la conexión del prototipo hacia los equipos de Networking del Servicio de Computación en la Nube.

Basados en Python Software Foundation (2016) se utilizó:

**SYS:** Es un módulo que brinda acceso a variables y funciones utilizadas por Python.

**OS:** Módulo que permite acceder a funcionalidades dependientes del Sistema Operativo, como referir información sobre el entorno y rutas del mismo y nos permiten manipular la estructura de directorios (lectura y escritura de archivos).

---

<sup>57</sup> Protocolo SSH: Está definido en RFC4251, 4252, 4253 y 4254. Protocolo utilizado para acceso remoto, la implementación principal del protocolo está a cargo del proyecto OpenSSH.

<sup>58</sup> Criptografía: Técnica que resguarda datos a través de cifrado o códigos.

GETPASS: Solicita una contraseña al usuario ocultando los caracteres.

TIME: Funciones relacionadas con el tiempo.

IPCALC: Librería que realiza cálculos para definir una red, segmento de red o

IP.

Para instalar un módulo o librería se lo realiza por línea de comandos, por ejemplo se instala Paramiko ejecutando el comando: “sudo apt-get install python-paramiko”.

```
cloudcnt@sdnserver:~/codigo$ sudo apt-get install python-paramiko
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  python-crypto python-ecdsa python-six
Suggested packages:
  python-crypto-dbg python-crypto-doc
The following NEW packages will be installed:
  python-crypto python-ecdsa python-paramiko python-six
0 upgraded, 4 newly installed, 0 to remove and 0 not upgraded.
Need to get 398 kB of archives.
After this operation, 2,269 kB of additional disk space will be used.
Do you want to continue? [Y/n] _
```

Figura 35. Instalación del módulo Paramiko en Python (El Autor, 2016)

Python tiene definido el acceso a base de datos de manera estándar de acuerdo a las especificaciones del API<sup>59</sup> para base de datos DB-API de acuerdo con Python Software Foundation (2016) PEP 249, esto permite que al utilizar Python, sea independiente la base de datos que se utilice, la conexión, métodos, lectura y escritura de datos. De acuerdo a la Figura 36, se debe instalar los módulos de la librería “MySQLdb” para empezar a trabajar con la base de datos.

---

<sup>59</sup> Application Programming Interfaz API: La Interfaz de Programación de Aplicaciones es un conjunto de funciones y procedimientos que pueden ser utilizadas por otro software, permitiendo una interacción y comunicación entre distintos sistemas y programas.

```

cloudcnt@sdnserver:~/codigo$ sudo apt-get install python-mysqldb
[sudo] password for cloudcnt:
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following packages were automatically installed and are no longer required:
  linux-headers-4.4.0-45 linux-headers-4.4.0-45-generic linux-image-4.4.0-45-generic linux-image-extra-4.4.0-45-generic
Use 'sudo apt autoremove' to remove them.
The following additional packages will be installed:
  libmysqlclient20
Suggested packages:
  python-egenix-mxdatetime python-mysqldb-dbg
The following NEW packages will be installed:
  libmysqlclient20 python-mysqldb
0 upgraded, 2 newly installed, 0 to remove and 10 not upgraded.
Need to get 852 kB of archives.
After this operation, 4,523 kB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://ec.archive.ubuntu.com/ubuntu xenial-updates/main amd64 libmysqlclient20 amd64 5.7.16-0ubuntu0.16.04.1 [809 kB]
Get:2 http://ec.archive.ubuntu.com/ubuntu xenial/main amd64 python-mysqldb amd64 1.3.7-1build2 [42.4 kB]
Fetched 852 kB in 1s (602 kB/s)
Selecting previously unselected package libmysqlclient20:amd64.
(Reading database ... 161845 files and directories currently installed.)
Preparing to unpack .../libmysqlclient20_5.7.16-0ubuntu0.16.04.1_amd64.deb ...
Unpacking libmysqlclient20:amd64 (5.7.16-0ubuntu0.16.04.1) ...
Selecting previously unselected package python-mysqldb.
Preparing to unpack .../python-mysqldb_1.3.7-1build2_amd64.deb ...
Unpacking python-mysqldb (1.3.7-1build2) ...
Processing triggers for libc-bin (2.23-0ubuntu4) ...
Setting up libmysqlclient20:amd64 (5.7.16-0ubuntu0.16.04.1) ...
Setting up python-mysqldb (1.3.7-1build2) ...
Processing triggers for libc-bin (2.23-0ubuntu4) ...
cloudcnt@sdnserver:~/codigo$

```

Figura 36. Instalación de la librería MySQLdb (El Autor, 2016)

## ANEXO 2

### CÓDIGO FUENTE DEL PROTOTIPO

#### Sección I: Código Fuente del prototipo del programa realizado en el lenguaje de Programación Python en Linux Ubuntu Server.

A continuación se detalla el código fuente del prototipo hecho en Python versión 2.7.12:

```

#####Librerias
import paramiko
import sys
import os
import getpass
import subprocess
import time
import re
import MySQLdb
from ipcalc import *
from netaddr import *
from progressbar import ProgressBar

#####
### PROTOTIPO : AUTOMAZACION DE ENLACES CLOUD VERSION 1.0 #####
### CREADO POR: JUAN ROLANDO CHICAIZA ACOSTA #####
### DERECHOS DE AUTOR #####
#####

#Variable Global
dssh = paramiko.SSHClient()

#Limpiar pantalla CLI
def clear():
    if os.name == 'posix':
        os.system('clear')
    elif os.name in ('ce', 'nt', 'dos'):
        os.system('cls')

#menu principal
def menu():
    opcion = True
    while opcion:

```

```

print '\n'+chr(27)+'[0;46m'+ '*****CNT E.P.*****'+chr(27)+'[0m'+'\n'
print '    1) Configurar nuevo enlace'
print '    2) Borrar enlace'
print '    3) Consultar enlace'
print '    0) Salir del programa\n'
print chr(27)+'[0;46m'+ '*****'+chr(27)+'[0m'
opcion = raw_input('Su opcion ? ')

if opcion == '0':
    print '\n Usted ha salido del programa, hasta pronto!!!\n'
    dssh.close()
    sys.exit(0)

elif opcion == '1':
    print '\n'+chr(27)+'[0;46m'+ ' + Configurar nuevo enlace '+chr(27)+'[0m'+'\n'
    try:
        conexion()
        crear_enlace()
    except Exception, e:
        print '\n'
        print e
        print '\n'
        time.sleep(3)

elif opcion == '2':
    print '\n'+chr(27)+'[0;46m'+ ' - Borrar Enlace '+chr(27)+'[0m'+'\n'
    try:
        conexion()
        borrar_enlace()
    except Exception, e:
        print '\n'
        print e
        print '\n'
        time.sleep(3)

elif opcion == '3':
    print '\n'+chr(27)+'[0;46m'+ ' ? Consultar Enlace '+chr(27)+'[0m'+'\n'
    try:
        conexion()
        consultar_enlace()
    except Exception, e:
        print '\n'
        print e
        print '\n'
        time.sleep(3)

elif opcion != ('0' or '1' or '2' or '3'):
    print '\nOpcion no valida, intente otra vez \n'

#nombre del equipo
def hostname():
    stdin, stdout, stderr = dssh.exec_command('show hostname')
    host = stdout.read()
    host1 = host.rstrip('\n')
    return host1[:-1]

#barra de progreso
def barra(tiempo):
    pbar = ProgressBar()

    for i in pbar(range(24)):
        time.sleep(tiempo)

#Conexion BDD
def conexion_db():
    #Cadena de conexion
    conn = MySQLdb.connect(host="localhost", user="root", passwd="mypassword", db="bdd_sdn")
    puntero = conn.puntero()
    return (puntero,conn)

#Validar rango y valor de vlans
def validar_rangovlan():
    bucle = 0

    while bucle < 3:

        try:
            valorvlan = int (raw_input('VLAN ID: ').strip())
            if isinstance(valorvlan,int):
                if (valorvlan>1 and valorvlan<3968):
                    print 'VLAN ID VALIDO'
                    bucle = 3
                    return valorvlan
                else:
                    bucle += 1
                    print 'VLAN ID FUERA DE RANGO'+'\t(Intento '+str(bucle)+' DE 3)'

        except ValueError:
            bucle += 1
            print 'VLAN ID NO VALIDO'+'\t(Intento '+str(bucle)+' DE 3)'

#RespalDOS a un servidor TFTP

```

```

def backup ():
    timestamp = time.strftime("%Y-%m-%d-%H%M%S")
    host = hostname()
    com_respaldo = 'copy running-config tftp://100.100.2.100/' + host + '-cfg.' + timestamp + ' vrf management'
    print 'Estado:'
    print 'Respaldo de la configuracion del equipo por favor espere...'
    print com_respaldo
    stdin, stdout, stderr = dssh.exec_command(com_respaldo)
    barra(0.64)
    print ('Respaldo realizado en '+ time.strftime('%c')+'\n')

#Buscar VRF
def buscar_vrf(pcomm_vlan):
    ejecutar_vrf = 'show running-config interface vlan '+ pcomm_vlan + ' | grep "vrf member"'
    stdin, stdout, stderr = dssh.exec_command(ejecutar_vrf)
    cadena = stdout.read()
    if cadena.strip():
        lista = cadena.split()
        if lista[1] == 'member':
            return lista[2]
        else:
            return None

#Buscar OSPF ID
def buscar_idospf(pcomm_vlan):
    ejecutar_idospf = 'show running-config interface vlan '+ pcomm_vlan + ' | grep "ip router"'
    stdin, stdout, stderr = dssh.exec_command(ejecutar_idospf)
    cadena = stdout.read()
    if cadena.strip():
        lista = cadena.split()
        if lista[2] == 'ospf':
            return lista[3]
        else:
            return None

#Consultar enlaces configurado
def consultar_enlace():
    print ' ? Consulta de enlace'

    cvrf, cvlan, cidospf = validar_enlace()

    if (cvrf and cvlan and cidospf) is not None:

        print '\n-----'
        print ('VRF: \t\t' + cvrf)
        print ('VLAN: \t\t' + cvlan)
        print ('ID OSPF: \t' + cidospf)
        print '-----\n'

    else:
        print '-----'
        print 'La configuracion de este enlace no existe'
        print '-----\n'

#Borrar Enlaces
def borrar_enlace():

    print ' - Borrar Enlace'

    avrf, avlan, aidospf = validar_enlace()

    if (avrf and avlan and aidospf) is not None:

        print ' - El siguiente enlace ser@ eliminado:\n'
        print ('VRF: \t\t' + avrf)
        print ('VLAN: \t\t' + avlan)
        print ('ID OSPF: \t' + aidospf)
        resp = str(raw_input('Desea Continuar (s/n): ')).lower().strip()

        if (resp == 'si' or resp == 's'):
            #backup()
            shell = dssh.invoke_shell()

            delete_cmd1 = 'no vrf context '+ avrf
            delete_cmd2 = 'router ospf '+ aidospf
            delete_cmd3 = 'no vrf '+ avrf
            delete_cmd4 = 'no vlan '+ avlan
            delete_cmd5 = 'no interface vlan'+ avlan
            delete_cmd6 = 'no interface Ethernet17/25.'+ avlan
            delete_cmd7 = 'no interface Ethernet17/29.'+ avlan

            shell.send('configure terminal\n')
            time.sleep(3)
            shell.send(delete_cmd1 + '\n')
            shell.send(delete_cmd2 + '\n')
            shell.send(delete_cmd3 + '\n')
            shell.send('exit\n')
            shell.send(delete_cmd4 + '\n')
            shell.send(delete_cmd5 + '\n')
            shell.send(delete_cmd6 + '\n')
            shell.send(delete_cmd7 + '\n')
            shell.send('exit\n')
            shell.send('copy running-config startup-config\n')
            print 'Guardando configuracion ...'
            print 'Por favor espere'
            barra(1.7)
            time.sleep(40)
            puntero1, conn1 = conexion_db()

```

```

        puntero1.execute("DELETE FROM enlaces WHERE id_vlan = '%s'" % avrf)
        conn1.commit()
        print '\n - El enlace ha sido borrado'
    else:
        print 'Saliendo de la opcion eliminar ...'

else:
    print '-----'
        print 'La configuracion de este enlace no existe'
        print '-----\n'

def validar_ip():
    red_ip = raw_input('Ingrese la subred (R.R.R.H/MM): ')
    validacion = ('^(?:25[0-5]|2[0-4][0-9]|0[01]?[0-9]?[0-9])\.(?:25[0-5]|2[0-4][0-9]|0[01]?[0-9]?[0-9])\.(?:25[0-5]|2[0-4][0-9]|0[01]?[0-9]?[0-9])\.(?:25[0-5]|2[0-4][0-9]|0[01]?[0-9]?[0-9])$')
    ip = re.compile(validacion)
    comp = ip.search(red_ip)
    if comp is None:
        print ('Dirección IP incorrecta')
        return comp
    else:
        return red_ip

#Ejecuta comandos en los equipos
def ejecutar_comandos(cvlan, cidospf,direccion_ip)

    shell = dssh.invoke_shell()
    shell.send('configure terminal\n')
    time.sleep(3)
    cvrf = str(raw_input('Nombre VRF:')).lower().strip()
    aidospf = str(raw_input('ID Process OSPF:')).lower().strip()
    create_cmd1 = 'vrf context ' + cvrf + '_i'
    create_cmd2 = 'vlan ' + cvlan
    create_cmd3 = 'name v' + cvlan + ' ' + cvrf + '_internet'
    create_cmd4 = 'interface Vlan' + cvlan
    create_cmd5 = 'no shutdown'
    create_cmd6 = 'vrf member ' + cvrf + '_i'
    create_cmd7 = 'ip address ' + direccion_ip
    create_cmd8 = 'ip router ospf ' + cidospf + ' area 0.0.0.0'
    create_cmd9 = 'router ospf ' + aidospf
    create_cmd10 = 'vrf ' + cvrf + '_i'

#Comandos para las demas subinterfaces

    shell.send('configure terminal\n')
    time.sleep(3)
    shell.send(create_cmd1 + '\n')
    shell.send(create_cmd2 + '\n')
    shell.send(create_cmd3 + '\n')
    shell.send(create_cmd4 + '\n')
    shell.send(create_cmd5 + '\n')
    shell.send(create_cmd6 + '\n')
    shell.send(create_cmd7 + '\n')
    shell.send(create_cmd8 + '\n')
    shell.send('exit\n')
    shell.send(create_cmd9 + '\n')
    shell.send(create_cmd10 + '\n')

#Ejecucion para las demas subinterfaces

    shell.send('copy running-config startup-config\n')
    puntero1, conn1 = conexion_db()
    puntero1.execute("INSERT INTO enlaces VALUES ('%s','%s','%s','%s')", (% cvrf, % cvlan, % cidospf, % direccion_ip))
    conn1.commit()
    conn1.close()

```

```

        print 'Guardando configuracion ...'
        print 'Por favor espere'
        barra(1.7)

#Crear Enlaces
def crear_enlace():
    print ' + Configurar Enlace'
    cvrf, cvlan, cidospf = validar_enlace()

    if (cvrf and cvlan and cidospf) is None:
        print ('Se creara su enlace ...')
        tipo_enlace = str(raw_input('Tipo de enlace [Internet (i) / Datos (d)]: ')).lower().strip()

        if (tipo_enlace == 'I' or tipo_enlace == 'i'):
            cvlan = str(validar_rangovlan())
            direccion_ip = validar_ip()
            if (cvlan is not None and direccion_ip is not None):
                ejecutar_comandos(cvlan, cidospf,direccion_ip)
            else:
                if (tipo_enlace == 'D' or tipo_enlace == 'd'):
                    ejecutar_comandos(cvlan, cidospf,direccion_ip)
                else:
                    print ('No se puede crear su enlace')
        else:
            print '\n - La configuracion ya existe en este equipo:\n'
            print ('VRF: \t\t' + cvrf)
            print ('VLAN: \t\t' + cvlan)
            print ('ID OSPF: \t' + cidospf + '\n')

#Validar si existe VLAN ID

```

```

def validar_enlace():
    comm_vlan = str(validar_rangovlan())

    ejecutar_vlan = 'show running-config | grep "vlan '+ comm_vlan + '"'
    stdin, stdout, stderr = dssh.exec_command(ejecutar_vlan)
    a = stdout.read()
    if a.strip():
        lista = a.split()
        if (lista[1]==comm_vlan):
            vrf_array = buscar_vrf(comm_vlan)
            if vrf_array is not None:
                idospf_array = buscar_idospf(comm_vlan)
                if idospf_array is not None:
                    return (vrf_array, comm_vlan, idospf_array)
                else:
                    print '\n-----'
                    print 'No existe id ospf configurado'
                    print '-----\n'
                    return None, None, None
            else:
                print '\n-----'
                print 'No existe configuracion de vrf'
                print '-----\n'
                return None, None, None
        else:
            print '\n-----'
            print 'La vlan no existe'
            print '-----\n'
            return None, None, None
    else:

```

```
print '\n-----'
print 'La vlan no existe'
print '-----\n'
return None, None, None

#Acceso remoto SSHClient
def conexion():

    ip = raw_input('Ip/Hostname: ')
    username = raw_input('Usuario: ')
    password = getpass.getpass('Password: ')
    dssh.set_missing_host_key_policy(paramiko.AutoAddPolicy())
    dssh.connect(ip, username=username, password=password, look_for_keys=False, allow_agent=False)
    print '\nBienvenido a '+ hostname()

menu()
```