

**MODELO, DISEÑO Y TÉCNICAS BÁSICAS PARA IMPLANTAR UN SISTEMA  
DE COMPUTACIÓN EN LA NUBE**

**VIVIANA YISETH CARDOZO ESCOBAR**

**MARIA TRINIDAD BERNAL CELIS**

**JOSEMAR DAVID SIERRA RAMIREZ**



**Institución Universitaria**

**Politécnico Grancolombiano**

**INSTITUCIÓN UNIVERSITARIA POLITÉCNICO GRANCOLOMBIANO**

**FACULTAD DE INGENIERÍA**

**PROYECTO CURRICULAR DE INGENIERÍA DE SISTEMAS**

**BOGOTÁ D.C.**

**2015**

MODELO, DISEÑO Y TÉCNICAS BÁSICAS PARA IMPLANTAR UN SISTEMA DE  
COMPUTACIÓN EN LA NUBE

VIVIANA YISETH CARDOZO ESCOBAR 1310650357

MARIA TRINIDAD BERNAL CELIS 1310650922

JOSEMAR DAVID SIERRA RAMIREZ 1310650475

Trabajo de Grado para optar al título de  
Ingeniero de Sistemas

Director

Dr. Alexis Rojas Cordero

INSTITUCIÓN UNIVERSITARIA POLITÉCNICO GRANCOLOMBIANO

FACULTAD DE INGENIERÍA

PROYECTO CURRICULAR DE INGENIERÍA DE SISTEMAS

BOGOTÁ D.C.

2015

Nota de Aceptación:

---

---

---

---

---

---

Firma del Director

---

Firma del Jurado

---

Firma del Jurado

Bogotá D.C. Julio 2015

## **AGRADECIMIENTOS**

Primero y como más importante, agradecer a Dios y nuestras familias por darnos la fuerza y motivaciones para llevar a cabo nuestro proyecto.

A nuestro tutor el Dr. Alexis Rojas por su esfuerzo, su dedicación, sus conocimientos, sus orientaciones, su paciencia, su manera de trabajar han sido fundamentales para nuestra formación como investigadores.

A la Institución Universitaria Politécnico Grancolombiano por brindarnos los recursos y permitirnos realizar nuestro proyecto para que nuestra querida institución este a la vanguardia de la tecnología y para los futuros grancolombianos.

## TABLA DE CONTENIDO

<b>LISTA DE FIGURAS</b>	<b>7</b>
<b>LISTA DE TABLAS</b>	<b>8</b>
<b>LISTA DE ANEXOS</b>	<b>9</b>
<b>GLOSARIO</b>	<b>11</b>
<b>INTRODUCCIÓN</b>	<b>15</b>
<b>1. JUSTIFICACIÓN</b>	<b>16</b>
<b>2. OBJETIVOS</b>	<b>17</b>
2.1 <i>OBJETIVO GENERAL</i>	17
2.2 <i>OBJETIVOS ESPECIFICOS</i>	17
<b>3. PLANTEAMIENTO DEL PROBLEMA</b>	<b>18</b>
<b>4. ALCANCE Y LIMITACIONES</b>	<b>19</b>
4.1 <i>ALCANCE</i>	19
4.2 <i>LIMITACIONES</i>	19
<b>5. RECURSOS DEL PROYECTO</b>	<b>20</b>
<b>6. ESTADO DEL ARTE</b>	<b>21</b>
6.1 <i>EVOLUCION DEL ALMACENAMIENTO INFORMÁTICO</i>	21
6.2 <i>CLOUD COMPUTING</i>	23
6.2.1 Orígenes e influencias	23
6.2.2 Proyección	25
6.2.3 Concepto	26
6.2.4 Características	28
6.2.5 Arquitectura de la nube	30
6.3 <i>OPENSTACK</i>	35
6.3.1 Orígenes y concepto	35
6.3.2 Virtualización (Hipervisor KVM)	36
6.3.3 Componentes	37
6.3.4 Características de las versiones	42
6.4 <i>CONCEPTOS Y COMPONENTES UBUNTU</i>	44
6.4.1 Concepto	44
6.4.2 Historia	44

6.4.3	Características	45
6.4.4	Organización de paquetes	46
<b>7.</b>	<b>CONCLUSIONES</b>	<b>48</b>
<b>8.</b>	<b>BIBLIOGRAFIA</b>	<b>49</b>

## **LISTA DE FIGURAS**

Ilustración 1. Infografía de la evolución del almacenamiento de información

Ilustración 2. Crecimiento

Ilustración 3. Símbolo

Ilustración 4. Modelo de Servicios

Ilustración 5. Prototipo de arquitectura Openstack

## **LISTA DE TABLAS**

Tabla 1. Características servidor master

Tabla 2. Software a utilizar

Tabla 3. Extensiones permitidas para las instancias

Tabla 4. Versiones de OpenStack



## LISTA DE ANEXOS

### 1. SCRIPTS INSTALACIÓN OPENSTACK

#### 1.1 Essex

- 1.1.1 Openstack\_base\_1.sh
- 1.1.2 Openstack\_base\_2.sh
- 1.1.3 Openstack\_glance.sh
- 1.1.4 Openstack\_horizon.sh
- 1.1.5 Openstack\_keystone.sh
- 1.1.6 Openstack\_mysql.sh
- 1.1.7 Openstack\_nova.sh
- 1.1.8 Openstack\_nova\_single\_nic.sh
- 1.1.9 Openstack\_restart\_nova.sh

#### 1.2 Grizzly

- 1.2.1 Openstack\_cinder.sh
- 1.2.2 Openstack\_cleanup.sh
- 1.2.3 Openstack\_disable\_tracking.sh
- 1.2.4 Openstack\_drop\_databases.sh
- 1.2.5 Openstack\_glance.sh
- 1.2.6 Openstack\_horizon.sh
- 1.2.7 Openstack\_ipv6.sh
- 1.2.8 Openstack\_keystone.sh
- 1.2.9 Openstack\_loop.sh
- 1.2.10 Openstack\_mysql.sh
- 1.2.11 Openstack\_networking.sh
- 1.2.12 Openstack\_nova.sh
- 1.2.13 Openstack\_nova\_compute.sh
- 1.2.14 Openstack\_restart\_nova.sh
- 1.2.15 Openstack\_server\_test.sh
- 1.2.16 Openstack\_setup.sh
- 1.2.17 Openstack\_stackmonkey\_va.sh
- 1.2.18 Openstack\_system\_update.sh
- 1.2.19 Postcreation.sh

### **1.3 Icehouse**

- 1.3.1 Openstack\_ceilometer.sh
- 1.3.2 Openstack\_cinder.sh
- 1.3.3 Openstack\_cleanup.sh
- 1.3.4 Openstack\_disable\_tracking.sh
- 1.3.5 Openstack\_drop\_databases.sh
- 1.3.6 Openstack\_glance.sh
- 1.3.7 Openstack\_horizon.sh
- 1.3.8 Openstack\_ipv6.sh
- 1.3.9 Openstack\_keystone.sh
- 1.3.10 Openstack\_loop.sh
- 1.3.11 Openstack\_multi-node\_compute.sh
- 1.3.12 Openstack\_multi-node\_controller.sh
- 1.3.13 Openstack\_mysql.sh
- 1.3.14 Openstack\_networking.sh
- 1.3.15 Openstack\_nova.sh
- 1.3.16 Openstack\_nova\_compute.sh
- 1.3.17 Openstack\_restart\_nova.sh
- 1.3.18 Openstack\_server\_test.sh
- 1.3.19 Openstack\_setup.sh
- 1.3.20 Openstack\_splunk.sh
- 1.3.21 Openstack\_stackmonkey\_va.sh
- 1.3.22 Openstack\_system\_update.sh
- 1.3.23 Postcreation.sh

## **2. SISTEMA CLOUD COMPUTING**

- 2.1 Instalación y Configuración de Sistema Cloud Computing\_OpenStack.pdf

## **3. APRENDIENDO ADMINISTRAR LA NUBE**

- 3.1 Aprendiendo administrar la nube en el poli.pdf

## **4. ARTICULO SISTEMA E-LEARNING**

- 4.1 Arquitectura de un sistema de computación en la nube para e-learning.pdf

## **5. VIDEO DE DESARROLLO PROTOTIPO OPENSTACK EN EL POLITÉCNICO GRANCOLOBIANO**

5.1 Enlace: [https://www.youtube.com/watch?v=G9Bt5\\_XmdZo](https://www.youtube.com/watch?v=G9Bt5_XmdZo)

**GLOSARIO**

**Biometría:** Estudio de métodos automáticos para el reconocimiento único de humanos basados en uno o más rasgos conductuales o rasgos físicos intrínsecos.

**Clúster computacional:** Conjuntos de computadores construidos mediante la utilización de hardware común, que hacen procesamientos computacionales como si fuesen un único computador.

**Código Abierto:** Es la expresión con la que se conoce al software distribuido y desarrollado libremente. Se focaliza más en los beneficios prácticos (acceso al código fuente) que en cuestiones éticas o de libertad que tanto se destacan en el software libre.

**DHCP:** Protocolo de configuración dinámica de host, que permite a los clientes de una red IP obtener sus parámetros de configuración automáticamente.

**Django:** Es un framework de desarrollo web de código abierto, escrito en Python, que respeta el paradigma conocido como Model Template View. La meta fundamental de Django es facilitar la creación de sitios web complejos

**FTP:** Es un protocolo para transferencia de archivos entre sistemas conectados a una red TCP, basado en la arquitectura cliente-servidor.

**Ganglia:** Sistema de monitoreo escalable para sistemas computacionales de alto rendimiento, como clústeres y grids.

**Gartner Inc.** Es una empresa consultora y de investigación de las tecnologías de la información con sede en Stamford, Connecticut, Estados Unidos. Hasta 2001 era conocida como Gartner Group.

**George Gilder:** Es un alto miembro del Discovery Institute y publica el Informe Tecnología Gilder.

**Grid Computacional:** es una tecnología innovadora que permite utilizar de forma coordinada todo tipo de recursos (entre ellos cómputo, almacenamiento y aplicaciones específicas) que no están sujetos a un control centralizado.

**Hipervisor:** Es una plataforma que permite aplicar diversas técnicas de control devirtualización para utilizar, al mismo tiempo, diferentes sistemas operativos

**Icehouse:** La novena versión del software de código abierto para la construcción, privado, y las nubes híbridas públicos, cuenta con cerca de 350 nuevas características para apoyar el desarrollo de software, gestión de datos y la infraestructura de aplicaciones a gran escala.

**IDC (Analyze the future):** Es proveedor líder en el mundo en análisis e inteligencia de mercados, y en soporte táctico a fabricantes y usuarios, en el sector de Informática y Telecomunicaciones.

**Interoperabilidad:** Habilidad de dos o más sistemas o componentes para intercambiar información y utilizar la información intercambiada.

**IP Fija:** Es la dirección IP de la instancia. Se asigna automáticamente por DHCP en la creación de la instancia y tiene la misma duración que la vida de la instancia.

**IP Flotante:** Es una IP que se asocia a una instancia. Se realiza esta asociación para acceder desde fuera de la instancia, se realiza un NAT. Pueden existir instancias que no tengan una IP flotante, porque no interese que se acceda desde fuera, ya que se puede querer que forme parte de la red interna solamente.

**Kernel:** Software que constituye la parte más importante de un sistema operativo, encargado de gestionar recursos, a través de servicios de llamada al sistema.

**Máquinas Virtuales:** es un software que simula a una computadora y puede ejecutar programas como si fuese una computadora real.

**Organización Virtual (VO):** Conjunto de individuos y/o instituciones definidas por reglas que controlan el modo en que comparten sus recursos.

**Prototipo:** Un objeto diseñado para una demostración de cualquier tipo.

**QEMU:** Es un emulador de procesadores basado en la traducción dinámica de binarios (conversión del código binario de la arquitectura fuente en código entendible por la arquitectura huésped).

**SLA:** Acuerdo a nivel de servicio, suele ser un tipo de contrato reconocido a nivel internacional.

**Ubicuo:** Acceso por parte de los usuarios desde cualquier lugar, en cualquier dispositivo y en cualquier momento.

**Virtualización:** Es un mecanismo mediante el cual se crea una versión virtual de algún medio tecnológico, como una plataforma de hardware, un dispositivo de almacenamiento o recursos de red

**WMS:** Servicio usado para creación dinámica de entornos de aplicación aislados, usando máquinas virtuales o cuentas Unix.

**Xinetd:** Servicio que usan gran parte de los sistemas Unix, dedicado a administrar la conectividad basada en internet.

## INTRODUCCIÓN

La computación en la nube (cloud computing) es un paradigma que permite ofrecer servicios de computación a través de Internet a los que podemos acceder desde cualquier lugar teniendo la autorización.

En este tipo de computación todo lo que puede ofrecer un sistema informático se ofrece como servicio, de modo que los usuarios puedan acceder a los servicios disponibles "en la nube de Internet" sin ser expertos en la gestión de los recursos que usan.

Cloud computing es un nuevo modelo de prestación de servicios de negocio y tecnología, que permite al usuario acceder a un catálogo de servicios estandarizados y responder a las necesidades de su negocio, de forma flexible y adaptativa, en caso de demandas no previsibles o de picos de trabajo, pagando únicamente por el consumo efectuado. El cambio paradigmático que ofrece computación en nube es que permite aumentar el número de servicios basados en la red. Esto genera beneficios tanto para los proveedores, que pueden ofrecer, de forma más rápida y eficiente, un mayor número de servicios, como para los usuarios que tienen la posibilidad de acceder a ellos, disfrutando de la inmediatez del sistema y de un modelo de pago por consumo.

Computación en nube consigue aportar estas ventajas, apoyándose sobre una infraestructura tecnológica dinámica que se caracteriza, entre otros factores, por un alto grado de automatización, una rápida movilización de los recursos, una elevada capacidad de adaptación para atender a una demanda variable, así como virtualización avanzada y un precio flexible en función del consumo realizado evitando además el uso fraudulento del software y la piratería.

## 1. JUSTIFICACIÓN

En los últimos años las necesidades de explotación en el uso de las tecnologías de la información y procesamiento de datos han crecido exponencialmente. Frente a este crecimiento las Arquitecturas informáticas han evolucionado para responder a este crecimiento. Linux fue fundamental en los inicios para crear nuevas arquitecturas favorecedoras del procesamiento paralelo. Otro elemento es la Internet, que además de la navegación web, facilitaba el acceso remoto (conectividad) a Sistemas distantes por parte de usuarios. A mediados del 2000 se comenzaron a utilizar arquitecturas tipo Clúster, Grids de servidores, con amplias limitaciones, con esto se popularizaron las tecnologías de Virtualización que hacían posible implementar máquinas virtuales -que desacoplan en Hardware del software y permiten “replicar” el entorno de usuario sin tener que instalar todo el Software, para soportar aplicaciones.

Esta nueva arquitectura permitía distribuir la carga de trabajo/procesos, de forma sencilla y abría una nueva puerta que se llamó Cloud Computing. Este nuevo modelo inicia como un nuevo paradigma capaz de proporcionar recursos de almacenamiento que, además, resulta especialmente apto para la explotación comercial de las grandes capacidades de cómputo de proveedores de servicios en Internet.

Cloud Computing significa una revolución en la operativa de procesamiento de la información y gestionar las áreas IT. Tradicionalmente se realizaban inversiones enormes en hardware, licencias, CPDs, redes, personal, seguridad, entre otros. Mientras que con los servicios en la nube el Politécnico Grancolombiano obtendrá una reducción drástica en los costes fijos, y los suministros ofrecidos por los nuevos proveedores son flexibles e instantáneos bajo demanda.



## **2. OBJETIVOS**

### **2.1 OBJETIVO GENERAL**

Crear y depurar un prototipo de un sistema de computación en nube para trabajar dentro de un marco emulado, que sea escalable basado en OpenStack. Para ello se instalará una solución distribuida en servidores del Politécnico Grancolombiano, la cual debe estar visible para la evaluación por los asesores y respectivos entes.

### **2.2 OBJETIVOS ESPECIFICOS**

- Conocer las principales características de Cloud Computing y su aplicación en el aula, principalmente las capas de infraestructura (IaaS) y plataforma (PaaS).
- Instalar y utilizar el software OpenStack (IaaS). Software libre que permite gestionar infraestructura en nubes privadas o públicas.
- Publicar un artículo de resultados en la revista elementos del Politécnico Grancolombiano.
- Crear imágenes de sistemas operativos (Linux y Windows) para la creación de máquinas virtuales.
- Creación documento maestro del proyecto.

### 3. PLANTEAMIENTO DEL PROBLEMA

El foco de este proyecto ha sido el análisis, y posterior implantación de una solución real Cloud computing, orientada a explotación de servicios de todo tipo, 100% operativa, basada en Openstack. Disponible, para su manejo por cualquier facultad de la universidad, como ya se ha indicado, en su doble funcionalidad:

- (1) Tanto como usuario de la plataforma o
- (2) Como proveedor de servicios a terceros.

OpenStack se compone de más de una docena de elementos que controlan los aspectos más importantes de una nube. Hay recursos para la gestión de cómputo, redes y almacenamiento. Otros para la gestión y los de la orquestación de aplicaciones que se ejecutan en la misma entidad. En conjunto, estos componentes permiten a las empresas y proveedores de servicios ofrecer los recursos de computación mediante el aprovisionamiento y la gestión de grandes redes de máquinas virtuales.

OpenStack es en realidad un conjunto de proyectos Opensource que operan como un “Lego” dentro de una arquitectura abierta y altamente escalable. Este aspecto es el que nos resulta más fascinante.

El proyecto incluye la versión, OpenStack Icehouse, que contiene la totalidad de los proyectos de la Arquitectura.

La arquitectura diseñada constará de 3 servidores que contendrán todos los Componentes de Openstack y las herramientas de Monitorización y Gestión de alarmas precisas, con conexión a internet e integrará fácilmente todo tipo de imágenes para instanciar servidores tipo Linux y Windows, orientados a dar servicios reales internos o externos.

Los aspectos operativos como pudieran ser: la automatización de la instalación, la escalabilidad, el rendimiento, la monitorización, la alta disponibilidad, la gestión y administración vía interface web, serán aspectos fundamentales de consideración del proyecto.

## **4. ALCANCE Y LIMITACIONES**

### **4.1 ALCANCE**

El presente proyecto contempla la instalación de un prototipo para la ejecución de la plataforma computacional en la nube con OpenStack. Se implementará en tres servidores situados en la Institución Universitaria Politécnico Grancolombiano con el fin de interactuar con recursos cloud (procesamiento, almacenamiento e instancias) de forma autónoma.

### **4.2 LIMITACIONES**

Un prototipo de una cloud computing con OpenStack a nivel académico

La arquitectura para OpenStack es un prototipo controlador, el cual compila todos los servicios con recursos de un solo servidor.

De acuerdo a las características del servidor estará acotada la creación de un número de instancias específico.

El estado del servicio de virtualización en el procesador del servidor afecta directamente el funcionamiento correcto de OpenStack.

Las reglas configuradas dentro de la red privada donde está alojado el prototipo deben ser administradas por sus propios servicios.

El banco de imágenes de las instancias está delimitado por un conjunto de extensiones y/o formatos.

## 5. RECURSOS DEL PROYECTO

### 5.1 Hardware

<b>Referencia servidor:</b>	PowerEdge 2950 Dell
<b>Memoria Ram:</b>	8GB
<b>Cores:</b>	4
<b>Disco Duro:</b>	292 GB Raid 1 + 0

Tabla 1 - Características servidor master

### 5.2 Software

<b>Sistema Operativo:</b>	Linux Ubuntu 14.02
<b>Dashboard:</b>	Horizon
<b>Log:</b>	Splunk
<b>Motor base de datos:</b>	Mysql

Tabla 2 - Software a utilizar

### 5.3 Servicios

Listado de servicios a utilizar:

- Networking
- Cinder
- Nova
- Glance

## 6. ESTADO DEL ARTE

### 6.1 EVOLUCION DEL ALMACENAMIENTO INFORMÁTICO

El almacenamiento de gran cantidad de información, creó la necesidad de desarrollar dispositivos que logran responder a la demanda de mayor espacio y durabilidad para los datos.

El avance para el almacenamiento informático en las primeras décadas se dio a ritmo lento pero constante, siguiendo el orden cronológico mostrado a continuación:  
[10]

**Bits (b):** La tecnología de almacenamiento durante la década de los 50 consistía en tarjetas perforadas que podían almacenar alrededor de 960 b de información.

#### **Megabytes (MB):**

1. **Cinta magnética:** Surge en la década de los 60 en forma de bobina, con una longitud de hasta 4800 pies. También podía almacenar el mismo volumen de información que 10 000 tarjetas perforadas.
2. **Disquete de 5,25”:** Su atractivo radicaba en la posibilidad de utilizarlo como dispositivo de almacenamiento más portátil para los PC domésticos. En ellos se distribuían también los juegos de PC clásicos que se popularizaron en la década de los 70.
3. **Disquete de 3,5”:** Estos discos mantuvieron su popularidad durante tres décadas por sus capacidades de almacenamiento, su carácter portátil y su resistencia a los daños. El disquete de 3,5” podía almacenar alrededor de 1,44 MB.
4. **CD:** Con una capacidad de almacenamiento 450 veces superior a la de un disquete convencional, el CD fue un producto con una gran demanda durante la década de los 90 y aún sigue utilizándose.
5. **Disco ZIP:** Este dispositivo podía almacenar entre 100 y 750 MB y se convirtió en la opción idónea para hacer backups de archivos.

#### **Gigabytes (GB):**

Las unidades flash supusieron el siguiente paso en materia de almacenamiento de grandes volúmenes de datos, ya que pueden contener hasta 256 GB. Además de ser rentables y fáciles de utilizar, permiten almacenar una gran cantidad de datos.

### **Terabytes (TB):**

Los discos duros portátiles pueden almacenar hasta 4 TB y son útiles para hacer backups de archivos importantes o de gran tamaño.

### **Almacenamiento ilimitado:**

La oportunidad de utilizar almacenamiento en cloud ofrece a los consumidores posibilidades de almacenamiento prácticamente ilimitadas. Gracias al almacenamiento en cloud, puede archivar, compartir, enviar y almacenar archivos, así como acceder a ellos, desde cualquier dispositivo con conexión a Internet.



**Ilustración 1** – Infografía de la evolución del almacenamiento de información [10]

## 6.2 CLOUD COMPUTING

### 6.2.1 Orígenes e influencias

La idea de la computación en una "nube" remonta a los orígenes de la informática de utilidad, un concepto que el informático John McCarthy propuso públicamente en 1961:

*"Si los ordenadores de la clase por la cual he abogado se hacen los ordenadores del futuro, entonces la computación algún día puede ser organizada como un servicio público, así como el sistema telefónico que aplica un servicio de utilidad pública. La utilidad de las computadoras podría convertirse en la base de una nueva e importante industria."*

En 1969, Leonard Kleinrock, jefe científico de la Agencia de Proyectos de Investigación Avanzada Red o proyecto ARPANET que siembra la Internet, declaró lo siguiente:

*"Las redes de computadoras están todavía en su infancia, pero a medida que crecen y se vuelven sofisticadas, veremos probablemente la extensión de 'herramientas del ordenador' ".*

El público en general ha estado aprovechando las formas de servicios basados en Internet desde mediados de la década de 1990 a través de diversos motores de búsqueda (Yahoo!, Google, etc.), servicios de correo electrónico (Hotmail, Gmail), plataformas de publicación abierta (MySpace, Facebook, YouTube), y otros tipos de medios sociales (Twitter, LinkedIn). Aunque centrada en el consumidor, estos servicios popularizaron y validaron conceptos básicos que forman la base moderna de la computación en nube.

A finales de 1990, Salesforce.com fue pionera en la idea de traer remotamente provisionados en la empresa. En 2002, [www.Amazon.com](http://www.Amazon.com), lanzó la plataforma Amazon Web Services (AWS), un conjunto de servicios orientados a empresas que proveen remotamente almacenamiento de información, recursos y funciones de negocios de computación.

Una evocación ligeramente diferente del término "Red de la Nube" o "Nube" se introdujo a principios de los años 1990 en toda la industria de redes. Se refirió a una

capa de abstracción derivada de los métodos de entrega de datos heterogéneos públicos y semipúblicos que eran principalmente las redes de conmutación de paquetes, aunque las redes celulares usaran el término de la "Nube" también. El método de red en este momento apoyó la transmisión de datos de un punto final (red local) a la "Nube" (red de área amplia) y, después, se descompone en otro punto final intencionado. Esto es relevante, ya que la industria de las redes hace referencia al uso de este término, y es considerado como uno de los primeros en adoptar los conceptos base de servicios informáticos.

Fue hasta el 2006 que el término "*cloud computing*" surgió en el ámbito comercial, cuando **George Gilder** publicó su artículo "*Las fábricas de la información*" en la revista Wired [8]. En él expuso un modelo de nube virtual, similar en estructura a la computación en grid pero enfocada a su uso en la web.

La idea de esta "nube" es que es un intermediario virtual entre el proveedor del servicio y el usuario o cliente, y se puede acceder a esta nube desde cualquier dispositivo con acceso a internet. El proveedor maneja desde sus propias oficinas los recursos que presta a sus clientes, y estos recursos se "*virtualizan*" y pasan a ser parte de una "*nube virtual*".

Fue durante este tiempo que Amazon lanzó el servicio *Elastic Compute Cloud* (EC2) que permitió a las organizaciones "*arrendar*" capacidad de computación y potencia de procesamiento para ejecutar sus aplicaciones empresariales. Google Apps también comenzó a proporcionar aplicaciones empresariales basadas en navegador en el mismo año, y tres años más tarde, Google App Engine se convirtió en un hecho histórico.

Según el informe de **Gartner** se necesitarían siete años para que este modelo computacional esté completamente desarrollado. Para poder aprovechar la tendencia y, aquellas empresas que estén enfocadas en este nicho deberán focalizarse en proyectos de mediano plazo. A partir de 2011, el mercado comenzó a estar dominado por menos proveedores. Ya más a largo plazo, para 2015, el *cloud computing* será de uso masivo y será el modelo de entrega de aplicación preferida por los usuarios, anticipan desde la consultora "Para los próximos siete años, Gartner prevé la evolución del mercado del cloud computing en tres fases".

Algunas de las grandes innovaciones tecnológicas que vienen asociadas a la nube y que producirán un cambio social, además del cambio tecnológico, difícil de predecir: la web en tiempo real, la geolocalización, la realidad aumentada, el internet de las cosas y la pronta llegada de la telefonía móvil LTE de cuarta



generación, 4G, unida a los nuevos estándares de USB, Bluetooth e implantación de redes inalámbricas. [12]

### **2007- 2011: Los pioneros**

En estos años, el mercado experimentará su fase de desarrollo. La oportunidad estará en soluciones puntuales. Se tratará de oportunidades de carácter táctico, en las que los tiempos de llegada al mercado y la productividad del desarrollo pesarán sobre la viabilidad del programa a largo plazo.

### **2010- 2013: La consolidación del mercado**

A finales de junio de 2010, la consultora estadounidense, con presencia en todo el mundo, publicó un informe en que confirmaba el crecimiento de la computación en la nube. Solo los servicios vinculados a estas tecnologías registrarían ingresos de aproximadamente 68.300 millones de dólares. [12]

A partir de 2011, el mercado ha venido mostrando una oferta mucho más amplia que se repartirá entre vendedores grandes y chicos. Para entonces y por la mayor competencia, varios de los primeros y más pequeños proveedores se verán forzados a vender o integrarse a otros más grandes. La infraestructura de servicios en general se volverá más atractiva para un mayor número de usuarios.

### **2012 - 2015: Masa crítica y masificación**

De 2013 en adelante, el mercado ha estado dominado por un número más reducido de proveedores. Su oferta se centró en tecnologías de carácter propietario que fueron desarrolladas y perfeccionadas en los cinco años anteriores. Han servido como piedra angular para la entrega de interfaces que relacionan soluciones macro que vienen respondiendo y responderán a plataformas de diferentes proveedores.

En contrapartida se incrementará la oferta de Servicios de código fuente abierto. El resultado: más masa crítica y la acomodación del producto.

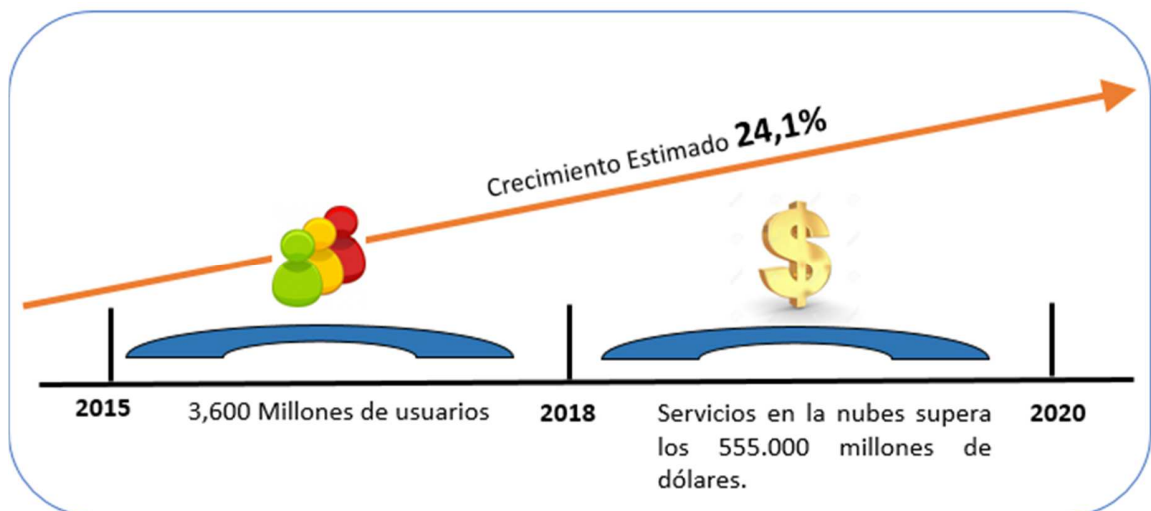
## **6.2.2 Proyección**

**Garther** en 2009 escribió: - Los ingresos de la nube computacional se disparan más rápido de lo esperado y superara \$150 billones para el año 2013. Representará el 19% del gasto en TI en 2015. [8]

**IDC** en 2009 - El gasto de servicios en la nube se triplicará en los próximos 5 años, alcanzando \$42 billones. [12]

**Forrester** en 2010 - La computación en la nube pasará de \$49,7 billones en 2010 hasta \$241 billones en 2020. [8]

**Forecasts** en 2014 – Según el estudio *Cloud Computing Consumer Markets* de Juniper Research, la demanda de servicios en la nube por parte de los consumidores aumentará en gran medida durante los próximos cinco años, y se estima que alcanzará los 3.600 millones de usuarios en 2018. Los analistas apuntan que en el 2020 el mercado global de servicios en la nube superará los 555.000 millones de dólares, y serán las empresas las que liderarán el uso de estos servicios con un crecimiento estimado del 24,1%. [11]



**Ilustración 1** - Crecimiento Fuente: Los autores. Acoplado [11]

### 6.2.3 Concepto

## Cloud (Nube)

Nube se refiere a un entorno de TI que está diseñado con el propósito de ofrecer recursos computacionales, escalables, seguros, medidos y fiables con tolerancia a fallos. El término se originó como una metáfora de la Internet, lo cual es, en esencia, una red de redes que ofrecen acceso remoto a un conjunto de servicios descentralizados TI. El símbolo de una nube se utiliza comúnmente para representar a la Internet en una variedad de especificaciones y documentación general de arquitecturas basadas en Web. Este mismo símbolo es utilizado actualmente para representar un entorno nube.

Este es el símbolo utilizado para indicar el límite de un entorno de cloud



**Ilustración 3 - Símbolo**

Considerando que Internet ofrece acceso abierto a muchos recursos basados en la Web, una nube es normalmente propiedad privada que brinda acceso a los servicios informáticos que se miden. Gran parte de la Internet está dedicado al acceso de contenidos publicados a través de la World Wide Web.

Por otra parte, los servicios en entornos cloud, se dedica a proveer procesamiento back-end (atrás de la interfaz) y las capacidades de acceso basado en el usuario. Otra diferencia clave es que no es necesario para la formación de nubes a través de Internet, aunque comúnmente están basados en las tecnologías y protocolos de Internet. Los protocolos se refieren a estándares y métodos que permiten que los ordenadores se comuniquen el uno con el otro de manera estructurada.

Una nube puede estar basada en el uso de cualquier protocolo que tenga el acceso remoto a servicios TI.

El uso de la palabra “cloud” se refiere a dos conceptos esenciales:

**Abstracción:** La computación en nube abstrae los detalles de la implementación del sistema de los usuarios y desarrolladores. Las aplicaciones se ejecutan en sistemas físicos que no están específicos, los datos se almacenan en posiciones que son

desconocidas, la administración de sistemas está externalizada a otros y el acceso por parte de los usuarios es ubicuo.

Virtualización: La computación en nube virtualiza sistemas agrupando y compartiendo recursos. Los sistemas y el almacenamiento son provistos a medida que se requieren desde una infraestructura centralizada.

#### **6.2.4 Características**

Con el fin de facilitar la comprensión del Cloud computing a continuación se detallan sus principales características según NIST (National Institute of Standards and Technology).

##### **A. Bajo demanda y autoservicio**

Ofrecer al usuario disponer de forma automática a las capacidades y necesidades de recursos, tales como el tiempo de almacenamiento y servidores de red, sin que sea necesaria la intervención manual del proveedor de servicios cloud.

Normalmente para los servicios ofertados y disponibles se reducirán los periodos de tiempo, porque probablemente solo requieran el registro o contrato del servicio y en corto tiempo e decir inmediato, el servicio estará disponible, ya que no requerirá instalaciones sino la ejecución e incluso el mantenimiento o lanzamiento de nuevas versiones, no se requerirá.

Es un gran beneficio para el usuario, ya que reduce las complicaciones que conlleva la adquisición de recursos propios TI (Hardware, Software) y el mantenimiento de los mismos.

Los servicios prestados por cloud computing no son gratis, sin embargo existen tarifas especiales, por lo que es indispensable que las necesidades y requerimientos de determinen antes que se proporcione el servicio especificado y con el contrato comercial correspondiente "SLA".

##### **B. Acceso amplio de la red**

La información está disponible a través de la red, desde cualquier lugar. No es necesario instalar un software determinado o tener un sistema operativo definido para cada cliente.

Es suficiente contar con una conexión a internet y un navegador web para disfrutar de los servicios en la nube, los cuales pueden ser accedidos a través de múltiples dispositivos (smartphones, tablets, portátiles y estaciones de trabajo).

Esta característica es importante y útil para las compañías u organizaciones distribuidas geográficamente, permite acceso a los recursos sin depender en aspectos como ubicación o zona horaria.

### **C. Agrupación de recursos**

Los proveedores de servicios informáticos agrupan y comparten sus recursos para servir a varios consumidores, utilizando un modelo multiusuario, estableciendo unos recursos de acceso y prestaciones distintos para cada usuario, con el fin de disminuir los costes y ampliar la disponibilidad. Al ser aplicaciones multiusuario, pueden ser miles de internautas utilizando la misma herramienta a la vez, por ende, la asignación de la capacidad de computación, almacenamiento y velocidad, debe ser asignada automáticamente en la base a las peticiones de los usuarios.

### **D. Elasticidad y escalabilidad**

Los conceptos de escalabilidad y elasticidad están orientados a la posibilidad de añadir o eliminar recursos en cuestión de minutos según demanda, sin que las aplicaciones se vean afectadas y sea transparente para los usuarios. La capacidad de escalar se consigue mediante la propiedad de elasticidad o flexibilidad en el cambio. Además de ello, la capacidad de adaptarse a cualquier sistema sobre el que se estén ejecutando.

### **E. Servicio medido**

El servicio prestado al usuario final es medible, lo que quiere decir, que tanto proveedor como usuario tienen acceso transparente al consumo real de los recursos, esto posibilita el pago por el uso o disponibilidad real de los servicios, la capacidad de medición debe ser a nivel de abstracción apropiado al tipo de servicio (almacenamiento, procesamiento, ancho de banda y cuentas activas de usuario).

### **F. Interfaces de programación de aplicaciones API estándares**

Los servicios en la nube necesitan que sus API estén estandarizadas con el objetivo de conseguir la interoperabilidad con las aplicaciones de los clientes, se requiere que cumplan con características como la portabilidad, la integración y la seguridad.

### **G. Monitorización y medida del rendimiento**

El proveedor de servicios de la nube debe incluir un entorno de gestión de servicios que permita gestionar sus sistemas de TI y sus entornos físicos; debe poder mantener y garantizar el nivel de servicio requerido por la organización, además de ello garantiza la monitorización y la medida del rendimiento del sistema globalmente, tanto en los aspectos de seguridad como en el propio rendimiento e informar respectivamente.

## **H. Seguridad**

La primera prioridad en un estudio de migración a la nube es la seguridad, por ello para generar confianza en el uso de la misma depende de los mecanismos de computación, de la gestión de la identidad robusta y de las técnicas de control de acceso que proporcionan un entorno seguro de ejecución, aseguramiento de las condiciones de la nube y el soporte de las infraestructura; se debe determinar cuáles son los recursos autorizados a un usuario para el acceso, o el proceso de uso de tecnologías como la biometría o tarjetas inteligentes, y determinar cuándo se ha accedido a un recurso por entidades o usuarios no autorizados, es indispensable tener presente la implementación de políticas de seguridad.

### **6.2.5 Arquitectura de la nube**

La arquitectura *Cloud Computing* consiste de un conjunto de capas que se encuentran acopladas entre sí para brindar la funcionalidad del sistema, en este caso la arquitectura de Cloud Computing es similar a la arquitectura de red, desde un nivel físico hasta un nivel de aplicación. Esto debido a que Cloud Computing utiliza protocolos similares a los que se usan en Internet como medio de comunicación.

La arquitectura o soluciones de Cloud Computing existentes se clasifican según diferentes criterios, que fundamentalmente son: Familias, y Formas de Implementación. Las cuales se presentan como software, infraestructura o plataforma como servicio.

#### **• Familias (Modelos de Servicio):**

Se refieren a los tipos específicos de servicios a los que se puede acceder en una plataforma de computación en la nube

#### **1. Infraestructura como Servicio (IaaS).**

En el modelo IaaS, se parte de la idea de la externalización de servidores para espacio en disco, base de datos etc., en lugar de tener un control completo de los mismos con el DATA CENTER dentro de la empresa, u optar por un centro de datos y sólo administrarlo. Mediante este modelo de despliegue en Cloud, lo que se tiene es una solución basada en la virtualización, en la que se paga por el nivel de consumo de los recursos: espacio en disco utilizado, tiempo de CPU, espacio en base de datos, transferencia de datos.

En un modelo IaaS, un proveedor de terceros acoge hardware, software, servidores, almacenamiento y otros componentes de la infraestructura en nombre de sus usuarios. Los proveedores de IaaS también reciben solicitudes de los usuarios y manejar tareas como el mantenimiento del sistema, copia de seguridad.

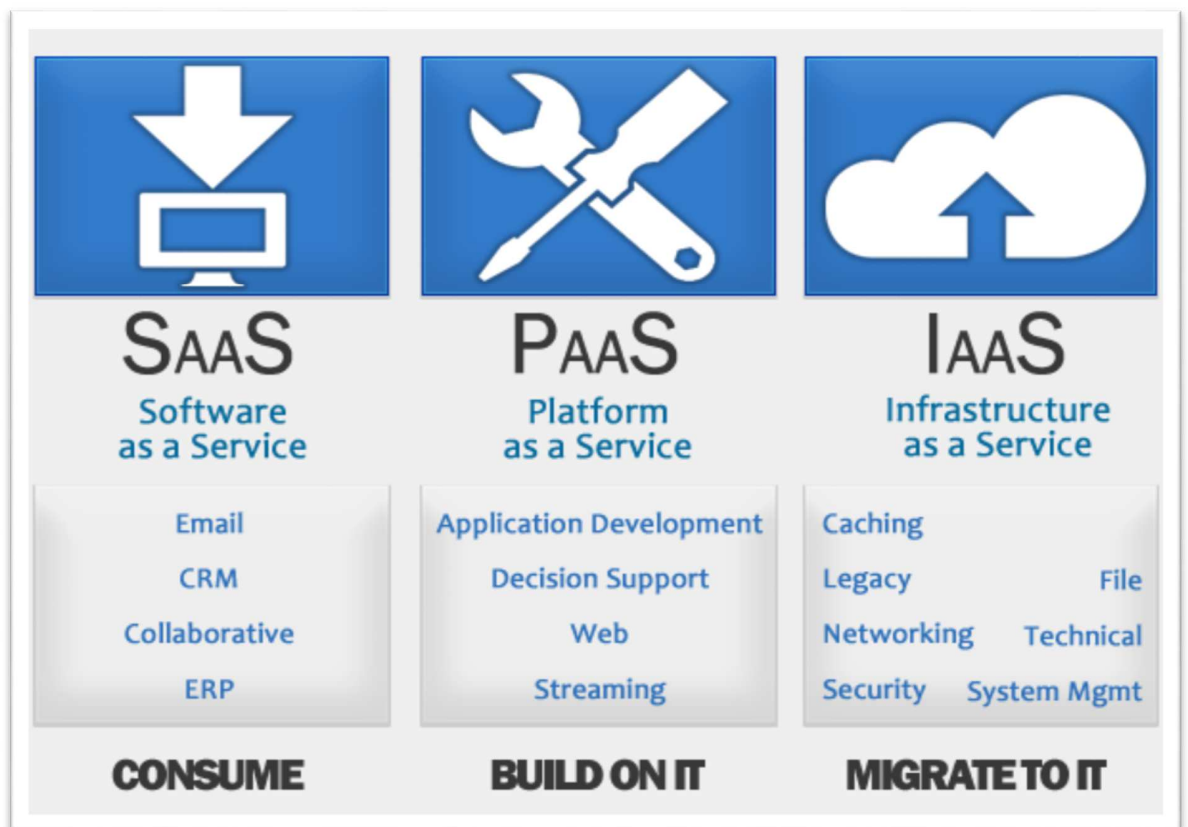


Ilustración 4 - Modelo de Servicios Fuente [10]

Plataformas IaaS ofrecen recursos altamente escalables que se pueden ajustar a la carta por demanda. Esto hace que IaaS sea muy adecuado para las cargas de trabajo que son temporales, experimental o cambian inesperadamente.

Otras características de los entornos IaaS incluyen la automatización de tareas administrativas, escalamiento dinámico, la virtualización de escritorios y los servicios basados en políticas. Los Clientes IaaS pagan en función de cada uso, típicamente por horas, semanas o meses. Algunos proveedores también cobran a los clientes sobre la base de la cantidad de espacio de la máquina virtual que utilizan. Esto elimina el gasto de capital de desplegar en el local de hardware y software.

La ventaja más inmediata de elegir este tipo de soluciones es la de desplazar una serie de problemas al proveedor relacionados con la gestión de las máquinas y llegar a un ahorro de costes importante, ya que pagaremos solo por lo consumido en función del nivel servicio que nos ofrezca dicho proveedor. Otro aspecto fundamental a tener en cuenta, es que las Infraestructura como servicio pueden permitir una escalabilidad automática o semiautomática, de forma que podamos contratar más recursos según los vayamos necesitando.

Como ejemplos de IaaS, se pueden citar: AbiCloud, Amazon, Web Services EC2, GoGrid, entre otros. Existen diversas soluciones de software para generar IaaS, tanto OpenSource como de ámbito privado: Vmware, Citrix, 3Tera, Abiquo, Enomaly, Eucalyptus, Proxmox, OpenNebula.org, etc.

## **2. Software como Servicio (SaaS)**

SaaS es aquella aplicación ofrecida por un fabricante de software o proveedor de servicios informáticos a través de Internet, para su uso o utilización por varios clientes.

El fabricante es el que en última instancia se ocupa del manteniendo de la privacidad de los datos y la personalización de la aplicación. En este modelo de servicio, el usuario paga por el uso y por la infraestructura necesaria (almacenamiento, seguridad, alojamiento, etc.) para el correcto funcionamiento de la aplicación y, a excepción de unos pocos parámetros de configuración, se limita a utilizar la herramienta y sus funcionalidades.

Algunos ejemplos de SaaS son: Google Apps, DocuMany, TeamBox, Kubbos, Gupigupi, Salesforce, Basecamp, Gmail, Salesforce.com, MediaWiki, Moodle, WordPress, etc.



### 3. Plataforma como Servicio (PaaS)

Son servicios de plataformas orientadas al desarrollo, testing, despliegue, entre otros de aplicaciones del cliente. La información de los Sistemas se encuentra ubicada en la “*nube pública*”. Puede originar conflictos por las exigencias de seguridad en la red.

En este modelo de nube, el usuario no podrá gestionar la infraestructura de la nube, pero tendrá acceso tanto en las aplicaciones desplegadas en ella como sobre la configuración de las diversas herramientas que utilice.

Resumiendo, se puede decir que PaaS es la encapsulación de una abstracción de un ambiente de desarrollo y el empaquetamiento de una serie de módulos o complementos que proporcionan, normalmente, una funcionalidad horizontal (persistencia de datos, autenticación, mensajería, etc.). De esta forma, un arquetipo de plataforma como servicio podría consistir en una lista de componentes o APIs pre-configuradas y listas para integrarse sobre una tecnología concreta de desarrollo (por ejemplo, un servidor web, un sistema Linux, AIX, Solaris, etc. o un ambiente de programación como Perl, Fortran, C++, Java, Ruby, etc.)

Ejemplos de PaaS son: Velneo, Abiquo.com, SimpleDB SQS Google App Engine, entre otros.

#### • Formas de Implementación (Formas de Integración y Explotación):

Se refieren a la *posición* (localización) y administración (gestión) de infraestructura de la nube (pública, privada, comunitaria, híbrida).

#### 1. Cloud Público (Externo)

Se ofrecen a los servicios de computación virtualizados (Sistemas operativos, plataformas, bases de datos, etc...) por parte de los proveedores a través de Internet (y/o VPN seguras).

La ventaja más clara de las nubes públicas es la capacidad de procesamiento y almacenamiento sin instalar máquinas localmente, por lo que no tiene una inversión inicial o gasto de mantenimiento en este sentido, si no que se paga por el uso. La carga operacional y la seguridad de los datos (backup, accesibilidad, etc.) recae íntegramente sobre el proveedor del hardware y software, debido a ello, el riesgo por la adopción de una nueva tecnología es bastante bajo. El retorno de la inversión se hace rápido y más predecible con este tipo de nubes.

Como inconvenientes se cuenta con el acceso de toda la información a terceras empresas, y la dependencia de los servicios en línea (a través de Internet). También puede resultar difícil integrar estos servicios con otros sistemas propietarios.

## **2. Cloud Privado (Interno)**

En las nubes privadas, sin embargo, la plataforma se encuentra dentro de las instalaciones del usuario y no suele ofrecer servicios a terceros.

En general, una nube privada es una plataforma para la obtención solamente de hardware, es decir, máquinas, almacenamiento e infraestructura de red (IaaS), pero también se puede tener una nube privada que permita desplegar aplicaciones (PaaS) e incluso aplicaciones (SaaS).

Como ventaja de este tipo de nubes, al contrario que las públicas, es la localización de los datos dentro de la propia empresa, lo que conlleva a una mayor seguridad de estos, corriendo a cargo del sistema de información que se utilice, incluso será más fácil integrar estos servicios con otros sistemas propietarios.

Sin embargo, como inconveniente se encuentra la inversión inicial en infraestructura física, sistemas de virtualización, ancho de banda y seguridad, lo que llevará a su vez a pérdida de escalabilidad y desestabilidad de las plataformas, sin olvidar el gasto de mantenimiento que requiere. Esta alta inversión supondrá un retorno más lento de la inversión.

## **3. Cloud Híbrido**

Las nubes híbridas consisten en combinar las aplicaciones locales con las de la nube pública. Se puede ver también como aplicación privada que se ve aumentada con los servicios de Cloud Computing y la infraestructura.

Una nube híbrida tiene la ventaja de una inversión inicial más moderada y a la vez contar con SaaS, PaaS o IaaS bajo demanda. En el momento necesario, utilizando las APIs de las distintas plataformas públicas existentes, se tiene la posibilidad de escalar la plataforma todo lo que se quiera sin invertir en infraestructura.

Este tipo de nube tiene una gran aceptación, ya que se están desarrollando softwares de gestión de nubes para poder gestionar la nube privada y a su vez adquirir recursos en los grandes proveedores públicos.

El mayor inconveniente puede estar en la mayor complejidad de la solución ya que requiere la integración de dos tipos de nube, con interfaces quizás no compatibles.

## **6.3 OPENSTACK**

### **6.3.1 Orígenes y concepto**

A día de hoy, hablar de software de computación en la nube supone hablar de OpenStack.

OpenStack es un proyecto de software libre para proporcionar computación en la nube de tipo IaaS. Nacido de una iniciativa conjunta de Rackspace y la NASA en Julio de 2010, su popularidad y uso desde su lanzamiento se ha disparado. Más de 200 compañías, incluyendo a IBM, Cisco, Canonical o Red Hat3 apoyan el proyecto en la actualidad.

Este ascenso debe, entre otras cosas, a la licencia Apache4 en la que se distribuye.

OpenStack cuenta con una licencia más permisiva que la licencia GPL5 (el estándar de facto del software libre). Esto implica que cualquiera puede notificar bugs, desarrollar complementos o incluso programas enteros que utilicen su interfaz de programación de aplicaciones (API, por sus siglas en inglés). Además, OpenStack provee de compatibilidad con la API de Amazon EC2, lo cual facilita el proceso de portar software desde esta popular plataforma.

### **6.3.2 Virtualización (Hipervisor KVM)**

Openstack puede ser utilizado con hipervisores KVM, a continuación se hace una introducción sobre éste y el porqué de su elección. También mencionaremos por encima un componente crucial para la gestión de las instancias virtualizadas que es la librería libvirt.

#### **6.3.2.1 KVM**

KVM es un módulo más del kernel. Esto significa que no es un kernel “diferente”. Al ser un módulo del kernel:

- Viene por defecto del Kernel.

Por otra parte, debido a que es parte del kernel, utiliza muchas de las ventajas implícitas del kernel como son:

- Escalabilidad
- Soporte para diferentes procesadores
- Gestión de memoria

KVM puede compartir páginas de memoria, y de esta forma nos permite ahorrar memoria y por tanto, dinero.

Finalmente siendo parte del kernel, tiene menos líneas de código, ya que mucha funcionalidad la aprovecha del propio kernel. El desarrollo es más fácil y está centralizado. [17]

#### **6.3.2.2 Libvirt.**

La biblioteca libvirt es una API de Linux sobre las capacidades de virtualización de Linux que soporta una variedad de hipervisores, incluyendo Xen y KVM, como así también QEMU y algunos productos de virtualización para otros sistemas operativos. Este software es ideal y crucial para la gestión de las máquinas virtuales y su administración remota a través de conexiones seguras con SSH. Además incluye una librería en C para el desarrollo de aplicaciones e incluye además interfaces para diferentes lenguajes como Python, Perl, Ruby, Java y PHP. [18]

#### **6.3.3 Componentes**

OpenStack está formado por diferentes componentes. Estos componentes pueden ser instalados en cualquier máquina de la nube e incluso movidos de una máquina a otra sin problemas. Algunos de ellos pueden ser incluso sustituidos por otras alternativas, siempre que utilicen una API soportada.



- **DB:** La base de datos SQL para almacenar datos
- **Panel de control web:** Componente externo que permite la conexión con la API.
- **API:** Conjunto de subrutinas, funciones y procedimientos que representa la capacidad de comunicación entre componentes de software, con el fin de recibir solicitudes http.
- **Directorio Auth:** Una clase de Python utilizada por todos los elementos para comunicarse con el servidor DB o LDAP. Además, es responsable de los usuarios, proyectos y roles.
- **Almacén de objetos:** Réplica de API S3 que permite el almacenamiento y recuperación de imágenes.
- **Schedule (Planificador):** Asigna los host adecuados para los sistemas de memoria virtual.
- **Red:** Responsable de asignar y reenviar las IP, puentes y redes VLAN.
- **Compute (Procesamiento):** Controla la comunicación entre el hipervisor y máquinas virtuales.

### 6.3.3.2 OpenStack Object Storage (Swift)

Es un código abierto de almacenamiento masivo de objetos a través de un sistema escalable, redundante y tolerante a fallos. Las posibles aplicaciones de Object Storage son numerosas, como por ejemplo: almacenamiento simple de ficheros, copias de seguridad, almacenamiento de streamings de audio/vídeo, almacenamiento secundario/ terciario, desarrollo de nuevas aplicaciones con almacenamiento integrado, etc.

Ficheros y archivos se escriben en varias unidades, y el software Swift asegura que los datos se replican a través de un servidor clúster.

### 6.3.3.3 OpenStack Identity Service (Keystone)

Proporciona a un servicio común para la autenticación y autorización. Es encargado de gestionar usuarios, roles y a qué proyectos pertenecen, proporcionando un catálogo de todos los servicios OpenStack. Este servicio utiliza un sistema de autenticación basado en tokens y se incorporó en la versión 2012.1 de OpenStack.

Es importante resaltar, que todos los servicios OpenStack utilizan Keystone para verificar las peticiones.

Cada función de Keystone tiene un back-end vinculable que permite diferentes formas de usar el servicio. Se brinda soporte para los backends estándares como LDAP o SQL, y KVS (Key-Value Stores).

### 6.3.3.4 OpenStack Image Service (Glance)

Glance proporciona un mecanismo de acceso a plantillas de instancias. Es un servicio para la búsqueda y recuperación de imágenes de máquinas virtuales. Este servicio puede almacenar las imágenes directamente o utilizar mecanismos más avanzados como: usar Object Storage como servicio de almacenamiento, usar Amazon's Simple Storage Solution (S3) directamente, o usar Object Storage como almacenamiento intermedio de S3.

Dentro de Glance encontraremos:

- ***glance-api*** acepta los pedidos para la búsqueda, obtención y almacenamiento de imágenes.
- El registro de almacenamiento ***glance-registry*** procesa y recupera los metadatos de las imágenes.
- Posee una base de datos para los metadatos de las imágenes.
- También se corren servicios de replicación, para proveer consistencia y disponibilidad a través del clúster, auditoría y actualización.

Los diferentes formatos soportados por las imágenes son los mostrados en la siguiente tabla:



Extensiones Instancias	
AMI (El formato de AWS)	QCOW2 (Qemu/KVM)
VHD (Hyper-v)	VMDK (Vmware)
VDI (VirtualBox)	OVF (Vmware, otros)
Raw	

Tabla 3 - Extensiones permitidas para las instancias

### 6.3.3.5 OpenStack Dashboard (Horizon)

Es un panel web de instrumentos OpenStack proporciona a los administradores y usuarios una interfaz gráfica para el acceso, provisión y automatizar los recursos basados en la nube. El diseño extensible hace que sea fácil de conectar y exponer los productos y servicios de terceros, como la facturación, seguimiento y herramientas de gestión adicionales.

El tablero de instrumentos es sólo una forma de interactuar con los recursos OpenStack. Los desarrolladores pueden automatizar el acceso o construir herramientas para gestionar sus recursos mediante la API de OpenStack nativo o la API de compatibilidad EC2.

### Capacidad Dashboard

- El tablero de instrumentos es una aplicación web extensible que permite a los administradores de la nube y los usuarios controlar sus recursos informáticos, de almacenamiento y de redes.
- Como administrador de la nube, el tablero de instrumentos ofrece una visión global del tamaño y estado de su nube. Puede crear usuarios y proyectos, asignar usuarios a los proyectos y establecer límites en los recursos para esos proyectos.
- El panel web proporciona a los usuarios un portal de autoservicio para la prestación de sus propios recursos dentro de los límites establecidos por los administradores.

### 6.3.3.6 OpenStack Block Storage (Cinder)

Es un dispositivo de almacenamiento a nivel de bloque persistente para uso con instancias de proceso OpenStack. Gestiona la creación, fijación y separación de dispositivos de bloque a los servidores. Además de utilizar almacenamiento simple, proporciona soporte unificado para numerosas plataformas de información, incluyendo Ceph, NetApp, Nexenta, SolidFire y Zadara.

Proporciona funcionalidad de administración para copias de seguridad de los datos guardados en los bloques, los cuales se pueden restaurar o utilizar para crear nuevos volúmenes de almacenamiento.

No se puede asociar un volumen a más de una instancia, y tampoco se puede modificar el volumen en caliente. Pero podemos crear instancias a partir de volúmenes y que estos volúmenes dispongan de las aplicaciones que queramos.

Cinder incluye los siguientes componentes:

- **Cinder-api:** Acepta solicitudes y las dirige a los bloques de volúmenes para la acción.
- **Cinder-volume:** Responde a las peticiones para leer y escribir a una base de datos con el fin mantener el estado interactuando con otros procesos, como el programador de cinder, que a través de una cola de mensajes actúa directamente sobre el software o hardware.
- **Cinder-scheduler:** Escoge el nodo proveedor más óptimo para crear el volumen.
- **Lista de espera de mensajería:** Rutas de información entre los procesos de servicio de almacenamiento en bloque.

### 6.3.3.7 OpenStack Networking (Quantum)

Es el servicio de control de red. Además de controlar la red física de las máquinas, permite a los usuarios crear redes y con ellas conectar instancias entre sí.

### 6.3.4 Características de las versiones

OpenStack ha pasado por diferentes versiones, la versión mas reciente tiene por nombre IceHouse, la cual se ha utilizado en el presente proyecto junco con Essex.

A continuación se listan las diferentes versiones que ha tenido OpenStack, sus incorporaciones y cambios en los servicios por cada versión.

Nombre de Versión	Fecha Creación	Componentes incluidos
Austin	21 de Octubre de 2010	Nova, Swift
Bexar	3 de Febrero de 2011	Nova, Glance, Swift
Cactus	15 de Abril de 2011	Nova, Glance, Swift
Diablo	22 de Septiembre de 2011	Nova, Glance, Swift
Essex	5 de Abril de 2012	Nova, Glance, Swift, Horizon, Keystone
Folsom	27 de Septiembre de 2012	Nova, Glance, Swift, Horizon, Keystone, Quantum, Cinder
Grizzly	4 de Abril de 2013	Nova, Glance, Swift, Horizon, Keystone, Quantum, Cinder
Havana	17 de Octubre de 2013	Nova, Glance, Swift, Horizon, Keystone, Neutron, Cinder
IceHouse	Abril de 2014	Nova, Glance, Swift, Horizon, Keystone, Neutron, Cinder

Tabla 4 - Versiones de OpenStack [16]

## **6.4 CONCEPTOS Y COMPONENTES UBUNTU**

### **6.4.1 Concepto**

Ubuntu es una distribución Linux que ofrece un sistema operativo predominantemente enfocado a ordenadores de escritorio aunque también proporciona soporte para servidores.

Basada en Debían GNU/Linux, Ubuntu concentra su objetivo en la facilidad de uso, la libertad en la restricción de uso, los lanzamientos regulares (cada 6 meses) y la facilidad en la instalación. Ubuntu es patrocinado por Canonical Ltd., una empresa privada fundada y financiada por el empresario sudafricano Mark Shuttleworth.

El nombre de la distribución proviene del concepto zulú y xhosa de Ubuntu, que significa “humanidad hacia otros” o “yo soy porque nosotros somos”. Ubuntu es un movimiento sudafricano encabezado por el obispo Desmond Tutu, quien ganó el Premio Nobel de la Paz en 1984 por sus luchas en contra del Apartheid en Sudáfrica. El sudafricano Mark Shuttleworth, mecenas del proyecto, se encontraba muy familiarizado con la corriente. Tras ver similitudes entre los ideales de los proyectos GNU, Debían y en general con el movimiento del software libre, decidió aprovechar la ocasión para difundir los ideales de Ubuntu. El eslogan de Ubuntu – “Linux para seres humanos” (en inglés “Linux for Human Beings”) – resume una de sus metas principales: hacer de Linux un sistema operativo más accesible y fácil de usar.

### **6.4.2 Historia**

Mark Shuttleworth de la Fundación Ubuntu en la WSIS 2005 de Túnez

El 8 de julio de 2004, Mark Shuttleworth y la empresa Canonical Ltd. anunciaron la creación de la distribución Ubuntu. Ésta tuvo una financiación inicial de 10 millones de dólares (US\$). El proyecto nació por iniciativa de algunos programadores de los proyectos Debían, Gnome porque se encontraban decepcionados con la manera de

operar del proyecto Debían, la distribución Linux sin ánimo de lucro más popular del mundo.

De acuerdo con sus fundadores, Debían era un proyecto demasiado burocrático donde no existían responsabilidades definidas y donde cualquier propuesta interesante se ahogaba en un mar de discusiones. Asimismo, Debían no ponía énfasis en estabilizar el desarrollo de sus versiones de prueba y sólo proporcionaba auditorías de seguridad a su versión estable, la cual era utilizada sólo por una minoría debido a la poca o nula vigencia que poseía en términos de la tecnología Linux actual.

Tras formar un grupo multidisciplinario, los programadores decidieron buscar el apoyo económico de Mark Shuttleworth, un emprendedor sudafricano que vendió la empresa Thawte a VeriSign, cuatro años después de fundarla en el garaje de su domicilio, por 575 millones de dólares estadounidenses.

Shuttleworth vio con simpatía el proyecto y decidió convertirlo en una iniciativa auto sostenible, combinando su experiencia en la creación de nuevas empresas con el talento y la experiencia de los programadores de la plataforma Linux. De esta forma nació la empresa Canonical, la cual se encarga de sostener económicamente el proyecto mediante la comercialización de servicios y soporte técnico a otras empresas. Mientras los programadores armaban el sistema, Shuttleworth aprovechó la ocasión para aplicar una pequeña campaña de mercadotecnia para despertar interés en la distribución sin nombre (en inglés: the no-name-distro).

Tras varios meses de trabajo y un breve período de pruebas, la primera versión de Ubuntu (Warty Warthog) fue lanzada el 20 de octubre de 2004.

### **6.4.3 Características**

- ✓ Basada en la distribución Debían.
- ✓ Disponible en 4 arquitecturas: Intel x86, AMD64, SPARC (para esta última sólo existe la versión servidor).
- ✓ Los desarrolladores de Ubuntu se basan en gran medida en el trabajo de las comunidades de Debían y GNOME.
- ✓ Las versiones estables se liberan cada 6 meses y se mantienen actualizadas en materia de seguridad hasta 18 meses después de su lanzamiento.
- ✓ La nomenclatura de las versiones no obedece principalmente a un orden de desarrollo, se compone del dígito del año de emisión y del mes en que esto ocurre. La versión 4.10 es de octubre de 2004, la 5.04 es de abril de

2005, la 5.10 de octubre de 2005, la 6.06 es de junio de 2006, la 6.10 es de octubre de 2006 y la 7.04 es de abril de 2007.

- ✓ El entorno de escritorio oficial es Gnome y se sincronizan con sus liberaciones.
- ✓ Para centrarse en solucionar rápidamente los bugs, conflictos de paquetes, etc. se decidió eliminar ciertos paquetes del componente main, ya que no son populares o simplemente se escogieron de forma arbitraria por gusto o sus bases de apoyo al software libre. Por tales motivos inicialmente KDE no se encontraba con más soporte de lo que entregaban los mantenedores de Debían en sus repositorios, razón por la que se sumó la comunidad de KDE distribuyendo la distro llamada Kubuntu.
- ✓ De forma sincronizada a la versión 6.06 de Ubuntu, apareció por primera vez la distribución Xubuntu, basada en el entorno de escritorio XFce.
- ✓ El navegador web oficial es Mozilla Firefox.
- ✓ El sistema incluye funciones avanzadas de seguridad y entre sus políticas se encuentra el no activar, de forma predeterminada, procesos latentes al momento de instalarse. Por eso mismo, no hay un firewall predeterminado, ya que no existen servicios que puedan atentar a la seguridad del sistema.
- ✓ Para labores/tareas administrativas en terminal incluye una herramienta llamada sudo (similar al Mac OS X), con la que se evita el uso del usuario root (administrador).
- ✓ Mejora la accesibilidad y la internacionalización, de modo que el software está disponible para tanta gente como sea posible. En la versión 5.04, el UTF-8 es la codificación de caracteres en forma predeterminada.
- ✓ No sólo se relaciona con Debían por el uso del mismo formato de paquetes deb, también tiene uniones muy fuertes con esa comunidad, contribuyendo con cualquier cambio directa e inmediatamente, y no solo anunciándolos. Esto sucede en los tiempos de lanzamiento. Muchos de los desarrolladores de Ubuntu son también responsables de los paquetes importantes dentro de la distribución de Debían.
- ✓ Todos los lanzamientos de Ubuntu se proporcionan sin costo alguno. Los CDs de la distribución se envían de forma gratuita a cualquier persona que los solicite mediante el servicio Shiplt (la versión 6.10 no se llegó a distribuir de forma gratuita en CD, pero la versión 7.04 sí). También es posible descargar las imágenes ISO de los discos por transferencia directa o bajo la tecnología Bittorrent.
- ✓ Ubuntu no cobra honorarios por la suscripción de mejoras de la “Edición Enterprise”.

#### **6.4.4 Organización de paquetes**

Sistema de gestión de paquetes Synaptic.

Ubuntu divide todo el software en cuatro secciones, llamadas componentes, para mostrar diferencias en licencias y la prioridad con la que se atienden los problemas que informen los usuarios. Estos componentes son: main, restricted, universe y multiverse.

Por defecto, se instala una selección de paquetes que cubre las necesidades básicas de la mayoría de los usuarios de computadoras.

#### 1. El componente main

El componente main contiene solamente los paquetes que cumplen los requisitos de la licencia de Ubuntu, y para los que hay soporte disponible por parte de su equipo. Éste está pensado para que incluya todo lo necesario para la mayoría de los sistemas Linux de uso general. Los paquetes de este componente poseen ayuda técnica garantizada y mejoras de seguridad oportunas.

#### 2. El componente restricted

El componente restricted contiene el programa soportado por los desarrolladores de Ubuntu debido a su importancia, pero que no está disponible bajo ningún tipo de licencia libre para incluir en main. En este lugar se incluyen los paquetes tales como los controladores propietarios de algunas tarjetas gráficas, como por ejemplo, los de nVIDIA. El nivel de la ayuda es más limitado que para main, puesto que los desarrolladores pueden que no tengan acceso al código fuente.

#### 3. El componente universe

El componente universe contiene una amplia gama del programa, que puede o no tener una licencia restringida, pero que no recibe apoyo por parte del equipo de Ubuntu. Esto permite que los usuarios instalen toda clase de programas en el sistema guardándolos en un lugar aparte de los paquetes soportados: main y restricted.

#### 4. El componente commercial

Como lo indica su clasificación, contiene programas comerciales.

#### 5. El componente multiverse

Finalmente, se encuentra el componente multiverse, que contiene los paquetes sin soporte debido a que no cumplen los requisitos de Software Libre.

## 7. CONCLUSIONES

El servicio Cloud privado en la institución permitió ofrecer servicios como Infraestructura, presentando características y oportunidades de desarrollo para los estudiantes bastante complejas, gracias a la calidad y rapidez de los servicios que ofrece OpenStack.

Se comprueba que la evolución acelerada de cloud computing dentro de la Universidad depende de las necesidades de los usuarios y factores como capacidad, seguridad, servicios y otros que la institución vea adecuado para su propósito.

Se comprueba la eficacia y ahorro de dinero en el manejo de máquinas virtuales a través de internet. Hemos construido un entorno de aprendizaje virtual y personal que combina una amplia gama de servicios para crear una herramienta interactiva para la educación sobre la base de los servicios disponibles en la nube.

Debido a la complejidad en la configuración e instalación de OpenStack, se realizaron varias instalaciones, para finalmente obtener la que se adopta a los recursos ofrecidos por la institución.

En este trabajo se comprobó que las tecnologías de computación en nube puede ser explotadas para construir la próxima generación de herramientas e-Learning, para proporcionar el aprendizaje formal e informal inteligente apoyados en tecnologías Cloud. Este conjunto de herramientas tiene la fuerza para distribuir servicios a través de un conjunto amplio de dispositivos y reducir considerablemente el costo global de los elementos de cómputo de una institución.



## 8. BIBLIOGRAFIA

- [1]. OpenStack Foundation. 2014-05-19. OpenStack Installation Guide for Ubuntu 12.04/14.04 (LTS)
- [2]. Academia.edu. OPENSTACK Icehouse,  
[http://www.academia.edu/9785529/OPENSTACK\\_Icehouse](http://www.academia.edu/9785529/OPENSTACK_Icehouse)
- [3]. Openstack. <https://www.openstack.org/>
- [4]. Debitoor. Definición de Cloud computing.<https://debitoor.es/glosario/definicion-cloud-computing>
- [5]. Techtarget.Cloud Computing.<http://searchcloudcomputing.techtarget.com/definicion/cloud-computing>
- [6]. Linux.<http://www.linux.com/>
- [7]. Coursera.org.Cloud Computing.  
<https://www.coursera.org/course/cloudcomputing>
- [8]. Revista Wired,  
[http://archive.wired.com/wired/archive/14.10/cloudware.html?pg=5&topic=cloudware&topic\\_set](http://archive.wired.com/wired/archive/14.10/cloudware.html?pg=5&topic=cloudware&topic_set)
- [9]. Kit Ingeniería Electronica SRL, <http://www.kit.com.ar/boletines-a.php?id=0000032>
- [10]. The NIST Definition of Cloud Computing,  
<http://csrc.nist.gov/publications/nistpubs/800-145/SP800-145.pdf>
- [11]. Revista Cloud Computing  
<http://www.revistacloudcomputing.com/2015/03/cloud-openstack-y-su-madurez/>
- [12]. Luis Joyanes Aguilar.Computación en la nube:Estrategias de cloud computing en las empresas,Primera edición,Alfaomega, México, julio 2012

[13]. The Cloud Computing and Distributed Systems (CLOUDS) Laboratory,  
University of Melbourne <http://www.cloudbus.org/>

[14]. Revista Cloud Computing  
<http://www.revistacloudcomputing.com/2014/10/desde-2007-ibm-ha-invertido-7-000-millones-de-dolares-en-adquisiciones-alianzas-y-lanzamiento-de-soluciones-cloud/>

[15]. Red Hat Enterprise Linux OpenStack Platform  
[https://access.redhat.com/documentation/en-US/Red\\_Hat\\_Enterprise\\_Linux\\_OpenStack\\_Platform/2/html-single/Getting\\_Started\\_Guide/](https://access.redhat.com/documentation/en-US/Red_Hat_Enterprise_Linux_OpenStack_Platform/2/html-single/Getting_Started_Guide/)

[16]. OpenStack Training Guides, <http://docs.openstack.org/icehouse/training-guides/content/>

[17]. Kernel Virtual Machine, [http://www.linux-kvm.org/page/Main\\_Page](http://www.linux-kvm.org/page/Main_Page)

[18]. The virtualization API, <http://libvirt.org/>