

**EL USO DE SOFTWARE LIBRE EN CLOUD COMPUTING
(COMPUTACIÓN EN LA NUBE)**

**JESÚS MAURICIO LÓPEZ LÓPEZ
HENRY LUIS RODRIGUEZ CARDENAS**

**UNIVERSIDAD AUTÓNOMA DE BUCARAMANGA, UNAB
FACULTAD DE INGENIERÍA DE SISTEMAS
MESTRIA EN SOFTWARE LIBRE
BUCARAMANGA
2012**

**EL USO DE SOFTWARE LIBRE EN CLOUD COMPUTING
(COMPUTACIÓN EN LA NUBE)**

**JESÚS MAURICIO LÓPEZ LÓPEZ
HENRY LUIS RODRIGUEZ CARDENAS**

Tesis para optar el Título de Magister en Software Libre

Director: PHD. EDUARDO CARRILLO ZAMBRANO

**UNIVERSIDAD AUTÓNOMA DE BUCARAMANGA, UNAB
FACULTAD DE INGENIERÍA DE SISTEMAS
MESTRIA EN SOFTWARE LIBRE
BUCARAMANGA
2012**

Nota de Aceptación

Firma Presidente del Jurado

Firma del Jurado

Firma del Jurado

Bucaramanga, 25 de Enero de 2012

AGRADECIMIENTOS

Agradecemos a Dios por darnos su bendición y sabiduría para sacar a delante los objetivos propuestos.

A nuestros padres por su apoyo incondicional y soporte en nuestras familias para la realización de nuestros estudios.

A la universidad y en especial a nuestro director de tesis por ser nuestro mentor y con su apoyo y orientación nos permitió que este proyecto saliera a delante.

CONTENIDO

INTRODUCCIÓN.....	17
1. METODOLOGÍA	19
1.1. ENFOQUE Y TIPO DE ESTUDIO.....	19
1.1.1. <i>Seleccionar el diseño apropiado de investigación:</i>	19
1.1.2. <i>Delimitación de la población y de la muestra.</i>	19
1.2. PRODUCTOS ESPERADOS.....	20
2. ESTADO DEL ARTE SOBRE PROYECTOS DE INVESTIGACIÓN EN CLOUD COMPUTING	22
2.1. CLOUD COMPUTING EN EL CAMPO EDUCATIVO	22
2.2. CLOUD COMPUTING Y LA GESTIÓN DE DATOS A GRAN ESCALA	23
2.3. CLOUD COMPUTING EN LOS PROYECTOS CIENTÍFICOS	24
2.4. CLOUD COMPUTING Y VIRTUALIZACIÓN	26
2.5. CLOUD COMPUTING EN GESTIÓN DE RECURSOS Y SERVICIOS	27
2.6. CLOUD COMPUTING Y LA GESTIÓN DE SEGURIDAD	31
2.7. CLOUD COMPUTING Y SUS PERSPECTIVAS	33
3. MARCO TEÓRICO.....	35
3.1. COMPUTACIÓN EN LA NUBE (DEFINICIÓN).....	35
3.1.1. <i>Antecedentes históricos de la Computación en la nube:</i>	37
3.1.2. <i>Clasificación de los modelos de Computación en la Nube:</i>	38
3.1.2.1. La Clasificación de la nube modelo SPI:.....	39
3.1.2.2. La Ontología de la nube de UCSB-IBM:	41
3.1.2.3. Modelo de nube de Hoff:.....	42
3.1.3. <i>Tipos de Computación en la Nube</i>	44
3.1.4. <i>Proveedores actuales de Computación en la nube:</i>	45
3.1.5. <i>Software libre para la implementación de Computación en la nube</i>	45
3.1.6. <i>Impacto de computación en la nube en organizaciones:</i>	48
3.2. EL SOFTWARE LIBRE	49
3.2.1. <i>Historia del Software Libre:</i>	53
3.2.2. <i>Tipos de Licencias en el software libre</i>	54
4. EVOLUCIÓN DE GRID COMPUTING A CLOUD COMPUTING.....	55
4.1. GRID COMPUTING O COMPUTACIÓN DE MALLA.....	55
4.1.1. <i>Arquitectura y Funcionalidad de Grid Computing:</i>	59
4.1.1.1. La capa Fábrica (Fabric):	60
4.1.1.2. La capa Conectividad (Connectivity):.....	60
4.1.1.3. La capa Recursos (Resource):.....	60
4.1.1.4. La capa Colectivo (Collective):.....	60

4.1.1.5.	La capa de aplicación (Application):.....	61
4.1.2.	<i>Clasificación de Grid Computing:</i>	62
4.1.2.1.	Clasificación de Grids de acuerdo con el enfoque de recursos:.....	62
4.1.2.2.	Clasificación de Grids en función del alcance de uso compartido de recursos:	62
4.1.3.	<i>Grid Computing hacia nuevas tecnologías:</i>	66
4.1.3.1.	Convergencia de Grid Computing a computación orientada a servicios (SOC):	67
4.1.3.2.	Convergencia de Grid Computing y Software como servicio (SaaS):.....	69
4.1.3.3.	Evolución hacia Cloud Computing	71
4.2.	CLOUD COMPUTING (COMPUTACIÓN EN LA NUBE)	72
4.2.1.	<i>Modelos de despliegue de Computación en la Nube.</i>	78
4.2.1.1.	Nubes públicas:	78
4.2.1.2.	Nubes privadas:.....	79
4.2.1.4.	Nubes híbridas:.....	79
4.2.2.	<i>Las Tres capas de la arquitectura de la computación en la nube (Cloud Computing)</i>	79
4.2.2.1.	Infraestructura como servicio (IaaS).	80
4.2.2.2.	Plataforma como servicio (PaaS).	81
4.2.2.3.	Software como servicio (SaaS).....	83
5.	EL USO DEL SOFTWARE LIBRE EN LA EVOLUCION DE GRID COMPUTING A CLOUD COMPUTING	84
5.1.	EL USO DE SOFTWARE LIBRE EN GRID COMPUTING (COMPUTACIÓN DE MALLA)	84
5.1.1.	<i>Open Grid Forum (OGF)</i>	84
5.1.2.	<i>El Open Grid Services Architecture (OGSA)</i> :.....	85
5.1.3.	<i>Open Grid Services Infrastructure (OGSI)</i>	85
5.1.4.	<i>Globus Toolkit</i>	86
5.1.5.	<i>Estándares y API's de Grid Computing</i>	86
5.1.6.	<i>Los principales Grid middleware de diferentes proveedores todos bajo licencias libres.</i>	88
5.2.	EL USO DE SOFTWARE LIBRE EN CLOUD COMPUTING (COMPUTACIÓN EN LA NUBE)	89
5.2.1.	<i>Organizaciones que apoyan los estándares abiertos en cloud computing</i>	89
5.2.2.	<i>Herramientas de Software libre en la capa infraestructura como servicio (IaaS)</i>	90
5.2.2.1.	Openstack	90
5.2.2.2.	Open Nebula.	94
5.2.2.3.	Eucalyptus	99
5.2.2.4.	Nimbus:.....	103
5.2.2.5.	Ubuntu Enterprise Cloud Architecture (UEC)	105
5.2.2.6.	Cuadro Comparativo Proveedores de IaaS, basados en Software Libre	106
5.2.3.	<i>Herramientas de Software libre en la capa de plataforma como servicio (PaaS)</i>	108
5.2.3.1.	Google AppEngine.	108
5.2.3.2.	Appscale. Según App scale	112
5.2.3.3.	WSO2 Stratos.	114
5.2.3.4.	Cloud Foundry.	117
5.2.3.5.	Wavemaker.....	119
5.2.3.6.	Tabla Comparativa Proveedores de PaaS, basada en Software Libre.....	123

5.3.1.	<i>Herramientas de software como servicio (SaaS) de Software Libre</i>	125
5.3.1.1.	Aplicaciones Desktop en la nube.	128
5.3.1.2.	Desarrollo de pruebas a EyeOs y Ubuntu One.....	130
5.4.	APLICACIÓN TIPO ESCRITORIO EN CLOUD	139
5.5.	MAPA CONCEPTUAL DE CLOUD COMPUTING (COMPUTACIÓN EN LA NUBE).....	140
6.	PROTOTIPO FUNCIONAL BASADO EN EL MODELO DE PLATAFORMA COMO SERVICIO (PAAS) EN UNA NUBE PRIVADA, CON HERRAMIENTAS DE SOFTWARE LIBRE.	142
6.1.	PROTOTIPO 1 PAAS PRIVADA BASADO EN MICRO CLOUD FOUNDRY	142
6.1.1.	<i>Pasos de implementación del prototipo 1</i>	143
6.1.1.1.	Descarga e Instalación Software de virtualización (Vmware Player v.4.0.1).....	143
6.1.1.2.	Ingreso y registro en el sitio web de Cloud Foundry.	145
6.1.1.3.	Carga y configuración de la maquina virtual de Micro Cloud Foundry.	148
6.1.1.4.	Ejecución y Configuración MV de Micro Cloud Foundry.	149
6.1.1.5.	Instalación de Ruby, RubyGems y VMC (Cliente de línea de Comandos de Vmware)	154
6.1.1.6.	Conexión y creación de usuario en el prototipo1 (PaaS) privado.	157
6.1.1.7.	Uso y Pruebas del Prototipo1 (PaaS) privado.	158
6.1.1.8.	Administración de las aplicaciones.	166
6.1.1.9.	Dificultades en la implementación del Prototipo 1.	168
6.1.1.10.	Conclusiones en la implementación del Prototipo1.....	174
6.2.	PROTOTIPO 2 PAAS PRIVADA BASADO EN WAVEMAKER Y EUCALYPTUS.....	175
6.2.1.	<i>La topología de red</i>	175
6.2.2.	<i>Instalación de Ubuntu Enterprise Cloud (UEC)</i>	177
6.2.2.1.	Instalación UEC maquina A (Front-end)	177
6.2.2.2.	Instalación UEC maquina B (Nodo)	188
6.2.2.3.	Configuración de la nube privada en Ubuntu Enterprise Cloud (UEC).	190
6.2.3.	<i>Instalación y configuración de Plataforma como servicio (PaaS) en la nube privada.</i>	196
6.2.3.1.	Instalación y configuración de Wavemaker.....	196
6.2.3.2.	Inicio de Wavemaker studio PaaS.....	198
6.2.3.3.	Pruebas de Wavemaker studio PaaS.....	200
7.	REFERENCIAS BIBLIOGRÁFICAS.	206

TABLA DE FIGURAS

Figura 1, “Modelo de Ontología de la nube de UCSB-IBM, con tres componentes de la capa de infraestructura en la nube”	42
Figura 2, “Modelo de la Nube de Hoff, Resultado de colaboración en línea y la discusión entre los diferentes expertos en la computación en nube”	43
Figura 3, Arquitectura genérica de Grid Computing.....	59
Figura 4, “Evolución de Grid Computing a través del tiempo”	66
Figura 5, “Definición de las características de la computación en la nube” (Cloud Computing)	77
Figura 6, “Las tres capas de la Computación en la nube: SaaS, PaaS y IaaS”	79
Figura 7, “Componentes Openstack”	91
Figura 8, “Ejemplo de arquitectura Openstack”.....	94
Figura 9 “Funcionamiento OpenNebula”	95
Figura 10 “Arquitectura Interna de OpenNebula”	97
Figura 11 “Infraestructura de OpenNebula”	98
Figura 12 “Descripción detallada de cada uno de los componentes de Eucalyptus”	101
Figura 13 “Elementos de (UEC)”	105
Figura 14 Página principal Google App Engine en español.....	112
Figura 15, Ejemplos de despliegue de AppScale	113
Figura 16, Sitio Web oficial de WSO2 Stratos	116
Figura 17, Página oficial de Cloudfoundry.org	119
Figura 18, Interface de Wavemaker Studio.....	120
Figura 19, Componentes de código abierto de la arquitectura WaveMaker	121
Figura 20, Ingreso a EyeOs.....	131
Figura 21, Interface inicial de EyeOS.....	131
Figura 22, Directorio en EyeOs	132
Figura 23, Abrir documentos en EyeOs.....	133
Figura 24, Manejo de Contactos en EyeOS	133
Figura 25,Chat en EyeOs.....	134
Figura 26, Correo Electrónico en EyOs	134
Figura 27, Sincronización con S.O Locales de EyeOs	135
Figura 28, Instalación de Ubuntu One	135
Figura 29, Sincronizar documentos con Ubuntu one	136
Figura 30, Compartir carpetas en la nube con Ubuntu One	136
Figura 31, Configuración de la cuenta personal en Ubuntu One	137
Figura 32, Mapa Conceptual Computación en la nube, desde el software libre (Parte1)	140
Figura 33, Mapa Conceptual Computación en la nube, desde el software libre (Parte2)	141
Figura 34, Topología de Red Prototipo1	143
Figura 35, Descarga Vmware Player 4.0.1.....	143
Figura 36, Ejecutar el Script de instalación del Vmware Player 4.0.1	144

Figura 37, Inicio de instalación del Vmware Player 4.0.1	145
Figura 38, Instalación Completa del Vmware Player 4.0.1	145
Figura 39, Ingreso a la cuenta de micro cloud foundry	146
Figura 40, Creación nombre de dominio único para micro Cloud Foundry.....	147
Figura 41, Asignación del Token único de Micro Cloud Foundry	147
Figura 42, Descarga MV Micro cloud foundry.....	148
Figura 43, Configuración MV Micro Cloud Foundry	149
Figura 44, Inicio instancia MV Micro Cloud Foundry.....	150
Figura 45, Establecer Pass instancia MV Micro Cloud Foundry	150
Figura 46, Configuración de red instancia MV Micro Cloud Foundry	151
Figura 47, Autenticación de credenciales MV Micro Cloud Foundry	152
Figura 48, Instancia MV Micro Cloud Foundry en la nube privada	153
Figura 49, Prueba funcionamiento Micro Cloud Foundry en la nube privada.	153
Figura 50, Instalación de Ruby y RubyGems	154
Figura 51, Finalización de Instalación de Ruby y RubyGems	155
Figura 52, Instalación de Sinatra Framework.....	155
Figura 53, Instalación de VMC	156
Figura 54, Finalización de Instalación de VMC.....	156
Figura 55, Prueba de funcionamiento de VMC.	157
Figura 56, Conexión y creación de usuario en el Prototipo1 (PaaS) privado.	158
Figura 57, Inicio de sesión en la PaaS privada.....	159
Figura 58, Despliegue de la aplicación Hello2 en Prototipo1 (PaaS) privada	161
Figura 59, Aplicación Hello2 en ejecución Prototipo1 (PaaS) privada	161
Figura 60, Actualización de una aplicación de prueba en Prototipo1 (PaaS) privada.	162
Figura 61, Aplicación Hello2 modificada en ejecución Prototipo1 (PaaS) privada.....	162
Figura 62, Aplicación Hello2 en ejecución desde Windows Prototipo1 (PaaS) privada	163
Figura 63, Visualización de los servicios disponibles en Prototipo1 (PaaS) privada.....	164
Figura 64, Creación y visualización de los servicios disponibles en Prototipo1 (PaaS) privada	165
Figura 65, Despliegue de la aplicación con servicios en Prototipo1 (PaaS) privada.....	166
Figura 66, Administración de las aplicaciones en Prototipo1 (PaaS) privada.	167
Figura 67, Detener una aplicación en el Prototipo1 (PaaS) privada.	167
Figura 68, Error de autenticación Prototipo1 (PaaS) privada	168
Figura 69, Inicio de sesión instancia virtual Prototipo1 (PaaS) privada.....	169
Figura 70, Configuración manual de la interface de red en la MV.....	170
Figura 71, Archivo Interface en la MV de Micro Cloud Foundry.....	171
Figura 72, Reinicio de los servicios de red de la MV.....	172
Figura 73, Error después de instalar VMC, en el cliente	173
Figura 74, Error el desplegar una aplicación por falta de librerías	174
Figura 75, Topología de red de 2 maquinas (IaaS).	176
Figura 76, Componentes de Eucalyptus distribuidos en las maquinas del prototipo.....	176

Figura 77, Inicio Instalación UEC.....	177
Figura 78, Instalación Fron-End	178
Figura 79, Seleccionar Ubicación	178
Figura 80, Conjunción de teclado en UEC	179
Figura 81, Confirmación del teclado	179
Figura 82, Configuración de Red UEC.....	180
Figura 83, Modo de Instalación de la nube en UEC.....	180
Figura 84, Componentes a instalar en UEC	181
Figura 85, Configuración del Reloj	181
Figura 86, partición del Disco Duro en UEC.....	182
Figura 87, Salvar cambio del disco duro y confirmación	182
Figura 88, Inicio Instalación sistema base.....	183
Figura 89, Registro de usuario y súper usuario.	183
Figura 90, Configuración de contraseña usuario nuevo	184
Figura 91, Creación de la carpeta personal.	184
Figura 92, Configuración gestión de paquetes.	185
Figura 93, Configuración potfix.	185
Figura 94, Configuración del Cluster controller.....	186
Figura 95, Configuración de IP publicas de la nube.	186
Figura 96, Configuración de inicio Grub.....	187
Figura 97, Finalización de la Instalación, maquia A.....	187
Figura 98, Inicio de la Maquia A Fron-end	188
Figura 99, Configuración de red nodo 1.....	189
Figura 100, Configuración del modo de instalación del nodo.....	189
Figura 101, Interfaz Web de UEC, en la nube privada, riesgo de seguridad.....	190
Figura 102, Interfaz Web de UEC, en la nube privada, login.....	191
Figura 103, Interfaz Web de UEC, cambio de password.....	191
Figura 104, Interfaz Web de UEC, Acceso al menú de opciones.	192
Figura 105, Interfaz Web de UEC, Parámetros de configuración.	193
Figura 106, Interfaz Web de UEC, Descarga de credenciales de seguridad.....	193
Figura 107, Comandos de obtención y registro de credenciales de autenticación en el servidor.	194
Figura 108, Comprobación disponibilidad del Cluster en la nube privada.....	195
Figura 109, Comando euca_conf - -help en el Cloud Controller	196
Figura 110, Descarga Instalador Wavemaker 6.3 64bits	197
Figura 111, Instalación de Wavemaker 6.3	197
Figura 112, ejecutar el script de arenuque de wavemaker	198
Figura 113, Consola de inicio de wavemaker	198
Figura 114, iniciar wavemaker.....	199
Figura 115, Interface grafica de wavemaker studio	199
Figura 116, Ejecutar una aplicación en wavemaker	200

Figura 117, Aplicación corriendo en una nueva ventana del navegador.	201
Figura 118, Despliegue de una aplicación Wavemaker con IaaS	201
Figura 119, Administración de los servidores en la nube con wavemaker	202
Figura 120, Acceso a la cuenta de Eucalyptus con wavemaker	202
Figura 121, Credenciales de la cuenta de Eucalyptus	203
Figura 122, Lista de instancias en IaaS desde wavemaker	203

ÍNDICE DE TABLAS

Tabla 1, Productos Esperados.....	20
Tabla 2, “Evolución de Grid Computing a Cloud Computing”	71
Tabla 3, Estándares y API’s de Grid Computing.	86
Tabla 4, Grid middleware y sus tipo de licencia.....	88
Tabla 5, Comparación de Proveedores de IaaS basados en Software Libre.....	107
Tabla 6, Cuadro Comparativo proveedores PaaS, basados en Software Libre.....	123
Tabla 7, Cuadro Comparativo Soluciones de SaaS, basados en Software Libre.	139
Tabla 8, Prerrequisitos prototipo 1.....	142
Tabla 9, Prerrequisitos Prototipo 2.....	175
Tabla 10, Análisis de pruebas Plataforma como servicio en una nube privada	204

GLOSARIO

API: Acrónimo en Inglés de (Application Programming Interface) Interfaz de programación de aplicaciones es un conjunto de funciones y procedimientos para ser utilizado por otro software como una capa de abstracción. El API describe la sintaxis de las operaciones del sistema. Para cada operación, la API especifica la información que debe ser enviada al sistema, la información que el sistema le enviará de vuelta, y cualquiera de las condiciones de error que pueda ocurrir.

Arquitectura: Término que identifica la estructura lógica de elementos tecnológicos. Puede hacer referencia a software, hardware o combinación de ambos.

Cloud Computing (Computación en la Nube): Paradigma o Modelo tecnológico de provisión de capacidades computacionales en modo servicio que utiliza Internet como medio de comunicación.

Grid Computing (Computación en la Nube): Tecnología de computación que mediante sistemas distribuidos de recursos computacionales obtiene características de súper computador.

Hipervisor: También denominado monitor de máquinas virtuales, es una plataforma software que permite ejecutar distintos sistemas operativos, virtualizados, sobre un mismo hardware.

IaaS: Acrónimo inglés de Infrastructure as a Service o lo que es lo mismo infraestructura como servicio.

Infraestructura como Servicio: Capa de distribución de computación en la nube que suministra servicios de infraestructura computacional básica, esto incluye hardware, sistemas operativos, almacenamiento o comunicaciones. Acrónimo (IaaS)

Integración: La integración es el proceso de combinación de componentes o sistemas en un sistema global.

Interoperabilidad: la interoperabilidad se refiere a la capacidad de los sistemas de comunicarse. Se requiere que la información comunicada es entendida por el sistema de recepción. En el mundo de la computación en la nube, esto significa la capacidad de escribir código que funciona con el proveedor de la nube más de una vez, independientemente del proveedor.

Maquina Virtual (Virtual Machine VM): archivo normalmente llamado imagen que, al ejecutarse, funciona como una máquina real. En la capa de Infraestructura como Servicio (IaaS) se utiliza a menudo las imágenes en una máquina virtual que puede ser iniciada o detenida, según sea necesario.

PaaS: Acrónimo inglés de Platform as a Service o lo que es lo mismo plataforma como servicio.

Plataforma como servicio: Capa de distribución de computación en la nube que suministra servicios de plataformas de desarrollo e software (programación) de aplicaciones SaaS y de despliegue de las mismas. Acrónimo (PaaS)

Portabilidad: La portabilidad es la capacidad de ejecutar los componentes o sistemas por escrito para un entorno. En el mundo de la computación en nube, incluye los entornos de software y el hardware (tanto físicos como virtuales).

SaaS: Acrónimo inglés de Software as a Service o lo que es lo mismo software como servicio.

Software como Servicio: Capa de distribución de computación en la nube donde se accede a recursos de software o aplicaciones web a través de internet proporcionadas y mantenido por el proveedor del servicio y. Acrónimo (SaaS)

Software Libre: Software licenciado con licencias libres. Software que hace referencia a conceptos similares que permite y la libertad a los usuarios para ejecutar, copiar, distribuir, estudiar, cambiar y mejorar el software.

RESUMEN

El paradigma de computación en la nube ha tenido un desarrollo importante en su estructura y en los servicios que presta a las diferentes organizaciones que miran en este modelo una alternativa a los grandes procesos computacionales e informáticos que manejan. El software libre y sus herramientas han sido uno de los elementos preponderantes para la construcción de soluciones de cloud computing que tengan disponibilidad para cualquier tipo de usuarios y organizaciones.

En el desarrollo del presente documento se hace una descripción teórica acerca de la idea de cloud computing desde sus inicios en el concepto de computación en malla (Grid Computing) hasta la implementación de las nuevas tecnologías de computación en nube que se están aplicando en diferentes áreas de las ciencias, educación y los negocios poniendo especial atención a la participación del software libre en la variedad de soluciones de software de la actualidad.

El presente estudio hace un recorrido por los diferentes tipos de prestaciones de cloud computing especialmente las que han sido desarrolladas con software libre y muestra la implementación de un prototipo de plataforma como servicio utilizando aplicaciones de software libre para una nube privada así como también la descripción de las pruebas de funcionalidad de dos aplicaciones de software como servicio.

Con la implementación de las diversas soluciones de cloud computing en software libre se establece un estado comparativo donde se evidencia aquellas que presentan mejores capacidades y factibilidad de uso.

Palabras claves: cloud computing, software libre, software como servicio, plataforma como servicio, infraestructura como servicio.

ABSTRACT

The paradigm of cloud computing has been an important development in its structure and services provided to organizations that look at this model an alternative to large computing and computer processes they manage. Free software and tools have been one of the predominant elements for building cloud computing solutions that are available for any type of users and organizations.

In the development of this paper provides a theoretical description of the idea of cloud computing since its inception in the concept of grid computing (Grid Computing) to implementation of new cloud computing technologies are being applied in different areas science, education and business with special attention to the participation of free software in a variety of software solutions available today.

This study makes a journey through the different types of cloud computing services especially those that have been developed with free software and shows the implementation of a prototype platform as a service using open source applications for private cloud as well as the results of functional testing of both software applications as a service.

With the implementation of various solutions of open source cloud computing provides a comparative statement which demonstrate those that have better capabilities and feasibility of use.

Keywords: cloud computing, open source, software as a service, platform as a service and infrastructure as a service.

INTRODUCCIÓN

En estos últimos años en el mundo de las tecnologías de la información y las comunicaciones se viene hablando sobre un tema que está cambiando la forma tradicional de implementar y usar los recursos informáticos en la industria y en la vida cotidiana, se trata de la computación en la nube o “Cloud Computing”, un nuevo paradigma, según algunos entendidos de acceder y usar recursos informáticos, pero dentro de estos nuevos avances en el mundo de la tecnología no podemos dejar a un lado la gran influencia y aporte que realiza del Software Libre en todos sus campos.

Esta tesis de investigación para la maestría de software libre se centra en el estudio de la gran influencia que el software libre ha tenido en el desarrollo de las nuevas tecnologías de la información y las comunicaciones y su participación dentro de la computación en la nube “Cloud Computing”.

El presente trabajo realiza una exploración teórica del concepto de computación en la nube (cloud computing) en los diversos proyectos de investigación desarrollados por la comunidad científica los últimos tres años para establecer las líneas de acción en las cuales el concepto de cloud computing ha tenido mayor repercusión, además define los horizontes en los cuales la comunidad científica está trabajando para dar mayor aplicabilidad a las nuevas tecnologías de cloud computing. La presente investigación toma como base las aplicaciones de cloud computing tanto en Infraestructura como servicio (IaaS), Plataforma como servicio (PaaS) y software como servicio (SaaS) con el fin de encontrar aquellas que cumplan con los principios del software libre y establecer estados comparativos de las mismas para determinar cuáles de ellas pueden ser las que tengan mejores prestaciones y funcionalidades con el objetivo de que sean utilizadas con un nivel de calidad y confianza.

Esta investigación seleccionó soluciones plenamente de software libre que tienen su aplicación en cloud computing y con ellas se desarrolló el proceso de instalación e implementación lo cual permitió definir puntos de comparación entre las mismas y presentar un estado de calidad y un nivel de utilización de acuerdo a sus características y prestaciones.

Con el presente trabajo se espera fomentar el uso de aplicaciones de software libre en el paradigma de cloud computing estableciendo los parámetros más recomendables para la implementación de soluciones de nubes privadas o públicas donde su factibilidad de puesta en marcha sea más cercana con la utilización de herramientas de software libre.

1. METODOLOGÍA

1.1. ENFOQUE Y TIPO DE ESTUDIO.

La investigación realizada corresponde a una aplicada documental y es de carácter exploratorio.

1.1.1. Seleccionar el diseño apropiado de investigación: Para el logro de los objetivos propuestos se aplicó una investigación de tipo exploratoria con el fin de determinar los conceptos generales acerca de la computación en la nube y sus aplicaciones como también del concepto de software libre y su relación con cloud computing.

1.1.2. Delimitación de la población y de la muestra.

Universo: Soluciones o aplicaciones web de cloud computing (Computación en la nube)

Muestra: Soluciones cloud computing (Computación en la nube) basades en Software Libre.

El proyecto se desarrolló con una metodología estructurada en las siguientes fases:

FASE 1: Para el desarrollo del proyecto en primer término se hizo una revisión bibliográfica de los conceptos de Grid Computing como también la exploración teórica de los conceptos de Cloud Computing para establecer la línea de evolución y la relación que los vincula.

FASE 2: También se desarrolló una exploración teórica de los proyectos de investigación los últimos tres años respecto al tema de computación en la nube y sus campos de aplicación, esta actividad se ejecutó con la lectura sistemática de artículos, ponencias y escritos que se produjeron sobre el tema en los principales sitios web de revistas científicas y especializadas.

FASE 3: Con la conceptualización de Cloud Computing y sus campos de acción se analizaron las soluciones y aplicaciones de software libre que tienen su utilidad en la computación en la nube con el fin de desarrollar instalaciones y pruebas técnicas de funcionalidad para establecer un marco de comparación entre

infraestructuras, plataformas y aplicaciones de software como servicio en donde el paradigma de cloud computing interactúa.

1.2. PRODUCTOS ESPERADOS

Tabla 1, Productos Esperados

OBJETIVOS	PRODUCTOS ESPERADOS	ACTIVIDADES
Comparar elementos conceptuales y de desarrollo de software en relación con desarrollo de software para Grid y Cloud Computing.	Documento donde se evidencie la evolución de grid computing y su relación con cloud computing.	<ul style="list-style-type: none"> • Revisiones bibliográficas en sitios web y publicaciones científicas. • Resumen y análisis documental de publicaciones. • Redacción del documento de resumen.
Generar un estado del arte actualizado sobre proyectos de investigación que se estén desarrollando sobre cloud computing.	Documento de evidencia del desarrollo de las investigaciones que sobre cloud computing se están desarrollando actualmente.	<ul style="list-style-type: none"> • Revisión de fuentes bibliográficas de sitios web y publicaciones científicas en medio digital. • Categorización, resumen y análisis de las investigaciones. • Redacción del documento.
Realizar un prototipo funcional basado en la arquitectura de plataforma como servicio (PaaS software as a service) a partir de la reutilización de software libre existente.	- Entrega de un documento donde se evidencien los resultados de la implementación y pruebas del prototipo mediante el uso de herramientas libres.	<ul style="list-style-type: none"> • Investigación de soluciones de software libre sobre cloud computing. • Clasificación de soluciones sobre Infraestructura, Plataforma y Software como servicio. • Selección de

		<p>herramientas de software factibles de ser implementadas.</p> <ul style="list-style-type: none"> • Desarrollo de instalación y pruebas de funcionalidad. • Entrega de cuadro de resultados sobre implementación.
<p>Generar un documento que resuma el estado actual del desarrollo de soluciones basadas en computación en la nube (cloud computing), haciendo énfasis en software como servicio (SaaS), incluyendo la documentación de dos pruebas realizadas en software libre existente basado en la arquitectura de Software como servicio (SaaS software as a services).</p>	<p>- Entrega un artículo publicando las pruebas de soluciones de en computación en la nube (cloud computing), basadas en el modelo de software como servicio (SaaS software as a services).</p>	<ul style="list-style-type: none"> • Selección del marco teórico de la investigación. • Redacción de conceptos relevantes de la investigación. • Presentación de cuadros de resultados sobre los procesos de instalación y pruebas. • Redacción del documento resumen con conclusiones.

Fuente, Elaboración Propia.

2. ESTADO DEL ARTE SOBRE PROYECTOS DE INVESTIGACIÓN EN CLOUD COMPUTING

De acuerdo a los proyectos de investigación analizados sobre cloud computing en los últimos tres años se puede determinar unas tendencias en las cuales se ha profundizado en el campo investigativo y que se clasifican de la siguiente manera:

2.1. Cloud Computing en el campo educativo

Una primera aplicación comprende el concepto de computación oportunista que consiste en aprovechar los recursos existentes entre equipos de cómputo y servidores subutilizados con el fin de construir una nube de computación para prestar servicios educativos óptimos sin acarrear en costos representativos, es la que la Universidad de los Andes experimentó con su modelo UNACLOUD¹, la investigación aplica un modelo IaaS para prestar servicios computacionales básicos haciendo uso de recursos existentes en la universidad y la utilización de software libre en su mayoría.

La puesta en práctica del desarrollo cooperativo de software para la enseñanza es un proyecto que toma como base herramientas libres de cloud computing utilizando la plataforma EyeOS, un sistema operativo libre para montar aplicaciones de internet desarrolladas en JavaScript y PHP, tomando estas aplicaciones se construye un proyecto educativo para enseñanza de asignaturas en la Universidad de Belgrano en Argentina².

Las funcionalidades de cloud computing pueden hacer servicios de aprendizaje electrónico más inteligente y más eficiente por ello los servicios de e-learning incluyen una arquitectura más cercana al usuario, con esta investigación, se propone un nuevo concepto de servicio que proporciona el contexto para la conciencia inteligente de contenidos de aprendizaje en un entorno de cloud computing. Se sugiere las 4 inteligencias elásticas (E4S). El E4S se enfoca en satisfacer las necesidades de recolección y análisis de los usuarios el comportamiento de los usuarios, los servicios de prospección de futuro, la creación de contenidos correspondientes, y entregar el contenido a través del

¹ ROSALES FORERO UNA-CLOUD: Infraestructura como servicio para Cloud Computing Oportunista, 2011

² AGUILERA, Sergio Omar. Desarrollo de Software Libre aplicado a la enseñanza de Sistemas Operativos sobre plataforma Cloud Computing. Universidad de Belgrano, Argentina 2011.)

entorno de cloud computing. El comportamiento de los usuarios se pueden recoger a través de dispositivos móviles como teléfonos inteligentes que se han incorporado en los sensores. Como resultado, la propuesta inteligente de e-learning en el modelo de entorno de cloud computing ofrece servicios de formación personalizada y adaptada a sus usuarios³.

Una iniciativa importante para el tratamiento del lenguaje es CluE, este proyecto busca implementar un sistema de traducción automática, que promete reducir la brecha del lenguaje en la sociedad multicultural. Sistemas capaces de convertir el texto de un idioma a otro tienen el potencial de transformar la manera en que los diversos individuos y comunidades se comunican. Mediante el acoplamiento de análisis de red con técnicas de lenguaje cruzadas el resultado es un rico modelo contextual multilingüe que guiará un sistema de traducción automática en la traducción de diferentes tipos de texto. El impacto potencial más amplio de este proyecto es nada menos que la difusión del conocimiento a través de las fronteras lingüísticas⁴.

2.2. Cloud Computing y la Gestión de datos a gran escala

Para el proyecto Cloud_Store de la Universidad de San Diego California uno de sus objetivos investigativos es explorar nuevas formas de almacenar cantidades extremadamente grandes de datos en el modelo de computación en la nube, específicamente en aplicaciones de datos topográficos⁵.

El proyecto estudia estrategias dinámicas para el aprovisionamiento de aplicaciones haciendo una evaluación del desempeño de los enfoques alternativos para servir a grandes conjuntos de datos. Las plataformas de nubes que se utilizan son el grupo Pista Google-IBM y HP-Intel-Yahoo Cluster y el banco de pruebas de Open Cloud Computing Cirrus de la Universidad de Illinois. La aplicación permite a los usuarios obtener un subconjunto de datos de teledetección, procesamiento mediante diferentes algoritmos y muestra de salida de resultados.

³ KIM Svetlana. Smart Learning Services Based on Smart Cloud Computing. SENSORS 2011

⁴ LIN Jimmy, Research and Education with MapReduce/Hadoop: Data Intensive Text Processing and Beyond. University of Maryland 2009

⁵ CHAITAN Baru, Performance Evaluation of On-Demand Provisioning Strategies for Data Intensive Applications, Universidad de California San Diego.

Otra faceta en la que cloud computing tiene su radio de acción es el análisis de datos a gran escala en enfoques de clusters, este proyecto desarrollado por MIT, la universidad de Wisconsin y la Universidad de Yale compara estos diferentes sistemas de bases de datos como MapReduce y el procesamiento paralelo escalable a cientos de nodos, sin embargo es importante para los investigadores conocer las diferencias de rendimiento y la escalabilidad de estos dos enfoques para conocer cual es el más adecuado en el diseño de nuevas aplicaciones intensivas de datos informáticos⁶.

2.3. Cloud Computing en los proyectos científicos

Son múltiples los campos de la ciencia en los que los servicios de cloud computing pueden ser de utilidad uno de ellos puede ser el proyecto CloVR una herramienta de análisis de la secuencia genómica automática utilizando máquinas virtuales, esta investigación desarrollada por la Universidad de Maryland y Baltimore, toma como base un software portable e independiente que se ejecuta en máquinas virtuales para el análisis del genoma, el tratamiento de los datos de la secuencia de gran tamaño puede ser subcontratados a las grandes nubes de computación. Los protocolos de análisis proporcionado en el paquete virtualizado permiten replicar y extender los protocolos establecidos de bioinformática e incluyen herramientas para la totalidad de su genoma y el análisis comparativo, incluido el montaje de secuencias, predicción de genes, anotación funcional, la reconstrucción de vías metabólicas, y la clasificación filogenética. Con la disponibilidad del software en código abierto y su disponibilidad en la nube puede facilitar la secuenciación de los genomas microbianos a una amplia comunidad de científicos⁷.

Un área de aplicación de cloud computing esta relacionada en los dominios del concepto de e-ciencia, con la plataforma de computación se puede explotar el poder de la computación en la nube mientras que proporciona abstracciones para los científicos y crear flujos de trabajo de datos altamente escalable de procesamiento. La reciente masificación de sitios web basados en internet y las

⁶ MADDEM Sam, A Comparison of Approaches to Large-Scale Data Analysis, MIT - University of Wisconsin, University of Yale. 2009

⁷ FRICKE Florian, CLOVR, A Genomics tool for automated and Portable Sequence Analysis using virtual machines and cloud computing. University Maryland and Baltimore.

herramientas de e-Ciencia han proporcionado a los científicos con una amplia gama de nuevos métodos para recopilar, informar, analizar y compartir sus datos. Con el proyecto de investigación ESCALA se presenta una plataforma de análisis de e-Ciencia, los datos experimentales. ESCALA apoyan para facilitar la colaboración a través de una comunidad de los científicos con el apoyo mínimo en las operaciones de IT y especialistas en computación, y el uso de los recursos de computación en la nube (privado o del público) de manera independiente de la plataforma para ofrecer computación de alto rendimiento.

Estas tecnologías basadas en la Web han demostrado el potencial de permitir mayor colaboración y facilitar el intercambio y reutilización de los datos experimentales. La amplia disponibilidad de conjuntos de datos de calidad y los flujos de proceso de prueba son hechos de buen complemento para el e-científicos⁸.

Hoy, muchas aplicaciones en tiempo real como ejemplo las geoespaciales (la navegación y servicios basados en la localización) se refieren a datos o a tareas intensivas de geoprocesamiento donde el rendimiento es de gran importancia. La computación en nube, una plataforma prometedora con una gran cantidad de recursos de almacenamiento y computación, podría ser una solución práctica para albergar grandes cantidades de datos y de procesamiento en tiempo real. En este proyecto ha explorado la posibilidad de utilizar Google App Engine (GAE), la tecnología cloud computing de Google, para un módulo de servicios de navegación, llamada GNSS (iGNSS). El objetivo de este módulo es predecir la calidad de soluciones de posicionamiento para las rutas posibles previamente. iGNSS QoS consiste en el cálculo en tiempo real de las grandes redes Triangular Irregular (TIN) generados a partir de datos LiDAR. Los resultados experimentales revelan la computación en nube puede potencialmente ser utilizado para el desarrollo y despliegue de datos y / o aplicaciones de cálculo intensivo geoespacial⁹.

La necesidad de compartir una plataforma donde se pueda existan recursos de investigación sobre cloud computing llevó al diseño de Open Cirrus que es un proyecto que se constituye en un banco de pruebas de cloud computing y tiene

⁸ AJITH Ranabahu, the Cloud Agnostic e-Science Analysis Platform. *Wright State University*

⁹ KARIMI Hassan Ali, Exploring Real-Time Geoprocessing in Cloud Computing: Navigation Services Case Study. *TRANSACTIONS IN GIS*. 2011

como objetivo de proporcionar una plataforma para pruebas individuales de centros de datos heterogéneos distribuidos de los sistemas, aplicaciones, servicios, investigación y desarrollo de código abierto.

El proyecto es una iniciativa conjunta patrocinada por Hewlett-Packard, Intel y Yahoo en colaboración con la NSF, la Universidad de Illinois, el Institute of Technology de Singapur y la academia rusa de las ciencias.

Open Cirrus ofrece una colección de experimentaciones de datos, Investigadores de cloud computing a menudo carecen de los conjuntos de datos para llevar a cabo una alta calidad experimental de evaluaciones. Open Cirrus permite importar a los investigadores, almacenar y compartir grandes conjuntos de datos. Con estas instalaciones, Open Cirrus podría convertirse en un "pozo de agua", donde los investigadores con intereses similares pueden intercambiar conjuntos de datos y desarrollar puntos de referencia estándar de la computación en nube.

Open Cirrus tiene como objetivo desarrollar las pilas de código abierto y las API para la nube. Open Cirrus se constituye como una plataforma que la comunidad de código abierto puede utilizar para diseñar, implementar y evaluar dichos códigos e interfaces para todos los niveles.

Open Cirrus se concibió por tres razones:

- colaborar en un solo esfuerzo, lograr un mayor impacto que los participantes de forma individual.
- Las pruebas en los entornos de sitio diferente para mejorar la calidad de software y servicios,
- (Timothy.)La concentración de recursos para mejorar la eficiencia, ya que los sitios van a compartir las innovaciones¹⁰.

2.4. Cloud Computing y Virtualización

Uno de los propósitos de los proveedores de Cloud Computing es la reducción de los consumos de energía tanto para disminuir los costos operativos como también para garantizar la fiabilidad de los sistemas.

¹⁰ AVETISYAN Arutyun, Open Cirrus: A global cloud computing testbed. Institute for System Programming of the Russian Academy of Sciences.2010.

Para Kim¹¹ con su investigación, “Power-aware provisioning of virtual machines for real-time Cloud services” propone el aprovisionamiento de máquinas virtuales para servicios de tiempo real, este enfoque permite virtualizar los centros de datos dinámicos con planes de escalado de frecuencia.

A su vez una tecnología que permite la migración de máquinas virtuales en una red de área local LAN ha permitido la facilidad en el procesamiento de la información, sin embargo el alcance de gestión de los recursos ha cambiado, ahora se espera la migración de máquinas virtuales a redes de área extendida WAN para transformar el nivel de provisión de los recursos hacia múltiples centros de datos alrededor del país o en el mundo entero.

Con este objetivo el proyecto CloudNet¹² pretende construir un modelo donde varias plataformas de computación en la nube se vinculan en una infraestructura de red basada en redes privadas virtuales VPN para proporcionar una conectividad segura y sin problemas entre las empresas y los centros de datos en la nube.

Para hacer realidad esta visión de colocar en común recursos de datos diseminados en diversos centros de datos distantes geográficamente, CloudNet ofrece soporte optimizado para la migración en vivo de máquinas virtuales en redes WAN. En concreto son una serie de optimizaciones que reducen al mínimo los gastos de transporte y almacenamiento de la memoria de la máquina virtual durante migraciones que demanden más ancho de banda y bajar la latencia en los enlaces de internet. Este proyecto se evaluó en una nube de los Estados Unidos con cuatro máquinas virtuales en centros de datos de Texas e Illinois para bajar el tiempo de migración en un 65% y disminuir el consumo de ancho de banda para almacenamiento y transferencia en un 50%.

2.5. Cloud Computing en Gestión de recursos y servicios

Los servicios de Cloud Computing se hacen más numerosos y dinámicos y el aprovisionamiento de los recursos es cada vez más difícil. Una asignación en calidad de servicio con recursos limitados es un problema que se considera en la

¹¹ (KIM Kyong Hoon. Power-aware provisioning of virtual machines for real-time Cloud services. CONCURRENCY AND COMPUTATION-PRACTICE & EXPERIENCE. 2011.

¹² WOOD Timothy. CloudNet: Dynamic Pooling of Cloud Resources by Live WAN Migration of Virtual Machines. ACM SIGPLAN NOTICES. 2011

investigación de WEI Guiyi¹³ “A game theoretic method of fair resource allocation for cloud computing services” en el que el demandante de servicio busca prestaciones sofisticadas a través de una red basada en la nube con un calculo dependiendo de la cantidad de servicio prestado.

La teoría de juegos se utiliza para resolver el problema de la asignación de recursos y en este caso una es una solución que se propone en la cual en primer lugar, cada participante resuelve su problema óptimo de forma independiente, sin tener en cuenta la multiplexación de las asignaciones de recursos. Una programación entera binaria es un método que se propone para resolver la optimización independiente. En segundo lugar, un mecanismo de proceso evolutivo, el cual cambia las estrategias de multiplexado de la solución inicial óptima de los distintos participantes para minimizar sus pérdidas de eficiencia. Además con el desarrollo de servicios de aplicación que proporciona la nube informática, cada vez más empresas medianas o pequeñas invierten en sus propios sistemas de información y el uso de servicios de software e incluso plataforma de servicios prestados por profesionales de empresas de servicios de información.

Estas compañías ofrecen servicios de información del sistema de gestión de información servicios como los recursos públicos para apoyar la operación de procesos de negocio de sus clientes. Sin embargo, para la transformación entre los sistemas de información empresarial y los de cloud computing, no hay ningún proveedor de servicios de aplicaciones pueden satisfacer la totalidad de requisitos funcionales de los sistemas de información de una empresa. Por lo tanto, las empresas tienen que utilizar sistemas distribuidos en diferentes nubes.

Con la investigación de Quing¹⁴ “Application-Services Integration among Multi-clouds for Small and Middle Scale Enterprises” se propone un marco para integrar las aplicaciones desplegadas en diferentes nubes. Una plataforma de operación a través de diferentes nubes de una empresa, esta técnica de los procesos de negocio orientados también se desarrolla para mejorar la viabilidad de sistemas de información en un entorno de cloud computing.

¹³ WEI Guiyi. A game theoretic method of fair resource allocation for cloud computing services. Springer Science+Business Media, LLC 2009.

¹⁴ QUING, Li, Application-Services Integration among Multi-clouds for Small and Middle Scale Enterprises. Tsinghua University Beijing. 2010.

Otra faceta que hay que tomar en cuenta es la vigilancia de los recursos para la buena gestión de cloud computing y una adecuada programación de tareas y balanceo de carga.

En el análisis de aspectos como la facturación de aplicaciones comerciales por utilización de recursos por parte de los usuarios, la virtualización, la escalabilidad y la transparencia del ambiente de computación en la nube, los métodos de vigilancia de los recursos existentes de la computación distribuida y el grid computing no puede satisfacer el ambiente de computación en la nube por completo. Por lo tanto, de acuerdo a las características de las plataformas de computación en cloud, Junwei Ge¹⁵ presenta un modelo de supervisión debidamente adaptado al ambiente de computación en la nube, que combina el VMM (Monitor de máquina virtual) y los lenguajes C / C++ y Java para obtener la información sobre el estado de los recursos.

Una dificultad en cuanto a los modelos de cloud computing actuales radica en que los sistemas existentes no son compatibles mecanismos y políticas para la distribución de la carga dinámica de coordinación entre diferentes centros de datos en la nube con el fin de determinar la ubicación óptima para servicios de alojamiento de aplicaciones para lograr razonables niveles de calidad en el servicio. Además los proveedores de cloud computing son incapaces de predecir la distribución geográfica del consumo de los usuarios de acuerdo a sus servicios, por lo tanto, la coordinación de la carga que debe suceder de forma automática, y la distribución de los servicios debe cambiar en respuesta a cambios en el de carga.

Para contrarrestar este problema, se propone la creación de un ambiente de cloud computing de tipo federado (InterCloud)¹⁶ que facilite el servicio oportunista y exista un aprovisionamiento escalable de servicios de aplicaciones, la manera de lograr los objetivos de calidad de servicio bajo condiciones variables de carga de trabajo, de recursos y de la red.

El objetivo general es crear un entorno informático que apoye la expansión o contracción dinámica de las capacidades (maquinas virtuales, servicios,

¹⁵ JUNWEI Ge. Research on the Resource Monitoring Model Under Cloud Computing Environment. Springer-Verlag Berlin Heidelberg 2010

¹⁶ KIM Svetlana, Smart Learning Services Based on Smart Cloud Computing. SENSORS. 2011

almacenamiento y base de datos) para el manejo variable en la demanda de servicios. Con este proyecto se enmarca la visión, los retos, y los elementos arquitectónicos de la InterCloud de servicios públicos orientados a una distribución descentralizada de ambientes cloud.

La propuesta de entorno InterCloud apoya la ampliación de las aplicaciones a través de múltiples nubes de proveedores. Se ha validado este enfoque mediante la realización de una serie de rigurosos estudios de evaluación de desempeño utilizando el kit de herramientas denominado CloudSim. Los resultados demuestran que el modelo de cloud computing federada tiene un inmenso potencial, ya que ofrece importantes mejoras de rendimiento en lo que respecta al tiempo de respuesta y ahorro de costes en escenarios de carga de trabajo dinámica.

El nivel de utilidad de los servicios de Cloud computing también se amplía a los dispositivos electrónicos de mano como Smartphones o Notebooks que constituyen unos ambientes de computación personal generalizada.

El almacenamiento de datos y el intercambio es difícil para estos dispositivos por la cantidad de datos y las limitaciones naturales de los aparatos móviles, tales como el espacio de almacenamiento limitado y la capacidad de computación limitada. Puesto que las soluciones de almacenamiento de nube emergentes puede proporcionar almacenamiento fiable y sin límites, que cumplan a la exigencia de la computación ubicua. Así, se ha diseñado una plataforma de almacenamiento en la nube, que incluye nuevas una serie de servicios de almacenamiento de instantáneas para hacer frente a estos nuevos retos de gestión de datos en ambientes de computación ubicua, que se llama como "SmartBox"¹⁷.

En SmartBox, cada dispositivo está asociado a su almacenamiento con una única cuenta, y el almacenamiento actúa como centro de respaldo, así como repositorio personal cuando el dispositivo está conectado. Al facilitar la navegación por los archivos, todos los conjuntos de datos en el almacenamiento se organizan en base a atributos de archivo apoyan a los usuarios para buscar los archivos por las consultas semánticas. Se implementó un prototipo SmartBox que se centra en los entornos omnipresentes compuestos por dispositivos accesibles a Internet.

¹⁷ WEIMING Zheng, Design a cloud storage platform for pervasive computing environments.

2.6. Cloud Computing y la gestión de Seguridad

Como garantizar niveles de seguridad adecuados tanto para las plataformas de servicios como también para la capa de los usuarios es una inquietud que se ha planteado entre los investigadores del paradigma de cloud computing, los siguientes son algunos de los proyectos que abordan esta temática:

Los primeros requerimientos por parte de los clientes de cloud computing son mantener y verificar la integridad de las capas de software donde se ejecutan los sistemas remotos y también proteger los datos sensibles de los usuarios.

Sin embargo, debido al hecho de que las capas de software que se ejecutan en plataformas en la nube son generalmente suministradas por diferentes proveedores que son potencialmente desconfiados el uno del otro, el problema de la medición y la protección de la integridad del sistema de ejecución se hace muy difícil y no ha sido bien abordado todavía.

Cheng¹⁸ en su investigación “Building dynamic and transparent integrity measurement and protection for virtualized platform in cloud computing” presenta una medición de la integridad y la arquitectura de la protección de capas de software se ejecuta en un del sistema operativo invitado (OS) de una plataforma virtual en el entorno de nube. Esta solución no cambia el sistema operativo invitado, y por lo tanto es transparente al sistema operativo autorizado. Además, la arquitectura asegura que información confidencial de los usuarios está protegida una vez que la integridad de los paquetes de software se divide en tiempo de ejecución.

En este trabajo se pone en práctica la solución en Xen, y se presenta un modelo Nimbus sencillo basado en prototipos. Se demuestra la capacidad de detectar dinámicamente el cambio de la integridad de los programas de computación en la nube, y los resultados de la evaluación muestran que la solución es efectiva para la protección de la integridad con un rendimiento aceptable.

La característica principal de esta solución es que se puede mejorar la seguridad de un tráfico comercial ordinario plataforma con las mismas capacidades

¹⁸ CHENG. Ge. Building dynamic and transparent integrity measurement and protection for virtualized platform in cloud computing, Huazhong University of Science and Technology, People's Republic of China. 2010.

proporcionadas por el coprocesador de seguridad de IBM 4785. La solución no sólo indica las medidas y los informes de la integridad de un sistema, sino que también protege la información confidencial y garantiza la integridad del sistema. El enfoque se puede aplicar en la computación en nube o la red entornos con varios organismos independientes para proteger sus datos sensibles y de mantener la integridad de todo un sistema.

El segundo aspecto importante es el de la seguridad en la plataforma de servicios para cloud computing, especialmente en las organizaciones que descargan sus servicios de software de proveedores. El nivel de servicio de estos proveedores puede correr el riesgo de una falta de concordancia con la capa de seguridad, no solamente se quiere que los proveedores de cloud aislen unos clientes de los demás, también se necesita que existan los medios para aislar los componentes y los usuarios de la aplicación para cada cliente.

Bacon¹⁹ en su investigación “Enforcing End-to-End Application Security in the cloud” propone un enfoque basado en principios para el diseño y despliegue final de software seguro, distribuido a través del etiquetado minucioso, implacable del sentido de seguridad de datos, de forma análoga a lo que ya está hecho para los tipos de datos. El objetivo es garantizar que, por encima de un pequeño código de confianza de base de datos no pueda ser filtrada por los componentes de software defectuoso o malicioso.

Esto es crucial para las infraestructuras de nube, en la que los datos almacenados y los servicios alojados todos tienen distintos propietarios cuyos intereses no son alineados (e incluso puede estar en la competencia), el proyecto desarrolla un sistema de etiquetado de datos y técnicas de aplicación que puede ayudar a formar una base de código de confianza.

Por otro lado los usuarios se preocupan considerablemente por la seguridad de sus datos personales, y las plataformas de cloud computing no han proporcionado un ambiente informático que sea de plena confianza para los clientes, mientras tanto, la gestión del dominio de plataforma de cloud computing está sujeta a ataques maliciosos, que pueden tener afectar a la fiabilidad del entorno informático virtual. Con el proyecto de diseño de TRIOB²⁰ un mecanismo de enlace con un

¹⁹ BACON Jean. Enforcing End-to-End Application Security in the cloud. University of Cambridge. 2010.

²⁰ FANG Haifeng. TRIOB: A trusted virtual computing Environment based on Remote I/O binding mechanism. Institute of Computing Technology, Chinese Academy of Sciences. 2010

ambiente de computación virtual confiable, se presenta un nuevo enfoque para construir una computación virtual confiable en los centros de datos. Por medio de tecnologías innovadoras, los datos del usuario pueden ser almacenados de forma remota en recursos de almacenamiento de confianza, en un entorno del usuario de computación virtual aislado y el usuario puede detectar automáticamente los ataques contra el gestor de dominios de cloud computing.

2.7. Cloud Computing y sus perspectivas

Una visión que se busca hacer realidad es un modelo de cloud computing que permita a los ciudadanos componer libremente sus servicios de forma personalizada en la infraestructura de administración pública, se pretende consolidar una arquitectura adecuada para la orientación a servicios (SOA), con un enfoque de funcionalidades escalables e interoperables para que cualquier persona pueda diseñar un servicio convenientemente.

Con este modelo flexible los ciudadanos podrían auto-componer sus servicios de diversas maneras orientado tanto a los llamados nativos digitales como a individuos con mínimas experiencias en tecnologías.

Las características de este nuevo servicio a los ciudadanos basado en cloud computing está destinado a proporcionar funcionalidad en gestión de información del sector público en un modelo llamado "silo" donde se permita proporcionar múltiples puntos de acceso que se combinen con tareas complejas.

El modelo propuesto se basa en tres niveles diferentes:

- El nivel de conectividad basadas en la web, destinadas a garantizar que las señales de los viajes entre los nodos de nube sea en forma encriptado.
- El nivel de señalización: la clásica arquitectura cliente-servidor utiliza actualmente en la web sistemas se unirán a un sistema basado en señales y conectores, donde cada nube agente se emiten señales, todos los datos de viaje en el formato estándar de la señal y será dirigida hacia conectores específicos expuestos por los nodos de la nube.

- El nivel de tarea, es necesario establecer una conexión entre la nube y los nodos con el fin de lograr una tarea, a este nivel, la agencia y el ciudadano deben

alinearse en el mismo protocolo, aunque el paquete de servicios depende de otros nodos en la nube²¹.

En cuanto a servicios de Cloud Computing, el sector de los países en desarrollo ha recibido considerable atención de empresas locales y globales, los gobiernos nacionales, y los organismos internacionales. Por ejemplo, IBM ha establecido la computación en nube, centros en China, India, Vietnam, Brasil y Corea del Sur. Otras entidades del orden mundial como Microsoft, VMware, Salesforce, Dell, y Parallels están activamente buscando oportunidades en países en desarrollo, clouds relacionados con el capital riesgo y otras inversiones también están fluyendo hacia economías en desarrollo.

Es probable que sea justo decir que en ninguna otra las innovaciones tecnológicas importantes del mundo en desarrollo ha recibido este nivel de atención, sin embargo, los hallazgos y conclusiones sobre el potencial y el impacto de la computación en nube en el mundo en desarrollo a partir de encuestas, estudios y experiencias de las empresas son muy confusas y contradictorias. Algunos analistas sugieren que los países en desarrollo serán los mercados más atractivos para los servicios en la nube, y predicen que esta tecnología pronto estará en la banca, la educación, salud. Las economías en desarrollo podrían alcanzar a los países desarrollados, como la nube que les da acceso a la misma infraestructura de TI, centros de datos y aplicaciones. La nube podría decirse que reduce los costes de infraestructura y los niveles del campo de juego para las pequeñas y medianas empresas.

Además, el software en la nube sería más fácil de instalar, mantener y actualizar a los clientes basada en la computación, que requiere la instalación y configuración de software y su actualización con cada nueva versión, así como la revisión de otros programas con uno de cada tres actualizaciones, servicios en la nube a los usuarios la flexibilidad de ampliación cuando la demanda se incrementa, sus defensores también argumentan que la nube como el software libre se convierte en a través de aplicaciones basadas en Web o disponibles en el software como servicio (SaaS), la piratería de software podría disminuir. Una se refiere al punto final, pero no secundaria del mundo en desarrollo basada en las actividades de TI de las empresas del sector de la nube.

²¹ MARASSO Lanfranco. *Allowing Citizens to Self-compose Personalized Services: A Cloud Computing Model*. Springer-Verlag Berlin Heidelberg 2010

Poca, o ninguna, evidencia empírica muestra la eficacia de estas teorías, ideas y especulaciones se puede traducir en la práctica, sin embargo. Una mirada de cerca en las primeras etapas del desarrollo de la industria de la nube indica que estas observaciones podrían subrayar cómo los problemas económicos e institucionales siguen siendo fundamentales para el desarrollo y la difusión de la información y las comunicaciones (TIC) y el rendimiento empresarial en el mundo en desarrollo. Por un lado, la nube está inherentemente vinculada a las múltiples facetas de la seguridad de los datos. Los estudios recientes indican que muchas organizaciones no están preparadas para la seguridad de sus nubes.

La nube es una espada de doble filo desde el punto de vista de seguridad. A pesar del potencial de la nube para ofrecer seguridad de bajo costo, las pequeñas empresas pueden aumentar los riesgos de seguridad mediante el almacenamiento de datos sensibles en la nube. Los crímenes de las fuentes objeto de valor. Las grandes redes de empresa ofrecen más objetivos a los piratas informáticos. Los proveedores de nubes, que a menudo son más grandes que sus clientes, son objetivos atractivos. Más importante aún, la información almacenada en las nubes es un potencial mina de oro para los cibercriminales²².

3. MARCO TEÓRICO.

3.1. COMPUTACIÓN EN LA NUBE (DEFINICIÓN).

²² Weimar Díaz Garzón, realidad de la computación Grid en América Latina, [En línea],[Julio2011]. Disponible en: http://www.sociedadelainformacion.com/12/Paper_Grid.pdf

Según el National Institute of Standards and Technology NIST²³ “Computación en la nube”, es un modelo para habilitar el acceso a un conjunto de servicios computacionales (redes, servidores, almacenamiento, aplicaciones y servicios) de manera conveniente y por demanda, que pueden ser rápidamente aprovisionados y liberados con un esfuerzo administrativo y una interacción con el proveedor del servicio mínimos”.

Según el manifiesto de la nube abierta (Open Cloud Manifiesto)²⁴,” la computación en la nube es la culminación de muchas tecnologías, computación de maya (Grid Computing), utility computing, SOA, Web 2.0 y otras tecnologías”.

Las características principales de la computación de la nube es su capacidad de escalar el poder de computación de forma rentable y dinámica para el usuario final ya sea una organización o un usuario de TI.

La arquitectura de la nube puede ser privada (alojada en un servidor detrás del firewall de una empresa) o pública (alojada en Internet).

Sus características conducen a un conjunto de propuestas principales:

Escalabilidad bajo demanda: Todas las organizaciones tienen que lidiar con los cambios de sus entornos tecnológicos, la necesidad de aumentar sus recursos mientras otros se desperdician o la necesidad de un alto grado de disponibilidad en determinado instante de tiempo.

Las tecnologías de nube tiene la capacidad de enfrentar o solucionar estos cambios o necesidades de recursos por demanda tanto en entornos públicos o privados y esto permite a las personas u organizaciones pagar solo por los recursos TI que realmente necesitan.

Racionalización de los centros de datos: Las organizaciones sin importar el tamaño tienen una inversión considerable en centros de datos los cuales incluyen hardware y software en determinadas instalaciones y con la contratación de personal.

²³ National Institute of Standards and Technology, Definition Of Cloud Computing, Disponible <http://csrc.nist.gov/publications/nistpubs/800-145/SP800-145.pdf>, Consultado 11/11/2010

²⁴ [Opencloudmanifesto.org](http://www.opencloudmanifesto.org/), manifiesto de la nube libre, Disponible <http://www.opencloudmanifesto.org/>, Consultado 08/01/2011

Las tecnologías de nube permiten optimizar los centros de datos mediante la descarga de cargas de trabajo en otras nubes o en nubes públicas.

Mejora de Procesos de Negocios: La infraestructura de tecnologías de la nube mejora los procesos e negocio da las organizaciones, permitiendo a los clientes, socios y proveedores compartir los datos y aplicaciones en la nube para centrarse en el proceso del negocio en lugar de tener que preocuparse por la infraestructura propia.

Minimizar los costos de inicio: Para las empresas que están iniciando o emergentes las tecnologías en la nube les permite acceder a unos muy buenos recurso computacionales sin tener que incurrir en grandes gastos en infraestructuras TI propias.

Computación en la nube es un concepto donde se pretende utilizar internet (nube) como centro de operaciones computacionales y liberar al computador personal y a los servidores empresariales de estas responsabilidades.

La computación en la nube establece ciertas características a cumplir como son el servicio a demanda de los recursos computacionales de acuerdo a la necesidad del cliente, los recursos compartidos que están disponibles por internet, la independencia de la ubicación donde ya no importa donde se encuentre el usuario siempre tendrá disponibles los recursos informáticos, API de consumo que son aplicaciones que están presentes en la nube las cuales son usadas por demanda por los clientes, los sistemas escalables que permiten ampliar los requerimientos de las empresas de acuerdo a su crecimiento y complejidad de los procesos y el pago por uso donde de acuerdo al tiempo y recursos utilizados se establece el valor a pagar por parte del cliente.

3.1.1. Antecedentes históricos de la Computación en la nube: Desde el año 1960 el doctor John McCarthy²⁵, profesor emérito en ciencias de la computación de la Universidad de Stanford había manifestado “Algún día la computación podrá ser organizada como un servicio público”.

²⁵ Gutiérrez Juan David. Cloud Computing: una opción viable para su negocio? Video en Línea. Disponible: www.acis.org.co/fileadmin/Conferencias/ConferenciaJueves11deNoviembrede2010PrimeraParte [Citado 30 de Noviembre de 2010]

Para 1984 John Burdette Gage²⁶ de Sun Microsystems habla “The Network is the computer” haciendo alusión a que la nube (internet) es el recurso de computación a usarse en los próximos años. En el 2006 George Gilder establece el concepto de “Information factories” anunciando la muerte del computador personal y dando la bienvenida a la nube de internet como repositorio de todos los datos que los usuarios podrán usar en algún momento de su vida.

Según Mather Tim²⁷, en décadas desde 1960 hasta finales de 1970 fueron para los mainframes, equipos enormes de IBM para cómputo empresarial, entre 1980 a 1985 tuvieron su apogeo los minicomputadores de DEC con su capacidad de cómputo representativo, del año 1985 a 1990 la computación llegó a manos de personas normales con INTEL por medio del computador personal, de 1990 a 1995 SUN y MICROSOFT establecen un modelo novedoso el Cliente-Servidor como alternativa de cómputo en las organizaciones, del año 1995 al 2000 CISCO masifica las redes IP fortaleciendo las comunicaciones por internet, desde 2001 a 2008 Nokia coloca en manos de los usuarios la computación por medio de dispositivos móviles de allí que el computador deja de ser un recurso indispensable para proveer recursos de información a los clientes y finalmente después del 2008 la computación en nube se impone como alternativa con información como servicio, en este hito Salesforce, Amazon y Google se consolidan como las empresas más fuertes en proveer este tipo de servicios.

3.1.2. Clasificación de los modelos de Computación en la Nube: Según Syed A. Ahson²⁸ Existen tres diferentes modelos de computación en la nube los cuales nos permitirá tener una comprensión de la interdependencia de los principales sistemas de computación en la nube, así como mostrar su potencial y limitaciones estos tres modelos son:

- La Clasificación de la nube modelo SPI
- La Ontología de la nube de UCSB-IBM
- El modelo de nube de Hoff.

²⁶ George Gilder. Information Factories Disponible:
<http://www.wired.com/wired/archive/14.10/cloudware.html> [Citado 30 de Noviembre de 2010]

²⁷ Mather Tim Cloud. Security and Privacy Disponible:
<http://www.demosdesoftware.com/videos/864/ciberseguridad/oreilly-webcast-cloud-security-privacy/> [Citado 30 de Noviembre de 2010]

²⁸ Syed A. Ahson & Mohammad Ilyas, (CRC Press; 1 edition (July 19, 2010)) Cloud Computing and Software Services Theory and Techniques

3.1.2.1. La Clasificación de la nube modelo SPI: Según Syed A. Ahson²⁹, dentro de los sistemas que se estaban desarrollando rápidamente para la Computación de nube, se clasificaron tres subgrupos principales de los sistemas, que fueron: Software como servicio (SaaS), Plataforma como servicio (PaaS), y de Infraestructura como Servicio (IaaS).

Estos tres subgrupos fueron discutidos por varios expertos en computación en la nube y con base a estas discusiones el modelo SPI se formo y se denoto el Software, la Plataforma y la Infraestructura como sistemas de la computación en la nube.

- Sistema de Software como servicio en la nube (SaaS): Este sistema se refiere a las aplicaciones construidas por los desarrolladores para la nube y normalmente estas basadas en Un navegador y esta predefinidas su funcionalidad, acceso y predefinido su alcance, por lo general los usuarios de este sistema son los usuarios finales de las aplicaciones.

SaaS es considerado como una alternativa atractiva para las aplicaciones de escritorio de los usuarios finales por varias razones: con la aplicación desplegada por el proveedor de centro de datos se reduce los requisitos de hardware y mantenimiento de los usuarios finales, por otra parte, se simplifica el proceso de mantenimiento de software, ya que permite a los desarrolladores del software aplicar posteriores actualizaciones frecuentes y soluciones a sus aplicaciones. Ejemplos de SaaS son los clientes de Salesforce gestión de las relaciones (CRM) y Google Apps como Google Docs.

- Sistema de Plataforma como servicio en la nube (PaaS): En este segunda clase de sistema, se denominado Plataforma como servicio (PaaS), y es donde el proveedor suministra una plataforma de entornos de software e interfaces de programación de aplicaciones (API) que pueden ser utilizados en el desarrollo de aplicaciones en la nube. Naturalmente, los usuarios de esta clase de sistemas son los desarrolladores que utilizan las API específicas para crear, probar, implementar y ajustar sus aplicaciones en la plataforma de la nube.

²⁹ Syed A. Ahson & Mohammad Ilyas, (CRC Press; 1 edition (July 19, 2010)) Cloud Computing and Software Services Theory and Techniques

El desarrollo de aplicaciones para una plataforma en la nube es analógico en cierta medida al desarrollo de una aplicación para la web, el viejo modelo de servidores, en el sentido de que los desarrolladores escriben códigos y lo implementan en un servidor remoto, para los usuarios finales, el resultado final es una aplicación basada en navegador.

Sin embargo, el modelo PaaS es diferente, ya que puede proporcionar servicios adicionales para simplificar el desarrollo de aplicaciones, implementación y ejecución, tales como capacidad de ampliación automática, monitoreo y balanceo de carga. Por otra parte, los desarrolladores de aplicaciones pueden integrar otros servicios prestados por el sistema de PaaS a su aplicación, tales como los servicios de autenticación, servicios de correo electrónico, y componentes de la interfaz de usuario, todo lo que se proporciona a través de un conjunto de API es suministrada por la plataforma. Como resultado, el sistema PaaS se considera generalmente para acelerar el desarrollo software y el tiempo de implementación. Un ejemplo de los sistemas de esta categoría es Google App Engine, que proporciona Python y entornos de ejecución Java y API para aplicaciones para interactuar con el entorno de ejecución de Google.

- Sistema de Infraestructura como servicio en la nube (IaaS): El tercer sistema, de acuerdo con el modelo de clasificación de SPI, es el que proporciona los recursos de infraestructura tales como computación, almacenamiento y servicios de comunicación en una manera flexible. Estos sistemas se denominan de Infraestructura como Servicio (IaaS).

Los avances recientes en el sistema operativo (OS) de virtualización han facilitado la aplicación de IaaS en el hardware existente.

En este sentido utilizar tecnología de virtualización permite un nivel de direccionamiento indirecto con respecto a la directa del uso del hardware, permite el uso del equipo directamente a ser encapsulado y aislado en el contenedor de la instancia de una máquina virtual (VM).

Por lo tanto, el sistema operativo de virtualizado permite a los proveedores de IaaS controlar y gestionar la utilización eficiente de los recursos físicos.

Este enfoque es particularmente atractivo para los proveedores de IaaS dada la subutilización de los sistemas de energía, los procesadores de alta velocidad que constituyen la infraestructura de los centros de datos.

Amazon infraestructura EC2³⁰, es un ejemplo de los sistemas de IaaS, donde los usuarios pueden alquilar potencia de cálculo en su infraestructura por hora. En este espacio, también hay varios proyectos académicos de código abierto en la nube.

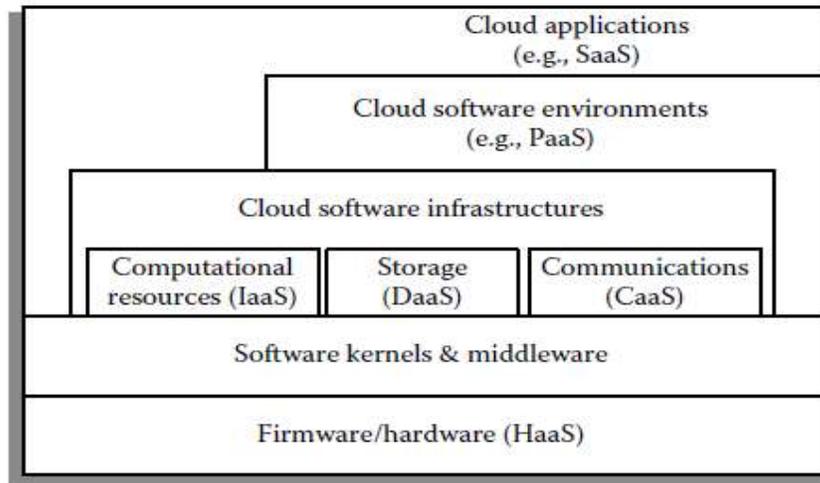
3.1.2.2. La Ontología de la nube de UCSB-IBM: La ontología de la nube de UCSB-IBM surgió a través de un esfuerzo de colaboración entre la Universidad de California en Santa Bárbara y la industria (IBM Centro de Investigación TJ Watson) en un intento de entender el panorama de la computación en la nube. El objetivo final de este esfuerzo era el de facilitar la exploración del área de la computación en la nube, así como para avanzar en los esfuerzos educativos en la enseñanza y la adopción de la nube en el área de computación. En esta clasificación, los autores utilizaron el principio de compatibilidad de la Arquitectura Orientada a Servicios (SOA) para clasificar las distintas capas de la nube.

Compatibilidad en SOA es la capacidad para coordinar y reunir un conjunto de servicios, servicios para formar compuestos. En este sentido, los servicios en la nube también pueden ser integrados por uno o más servicios de otras nubes. Por el principio de compatibilidad, el modelo de UCSB-IBM clasifica la nube en cinco capas; cada capa comprende uno o más servicios en la nube, servicios en la nube que pertenecen a la misma capa si tienen un nivel de abstracción.

El modelo de UCSB- IBM de computación en la nube se muestra en la Figura 1, las tres primeras capas de la nube de UCSB-IBM son similares a la clasificación de SPI, salvo que los autores rompen la capa de la infraestructura en tres componentes, los tres componentes que componen la capa de infraestructura de UCSB -IBM son recursos computacional, almacenamiento y comunicaciones.

³⁰ Syed A. Ahson & Mohammad Ilyas, (CRC Press; 1 edition (July 19, 2010)) Cloud Computing and Software Services Theory and Techniques

Figura 1, “Modelo de Ontología de la nube de UCSB-IBM, con tres componentes de la capa de infraestructura en la nube”³¹



Fuente: Syed A. Ahson & Mohammad Ilyas, (CRC Press; 1 edition (July 19, 2010)) Cloud Computing and Software Services Theory and Techniques, pp. 5.

3.1.2.3. Modelo de nube de Hoff: El modelo de nube de Hoff inspirado por el modelo de SPI y la ontología de nube de UCSB-IBM, quien organizo una discusión en línea entre varios expertos en computación en la nube quienes construyeron una ontología sobre los modelos anteriores.

El modelo de Hoff, se muestra en Figura 2, presenta una ontología de nube de nuevo con más detalle. Este modelo centrado en el análisis de los tres principales servicios de la nube: IaaS, PaaS y SaaS. El modelo analiza la capa IaaS con varios componentes, datos, centro de instalaciones, que incluyen la energía y el espacio.

El hardware es el segundo componente de la capa de IaaS, que consiste en el nodo de cómputo, los datos, el almacenamiento y la red. La abstracción es el siguiente componente, que la componen el hardware a través de sistemas de máquinas virtuales, la red y los servicios de Grid. El siguiente componente es la conectividad básica y la entrega, que proporciona los diversos servicios de apoyo

³¹ Syed A. Ahson & Mohammad Ilyas, (CRC Press; 1 edition (July 19, 2010)) Cloud Computing and Software Services Theory and Techniques

de los sistemas de utilización de la capa de IaaS, como la autenticación y servicios de DNS.

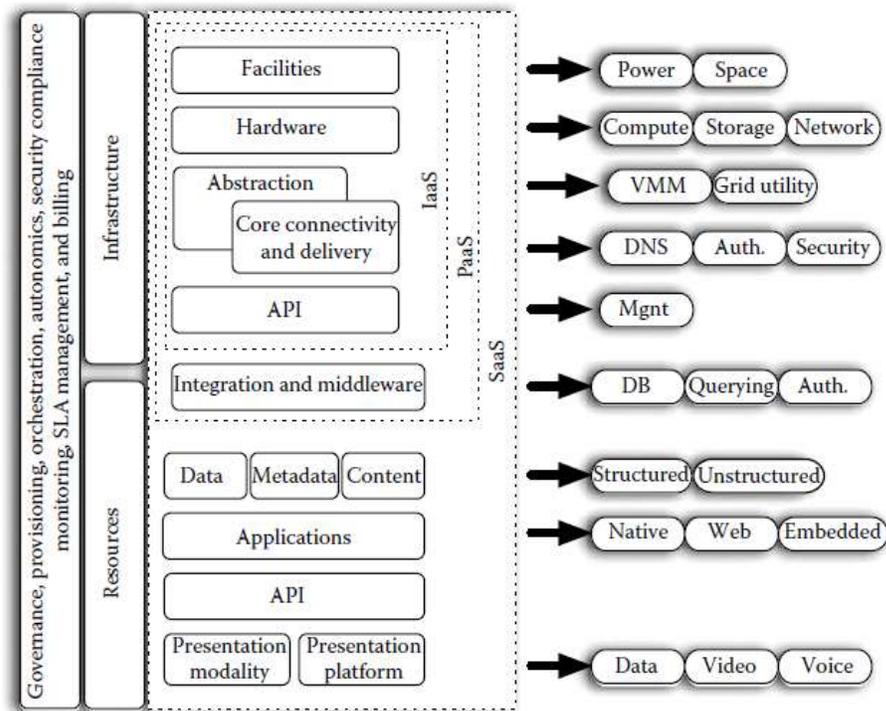
El componente API presenta la gestión de servicios, así como una interfaz simplificada para la capa siguiente en la nube, esta capa siguiente en el modelo de Hoff, que es el PaaS, se compone de una sub-capa que proporciona los servicios de integración en la nube, esta sub-capa proporciona varios servicios, tales como la autenticación, la base de datos, y servicios de consulta. La capa de SaaS en el modelo de Hoff, también se subdivide en varias subcapas y componentes.

El componente de aplicación en el SaaS capa se clasifican en tres categorías: aplicaciones nativas, aplicaciones web, y aplicaciones integradas, una aplicación nativa puede ser una aplicación de escritorio que utiliza un servicio en la nube, una aplicación Web es una aplicación de nube que se accede a través del navegador web. Por último, una aplicación embebida es una aplicación de nube que está incrustado en otra aplicación.

Las últimas dos sub-capas en la capa de SaaS en el modelo de Hoff son API de las aplicaciones y la presentación de sub-capas. En la sub-capas de presentación se compone de la presentación de datos, presentación de video, voz y presentación, reconociendo las diferentes formas de presentaciones de datos de nubes.

Figura 2, “Modelo de la Nube de Hoff, Resultado de colaboración en línea y la discusión entre los diferentes expertos en la computación en nube”³²

³² Syed A. Ahson & Mohammad Ilyas, (CRC Press; 1 edition (July 19, 2010)) Cloud Computing and Software Services Theory and Techniques



Fuente: Syed A. Ahson & Mohammad Ilyas, (CRC Press; 1 edition (July 19, 2010)) Cloud Computing and Software Services Theory and Techniques, pp. 12.

3.1.3. Tipos de Computación en la Nube: El Instituto Nacional de Estándares y Tecnología de los Estados Unidos de América (NIST, 2009)³³ define cuatro modelos de despliegue. Estos pueden ser considerados como la estructura básica de los modelos de negocios de Computación en la nube:

- Nubes públicas: La infraestructura de la Nube está disponible para el público en general o puede estar orientado a una industria en particular. Esta infraestructura es propiedad de la organización que ofrece los servicios en la nube.
- Nubes privadas: En el modelo de Computación de nube privado, la infraestructura está orientada a servir a una sola organización. Su gestión puede ser realizada por la organización o por un tercero.
- Nubes comunitarias: La infraestructura es compartida por diferentes organizaciones y está orientada a servir a un tipo de comunidad específica de

³³ National Institute of Standards and Technology, Definition Of Cloud Computing, Disponible <http://www.opencloudmanifesto.org/>, Consultado 11/11/2010

acuerdo a sus intereses comunes. Su gestión puede ser realizada por las organizaciones o por un tercero.

- Nubes híbridas: Este modelo es la unión del modelo público y el modelo privado. Bajo este modelo dos o más nubes (privadas, públicas o comunitarias) se unen a través de tecnología estándar o propietaria que facilite la portabilidad de datos y aplicaciones, por ejemplo, en caso de *cloud bursting*, donde se balancea la carga entre las nubes cuando una de ellas ha ocupado su capacidad máxima.

3.1.4. Proveedores actuales de Computación en la nube: Según Castro³⁴ en la actualidad los principales proveedores de Computación en la nube son:

- Amazon EC (Elastic Compute Cloud), es un servicio Web que provee capacidades de cómputo elásticas, disponibles a través de una infraestructura de nube diseñada con la finalidad de proveer computación escalable a entornos Web, bajo demanda, siguiendo un modelo comercial de pago por uso. (Amazon Web Services, LLC).

-Google App Engine es una plataforma para la creación y alojamiento de aplicaciones web utilizando la infraestructura de Google. El servicio se encuentra actualmente en prueba, permitiendo a los desarrolladores inscribirse gratuitamente y utilizar hasta 500 MB de almacenamiento.

-Salesforce.com es un proveedor de Customer Relationship Management (CRM), que ofrece la utilización del software como servicio, que incluyen aplicaciones para ventas, servicio, soporte y comercialización.

-Microsoft con la Plataforma de Servicios Azure es una plataforma de servicios Cloud, alojados en centros de datos de Microsoft, que ofrece un sistema operativo y un conjunto de servicios de desarrollo que pueden ser utilizados individualmente o en conjunto.

3.1.5. Software libre para la implementación de Computación en la nube: El software libre cumple un papel importante dentro del desarrollo e implementación de tecnologías de Computación en la nube, gracias a este se debe el crecimiento

³⁴ Castro Jorge. Cloud Computing una perspectiva para Colombia. Disponible: www.interactic.com.co/dmdocuments/clud_computing.pdf [Citado 2 de Diciembre de 2010]

de grandes iniciativas de software como lo son GNU/Linux, Open Office, Mysql, Mozilla Firefox, entre otras.

Gracias al uso de herramientas de software libre es el desarrollo e implantación de la infraestructura de Computación en la nube, pero surge un debate debido a esto ¿que tan libre es la nube?, ya sea pública o privada independiente de la infraestructura que se utilice, el modelo que se ofrezca como servicio la computación de la nube debe cumplir unos principios para que sea y se mantenga libre, por esta razón varias empresas entre las que destacan IBM, Sun Microsystems, Cisco, Novell, Red Hat, EMC y otras más tomaron la iniciativa de readaptar un documento como manifiesto en el cual se definen lo que serán las reglas de juego de las iniciativas de Computación en la nube que estamos ya viendo y que se generalizarán como el nuevo esquema computacional.

El manifiesto de la nube abierta (open cloud manifesto)³⁵, consta de seis principios que son:

1. Trabajar juntos para que los retos fundamentales en la adopción sean solucionados mediante colaboración abierta y el uso adecuado de los estándares.
2. No utilizar su posición de mercado para convertir a sus clientes en cautivos de una plataforma concreta y limitar su libertad de elección.
3. Usar y adoptar los estándares existentes siempre que sea posible, para evitar así reinventarlos o duplicarlos.
4. Recurrir con prudencia a la creación de nuevos estándares, y cuando así sea por necesidad, hacerlo con pragmatismo, reduciendo el número de estándares necesarios, y asegurando que éstos promueven la innovación en lugar de inhibirla.
5. Llevar a cabo iniciativas en función de las necesidades del cliente, no de las necesidades técnicas de los proveedores.
6. Trabajo conjunto y coordinado de todos los actores implicados para evitar que sus iniciativas entren en conflicto o se solapen³⁶.

Siguiendo estos principios existen múltiples iniciativas de organizaciones y empresas las cuales son fieles a la filosofía del Software libre y ofrecen sus

³⁵ Openccloudmanifesto.org, manifiesto de la nube libre, Disponible <http://www.openccloudmanifesto.org/openccloudmanifesto5.htm> [Citado 11 de Noviembre de 2010]

³⁶ Openccloudmanifesto.org, manifiesto de la nube libre, Disponible <http://www.openccloudmanifesto.org/openccloudmanifesto5.htm> [Citado 11 de Noviembre de 2010]

productos y desarrollos a la comunidad para su estudio mejora y utilización del software en la Computación en la nube, entre estos se destacan:

- **OpenNebula:** Según OpenNebula³⁷ es una herramienta totalmente de código abierto para construir cualquier tipo de nube de IaaS: privado, público e híbridos. La tecnología OpenNebula es el resultado de muchos años de investigación y desarrollo en la gestión eficiente y escalable de máquinas virtuales en gran escala infraestructuras distribuidas. Sus características innovadoras han sido desarrolladas para satisfacer los requisitos de casos de uso empresarial de las principales empresas de TI y de múltiples sectores en el contexto de proyectos pioneros en Computación en la nube.

OpenNebula está siendo utilizado como una plataforma abierta para la innovación en varios proyectos internacionales de investigación para los desafíos que se plantean en la gestión empresarial de nubes, y también como herramienta de producción, lista en la academia y la industria para administrar las nubes.

- **Eucalyptus:** Según Eucalyptus³⁸ es un software que implementa nubes privadas e híbridos el estilo de IaaS. La arquitectura de Eucalyptus es altamente modular con componentes internos consistentes en servicios Web, lo que las hace fáciles de reemplazar y ampliar. La flexibilidad de Eucalyptus " permite exportar una gran variedad de APIs a los usuarios a través de las herramientas de cliente. Actualmente Eucalyptus implementa el servicio web de Amazon (AWS) y su API, que permite la interoperabilidad con los servicios de AWS-compatibles y herramientas. Esto permite a los usuarios utilizar los recursos establecidos de Eucalyptus, desde una nube privada interna y nubes públicas externas para formar una nube híbrida.

- **Ubuntu Enterprise Cloud:** Según Simon Wardley³⁹ en la actualidad Ubuntu es una de las distribuciones de GNU/Linux más usadas gracias a su simplicidad y al soporte ofrecido por la empresa Canonical Ltd.

³⁷ OpenNebula.org, Acerca de la tecnología de OpenNebula Disponible <http://www.opennebula.org/about:technology> [Citado 10 de Diciembre de 2010]

³⁸ Eucalyptus System Inc. Sobre la plataforma de código abierto de nube, Disponible <http://open.eucalyptus.com/>, [Citado 10 de Diciembre de 2010]

³⁹ Simon Wardley, Etienne Goyer & Nick Barcet Ubuntu Enterprise Cloud Architecture, Agosto 2009, Disponible, <http://www.canonical.com/about-canonical/resources/white-papers/ubuntu-enterprise-cloud-architecture> [Citado 10 de Enero de 2011]

Dentro de su portafolio de productos Canonical ofrece Ubuntu Enterprise Cloud, el cual permite la implementación de la infraestructura de computación en la nube ya sea pública o privada con todo el potencial de Amazon Elastic Computing Cloud (EC2) al interior de la organización mediante el uso de Eucalyptus una solución de código abierto con los estándares y API de EC2.

Ubuntu Enterprise Cloud (UEC), permite desplegar todo el potencial de computación en la nube al interior de una organización de cualquier tamaño gracias al uso de aplicaciones de código abierto sin restricción de licencias o suscripción por uso.

3.1.6. Impacto de computación en la nube en organizaciones: Para Castro Jorge⁴⁰ New York Times, como se detalla en (Gottfrid., 2007), el periódico The New York Times necesitaba convertir 11 millones de artículos e imágenes desde su archivo (de 1851 a 1980) al formato PDF. Su departamento interno de TI dijo que les tomaría siete semanas.

Mientras tanto, un desarrollador utilizando 100 instancias de Amazon EC2 con una interface de servicios Web simple para operar Hadoop (una implementación de código abierto similar al MapReduce) completó el trabajo en 24 horas por menos de \$300.

La empresa Netflix (www.netflix.com) dedicada al servicio de streaming de películas y series de televisión cuenta con 15 millones de suscriptores y acoge el 20% de descarga en los Estados Unidos desarrolló la migración de su plataforma tecnológica en Cloud Computing excepto la pasarela de pagos.

La empresa Vivareal (www.vivareal.com.co) pudo probar sus aplicaciones sin realizar inversión en infraestructura, esta empresa desarrollo pruebas de carga de sus bases de datos NoSQL con un costo total de 15 dólares por medio de Computación en la nube.

El Grupo Manuelita por medio de Microsoft Business Productivity Online Suite pudo desarrollar operaciones de migración de la compañía ahorrando entre un 30 y 40% de presupuesto de gastos en tecnología, garantizando la movilidad y portabilidad de las aplicaciones.

El Servicio Nacional de Aprendizaje SENA por medio de Google Apps permitió el uso de su plataforma educativa a 3,4 millones de personas en el país con

⁴⁰ Castro Jorge. Cloud Computing una perspectiva para Colombia. Disponible: www.interactic.com.co/dmdocuments/clud_computing.pdf [Citado 2 de Diciembre de 2010]

funcionalidades como mensajería instantánea, agenda y calendario, procesamiento de textos y hojas de cálculo.

3.2. EL SOFTWARE LIBRE

El Software Libre viene ocupando un papel muy importante en el mundo de la tecnologías de la información y las comunicaciones, su participación cada vez es más grande gracias a la adopción de normas y estándares abiertos por iniciativa y organización de la comunidad conformada por (científicos, académicos, profesionales y entusiastas), con el apoyo y soporte de la industria y entes gubernamentales, principalmente en los países desarrollados.

Pero en la actualidad aun existe cierta ambigüedad cuando se habla de software libre, es más que un tipo de licenciamiento o beneficios económicos, el software libre es cuestión de la libertad de los usuarios de ejecutar, copiar, distribuir, estudiar, cambiar y mejorar el software, pero para entender mejor el concepto de Software Libre se cita la definición de la Fundación de Software Libre (FSF)⁴¹.

El «software libre» es una cuestión de libertad, no de precio. Para entender el concepto, debería pensar en «libre» como en «libre expresión», no como en «barra libre».

El software libre es una cuestión de la libertad de los usuarios de ejecutar, copiar, distribuir, estudiar, cambiar y mejorar el software. Más precisamente, significa que los usuarios de programas tienen las cuatro libertades esenciales.

- La libertad de ejecutar el programa, para cualquier propósito (libertad 0).
- La libertad de estudiar cómo trabaja el programa, y cambiarlo para que haga lo que usted quiera (libertad 1). El acceso al código fuente es una condición necesaria para ello.
- La libertad de redistribuir copias para que pueda ayudar al prójimo (libertad 2).
- La libertad de distribuir copias de sus versiones modificadas a terceros (libertad 3). Si lo hace, puede dar a toda la comunidad una oportunidad de beneficiarse de sus cambios. El acceso al código fuente es una condición necesaria para ello.

Un programa es software libre si los usuarios tienen todas esas libertades. Entonces, debería ser libre de redistribuir copias, tanto con o sin modificaciones, ya sea gratis o cobrando una tarifa por distribución, a cualquiera en cualquier

⁴¹ FREE SOFTWARE FOUNDATION en español, sitio web [En línea], [Diciembre2011]. Disponible en: <http://www.gnu.org/home.es.html>

parte. El ser libre de hacer estas cosas significa, entre otras cosas, que no tiene que pedir o pagar el permiso.

También debería tener la libertad de hacer modificaciones y usarlas en privado, en su propio trabajo u obra, sin siquiera mencionar que existen. Si publica sus cambios, no debería estar obligado a notificarlo a alguien en particular, o de alguna forma en particular.

La libertad de ejecutar el programa significa la libertad para cualquier tipo de persona u organización de usarlo en cualquier tipo de sistema de computación, para cualquier tipo de trabajo y propósito, sin estar obligado a comunicarlo a su programador, o alguna otra entidad específica. En esta libertad, el propósito de los usuarios es el que importa, no el propósito de los programadores. Como usuario es libre de ejecutar un programa para sus propósitos; y si lo distribuye a otra persona, también es libre para ejecutarlo para sus propósitos, pero usted no tiene derecho a imponerle sus propios propósitos.

La libertad de redistribuir copias debe incluir las formas binarias o ejecutables del programa, así como el código fuente; tanto para las versiones modificadas como para las no lo están. (Distribuir programas en forma de ejecutables es necesario para que los sistemas operativos libres se puedan instalar fácilmente). Resulta aceptable si no existe un modo de producir un formato binario o ejecutable para un programa específico, dado que algunos lenguajes no incorporan esa característica, pero debe tener la libertad de redistribuir dichos formatos si encontrara o programara una forma de hacerlo.

Para que la libertad 1 y 3 para realizar cambios y publicar versiones mejoradas, tengan sentido; debe tener acceso al código fuente del programa.

Por consiguiente, el acceso al código fuente es una condición necesaria para el software libre. El «código fuente» ofuscado no es código fuente real, y no cuenta como código fuente.

La libertad 1 incluye la libertad de usar su versión modificada en lugar de la original. Si el programa se entrega con un producto diseñado para ejecutar versiones modificadas de terceros, pero rechaza ejecutar las suyas, una práctica conocida como «tivoization» o «lockdown» o «arranque seguro» (en la terminología perversa de los que la practican); la libertad 1 se convierte más en una ficción teórica que en una libertad práctica. Esto no es suficiente. En otras palabras, estos binarios no son software libre, incluso si se compilaron desde un código fuente que es libre.

Una manera importante de modificar un programa es fusionando subrutinas y módulos libres disponibles. Si la licencia del programa dice que no puede fusionar un módulo existente con una debida licencia, así como si le requiere ser el titular de los derechos de autor de lo que agregue, entonces la licencia es demasiado restrictiva para calificarla como libre.

La libertad 3 incluye la libertad de liberar sus versiones modificadas como software libre. Una licencia también puede permitir otras formas de relicenciarlas, en otras palabras, no tiene que ser una licencia de copyleft. No obstante, una licencia que requiera que las versiones modificadas no sean libres, no se puede considerar como una licencia libre.

Para que estas libertades puedan ser reales, deben ser irrevocables siempre que usted no cometa ninguna equivocación; si el programador del software tiene el poder de revocar la licencia, o de cambiar retroactivamente sus términos, sin que usted se haya equivocado para justificarlo, el software no es libre.

Sin embargo, ciertos tipos de reglas sobre la manera de distribuir software libre son aceptables, cuando no entran en conflicto con las libertades principales. Por ejemplo, el copyleft (definido muy resumidamente) es la regla en base a la cual, cuando redistribuye el programa, no puede agregar restricciones para denegar a las demás personas las libertades principales. Esta regla no entra en conflicto con las libertades principales; más bien las protege.

«Software libre» no significa «que no sea comercial». Un programa libre debe estar disponible para el uso comercial, la programación comercial y la distribución comercial. La programación comercial de software libre ya no es inusual; tal software libre comercial es muy importante. Puede haber pagado dinero para obtener copias de software libre, o puede haber obtenido copias sin costo. Pero sin tener en cuenta cómo obtuvo sus copias, siempre tiene la libertad de copiar y modificar el software, incluso de vender copias.

Si una modificación constituye una mejora es un asunto subjetivo. Si sus modificaciones se limitan, en esencia, a los cambios que otra persona considera una mejora, eso no se trata de libertad.

No obstante, las reglas acerca cómo empaquetar una versión modificada son aceptables si no limitan substancialmente su libertad para publicar versiones modificadas, o su libertad para hacer y usar versiones modificadas en privado. Así que es aceptable que una licencia le obligue a cambiar el nombre de la versión modificada, eliminar el logotipo o a identificar sus modificaciones como suyas. Son aceptables siempre y cuando esas obligaciones no sean tan agobiantes que le dificulten la publicación de sus modificaciones. Como ya está aplicando otras modificaciones al programa, no le supondrá un problema hacer algunas más.

Las normas del estilo «si pone a disposición su versión de este modo, también debe hacerlo de este otro modo» también pueden ser, bajo la misma condición, admisibles. Un ejemplo de una norma admisible, sería una que planteara que si ha distribuido una versión modificada, y uno de los programadores de versiones anteriores le pide una copia, deberá mandarle una (tenga en cuenta que esta norma le sigue permitiendo elegir si distribuye, o no, su versión.). Las normas que obligan a liberar el código fuente a los usuarios de las versiones que publica también son admisibles.

En el proyecto GNU, usamos copyleft para proteger legalmente estas libertades para todos. Pero también existe software libre sin copyleft. Creemos que existen razones importantes por las que es mejor usar copyleft, pero si su programa es software libre sin copyleft, sigue siendo ético de todos modos.

En algunos casos las regulaciones de control de exportación y las sanciones comerciales pueden limitar sus libertades de distribuir copias de programas intencionalmente. Los desarrolladores de software no tienen el poder de eliminar o pasar por alto estas restricciones, pero lo que pueden y deben hacer es rechazar imponerlas como condiciones para el uso del programa. De este modo, las restricciones no afectarán a las actividades ni a las personas fuera de las jurisdicciones de dichos gobiernos. Por ende, las licencias de software libre no deben requerir la obediencia a ninguna regulación de exportaciones como condición de cualquiera de las libertades esenciales.

La mayoría de las licencias de software libre están basadas en el copyright, y existen límites en los tipos de requisitos que pueden ser impuestos a través del copyright. Si una licencia basada en el copyright respeta la libertad en las formas antes mencionadas, es poco probable tener otro tipo de problema que no hayamos anticipado (a pesar de que esto ocurre ocasionalmente). Sin embargo, algunas licencias de software libre están basadas en contratos, y los contratos pueden imponer un rango mucho más grande de restricciones posibles. Esto significa que existen muchas maneras posibles de que tal licencia pueda ser inaceptablemente restrictiva y que no sea libre.

Posiblemente no podamos enumerar todas las formas en las que eso puede pasar. Si una licencia basada en un contrato restringe al usuario de un modo que no puedan hacer las licencias basadas en el copyright, y que no está mencionado aquí como legítimo, tendremos que pensar sobre ello; y probablemente concluyamos que no es libre.

Cuando se habla de software libre, es mejor evitar usar términos como «regalar» o «gratis», porque dichos términos implican que el asunto pasa por el precio, no la libertad. Algunos términos comunes como «piratería» implican opiniones con las que esperamos no concuerde.

Finalmente, tenga en cuenta que los criterios, como los establecidos en esta definición de software libre, requieren pensar con cuidado su interpretación. Para decidir si una licencia de software específica es una licencia de software libre, la juzgamos en base a estos criterios para determinar si concuerda su espíritu, conjuntamente con la terminología precisa. Si una licencia incluye restricciones demasiado grandes, la rechazamos, incluso si no anticipamos la cuestión en este criterio. Algunas veces, los requisitos de una licencia muestra una cuestión que hace necesaria una reflexión más profunda, incluyendo la discusión con un abogado, antes que podamos decidir si el requisito es aceptable. Cuando llegamos

a una conclusión sobre una nueva cuestión, solemos actualizar estos criterios para que resulte más fácil ver por qué ciertas licencias se califican o no.⁴²

3.2.1. Historia del Software Libre: Los comienzos del software libre se remonta hacia los años sesenta donde los computadores como mainframes o minicomputadores se distribuían con el software y específicamente con el código fuente, tal es el caso de las maquinas que fabricaba IBM, para la década de los setenta el software ya tomó un valor y se vendía por separado, lo que definió una distribución privativa que no entrega el código fuente donde los permisos de modificación y redistribución estaban vetados.

Un evento significativo de 1970 lo marca la creación del sistema operativo Unix en los laboratorios Bell de AT&T, que originó posteriormente Unix BSD, conjuntamente con el nacimiento de internet se establece el campo de pruebas para novedosas formas de colaboración las cuales se perpetuaron posteriormente en el software libre.

Para 1984 Richard Stallman comenzó a trabajar en el proyecto GNU y funda la Free Software Foundation (FSF), escribe la licencia GPL y propone los fundamentos del software libre tal como lo conocemos hoy.

Para la década de 1990 con el Internet ya evolucionado se proporciona a las comunidades de software libre nuevos canales de comunicación y distribución.

En 1991 Linus Torvalds comenzó a desarrollar un núcleo libre (Linux) que permitió completar el sistema GNU, que contaba ya con casi todas las partes para convertirse en un sistema completo similar a Unix con compilador de lenguaje c, editor de texto Emacs, sistema de ventanas X-windows. Nacieron los sistemas operativos GNU/Linux, que fructificaron en variedad de distribuciones, como Red Hat Linux y Debian GNU/Linux, para el final de 1990 estos sistemas se complementan con los entornos de escritorio KDE y GNOME.

Después del año 2000 el software libre llega a liderar algunos sectores como los Servidores web específicamente por Apache y aparecen nuevas herramientas que cubren variedad de aplicaciones informáticas.⁴³

⁴² FREE SOFTWARE FOUNDATION en español, sitio web [En línea], [Diciembre2011]. Disponible en: <http://www.gnu.org/home.es.html>

⁴³ FREE SOFTWARE FOUNDATION en español, sitio web [En línea], [Diciembre2011]. Disponible en: <http://www.gnu.org/home.es.html>

3.2.2. Tipos de Licencias en el software libre: En el aspecto legal la situación de los programas libres respecto de los privativos no es diferente, los dos se distribuyen bajo licencia, la diferencia está en lo que permite la licencia. En las licencias de los programas libres, la redistribución y la modificación tiene ciertas condiciones para esta práctica.

La variedad de licencias libres es amplia sin embargo se pueden agrupar en Cuatro o cinco tipos de ellas. Es posible dividir las licencias de software libre en Dos grandes familias. La primera compuesta por las licencias que no imponen Condiciones especiales en la segunda redistribución, esto implica que sólo Especifican que el software se puede redistribuir o modificar pero que no imponen condiciones especiales si se hace, lo que permite por ejemplo, que alguien que reciba el programa pueda después redistribuirlo como software propietario estas licencias se denominan permisivas. La segunda familia, que se denominan licencias robustas o licencias de copyleft incluyen las que al estilo de la GNU GPL, imponen condiciones en el caso de que se requiera redistribuir el software, condiciones que van en la línea de forzar que se sigan cumpliendo las condiciones de la licencia después de la primera redistribución. Mientras que el primer grupo hace énfasis en la libertad de quien recibe un programa, que le permite hacer casi todo lo que quiera con él, el segundo hace énfasis en la libertad de cualquiera que potencialmente pueda recibir algún día un trabajo derivado del programa, que obliga a que las sucesivas modificaciones y redistribuciones respeten los términos de la licencia original.⁴⁴

⁴⁴ FREE SOFTWARE FOUNDATION en español, sitio web [En línea], [Diciembre2011]. Disponible en: <http://www.gnu.org/home.es.html>

4. EVOLUCIÓN DE GRID COMPUTING A CLOUD COMPUTING

En los últimos años dentro de la industria y las comunidades científicas de las tecnologías de la información y las comunicaciones se ha venido posesionado con fuerza un nuevo termino informático, la computación en la nube o (Cloud Computing en ingles), pero realmente que es la computación en la nube (Cloud Computing)? y qué relación tiene con otras tecnologías como Grid Computing, cuáles son sus orígenes o por lo contrario es una nueva tecnología que va a marcar el futuro de la informática moderna, lo cierto es que no es un concepto nuevo, en 1961, el pionero informático John McCarthy⁴⁵ predijo que "algún día la Computación puede ser organizado como un servicio público" y llegó a especular cómo esto podría ocurrir, pero para lograr materializar esta predicción fue necesario que pasaran décadas y cambios de paradigmas que llevaron al desarrollo de nuevas y mejores tecnologías cada vez mas robustas y complejas para poder suplir las necesidades que la industria demandaba.

La necesidad de realizar grandes cálculos y de procesar cada vez mayores volúmenes de datos demandaba el uso de supercomputadores o computación de altas prestaciones (HPC por sus siglas en ingles) que fueran capaces de soportar trabajos complejos, pero debido a las características de hardware, sus costos eran muy elevados. Gracias al desarrollo acelerado de las tecnologías de la información y las comunicaciones con la creación de computadores personales con hardware robusto a bajo costo, la masificación de Internet debido al mejoramiento y disminución de costos de las comunicaciones, permito a los científicos llevar estas ideas de la teoría a la realidad con la aparición de sistemas distribuido basado en modelos tecnológicos y comerciales.

Para comprender mejor los conceptos de las nuevas y mejores formas de computación es necesario profundizar y entender cada una de ellas.

4.1. GRID COMPUTING O COMPUTACIÓN DE MALLA.

⁴⁵ Ian Foster, Yong Zhao, Ioan Raicu Cloud Computing and Grid Computing 360-Degree Compared, IEEE 06 enero 2009 pdf.
http://ieeexplore.ieee.org/xpl/freeabs_all.jsp?tp=&isnumber=4738437&arnumber=4738445

En la actualidad existen una variedad de definiciones desde varios puntos de vista, de la industria, académicos y analistas los cuales generan cierta confusión a la hora de presentar una única definición, pero teniendo en cuenta varios puntos de vista podemos definir a Grid Computing o Computación de malla como el resultado de la integración de múltiples tecnologías las cuales han ido evolucionando desde el mismo inicio de los sistemas informáticos los cuales no han parado de crecer, debido a las necesidades de mejorar todas estas tecnologías para poder obtener mayor poder de computo logrando compartir recursos de hardware dentro de una infraestructura tecnológica local o varias de forma remota aprovechando el despliegue de recursos de interconexión de Internet, logrando reducir los costos de almacenamiento y calculo de procesamiento de datos mediante la integración de software de coordinación de recursos descentralizados logrando superar las fronteras de las organizaciones por medio del uso de protocolos y estándares abiertos.

Según Katarina Stanoevska⁴⁶ Grid Computing o computación en malla surge gracias el resultado de estudios e investigaciones realizadas por la comunidad científica durante la década de los 90 donde la Computación de altas prestaciones (HPC por sus siglas en ingles), Meta Computing e Hyper Computing era la solución a la demanda de cálculo y procesamiento de datos por medio de conexiones de datos rápidas haciendo hincapié en la coordinación de recursos informáticos disponibles para aplicaciones de alto rendimiento, pero el alto costo que demandaban este tipo de computación permitió la evolución de nuevas tecnologías tomando como base el modelo del computo distribuido.

Inspirados por la penetración de la red de energía eléctrica, la facilidad de uso y fiabilidad, científicos de computación a mediados de 1990 comenzaron a explorar el diseño y desarrollo de una infraestructura similar llama la red de energía computacional que permite el acceso a la potencia de cálculo y aplicación en cualquier momento o lugar, según sea necesario sin la necesidad de poseer la infraestructura necesaria para producir el servicio.

Para entrar más en detalle vamos a nombrar algunas de las definiciones más importantes desde varios puntos de vista acerca de Grid Computing.

⁴⁶ Katarina Stanoevska Thomas Wozniak, Santi Ristol "Grid and Cloud Computing A Business Perspective on Technology and Applications" Springer Dec 2010.

Una de las primeras definiciones más citadas de Grid Computing que refleja sus Orígenes que fue sugerida por Foster y Kesselman⁴⁷:

"Grid Computing es una infraestructura de hardware y software que proporciona confiabilidad, consistencia, penetración y acceso a bajo costo a computación de alto nivel. "

Según Foster⁴⁸, un sistema de Grid Computing es por lo tanto, un sistema que:

- Coordina los recursos que no están sujetos a un control centralizado.
- Utiliza estándares abiertos de uso general, protocolos e interfaces.
- Ofrece una calidad de servicio no trivial.

Según Lilienthal⁴⁹, los principales recursos que se pueden compartir en Grid Computing son:

- Informática / potencia de procesamiento
- Almacenamiento de datos / sistemas en red de archivos
- Comunicaciones y ancho de banda
- Aplicaciones de Software.
- Instrumentos científicos.

Una de las definiciones más nuevas y precisa fue tomada por la comunidad Científica

"Grid Computing es considerado por la comunidad científica como una capa de middleware que permite el intercambio seguro, confiable y eficiente de los recursos de computo y datos entre entidades independientes de la organización."⁵⁰

Otra definición y enfoque de Grid Computing fue aportado por la industria, con interpretaciones diferentes, IBM por ejemplo, describe Grid Computing indirectamente, al referirse a sus características:

"La computación Grid permite unir grupos de servidores, sistemas de almacenamiento y las redes en un gran sistema único, para poder ofrecer la

⁴⁷ Katarina Stanoevska Thomas Wozniak, Santi Ristol "Grid and Cloud Computing A Business Perspective on Technology and Applications" Springer Dec 2010, pp. 23

⁴⁸ Katarina Stanoevska Thomas Wozniak, Santi Ristol "Grid and Cloud Computing A Business Perspective on Technology and Applications" Springer Dec 2010, pp. 24

⁴⁹ Katarina Stanoevska Thomas Wozniak, Santi Ristol "Grid and Cloud Computing A Business Perspective on Technology and Applications" Springer Dec 2010, pp. 24

⁵⁰ Katarina Stanoevska Thomas Wozniak, Santi Ristol "Grid and Cloud Computing A Business Perspective on Technology and Applications" Springer Dec 2010, pp. 24

potencia de recursos de múltiples sistemas en un solo punto a un usuario para un propósito específico.”⁵¹

Algunos analistas, como por ejemplo Quocirca, define Grid Computing como una arquitectura específica:

"La computación Grid es un enfoque arquitectónico para la creación de una infraestructura tecnológica flexible, permitiendo la puesta en común recursos de red, hardware y software para satisfacer los requisitos de los procesos de negocio. Los componentes de una arquitectura de Grid (por ejemplo, unidades de computación, almacenamiento, bases de datos, aplicaciones funcionales y servicios) trabajan juntos para maximizar la utilización de los componentes y reducir al mínimo la necesidad de actualización continua de la capacidad de los componentes individuales. "⁵²

Partiendo de estudios de mercados podemos encontrar otra definición de Grid Computing desde el punto de vista económico como en modelo de negocio, Un grupo de expertos en Grid acuñó el término Grids de negocios definiendo la red como una infraestructura específica:

"Tenemos la visión de Grids de negocios como la adaptación orientada a servicios de infraestructura como servicios públicos para aplicaciones de negocios. Se convertirán en la columna vertebral de las TIC en general, logrando gran impacto económico en las economías del futuro. "⁵³

Los primeros éxitos con las redes nacionales en el área de eCiencia, así como las iniciativas libres, por ejemplo, Seti@Home dio lugar a escenarios más hacia un modelo de Utility Computing o el suministro de potencia de cálculo y aplicaciones como un servicio. Se hizo evidente que la Computación de Grid utilizara Internet como un medio de transporte y comunicación, y es una generalización posterior de la Web a medida que se extiende la clase de recursos accesibles con las aplicaciones, los datos, los recursos informáticos, instrumentos, sensores y similares.

⁵¹ Katarina Stanoevska Thomas Wozniak, Santi Ristol "Grid and Cloud Computing A Business Perspective on Technology and Applications" Springer Dec 2010, pp. 24

⁵² Katarina Stanoevska Thomas Wozniak, Santi Ristol "Grid and Cloud Computing A Business Perspective on Technology and Applications" Springer Dec 2010, pp. 24

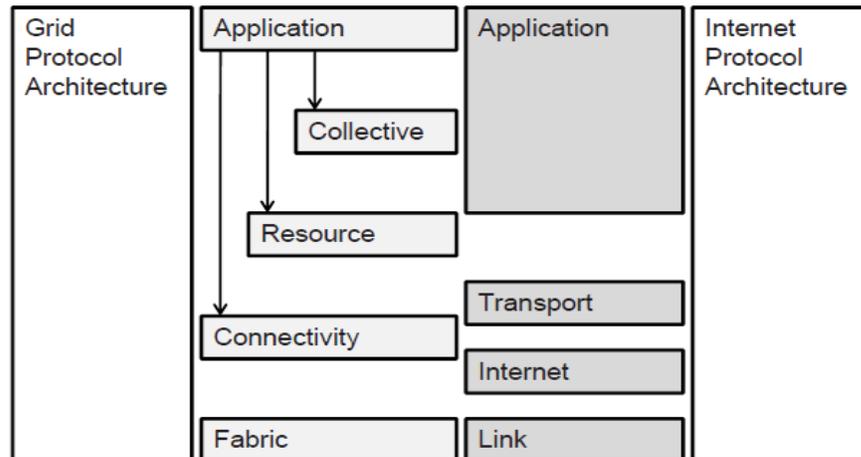
⁵³ Katarina Stanoevska Thomas Wozniak, Santi Ristol "Grid and Cloud Computing A Business Perspective on Technology and Applications" Springer Dec 2010, pp. 25

4.1.1. Arquitectura y Funcionalidad de Grid Computing: Para Katarina Stanoevska⁵⁴ Por medio de la arquitectura de Grid Computing podemos tener una visión general de los diferentes componentes, sus objetivos, funciones y como interactúa cada uno de ellos entre sí.

El objetivo principal de la arquitectura de Grid Computing está en la interoperabilidad de los protocolos entre los proveedores y usuarios de los recursos con el fin de establecer las relaciones de intercambio.

Los protocolos requeridos están organizados en capas que se presentan en la figura 3.

Figura 3, **Arquitectura genérica de Grid Computing**⁵⁵



Fuente: Katarina Stanoevska Thomas Wozniak, Santi Ristol “Grid and Cloud Computing A Business Perspective on Technology and Applications” Springer Dec 2010, pp. 26.

Según Katarina Stanoevska⁵⁶ la principal funcionalidad de cada capa se puede resumir de la siguiente manera:

⁵⁴ Katarina Stanoevska Thomas Wozniak, Santi Ristol “Grid and Cloud Computing A Business Perspective on Technology and Applications” Springer Dec 2010,

⁵⁵ Katarina Stanoevska Thomas Wozniak, Santi Ristol “Grid and Cloud Computing A Business Perspective on Technology and Applications” Springer Dec 2010,

4.1.1.1. La capa Fábrica (Fabric): Cuenta con los recursos físicos que se comparten dentro del Grid. Según Foster y Kesselman⁵⁷, esto incluye recursos computacionales, sistemas de almacenamiento, recursos de red, catálogos, módulos de software, sensores y otros recursos del sistema.

4.1.1.2. La capa Conectividad (Connectivity): Contiene el núcleo de los protocolos de comunicación y autenticación necesarios en la conexión de red en una Grid específica. Los protocolos de comunicación permiten el intercambio de datos entre los recursos de la capa fábrica. Las funciones más importantes en la capa de conectividad incluyen: el transporte, enrutamiento y nombres, así como apoyo para una comunicación segura, los requisitos más importantes de seguridad incluyen: un único inicio de sesión, delegación de los recursos que el usuario tiene acceso para acceder y ejecutar, integración e interoperabilidad con soluciones de seguridad y reglas locales.

4.1.1.3. La capa Recursos (Resource): Utiliza la comunicación y protocolos de seguridad (que son definidos por la capa de conectividad) permitiendo negociación segura, iniciación, control, contabilidad y el método de pago para el intercambio de recursos individuales. Se compone principalmente de información y protocolos de gestión. Los protocolos de información se utilizan para obtener información sobre la estructura y el estado de disposición de los recursos, los protocolos de gestión se utilizan para negociar el acceso a los recursos y servir como una política de punto de aplicación, asegurando que el uso de los recursos sea consistente con la política de cual recurso debe ser compartido.

4.1.1.4. La capa Colectivo (Collective): Es responsable de toda la gestión de los recursos globales y de interacción con las colecciones de recursos. Los protocolos de la capa Colectivo implementa una amplia variedad de conductas a compartir. Las más importantes funcionalidades de esta capa son: servicios de directorio, colocación, programación y servicios de intermediación, monitoreo y diagnóstico de servicios y servicios de replicación de datos. Los servicios de la capa colectivo suelen ser invocados mediante la programación modelos y herramientas: Esta capa

⁵⁶ Katarina Stanoevska Thomas Wozniak, Santi Ristol "Grid and Cloud Computing A Business Perspective on Technology and Applications" Springer Dec 2010,

⁵⁷ Katarina Stanoevska Thomas Wozniak, Santi Ristol "Grid and Cloud Computing A Business Perspective on Technology and Applications" Springer Dec 2010,

también se ocupa de comunidad de autorización, junto con la contabilidad y los servicios de pago.

4.1.1.5. La capa de aplicación (Application): Incluye las aplicaciones de usuario que se implementan en el Grid. Es importante señalar que no cualquier usuario de la aplicación se puede implementar en Grid. Sólo una aplicación de Grid o Gridified, es decir, una aplicación que se diseñó o ajustó para funcionar en paralelo y estableciendo el uso de múltiples procesadores de un Grid que se puede ejecutar en diferentes máquinas heterogéneas, se puede tomar ventaja de una infraestructura Grid.

Las cinco capas de Grid Computing se relacionan entre sí y dependen unas de otras. Cada capa posterior utiliza las interfaces de la capa subyacente. Juntas crean el Grid middleware y proporcionan un conjunto completo de funcionalidades necesarias que permite compartir de forma segura, fiable y eficiente los recursos (computadores, datos) entre entidades independientes.

Las principales funcionalidades de un Grid middleware son:

- Virtualización e integración de recursos heterogéneos autónomos.
- Provisión de información sobre los recursos y su disponibilidad.
- Asignación flexible y dinámica de los recursos y gestión.
- Intermediación de los recursos ya sea sobre políticas internas de una organización o la apertura a otras organizaciones.
- Seguridad y confianza, la seguridad incluye la autenticación (afirmación y confirmación de la identidad de un usuario) y la autorización (control de derechos para acceder a determinados servicios o datos) de los usuarios, así como su responsabilidad.
- Gestión de licencias
- Facturación y métodos de pago.
- Reparto no-trivial de Calidad de Servicio (QoS).

Teniendo en cuenta las anteriores funcionalidades, es evidente que Grid Computing es un sistema complejo y una sola tecnología no constituye un Grid.

Según Smith et al⁵⁸, los típicos componentes de servicios orientados a Grid middleware son: Globus Toolkit 4.0 (GT4), Tomcat 5.5 y Axis. Cada uno de estos

⁵⁸ Katarina Stanoevska Thomas Wozniak, Santi Ristol "Grid and Cloud Computing A Business Perspective on Technology and Applications" Springer Dec 2010.

tres, es un software de gran escala que abarca sistemas de miles de clases de Java, en muchos casos la funcionalidad necesaria es ensamblada por varios programas y proveedores de middleware.

4.1.2. Clasificación de Grid Computing: Según Katarina Stanoevska⁵⁹, para lograr entender la evolución de Grid Computing como una tecnología madura y su semejanza con nuevas tecnologías es necesario tener clara su clasificación. Grid Computing se pueden clasificar según diferentes criterios:

- ✓ Enfocado en los recursos.
- ✓ Alcance de uso compartido de los recursos involucrados

4.1.2.1. Clasificación de Grids de acuerdo con el enfoque de recursos: A pesar de que el objetivo final de Grid Computing es proporcionar el intercambio de cualquier tipo de los recursos, históricamente Grid middleware surgido con especial atención a determinados tipos de los recursos, de acuerdo con el enfoque de los recursos podemos distinguir los siguientes tipos de Grid middleware:

- Compute Grids: se centran en el intercambio de recursos informáticos, es decir, de la CPU.
- Data Grids: se centran en la gestión del almacenamiento controlado y la distribución a gran escala de datos heterogéneos y distribuidos.
- Application Grids: "tienen que ver con la gestión de aplicaciones proporcionando acceso a software de control remoto y bibliotecas de forma transparente "
- Service Grids: es el resultado de la convergencia de Grids orientadas al servicio de cómputo y apoyar el intercambio eficiente de los recursos. Estos cuatro tipos diferentes de Grid Computing convergieron en un solo Grid middleware genérico con funcionalidad combinada.

4.1.2.2. Clasificación de Grids en función del alcance de uso compartido de recursos: Dependiendo del alcance de la distribución de los recursos involucrados, podemos distinguir los siguientes enfoques de Grid Computing:

- Cluster Grids
- Enterprise Grids
- Utility Grid

⁵⁹ Katarina Stanoevska Thomas Wozniak, Santi Ristol "Grid and Cloud Computing A Business Perspective on Technology and Applications" Springer Dec 2010.

- Partner/Community Grids.
- **Cluster Grids: Cluster Grids, o clusters, son un conjunto homogéneo de computadoras las cuales se diferencian principalmente por su configuración, conectadas entre sí por una red de área local de alta velocidad que está diseñada para utilizar la integración de los recursos de datos y procesamiento de todos los equipos.**

En Cluster Grids los recursos locales operan dentro del firewall de la organización y están controlados por una sola entidad administrativa que tiene el control completo sobre cada componente, por lo tanto los grupos en realidad no implican el reparto de los recursos y no pueden ser consideradas como Grids en un sentido estricto.

Pero sin embargo, son generalmente el punto de partida para la construcción de Grids y un primer paso hacia Grid Computing.

- Enterprise Grid: El término Enterprise Grid se utiliza para referirse a la aplicación de Grid Computing que comparten los recursos dentro de los límites de una organización.

Todos los componentes de Enterprise Grid operan dentro del firewall de la organización, pueden ser heterogéneos y distribuidos físicamente en varias ubicaciones o sitios de la organización y pueden pertenecer a diferentes dominios administrativos.

Con un específico Enterprise Grid middleware, los recursos de TI disponibles están virtualizados y pueden ser gestionados de una manera unificada y central, también se pueden asignar los procesos de acuerdo a la demanda.

De acuerdo con NESSI-Grid⁶⁰, las soluciones disponibles en el mercado de Enterprise Grid se basan en políticas de programación y gestión de cargas de trabajo en infraestructuras heterogéneas, hechas para equipos de escritorio, servidores y clusters.

⁶⁰ Katarina Stanoevska Thomas Wozniak, Santi Ristol "Grid and Cloud Computing A Business Perspective on Technology and Applications" Springer Dec 2010.

Estos sistemas de control de los recursos básicos contienen los mecanismos de tolerancia a fallos, así como herramientas de análisis de rendimiento y depuración, debido a la falta de estandarización en este tipo de Grids, estas soluciones suelen apoyar una variedad de estándares de facto y convertirse en un formato de solución específica.

- **Utility Grid: Cuando la infraestructura tecnológica es propiedad de un tercero y está implementada por un proveedor de servicios se denomina Utility Grid. Los servicio que se ofrece a través de Utility Grid es denominado utility computing, es decir capacidad de cálculo y / o almacenamiento en forma de pago por uso.**

Utility Grid funciona fuera del firewall del usuario u organización, el usuario de Utility Grid no tiene control sobre su funcionamiento, esto quiere decir que el usuario u organización tiene que transmitir los datos y las solicitudes de computo hacia Utility Grid y recopilar los resultados también de ella.

Mediante el uso de Utility Grid, los riesgos de seguridad y privacidad, así como las preocupaciones respecto a la fiabilidad se incrementan, esto tiene un impacto fuerte en la decisión al utilizar Utility Grid o no.

El lado positivo de Utility Grid es que no requiere inversión inicial en infraestructura de TI y permite la transformación de los costos de inversión de capital en costes variables, Utility Computing además, proporciona recursos de escalabilidad y flexibilidad de TI por demanda.

- **Partner/Community Grids: La idea Partner/Community Grids procede de eScience, donde los multiplex esfuerzos de investigación, en particular en las ciencias naturales requirieron los esfuerzos conjuntos de de los científicos y el intercambio de las infraestructuras entre instituciones de investigación de todo el mundo.**

La cooperación por lo general es el resultado en una organización virtual (VO), dentro del cual se lleva a cabo el intercambio de recursos. Hoy en día, la necesidad de cooperación es cada vez mayor en el mundo de los negocios, debido a la globalización, las empresas están cada vez más involucradas en las cadenas de suministro globales y el éxito de una empresa depende cada vez más en la colaboración eficaz dentro de ellos.

La colaboración rápida y eficaz es a menudo obstaculizada por infraestructuras de TI heterogéneas e inflexibles que dificultan el intercambio eficiente de datos e implementación de procesos de la entre compañías. Partner/Community Grids son un tipo específico de Grids que pueden proporcionar soporte para el establecimiento de un VO basada en el intercambio de recursos de TI entre entidades colaboradoras. A pesar de Partner/Community Grids tienen funcionalidades similares, pueden soportar diferentes tipos de VOs: Partner/Community Grids son más bien establecida en el contexto de negocios entre empresas y universidades con objetivos comunes y definiendo las políticas de intercambio de recursos y relaciones, también se basan sobre la donación de recursos en su mayoría de instituciones privadas. Un ejemplo bien conocido de un Community Grid es SETI@HOME.

Partner/Community Grids están habilitadas por un middleware específico, que tiene las siguientes funcionalidades:

- La virtualización y exposición de los recursos de TI de cada organización participante en la organización virtual (VO).
- Soporte en la ejecución de las políticas de distribución de recursos.
- Coordinación de la ejecución de procesos y flujos de trabajo comunes, esto en particular implica la asignación de los recursos compartidos de las tareas más comunes.
- Soporte a la ejecución de seguridad y políticas de privacidad individuales y comunes.
- Soporte en el monitoreo de los recursos compartidos.
- Soporte en la medición del uso de los recursos comunes y compartidos y soporte para los procedimientos de autenticación, contabilidad y procedimientos de pago.
- Soporte opcional para el acceso a través de navegador a través de una interfaz Web.

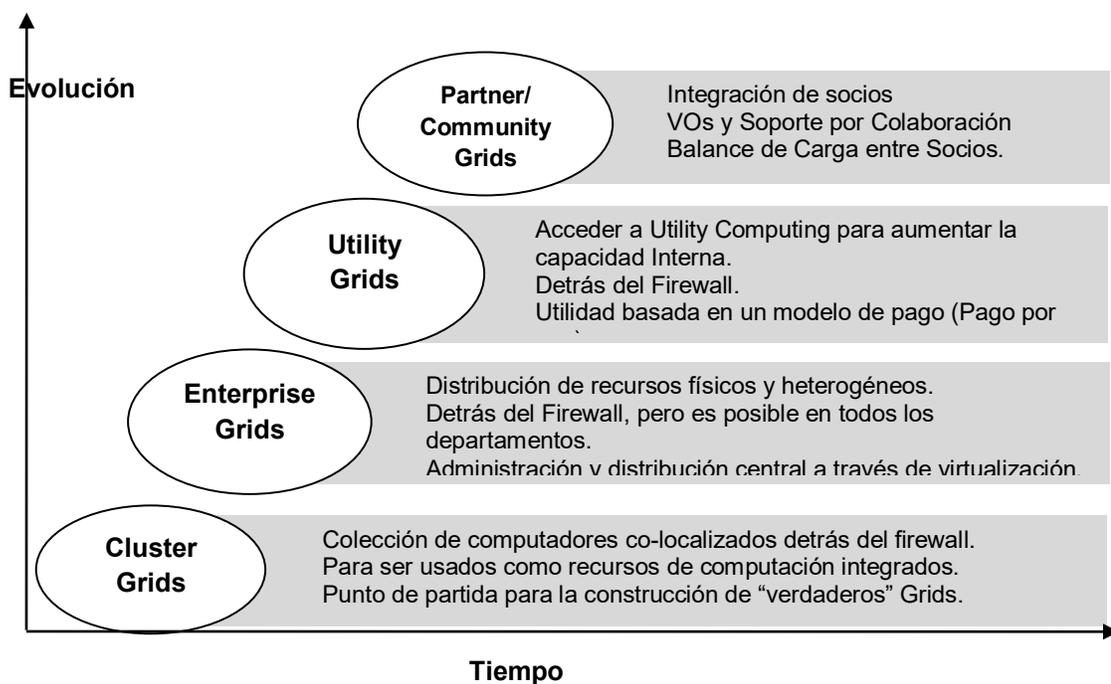
En una Partner/Community Grid cada socio participante ofrece una parte de su infraestructura para el intercambio de los recursos definiendo las reglas bajo las cuales los recursos pueden ser utilizados por otros socios y acepta las reglas de la comunidad para la donación de recursos, cada socio participante ofrece acceso a sus recursos de TI a sus socios y tiene acceso a la infraestructura de los socios.

La arquitectura de Partner/Community Grids puede ser vista como una colección de recursos independiente (por ejemplo Cluster Grids u otros recursos)

interconectados a través de un Grid middleware global, accesible opcionalmente a través de una interfaz Web.

4.1.3. Grid Computing hacia nuevas tecnologías: Los diferentes tipos de Grids descritos anteriormente permiten tener una clara perspectiva de la evolución y maduración de Grid Computing a través del tiempo.

Figura 4, “Evolución de Grid Computing a través del tiempo”⁶¹.



Fuente: Katarina Stanoevska Thomas Wozniak, Santi Ristol “Grid and Cloud Computing A Business Perspective on Technology and Applications” Springer Dec 2010, pp. 39.

Según Katarina Stanoevska⁶², la evolución de Grid Computing y la influencia de otros fenómenos TI que prevalecen al mismo tiempo, ha permitido el desarrollo de

⁶¹ Katarina Stanoevska Thomas Wozniak, Santi Ristol “Grid and Cloud Computing A Business Perspective on Technology and Applications” Springer Dec 2010, pp. 39

⁶² Katarina Stanoevska Thomas Wozniak, Santi Ristol “Grid and Cloud Computing A Business Perspective on Technology and Applications” Springer Dec 2010

nuevas tendencias tecnológicas influenciado el concepto inicial de Grid Computing, en particular se pueden identificar tres nuevas formas de computación:

- Computación orientada a servicios.
- Software como Servicio (SaaS).
- Cloud Computing (Computación en la Nube).

4.1.3.1. Convergencia de Grid Computing a computación orientada a servicios (SOC): La computación orientada a servicios (SOC) es un nuevo paradigma informático que se desarrolló en paralelo a Grid Computing, fue motivada e impulsada por la evolución y la necesidad del comercio electrónico de ser una aplicación fácil y eficiente de integrar dentro de las empresas.

De acuerdo con Papazoglou et al⁶³, la computación orientada a servicios (SOC) se define como:

Un nuevo paradigma informático que utiliza los servicios como la estructura básica para apoyar el desarrollo rápido, a bajo costo y de fácil composición en aplicaciones distribuidas incluso en entornos heterogéneos.

La promesa visionaria de la computación orientada a servicios es un mundo de servicios en cooperación, donde las solicitudes de aplicación son ensamblados con un bajo esfuerzo dentro una red de servicios que pueden ser libremente unida para crear flexibilidad en los procesos de negocios dinámicos y aplicaciones ágiles, que puede extenderse a muchas organizaciones y plataformas de computación⁶⁴.

Los elementos básicos que componen la computación orientada a servicios (SOC) son los servicios, que son autónomos, independientes de la plataforma entidades computacionales que pueden ser descritos, publicados, descubiertos y fácilmente acoplados de manera novedosa.

Hasta ahora los servicios más maduros y de mayor interés desde la perspectiva de Grid Computing son los Servicios Web. Los Servicios Web utilizan Internet como medio de comunicación y se definen basados en los estándares abiertos de Internet.

Las normas pertinentes para los servicios Web son:

⁶³ Katarina Stanoevska Thomas Wozniak, Santi Ristol "Grid and Cloud Computing A Business Perspective on Technology and Applications" Springer Dec 2010.

⁶⁴ Katarina Stanoevska Thomas Wozniak, Santi Ristol "Grid and Cloud Computing A Business Perspective on Technology and Applications" Springer Dec 2010, pp. 40

- Simple Object Access Protocol (SOAP) - el estándar para la transmisión de datos.
- Web Service Description Language (WSDL) - el estándar para la descripción unificada de los servicios
- Universal Description and Integration (UDDI), es una plataforma independiente, basada en el registro de los servicios.

Todos los estándares de servicios Web se basan en Extensible Markup Language (XML), el cual es legible por la máquina y permite la comunicación entre servicios. Las definiciones anteriores muestran que la computación orientada a servicios (SOC) tiene similitudes con Grid Computing, es decir la visión de Grid Computing respecta al intercambio y la interoperabilidad de recursos a nivel de hardware, al igual que la visión de la computación orientada a servicios (SOC) pero al nivel de software y aplicación, Otra característica común entre los dos conceptos es la noción de los servicios, la arquitectura de Grid Computing se compone de protocolos, es decir servicios necesarios para la descripción y el intercambio de recursos físicos disponibles.

La convergencia de la computación orientada a servicios (SOC) y los paradigmas de Grid Computing ofrece varias oportunidades:

- Mediante la aplicación de los estándares de servicios Web, protocolos y servicios Grid que pueden ser encapsulados y descritos de una manera estandarizada.
- Una vez que los paradigmas complementarios, Grid Computing y computación orientada a servicios (SOC) estén basados en la misma norma, su combinación es posible, esto significa que no sólo los recursos de hardware y el sistema se puede compartir, sino también correr aplicaciones en ellos.

La convergencia de Grid Computing con la computación orientada a servicios (SOC), quiere decir que la funcionalidad del Grid se presenta en forma de servicios.

La aplicación de la computación orientada a servicios (SOC) y el paradigma de Grid Computing tiene varias ventajas:

Primero que toda la tecnología Grid en línea actualmente es adoptada a gran escala por las empresas, además, el paradigma grid orientado a servicios ofrece la posibilidad de ofrecer una virtualización de los recursos disponibles para aumentar la flexibilidad de un Grid y también proporciona un elemento de enlace entre los servicios Grid específicos a nivel de hardware y los servicios de aplicación.

4.1.3.2. Convergencia de Grid Computing y Software como servicio (SaaS): Otro paradigma importante que ha ganado impulso, junto a Grid Computing y la computación orientada a servicios (SOC) es el paradigma de Software como servicio (SaaS).

El término de Software como servicio (SaaS) se refiere al software que es propiedad y se gestiona de forma remota por uno o más proveedores de software independientes y que se ofrece en la modalidad de pago por uso.

El Software como servicio (SaaS) se consume a través de redes de comunicación (por lo general Internet) y el usuario puede acceder ya sea a través de un navegador web o accediendo directamente a las interfaces de programación de aplicaciones (API).

El concepto de Software como servicio (SaaS) implica cambios sustanciales en la forma cómo se desarrolla y se consume el software, el software tradicional es empaquetado y está preparado para trabajar en los computadores de los usuarios finales, por esta razón para ejecutar los paquetes de software el usuario debe poseer su propia infraestructura, lo cual puede resultar en inversiones adicionales además debe contar con funcionarios calificados que sea capaz de proporcionar mantenimiento y apoyo al software y hardware que se necesita.

Al comprar una licencia de software a un usuario se le otorga el derecho a utilizar el software, las licencias para el software típico tradicional son: licencias para el uso por tiempo (por ejemplo un año), las licencias para utilizar el software durante un cierto tiempo y el número de equipos (por ejemplo, un año y máximo de seis computadoras), las licencias para un período de tiempo determinado y por número de usuarios (por ejemplo, un año y máximo de seis personas) y similares. Por uso, el usuario paga por la funcionalidad del software sólo por el tiempo y la intensidad de cada uso específico, el usuario no posee el software y no tiene que preocuparse de invertir en infraestructura para el funcionamiento del software y en el personal necesario para el mantenimiento del mismo.

El Software como servicio (SaaS) no es un fenómeno nuevo, la idea de compartir el software, es decir, el acceso remoto de software por varios usuarios ha sido una

visión desde el principio de distribución informática en las empresas. En ese momento fue llamado "tiempo compartido" de software que se ejecuta en un servidor remoto y era utilizado por varios usuarios en una red privada.

El siguiente impulso que evoluciono la provisión de aplicaciones como un servicio llegó con la creación de la Internet como el principal medio de comunicación.

Según Katarina Stanoevska⁶⁵, en 1998, el término Prestación de Servicios de Aplicaciones (ASP) fue presentado. ASP evolucionado a partir de outsourcing de TI y se basa en la idea que una aplicación web que se puede proporcionar en línea a través de IP basados en la infraestructura de telecomunicaciones por un proveedor de servicios de aplicaciones centrales.

Las principales ventajas con las que se propagó el modelo de negocio ASP a los clientes eran las siguientes: ahorro de costos sin necesidad de desarrollo y mantenimiento de infraestructura propia y habilidades.

A pesar de que ASP fue considerado como un modelo de negocio muy prometedor a finales de 1990, no pasó mucho tiempo hasta que su adopción en el mercado eran muy lenta y una de las razones principales del fracaso de la ASP fue la incapacidad de los primeros proveedores de ofrecer servicios de aplicaciones personalizados. La aplicación fue siempre de la misma forma para cualquier cliente, había pocas posibilidades de personalización y debido a la fase inicial de desarrollo de Internet y las tecnologías Web los proveedores no fueron capaces y ofrecer servicios fiables y robustos. Otras razones para el fracaso de la ASP había sido: el enfoque centralizado para la informática, lo que requiere el envío de entrada y salida datos y la falta general de confianza en el paradigma prestación de Servicios de Aplicaciones (ASP).

En la actualidad, Grid Computing y las prestaciones de Servicios de Aplicaciones (ASP) están convergiendo hacia el Software como servicio (SaaS). El acrónimo Software como servicio SaaS denota una nueva evolución en la entrega de software como un servicio basado en servicios Web (Web Services) y tecnología Grid.

La convergencia de los servicios Web y la tecnología de Grid Computing proveen nuevas oportunidades para resolver los problemas de distribución de Prestación de Servicios de Aplicaciones (ASP).

⁶⁵ Katarina Stanoevska Thomas Wozniak, Santi Ristol "Grid and Cloud Computing A Business Perspective on Technology and Applications" Springer Dec 2010.

Los servicios Web permiten la modularización de las aplicaciones en varios servicios que pueden ser combinados y personalizados por los usuarios.

La Tecnología de Grid tiene el potencial para proporcionar la necesaria flexibilidad y escalabilidad de la infraestructura al lado de las ofertas Software como servicio (SaaS).

4.1.3.3. Evolución hacia Cloud Computing. La maduración de la tecnología Grid permitió nuevos modelos de negocio como utility computing, proporcionado potencia de cálculo bajo demanda basado en una forma de pago por uso.

Mientras que el desarrollo de la tecnología Grid eran básicamente impulsada por los proveedores de hardware y sistema como Sun e IBM, a la vez una evolución en la industria del software hacia el Software como servicio (SaaS) impulsadas por los proveedores de software como, por ejemplo, Microsoft y SAP.

Ambos acontecimientos Utility Computing y Software como servicio (SaaS), ilustran la creciente tendencia hacia la implementación y el aprovisionamiento externo de la computación y las aplicaciones.

Utility computing y Software como servicio (SaaS), son dos tendencias complementarias: utility computing sólo puede tener éxito en el mercado si es capaz de correr una gran cantidad crítica de aplicaciones en él y el Software como servicio (SaaS), necesita una infraestructura flexible, escalable y de fácil acceso en el que se puede ejecutar las aplicaciones. Con el fin de satisfacer la demanda del mercado, la integración de estas dos tendencias se presenta en nuevo enfoque global que ofrece las siguientes funciones:

- Infraestructura física flexible, escalable, robusta y fiable.
- Una Plataforma de servicios que permiten programar el acceso a la infraestructura física a través de interfaces abstractas.
- Software como servicio (SaaS) desarrollado, implementado y corriendo una infraestructura física flexible y escalable.

Tabla 2, “Evolución de Grid Computing a Cloud Computing”⁶⁶

⁶⁶ Katarina Stanoevska Thomas Wozniak, Santi Ristol “Grid and Cloud Computing A Business Perspective on Technology and Applications” Springer Dec 2010, pp. 45

Grid Computing	Utility Computing	Software as a Service	Cloud Computing
La solución a grandes problemas con la computación paralela	Ofrece los recursos informáticos como servicios a la medida.	Basados en la red por suscripción a aplicaciones	Nueva Generación de la Computación en Internet.
Incorporaciones hechas por Globus Alliance	Incorporaciones hechas por Globus Alliance	Impulso generado en 2001	Revolución de los centros de datos.
A finales de 1980	A finales de 1990		

Fuente: Katarina Stanoevska Thomas Wozniak, Santi Ristol “Grid and Cloud Computing A Business Perspective on Technology and Applications” Springer Dec 2010, pp. 45.

Todo esto se está convirtiendo en nuevas plataformas en línea conocido como nubes (Clouds) o Computación en la nube (Cloud Computing).

Cloud Computing es resultado de la convergencia de Grid Computing, Utility Computing y Software como servicio (SaaS), fundamentalmente representa la creciente tendencia del despliegue externo de los recursos de TI, tales como la potencia de cálculo, almacenamiento, aplicaciones empresariales obtenidas como servicios.

Esta nueva tendencia computacional basada a un modelo de X como servicio (XaaS) que tiene la capacidad de cambiar la distribución y manejo de los recursos TI se explica con más detalle en el siguiente apartado que hace referencia a la computación en la nube (Cloud Computing).

4.2. CLOUD COMPUTING (COMPUTACIÓN EN LA NUBE).

Para Katarina Stanoevska⁶⁷ se podría decir que la computación en la nube nació gracias a la evolución de múltiples tecnologías y como resultado de la convergencia de Grid Computing, Utility Computing y Software como servicio (SaaS), pero esto sería solo un concepto desde el punto de vista técnico, lo cierto es que la computación en la nube es un modelo tecnológico joven el cual ha revolucionado la industria TI en los últimos años a pesar de ser el resultado de evolución de tecnologías maduras, sin embargo actualmente se ha definido la computación en la nube de muchas maneras por las firmas de analistas, académicos, profesionales de la industria y las empresas de TI.

Según Gartner⁶⁸ "Un estilo de computación donde las capacidades de TI son masivamente escalables, proporcionadas como un servicio, utilizando las tecnologías de Internet a múltiples clientes externos"

Según el grupo 451⁶⁹ "Un modelo de servicio que combina un principio general de organización para la prestación de TI, componentes de infraestructura, un enfoque de arquitectura y un modelo básicamente económico, una confluencia de grid computing, la virtualización, utility computing, hosting y software como servicio (SaaS)".

La mayoría de definiciones de las firmas de analistas tiene una característica común, tratar de describir y definir la computación en la nube (Cloud Computing) desde la perspectiva del usuario final y su atención se centra en cómo puede ser experimentado por ellos.

De acuerdo con estas definiciones, las características básicas de la computación en la nube (Cloud Computing) es la provisión de infraestructura de TI y las aplicaciones como un servicio de una manera escalable.

A diferencia de las definiciones de los analistas, la comunidad científica no sólo se centra en la perspectiva del usuario final, sino también en los aspectos

⁶⁷ Katarina Stanoevska Thomas Wozniak, Santi Ristol "Grid and Cloud Computing A Business Perspective on Technology and Applications" Springer Dec 2010.

⁶⁸ Katarina Stanoevska Thomas Wozniak, Santi Ristol "Grid and Cloud Computing A Business Perspective on Technology and Applications" Springer Dec 2010, pp. 48

⁶⁹ Katarina Stanoevska Thomas Wozniak, Santi Ristol "Grid and Cloud Computing A Business Perspective on Technology and Applications" Springer Dec 2010, pp. 48

arquitectónicos. Por ejemplo, RAD Berkeley Lab⁷⁰. Define a la computación en la nube (Cloud Computing), de la siguiente manera.

La computación en la nube (Cloud Computing) se refiere a la entrega de servicios a través de Internet de hardware y sistemas de software de los centros de datos que proporcionan esos servicios. Los mismos servicios han sido durante mucho tiempo conocidos como Software como Servicio (SaaS). El hardware y el software de los centro de datos es a lo que se le llama una nube. Cuando una nube se hace disponible de manera pay-as-you-go al público en general, la llamamos una nube pública, el servicio que se vende es utility computing. Nosotros usamos la nube privada para referirse a los centros de datos internos de una empresa u otra organización, que no está a disposición del público en general. De este modo, Cloud Computing es la suma de SaaS y Utility Computing, pero no incluye las nubes privadas en donde las personas pueden ser usuarios o proveedores de SaaS, o los usuarios o proveedores de utility Computing⁷¹.

La anterior definición reúne diferentes perspectivas sobre la nube, desde la perspectiva de un proveedor el componente de la nube más importante es el centro de datos. El centro de datos contiene los recursos de hardware para la computación y el almacenamiento, que junto con el software se ofrecen de manera pay-as-you-go.

Desde la perspectiva de su propósito, las nubes se clasifican en públicas y privadas, independiente de la finalidad de las Nubes, una de las características más importantes de las nubes es la integración de hardware y software con las aplicaciones, es decir, la integración de utility Computing y SaaS.

También Reese⁷², señala que una nube puede ser tanto de software e infraestructura y hace hincapié en la manera de cómo los servicios de la nube pueden ser consumidos.

⁷⁰ Katarina Stanoevska Thomas Wozniak, Santi Ristol "Grid and Cloud Computing A Business Perspective on Technology and Applications" Springer Dec 2010.

⁷¹ Katarina Stanoevska Thomas Wozniak, Santi Ristol "Grid and Cloud Computing A Business Perspective on Technology and Applications" Springer Dec 2010, pp. 48.

⁷² Katarina Stanoevska Thomas Wozniak, Santi Ristol "Grid and Cloud Computing A Business Perspective on Technology and Applications" Springer Dec 2010.

"La (Nube) El servicio es accesible mediante un navegador web (no propietario) o servicios web API's. Cero gastos de inversión son necesario para empezar, usted sólo paga por lo que se utiliza, a medida que lo utiliza"⁷³.

Foster et al. Define la computación en la nube (Cloud Computing) como: "Paradigma a gran escala de computación distribuida que está impulsado por las economías de escala, en que un grupo de abstracción, virtualizados, dinámicamente escalable, gestionan el poder de cómputo, de almacenamiento, plataformas y servicios se entregan bajo demanda a clientes externos a través de Internet"⁷⁴.

Dos aspectos importantes añadidos por la definición de Foster et al. (2008), son virtualización y escalabilidad. La computación en la nube (Cloud Computing) es la abstracción del hardware subyacente y sistema de software mediante la virtualización. Los recursos virtualizados se proporcionan a través de una interfaz definida de abstracción (una interfaz de programación de aplicaciones (API) o un servicio). Por lo tanto a nivel de hardware puro, los recursos se pueden agregar o retirar de acuerdo a la demanda, publicado a través de la interfaz, mientras que la interfaz del usuario no está cambiando. Esta arquitectura permite la escalabilidad y la flexibilidad en la capa física de una nube, sin impacto en la interfaz para el usuario final.

Finalmente, Vaquero et al⁷⁵. Analizo mas de 22 definiciones de la computación en la nube, todas las propuestas en 2008. Con base en ese análisis, Vaquero et al propone la siguiente definición que intenta reflejar cómo es Cloud Computing concebido en la actualidad:

Las nubes son una gran reserva de recursos virtualizados fácilmente utilizables y accesibles (por ejemplo, hardware, plataformas de

⁷³ Katarina Stanoevska Thomas Wozniak, Santi Ristol "Grid and Cloud Computing A Business Perspective on Technology and Applications" Springer Dec 2010, pp. 49

⁷⁴ Katarina Stanoevska Thomas Wozniak, Santi Ristol "Grid and Cloud Computing A Business Perspective on Technology and Applications" Springer Dec 2010, pp. 49

⁷⁵ Katarina Stanoevska Thomas Wozniak, Santi Ristol "Grid and Cloud Computing A Business Perspective on Technology and Applications" Springer Dec 2010.

desarrollo y / o servicios). Estos recursos pueden ser dinámicamente reconfigurados para adaptarse a una carga variable (escala), contemplando asimismo la óptima utilización de los recursos. Este conjunto de recursos suele ser ofrecido por un modelo de pago por uso con garantías ofrecidas por el proveedor de la infraestructura por medio de medidas de SLAs⁷⁶.

Adicional se puede encontrar el aporte a la definición de la computación en la nube por los grupos y organizaciones de estándares mundiales:

Según el National Institute of Standards and Technology NIST⁷⁷ “Computación en la nube”, es un modelo para habilitar el acceso a un conjunto de servicios computacionales (redes, servidores, almacenamiento, aplicaciones y servicios) de manera conveniente y por demanda, que pueden ser rápidamente aprovisionados y liberados con un esfuerzo administrativo y una interacción con el proveedor del servicio mínimos”.

Según el manifiesto de la nube abierta (Open Cloud Manifiesto)⁷⁸,” la computación en la nube es la culminación de muchas tecnologías, computación de maya (Grid Computing), utility computing, SOA, Web 2.0 y otras tecnologías”.

Con base en el análisis de las definiciones, un resumen se podría definir las principales características de la computación en nube (Cloud Computing):

- Cloud Computing es un nuevo paradigma informático.
- Recursos de Infraestructura (hardware, almacenamiento y software) y aplicaciones se presentan de una manera X como Servicio. Cuando estos servicios se ofrecen por un proveedor independiente a clientes externos, la computación en la nube se basa en el modelo de negocio de pago por uso.
- Las características principales de la computación en la nube son la virtualización y escalabilidad dinámica por demanda.

⁷⁶ Katarina Stanoevska Thomas Wozniak, Santi Ristol “Grid and Cloud Computing A Business Perspective on Technology and Applications” Springer Dec 2010, pp. 49

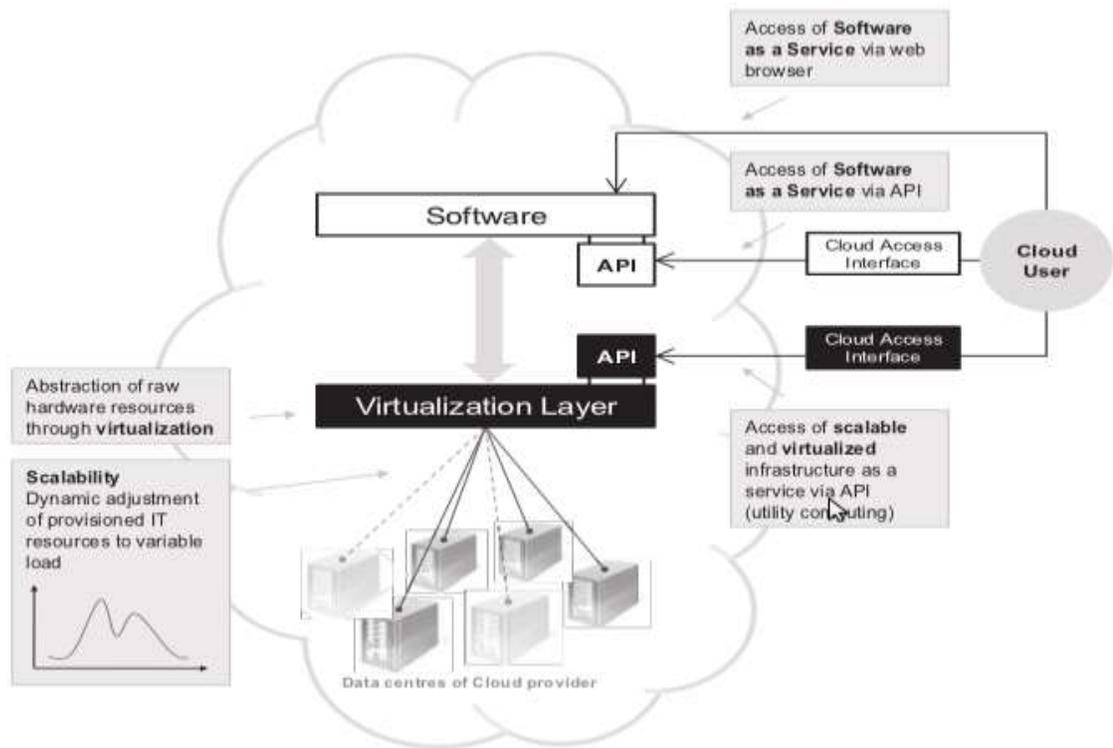
⁷⁷ National Institute of Standards and Technology, Definition Of Cloud Computing, Disponible <http://csrc.nist.gov/publications/nistpubs/800-145/SP800-145.pdf>, Consultado 11/11/2010

⁷⁸ Opencloudmanifesto.org, manifiesto de la nube libre, Disponible <http://www.opencloudmanifesto.org/>, Consultado 08/01/2011.

- Utility Computing y Software como servicio (SaaS) se proporcionan de manera integrada, a pesar de que Utility Computing pueden ser consumido por separado.
- Los servicios de la nube se consumen ya sea a través del navegador Web o a través de un API definida.

**Figura 5, “Definición de las características de la computación en la nube”
(Cloud Computing) ⁷⁹**

⁷⁹ Katarina Stanoevska Thomas Wozniak, Santi Ristol “Grid and Cloud Computing A Business Perspective on Technology and Applications” Springer Dec 2010, pp. 50



Fuente: Katarina Stanoevska Thomas Wozniak, Santi Ristol “Grid and Cloud Computing A Business Perspective on Technology and Applications” Springer Dec 2010, pp. 45.

4.2.1. Modelos de despliegue de Computación en la Nube. El Instituto Nacional de Estándares y Tecnología de los Estados Unidos de América NIST⁸⁰, define cuatro modelos de despliegue. Estos pueden ser considerados como la estructura básica de los modelos de negocios de Computación en la nube.

4.2.1.1. Nubes públicas: La infraestructura de la Nube está disponible para el público en general o puede estar orientado a una industria en particular. Esta infraestructura es propiedad de la organización que ofrece los servicios en la nube.

4.2.1.2. Nubes privadas: En el modelo de Computación de nube privado, la infraestructura está orientada a servir a una sola organización. Su gestión puede ser realizada por la organización o por un tercero.

⁸⁰ National Institute of Standards and Technology, Definition Of Cloud Computing, Disponible <http://csrc.nist.gov/publications/nistpubs/800-145/SP800-145.pdf>, Consultado 11/11/2010

4.2.1.3. Nubes comunitarias: La infraestructura es compartida por diferentes organizaciones y está orientada a servir a un tipo de comunidad específica de acuerdo a sus intereses comunes. Su gestión puede ser realizada por las organizaciones o por un tercero.

4.2.1.4. Nubes híbridas: Este modelo es la unión del modelo público y el modelo privado. Bajo este modelo dos o más nubes (privadas, públicas o comunitarias) se unen a través de tecnología estándar o propietaria que facilite la portabilidad de datos y aplicaciones, por ejemplo, en caso de cloud bursting, donde se balancea la carga entre las nubes cuando una de ellas ha ocupado su capacidad máxima.

4.2.2. Las Tres capas de la arquitectura de la computación en la nube (Cloud Computing). Según Barrie Sosinsky⁸¹, la computación en la nube (Cloud Computing) cuenta con diferentes capacidades de TI, infraestructura, plataforma y software, esto también se conoce como diferentes 'formas', 'segmentos', 'estilo', 'tipo', 'niveles' o "Capas" de la computación en la nube.

En vez de hablar de diferentes capacidades pensando en ella como diferentes capas tiene mucho más sentido porque la infraestructura, la plataforma y el software se desarrollan luego sobre el nivel precursor y son lógicamente conectados con las diferentes capas de la arquitectura de la nube.

Independiente del término utilizado, esta triple clasificación de la computación en la nube se ha vuelto común.

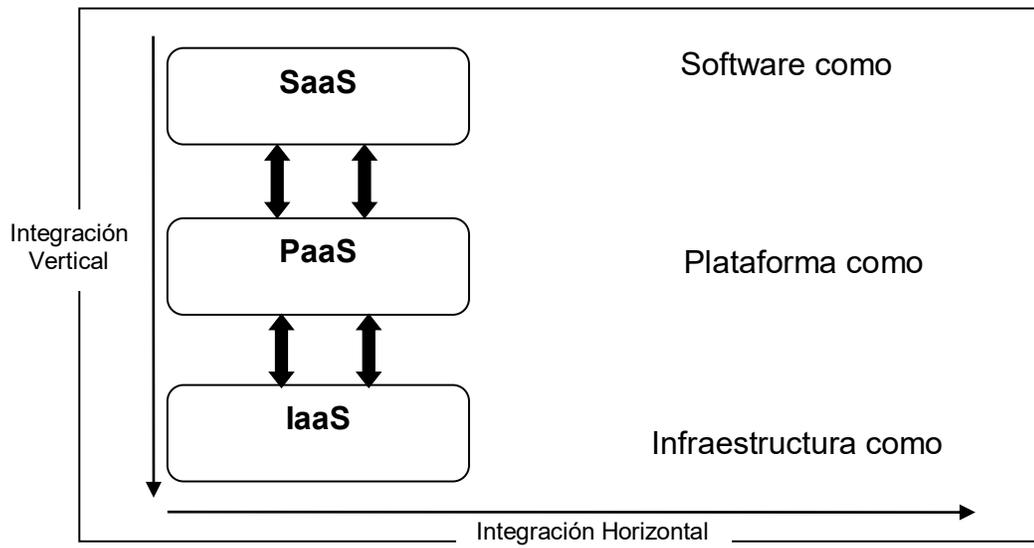
Como la distribución de los recursos de TI o de capacidades como un servicio es una característica importante de Cloud Computing, las tres capas de la arquitectura de Cloud Computing son:

- Infraestructura como servicio (IaaS)
- Plataforma como Servicio (PaaS)
- Software como Servicio (SaaS).

Figura 6, “Las tres capas de la Computación en la nube: SaaS, PaaS y IaaS”⁸²

⁸¹ Barrie Sosinsky, Cloud Computing Bible, Wiley Publishing, Indiana 2011

⁸² Katarina Stanoevska Thomas Wozniak, Santi Ristol “Grid and Cloud Computing A Business Perspective on Technology and Applications” Springer Dec 2010, pp. 52



Fuente: Katarina Stanoevska Thomas Wozniak, Santi Ristol "Grid and Cloud Computing A Business Perspective on Technology and Applications" Springer Dec 2010, pp. 52.

4.2.2.1. Infraestructura como servicio (IaaS). Para Barrie Sosinsky⁸³, la capa de Infraestructura como servicio (IaaS), es la que proporciona los recursos informáticos tales como el procesamiento, el almacenamiento y recursos de red, los cuales pueden ser obtenidos como un servicio.

Ejemplos de ello son los servicios web de Amazon con su Elastic Compute Cloud (EC2) para su procesamiento y Simple Storage Service (S3) para el almacenamiento y Joyent que ofrecen gran escalabilidad bajo demanda de infraestructura para el funcionamiento de sitios Web y aplicaciones en Internet. Los proveedores de IaaS se caracterizan por ofrecer infraestructura virtualizada como un servicio, indicando el nivel de recursos de hardware, tales como procesamiento, almacenamiento y recursos de red.

Normalmente por la virtualización, los recursos a nivel de hardware se abstraen y se encapsulan, por lo tanto pueden estar expuestos en la capa superior a usuarios finales a través de una interfaz estandarizada, como recursos unificados, el clientes no gestionan ni controlan la infraestructura de la Nube, pero tienen control

⁸³ Barrie Sosinsky, Cloud Computing Bible, Wiley Publishing, Indiana 2011.

sobre los sistemas operativos, almacenamiento, aplicaciones desplegadas y la posibilidad de tener un control limitado de componentes de red seleccionados.

La infraestructura como servicio (IaaS) denota su evolución hacia el soporte integrado en las tres capas (IaaS, PaaS y SaaS) dentro de la nube. Los proveedores de Utility computing se dieron cuenta que para tener éxito, era necesario proporcionar una interfaz fácil de acceder, comprender, programar y utilizar, es decir una API que permite una fácil integración con la infraestructura de los clientes, los desarrolladores de aplicaciones y los usuarios de Software.

4.2.2.2. Plataforma como servicio (PaaS). Para Barrie Sosinsky⁸⁴, la capa de plataforma como servicio (PaaS) se encuentra entre las capas de infraestructura como servicio (IaaS) y software como servicio (SaaS), siendo esta la que provee las herramientas dentro de un entorno de desarrollo para crear aplicaciones que pueden ejecutarse en el modelo de Software como Servicio (SaaS).

La capa de Plataforma como servicio (PaaS) es una de los mas interesantes dentro de todos los servicios alojados en la Computación en la nube, la capa de Infraestructura como servicio (IaaS) ofrece un servicio que es similar al de instalar el software en un equipo, a pesar de ser virtual, se considera todavía un equipo con su sistema operativo, configuración de hardware, red y demás aplicaciones, la capa de Software como servicio (SaaS), el software está muy bien diseñado para el usuario, pero el usuario solo puede hacer algún tipo de cambios limitados, el usuario accede al software a través de Internet el cual se encuentran alojado en alguna parte que es transparente para el usuario.

En la capa de Plataforma como servicio al usuario se le dan un conjunto de herramientas para trabajar en una maquina virtual en donde tiene la capacidad de diseñar, desarrollar, probar y ejecutar el software de acuerdo a sus necesidades, encontrando lenguajes de programación basados es tecnologías Web orientados al servicio.

Así es como la capa de plataforma como servicio (PaaS) abarca una amplia gama de servicios, tales como.

- Desarrollo de aplicaciones: al usuario de le provee de una plataforma de desarrollo o se le proporcionan las herramientas de desarrollado visuales para

⁸⁴ Barrie Sosinsky, Cloud Computing Bible, Wiley Publishing, Indiana 2011.

escribir su propio código con lenguajes compatibles y crear sus propias aplicaciones.

- Colaboración: muchos sistemas de plataforma como servicio (PaaS) están configurados para permitir que varias personas trabajen en los mismos proyectos.
- Gestión de datos: Se proporcionan herramientas para acceder y utilizar datos en un almacén de datos.
- Instrumentación, ejecución y pruebas: las herramientas están disponibles para la medición de sus aplicaciones y la optimización de su rendimiento.
- Almacenamiento: Los datos pueden ser almacenados en cualquiera de los servicios del proveedor de la plataforma como servicio (PaaS) o acceder a un servicio de almacenamiento de terceros.
- Gestión de transacciones: Muchos sistemas de plataforma como servicio (PaaS) proporcionan servicios tales como administradores de transacciones o servicios de intermediación para mantener la integridad de las transacciones.

Un buen proveedor de plataforma como servicio (PaaS) tiene ciertas características deseables que son importantes en el desarrollo de aplicaciones robustas, escalables y portátiles. El cual debería tener los siguientes atributos.

- Separación de la gestión de los datos de la interfaz de usuario
- Dependencia de las normas de cloud computing
- Un ambiente de desarrollo integrado (IDE)
- Herramientas de Gestión del ciclo de vida
- Arquitectura de apoyo Multi-tenant, seguridad y escalabilidad, herramientas de rendimiento, monitoreo, pruebas y optimización.

Dentro de los servicios más importantes y utilizados en una plataforma como servicio (PaaS) es el desarrollo de aplicaciones en donde se debe proporcionar las herramientas necesarias para construir los diferentes tipos de aplicaciones que puedan trabajar juntas en el mismo entorno. Estos son algunos de los tipos de aplicaciones comunes.

- Las aplicaciones compuestas empresariales

- Portales de datos
- Aplicaciones web híbridas de múltiples fuentes de datos

Estas aplicaciones debe ser capaces de compartir datos y ejecutarse en un entorno multi-tenant, para hacer que las aplicaciones trabajen juntas con mayor facilidad, se utilizan lenguajes de desarrollo comunes, tales como Java o Python que se ofrece habitualmente para desarrolladores.

4.2.2.3. Software como servicio (SaaS). Para Barrie Sosinsky⁸⁵, el software como servicio (SaaS) es la capa más visible de Cloud Computing para los usuarios finales, porque se trata de las aplicaciones de software actuales a la que se tiene acceso.

Desde la perspectiva del usuario la obtención de software como servicio es principalmente motivado por las ventajas de costos, debido al modelo basado en la utilidad de pago, es decir no tener que asumir inversiones de infraestructura por adelantado.

Ejemplos bien conocidos de las ofertas de software como servicio (SaaS) son Salesforce.com y Google con Google Mail y Google Docs.

El usuario típico de una oferta de software como servicio (SaaS), normalmente no tiene ni el conocimiento ni el control sobre la infraestructura subyacente, ya sea software sobre la capa de plataforma como servicio (PaaS), infraestructura como servicio (IaaS) o una infraestructura de hardware tradicional.

Sin embargo, este modelo es muy pertinente para los proveedores de Software como Servicio (SaaS), ya que son necesarias y pueden ser subcontratadas, por ejemplo, una aplicación en el modelo Software como servicio (SaaS), puede ser desarrollada una plataforma como servicio (PaaS) existente y se ejecuta en la infraestructura de un tercero, así como la infraestructura como un servicio (IaaS) es atractivo para los proveedores de SaaS, ya que puede aliviar los costos de licencias e inversión en infraestructura y mantenimiento, también les permite centrarse en sus competencias principales, esto es similar a los beneficios que motivan a los usuarios a obtener el software como servicio (SaaS).

⁸⁵ Barrie Sosinsky, Cloud Computing Bible, Wiley Publishing, Indiana 2011.

El modelo SaaS ofrece a los clientes una forma económica de acceder a los servicios del software sin invertir grandes cantidades de dinero en licencias de uso o en infraestructura computacional para alojar las aplicaciones.

El software como servicio garantiza una disponibilidad de las aplicaciones todo el tiempo a todas horas ya que las aplicaciones son de tipo web, disponible permanentemente en la red.

5. EL USO DEL SOFTWARE LIBRE EN LA EVOLUCION DE GRID COMPUTING A CLOUD COMPUTING

5.1. EL USO DE SOFTWARE LIBRE EN GRID COMPUTING (COMPUTACIÓN DE MALLA).

Uno de los más grandes aportes que permitió el desarrollo y la evolución de Grid Computing como una tecnología robusta y madura es el papel del el Software Libre.

A pesar de que hoy en día se encuentra software y middleware con licencias propietarias para la implementación de Grids, se puede afirmar que Grid Computing desde sus inicios es 100% software libre gracias a los aportes de las comunidades científicas y de la industria. Una prueba de ello es la creación de los estándares Grid desarrollados por el Open Grid Forum (OGF) y el uso del Globus Toolkit para el desarrollo de nuevo software para Grid Computing.

5.1.1. Open Grid Forum (OGF): Según el Open Grid Forum (OGF)⁸⁶, es una comunidad de usuarios, desarrolladores y proveedores líderes en el esfuerzo de la estandarización global para Grid Computing. La comunidad OGF se compone de miles de personas en la industria y la investigación, lo que representa más de 400 organizaciones en más de 50 países. Juntos trabajan para acelerar la adopción de

⁸⁶ Open Grid Forum, Sitio Web [En línea], [Julio 2011]. Disponible en: http://www.ogf.org/About/abt_overview.php

Grid Computing en todo el mundo porque se cree que las redes se llevan a nuevos descubrimientos, nuevas oportunidades y mejores prácticas de negocio.

El trabajo de OGF se lleva a cabo a través de la comunidad iniciaron los grupos de trabajo, que se desarrollan las normas y especificaciones en cooperación con otras organizaciones líderes de las normas, los fabricantes de software, y los usuarios. OGF es financiado a través de sus miembros de la organización, incluyendo las compañías de tecnología y las instituciones académicas y gubernamentales de investigación.

5.1.2. El Open Grid Services Architecture (OGSA): Según el Open Grid Forum⁸⁷, es la arquitectura de sistemas Grid basada en Servicios Web, la cual aparece como la referencia clave para los proyectos en desarrollo Grid Computing, los Servicios Web es una arquitectura del W3C (Word Wide Web Consortium) para el desarrollo de servicios avanzados, a través de interfaces y protocolos consistentes y estándar, como: XML (eXtensible Markup Language), SOAP (Simple Object Access Protocol), WSDL (Web Service Description Language), y UDDI (Universal Description, Discovery and Integration).

Esta asegurando la interoperabilidad de sistemas heterogéneos de manera que diferentes tipos de recursos pueden comunicarse y compartir información. OGSA ha sido descrito como un refinamiento de la arquitectura de servicios web emergentes, específicamente diseñado para soportar los requerimientos de Grid Computing. OGSA ha sido adoptado como una arquitectura de red por una serie de proyectos de Grid incluyendo el Globus Alliance. Conceptualmente OGSA se sugirió por primera vez en un artículo por Ian Foster y más tarde desarrollada por Global Grid Forum (GGF), que resultó en un documento de información, titulado The Open Grid Services Architecture, versión 1.5.

5.1.3. Open Grid Services Infrastructure (OGSI). Publicado por el Open Grid Forum (OGF)⁸⁸, en Junio de 2003, fue pensado para proporcionar una capa de infraestructura para el Open Grid Services Architecture (OGSA).

⁸⁷ Open Grid Forum, Sitio Web [En línea], [Julio 2011]. Disponible en: http://www.ogf.org/About/abt_overview.php

⁸⁸ Open Grid Forum, Sitio Web [En línea], [Julio 2011]. Disponible en: http://www.ogf.org/About/abt_overview.php

5.1.4. Globus Toolkit. Según Globus Org⁸⁹, la principal implementación de Open Grid Services Architecture (OGSA), es el Globus Toolkit, una tecnología de código abierto gratuita desarrollada por la Globus Alliance, que proporciona un conjunto de herramientas de programación Java (librerías, servicios y API), para construir fácil y rápidamente sistemas y aplicaciones *grid* basándose en sus servicios y capacidades básicas; tales como la seguridad, la localización y gestión de los recursos y las comunicaciones. Los principales proyectos relacionados con grid, se están desarrollando en base al Globus Toolkit.

El Globus Toolkit ha crecido gracias a la estrategia de código abierto similar a la del sistema operativo Linux, fomentando una adopción más rápida y amplia, que conduce a una mayor innovación tecnológica, como la comunidad de código abierto ofrece mejoras continuas al producto.

5.1.5. Estándares y API's de Grid Computing.

Tabla 3, Estándares y API's de Grid Computing⁹⁰.

Estándar / API	Características	Desarrollado.
Distributed Resource Management Application API (DRMAA)	Especifica la presentación y control de los trabajos a una gestión de recursos distribuidos (DRM), un Cluster o Grid Computing infraestructura	Global Grid Forum
A technology-agnostic information model for a uniform representation of Grid resources (GLUE)	GLUE, estándar para Grid Laboratory Uniform Environment, es un modelo de información independiente de la tecnología para una representación uniforme de los recursos Grid.	Open Grid Forum, GLUE Working Group
Grid Remote Procedure Call (GridRPC)	Este paradigma ha sido propuesto por el grupo de trabajo GridRPC del Open Grid Forum (OGF), es un API definido	Open Grid Forum (OGF)

⁸⁹ Globus Toolkit, Sitio Web [En línea], [Julio 2011]. Disponible en: <http://www.globus.org/toolkit/>

⁹⁰ Grid Café, Sitio Web [En línea], [Julio 2011]. Disponible en http://www.gridcafe.org/que-es-la-computacion-grid_ES.html

	<p>por los clientes para acceder a servidores remotos tan simple como una llamada a función. Se utiliza entre Red numerosas middleware por su sencillez de aplicación, y ha sido estandarizado por la OGF en 2007.</p>	
Grid Security Infrastructure (GSI)	<p>Es una especificación para el secreto a prueba de falsificaciones, entre la comunicación del software delegatable en un entorno de Grid computing. La comunicación es segura gracias al uso de acifrado asimétrico .</p>	
Open Grid Services Architecture (OGSA)	<p>Describe una arquitectura para un entorno de red de computación orientada a servicios para uso comercial y científico, desarrollado en el Global Grid Forum (GGF). OGSA se basa en otras tecnologías de servicios Web, en particular, WSDL y SOAP, sino que quiere ser en gran parte independiente en relación con el manejo a nivel de transporte de datos.</p>	Global Grid Forum (GGF)
Open Grid Services Infrastructure (OGSI)	<p>fue publicado por el Open Grid Forum (OGF) en Junio de 2003, fue pensado para proporcionar una capa de infraestructura para el Open Grid Services Architecture (OGSA)</p>	Open Grid Forum
A Simple API for Grid Applications (SAGA)	<p>Es el nombre de una familia de normas relacionadas especificadas por el Open Grid Forum para definir una interfaz de programación de aplicaciones (API), para la funcionalidad común de computación distribuida.</p>	Open Grid Forum.
Web Services Resource Framework (WSRF)	<p>Es una familia de Organización para el Avance de Estándares de Información Estructurada OASIS especificaciones</p>	Los principales contribuyentes son la Globus Alliance

	publicadas por los servicios web . WSRF proporciona un conjunto de operaciones que pueden implementar los servicios web para convertirse en stateful, los clientes de servicios web pueden comunicarse con los servicios <i>de recursos</i> que permiten que los datos se almacenen y se recuperan.	e IBM.
--	--	--------

Fuente, Elaboración Propia

5.1.6. Los principales Grid middleware de diferentes proveedores todos bajo licencias libres.

Tabla 4, Grid middleware y sus tipo de licencia

Grid middleware	Desarrollado por:	S.O que soporta	Licencia
GLOBUS TOOLKIT	Globus Alliance	Linux, FreeBSD, Solaris, Mac OS X, HP-UX and AIX.	Apache License, Version 2.0 Public License (GTPPL) Version 3
GLITE	EGEE	Scientific Linux	Apache License, Version 2.0
GRIDWAY	dsa-research.org	UNIX-like	Apache license, Version 2.0
ADVANCED RESOURCE CONNECTOR (ARC)	NorduGrid, KnowARC and NDGF	Linux, Microsoft Windows, Mac OS X	Apache License
GRIDSAM	OMII-UK London e-Science Centre	Linux, Windows and Mac OS X	BSD licenses
ORACLE GRID ENGINE	Oracle Corporation	multi-plataforma	SISSL

Fuente, Elaboración Propia

5.2. El uso de Software libre en Cloud Computing (Computación en la nube).

Si se tiene en cuenta el desarrollo de la computación en la nube hasta la fecha, está claro que esta tecnología es el resultado de la convergencia de los diversos estándares.

Según Katarina Stanoevska⁹¹, la escalabilidad de la computación en nube cambio completamente la manera en que se implementan servicios y aplicaciones pero sin estándares, la industria crea sistemas de propiedad con los proveedores de tecnología.

Porque los usuarios no quieren estar encerrados en un solo sistema, hay un impulso fuerte en la industria para crear estándares abiertos basados en la nube. La propuesta de comunidad de la computación en nube es trabajar con estos estándares de arquitectura.

- Plataforma de recursos de virtualización
- Servicios orientados a la Arquitectura SOA
- Aplicaciones Web Frameworks
- Implementación de software de código abierto
- Servicios web estandarizados
- Sistemas autónomos
- Grid Computing

Estos estándares sirven para que los diferentes modelos de negocio puedan ser soportados por los proveedores de cloud computing, sobre todo el software como servicio y las aplicaciones Web 2.0. Las empresas requieren de estándares abiertos para que los datos sean portables y accesibles a todos.

5.2.1. Organizaciones que apoyan los estándares abiertos en cloud computing.

⁹¹ Katarina Stanoevska Thomas Wozniak, Santi Ristol "Grid and Cloud Computing A Business Perspective on Technology and Applications" Springer Dec 2010.

5.2.2. Herramientas de Software libre en la capa infraestructura como servicio (IaaS).

5.2.2.1. Openstack. Según Openstack⁹², es una colección de proyectos de software libre liderado por Rackspace Hosting y la NASA, quienes gracias a la colaboración global de desarrolladores, tecnólogos en computación y la combinación de tecnologías y estándares libres logran ofrecer un proyecto de código abierto que ayuda a dirigir a las organizaciones la implementación de nubes públicas y privadas.

El objetivo del proyecto es ofrecer soluciones para todo tipo de nubes que sean fáciles de implementar de una forma escalable y masiva ya sean públicas o privadas en grandes o pequeñas empresas.

OpenStack contiene una colección de proyectos de código abierto que son mantenidos por la comunidad como OpenStack Compute (Nova), OpenStack Object Storage (Swift), y OpenStack Imaging Service (Glance).

Todo el código de OpenStack está disponible libremente bajo licencia Apache 2.0. Cualquier persona puede ejecutar, construir sobre ella, o presentar cambios al proyecto. Se cree firmemente que un modelo de desarrollo abierto es la única manera de fomentar los estándares y normas de la nube, eliminar el miedo a la propia lock-in para los clientes de la nube, y crear un gran ecosistema que se extiende a los proveedores de cloud.

Actualmente hay tres componentes principales de **OpenStack: Compute, Object Storage, and Imaging Service**.

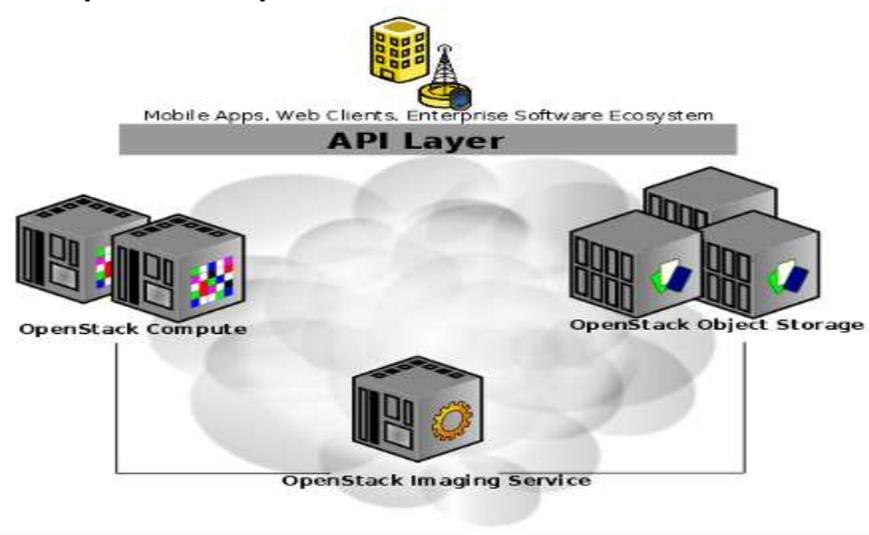
- **Compute: es un controlador de la estructura de la nube, que se utiliza para poner en marcha instancias virtuales, ya sea para un usuario o un grupo. También se utiliza para configurar la red para cada caso o proyecto que contiene varias instancias de un proyecto en particular.**
- OpenStack Object Storage: Es un sistema para almacenar objetos en un sistema de capacidad de escalabilidad masiva de gran tamaño con redundancia a errores.

⁹² Openstack, Sitio Web [En línea], [Agosto 2011]. Disponible en: <http://www.openstack.org/>

- OpenStack Imaging Service: es una búsqueda y recuperación de imágenes de máquinas virtuales. Se puede configurar de tres maneras: utilizando OpenStack tienda de objetos para almacenar las imágenes, el uso de almacenamiento S3 directamente, mediante el almacenamiento S3 con tienda de objetos como medio para acceder a S3.

El siguiente diagrama muestra las relaciones básicas entre los proyectos, cómo se relacionan el uno al otro, y cómo pueden cumplir con las metas de la computación en nube abierta de origen.

Figura 7, “Componentes Openstack”⁹³



Fuente, Openstack.org Disponible en: <http://docs.openstack.org/bexar/openstack-compute/admin/content/ch01s02.html>⁹⁴

OpenStack Compute: OpenStack Compute ofrece una herramienta para orquestar una nube, como las instancias en ejecución, gestión de redes, y controlar el acceso a la nube a través de los usuarios y proyectos. El nombre del proyecto OpenStack subyacente de código abierto es Nova, y proporciona el software que puede controlar una infraestructura como servicio (IaaS) en cloud computing. Se trata de un alcance similar al de Amazon EC2 y servidores de Rackspace Cloud.

⁹³ Openstack, Sitio Web [En línea], [Agosto 2011]. Disponible en: <http://www.openstack.org/>

⁹⁴ Openstack, Sitio Web [En línea], [Agosto 2011]. Disponible en: <http://www.openstack.org/>

OpenStack Compute no incluye ningún software de virtualización, sino que define los conductores que interactúan con los mecanismos subyacentes de virtualización que se ejecutan en el sistema operativo host, y expone la funcionalidad de más de un API basado en la web.

OpenStack Compute utiliza un sistema de gestión de derechos que emplea un papel de control de acceso basado en roles (RBAC) modelo y soporta los siguientes cinco funciones:

1. Cloud Administrador (admin): Los usuarios de esta clase de disfrutar de acceso al sistema completo.
2. IT Security (ITSEC): Esta función se limita al personal de seguridad de TI. Que permite a los titulares papel a los casos de cuarentena.
3. Project Manager (Administrador de proyectos): El valor por defecto para los propietarios del proyecto, esta función permite a los usuarios la posibilidad de añadir otros usuarios a un proyecto, interactuar con las imágenes del proyecto, e iniciar y terminar las instancias.
4. Network Administrator (netadmin): Los usuarios con este papel se les permite destinar y asignar las direcciones IP de acceso público, así como crear y modificar las reglas del cortafuegos.
5. Developer (desarrollador): Esta es una función de propósito general que se asigna a los usuarios por defecto.

Imágenes e instancias Openstack Compute, Una imagen es un archivo que contiene información acerca de un disco virtual que replica por completo toda la información sobre un equipo de trabajo en un momento en el tiempo, incluyendo información del sistema operativo y la información del sistema de archivos. Compute puede utilizar la administración de certificados para descifrar las imágenes incluidas, por ahora se Compute basa en el uso de euca2ools como herramienta de línea de comandos, distribuidas por el equipo de Eucalyptus para la inclusión, agrupación y borrado de imágenes.

Hay dos métodos para la gestión de imágenes, las imágenes pueden ser atendidas a través del Servicio OpenStack Imaging, un proyecto que se llama Glance, o utilizar el servicio de nova-ObjectStore, con el servicio de OpenStack, de servidor en sitio, el servicio de imagen obtiene la imagen en el equipo host y luego OpenStack Compute obtiene la imagen de la máquina host.

Para colocar las imágenes en servicio, se utiliza una interfaz REST stream, que es como una parte de Amazon S3, OpenStack Object Storage (que puede utilizar Amazon S3), o el archivo local en el servidor donde OpenStack Imaging Service está instalado.

Una instancia tiene un ciclo vital que es controlado por OpenStack Compute. Compute crea las instancias y es responsable de la construcción de una imagen en disco, el lanzamiento de ella, la presentación de informes de estado, adjuntando almacenamiento persistente, y darlo por terminado. Los tipos de normas de virtualización que puede ser utilizado con Compute incluyen: KVM, UML, XEN, Hyper-V QEMU.

Requisitos del sistema para OpenStack: Hardware, los componentes de OpenStack están destinados a funcionar en el hardware estándar.

Sistema operativo: OpenStack actualmente funciona en Ubuntu, los despliegues a gran escala en ejecución se ejecutan en Ubuntu 10.04 LTS. Miembros de la comunidad están poniendo a prueba las instalaciones de OpenStack Compute para CentOS y Red Hat Enterprise Linux

Redes: 1000 Mbps, se sugieren. Para OpenStack Compute, la red esté configurada en instalaciones multi-nodo entre las máquinas físicas en una única subred. Para establecer la conexión entre las instancias de máquina virtual, tres opciones de red están disponibles: DHCP plana, y VLAN.

Base de datos: Para OpenStack Compute, se necesita el acceso a cualquiera base de datos PostgreSQL o MySQL, o se puede instalar como parte del proceso de instalación OpenStack Compute.

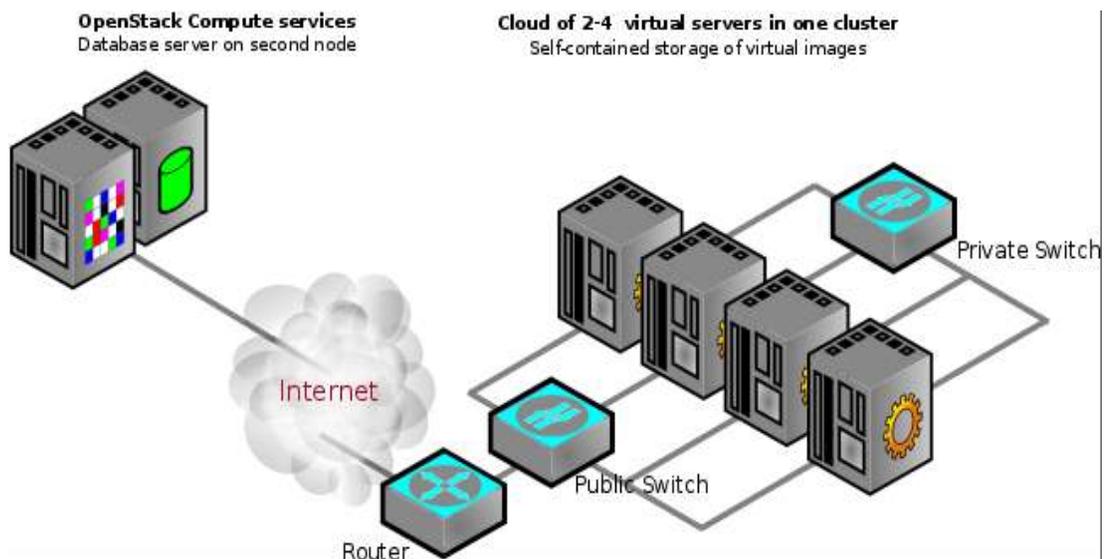
Permisos: Se puede instalar OpenStack Compute bien como root o como usuario con permisos de sudo si se configura el archivo sudoers para permitir todos los permisos.

Ejemplo de arquitectura Openstack: Este ejemplo de instalación a través de una instalación de varios nodos con dos servidores: un controlador de la nube que corre los nova-servicios, y un servidor de Compute que contiene el servidor de base de datos que también corre el nova-servicios.

Esta arquitectura está diseñada para ofrecer alta disponibilidad en la base de datos de lectura / escritura se sospecha que un factor limitante para la disponibilidad permanente de los servidores de la nube.

Esta es una ilustración de una posible instalación de múltiples servidores de OpenStack Compute la creación de redes de servidores virtuales del clúster puede variar.

Figura 8, “Ejemplo de arquitectura Openstack”⁹⁵



Fuente, Openstack.org Disponible en: <http://docs.openstack.org/bexar/openstack-compute/admin/content/ch03s02.html>⁹⁶

5.2.2.2. Open Nebula. Según OpenNebula⁹⁷, es una herramienta completa de código abierto para construir infraestructura como servicio (IaaS) en cualquier tipo de nube: privado, público e híbridas.

La tecnología OpenNebula es el resultado de muchos años de investigación y desarrollo en la gestión eficiente y escalable de máquinas virtuales en las grandes

⁹⁵ Openstack, Sitio Web [En línea], [Agosto 2011]. Disponible en: <http://www.openstack.org/>

⁹⁶ Openstack, Sitio Web [En línea], [Agosto 2011]. Disponible en: <http://www.openstack.org/>

⁹⁷ OpenNebula Sitio Web [En línea], [Agosto 2011]. Disponible en: [http://www. http://opennebula.org/](http://www.opennebula.org/)

infraestructuras distribuidas. Sus características innovadoras han sido desarrolladas para satisfacer los requisitos de casos de uso empresarial de las principales empresas de TI y de múltiples sectores en el contexto de proyectos innovadores en computación en la nube.

OpenNebula fue diseñada para satisfacer los requisitos de casos de uso empresarial de las principales empresas y en múltiples industrias, tales como Hosting, Telecom, la administración electrónica, servicios de cómputo, etc. Los principios que han guiado el diseño de OpenNebula son los siguientes:

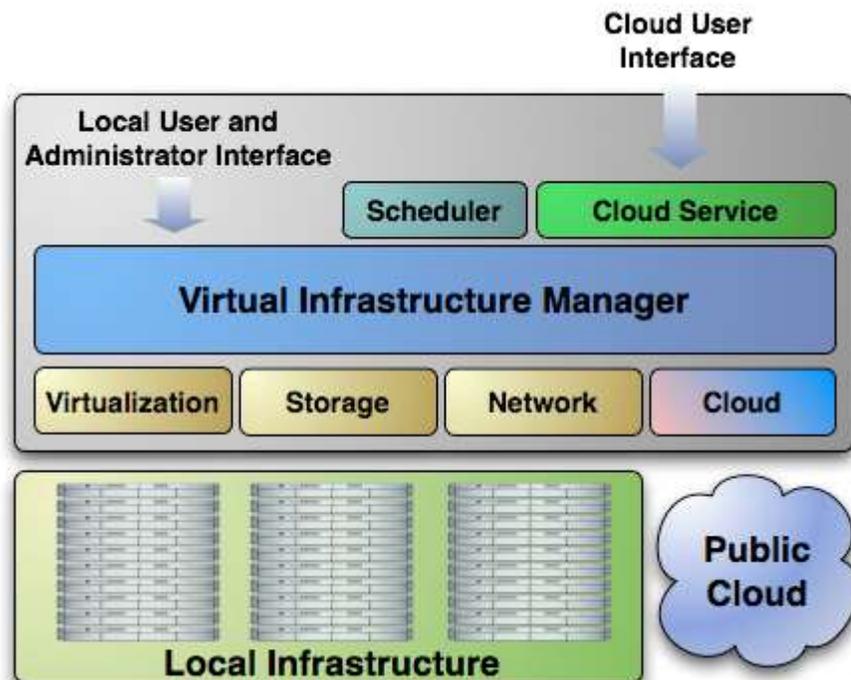
- La apertura de la arquitectura, interfaces, y el código
- Capacidad de adaptación para gestionar cualquier combinación de hardware y software, y para integrarse con cualquier producto y servicio en el ecosistema de la nube y la virtualización.
- La interoperabilidad y la portabilidad para evitar los proveedores de tecnología.
- Estabilidad para su uso en la producción de los ambientes de clase empresarial.
- Escalabilidad de las infraestructuras a gran escala.
- Estandarización mediante el aprovechamiento y aplicación de las normas.

OpenNebula es totalmente de código abierto, el software es distribuido y autorizado bajo los términos de Apache License, Version 2.0. Copyright 2002-2011 El proyecto de código abierto OpenNebula (OpenNebula.org), está a cargo de los laboratorios de C12G. [10]

Funcionamiento de OpenNebula. OpenNebula orquesta el almacenamiento, la red, la virtualización, monitorización, y tecnologías de seguridad para permitir la colocación dinámica de múltiples niveles de servicios (grupos de interconexión de máquinas virtuales) sobre las infraestructuras de distribución, que combina los recursos del centro de datos y recursos remotos de nubes, de acuerdo con las políticas de asignación.

Figura 9 “Funcionamiento OpenNebula”⁹⁸

⁹⁸ OpenNebula Sitio Web [En línea], [Agosto 2011]. Disponible en: <http://www.opennebula.org/>



Fuente, opennebula.org Disponible en: <http://opennebula.org/about:technology>⁹⁹

Arquitectura Interna de OpenNebula. La arquitectura interna de OpenNebula se divide en tres capas:

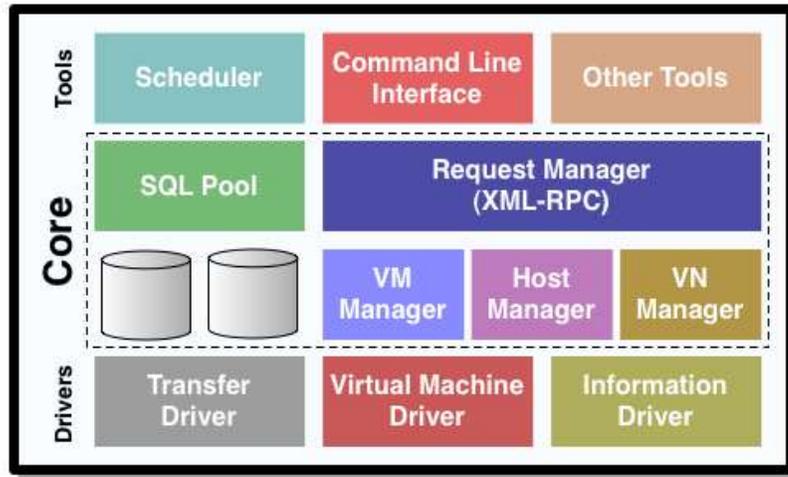
- Tools: Esta capa contiene las herramientas que se distribuyen con OpenNebula, como el CLI, el planificador, el API libvirt, así como las demás herramientas desarrolladas por terceras partes que emplean el interfaz XML-RPC de OpenNebula:
 - Command Line Interface: Permite a los usuarios y administradores de OpenNebula gestionar de forma manual la infraestructura virtual.
 - Scheduler: Es una entidad independiente en la arquitectura y puede desacoplarse del resto de los componentes, emplea la interfaz XML-RPC para invocar las acciones que se efectuarán en las máquinas virtuales. [14]
- Core: Es un conjunto de componentes por medio de los cuales se gestiona y se monitorea las máquinas virtuales, las redes virtuales, el almacenamiento y nodos. Los componentes principales del núcleo son:
 - Request Manager: Gestiona las peticiones de los clientes.
 - Virtual Machine Manager: Gestiona y monitorea las máquinas virtuales
 - Transfer Manager: Gestiona la transferencia de imágenes

⁹⁹ OpenNebula Sitio Web [En línea], [Agosto 2011]. Disponible en: <http://www.opennebula.org/>

- Virtual Network Manager: Gestiona las redes virtuales
- Host Manager: Gestiona y monitoriza los recursos físicos
- Database: Base de datos persistente que almacena los datos de OpenNebula. Soporta SQLite3 y MySQL.
- Drivers: Proporcionan nuevas tecnologías para la virtualización, el almacenamiento, la monitorización y los servicios de cloud. Interactúan con middleware específico (hipervisor, servicios de cloud, transferencia de imágenes, o servicios de información).

Figura 10 “Arquitectura Interna de OpenNebula”¹⁰⁰

¹⁰⁰ OpenNebula Sitio Web [En línea], [Agosto 2011]. Disponible en: <http://www.opennebula.org/>



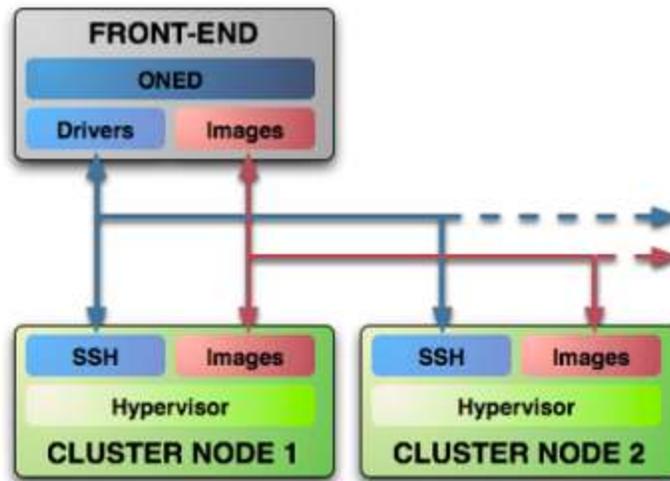
Fuente, opennebula.org Disponible en: <http://opennebula.org/about:technology>¹⁰¹

Infraestructura de OpenNebula. OpenNebula emplea en su infraestructura una arquitectura de cluster clásica, con un frontal y con conjunto de nodos donde serán ejecutadas las máquinas virtuales, al menos, debe haber una red que interconecte todos los nodos con el frontal.

Figura 11 “Infraestructura de OpenNebula”¹⁰²

¹⁰¹ OpenNebula Sitio Web [En línea], [Agosto 2011]. Disponible en: <http://www.opennebula.org/>

¹⁰² OpenNebula Sitio Web [En línea], [Agosto 2011]. Disponible en: <http://www.opennebula.org/>



Fuente, opennebula.org Disponible en: <http://opennebula.org/about:technology>¹⁰³

Los componentes de la Infraestructura de OpenNebula son:

- Front-end: Ejecuta OpenNebula y los servicios del cluster
- Nodos: Anfitriones que proporcionan los recursos necesarios a las máquinas virtuales, como el software hipervisor
- Repositorio de imágenes: Medio que almacena las imágenes virtuales.
- Demonio OpenNebula: Gestiona el ciclo de vida de las máquinas virtuales y los subsistemas (red, almacenamiento, hipervisores)
- Drivers: Programas empleados por el núcleo de OpenNebula para servir de interfaz para un hipervisor o un sistema de almacenamiento específicos.

5.2.2.3. Eucalyptus. Según Eucalyptus¹⁰⁴, es el acrónimo (Elastic utility Computing for Linking your Programs to Useful System) en español "Utilidad de arquitectura informática elástica para confiar sus programas a sistemas funcionales" desarrollada como software libre, bajo licencia BSD y GPL v3, sobre Linux que implementa nubes privadas e híbridas dentro de la infraestructura de TI de una

¹⁰³ OpenNebula Sitio Web [En línea], [Agosto 2011]. Disponible en: <http://www.opennebula.org/>

¹⁰⁴ Eucalyptus System Inc. Sobre la Infraestructura de código abierto de nube, Disponible <http://open.eucalyptus.com/>, [Citado 10 de Diciembre de 2010]

organización, inicialmente fue diseñado y desarrollado por el equipo del profesor Rich Wolski, como un proyecto del Computer Science Department de la Universidad de California.

Eucalyptus proporciona una capa de red virtual de tal forma que se aísla del tráfico de red de diferentes usuarios y permite que uno o más clusters pertenezcan a la misma LAN (sólo Enterprise Edition). Además, tiene la capacidad de interactuar con Amazon EC2 y los servicios S3 de la nube pública de Amazon, ofreciendo la posibilidad de crear una nube híbrida.

Las funcionalidades principales de la plataforma cloud de Eucalyptus son:

- API compatible con Amazon Web Services (AWS)
- Arquitectura agnóstica en relación al hipervisor (soporta Xen y KVM)
- Gestor Walrus: Gestiona el almacenamiento de las imágenes de los sistemas de archivos de las máquinas virtuales, es compatible con S3 de Amazon.
- Soporte EBS (Elastic Block Store) sobre AoE e iSCSI, múltiples modos de funcionamiento de red para adaptarse a diferentes arquitecturas
- Interfaz Web y herramientas CLI para la administración y la configuración del cloud
Arquitectura escalable: las peticiones del cloud se sirven de forma asíncrona.

Arquitectura de Eucalyptus. La arquitectura de Eucalyptus es simple, flexible y modular con diseño jerárquico que consta de cinco componentes basados en Servicios Web.

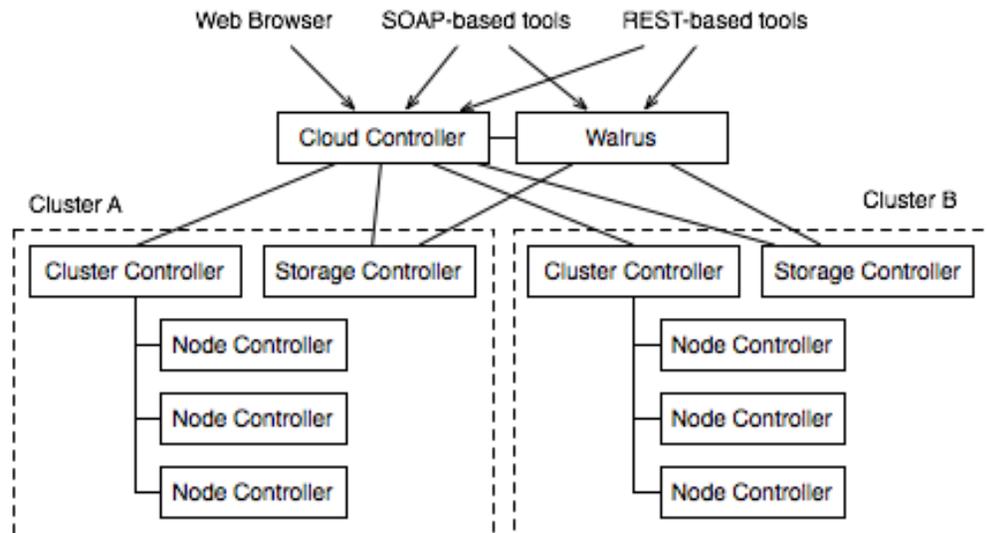
- Cloud Controller (CLC)
- Walrus Storage Controller (WS3)
- Elastic Block Storage Controller (EBS)
- Cluster Controller (CC)
- Node Controller (NC)

El cloud controller (CLC) y Walrus son componentes de alto nivel. En una instalación de cloud debe haber ambos. El cloud controller es un programa en Java que ofrece interfaces SOAP compatibles con Amazon Elastic Compute Cloud (EC2), así como una interfaz Web para todo el mundo. Además de gestionar las peticiones entrantes, el cloud controller aporta un planificador de recursos de alto nivel y un sistema de contabilidad. Walrus, también está escrito en Java, implementa almacenamiento bucket-based, el cual estaría disponible fuera y dentro del cloud mediante interfaces REST y SOAP compatibles con S3.

Los componentes de alto nivel pueden agregar recursos de múltiples clusters (conjuntos de nodos compartiendo un segmento de LAN, posiblemente ubicados detrás de un firewall). Cada cluster necesita un cluster controller (CC) para efectuar las tareas de planificación y control de la red, así como un storage controller (SC), para el almacenamiento basado en bloques EBS. Los componentes de segundo nivel deberían ser desplegados en el nodo cabecera de cada cluster. Finalmente, cada nodo necesitará un node controller (NC) para controlar el hipervisor que se encuentra instalado en el mismo. El CC y el NC están escritos en C y se despliegan como servicios Web dentro de Apache. El SC está escrito en Java. La comunicación entre esos componentes tiene lugar por medio de SOAP con seguridad WS.

Figura 12 “Descripción detallada de cada uno de los componentes de Eucalyptus”.¹⁰⁵

¹⁰⁵ Eucalyptus System Inc. Sobre la Infraestructura de código abierto de nube, Disponible <http://open.eucalyptus.com/>, [Citado 10 de Diciembre de 2010]



Fuente, open.eucalyptus.com Disponible en: <http://open.eucalyptus.com/>¹⁰⁶

Cloud controller (CLC): Proporciona una interfaz compatible con los servicios Web EC2/S3, además, proporciona una interfaz Web para los usuarios que permite gestionar ciertos aspectos de la infraestructura

- **Funciones:** Monitorea la disponibilidad de recursos en varios componentes de la infraestructura, incluyendo los nodos y los Cluster Controllers, efectúa el arbitraje de recursos: decide qué clusters se emplearán para provisionar las instancias, monitorea las instancias en ejecución.

Walrus: Implementa el sistema de almacenamiento persistente empleando APIs, REST y SOAP compatibles con Amazon S3

- **Funciones:** Almacena las imágenes maestras de las máquinas virtuales, almacena snapshots, almacena y sirve los archivos empleando el API de Amazon S3

Cluster controller (CC): Efectúa las tareas de planificación y control de la red a nivel de cluster.

Storage controller (SC): Gestiona el almacenamiento EBS a nivel de cluster

- **Funciones:** Gestiona la creación de los dispositivos EBS, proporciona a las instancias el acceso al almacenamiento por bloques mediante AoE o iSCSI, permite la creación de snapshots de los volúmenes.

¹⁰⁶ Eucalyptus System Inc. Sobre la Infraestructura de código abierto de nube, Disponible <http://open.eucalyptus.com/>, [Citado 10 de Diciembre de 2010]

Node controller: Se ejecuta en cada uno de los nodos y gestiona el ciclo de vida de las instancias, Interactúa con el hipervisor y con el Cluster Controller.

- Funciones: Gestiona el ciclo de vida de las instancias, recolecta los datos relacionados con la disponibilidad de los recursos e informa al Cluster Controller.

5.2.2.4. Nimbus: Según Nimbus¹⁰⁷, Nimbus es un conjunto de herramientas de código abierto, que en conjunto proporcionan una solución en la capa de Infraestructura como servicio (IaaS) en cloud computing.

Desarrollada y soportada por la Universidad de Chicago con el apoyo de la comunidad de scienceclouds, proporciona el conjunto de herramientas para el desligue de nubes privadas o comunitarias especialmente en entornos de ciencia, liberado bajo Licencia Apache 2.0.

Características de Nimbus, servicio de almacenamiento en la nube

Cumulus es el servicio de almacenamiento en la nube que es compatible con el API de Amazon S3.

La implementación y gestión remota de ciclo de vida de las máquinas virtuales, mediante el cliente remoto de Nimbus se pueden implementar, poner en pausa, reiniciar y apagar las máquinas virtuales.

Compatibilidad con los protocolos de red de Amazon y las API's SOAP y REST también se han implementado en Nimbus.

Soporte X509 Credenciales.

Los usuarios acceden a través de una interfase web que utiliza certificados X509, con el modelo de seguridad de PKI fuerte y soporte de certificados DOE o cualquier otra autoridad.

Nimbus se puede implementar en cualquier distribución Linux que cuenta con Java, Python, y bash en servidor y en el hipervisor nodos cuando con Python, bash, ebtabs, libvirt y KVM o Xen.

Principales componentes de Nimbus:

¹⁰⁷ Nimbus Sitio Web [En línea], [Noviembre 2011]. Disponible en: <http://www.nimbusproject.org/>

- El Workspace Service administrador del sitio
- Protocolo remoto WSRF
- Un EC2 la implementación del protocolo remoto SOAP y Query APIs
- Cumulus implementación de código abierto del API REST de Amazon S3. Se utiliza como depósito de la solución de Nimbus y también se puede instalar independiente.
- RM API puente entre los protocolos de control remoto / seguridad y implementaciones específicas de administrador del sitio.
- Cloud client tiene como objetivo conseguir que los usuarios y en cuestión de minutos con los lanzamientos de la instancia y grupos con un solo clic.
- reference client expone la característica de todo el conjunto en el protocolo WSRF como un cliente de línea de comandos (con la biblioteca cliente Java subyacente).
- Workspace Pilot permite integrar máquinas virtuales con recursos que ya se ha configurado para gestionar puestos de trabajo (es decir, ya está usando un programador batch como PBS).
- workspace-control las tareas de agente implementa VMM y de la red específica de cada hipervisor.
- Context Broker permite a los clientes coordinar cluster virtual se inicia automáticamente y repetible.
- Context Agent vida de las máquinas virtuales e interactúa con el Broker Contexto en el arranque del VM.

5.2.2.5. Ubuntu Enterprise Cloud Architecture (UEC). Según Canonical¹⁰⁸, Ubuntu Enterprise Cloud es un stack de aplicaciones de Canonical incluidas dentro de Ubuntu Server Edition.

Ubuntu es la única distribución Linux que puede posicionarse como un verdadero sistema operativo cloud, con tres componentes en la nube ya liberados. Dos de estos componentes están dirigidos a la infraestructura informática (IaaS o comúnmente llamada Infraestructura como Servicio), mientras que un componente está dirigido a la capa de software (comúnmente llamado SaaS).

¹⁰⁸ Simon Wardley, Etienne Goyer & Nick Barcet Ubuntu Enterprise Cloud Architecture, Agosto 2009, Disponible en: <http://www.canonical.com/about-canonical/resources/white-papers/ubuntu-enterprise-cloud-architecture> [Citado 10 de Enero de 2011]

Los tres componentes son:

- Ubuntu Server Edition en Amazon EC2 (IaaS)
- Ubuntu Enterprise Cloud impulsado por el Eucalypto (IaaS)
- UbuntuOne (SaaS)

Ubuntu Enterprise Cloud (UEC) brinda las capacidades de infraestructura dentro de la Firewall como las Amazon EC2. UEC se potencia de Eucalyptus, implementación de código abierto que utiliza el emergente de estándar de la API de EC2. Esta solución está diseñada para simplificar el proceso de construcción y la gestión de una nube interna para empresas de cualquier tamaño, lo que permite a las empresas crear sus propios servicios de infraestructura.

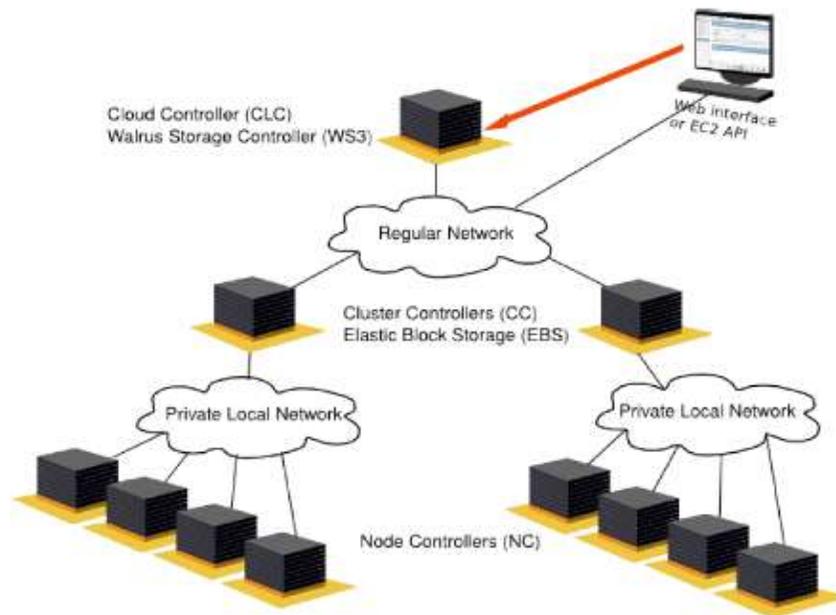
Elementos Ubuntu Enterprise Cloud Architecture (UEC). El componente principal de Ubuntu Enterprise Cloud es la arquitectura de Eucalyptus (descrita anteriormente) esta ha sido diseñado como sistema modular de 5 elementos simples que se pueden ampliar fácilmente:

- Cloud Controller (CTC)
- Walrus Controlador de almacenamiento (WS3)
- Elástica Bloque Controlador de almacenamiento (EBS)
- Cluster Controller (CC)
- Nodo de Control (NC)

Cada elemento está actuando como un servicio web independiente que expone Descripción del servicio Web Language (WSDL) define la API para interactuar con él. Es la arquitectura típico servicio web.

Figura 13 “Elementos de (UEC)”¹⁰⁹.

¹⁰⁹ Simon Wardley, Etienne Goyer & Nick Barcet Ubuntu Enterprise Cloud Architecture, Agosto 2009, Disponible en: <http://www.canonical.com/about-canonical/resources/white-papers/ubuntu-enterprise-cloud-architecture> [Citado 10 de Enero de 2011]



Fuente, Ubuntu Enterprise Cloud Disponible en: <http://www.ubuntu.com/cloud>¹¹⁰

5.2.2.6. Cuadro Comparativo Proveedores de IaaS, basados en Software Libre

¹¹⁰ Simon Wardley, Etienne Goyer & Nick Barcet Ubuntu Enterprise Cloud Architecture, Agosto 2009, Disponible en: <http://www.canonical.com/about-canonical/resources/white-papers/ubuntu-enterprise-cloud-architecture> [Citado 10 de Enero de 2011]

Tabla 5, Comparación de Proveedores de IaaS basados en Software Libre

Características	OpenStack	OpenNebula	Eucalyptus	Nimbus
S.O. Compatibles	S.O, Linux	Open Solaris, Open Suse, Debian, Ubuntu	S.O, Linux	S.O, Linux
Desarrollada en	Python	Java, Perl, PHP	Java, C	Java, Python
virtualización	Xen, KVM	Xen, KVM, Vmware,	Xen, KVM, VMware	Xen, KVM
Seguridad	Asignación de Credenciales de autenticación, rol a los usuarios.	Firewall, Red privada virtual.	Autenticación, y control de la nube por llave publica/privada Credenciales X509	Soporta Credenciales X509
Interface de Administración	Panel de Control, Aplicación Web.	Panel de Control, Aplicación Web y API	Panel de Control, Aplicación Web y Herramientas de gestión remota (euca2)	Interface de EC2, (línea de comandos.)
Tipos de Nubes	Publica, Privada, híbrida.	Privada, híbrida, compatibilidad con API de los servicios Web de Amazon.	Privadas, compatibilidad con API de los servicios Web de Amazon.	Privada y Comunitaria, compatibilidad con Amazon EC2.
Tipo de Licencia	Apache License	Apache 2.0 License	Propietaria, GPL v3	Apache 2.0 License
Creado Por	Rackspace y NASA.	C12G labs y dsa-research.org	Eucalyptus Systems, Inc	University of Chicago y ScienceCloud.org
Sitio Web	http://openstack.org	http://www.opennebula.org	http://open.eucalyptus.com	http://www.nimbusproject.org

Fuente, Elaboración Propia

5.2.3. Herramientas de Software libre en la capa de plataforma como servicio (PaaS). La mayoría de las herramientas para el desarrollo de aplicación en la capa de plataforma como servicio (PaaS), tienen su propio marco de trabajo.

Muchos de estas herramientas se basan en entornos visuales, y a menudo estas herramientas permiten a los desarrolladores ampliar las aplicaciones en un lenguaje común para el desarrollo de aplicaciones Web. Estas aplicaciones casi siempre adoptan un modelo de arquitectura orientada a servicios y el uso de SOAP / REST, con intercambio de datos XML.

5.2.3.1. Google AppEngine. Google es una de las más grandes y principales empresas promotoras de nuevas tecnologías en el mundo y dentro de su portafolio de productos y servicios se encuentra Google appEngine, que es su propuesta de plataforma como servicio en la nube, permitiendo el despliegue de aplicaciones web robustas dentro de su infraestructura tecnológica la cual se encuentra distribuida físicamente por todo el mundo.

A pesar de que Google pone a disposición su infraestructura para el despliegue de aplicaciones basada en la filosofía de Software Libre apoyando a la comunidad de desarrolladores, no se especifican un tipo de licencia, sin embargo la propuesta de Google esta orientada al desarrollo de aplicaciones en lenguajes JAVA y Python siendo ambos entornos de desarrollo libres y distribuyendo los ejemplos de código bajo la licencia Apache 2.0 y el contenido del sitio Web licenciado bajo la licencia Creative Commons Attribution 3.0.

Funciones de Google App Engine, Según Google¹¹¹:

- Servidor web dinámico, totalmente compatible con las tecnologías web más comunes,
- Almacenamiento permanente con funciones de consulta, clasificación y transacciones,
- Escalado automático y distribución de carga,
- API para autenticar usuarios y enviar correo electrónico a través de Google Accounts,
- Un completo entorno de desarrollo local que simula Google App Engine en tu equipo,
- Colas de tareas que realizan trabajos fuera del ámbito de una solicitud web,

¹¹¹ Google App Engine Sitio Web [En línea], [Noviembre 2011]. Disponible en: <http://code.google.com/intl/es-ES/appengine/>

- Tareas programadas para activar eventos en momentos determinados y en intervalos regulares.
- **Entornos de tiempo de ejecución. Google App Engine proporciona dos entornos en tiempo de ejecución uno en entorno Java y otro en entorno Python. Cada uno de ellos proporciona protocolos estándar y tecnologías comunes para el desarrollo de aplicaciones web.**

- El entorno de tiempo de ejecución Java. A través de herramientas de desarrollo web Java y de estándares y API's conocidos, las aplicaciones interactúan con el entorno a través del estándar Java Servlet y puede utilizar tecnologías de aplicaciones web conocidas como, por ejemplo, JavaServer Pages (JSP). El entorno de tiempo de ejecución Java utiliza Java 6. El SDK Java de App Engine permite desarrollar aplicaciones que utilicen tanto Java 5 como 6.

Las aplicaciones tienen acceso a la mayoría de los servicios de App Engine a través del API estándar Java. Para el almacén de datos de App Engine, el SDK Java incluye implementaciones de la interfaz de Objetos de datos Java (JDO) y de la interfaz del API de persistencia Java (JPA). Las aplicaciones pueden utilizar el API JavaMail para enviar mensajes de correo electrónico con el servicio de correo electrónico de App Engine. Las API HTTP, `java.net` acceden al servicio de extracción de URL de App Engine. App Engine también incluye las API de nivel inferior para sus servicios a fin de implementar adaptadores adicionales o para su uso directo desde la aplicación. Normalmente, los desarrolladores de Java utilizan el lenguaje de programación Java y las API para implementar aplicaciones web para JVM. Gracias al uso de intérpretes o de compiladores compatibles con JVM, también puedes utilizar otros lenguajes para desarrollar aplicaciones web como, por ejemplo, JavaScript, Ruby o Scala.

- El entorno de tiempo de ejecución Python. App Engine incluye varias API y herramientas para el desarrollo de aplicaciones web Python, así como un API de modelado de datos detallados, un marco de aplicaciones web fáciles de utilizar, y herramientas para administrar los datos y acceso a ellos. También cuenta de una amplia variedad de marcos y bibliotecas avanzados para el desarrollo de aplicaciones web Python como, por ejemplo, Django.

El entorno de tiempo de ejecución Python utiliza la versión 2.5.2. de Python, e incluye la biblioteca estándar Python.

El entorno Python proporciona varias API Python para servicios de almacén de datos, Google Accounts, extracción de URL y correo electrónico. App Engine también ofrece un sencillo marco para aplicaciones web Python denominado webapp que te permitirá empezar a crear aplicaciones fácilmente.

- **Servicios de Google App Engine. - El almacén de datos: App Engine proporciona un potente servicio de almacenamiento de datos distribuido que incluye un motor de búsqueda y transacciones. A medida que el servidor web distribuido crece con el tráfico, el almacén de datos distribuido crece con los datos.**

El almacén de datos de App Engine no es como una base de datos relacional tradicional, los objetos de datos, o "entidades", disponen de un tipo y un conjunto de propiedades.

- Google Accounts: App Engine admite la integración de una aplicación con Google Accounts para la autenticación de los usuarios, la aplicación puede permitir a un usuario acceder con una cuenta de Google y tener acceso a la dirección de correo electrónico y al nombre de visualización asociados a la cuenta. Google Accounts permite que el usuario pueda empezar a utilizar la aplicación de una forma más rápida, ya que no tiene que crear una cuenta nueva. También le ahorra el esfuerzo de implementar un sistema de cuentas de usuario solo para una aplicación.

El API de usuarios también puede indicar a la aplicación si el usuario actual es un administrador registrado de la aplicación.

- Extracción de URL: Mediante el servicio de extracción de URL de App Engine, las aplicaciones pueden acceder a recursos en Internet, como servicios web u otros datos, este servicio recupera recursos web mediante la misma infraestructura de alta velocidad de Google que obtiene páginas web para muchos otros productos de Google.

- Correo: Las aplicaciones pueden enviar mensajes de correo electrónico mediante el servicio de correo de App Engine. Este servicio utiliza la infraestructura de Google para enviar mensajes de correo electrónico.

- Memcache: Memcache proporciona a la aplicación el servicio de memoria caché de valores claves de alto rendimiento accesible desde varias instancias de la aplicación. Memcache resulta útil para los datos que no necesitan las funciones

de persistencia y transacciones del almacén de datos, como los datos temporales o los datos copiados del almacén de datos en la caché para un acceso a gran velocidad.

- Manipulación de imágenes: El servicio de imágenes permite a tu aplicación manipular imágenes. Con esta API, podrás recortar, girar, dar la vuelta o ajustar el tamaño de imágenes en formato JPEG PNG.

• **Kit de desarrollo de software de aplicaciones App Engine (SDK).** **Apaesar de que los servicios de Google App Engine están enfocados correr aplicaciones en la infraestructura de Google, App Engine suministra un kit de desarrollo de software de App Engine (SDK) para Java y Python, el cual emula todos los servicios de App Engine de un equipo local.**

Cada SDK incluye todas las API y bibliotecas disponibles en App Engine, e incluye una herramienta para subir las aplicaciones directamente a Google App Engine. El SDK de Java se ejecuta en cualquier sistema operativo con Java 5 o Java 6, el SDK está disponible en formato de archivo Zip. Si se utiliza el entorno de desarrollo Eclipse se puede usar el plug-in de Google para Eclipse, para crear, probar y subir aplicaciones App Engine.

El SDK de Python se implementa exclusivamente en Python y se ejecuta en cualquier plataforma que disponga de Python 2.5, como Windows, Mac OS X y Linux., el SDK está disponible en formato de archivo Zip y los instaladores están disponibles para Windows y Mac OS X.

El Kit de desarrollo SDK de AppEngine se distribuye bajo Licencia Apache 2.0

Figura 14 Página principal Google App Engine en español¹¹².



Fuente, Google App Engine Disponible en: <http://code.google.com/intl/es-ES/appengine/>¹¹³

5.2.3.2. Appscale. Según App scale¹¹⁴, AppScale es un framework que permite correr aplicaciones de Google AppEngine, implementada en la capa de plataforma como servicio (paaS), es de código abierto soportado por licencia BSD para sistemas operativos Linux, desarrollada y es mantenida por el RACELab en la Universidad de Santa Barbara.

AppScale implementa una serie de APIs populares como los de Google App Engine, MapReduce (a través de Hadoop), MPI, entre otros para desarrollar aplicaciones escritas en Python, Java y Go.

¹¹² Google App Engine Sitio Web [En línea], [Noviembre 2011]. Disponible en: <http://code.google.com/intl/es-ES/appengine/>

¹¹³ Google App Engine Sitio Web [En línea], [Noviembre 2011]. Disponible en: <http://code.google.com/intl/es-ES/appengine/>

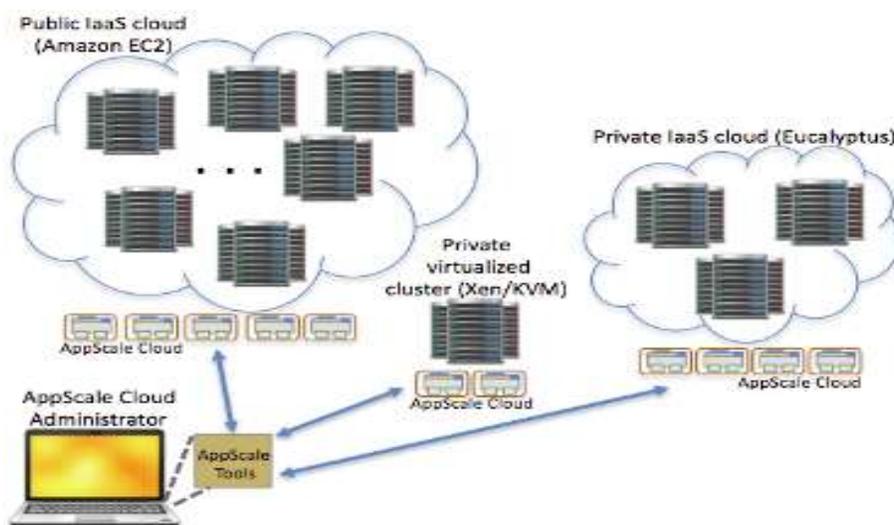
¹¹⁴ App Scale Sitio Web [En línea], [Noviembre 2011]. Disponible en: <http://appscale.cs.ucsb.edu/>

AppScale se ejecuta como una máquina virtual invitada sobre cualquier capa de virtualización que puede alojar una imagen Ubuntu Lucid (de Canonical). En este momento de prueba y apoyo a la hypervisor Xen y KVM.

AppScale también despliega automáticamente en la nube pública de Amazon EC2 y nubes privadas con Eucalyptus.

La última versión estable es AppScale 1.5, lanzado 28 de julio 2011.

Figura 15, Ejemplos de despliegue de AppScale¹¹⁵



Fuente, App Scale, Disponible en:

http://code.google.com/p/appscale/wiki/Deploy_AppScale¹¹⁶

- **Despliegues de nubes en AppScale.** Ya sea en nubes públicas o privadas, AppScale proporciona el software para desplegar aplicaciones potentes y escalables, AppScale también se puede instalar directamente en el hardware físico o el uso de la capa de virtualización favorita.

¹¹⁵ App Scale Sitio Web [En línea], [Noviembre 2011]. Disponible en: http://code.google.com/p/appscale/wiki/Deploy_AppScale

¹¹⁶ App Scale Sitio Web [En línea], [Noviembre 2011]. Disponible en: http://code.google.com/p/appscale/wiki/Deploy_AppScale

- Implementación en clústers virtualizados: El administrador de la nube AppScale copia las imágenes y crea una instancia AppScale manualmente en cada máquina del clúster virtual, una vez que las imágenes se han iniciado y tienen una dirección IP pública, el administrador utiliza las herramientas AppScale para invocar los componentes AppScale todo el sistema.
- Implementación en infraestructura en la nube: el administrador de la nube AppScale, carga los paquetes y registra la imagen AppScale en Eucalyptus o Amazon EC2 en la Infraestructura como servicio (IaaS) en la nube. Para Amazon EC2, el administrador alternativamente puede evitar este paso y el uso la imagen pública (AMI AppScale) que se suministra.

Una nube AppScale consiste en una o más instancias AppScale. Una instancia AppScale es una máquina virtual (VM) que se ejecuta sobre una capa de virtualización, la instancia implementa una o más de los componentes AppScale (roles aka) en tiempo de ejecución que interactúa con los componentes AppScale / roles de otras instancias.

Una máquina virtual es AppScale se crea con una imagen de disco AppScale junto con el kernel del sistema operativo correspondiente ramdisk. La imagen de disco AppScale consiste en una distribución del sistema operativo Ubuntu Karmic y el código del programa (código fuente y binario) para todos los componentes AppScale / roles, así como las herramientas AppScale. Una imagen AppScale contiene el código para todos los componentes / funciones de manera que una instancia puede interpretar cualquier papel en cualquier momento (para tolerancia a fallos y efectos de elasticidad). Una instancia utiliza las herramientas AppScale para controlar y gestionar otras instancias AppScale.

5.2.3.3. WSO2 Stratos. Según Stratos¹¹⁷, Wso2 Stratos es un conjunto de aplicaciones Middleware basada en WSO2 Carbon, el cual permite desplegar toda una suite de servicios en la nube.

Desarrollado por la empresa de Software Libre WSO2 de Sri Lanka, enfocada en proveer soluciones basadas en arquitecturas orientada al servicio (SOA), fundada en 2005, por el Dr.Sanjiva Weerawarana.

¹¹⁷ Wso2 Startos Sitio Web [En línea], [Noviembre 2011]. Disponible en: <http://wso2.com/cloud/stratos/>

WSO2 ofrece una pila de productos, bajo los estándares de la OSGi, desarrollando productos 100% de código abierto, dentro de los cuales se destaca WSO2 carbon como un middleware que facilita la implementación de arquitectura orientada al servicio (SOA) con servicios Web y aplicaciones alojadas en la Web, integrando la mediación, la orquestación de servicios, registros de servicios, gestión de identidades, seguridad.

Todos estos productos están ahora disponibles como plataforma como servicio (PaaS), bajo el nombre WSO2 Stratos, que es la primera plataforma completa de middleware para implementar una nube de servicios empresariales como arquitectura orientada al servicio SOA.

A diferencia de la mayoría de ofertas de plataforma como servicio (PaaS), que se centran en el desarrollo de nuevas aplicaciones en la nube, Stratos admite el modelo tradicional de programación al cual los usuarios ya están familiarizados y desplegar todo un conjunto de productos implementados mediante la arquitectura orientada al servicio (SOA).

WSO2 Stratos middleware como servicios en la nube incluye:

- WSO2 Enterprise Service Bus como servicio. Optimizado para trabajar con XML un servicio basado en Web, la ESB WSO2 también soporta colas de mensajes, formatos fijos y variables, intercambio electrónico de datos (IDE).
- WSO2 Application Server. WSO2 servidor de aplicaciones Web, se trata de una plataforma ligera y de alto rendimiento para Arquitecturas Orientadas a Servicios, lo que permite la lógica de negocio y aplicaciones. Que reúne una serie de proyectos de servicios Web de Apache, WSO2 Application Server proporciona un entorno de ejecución seguro, transaccional y fiable para el despliegue y la gestión de los servicios Web.
- WSO2 Business Process Server. WSO2 Servidor de Procesos de Negocio (BPS) es fácil de utilizar el servidor de procesos de negocio que ejecuta procesos de negocio por escrito WS-BPEL estándar. WS-BPEL se está convirtiendo en el estándar de facto para la composición de múltiples servicios web sincrónicos y asincrónicos en los flujos de proceso de colaboración y de transacciones que aumentan la flexibilidad y la agilidad de su arquitectura orientada a servicios. BPS WSO2 es impulsado por Apache ODE (Apache Director del motor de orquestación).

BPS WSO2 ofrece una web completa basada en consola gráfica para desplegar, gestionar y controlar procesos de negocio y las instancias de proceso.

- WSO2 Mashup Server, WSO2 Governance Registry, WSO2 Data Services Server, WSO2 Business Activity Monitor, WSO2 Gadget Server, WSO2 Business Rules Server WSO2
- Web Service Framework: WSO2 Web Services Framework es un framework de servicios web, basado en Apache Axis2 / C y varios paquetes relacionados. Tiene enlaces a varios idiomas, incluyendo C, PHP, Java, Ruby.

Figura 16, Sitio Web oficial de WSO2 Stratos¹¹⁸

¹¹⁸ Wso2 Startos Sitio Web [En línea], [Noviembre 2011]. Disponible en: <http://wso2.com/cloud/stratos/>



Fuente, Wso2 Startos, Disponible en: <http://wso2.com/cloud/stratos/>¹¹⁹

5.2.3.4. Cloud Foundry. Según Cloud Foundry¹²⁰, es una propuesta de plataforma como servicio (PaaS) desarrollada por VMware con apoyo de la industria, libreada a la comunidad de software libre bajo los términos de licencia de Apache 2.0 como versión Beta.

Cloud Foundry está basado en un sistema de arquitectura abierta con los principios fundamentales del desarrollo libre, VMware cree que la era de la nube debe ser flexible y permitir la participación de la comunidad, es decir de código abierto, con esta convicción fundamental, los componentes de VMware Cloud Foundry, el motor de ejecución de aplicaciones, servicios de aplicación y la interfaz de proveedor de la nube son de código abierto, la naturaleza abierta y extensible de Cloud Foundry, significa que los desarrolladores no se verán atrapados en un marco único, un único conjunto de servicios de aplicaciones o de una sola nube.

¹¹⁹ Wso2 Startos Sitio Web [En línea], [Noviembre 2011]. Disponible en: <http://wso2.com/cloud/stratos/>

¹²⁰ Cloud Foundry Sitio Web [En línea], [Octubre 2011]. Disponible en: <http://start.cloudfoundry.com/getting-started.html>

Cloud Foundry es una plataforma de desarrollo de aplicaciones en la nube que incluye el servicio de motor de ejecución de aplicaciones, un motor de automatización de implementación de aplicaciones y gestión de ciclo de vida, una scriptable interfaz línea de comandos (CLI), la integración con otras herramientas de desarrollo para facilitar los procesos de desarrollo y de despliegue y una arquitectura abierta para la integración rápida de Frameworks y la interfaz de proveedor de la nube.

Cloud Foundry hace más rápido y más fácil de construir, aplicaciones de prueba, despliegue y escala y está disponible a través de una variedad de distribuciones de nube privada y pública.

- **Acceso a Cloud Foundry.**

- CloudFoundry.com: Es un completo entorno de plataforma como servicio (PaaS), en una instancia pública de Cloud Foundry operado por VMware, que se ejecutan en la infraestructura vSphere. CloudFoundry.com soporta los frameworks sobre spring, Ruby on Rails, Sinatra, Scala y node.js y los servicios de Postgres, RabbitMQ, MongoDB, MySQL y Redis.

- CloudFoundry.org: El sitio de la comunidad de código abierto, en donde los desarrolladores y miembros de la comunidad puede colaborar y contribuir con el proyecto y en donde se puede obtener un catálogo completo de servicios de software disponible en la pila de código abierto.

- Micro Cloud Foundry: Es una versión completa descargable de Cloud Foundry que se ejecuta en una máquina virtual dentro de una infraestructura privada o en un PC para desarrolladores. Micro Cloud Foundry contiene una versión del software Cloud Foundry y ofrece una simetría con otras instancias de Cloud Foundry. Permite el acceso a los frameworks modernos y un rico ecosistema de servicios de aplicaciones de VMware, terceros y la comunidad de código abierto. Las aplicaciones desplegadas en Micro Cloud Foundry se ejecutará con una mínima modificación en cualquier nube pública o privada que corra Cloud Foundry.

- **Lenguajes de programación que soporta Cloud Foundry. En este momento, CloudFoundry.com (la instancia pública de la Cloud Foundry operado por VMware) es compatible con las aplicaciones escritas en Spring, Java, Rails y Sinatra para Ruby, Node.js. Scala y otros idiomas JVM / frames como Groovy y Grails. Además, el proyecto de Cloud Foundry de código**

abierto ofrece otros lenguajes/ frames de apoyo a través de socios como PHP, Python,. NET y mucho más.

Figura 17, Página oficial de Cloudfoundry.org¹²¹



Fuente, Cloud Foundry, Disponible en: <http://cloudfoundry.org/>¹²²

5.2.3.5. Wavemaker. Según Wavemaker¹²³, es un Framework de código abierto para desarrollo de aplicaciones Web en la capa de plataforma como servicio (PaaS) en la nube, anteriormente conocido como ActiveGrid, el cual fue adquirido por Vmware en marzo del 2011.

¹²¹ Cloud Foundry Sitio Web [En línea], [Octubre 2011]. Disponible en: <http://start.cloudfoundry.com/getting-started.html>

¹²² Cloud Foundry Sitio Web [En línea], [Octubre 2011]. Disponible en: <http://cloudfoundry.org>

¹²³ Wavemaker Sitio Web [En línea], [Octubre 2011]. Disponible en: <http://www.wavemaker.com/product/>

WaveMaker está licenciado bajo la Licencia Pública General de Affero versión 3 (GPL v3) y bajo una licencia no-comercial.

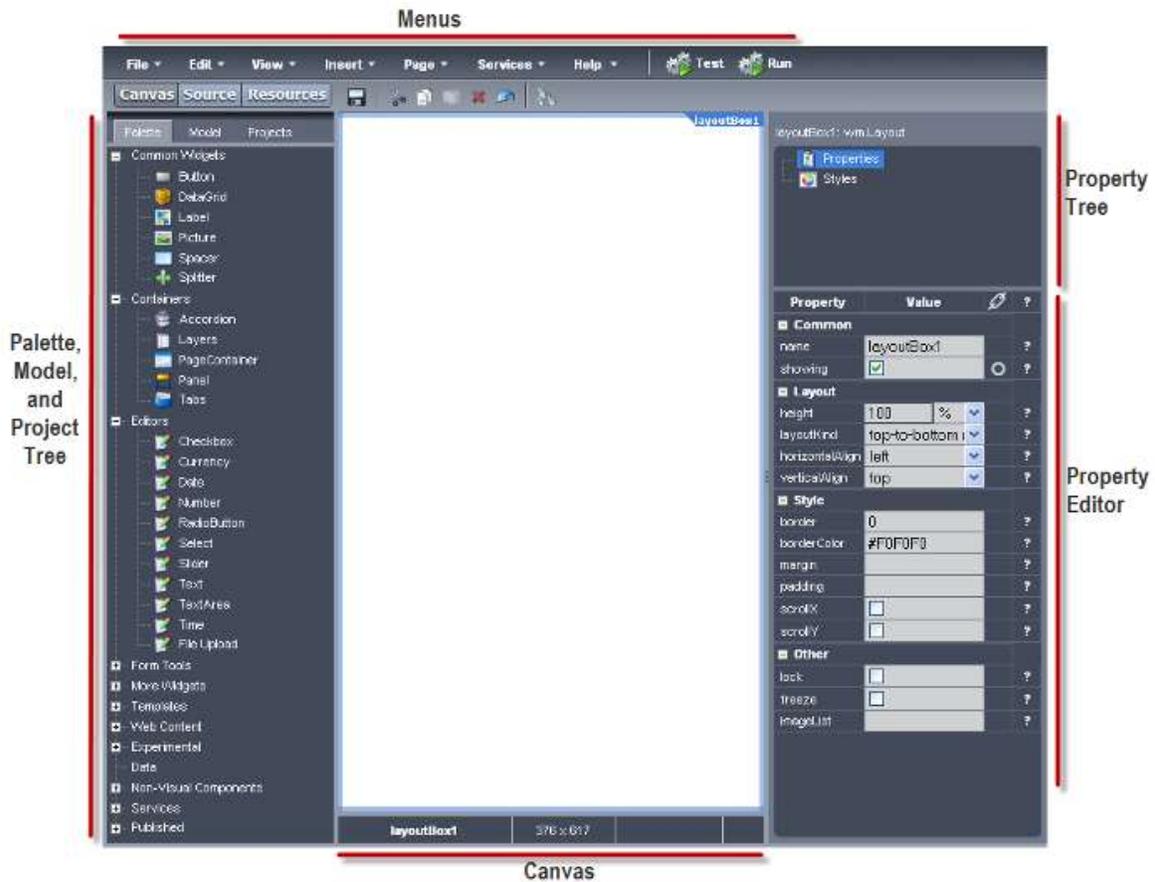
WaveMaker ofrece herramientas de desarrollo rápido de aplicaciones para construir aplicaciones web, como por ejemplo la entrada de pedidos, gestión de proyectos y formularios web con la combinación del poder de AJAX, las aplicaciones se ejecutan en un servidor estándar de Java basado en Apache Tomcat, Dojo Toolkit, Spring e Hibernate. Actualmente está soportado para Microsoft Windows, Linux y Macintosh, permitiendo el despliegue tanto en nubes públicas como privadas.

Componentes de Wavemaker. WaveMaker es un entorno de desarrollo rápido de aplicaciones que incluye dos componentes:

- WaveMaker Studio: Entorno de desarrollo visual con la facilidad de arrastrar y soltar y que se ejecuta a través de cualquier navegador web.
- WaveMaker Framework en tiempo de ejecución: El servidor de tiempo de ejecución se encuentra incrustados en la aplicación creada por el WaveMaker estudio, las aplicaciones desarrolladas con WaveMaker usan estándares de archivos JAVA WAR que se pueden ejecutar en cualquier servidor Java.

Figura 18, Interface de Wavemaker Studio¹²⁴

¹²⁴ Wavemaker Sitio Web [En línea], [Octubre 2011]. Disponible en: <http://www.wavemaker.com/product/>



Fuente, Wavemaker disponible en:

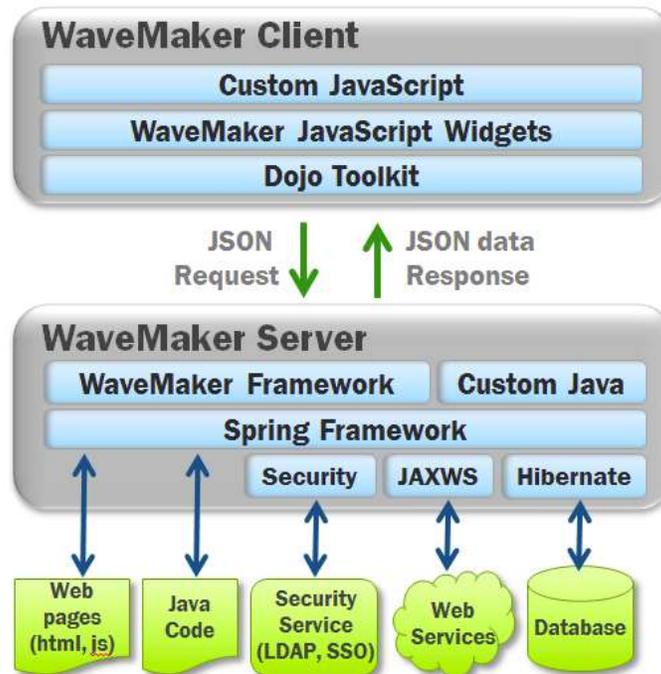
http://dev.wavemaker.com/wiki/bin/wmdoc_6.3/Intro¹²⁵

- Arquitectura de WaveMaker. El diagrama de la arquitectura se muestra cómo los diferentes elementos de WaveMaker encajan entre sí.

Figura 19, Componentes de código abierto de la arquitectura WaveMaker¹²⁶

¹²⁵ Wavemaker Sitio Web [En línea], [Octubre 2011]. Disponible en: http://dev.wavemaker.com/wiki/bin/wmdoc_6.3/Intro

¹²⁶ Wavemaker Sitio Web [En línea], [Octubre 2011]. Disponible en: <http://dev.wavemaker.com/wiki/bin/Dev/Architecture>



Fuente, Wavemaker disponible en:

<http://dev.wavemaker.com/wiki/bin/Dev/Architecture>¹²⁷

Estos componentes están incluidos tanto en la Community Edition y Enterprise Edition de WaveMaker.

- Dojo: widgets Ajax
- JSON: mensajes asíncrono
- Spring: Java contenedor
- Hibernate: acceso a base de datos
- Jaxws: acceso a servicios web
- Spring security: Autenticación de usuarios

¹²⁷ Wavemaker Sitio Web [En línea], [Octubre 2011]. Disponible en: <http://dev.wavemaker.com/wiki/bin/Dev/Architecture>

5.2.3.6. Tabla Comparativa Proveedores de PaaS, basada en Software Libre.

Tabla 6, Cuadro Comparativo proveedores PaaS, basados en Software Libre.

Características	Google AppEngine	Appscale	WSO2 Stratos	CloudFoundry	WaveMaker
Lenguajes que soporta	Python, Java, and Go	Python, Java, and Go	Middleware basado en SOA, que soporta Framework para php, C y C++	Java / Spring, Groovy/Grails, Ruby Rails y Sinatra, Node.js	Framework de Desarrollo de aplicaciones Web web en AJAX
Servicios, Base de datos y almacenamiento.	Serlets, MySQL.	5.3. HBase, Hypertable, Apache Cassandra, MySQL Cluster, y Redis	Cassandra como service, MySQL, y HDFS.	MongoDB, MySQL, Redis y .NET Framework	MySQL, PostgreSQL, HSQLDB, Oracle, Microsoft SQL Server y IBM DB2
Despliegue	SDK Remoto	Xen, KVM, Amazon EC2 AMI y Eucalyptus.	En Ubuntu Enterprise Cloud (UEC) basado en Eucalyptus	Vmware. Maquina virtual montada sobre Ubuntu 10.04 en una nube privada. (Vmware player, Workstation o Server multiplataforma.)	Instalación de Wavemaker Studio en Windows 7/XP/Vista/Server 2003/2008, Mac OS X, Red Hat Enterprise Linux, CentOS, Debian y Ubuntu.
Integración con Otros		Permite desplegar el las API's de Google AppEngine de manera local.	Google auth model y otros servicios WSO2 almacenamiento Amazon S3	En versión beta permite integran con .NET Framework	Tomcat, JBoss, GlassFish, WebSphere and WebLogic. WaveMaker corre en Ambientes standard

					J2EE
Interface de Administración.	Panel de Control, Aplicación Web.	Consola (Línea de comandos)	Panel de Control, Aplicación Web.	VMC, Cliente line de comandos y Cloud Foundry SpringSource Suite de Herramientas (STS) para Eclipse grafica	Wavemaker Studio Interface gráfica sobre cualquier Navegador Web.
Administración y Seguridad	Usuarios registrados en Google	Integración con el API de usuarios de Google.com	Integración con Eucalyptus, Soporta Credenciales X509	Registro y autenticación de Usuarios, validación de un DNS único.	Inicio de sesión único mediante LDAP, Josso, SiteMinder o SPNEGO.
Tipos de Nubes	Publica	Privada, Publica e Híbridas	Privada y publica	Publica y Privada	Privada.
Tipo de Licencia	Uso de Herramientas y lenguajes basados en software Libre	BSD License	Apache License 2.0.	Apache License 2.0	Apache License 2.0
Creado Por	Googele.inc	RACELab at UC Santa Barbara	WSO2	VMware	Wavemaker, adquirido por Vmware en marzo de 2011
Sitio Web	http://code.google.com/intl/es-ES/appengine/	http://appscale.cs.ucsb.edu/	http://wso2.com/cloud/stratos/	http://cloudfoundry.org/	http://www.wavemaker.com/

Fuente, Elaboración propia.

5.3.1. Herramientas de software como servicio (SaaS) de Software Libre. El modelo SaaS tiene disponible software libre y de código abierto, lo que permite su uso, modificación y mejora del diseño porque se cuenta con el código fuente.

Se puede mencionar las siguientes aplicaciones que son desarrolladas en software libre y que están disponibles para su uso:

Para el área financiera y contable existe una solución denominada Phreebooks desarrollada en PHP y en cuyas funcionalidades podemos encontrar, contabilidad, inventarios, bancos, proveedores, empleados clientes, manejo de fletes, manejo de varias sucursales entre otras. Una aplicación versátil disponible para pequeñas y medianas empresas que puede configurarse para varios idiomas.

En el manejo de copias de los archivos se puede considerar una solución denominada Amanda (Advanced Maryland Automatic Network Disk Archiver) esta aplicación permite hacer un respaldo de varios computadores que estén en la red ya sea de unidades de cinta, o medios de discos ópticos, este software usa en sus mecanismos de copia formatos tipo GNU tar con copias de seguridad de gran número de servidores o estaciones de diversas versiones en Linux o Unix, también puede utilizar clientes nativos de Windows para copias de seguridad en servidores de este sistema operativo.

En cuanto aplicaciones en el campo de la inteligencia de negocios podemos mencionar a Jaspersoft una aplicación que por medio del ambiente cloud puede manejar datos almacenados en fuentes relacionales por medio de OLAP, cuenta con un motor de reportes desarrollado en Java para combinar diferentes fuentes de datos, como también la herramienta Eclipse para el diseño de informes.

Una herramienta libre es Pentaho que puede ser una alternativa para inteligencia de negocios, tiene una plataforma integrada para análisis de datos de diversos tipos y magnitudes, cuenta con potentes utilidades para visualización, análisis y minería de datos, sus capacidades de integración radican en el un completo SDK en Java, una interfaz de servicios web que puede ser personalizable, inmersos sistemas de seguridad y arquitectura modular y extensible.

Jedox BI Suite proporciona funcionalidades de Business Intelligence que sirven para planificar, analizar, comunicar y visualizar la información de la empresa, posee un servidor OLAP que es una aplicación multiusuario que permite a los usuarios de la empresa acceder, modificar los datos que hacen parte del módulo de inteligencia de negocios de forma instantánea, el servidor ETL es una

aplicación web para la adquisición de datos que se extraen, transforman y cargan desde los sistemas transaccionales, bodegas de datos y otras fuentes externas, con el servidor web el diseñador puede administrar o crear informes basados en internet de la base de datos OLAP o un monitor ETL y a la vez los usuarios de los negocios pueden mirar los informes de planificación y análisis en paginas web, una funcionalidad adicional corresponde a la de los usuarios móviles que tienen acceso a las opciones de planificación, informes y análisis a través de dispositivos como iPad, iPhone o teléfonos celulares.

Open Intelligence OpenI una aplicación de código abierto orientada al tratamiento de datos tiene características como modelado tridimensional, diseños de cubos en OLAP, modelos de predicción, información interactiva y cuadros de mando haciendo uso de código abierto.

Process Maker edición Cloud está orientado al desarrollo de Workflow en las organizaciones, esta aplicación posee un modelo de definición de flujos de procesos, un constructor de formularios, un generador de salida de documentos y servicios web con el fin de eliminar el uso del papel, como también un sistema de notificación automática, definición de informes estadísticos y cuadros de mando para la gestión de Business Intelligence, todo esta orientado al modelo cloud. Este software puede obtenerse en código abierto para Linux y Windows que sea usado para pruebas.

Alfresco ofrece una aplicación para gestión de contenidos, gestión de documentos, modelo de colaboración, gestión de contenidos y conocimiento sobre esto desarrollado sobre código abierto. Este sistema ha usado las tecnologías Open Source como Spring, Hibernate, Lucene, Servicios Web, Java Server Faces para la construcción de su modelo funcional de gestión documental. Alfresco permite a los usuarios una interface de fácil acceso, un sistema inteligente de archivos virtual y también un portal con toda la potencia de un Sistema de Gestión de Contenido Empresarial.

Para Gestión de relaciones con clientes se pueden mencionar varias soluciones cloud como SugarCRM que proporciona una plataforma cloud con funcionalidades web para el registro de contactos, la información de oportunidades de negocio, la creación y modificación de documentos, la generación de informes de operaciones comerciales y la gestión de casos entre otros. También se puede encontrar a vTiger una solución adecuada para medianas y pequeñas empresas, proporciona un servicio bajo demanda, su tarifa es de 12 dólares mes con acceso a quince módulos relacionados con ventas, marketing, manejo de inventarios,

gestión de seguridad, flujos de trabajo entre otros. Orange Leap para entidades sin ánimo de lucro, es una solución bajo demanda basada en web, gestiona la operación de donaciones en organizaciones no lucrativas con una base de datos de código abierto y una interfaz de software de donación personalizable.

Para Sistemas para Gestión de documentos, los objetivos que proponen las aplicaciones de este tipo son: gestionar documentos para la operación de negocios, encontrar documentos con valor y efectividad, propender en los usuarios la participación en la construcción de documentos de calidad, permitir el acceso a los documentos desde cualquier lugar y garantizar la seguridad técnica de los documentos, en ese sentido se encuentran soluciones como KnowledgeTree para el manejo de workflow, OpenKm un sistema de gestión documental con control de versiones, búsqueda de contenidos, integración con diversas bases de datos, planificador de tareas disponible en versiones comerciales y host.

En el tema de correo electrónico y su aplicación en el área de cloud especialmente para el tema comercial se puede conseguir aplicaciones de software como Zimbra que comprende un servidor de colaboración con mayor sencillez en el manejo de correo electrónico, tiene funcionalidades como manejo de contactos, calendarios, gestión de documentos y la capacidad de movilidad y de escritorio para los usuarios en cualquier computador, como también la posibilidad de operar esta aplicación en la nube.

OpenMM es una aplicación basada en web para el marketing por correo electrónico, para el manejo de boletines electrónicos y servicios de transacciones o eventos por correo, esta aplicación de código abierto funciona en varios servidores distribuidos como software bajo demanda.

En el campo de sistemas de planificación de recursos de las empresas (Enterprise Resource Planning ERP) que son sistemas para el manejo de información que integran y automatizan a una organización se puede enunciar a OpenERP una plataforma que ofrece módulos para manejo de clientes, compras, manufacturas, gestión de almacenes, contabilidad, marketing y recursos humanos.

xTuple Postbooks es un sistema completo integrado de módulos como contabilidad, ventas, inventarios, manejo de clientes y planificación de recursos, está construido con un motor de base de datos abierto como es Postgresql y la herramienta Qt desarrollada en C/C++, este sistema funciona en Windows, Linux y Mac con aplicación de multi moneda y tiene un licenciamiento de tipo Common de tipo público.

Compiere ERP CRM Solution es una solución que ofrece un grupo amplio de funcionalidades empresariales integradas en una plataforma global, esta aplicación puede manejar varias organizaciones, almacenes, su capacidad está proyectada a varios servidores con grandes volúmenes de datos, cientos de usuarios y un número considerable de transacciones. Compiere tiene la opción de desplegar las aplicaciones en un centro de datos privado o en la nube de Amazon. Como complemento a las aplicaciones de planificación de recursos está Opentaps, una suite que integra proyectos de código abierto orientados al comercio electrónico, gestión de almacenes e inventarios, gestión de suministros, gestión financiera y business intelligence. Entre las herramientas de desarrollo de Opentaps se pueden mencionar a Java J2EE, Apache Tomcat, usuarios tipo AJAX, inteligencia de negocios con Pentaho y Jasper, como también compatibilidad con MySQL, PostgreSQL, Oracle y Microsoft SQL Server.

En cuanto a sistemas de gestión de Recursos Humanos las aplicaciones representativas son Orange HRM orientado a pequeñas y medianas empresas, con módulos de información sobre personal, servicio de información a los empleados y servicios de reclutamiento de personal, la nueva funcionalidad está en el servicio Live SaaS donde el software queda disponible en la nube y sus costos de operación se hacen por empleado registrado. Open Applicant aplicación proyectada para la selección de los mejores candidatos para empleos, permite la gestión de información de los candidatos, pruebas de selección, generación de informes.

Entre las aplicaciones cloud para la gestión de proyectos están OpenProj que es la alternativa de open source a Microsoft Project, tiene operatividad con diagramas Gantt y diagramas PERT, disponible para descarga o sobre demanda. También se puede optar por Onepoint Project para la gestión de proyectos empresariales. En el ambiente académico soluciones como openSIS permite el manejo de escuelas en la gestión demográfica, información de contactos, programación académica, libro de calificaciones, registros de salud, asistencia y diseñador de informes; se puede usar en sitio o en la nube por medio de un cobro por demanda con tarifas de 19 dólares mensuales.

5.3.1.1. Aplicaciones Desktop en la nube.

- EyeOs: Según EyeOs¹²⁸, es una aplicación tipo escritorio que se puede ejecutar desde un browser, el proyecto se inició en Barcelona desde 2005 y actualmente se consolida como una sistema de código abierto con renombre tanto así que IBM lo eligió como su plataforma Open Cloud en el año 2010.

El escritorio de eyeOs la nube presenta, organiza y gestiona los datos, archivos y aplicaciones de usuario. La versión profesional mantiene el control de los datos en la nube, lo que permite el acceso al espacio de trabajo del usuario desde cualquier dispositivo, esta solución por medio de la virtualización ahorra el costo en servidores y licencias de software.

- Funcionamiento: El software se instala en los servidores del cliente para la prestación del servicio a empleados, colaboradores externos o clientes. La aplicación crea una nube privada, lo que garantiza que los datos de los usuarios se encuentran bajo el control del administrador del sistema, el administrador toma la decisión de cuánto espacio se reserva a los usuarios, que permisos tienen a que aplicaciones y que funcionalidades pueden disfrutar, el escritorio se puede personalizar en el entorno visual, contenidos y menús. El software está programado con herramientas de software libre como PHP, Javascript y Ajax.

El modelo básico de eyeOS contiene las siguientes opciones:

- eyeFiles para la creación de carpetas, renombrado, copia de archivos, cambio de privilegios de usuario.
- eyeMail gestiona el correo de forma integrada, adjunta y descarga archivos.
- eyeDocs un editor de textos que permite abrir, editar, crear documentos e importarlos a otro formato.
- Integración para la creación de accesos directos a los recursos web y aplicaciones de la empresa que cuenten con un front-end de tecnología web.
- Compartición via URL permite crear un link desde el mismo escritorio que enlaza al fichero que se quiere compartir.
- Avisos en tiempo real recibe notificaciones al momento de lo que suceden en el entorno.
- Personalización para iconos y aplicaciones por defecto, de logos para introducir la imagen corporativa y el cambio del fondo de pantalla deseado.

¹²⁸ EyeOs Sitio Web [En línea], [Octubre 2011]. Disponible en: <http://eyeos.org/es/>

- Ubuntu One. Según Canonical¹²⁹, es un servicio de alojamiento de archivos, interado con el entorno de escritorio del sistema operativo Ubuntu orientado a la copia de seguridad y la sincronización de archivos e información entre computadoras conectadas a internet. Ubuntu One fue lanzado en mayo de 2009 para obtener ingresos a Ubuntu un sistema operativo respaldado por Canonical. Esta aplicación presenta una gama de servicios en la nube como el acceso inmediato a música, fotos, videos, documentos importantes en cualquier momento y desde cualquier dispositivo, el almacenamiento gratuito se presta hasta 5 GB o pagar por uso para streaming de música.

El software de cliente está escrito casi en su totalidad en Python. El código fuente está disponible en el sistema del proyecto de control de versiones, a diferencia de los componentes de cliente el software del servidor no está disponible bajo una licencia de código abierto.

- Funcionamiento.

- Panel de Control

Ubuntu One cuenta con un panel de control que sirve para administrar las cuentas, las carpetas en la nube, los dispositivos y los servicios en el computador personal del cliente. Ubuntu One permite el control de las carpetas que están en el escritorio y en la nube personal, además de acceder desde la web en cualquier momento.

- Compartir y Colaborar

Con esta opción se puede compartir archivos o crear enlaces cortos para colocar en espacios sociales como Facebook y Twitter.

- Music Store

Esta función permite comprar música y se entrega directamente a la nube personal que puede sincronizarse y transmitirse a teléfono celular, la colección de música puede ampliarse a varios dispositivos por medio de un streaming de música de 20 GB.

5.3.1.2. Desarrollo de pruebas a EyeOs y Ubuntu One. Para el sistema EyeOs se ingresa al sitio <http://eyeos.org> y se busca la versión profesional de la

¹²⁹ Ubuntu One Sitio Web [En línea], [Noviembre 2011]. Disponible en: <https://one.ubuntu.com/>

aplicación seleccionando la opción de prueba del sitio web, esta página solicita un nombre de usuario y una clave de acceso.

Figura 20, Ingreso a EyeOs



The image shows a login form for EyeOS. At the top left is the EyeOS logo, which consists of a green stylized 'e' inside a circle, followed by the text 'eye os' and the tagline 'solutions for a new world'. Below the logo, there are two input fields. The first is labeled 'Username:' and contains the text 'henryrodcar'. The second is labeled 'Password:' and contains a masked password represented by seven asterisks. To the right of the password field is a button labeled 'Login'. At the bottom left of the form, there is a link that says 'Create a new account'.

Fuente, Elaboración propia.

Con la cuenta creada se ingresa al escritorio de eyeOs que tiene las funcionalidades de eyeFiles, eyeCal,eyeContact,eyeSync, eyeMail, eyeChat, eyedDocs, eyeConfig todas ellas operando desde el navegador web.

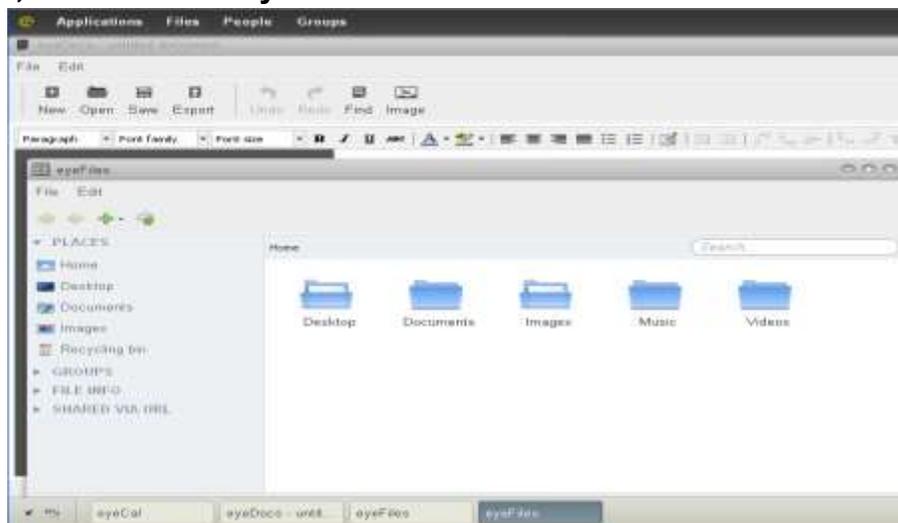
Figura 21, Interface inicial de EyeOS



Fuente, Elaboración propia.

La funcionalidad eyeFiles me permite gestionar todo tipo de archivos y folders de datos, existen varios lugares para alojar los archivos como home, desktop, documents, images y recycling bin

Figura 22, Directorio en EyeOs



Fuente, Elaboración propia.

El manejo de documentos se lo tiene con eyedDocs, un editor de texto que permite crear, modificar, buscar documentos, funcionalidades de formatos al texto, justificado entre otros.

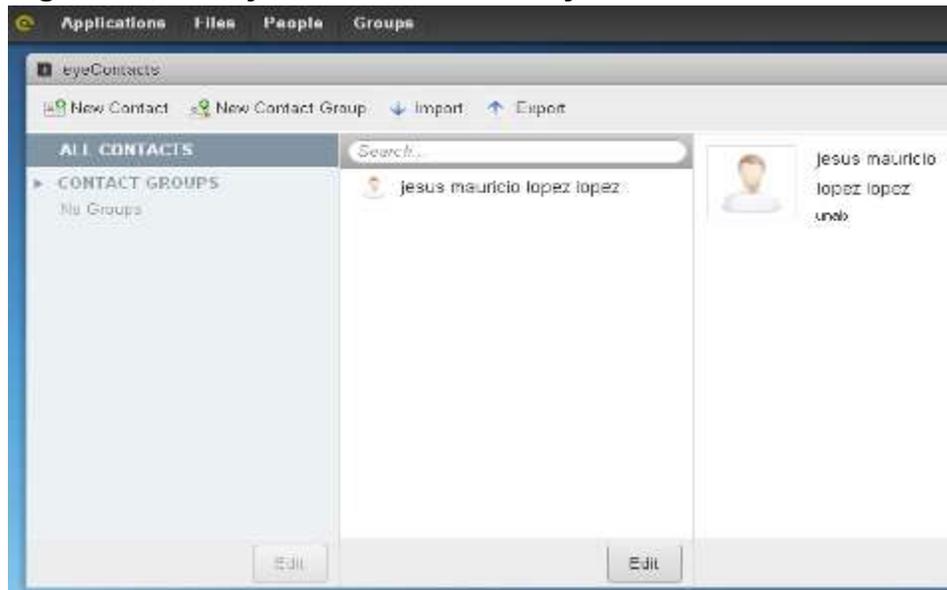
Figura 23, Abrir documentos en EyeOs



Fuente, Elaboración propia.

La opción de eyeContacts maneja la información de contactos que se pueden organizar por grupos.

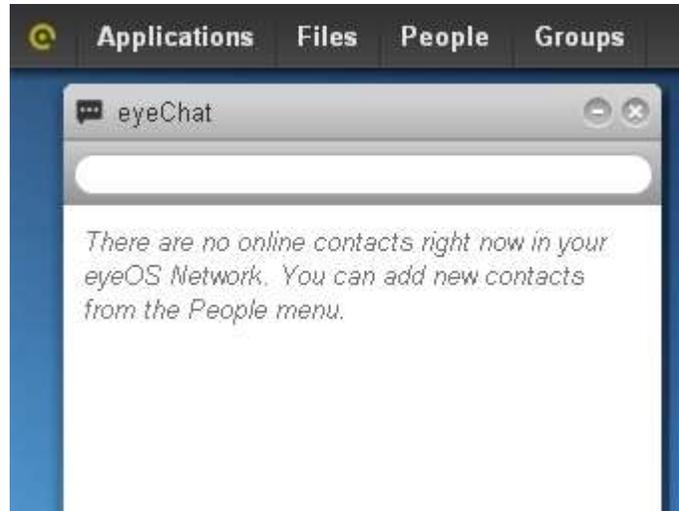
Figura 24, Manejo de Contactos en EyeOS



Fuente, Elaboración propia.

La funcionalidad de eyeChat permite establecer dialogo con otros usuarios de eyeOS que se encuentren conectados en la web.

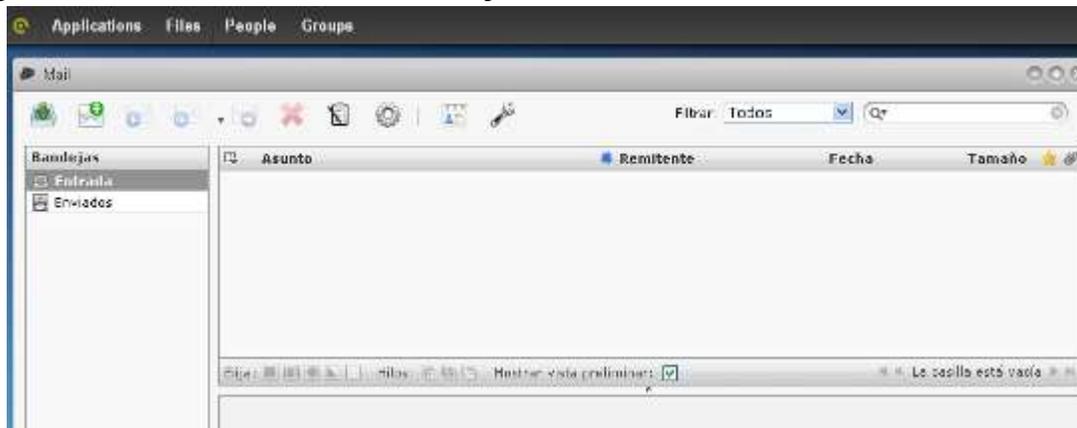
Figura 25, Chat en EyeOs



Fuente, Elaboración propia.

El manejo de correo electrónico esta disponible con eyeMail un buzón que se puede configurar para enviar y recibir mensajes de texto y archivos adjuntos.

Figura 26, Correo Electrónico en EyOs



Fuente, Elaboración propia.

La funcionalidad eyeSync permite una sincronización con otros sistemas operativos conectados a la red para envío y recepción de archivos.

Figura 27, Sincronización con S.O Locales de EyeOs



Fuente, Elaboración propia.

Existe la opción de notepad para desarrollar archivos de texto y la funcionalidad de preferencias que nos ayuda a configurar el entorno de apariencia del escritorio donde trabajamos la aplicación eyeOS.

Ubuntu One se ingresa al sitio web <http://one.ubuntu.com> y se selecciona el sistema operativo donde se va a instalar la aplicación, posteriormente se descarga el instalador y se procede a su instalación, en este proceso se requiere de una cuenta de correo electrónico y una clave para el ingreso al aplicativo.

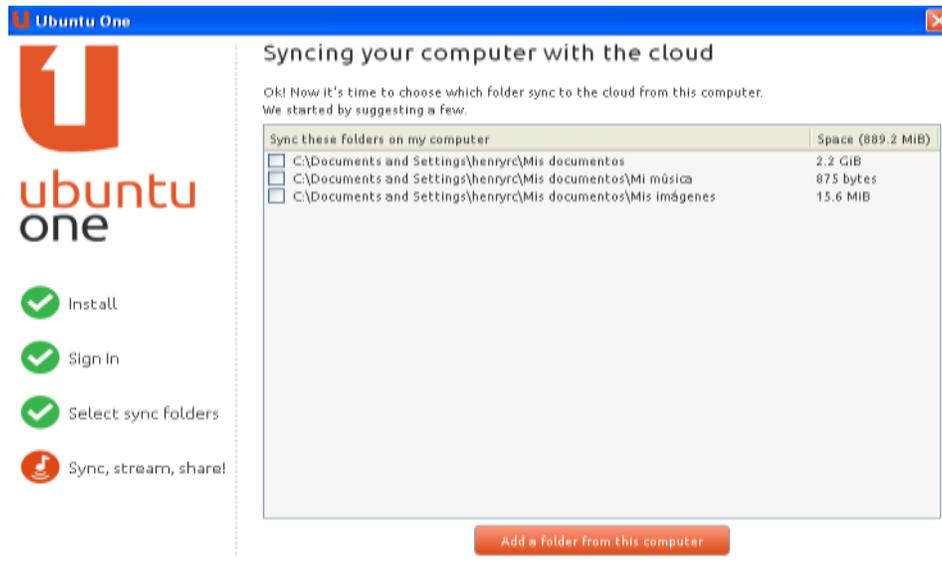
Figura 28, Instalación de Ubuntu One



Fuente, Elaboración propia.

El aplicativo solicita los directorios que se van a sincronizar en la nube cuya capacidad máxima es de 5 GB.

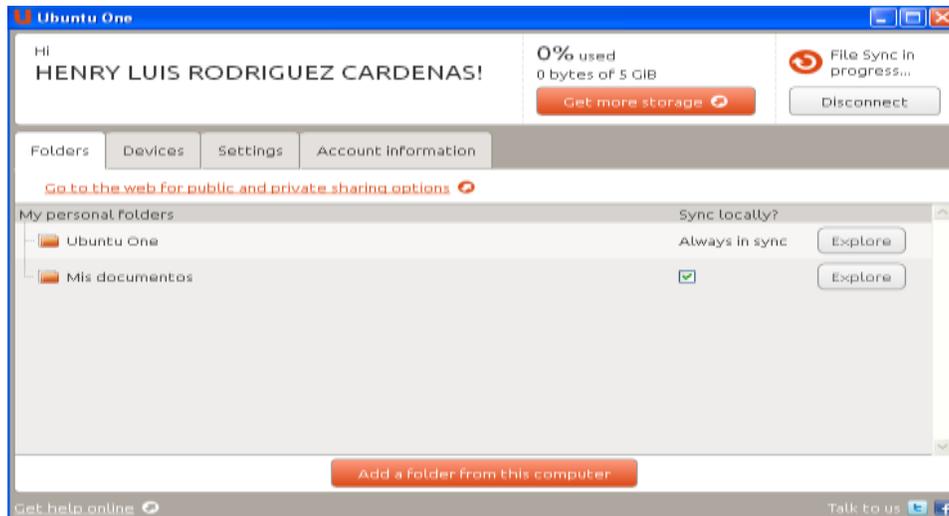
Figura 29, Sincronizar documentos con Ubuntu one



Fuente, Elaboración propia.

El aplicativo que se coloca en el cliente permite configurar los fólдерes a compartir en la web, los dispositivos y la información de la cuenta del usuario.

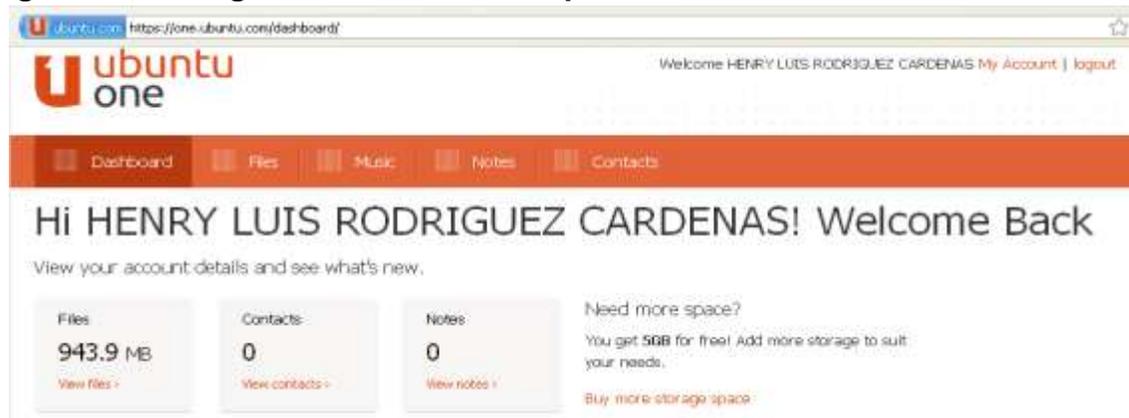
Figura 30, Compartir carpetas en la nube con Ubuntu One



Fuente, Elaboración propia.

El en sitio web con el usuario y la clave se puede ingresar a la nube con el acceso a los directorios que se habían sincronizado y los archivos que estaban contenidos.

Figura 31, Configuración de la cuenta personal en Ubuntu One



Fuente, Elaboración propia.

Ubuntu One en el menú de cloud posee las opciones Dashboard para configuración de archivos a almacenar, la lista de contactos y la relación de notas de texto se pueden crear.

Con la opción files se puede obtener una relación de archivos que se han dispuesto en la nube, aquí se suben archivos, se crean nuevas carpetas o se comparten archivos o folders.

La opción Music permite gestionar hasta 20 GB de archivos de música accesibles desde la web o dispositivos móviles.

- Análisis Aplicaciones (SaaS) eyeOs y Ubuntu One.
- Ventajas. De acuerdo al proceso de instalación y prueba de las soluciones eyeOS y Ubuntu One se percibe un modelo sencillo de manejo de estas aplicaciones de escritorio que reemplaza a un sistema operativo, la ventaja de este tipo de aplicaciones radica en que el computador del cliente ya no es necesario, simplemente debe tener una conexión a internet desde cualquier sitio para disfrutar de su sistema personal.

Potencialmente la necesidad de maquinas reales se esta eliminando, en el caso de empresas el costo por adquisición de computadores se va a descartar puesto que la capacidad de computación de las maquinas virtuales presentes en la nube son la alternativa mas viable para remplazo de la infraestructura computacional presente en la actualidad.

En estas soluciones eyeOS y Ubuntu One la actualización de versiones y el mantenimiento de funcionalidades se percibe en la presentación de las interfaces

y también en el respaldo de las comunidades de desarrollo de software libre que las soporten.

Estas aplicaciones de escritorio liberan a sus clientes o a empresas de las preocupaciones de tener equipos y software que mantener actualizados o de invertir grandes cantidades de dinero en su reposición, las funcionalidades de módulos de oficina, copia de documentos y compartición de los mismos quedan bajo la responsabilidad de los proveedores de los servicios en la nube.

La disponibilidad de la información y la seguridad de los datos es una bondad que provee este tipo de soluciones, ya los usuarios no tienen que estar pendientes sobre la copia de sus documentos, ni tampoco que puedan sufrir pérdida de información por casos fortuitos, la nube se encarga de resguardar la información y garantiza la perpetuidad de los datos.

- **Desventajas.** La información no relevante de los usuarios puede ser alojada en las soluciones cloud, pero aquellos datos empresariales de importancia o confidenciales no tendrán el nivel de confiabilidad en aplicaciones cloud ya que no existe una garantía al 100% de seguridad por parte de los altos ejecutivos de las organizaciones.

EyeOS y Ubuntu One presentan un alto nivel de compatibilidad y de intercomunicación con las maquinas de la empresa, sin embargo se necesita un nivel mayor interactividad con otras aplicaciones y sistemas internos a nivel empresarial.

Una desventaja radica en la conectividad de la empresa respecto al servicio de internet, ya que estas soluciones y su infraestructura funcionan en la web, los clientes quedan en un nivel de dependencia total respecto al canal de comunicación y su nivel de cumplimiento en la prestación del servicio.

Las aplicaciones cloud y sus servicios quedan en manos de los proveedores y su nivel de cumplimiento es una preocupación para organizaciones, porque las comunidades de desarrollo son las responsables de la continuidad o no del servicio a los clientes externos.

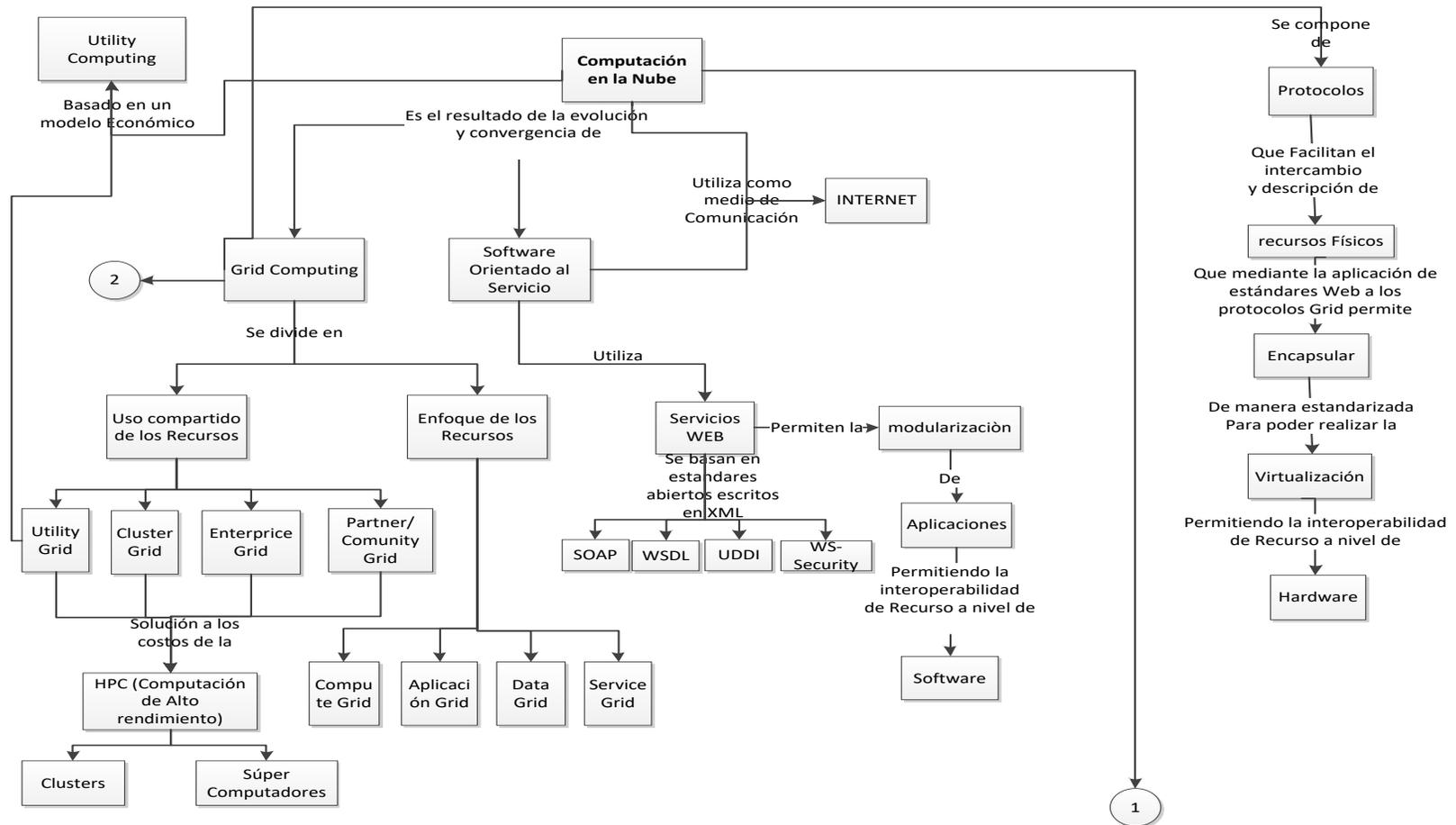
Tabla 7, Cuadro Comparativo Soluciones de SaaS, basados en Software Libre.

Características	Pentaho	EyeOS	Ubuntu One	Opentaps	Tuple PostBooks
Lenguajes que soporta	Php	Php JavaScript	Phyton	Java	Qt C/C++
Servicios que presta	Aplicación para Inteligencia de negocios	5.4. Aplicación tipo escritorio en cloud	Entorno de escritorio del sistema operativo Ubuntu en cloud.	Suite de integración De proyectos de código abierto orientados al comercio electrónico	sistema integrado de módulos como contabilidad, ventas, inventarios, manejo de clientes y planificación de recursos,
Interface de Administración.	Aplicación Web, API	Aplicación Web	Aplicación Web y API	Aplicación Web	Panel de Control, aplicación API
Seguridad	Permisos por usuarios y roles	Uso de cortafuegos, contenido cifrado	Cifrado de datos	Autenticación de usuario, Certificados de autenticación	Autenticación de usuarios.
Tipos de Nubes	Privada, Híbrida	Privada, Híbrida	Pública, Híbrida	Privada, Híbrida, compatibilidad con Amazon EC	Privada, Híbrida
Tipo de Licencia	Open Source	AGPL v3.	GPL	GPL/GNU	CPAL Licencia Pública Común
Creado Por	Corporación Pentaho	Pau Garcia Milla	Canonical	Opentaps	Xtuple
Sitio Web	www.pentaho.com	www.eyeos.org	www.one.ubuntu.com	www.opentaps.org	www.xtuple.com

Fuente, Elaboración propia

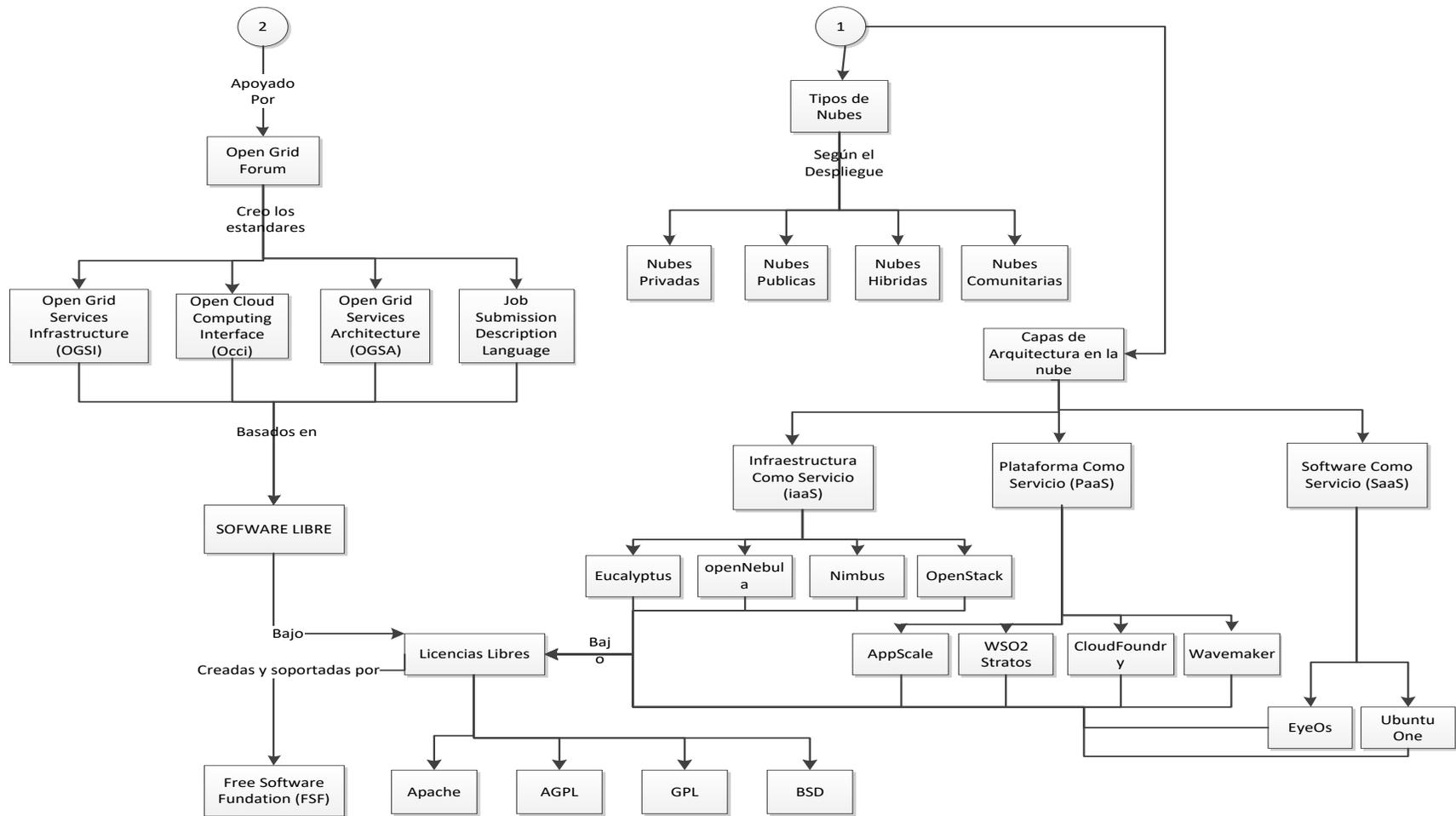
5.5. MAPA CONCEPTUAL DE CLOUD COMPUTING (COMPUTACIÓN EN LA NUBE), DESDE EL SOFTWARE LIBRE.

Figura 32, Mapa Conceptual Computación en la nube, desde el software libre (Parte1)



Fuente, Elaboración propia.

Figura 33, Mapa Conceptual Computación en la nube, desde el software libre (Parte2)



Fuente, Elaboración propia.

6. PROTOTIPO FUNCIONAL BASADO EN EL MODELO DE PLATAFORMA COMO SERVICIO (PAAS) EN UNA NUBE PRIVADA, CON HERRAMIENTAS DE SOFTWARE LIBRE.

6.1. PROTOTIPO 1 PAAS PRIVADA BASADO EN MICRO CLOUD FOUNDRY

- **Micro Cloud Foundry:** Es una versión completa de Cloud Foundry se realiza la descarga y se ejecuta en una máquina virtual dentro de una infraestructura privada o en un PC. Micro Cloud Foundry contiene una versión del software Cloud Foundry y ofrece una simetría con otras instancias de Cloud Foundry. Permite el acceso a los frameworks y un ecosistema de servicios de VMware, terceros y de la comunidad de software libre.

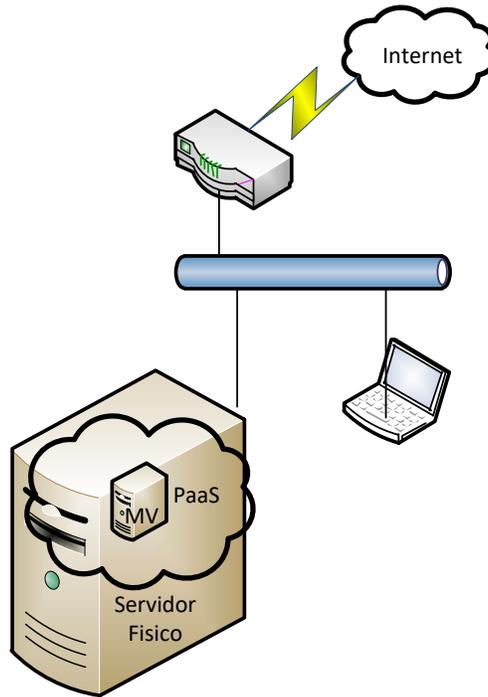
El prototipo se implemento dentro de una pequeña infraestructura privada permitiendo el despliegue completo de una plataforma como servicio (PaaS) privada.

Tabla 8, Prerrequisitos prototipo 1

PRERREQUISITO	DESCRIPCIÓN
Procesador equipo anfitrión:	X86 64Bits
Memoria Ram:	4 Gb (Anfitrión), 1Gb o más (MV)
S.O Anfitrión	Ubuntu Enterprise Cloud 11.04 (64bits)
Software de virtualización	VMWare Player v. 4.0.1
Maquina Virtual huésped (guest)	Ubuntu 10.04.3 (64bist)
Software de (PaaS)	Micro Cloud Foundry v.1.1.0 (Vmware.inc)
Red LAN	Ethernet 10Mb/100Mb puerta de enlace 192.168.1.1

Fuente, Elaboración propia.

Figura 34, Topología de Red Prototipo1



Fuente, Elaboración propia.

6.1.1. Pasos de implementación del prototipo 1

6.1.1.1. Descarga e Instalación Software de virtualización (Vmware Player v.4.0.1). Se ingresa al sitio Web oficial de Vmware (http://downloads.vmware.com/d/info/desktop_end_user_computing/vmware_player/4_0) , en donde con registro previo, se accede a la opciones de descarga, se busca para este caso la versión para Linux de 64bits.

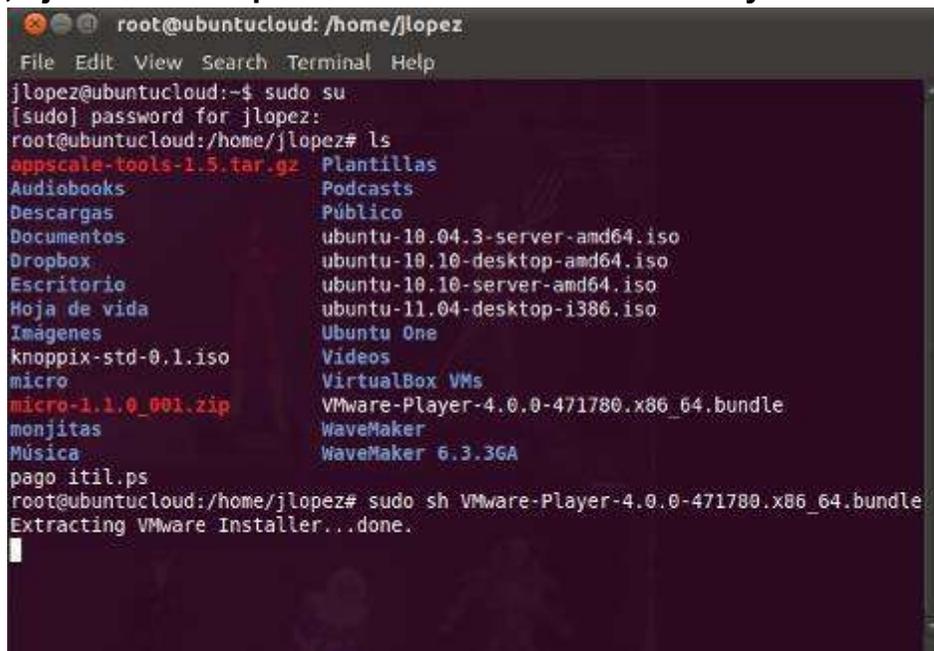
Figura 35, Descarga Vmware Player 4.0.1



Fuente, Elaboración propia.

Una vez realizada la descarga, se procede a la instalación de Vmware Player, como es un paquete .bunble, es necesario ingresar por consola a la ubicación, en donde se guardo el archivo y se corre el Script para iniciar la instalación del software.

Figura 36, Ejecutar el Script de instalación del Vmware Player 4.0.1



```
root@ubuntucloud: /home/jlopez
File Edit View Search Terminal Help
jlopez@ubuntucloud:~$ sudo su
[sudo] password for jlopez:
root@ubuntucloud:~# ls
appscale-tools-1.5.tar.gz  Plantillas
Audiobooks                Podcasts
Descargas                 Publico
Documentos                ubuntu-10.04.3-server-amd64.iso
Dropbox                   ubuntu-10.10-desktop-amd64.iso
Escritorio                ubuntu-10.10-server-amd64.iso
Hoja de vida              ubuntu-11.04-desktop-i386.iso
Imágenes                  Ubuntu One
knoppix-std-0.1.iso      Videos
micro                     VirtualBox VMs
micro-1.1.0_001.zip      VMware-Player-4.0.0-471780.x86_64.bundle
monjitas                  WaveMaker
Música                    WaveMaker 6.3.36A
pago itil.ps
root@ubuntucloud:~# sudo sh VMware-Player-4.0.0-471780.x86_64.bundle
Extracting VMware Installer...done.
```

Fuente, Elaboración propia.

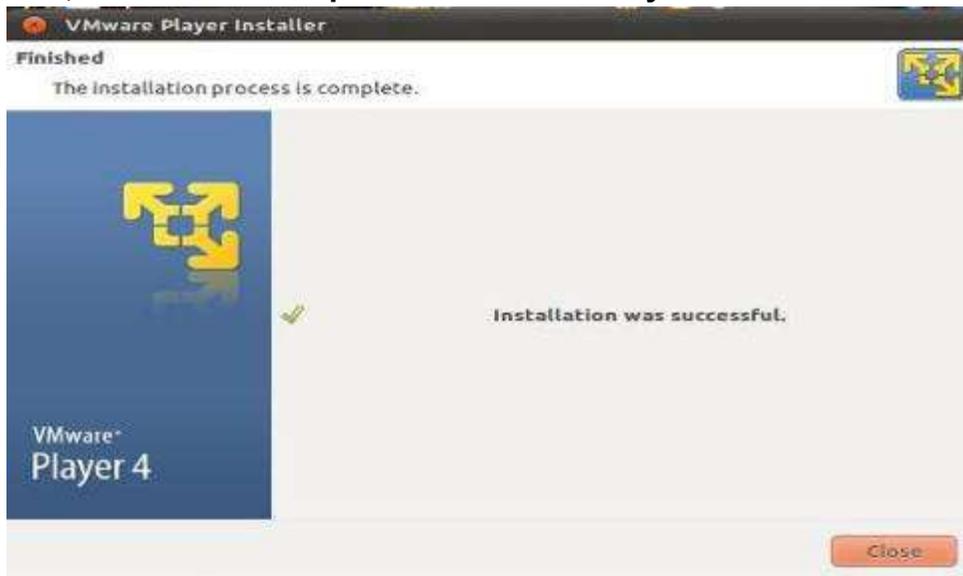
Una vez ejecutado el Script, se inicia el asistente de instalación el cual sesiguen las instrucciones hasta finalizar.

Figura 37, Inicio de instalación del VMware Player 4.0.1



Fuente, Elaboración propia.

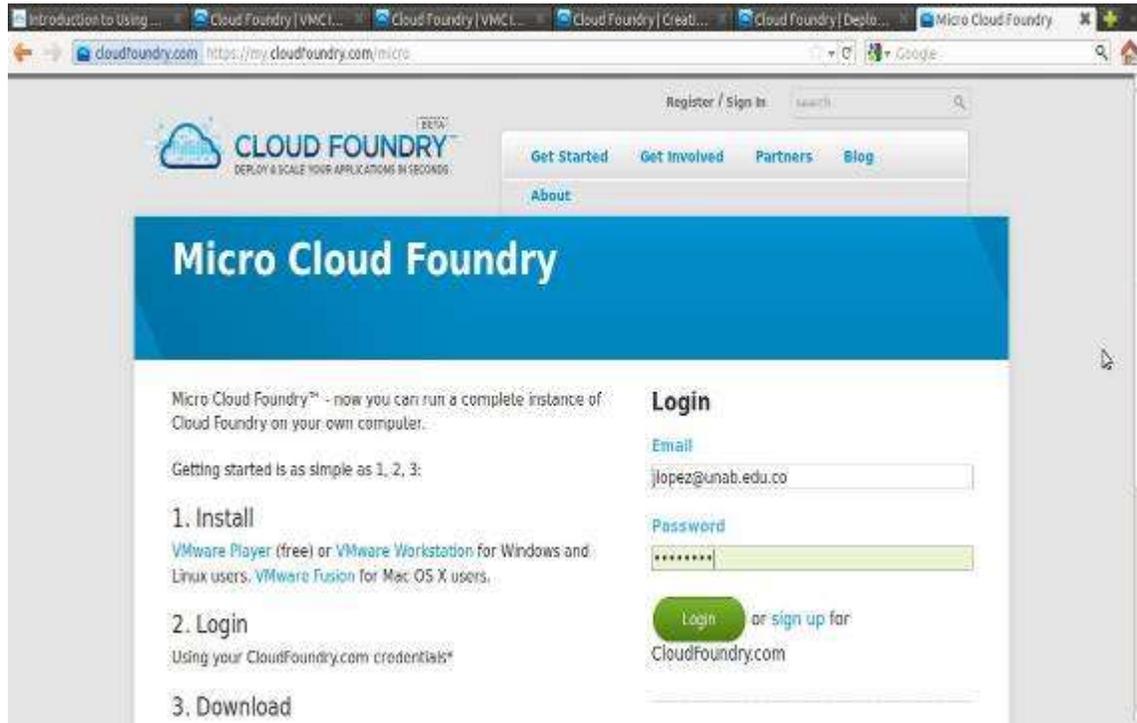
Figura 38, Instalación Completa del VMware Player 4.0.1



Fuente, Elaboración propia.

6.1.1.2. Ingreso y registro en el sitio web de Cloud Foundry.
Se ingresa el sitio web oficial (<https://my.cloudfoundry.com/micro>), en donde con registro previo se accede a la cuenta para la obtención y autenticación de las credenciales.

Figura 39, Ingreso a la cuenta de micro cloud foundry



Fuente, Elaboración propia.

Luego de ingresar a la cuenta personal se aceptan los términos de licencia de usuario final y se crea el nombre de dominio único personal para poder generar el Token que genera las credenciales que permiten autenticar la configuración de Micro Cloud Foundry en la nube privada.

Es este caso se registro el nombre de dominio único como.
jesuscloud1.cloudfoundry.me

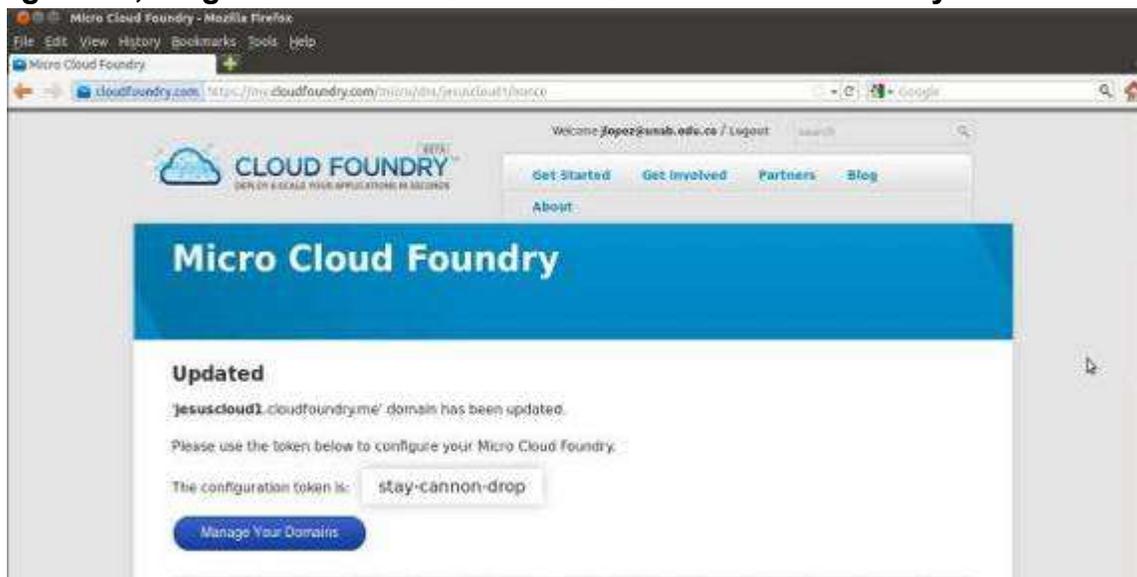
Figura 40, Creación nombre de dominio único para micro Cloud Foundry



Fuente, Elaboración propia.

Una vez creado y validado el nombre de dominio se actualiza y es asignado el Token que se utilizara como credencial de autenticación para configurar Micro Cloud foundry en la nube privada.

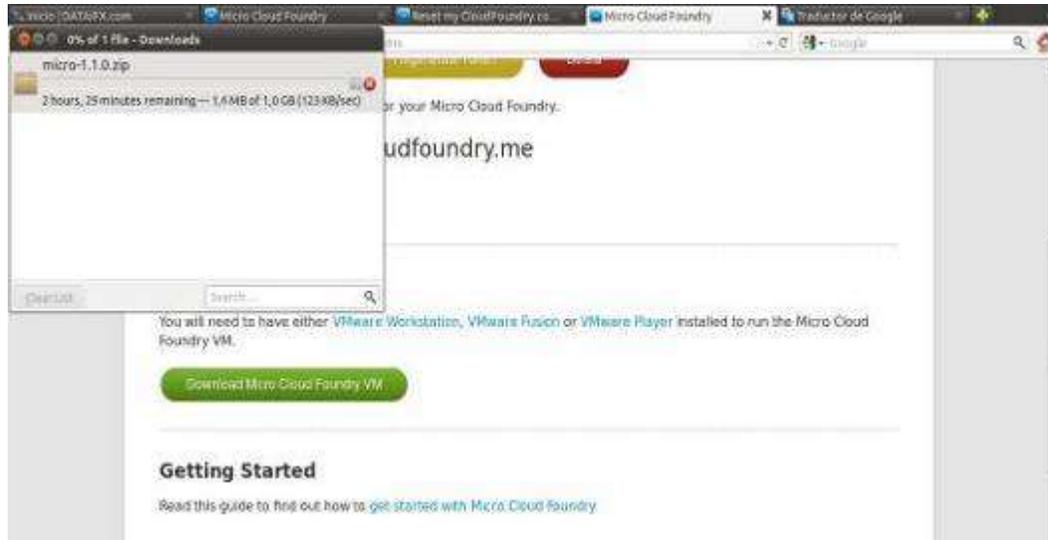
Figura 41, Asignación del Token único de Micro Cloud Foundry



Fuente, Elaboración propia.

Luego de tener registrado el nombre de dominio y el Token único se procede a descargar el archivo de la maquina virtual de Micro Cloud Foundry.

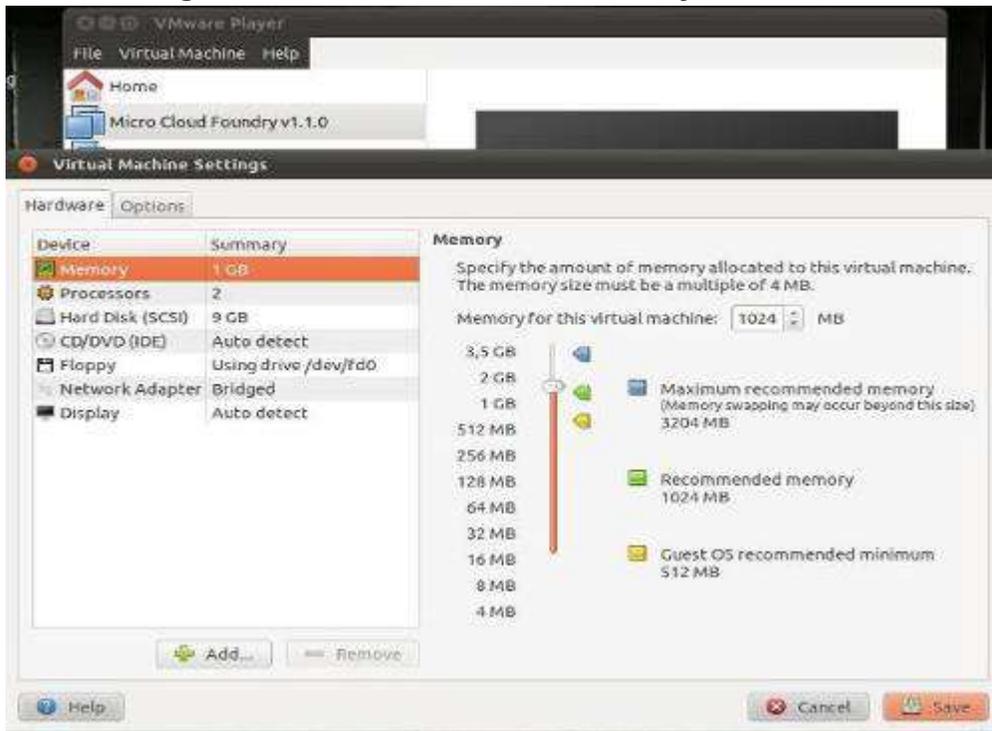
Figura 42, Descarga MV Micro cloud foundry



Fuente, Elaboración propia.

6.1.1.3. Carga y configuración de la maquina virtual de Micro Cloud Foundry. Se descomprime el archivo de la maquina virtual y se inicia Micro.vmx usando Vmware Player instalado previamente. Se realiza y se verifica la configuración de la maquina virtual antes de iniciar.

Figura 43, Configuración MV Micro Cloud Foundry



Fuente, Elaboración propia.

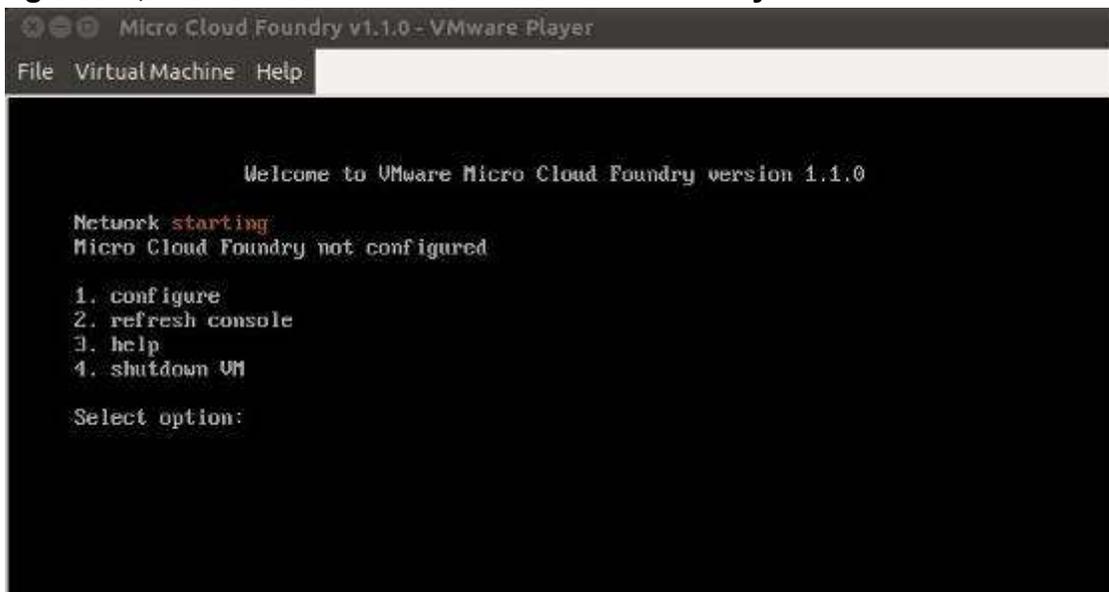
Para la implementación de este prototipo se utilizó la configuración por defecto: Memoria Ram 1Gb, Procesador 2 núcleos, Disco duro 9Gb, Adaptador de red en modo Bridged.

6.1.1.4. Ejecución y Configuración MV de Micro Cloud Foundry.

Al ejecutar la MV por primera vez se carga un menú por consola en donde se realiza la configuración de la instancia de la nube privada.

En la consola de la MV se escoge la opción 1 para iniciar la configuración seguido de enter.

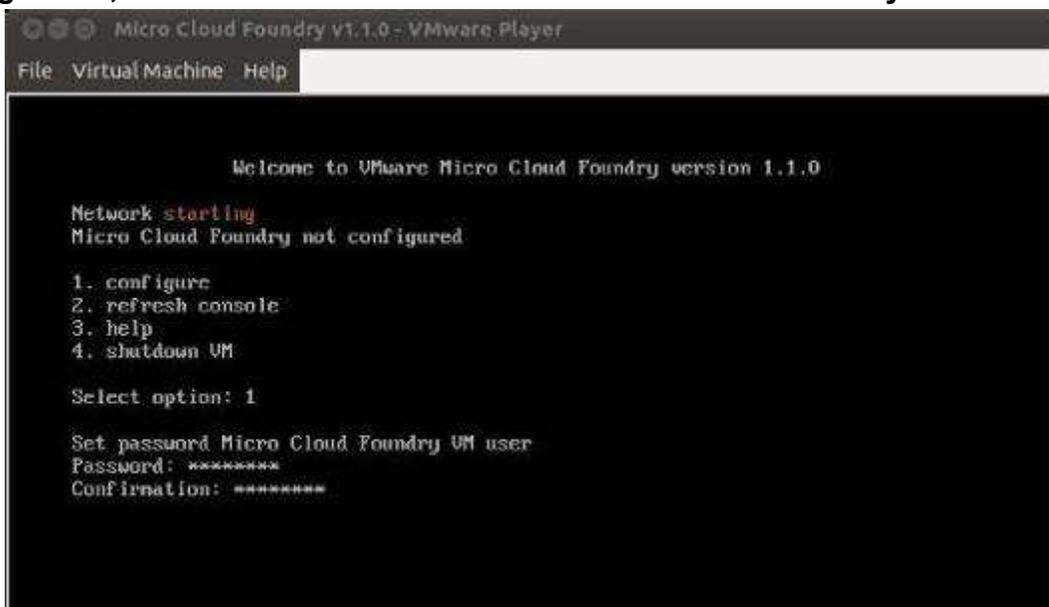
Figura 44, Inicio instancia MV Micro Cloud Foundry



Fuente, Elaboración propia.

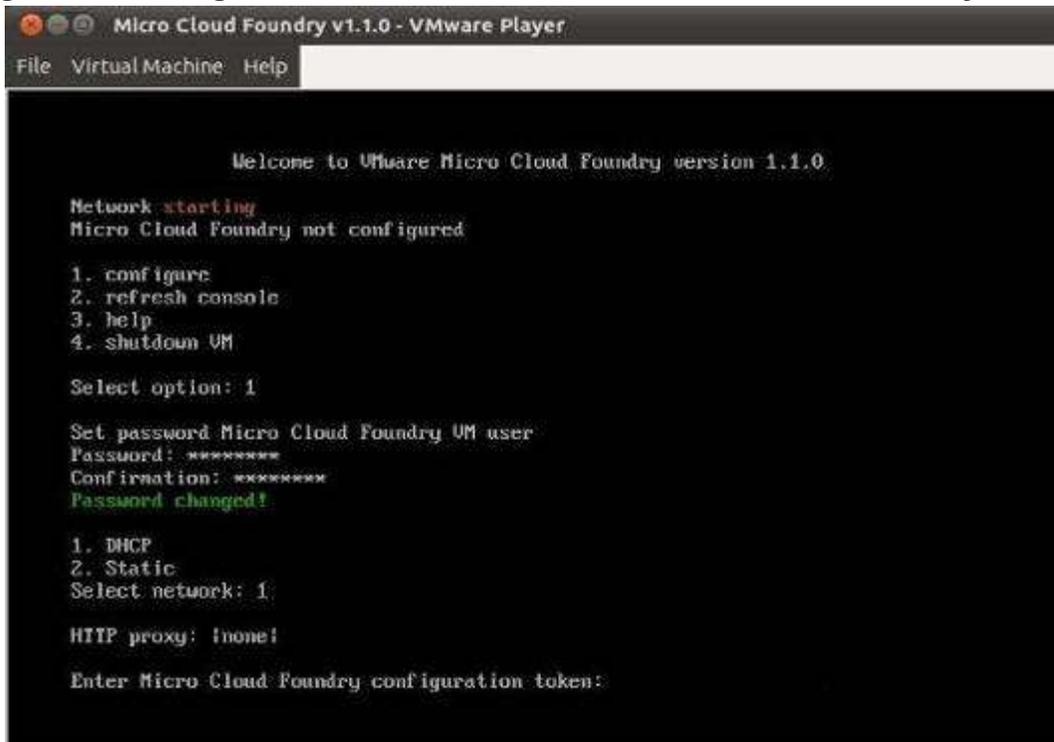
Luego se agrega el password para del usuario para la instancia virtual

Figura 45, Establecer Pass instancia MV Micro Cloud Foundry



Fuente, Elaboración propia.

Figura 46, Configuración de red instancia MV Micro Cloud Foundry



```
Micro Cloud Foundry v1.1.0 - VMware Player
File Virtual Machine Help

Welcome to VMware Micro Cloud Foundry version 1.1.0.

Network starting
Micro Cloud Foundry not configured

1. configure
2. refresh console
3. help
4. shutdown VM

Select option: 1

Set password Micro Cloud Foundry VM user
Password: xxxxxxxx
Confirmation: xxxxxxxx
Password changed!

1. DHCP
2. Static
Select network: 1

HTTP proxy: !none!

Enter Micro Cloud Foundry configuration token:
```

Fuente, Elaboración propia.

Una vez realizada la asignación del nuevo password, se configura los parámetros de red entre DHCP o estática, para la implementación de este prototipo la red esta configurada como DHCP.

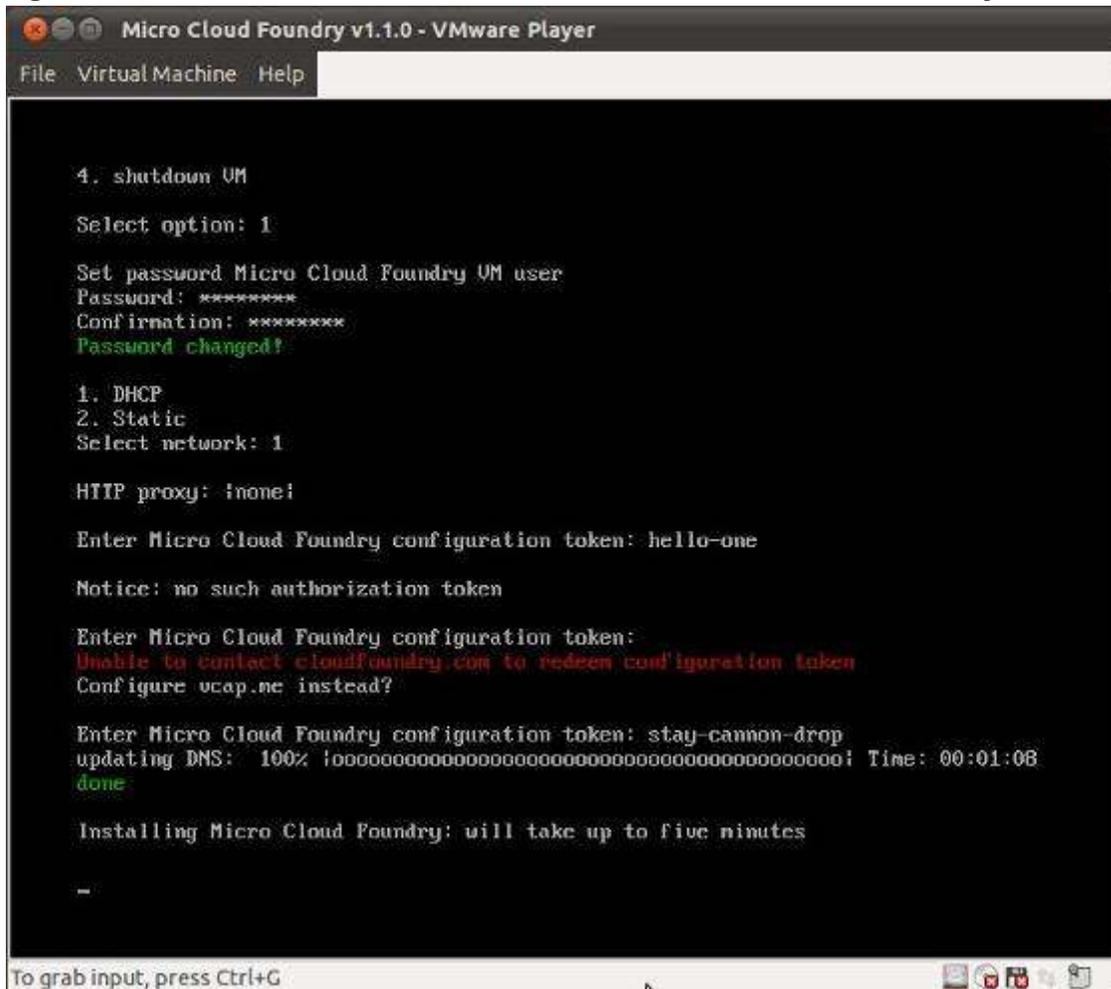
Luego se pregunta si para la conexión a Internet se esta detrás de un servidor proxy, si es asi se debe ingresar la dirección IP y puerto del proxy de lo contrario se deja en blanco y se tecléa enter para continuar.

Teniendo listo los parámetros de configuración de red, se procede a ingresar el Token para autenticar las credenciales de la cuenta única de Cloud Foundry, que se describió anteriormente.

En caso de ingresar un Token erróneo o no estar bien configurados los parámetros de red, la configuración de Micro Cloud Foundry no es exitosa.

(ver apartado 6.1.1.9. Dificultades en la implementación del Prototipo1.)

Figura 47, Autenticación de credenciales MV Micro Cloud Foundry



Fuente, Elaboración propia.

Si la conexión de red es exitosa se compara las credenciales del Token con la cuenta de cloudfoundry.com, actualizando el DNS personal e iniciando la instalación de Micro Cloud Foundry en la instancia virtual dentro de la infraestructura privada. Una vez finalizada la instalación con éxito se debe apreciar la siguiente pantalla.

Donde se observa la configuración presente de la plataforma como servicio (PaaS), en una nube privada.

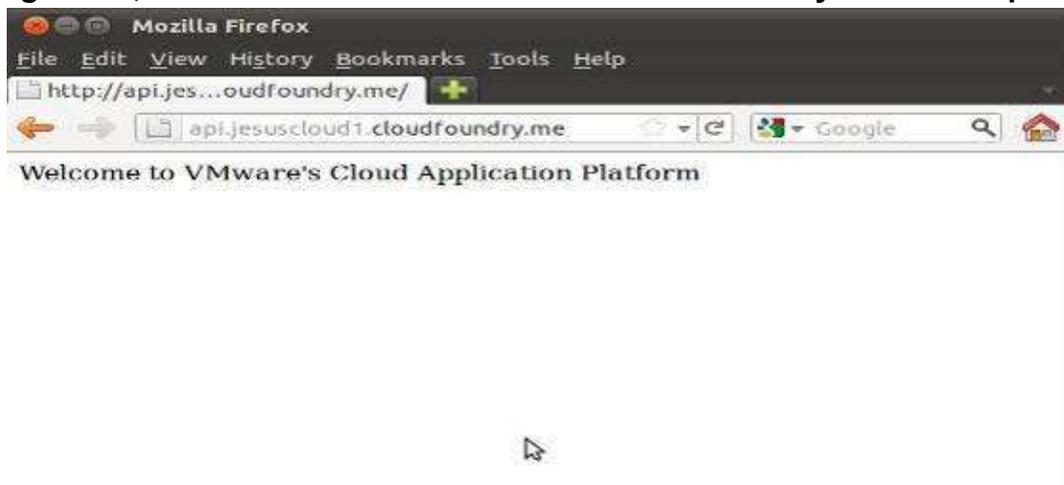
Figura 48, Instancia MV Micro Cloud Foundry en la nube privada



Fuente, Elaboración propia.

Al finalizar la configuración de la instancia virtual de la nube privada se verifica su funcionamiento, abriendo la URL de la nube privada en un navegador web.

Figura 49, Prueba funcionamiento Micro Cloud Foundry en la nube privada.



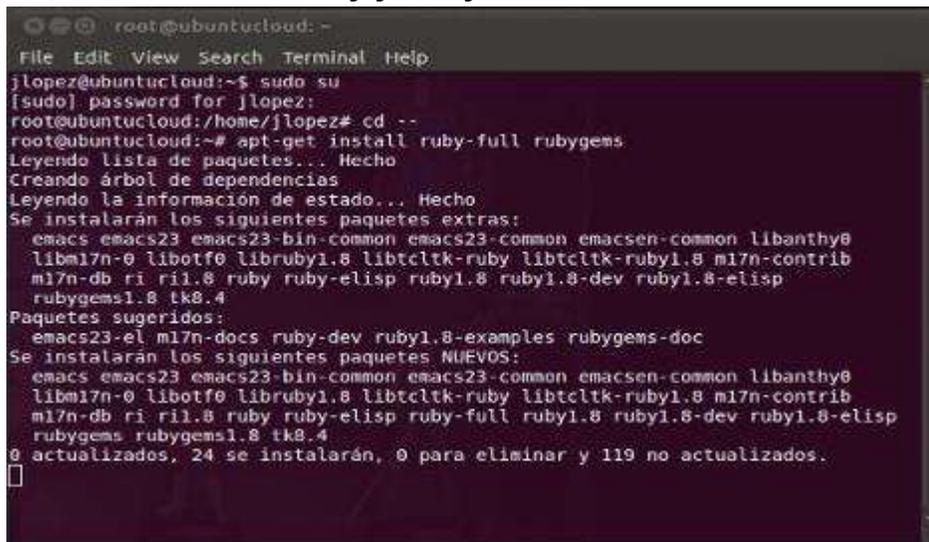
Fuente, Elaboración propia.

6.1.1.5. Instalación de Ruby, RubyGems y VMC (Cliente de línea de Comandos de Vmware)

Una vez instalada y configurada la instancia virtual de Micro Cloud Foundry plataforma como servicio en la nube privada, es necesario instalar y configurar Ruby, Ruby Gems y VMC (Cliente de línea de comandos de Vmware para Cloud Foundry), en los clientes (equipos remotos) para poder interactuar con la nube.

En la línea de comandos se ejecuta `$apt-get install ruby-full rubygems`, se inicia la instalación de Ruby con todos sus componentes.

Figura 50, Instalación de Ruby y RubyGems

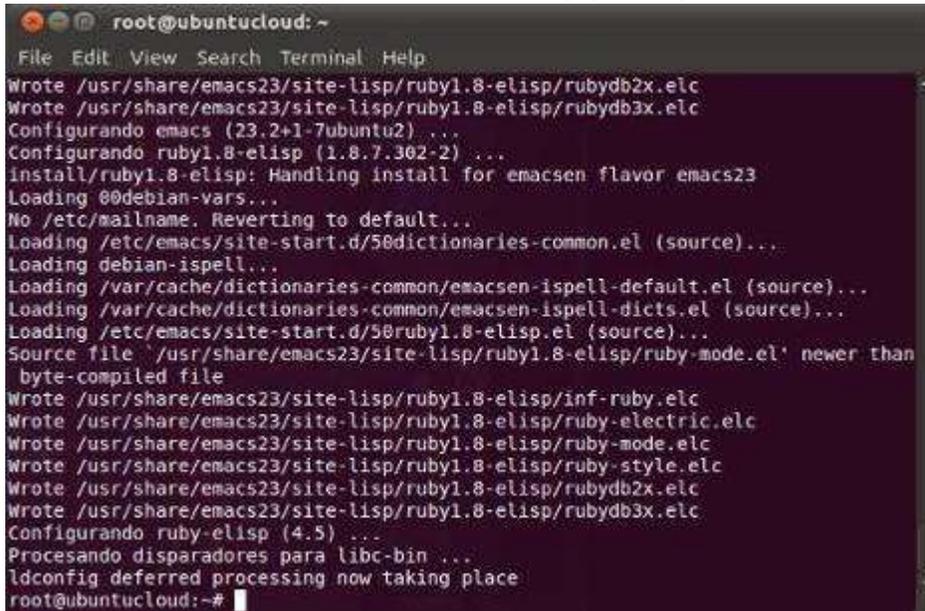


```
root@ubuntucloud:~#
File Edit View Search Terminal Help
jlopez@ubuntucloud:~$ sudo su
[sudo] password for jlopez:
root@ubuntucloud:/home/jlopez# cd --
root@ubuntucloud:~# apt-get install ruby-full rubygems
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
Se instalarán los siguientes paquetes extras:
  emacs emacs23 emacs23-bin-common emacs23-common emacsen-common libanthy8
  libm17n-0 libotf0 libruby1.8 libtcltk-ruby libtcltk-ruby1.8 m17n-contrib
  m17n-db ri ri1.8 ruby ruby-elisp ruby1.8 ruby1.8-dev ruby1.8-elisp
  rubygems1.8 tk8.4
Paquetes sugeridos:
  emacs23-el m17n-docs ruby-dev ruby1.8-examples rubygems-doc
Se instalarán los siguientes paquetes NUEVOS:
  emacs emacs23 emacs23-bin-common emacs23-common emacsen-common libanthy8
  libm17n-0 libotf0 libruby1.8 libtcltk-ruby libtcltk-ruby1.8 m17n-contrib
  m17n-db ri ri1.8 ruby ruby-elisp ruby-full ruby1.8 ruby1.8-dev ruby1.8-elisp
  rubygems rubygems1.8 tk8.4
0 actualizados, 24 se instalarán, 0 para eliminar y 119 no actualizados.
```

Fuente, Elaboración propia.

La instalación dura unos minutos dependiendo de la conexión a Internet, se espera a que finalice satisfactoriamente.

Figura 51, Finalización de Instalación de Ruby y RubyGems



```
root@ubuntucloud: ~
File Edit View Search Terminal Help
Wrote /usr/share/emacs23/site-lisp/ruby1.8-elisp/rubydb2x.elc
Wrote /usr/share/emacs23/site-lisp/ruby1.8-elisp/rubydb3x.elc
Configurando emacs (23.2+1-7ubuntu2) ...
Configurando ruby1.8-elisp (1.8.7.302-2) ...
install/ruby1.8-elisp: Handling install for emacsen flavor emacs23
Loading @debian-vars...
No /etc/mailname. Reverting to default...
Loading /etc/emacs/site-start.d/50dictionaries-common.el (source)...
Loading debian-ispell...
Loading /var/cache/dictionaries-common/emacs-en-ispell-default.el (source)...
Loading /var/cache/dictionaries-common/emacs-en-ispell-dicts.el (source)...
Loading /etc/emacs/site-start.d/50ruby1.8-elisp.el (source)...
Source file '/usr/share/emacs23/site-lisp/ruby1.8-elisp/ruby-mode.el' newer than
byte-compiled file
Wrote /usr/share/emacs23/site-lisp/ruby1.8-elisp/inf-ruby.elc
Wrote /usr/share/emacs23/site-lisp/ruby1.8-elisp/ruby-electric.elc
Wrote /usr/share/emacs23/site-lisp/ruby1.8-elisp/ruby-mode.elc
Wrote /usr/share/emacs23/site-lisp/ruby1.8-elisp/ruby-style.elc
Wrote /usr/share/emacs23/site-lisp/ruby1.8-elisp/rubydb2x.elc
Wrote /usr/share/emacs23/site-lisp/ruby1.8-elisp/rubydb3x.elc
Configurando ruby-elisp (4.5) ...
Procesando disparadores para libc-bin ...
ldconfig deferred processing now taking place
root@ubuntucloud:~#
```

Fuente, Elaboración propia.

Finalizada la instalación de Ruby con todos sus paquetes, se procede a instalar Sinatra un framework para aplicaciones Ruby, ya que este va ser utilizado en la pruebas.

Figura 52, Instalación de Sinatra Framework



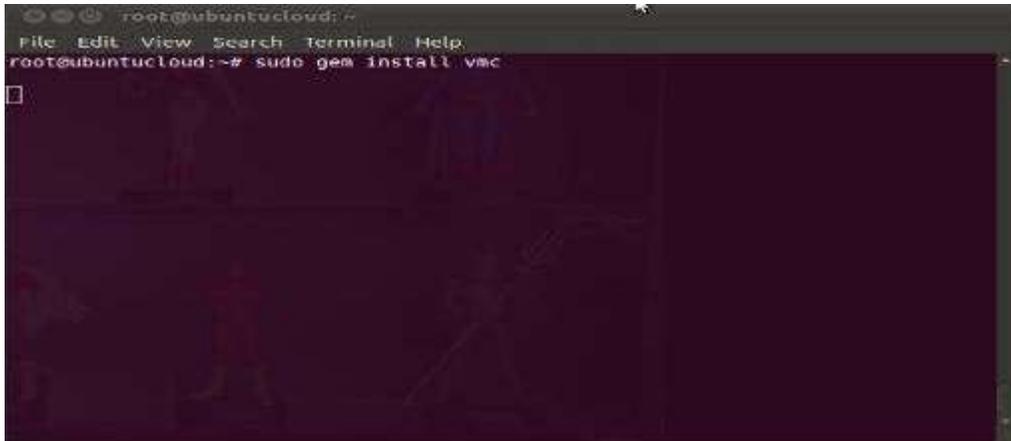
```
jlopez@ubuntucloud: ~
File Edit View Search Terminal Help
jlopez@ubuntucloud:~$ sudo gem install sinatra
[sudo] password for jlopez:
Successfully installed sinatra-1.3.2
1 gem installed
Installing ri documentation for sinatra-1.3.2...
Installing RDoc documentation for sinatra-1.3.2...
jlopez@ubuntucloud:~$
```

Fuente, Elaboración propia.

Ahora lo mas importante es la correcta instalación de vmc(linea de comando cliente de Vmware), para poder interactuar con la plataforma como servicio privada.

Para iniciar la instalación de VMC, en la línea de comandos se ejecuta `$sudo gem install vmc`

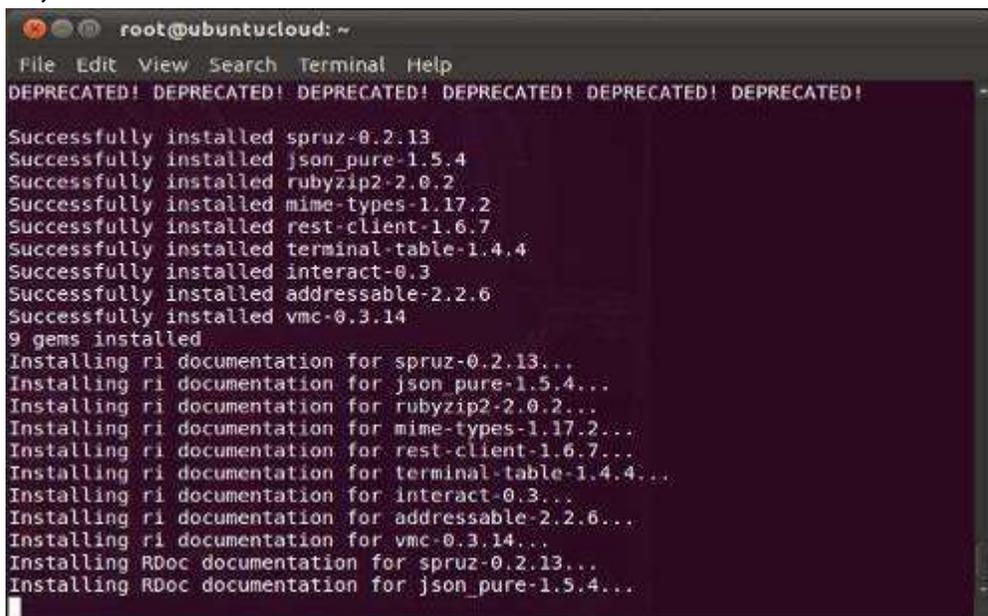
Figura 53, Instalación de VMC



Fuente, Elaboración propia.

Se espera que se inicie la descarga de paquetes y la instalación de vmc.

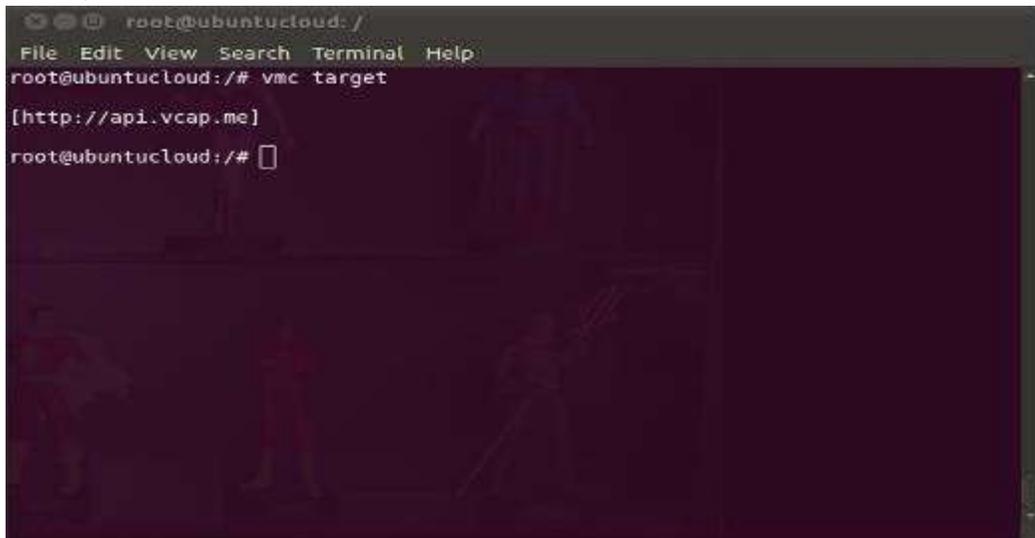
Figura 54, Finalización de Instalación de VMC.



Fuente, Elaboración propia.

Una vez finalizada la instalación. Se comprueba el funcionamiento de vmc con el comando `$vmc target`, debe devolver el formato de URL de una instancia en Cloud Foundry, [<http://api.vcap.me>].

Figura 55, Prueba de funcionamiento de VMC.

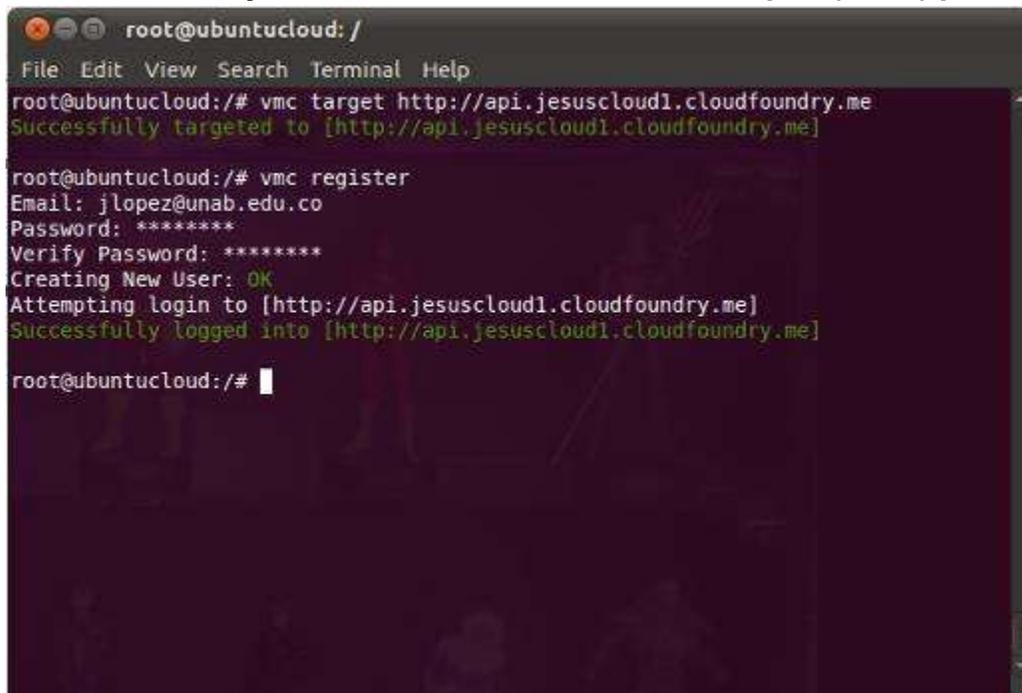


```
root@ubuntucloud: /
File Edit View Search Terminal Help
root@ubuntucloud:/# vmc target
[http://api.vcap.me]
root@ubuntucloud:/#
```

Fuente, Elaboración propia.

6.1.1.6. Conexión y creación de usuario en el prototipo1 (PaaS) privado. Luego de tener la instancia virtual de la plataforma como servicio activa y corriendo, las herramientas del cliente funcionando, se realiza la conexión a la nube por medio del comando **\$vmc target** “**la URL de la nube**”, en este prototipo la URL es **<http://api.jesuscloud1.cloudfoundry.me>** Luego de establecer la conexión a la plataforma en la nube con éxito, se realiza el registro de un usuario nuevo con el comando **\$vmc register**, el nombre de usuario debe ser una cuenta de correo electrónico, se ingresa el password y se verifica, se espera la confirmación exitosa de la creación del nuevo usuario con el cual se va a realizar la pruebas en la plataforma como servicio en la nube.

Figura 56, Conexión y creación de usuario en el Prototipo1 (PaaS) privado.

A terminal window titled 'root@ubuntucloud: /' with a menu bar (File, Edit, View, Search, Terminal, Help). The terminal shows the following commands and output:

```
root@ubuntucloud:~# vmc target http://api.jesuscloud1.cloudfoundry.me
Successfully targeted to [http://api.jesuscloud1.cloudfoundry.me]

root@ubuntucloud:~# vmc register
Email: jlopez@unab.edu.co
Password: *****
Verify Password: *****
Creating New User: OK
Attempting login to [http://api.jesuscloud1.cloudfoundry.me]
Successfully logged into [http://api.jesuscloud1.cloudfoundry.me]

root@ubuntucloud:~#
```

Fuente, Elaboración propia.

6.1.1.7. Uso y Pruebas del Prototipo1 (PaaS) privado.

Después de crear el usuario, se inicia sesión en la instancia de la nube Privada, con el comando **. vmc login**

Figura 57, Inicio de sesión en la PaaS privada

```
jlopez@ubuntucloud: /
File Edit View Search Terminal Help
ACTIVO DIFUSIÓN MULTICAST MTU:1500 Métrica:1
Paquetes RX:87 errores:0 perdidos:0 overruns:0 frame:0
Paquetes TX:113 errores:0 perdidos:0 overruns:0 carrier:0
colisiones:0 long.colaTX:1000
Bytes RX:42495 (42.4 KB) TX bytes:28071 (28.0 KB)

jlopez@ubuntucloud:/$ ping www.google.com
PING www.l.google.com (190.248.1.21) 56(84) bytes of data:
64 bytes from cable190-248-1-21.une.net.co (190.248.1.21): icmp_req=1 ttl=56 time=24.
1 ms
64 bytes from cable190-248-1-21.une.net.co (190.248.1.21): icmp_req=2 ttl=56 time=23.
7 ms
^C
--- www.l.google.com ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1001ms
rtt min/avg/max/mdev = 23.789/23.952/24.115/0.163 ms
jlopez@ubuntucloud:/$ vmc target

[http://api.jesuscloud1.cloudfoundry.me]

jlopez@ubuntucloud:/$ vmc login jlopez@unab.edu.co
Attempting login to [http://api.jesuscloud1.cloudfoundry.me]
Password: *****
Successfully logged into [http://api.jesuscloud1.cloudfoundry.me]

jlopez@ubuntucloud:/$
```

Fuente, Elaboración propia.

Después de autenticado el usuario ya esta listo para el despliega de aplicaciones en la nube privada.

- Despliegue de una aplicación sencilla (sin servicios).

Para el despliegue de la aplicación sencilla llamada *Hello2* , en un nuevo directorio se crea el archivo que se va a desplegar en la nube. `jlopez@ubuntucloud:/$ mkdir hello2`, dentro del directorio se crea un archivo Ruby llamado `hello2.rb` con el siguiente código

```
require 'rubygems'
require 'sinatra'
get '/' do
  "Hello from Cloud Foundry"
End
```

Para correr esta aplicación en la instancia privada de Micro Cloud Foundry, desde una ventana de terminal / línea de comandos, se ubica dentro del directorio donde se guardó "hello2.rb":

`jlopez@ubuntucloud:/$ cd hello2`, se ejecuta el comando "vmc push hello2" y se presiona enter

```
jlopez@ubuntucloud:/hello2$ vmc push hello2
```

```
Would you like to deploy from the current directory? [Yn]: y
```

A la pregunta si se desea desplegar desde el directorio actual, se responde Y , por defecto y se presiona enter para continuar

```
Application Deployed URL [hello2.jesuscloud1.cloudfoundry.me]:
```

Esto determina la URL donde la aplicación se a asignado, enter en este prototipo URL hello2.jesuscloud1.cloudfoundry.me

```
Detected a Sinatra Application, is this correct? [Yn]:y
```

El cliente vmc ha detectado que hello2.rb utiliza Sinatra framework, enter

```
Memory Reservation (64M, 128M, 256M, 512M, 1G) [128M]:
```

Esta es la asignación de memoria, el valor predeterminado para Sinatra es 128M para una aplicación Java sería 512M, se escribe la memoria que se desea asignar o se deja en blanco y se presiona enter para que tome por defecto que es 128M.

```
Creating Application: OK
```

```
Would you like to bind any services to 'hello'2? [yN]:n
```

Se requiere algún servicio para la aplicación, en este caso se responde "No", no se requiere que se unan servicios a la aplicación. Enter

```
Uploading Application:
```

```
Checking for available resources: OK
```

```
Packing application: OK
```

```
Uploading (OK): OK
```

```
Push Status: OK
```

```
Staging Application: OK
```

```
Starting Application: OK
```

La aplicación se ha desplegado con éxito, se debe ver la siguiente pantalla:

Figura 58, Despliegue de la aplicación Hello2 en Prototipo1 (PaaS) privada



```
jlopez@ubuntucloud: /hello2
File Edit View Search Terminal Help
jlopez@ubuntucloud:/$ ls
appscale  etc          initrd.img.old  lost+found  proc        srv          usr
bin       hello2       lib             media       root        sys          var
boot      home         lib32           mnt         sbin        logs         vmlinuz
dev       initrd.img  lib64           opt         selinux    ubuntu      vmlinuz.old
jlopez@ubuntucloud:/$ cd hello2/
jlopez@ubuntucloud:/hello2$ ls
hello2.rb
jlopez@ubuntucloud:/hello2$ vmc push hello2
Would you like to deploy from the current directory? [Yn]: y
Application Deployed URL [hello2.jesuscloud1.cloudfoundry.me]:
Detected a Sinatra Application, is this correct? [Yn]: y
Memory Reservation (64M, 128M, 256M, 512M, 1G) [128M]:
Creating Application: OK
Would you like to bind any services to 'hello2'? [yN]: n
Uploading Application:
  Checking for available resources: OK
  Packing application: OK
  Uploading (0K): OK
Push Status: OK
Staging Application: OK
Starting Application: OK
jlopez@ubuntucloud:/hello2$
```

Fuente, Elaboración propia.

Para ver la aplicación en ejecución, se abre un navegador web y se va a la URL donde se desplegó la aplicación en la nube.

Figura 59, Aplicación Hello2 en ejecución Prototipo1 (PaaS) privada



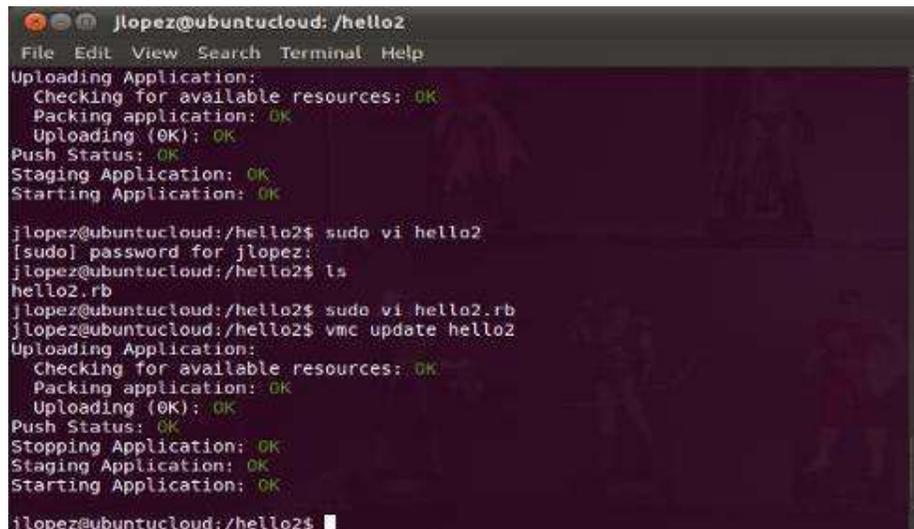
Fuente, Elaboración propia.

Se realizan modificaciones o actualizaciones a la aplicación, como esta aplicación ya quedo registrada dentro de la nube, para poder ejecutar las modificaciones se

debe realizar con el siguiente comando, `jlopez@ubuntucloud:/hello2$ vmc update hello2`

Y la aplicación nuevamente es desplegada en plataforma como servicio (PaaS) en la nube privada.

Figura 60, Actualización de una aplicación de prueba en Prototipo1 (PaaS) privada.



```
jlopez@ubuntucloud:/hello2
File Edit View Search Terminal Help
Uploading Application:
  Checking for available resources: OK
  Packing application: OK
  Uploading (OK): OK
Push Status: OK
Staging Application: OK
Starting Application: OK

jlopez@ubuntucloud:/hello2$ sudo vi hello2
[sudo] password for jlopez:
jlopez@ubuntucloud:/hello2$ ls
hello2.rb
jlopez@ubuntucloud:/hello2$ sudo vi hello2.rb
jlopez@ubuntucloud:/hello2$ vmc update hello2
Uploading Application:
  Checking for available resources: OK
  Packing application: OK
  Uploading (OK): OK
Push Status: OK
Stopping Application: OK
Staging Application: OK
Starting Application: OK
jlopez@ubuntucloud:/hello2$
```

Fuente, Elaboración propia.

Nuevamente se abre el navegador web y se comprueba la aplicación en ejecución con sus modificaciones.

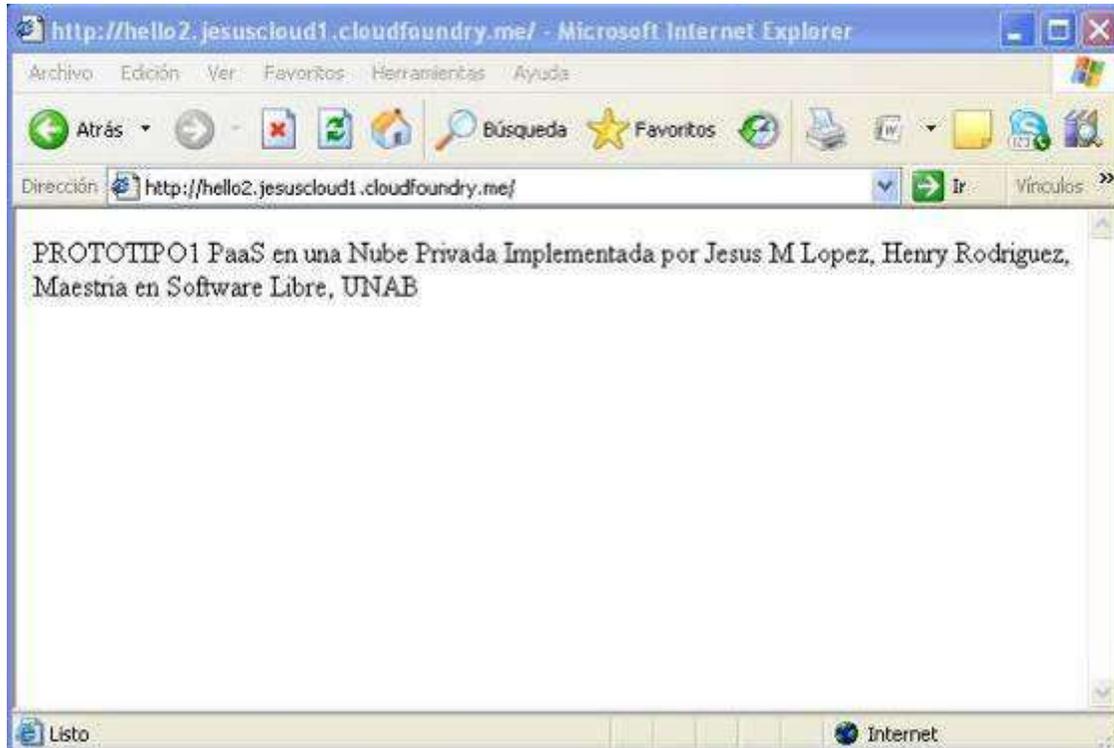
Figura 61, Aplicación Hello2 modificada en ejecución Prototipo1 (PaaS) privada



Fuente, Elaboración propia.

Se puede observar la aplicación en ejecución desde un equipo con S.O. Windows XP y navegador web Internet Explorer 7.

Figura 62, Aplicación Hello2 en ejecución desde Windows Prototipo1 (PaaS) privada



Fuente, Elaboración propia.

- Despliegue de una aplicación con servicios.

En la prueba anterior se implemento una aplicación sencilla sin servicio, una de las grandes ventajas de Micro Cloud Foundry es que permite el despliegue de aplicaciones con múltiples servicios.

Para verificar que servicios se tienen disponibles en la plataforma como servicio (PaaS) en la nube privada, en la línea de comandos se ejecuta el comando **vmc services**, al ejecutar el comando devuelve los servicios disponibles en dos tablas: la primera tabla muestra los tipos de servicios disponibles que se pueden utilizar para las aplicaciones, la segunda tabla muestra los servicios que ya se han creado una instancia.

Figura 63, Visualización de los servicios disponibles en Prototipo1 (PaaS) privada

```
jlopez@ubuntucloud: /
File Edit View Search Terminal Help
jlopez@ubuntucloud:/hello2$ cd ..
jlopez@ubuntucloud:/$ vmc services

===== System Services =====

+-----+-----+-----+
| Service | Version | Description |
+-----+-----+-----+
| mongodb | 1.8     | MongoDB NoSQL store |
| mysql   | 5.1     | MySQL database service |
| postgresql | 9.0   | PostgreSQL database service (vFabric) |
| rabbitmq | 2.4    | RabbitMQ messaging service |
| redis   | 2.2     | Redis key-value store service |
+-----+-----+-----+

===== Provisioned Services =====

+-----+-----+
| Name | Service |
+-----+-----+
| n    | mysql   |
+-----+-----+

jlopez@ubuntucloud:/$
```

Fuente, Elaboración propia.

Para crear una nueva instancia de un servicio se ejecuta el comando **vmc create-service** y se relaciona con el servicio que se desea, asignándole un nombre a la instancia del servicio para esta prueba se creo la instancia `mysql-js` para el servicio de la base de datos de `mysql`. **vmc create-service mysql mysql-sj**, al comprobar los servicios asignados a una instancia, se puede ver el servicio que se creo.

Figura 64, Creación y visualización de los servicios disponibles en Prototipo1 (PaaS) privada

```

jlopez@ubuntucloud: /
File Edit View Search Terminal Help

jlopez@ubuntucloud:/$ vmc create-service mysql mysql-js
Creating Service: OK

jlopez@ubuntucloud:/$ vmc services

===== System Services =====
+-----+-----+-----+
| Service | Version | Description |
+-----+-----+-----+
| mongodb | 1.8     | MongoDB NoSQL store |
| mysql   | 5.1     | MySQL database service |
| postgresql | 9.0    | PostgreSQL database service (vFabric) |
| rabbitmq | 2.4     | RabbitMQ messaging service |
| redis   | 2.2     | Redis key-value store service |
+-----+-----+-----+

===== Provisioned Services =====
+-----+-----+
| Name | Service |
+-----+-----+
| mysql-js | mysql |
| n       | mysql |
+-----+-----+

jlopez@ubuntucloud:/$

```

Fuente, Elaboración propia.

Si la aplicación que se va a desplegar requiere de mas de un servicio se crean las instancias necesarias para los servicios requeridos.

```

vmc create-service mysql mysql-jsn
vmc create-service mongodb mongodb-js
vmc create-service rabbitmq rabbitmq-js

```

Al igual que la prueba anterior se despliega esta aplicación en la nube, pero para esta prueba, la pregunta donde se requiere unirse a algún servicio, se responde si, y se escoge el servicio requerido.

```

prompt$ vmc push test1
Would you like to deploy from the current directory? [Yn]: y
Application Deployed URL: [test1.jesuscloud1.cloudfoundry.me]:
Detected a Sinatra Application, is this correct? [Yn]:y
Memory Reservation (64M, 128M, 256M, 512M or 1G) [128M]:
Creating Application: OK
Would you like to bind any services to 'test1'? [yN]: y
Would you like to use an existing provisioned service [yN]? y
The following provisioned services are available:
1. mysql-js
2. n
Please select one you wish to provision: 1

```

Binding Service [mysql-js]: OK
Uploading Application:
Checking for available resources: OK
Packing application: OK
Uploading (1K): OK
Push Status: OK
Staging Application: OK
Starting Application: OK

Figura 65, Despliegue de la aplicación con servicios en Prototipo1 (PaaS) privada.



```
root@ubuntucloud: /hello2
File Edit View Search Terminal Help

root@ubuntucloud:/hello2# vmc push test1
Would you like to deploy from the current directory? [Yn]: y
Application Deployed URL [test1.jesuscloud1.cloudfoundry.me]:
Detected a Sinatra Application, is this correct? [Yn]: y
Memory Reservation (64M, 128M, 256M, 512M, 1G) [128M]:
Creating Application: OK
Would you like to bind any services to 'test1'? [yN]: y
Would you like to use an existing provisioned service? [yN]: y
The following provisioned services are available
1: mysql-js
2: n
Please select one you wish to use: 1
Binding Service [mysql-js]: OK
Uploading Application:
  Checking for available resources: OK
  Packing application: OK
  Uploading (1K): OK
Push Status: OK
Staging Application: OK
Starting Application: OK

root@ubuntucloud:/hello2#
```

Fuente, Elaboración propia.

6.1.1.8. Administración de las aplicaciones.

Para saber que aplicaciones se han crea en la nube y administrar su ciclo de vida, la línea de comandos del cliente VMC permite realizar la estas actividades.

Para conocer el nombre exacto de las aplicaciones y saber cuales están corriendo, el comando **vmc apps**, permite visualizar por medio de una tabla el nombre de la aplicación, el estado actual del ciclo de vida (Corriendo,detenida), la URL de la instancia en la nube donde esta corriendo la aplicación y los servicios que usa.

Figura 66, Administración de las aplicaciones en Prototipo1 (PaaS) privada.

```

jlopez@ubuntucloud: /hello2
File Edit View Search Terminal Help
Staging Application: OK
Starting Application: OK

jlopez@ubuntucloud: /hello2$ sudo vi hello2.rb
jlopez@ubuntucloud: /hello2$ vmc update hello2
Uploading Application:
  Checking for available resources: OK
  Packing application: OK
  Uploading (8K): OK
Push Status: OK
Stopping Application: OK
Staging Application: OK
Starting Application: OK

jlopez@ubuntucloud: /hello2$ vmc apps

+-----+-----+-----+-----+-----+
| Application | # | Health | URLS | Services |
+-----+-----+-----+-----+-----+
| hello      | 1 | STOPPED | hello.jesuscloud1.cloudfoundry.me | n |
| hello2     | 1 | RUNNING | hello2.jesuscloud1.cloudfoundry.me |  |
+-----+-----+-----+-----+-----+

jlopez@ubuntucloud: /hello2$

```

Fuente, Elaboración propia.

Para detener una aplicación se usa el comando **vmc stop “nombre de la aplicación”** y si se dese eliminar la aplicación de la instancia en la nubecon el comando **vmc delete “nombre de la aplicación”**

Figura 67, Detener una aplicación en el Prototipo1 (PaaS) privada.

```

jlopez@ubuntucloud: /hello2
File Edit View Search Terminal Help
| redis      | 2.2 | Redis key-value store service |
+-----+-----+-----+-----+
===== Provisioned Services =====
+-----+-----+
| Name | Service |
+-----+-----+
| n    | mysql  |
+-----+-----+

jlopez@ubuntucloud: /hello2$ vmc apps

+-----+-----+-----+-----+-----+
| Application | # | Health | URLS | Services |
+-----+-----+-----+-----+-----+
| hello      | 1 | STOPPED | hello.jesuscloud1.cloudfoundry.me | n |
| hello2     | 1 | RUNNING | hello2.jesuscloud1.cloudfoundry.me |  |
+-----+-----+-----+-----+-----+

jlopez@ubuntucloud: /hello2$ vmc stop hello2
Stopping Application: OK

jlopez@ubuntucloud: /hello2$

```

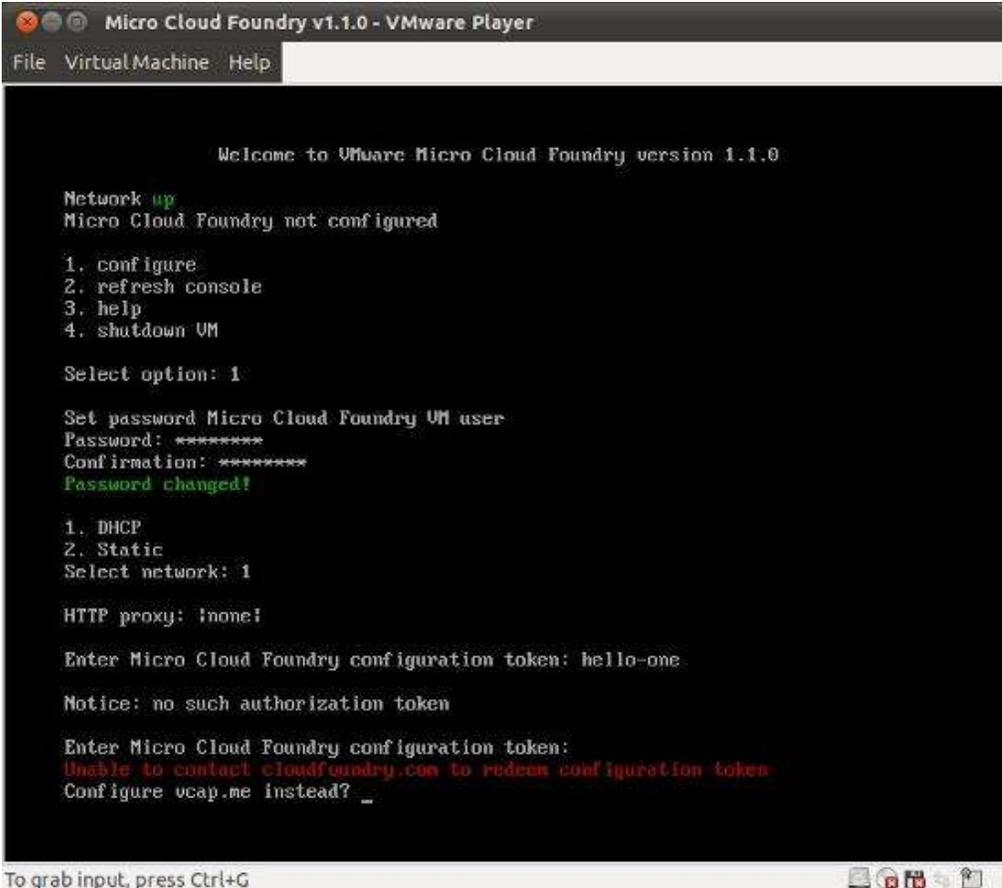
Fuente, Elaboración propia.

6.1.1.9. Dificultades en la implementación del Prototipo 1.

En un principio y según la documentación de Cloud Foundry la implementación de Micro Cloud Foundry dentro de una nube privada parecía sencillo y fácil de desplegar, pero durante el proceso de implementación se encontraron algunas dificultades las cuales se comenta a continuación ya que sirven como apoyo a la gestión del conocimiento en futuros trabajos.

- Error de autenticación.

Al realizar la configuración inicial de Micro Cloud Foundry, al ingresar el token para autenticar la cuenta única, se genera el error **“Unable to contact cloudfoundry.com to redeem configuration token”** Después de varios intentos se seguía presentando el error, se reviso los parámetros de configuración de la red cambiando modo Bridge a NAT de la MV y no había conexión, se identifico que la maquina virtual no tenia conexión de red, al consultar la pagina de soporte de Cloud Foundry se encontraron varios aportes, entre ellos cambiar los parámetros de red de la MV a NAT, cosa que ya se había intentado sin éxito.



```
Micro Cloud Foundry v1.1.0 - VMware Player
File Virtual Machine Help

Welcome to VMware Micro Cloud Foundry version 1.1.0

Network up
Micro Cloud Foundry not configured

1. configure
2. refresh console
3. help
4. shutdown VM

Select option: 1

Set password Micro Cloud Foundry VM user
Password: *****
Confirmation: *****
Password changed!

1. DHCP
2. Static
Select network: 1

HTTP proxy: Inone!

Enter Micro Cloud Foundry configuration token: hello-one

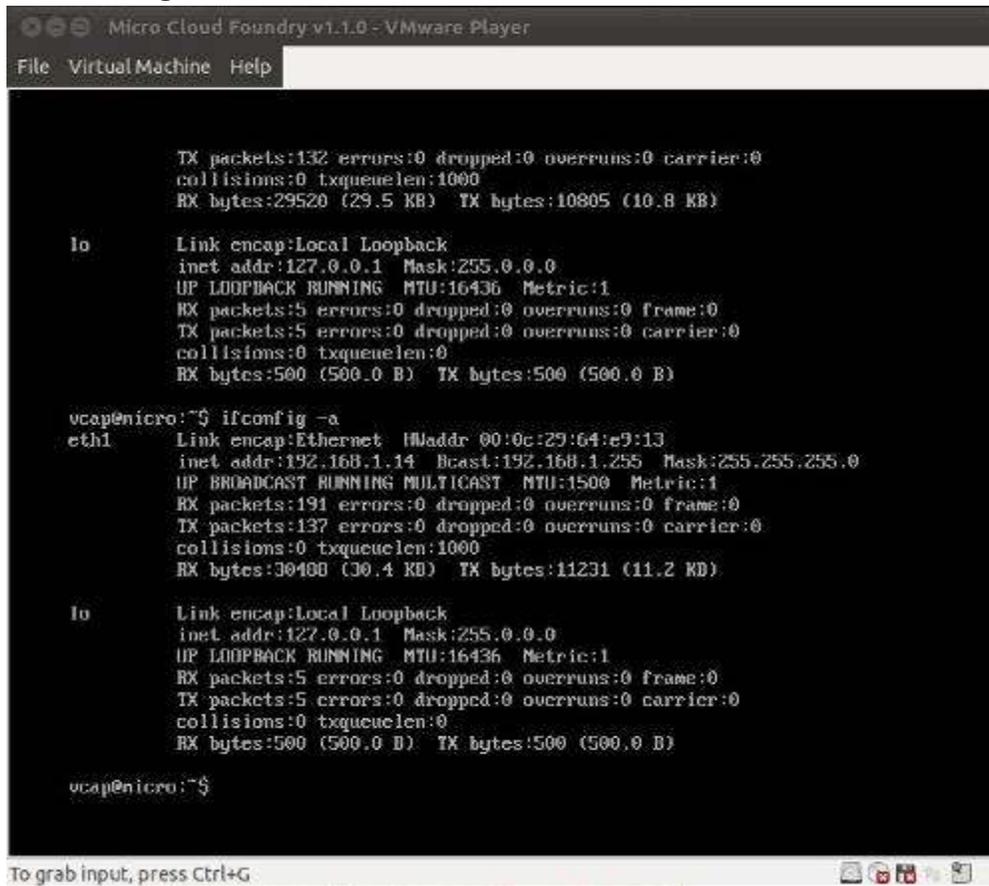
Notice: no such authorization token

Enter Micro Cloud Foundry configuration token:
Unable to contact cloudfoundry.com to redeem configuration token
Configure vcap.me instead? _

To grab input, press Ctrl+G
```

Figura 68, Error de autenticación Prototipo1 (PaaS) privada

Figura 70, Configuración manual de la interface de red en la MV



```
Micro Cloud Foundry v1.1.0 - VMware Player
File Virtual Machine Help

TX packets:132 errors:0 dropped:0 overruns:0 carrier:0
collisions:0 txqueuelen:1000
RX bytes:29520 (29.5 KB) TX bytes:10805 (10.8 KB)

io      Link encap:Local Loopback
inet addr:127.0.0.1 Mask:255.0.0.0
UP LOOPBACK RUNNING MTU:16436 Metric:1
RX packets:5 errors:0 dropped:0 overruns:0 frame:0
TX packets:5 errors:0 dropped:0 overruns:0 carrier:0
collisions:0 txqueuelen:0
RX bytes:500 (500.0 B) TX bytes:500 (500.0 B)

vcap@micro:~$ ifconfig -a
eth1    Link encap:Ethernet HWaddr 00:0c:29:64:e9:13
inet addr:192.168.1.14 Bcast:192.168.1.255 Mask:255.255.255.0
UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
RX packets:191 errors:0 dropped:0 overruns:0 frame:0
TX packets:137 errors:0 dropped:0 overruns:0 carrier:0
collisions:0 txqueuelen:1000
RX bytes:30400 (30.4 KB) TX bytes:11231 (11.2 KB)

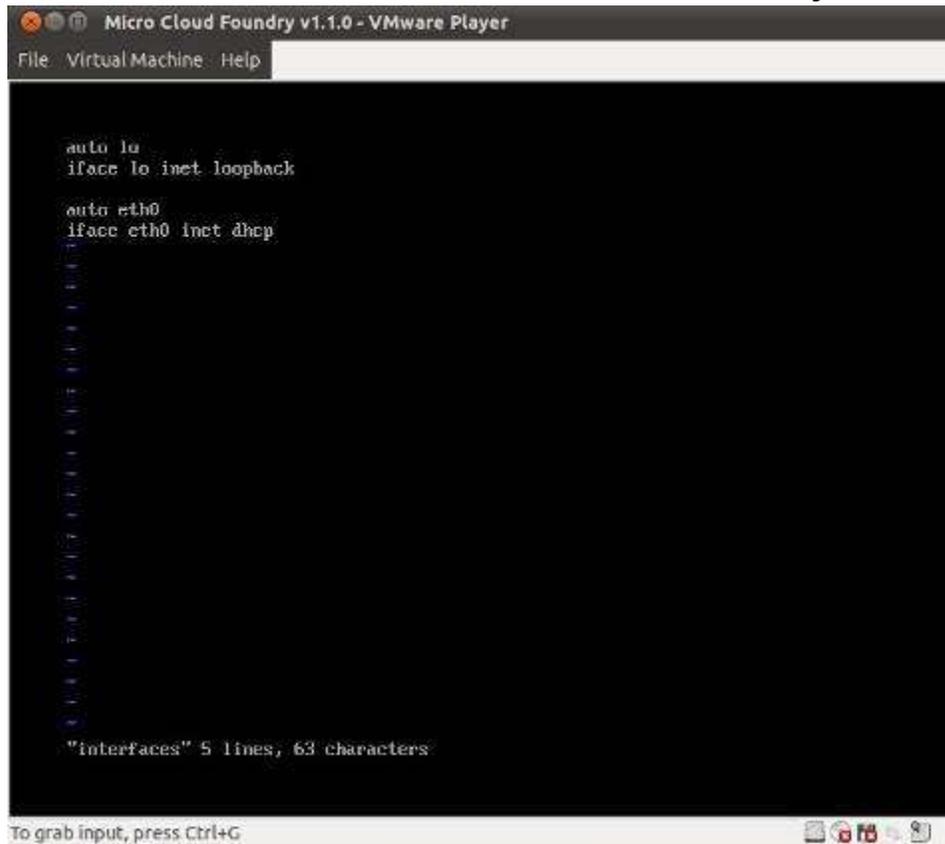
io      Link encap:Local Loopback
inet addr:127.0.0.1 Mask:255.0.0.0
UP LOOPBACK RUNNING MTU:16436 Metric:1
RX packets:5 errors:0 dropped:0 overruns:0 frame:0
TX packets:5 errors:0 dropped:0 overruns:0 carrier:0
collisions:0 txqueuelen:0
RX bytes:500 (500.0 B) TX bytes:500 (500.0 B)

vcap@micro:~$
```

Fuente, Elaboración propia.

se procedió a revisar el archivo de configuración de la interface red **\$/etc/network/interfaces**, y efectivamente estaba cargando por eth0 se edito y se cambio la interface de red a eth1.

Figura 71, Archivo Interface en la MV de Micro Cloud Foundry



```
Micro Cloud Foundry v1.1.0 - VMware Player
File Virtual Machine Help

auto lo
iface lo inet loopback

auto eth0
iface eth0 inet dhcp

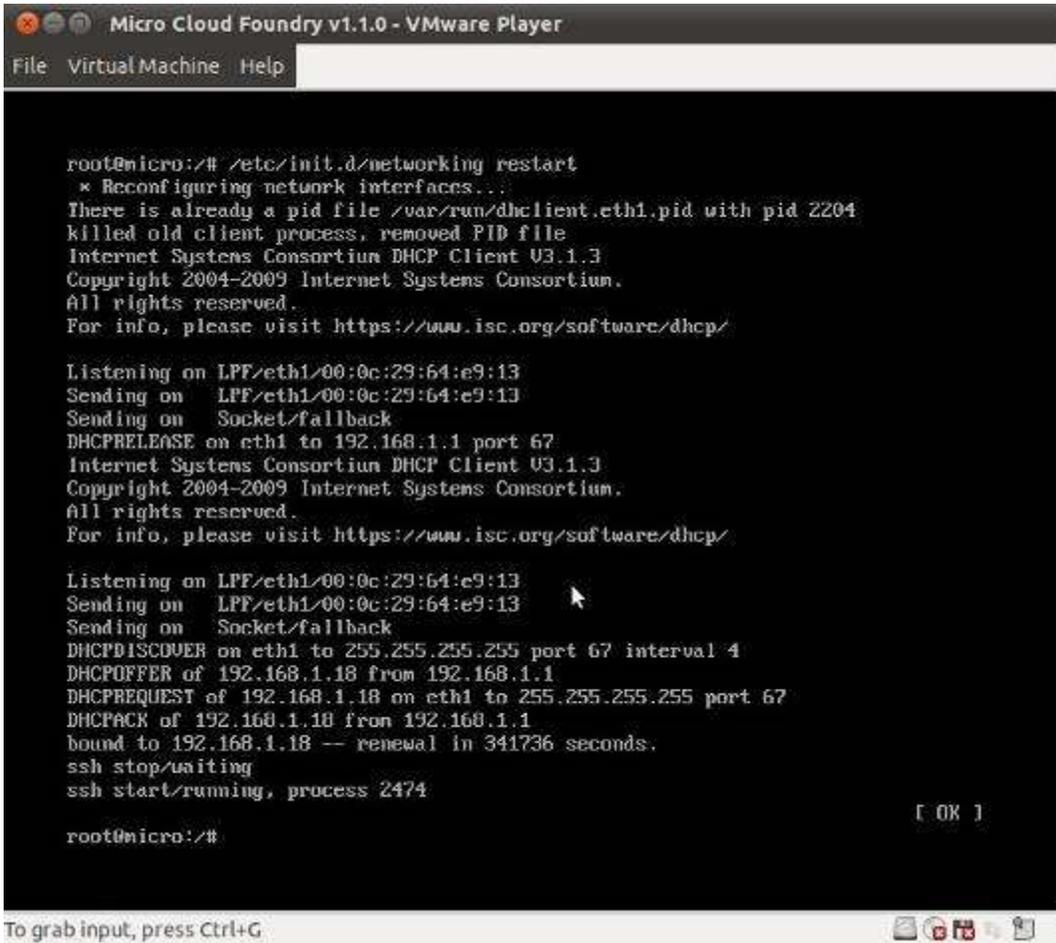
"interfaces" 5 lines, 63 characters

To grab input, press Ctrl+G
```

Fuente, Elaboración propia.

Se reinicio le demonio red `$/etc/init.d/networking restart`, se comprobó que la instancia virtual conectara y tomara dirección IP.

Figura 72, Reinicio de los servicios de red de la MV



```
Micro Cloud Foundry v1.1.0 - VMware Player
File Virtual Machine Help

root@micro:~# /etc/init.d/networking restart
 * Reconfiguring network interfaces...
There is already a pid file /var/run/dhclient.eth1.pid with pid 2204
killed old client process, removed PID file
Internet Systems Consortium DHCP Client U3.1.3
Copyright 2004-2009 Internet Systems Consortium.
All rights reserved.
For info, please visit https://www.isc.org/software/dhcp/

Listening on LPP/eth1/00:0c:29:64:e9:13
Sending on LPP/eth1/00:0c:29:64:e9:13
Sending on Socket/fallback
DHCPRELEASE on eth1 to 192.168.1.1 port 67
Internet Systems Consortium DHCP Client U3.1.3
Copyright 2004-2009 Internet Systems Consortium.
All rights reserved.
For info, please visit https://www.isc.org/software/dhcp/

Listening on LPP/eth1/00:0c:29:64:e9:13
Sending on LPP/eth1/00:0c:29:64:e9:13
Sending on Socket/fallback
DHCPDISCOVER on eth1 to 255.255.255.255 port 67 interval 4
DHCPOFFER of 192.168.1.18 from 192.168.1.1
DHCPREQUEST of 192.168.1.18 on eth1 to 255.255.255.255 port 67
DHCPACK of 192.168.1.18 from 192.168.1.1
bound to 192.168.1.18 -- renewal in 341736 seconds.
ssh stop/waiting
ssh start/running, process 2474

root@micro:~#
```

Fuente, Elaboración propia.

- Error de VMC (Línea de comando del cliente Vmware)

Después de instalar VMC como se comentó anteriormente, se realiza la prueba de funcionamiento con el comando **\$vmc target**, pero la consola devuelve comando desconocido, al consultar con los foros de soporte, se encuentra que hay que direccionar el directorio donde se encuentran los paquetes con el comando **\$export PATH=\$PATH:/var/lib/gems/1.8/bin/**, luego comprobamos nuevamente el comando **\$vmc target** y tiene que devolver el formato de URL de Micro Cloud Foundry.

Figura 73, Error después de instalar VMC, en el cliente

```
root@ubuntucloud: /
File Edit View Search Terminal Help
jlopez@ubuntucloud:/var/lib/gems$ cd 1.8
jlopez@ubuntucloud:/var/lib/gems/1.8$ ld
ld: no input files
jlopez@ubuntucloud:/var/lib/gems/1.8$ ls
bin cache doc gems specifications
jlopez@ubuntucloud:/var/lib/gems/1.8$ cd bin
jlopez@ubuntucloud:/var/lib/gems/1.8/bin$ ls
edit_json.rb enum prettify_json.rb restclient update_rubygems vmc
jlopez@ubuntucloud:/var/lib/gems/1.8/bin$ cd --
jlopez@ubuntucloud:~$ cd ..
jlopez@ubuntucloud:/home$ cd ..
jlopez@ubuntucloud:/$ sudo su
[sudo] password for jlopez:
root@ubuntucloud:/# export PATH=$PATH:/var/lib/gems/1.8/bin/
root@ubuntucloud:/# vmc target
vmc: Unknown command [target]
Usage: vmc [options] command [<args>] [command options]
Try 'vmc help [command]' or 'vmc help options' for more information.

root@ubuntucloud:/# vmc target

[http://api.vcap.me]
root@ubuntucloud:/#
```

Fuente, Elaboración propia.

- Error ejecutar una aplicación de prueba.

Al realizar la ejecución de la aplicación de prueba hello2, en la plataforma como servicio en la nube privada, se genero e siguiente error “**no such file or directory - /tmp/keyring-MTnqPL**”, se verifico permisos en el directorio y el archivo a ejecutar en este caso hello2 y se reinstalo Ruby completo nuevamente pero el problema persistía, según el soporte el problema ya era netamente de la librerías de Ruby mas no de la instancia de la plataforma como servicio, después leer varios foros de soporte se detecto que la aplicación requiera las librería de Sinatra como Framework el cual no se había instalado, se realizo la instalación de Sinatra con el comando **\$sudo gem install Sinatra**.

Figura 74, Error al desplegar una aplicación por falta de librerías

```
root@ubuntucloud: /
File Edit View Search Terminal Help
Error: Application 'hello' already exists, use update

root@ubuntucloud:/home/jlopez# vmc update hello
Uploading Application:
/var/lib/gems/1.8/gems/vmc-0.3.16.beta.1/lib/cli/commands/apps.rb:348:in `check_unreachable_links':
/usr/lib/ruby/1.8/pathname.rb:434:in `lstat'
/usr/lib/ruby/1.8/pathname.rb:434:in `realpath_rec'
/usr/lib/ruby/1.8/pathname.rb:441:in `realpath_rec'
/usr/lib/ruby/1.8/pathname.rb:467:in `realpath'
/var/lib/gems/1.8/gems/vmc-0.3.16.beta.1/lib/cli/commands/apps.rb:339:in `select'
/var/lib/gems/1.8/gems/vmc-0.3.16.beta.1/lib/cli/commands/apps.rb:339:in `check_unreachable_links'
/var/lib/gems/1.8/gems/vmc-0.3.16.beta.1/lib/cli/commands/apps.rb:366:in `upload_app_bits'
/var/lib/gems/1.8/gems/vmc-0.3.16.beta.1/lib/cli/commands/apps.rb:361:in `chdir'
/var/lib/gems/1.8/gems/vmc-0.3.16.beta.1/lib/cli/commands/apps.rb:361:in `upload_app_bits'
/var/lib/gems/1.8/gems/vmc-0.3.16.beta.1/lib/cli/commands/apps.rb:232:in `update'
/var/lib/gems/1.8/gems/vmc-0.3.16.beta.1/lib/cli/runner.rb:454:in `send'
/var/lib/gems/1.8/gems/vmc-0.3.16.beta.1/lib/cli/runner.rb:454:in `run'
/var/lib/gems/1.8/gems/vmc-0.3.16.beta.1/lib/cli/runner.rb:14:in `run'
/var/lib/gems/1.8/gems/vmc-0.3.16.beta.1/bin/vmc:5
```

Fuente, Elaboración propia.

6.1.1.10. Conclusiones en la implementación del Prototipo1

Micro Cloud Foundry permite desplegar una nube completa basada en el modelo de plataforma como servicio (PaaS), dentro de una infraestructura de red privada, para la prueba y ejecución de aplicaciones basadas en lenguajes de programación como Ruby, Spring, Java entre otros.

La puesta en marcha de Micro cloud Foundry es relativamente sencilla ya que al descargar se obtiene una maquina virtual basada en Ubuntu Linux 10.04 con todos paquetes y servicios que se requieren para la ejecución de aplicaciones en la nube.

La interacción con la plataforma como servicio es totalmente por línea de comandos a través de VMC (cliente de línea de comandos de Vmware), que se puede utilizar en modo consola de cualquier sistema operativo.

Una vez desplegada la instancia virtual la interacción con ella es sencilla a través de la línea de comandos siempre y cuando se tengan instaladas la librerías necesarias del lenguaje de programación en el equipo remoto, esto permite un despliegue fácil y rápido de aplicaciones Web multiplataforma.

6.2. PROTOTIPO 2 PAAS PRIVADA BASADO EN WAVEMAKER Y EUCALYPTUS

La implementación de este prototipo permitiera observar a interoperabilidad de las dos principales capas (IaaS y PaaS) de la computación en la nube dentro de una nube privada.

Primero se realizara la instalación y configuración de la capa de IaaS, y luego se desplegara Wavemaker como, Framework dentro de la nube privada en la capa de PaaS.

Tabla 9, Prerrequisitos Prototipo 2

PRERREQUISITO	DESCRIPCIÓN
S.O Anfitrión	Ubuntu Enterprise Cloud 11.04 (64bits)
Software (IaaS)	Eucalyptus 2.0.3 (64bits)
Software (PaaS)	Wavemaker 6.3.3 64bits
Hypervisor	KVM

Fuente, Elaboración propia.

6.2.1. La topología de red.

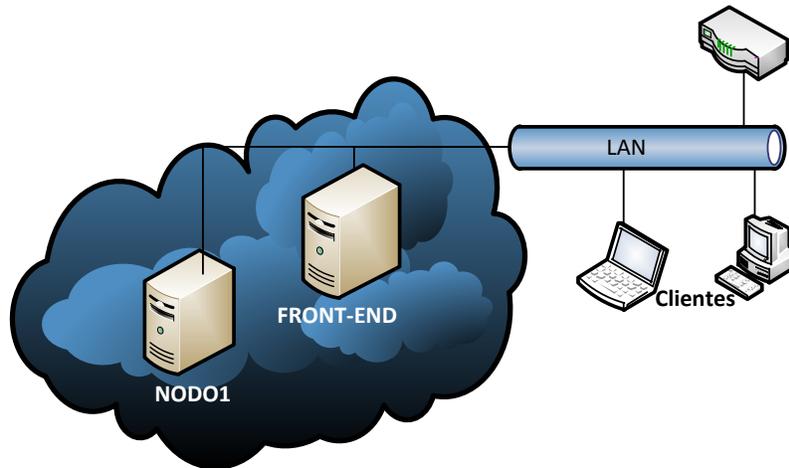
La topología de red escogida para la implementación del prototipo fue de dos maquinas para la capa de IaaS.

Maquina A: Front-end (CLC/Walrus/CC/SC)

Maquina B: Nodo1 (NC)

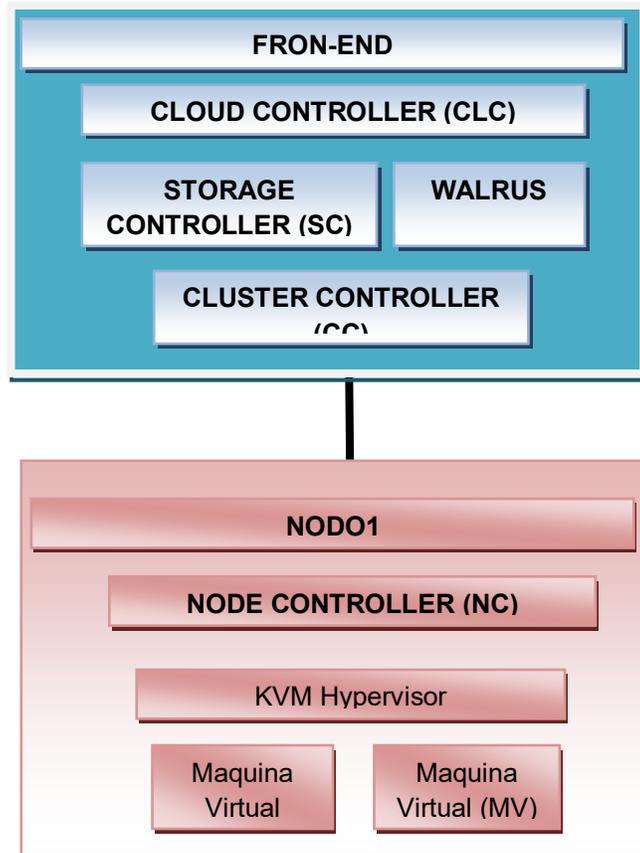
(more NCs...)

Figura 75, Topología de red de 2 maquinas (IaaS).



Fuente, Elaboración propia.

Figura 76, Componentes de Eucalyptus distribuidos en las maquinas del prototipo



6.2.2. Instalación de Ubuntu Enterprise Cloud (UEC)

Para realizar la instalación de Ubuntu Enterprise Cloud (UEC), los componentes de Eucalyptus como infraestructura como servicio (IaaS) y demás software del servidor, es necesario descarga de la página oficial de Ubuntu la versión deseada para este prototipo se utilizo la imagen de instalación de Ubuntu Server 11.04 de 64bits.

A continuación se presentan los pasos requeridos para la instalación de cada uno de los componentes en cada maquina

6.2.2.1. Instalación UEC maquina A (Front-end)

Al iniciar el asistente de instalación de UEC, primero se escoge el lenguaje de instalación.

Figura 77, Inicio Instalación UEC



Fuente, Elaboración propia.

Se selecciona la opción “Install Ubuntu Enterprise Cloud”, el cual permitirá instalar más adelante los componentes de Eucalyptus.

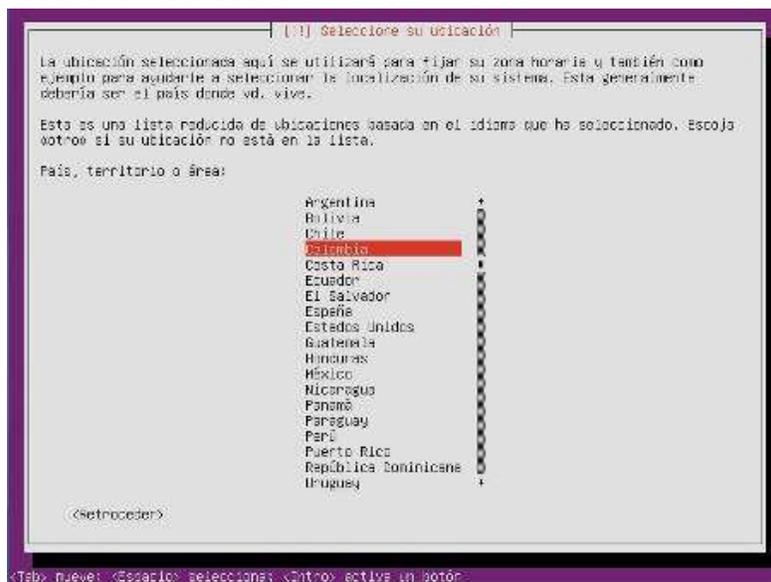
Figura 78, Instalación Fron-End



Fuente, Elaboración propia.

Se escoge la ubicación, para fijar la zona horaria.

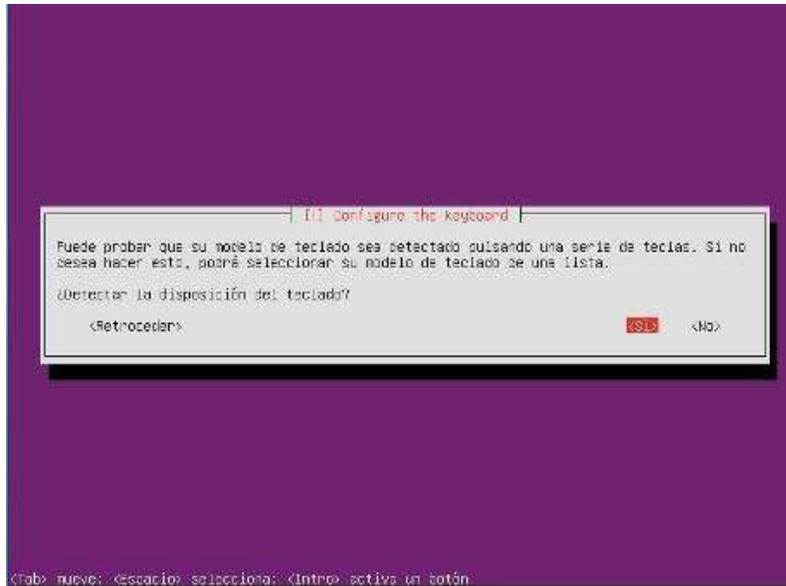
Figura 79, Seleccionar Ubicación



Fuente, Elaboración propia.

Se configura el teclado, para lo cual se escoge la detección de dispositivo del teclado.

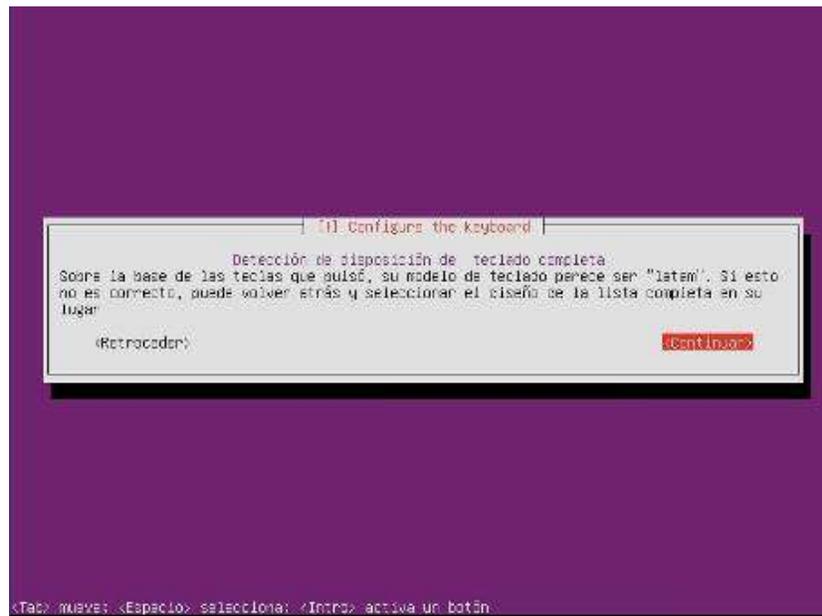
Figura 80, Conjunción de teclado en UEC



Fuente, Elaboración propia.

Se confirma la configuración del teclado y se continúa.

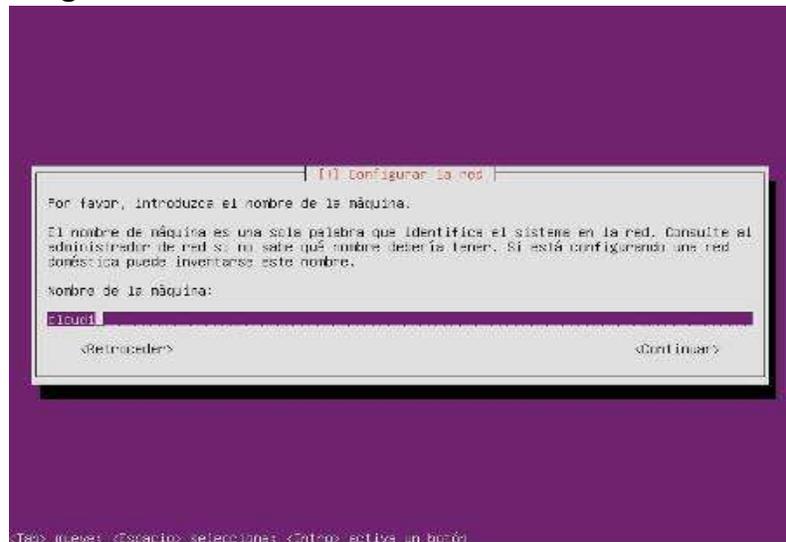
Figura 81, Confirmación del teclado



Fuente, Elaboración propia.

Se realiza la configuración de la red, se escribe el nombre de la maquina con la cual se va identificar en la red, debe ser un nombre único el cual va a identificar nuestro Cloud Controller en la nube privada para este caso es Cloud1.

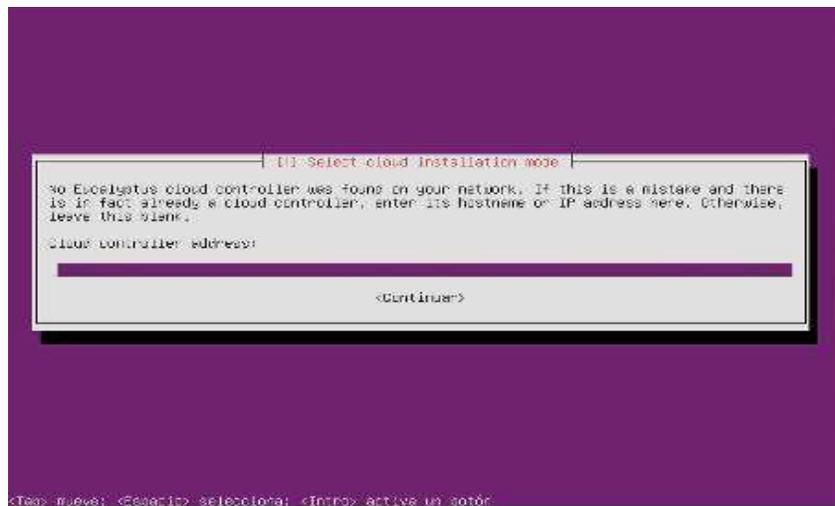
Figura 82, Configuración de Red UEC



Fuente, Elaboración propia.

El instalador pregunta por al modo de instalación de la nube, si no encuentra un cloud controller solicitará la dirección, en éste caso, la dejaremos en blanco (ésta máquina será nuestro Cloud Controller).

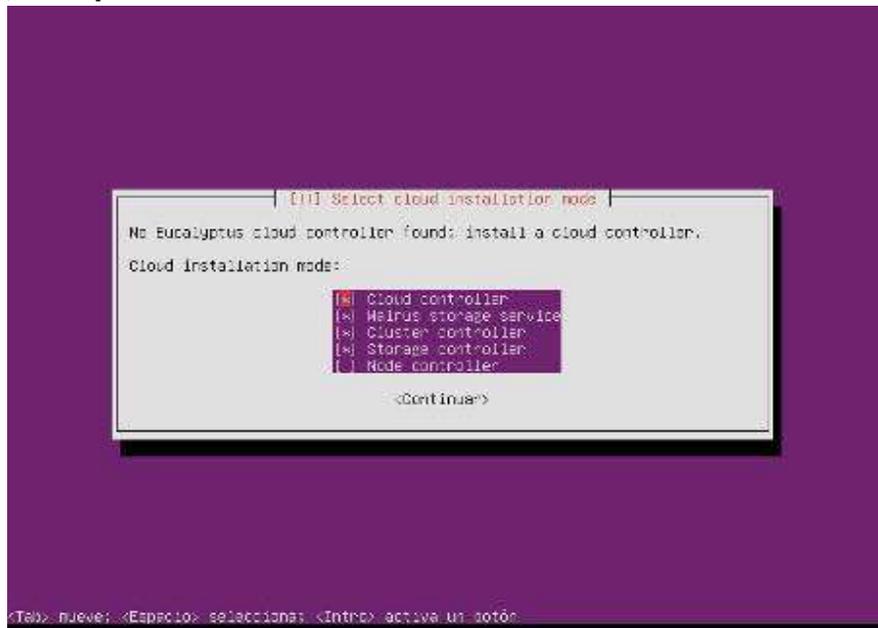
Figura 83, Modo de Instalación de la nube en UEC



Fuente, Elaboración propia.

Se escogen los componentes a instalar en la maquina, para este caso según la topología escogida de 2 maquinas, se selecciona (Cloud Controller, Walrus storage service, Cluster controller, storage controller).

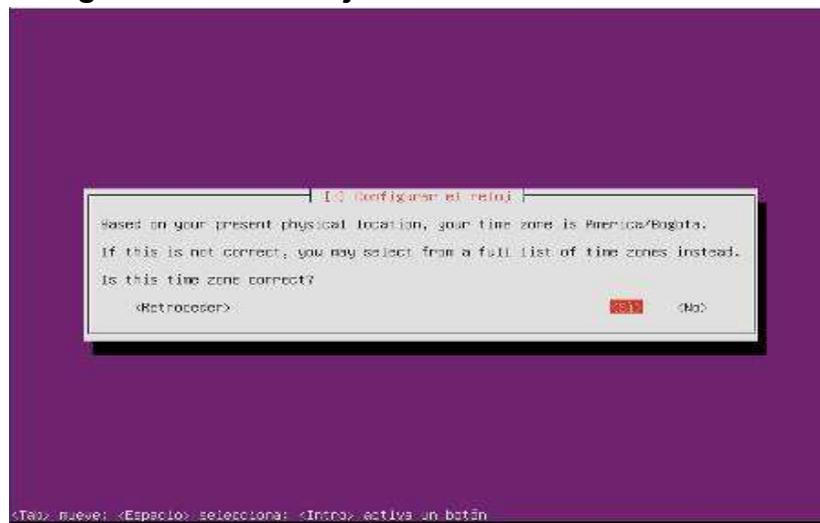
Figura 84, Componentes a instalar en UEC



Fuente, Elaboración propia.

Se confirma la configuración del reloj de acuerdo a la zona horaria.

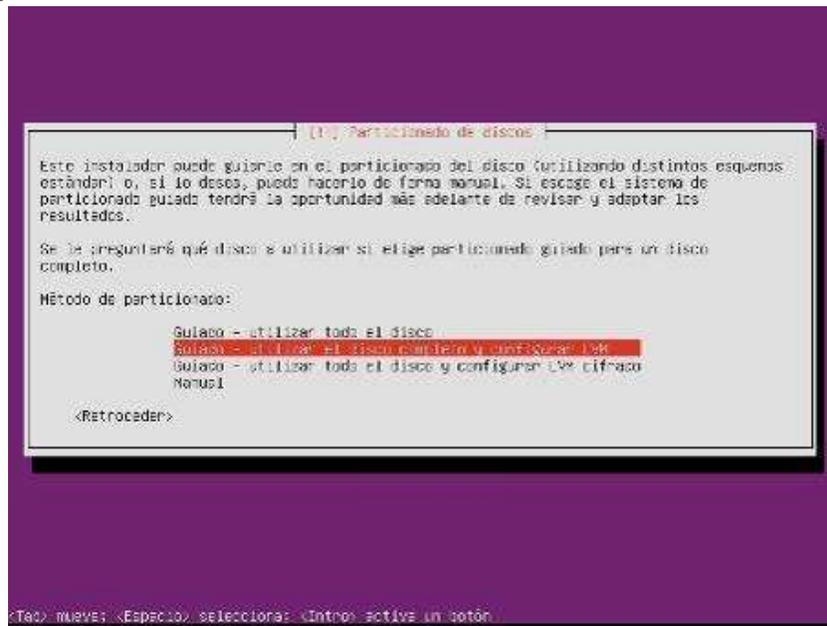
Figura 85, Configuración del Reloj



Fuente, Elaboración propia.

Se selecciona el método de partición del disco duro.

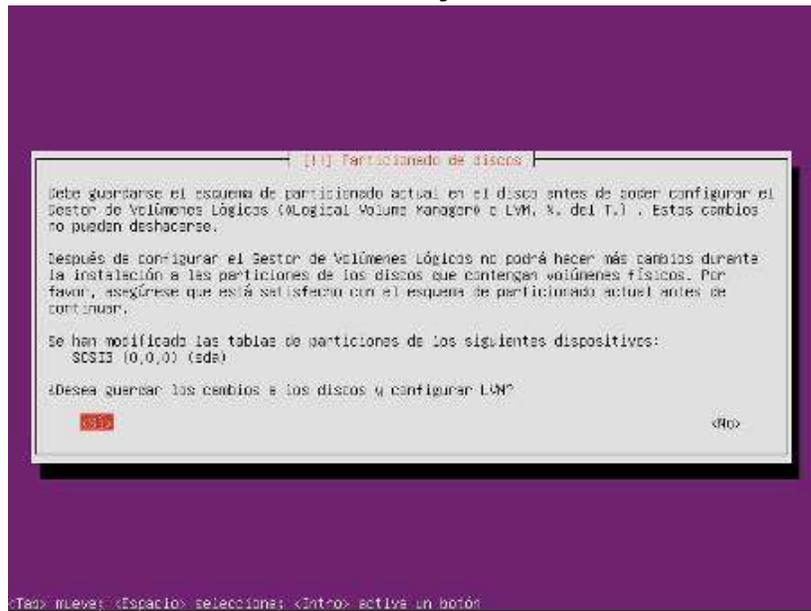
Figura 86, partición del Disco Duro en UEC



Fuente, Elaboración propia.

Se guardan los cambios de la nueva configuración del disco duro.

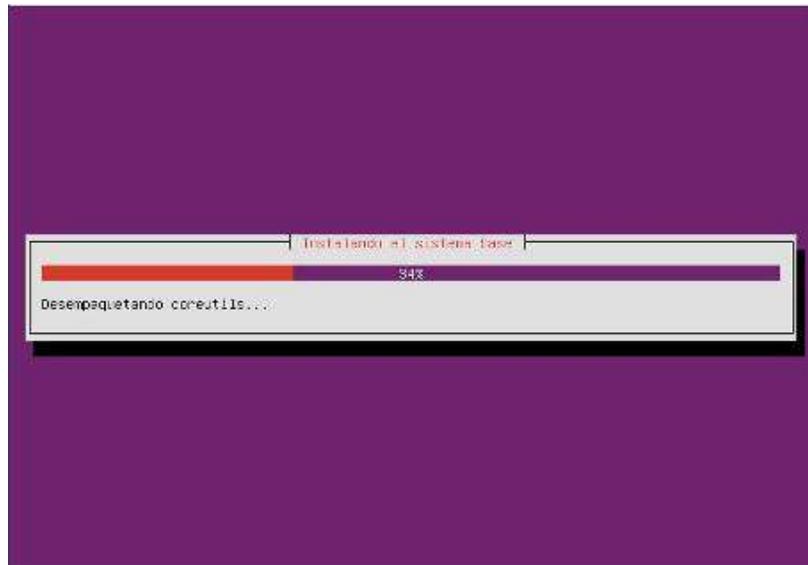
Figura 87, Salvar cambio del disco duro y confirmación



Fuente, Elaboración propia.

Se inicia la instalación del sistema base.

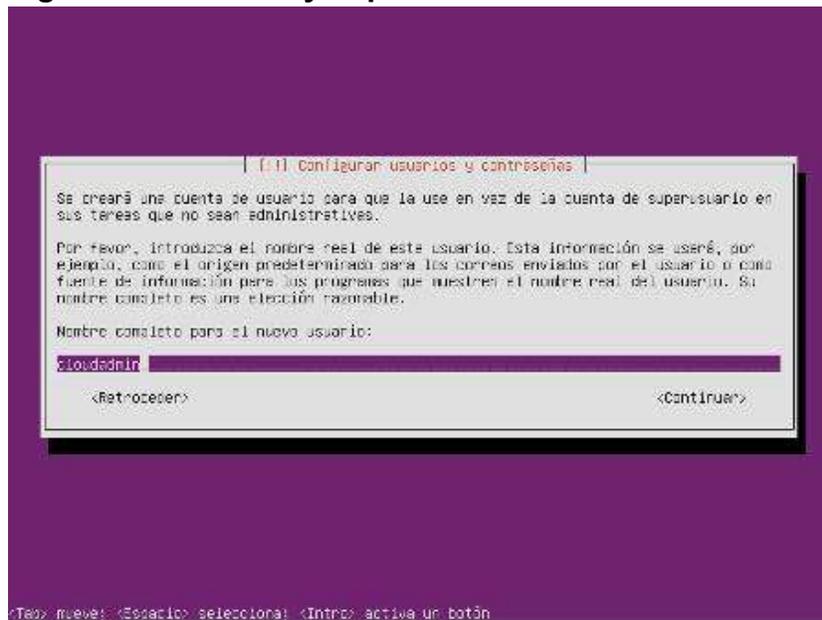
Figura 88, Inicio Instalación sistema base



Fuente, Elaboración propia.

Al finalizar la instalación sistema base, se realiza la configuración de la cuenta usuario para no tener que utilizar el supe usuario en tareas no administrativas.

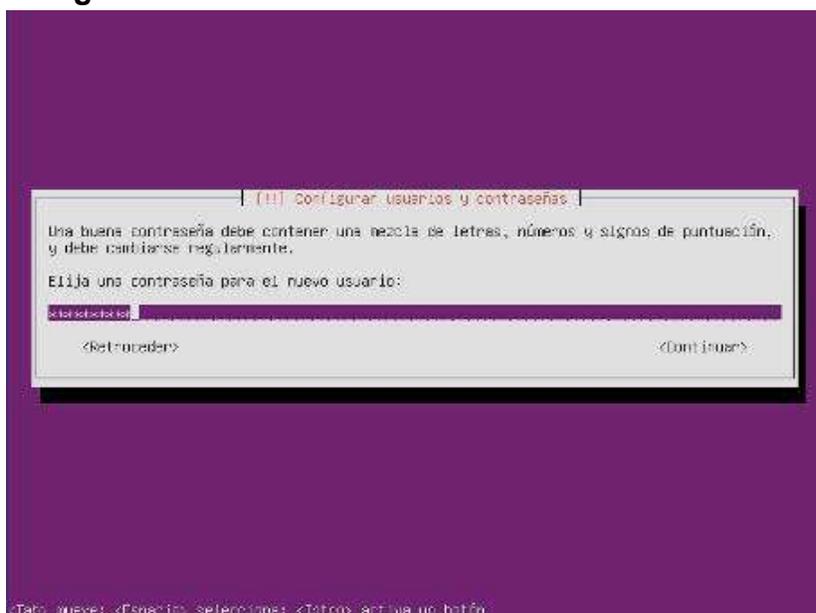
Figura 89, Registro de usuario y súper usuario.



Fuente, Elaboración propia.

Se configura la contraseña para el nuevo usuario.

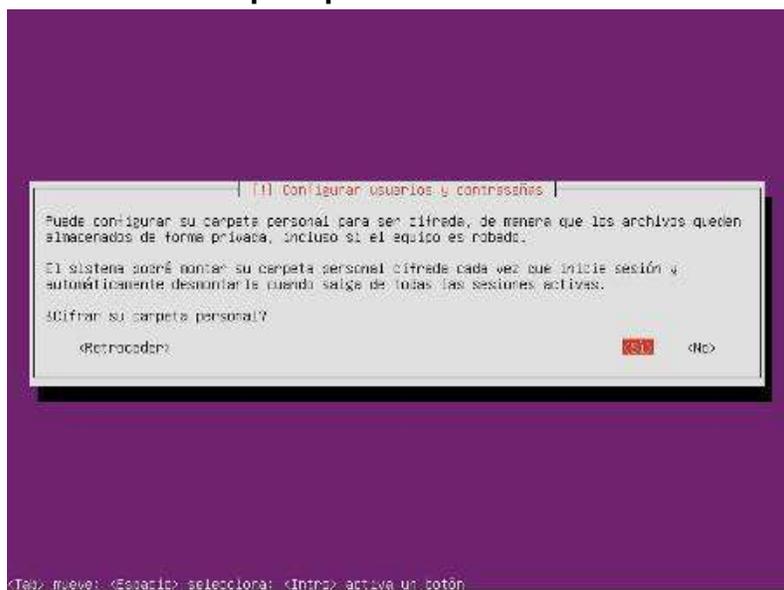
Figura 90, Configuración de contraseña usuario nuevo



Fuente, Elaboración propia.

Se confirma la creación de la carpeta personal y el cifrado de la misma.

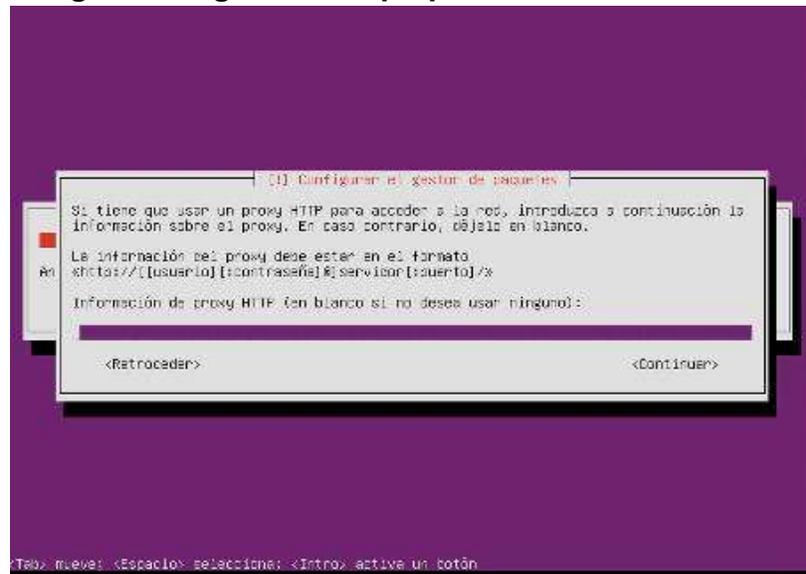
Figura 91, Creación de la carpeta personal.



Fuente, Elaboración propia.

Se configura la gestión de paquetes, en caso de tener un servidor proxy para conectar a internet, es necesario introducir la información de lo contrario se deja en blanco.

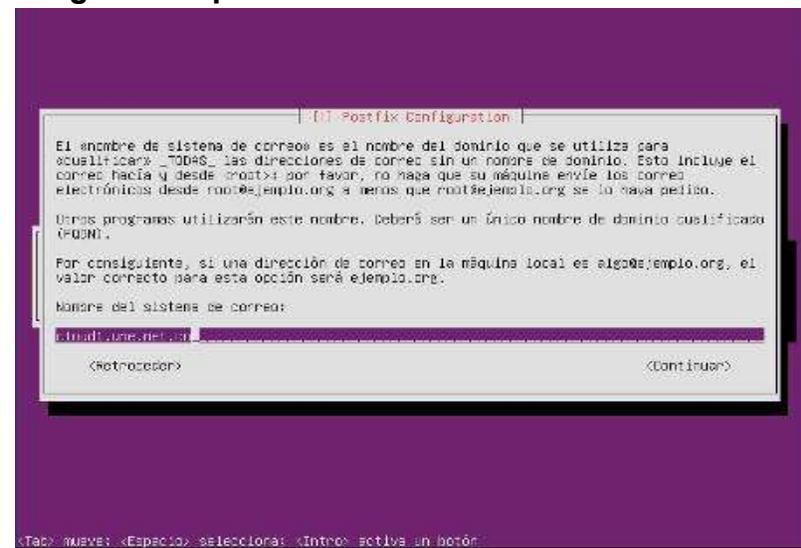
Figura 92, Configuración gestión de paquetes.



Fuente, Elaboración propia.

Se realiza la configuración de Postfix dando un nombre al sistema de correo en este caso es **cloud1.une.net.co**

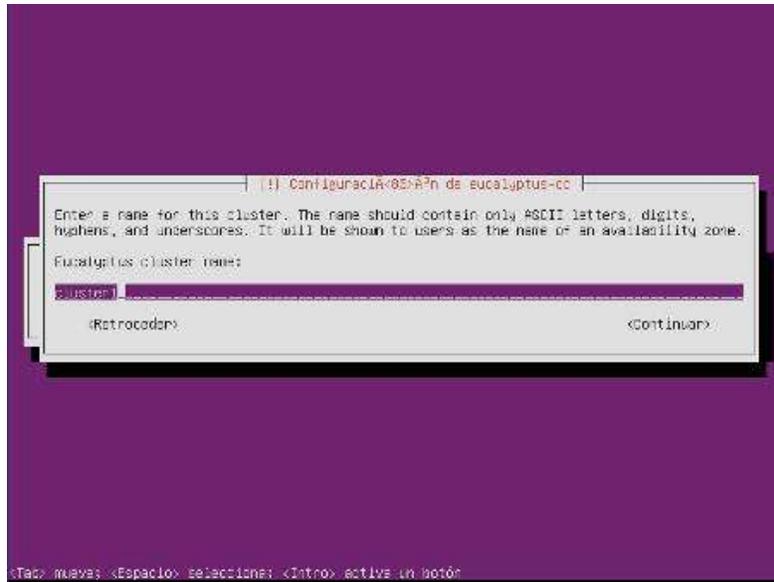
Figura 93, Configuración postfix.



Fuente, Elaboración propia.

Se realiza la configuración del Cluster Controller, agregando el nombre del Cluster, para nuestro caso **cluster1**

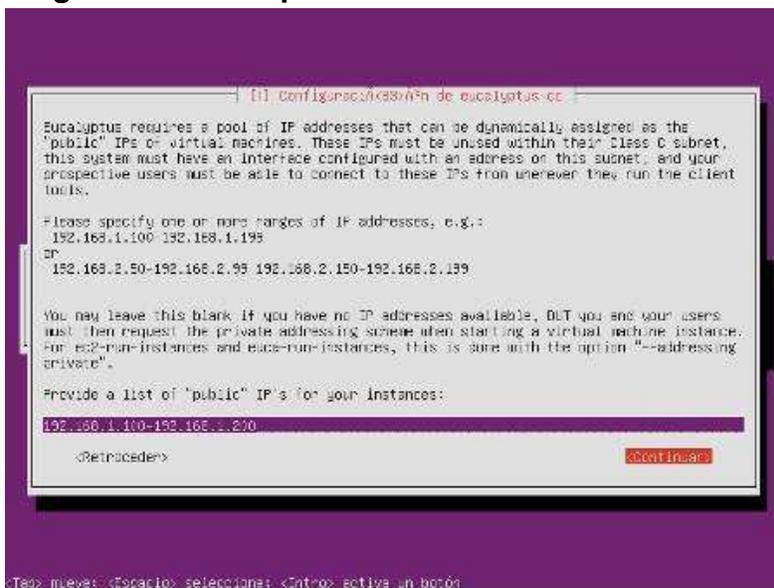
Figura 94, Configuración del Cluster controller.



Fuente, Elaboración propia.

Se especifica el rango de direcciones IP públicas en la red local que puedan ser asignadas a las instancias virtuales. 192.168.1.100-192.168.1.200.

Figura 95, Configuración de IP públicas de la nube.



Fuente, Elaboración propia.

Se configura el registro principal de arranque Grub.

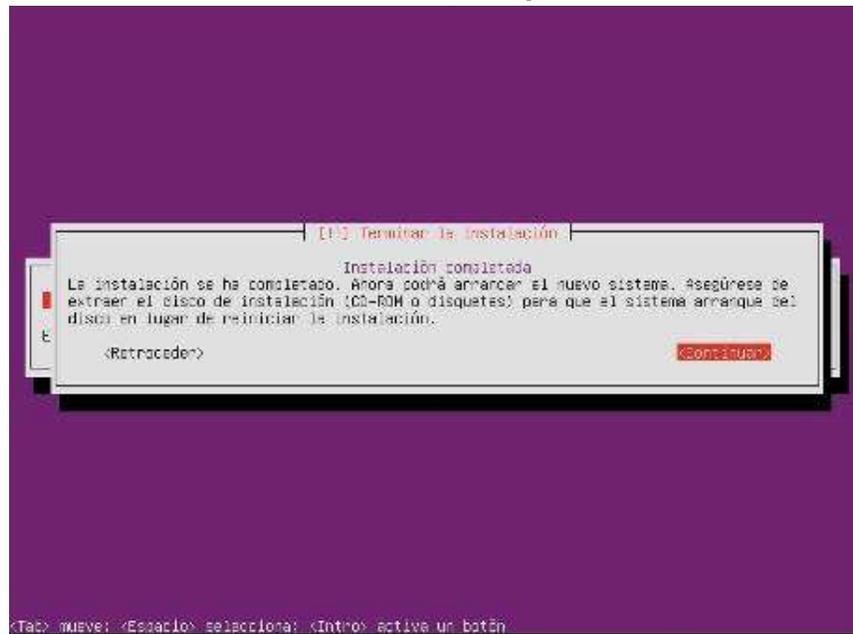
Figura 96, Configuración de inicio Grub



Fuente, Elaboración propia.

Se confirma la finalización de la instalación y se reinicia la maquina.

Figura 97, Finalización de la Instalación, maquia A



Fuente, Elaboración propia.

Se vez reiniciado el servidor se autentica en la máquina para continuar con el resto de la configuración.

Figura 98, Inicio de la Maquina A Fron-end

```
nde BIOS or use force_addr=0xaddr
Ubuntu 11.04 cloud1 tty1
cloud1 login: cloudadmin
Password:
Welcome to Ubuntu 11.04 (GNU/Linux 2.6.38-8-server x86_64)

* Documentation:  http://www.ubuntu.com/server/doc

System information as of Tue Aug 23 19:26:05 COT 2011

System load:  1.21          Processes:      121
Usage of /:   3.3% of 37.17GB Users logged in:  0
Memory usage: 15%         IP address for eth0: 192.168.1.109
Swap usage:   0%

Graph this data and manage this system at https://landscape.canonical.com/

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/*-copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

cloudadmin@cloud1:~$
```

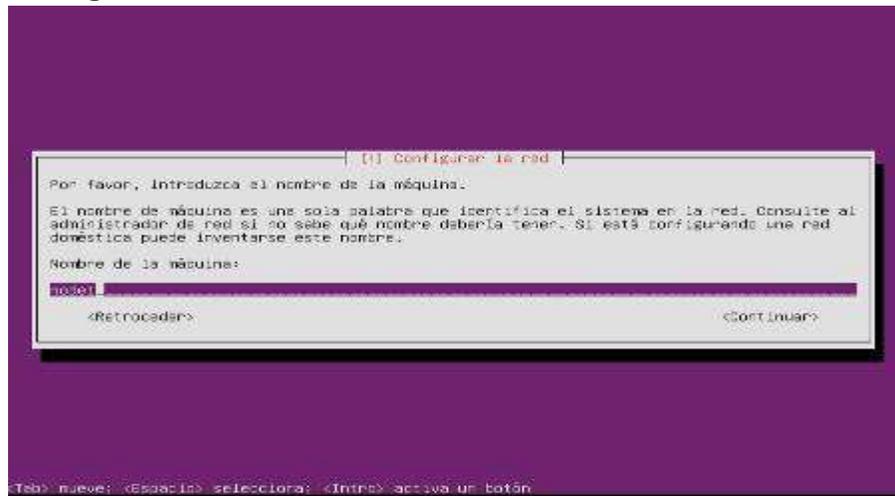
Fuente, Elaboración propia.

6.2.2.2. Instalación UEC maquina B (Nodo)

Para la instalación y configuración del nodo repetimos los pasos anteriores. Iniciamos la instalación de Ubuntu Enterprise Cloud (UEC) y en el paso donde se configura la red, se introduce el nombre de la máquina para esta nube la maquina nodo se llamara **nodo1**.

Configuración de la red, nombre de la maquina (nodo1)

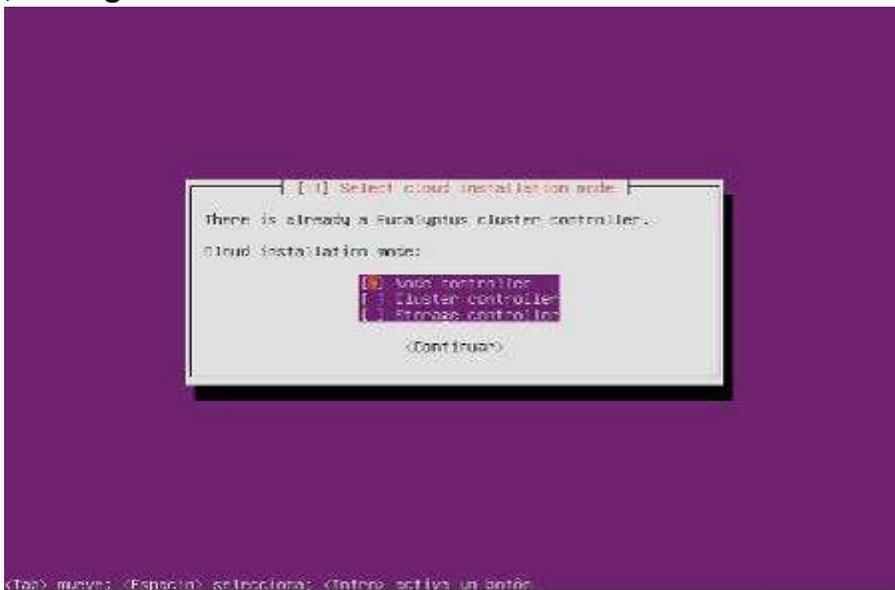
Figura 99, Configuración de red nodo 1



Fuente, Elaboración propia.

Se Continúa con la instalación normal como se describe anteriormente hasta el paso, donde se selecciona el modo de instalación de la nube, por defecto y automáticamente Ubuntu detecta el Cloud Controller (CLC) y se instala el node Controller. Se escoge el modo de instalación de la nube para la maquina B como Node Controller.

Figura 100, Configuración del modo de instalación del nodo



Fuente, Elaboración propia.

Se finalizada la instalación y configuración del Node Controller se reinicia la maquina y nos autenticamos con el mismo usuario y contraseña del Cloud Controller.

6.2.2.3. Configuración de la nube privada en Ubuntu Enterprise Cloud (UEC). Una vez finalizada la instalación y configuración de las maquinas se accede por medio del navegador Web de un cliente al servidor de Infraestructura como servicio (IaaS), dentro de nube privada.

Para ingresar a la interfaz web de UEC introducimos la siguiente URL:

http://ip_cloud_controller:8443

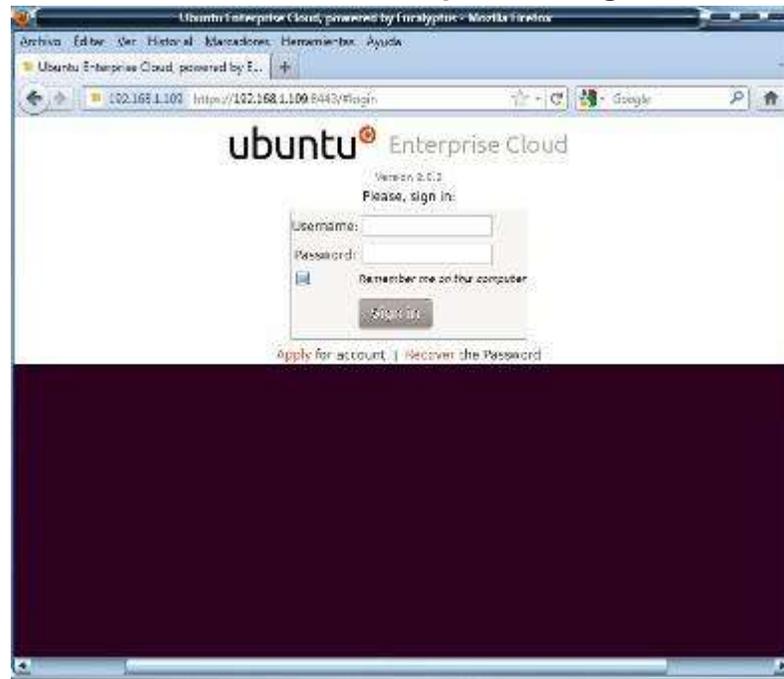
Figura 101, Interfaz Web de UEC, en la nube privada, riesgo de seguridad



Fuente, Elaboración propia.

Como es la primera vez que se ingresa el navegador pide la autenticación y seguridad del sitio, se acepta los riesgos y se añaden las firmas de seguridad.

Figura 102, Interfaz Web de UEC, en la nube privada, login



Fuente, Elaboración propia.

Al validar el sitio se accede con el usuario **admin** y password **admin**, Inmediatamente se realiza el cambio de password y el e-mail del administrador.

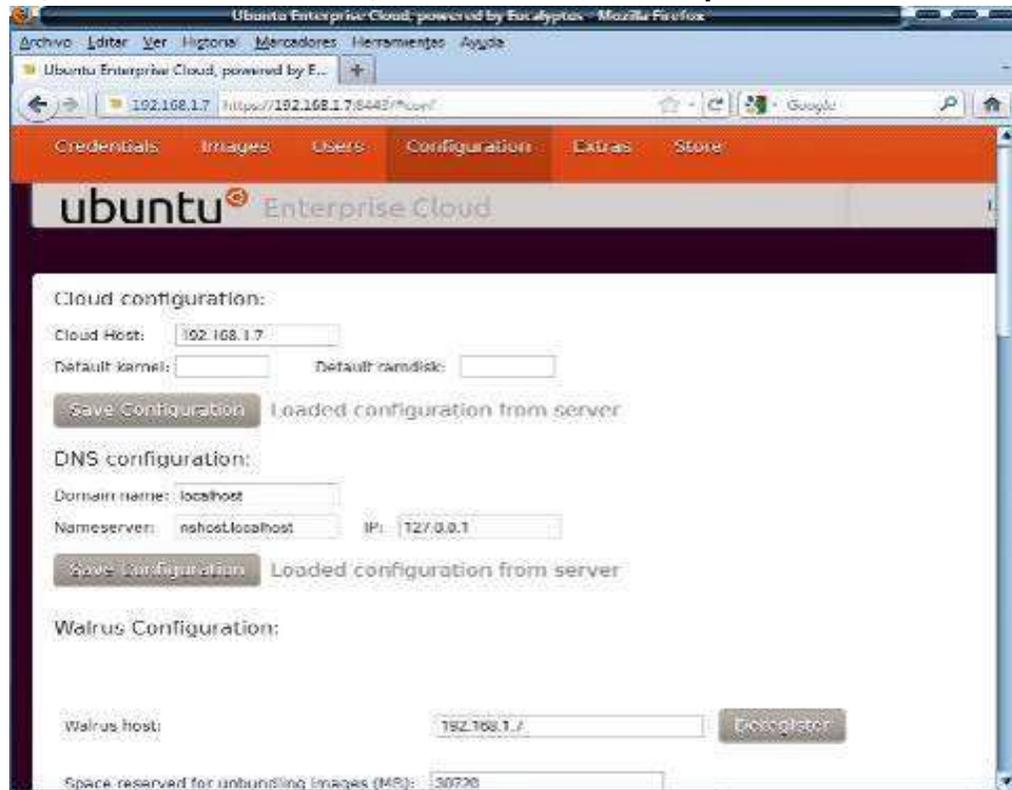
Figura 103, Interfaz Web de UEC, cambio de password.



Fuente, Elaboración propia.

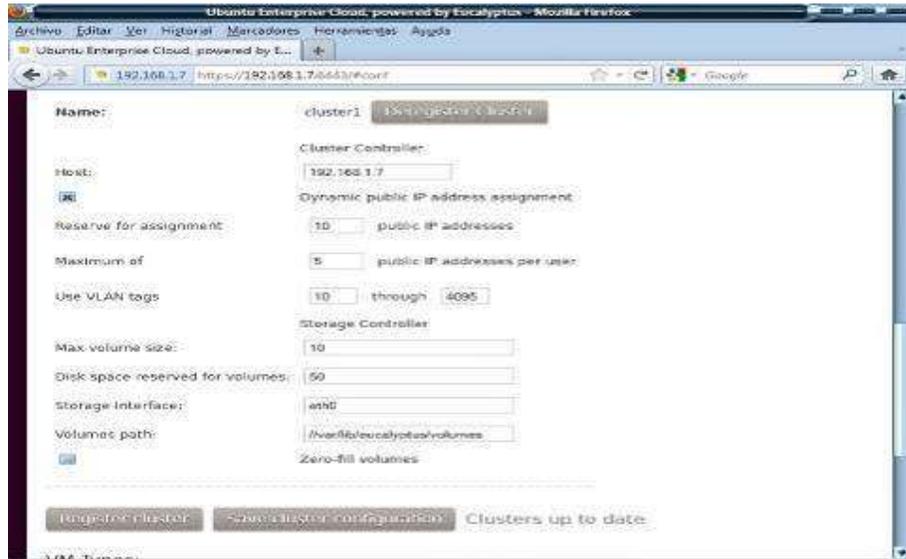
Se cambiada la contraseña se ingresa por completo a la interfaz web donde se encuentra el menú que permite acceder a las diferentes opciones de la nube, obtener la credenciales de autenticación de los usuarios, las imágenes instaladas, la administración de los usuarios, la configuración de los componentes de la nube, extras para la descarga de mas imágenes y software de gestión y el store donde se puede descargar mas imágenes de diferentes sistema operativos.

Figura 104, Interfaz Web de UEC, Acceso al menú de opciones.



Fuente, Elaboración propia.

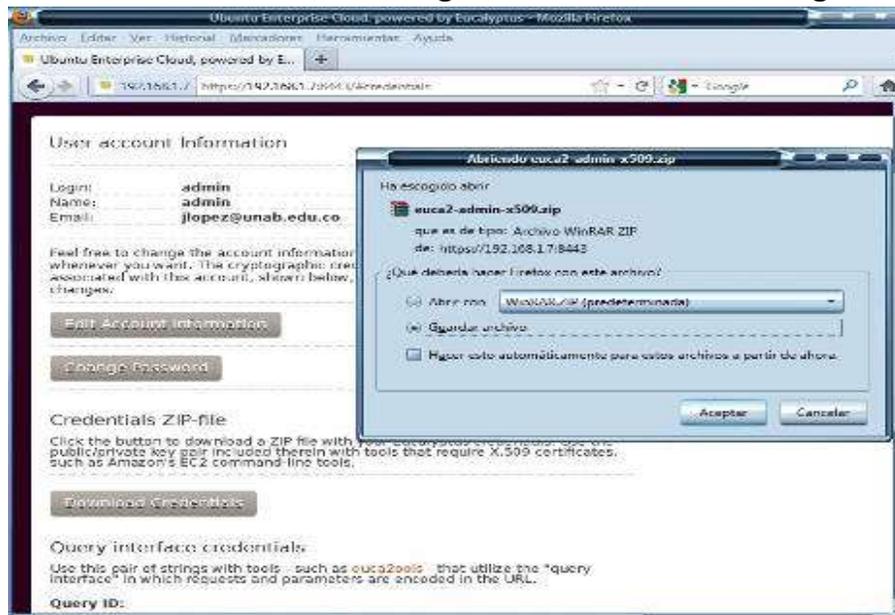
Figura 105, Interfaz Web de UEC, Parámetros de configuración.



Fuente, Elaboración propia.

Para poder continuar usar completamente la nube, se deben generar e instalar certificados de seguridad locales en un fichero script, esto se debe hacer por cada uno de los usuarios de la nube.

Figura 106, Interfaz Web de UEC, Descarga de credenciales de seguridad.



Fuente, Elaboración propia.

Una vez el usuario se autentica en la interfaz Web del Cloud Controller (CLC) en la opción Credentials, se debe pulsar el botón Download credentials, que están en un archivo .zip esto nos permite descargar el archivo euca2-admin-x509.zip el cual lo debemos descomprimir en una carpeta llamada .euca dentro del directorio del usuario actual en esta carpeta se deben encontrar los siguientes archivos, cloud-cert.pem, euca2-adminba25cbf5-pk.pem, euca2-adminba25cbf5-cert.pem, eucarc y jssecacerts, al fichero eucarc se le deben dar permisos de administrador ya que este contiene las variables de entorno con las rutas a diversos servicios como Walrus y Eucalyptus, así como el acceso a EC2 y las claves secretas necesarias para comunicarse con Eucalyptus.

Al lado de servidor el Cloud Controller (CLC), también se debe obtener las credenciales para poder comunicarse con el cliente, para esto desde la línea de comandos.

Figura 107, Comandos de obtención y registro de credenciales de autenticación en el servidor.

```

cloudadmin@cloud1:~/.euca$ sudo euca_conf --get-credentials mycreds.zip
[sudo] password for cloudadmin:
--2011-08-24 20:43:51-- https://localhost:8443/getX509?user=admin&code=1UREFDyF
kZKFx3iZoIRALA1e8P2ziCBNGhoUkUubFzod1fg3nXrkIUSwpBG1ourFBzyH2QMU5BqmcXr0Q
Resolviendo localhost... 127.0.0.1
Conectando a localhost[127.0.0.1]:8443... conectado.
AVISO: no se puede verificar el certificado de localhost, emitido por «/C=US/O=C
loud/OU=Eucalyptus/CN=db»:
Se encontró un certificado autofirmado.
AVISO: el nombre común certificado «db» no coincide con el nombre de anfitrión s
olicitado «localhost».
Petición HTTP enviada, esperando respuesta... 200 OK
Longitud: 4888 (4,8K) [application/zip]
Guardando en: «mycreds.zip»

100%[=====] 4.888      --.-K/s   en 0s

2011-08-24 20:43:52 (348 MB/s) - «mycreds.zip» guardado [4888/4888]

cloudadmin@cloud1:~/.euca$ ls
mycreds.zip
cloudadmin@cloud1:~/.euca$ unzip mycreds.zip
Archive:  mycreds.zip
To setup the environment run: source /path/to/eucarc
  inflating: eucarc
  inflating: cloud-cert.pem
  inflating: jssecacerts
  inflating: euca2-admin-ce8d8bc6-pk.pem
  inflating: euca2-admin-ce8d8bc6-cert.pem
cloudadmin@cloud1:~/.euca$ ln -s ~/.euca/eucarc ~/.eucarc

```

Fuente, Elaboración propia.

Si toda sale bien podemos ver la disponibilidad del cluster local con el siguiente comando **euca-describe-availability-zones verbose**

Figura 108, Comprobación disponibilidad del Cluster en la nube privada.

```
Archive: mycreds.zip
To setup the environment run: source /path/to/eucarc
  inflating: eucarc
  inflating: cloud-cert.pem
  inflating: jssecacerts
  inflating: euca2-admin-ce8d8bc6-pk.pem
  inflating: euca2-admin-ce8d8bc6-cert.pem
cloudadmin@cloud1:~/euca$ ln -s ~/euca/eucarc ~/euca/eucarc
cloudadmin@cloud1:~/euca$ cd -
No se ha encontrado la orden «cd-», quizás quiso decir:
La orden «cda» del paquete «xncd» (universe)
La orden «cdb» del paquete «tingcdb» (main)
La orden «cdu» del paquete «codeville» (universe)
La orden «cdp» del paquete «lrpas» (multiverse)
La orden «cdo» del paquete «cdo» (universe)
La orden «cd5» del paquete «cd5» (universe)
La orden «cdw» del paquete «cdw» (universe)
cd -: orden no encontrada
cloudadmin@cloud1:~/euca$ cd -
/
cloudadmin@cloud1:/$ . ~/euca/eucarc
cloudadmin@cloud1:/$ euca-describe-availability-zones verbose
AVAILABILITYZONE      cluster1      192.168.1.109
AVAILABILITYZONE      l- um types   free / max    cpu   ram   disk
AVAILABILITYZONE      l- n1.small   0000 / 0000   1     192   2
AVAILABILITYZONE      l- c1.medium  0000 / 0000   1     256   5
AVAILABILITYZONE      l- n1.large   0000 / 0000   2     512  10
AVAILABILITYZONE      l- n1.xlarge  0000 / 0000   2    1024  20
AVAILABILITYZONE      l- c1.xlarge  0000 / 0000   4    2048  20
cloudadmin@cloud1:/$
```

Fuente, Elaboración propia.

Desde el servidor de la nube privada con el comando **euca_conf**, también se puede usar para activar o desactivar los nodos o el controlador del cluster: Para obtener más detalles podemos ejecutar **euca_conf -help**.

Figura 109, Comando euca_conf - -help en el Cloud Controller

```
--register-cluster <clustername> <host> add new cluster to EUCALYPTUS
--deregister-cluster <clustername> remove cluster from EUCALYPTUS
--register-walrus <host> add walrus to EUCALYPTUS
--deregister-walrus <host> remove walrus from EUCALYPTUS
--register-sc <clustername> <host> add storage controller
--deregister-sc <clustername> remove storage controller from EUCALYPTUS
PTUS
--list-walruses list registered walrus(es)
--list-clusters list registered CCs
--list-nodes list registered NCs
--list-scns list registered SCs
--no-sync used only with --register-* to skip syncing
keys
--cc-port <port> set CC port
--nc-port <port> set NC port
--instances <path> set the INSTANCE path
--hypervisor <kvm|xen> set hypervisor to use
--user <euca_user> set the user to use
--dhcpd <dhcpd> set the dhcpd binary to <name>
--dhcp_user <user> set the username to run dhcpd as
--name <var> returns the value or <name>
--import-conf <file> import variables from <file> into //etc/euca
lyptus/eucalyptus, local.conf
--setup perform initial setup
--enable {cloud|walrus|sc} enable service at next start
--disable {cloud|walrus|sc} disable service at next start
--check {nc|cc|cloud|sc|walrus} pre-flight checks
--version eucalyptus version
cloudadmin@cloud1:~$
```

Fuente, Elaboración propia.

6.2.3. Instalación y configuración de Plataforma como servicio (PaaS) en la nube privada.

Para el despliegue de la plataforma como servicio se realizara el instalacion y configuración de Wavemaker, un Framework que permite e desarrollo de aplicaciones Web completas en AJAX, desarrollado por Wavemaker y comprado en marzo del 2011 por Vmware, se encuentra liberado bajo licencia Apache 2.0

6.2.3.1. Instalación y configuración de Wavemaker

Para poder instalar wavemaker se debe descargar la sitio web oficial en la zona de descargas (<http://www.wavemaker.com/downloads/>) el paquete deseado, para este prototipo se descargo el paquete Linux DEB Package, 64-bit (Debian/Ubuntu).

Figura 110, Descarga Instalador Wavemaker 6.3 64bits



Fuente, Elaboración propia.

Se descargado el paquete de instalación desde la consola se accede al directorio de la descarga y se ejecuta el comando
Sudo dpkg -i wavemaker_6.3.3GA_amd64.deb
El desempaqueta el archive y se realiza la instalación.

Figura 111, Instalación de Wavemaker 6.3



Fuente, Elaboración propia.

6.2.3.2. Inicio de Wavemaker studio PaaS

Una vez instalado y configurado Wavemaker 6.3, para iniciar el entorno de trabajo, se debe desde la consola ingresar al directorio donde se encuentra el script de ejecución.

Se accede al directorio `/opt/wavemaker-6.3.3GA/bin` y se inicia el script `wavemaker.sh` con el comando `./wavemaker.sh start`

Figura 112, ejecutar el script de arranque de wavemaker

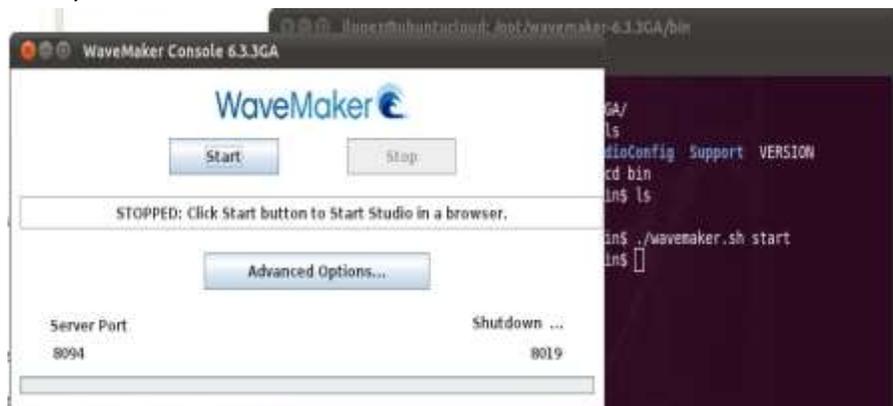


```
jlopez@ubuntucloud: /opt/wavemaker-6.3.3GA/bin
File Edit View Search Terminal Help
jlopez@ubuntucloud:/$ cd opt
jlopez@ubuntucloud:/opt$ cd wavemaker-6.3.3GA/
jlopez@ubuntucloud:/opt/wavemaker-6.3.3GA$ ls
bin launcher README Samples studio studioConfig Support VERSION
jlopez@ubuntucloud:/opt/wavemaker-6.3.3GA$ cd bin
jlopez@ubuntucloud:/opt/wavemaker-6.3.3GA/bin$ ls
wavemaker.sh
jlopez@ubuntucloud:/opt/wavemaker-6.3.3GA/bin$ ./wavemaker.sh start
jlopez@ubuntucloud:/opt/wavemaker-6.3.3GA/bin$
```

Fuente, Elaboración propia.

Se ejecuta el script carga una ventana donde tenemos la consola de inicio

Figura 113, Consola de inicio de wavemaker



Fuente, Elaboración propia.

Se hace clic en el botón start para cargar la interface grafica de wavemaker studio en el navegador web.

Figura 114, iniciar wavemaker



Fuente, Elaboración propia.

Figura 115, Interface grafica de wavemaker studio



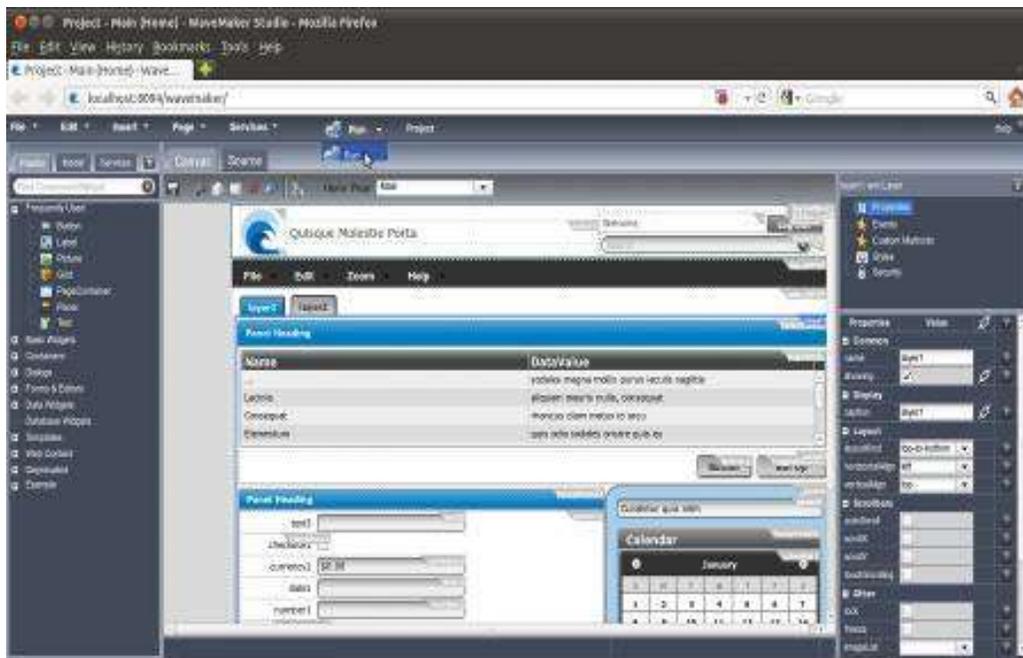
Fuente, Elaboración propia.

6.2.3.3. Pruebas de Wavemaker studio PaaS.

Wavemaker permite desarrollar aplicaciones web en Ajax y desplegarlas de varias formas, utilizando la infraestructura local o Infraestructura como servicio (IaaS), privada por medio de Eucalyptus o publica por medio de Amazon EC2.

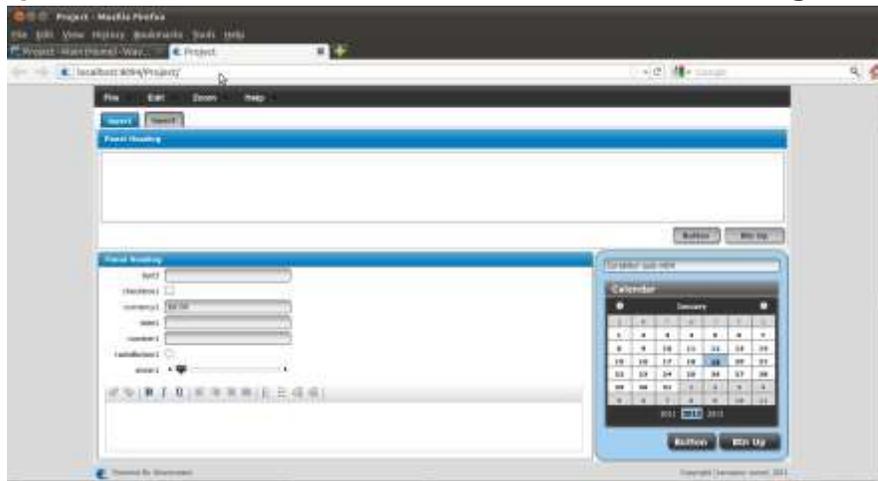
- Despliegue de wavemaker infraestructura local. Al acceder desde el navegador web a al interface de wavemaker, se abre el proyecto y en el menu Run, luego Run y se ejecuta la aplicación la cual carga en una nueva ventana del navegador web.

Figura 116, Ejecutar una aplicación en wavemaker



Fuente, Elaboración propia.

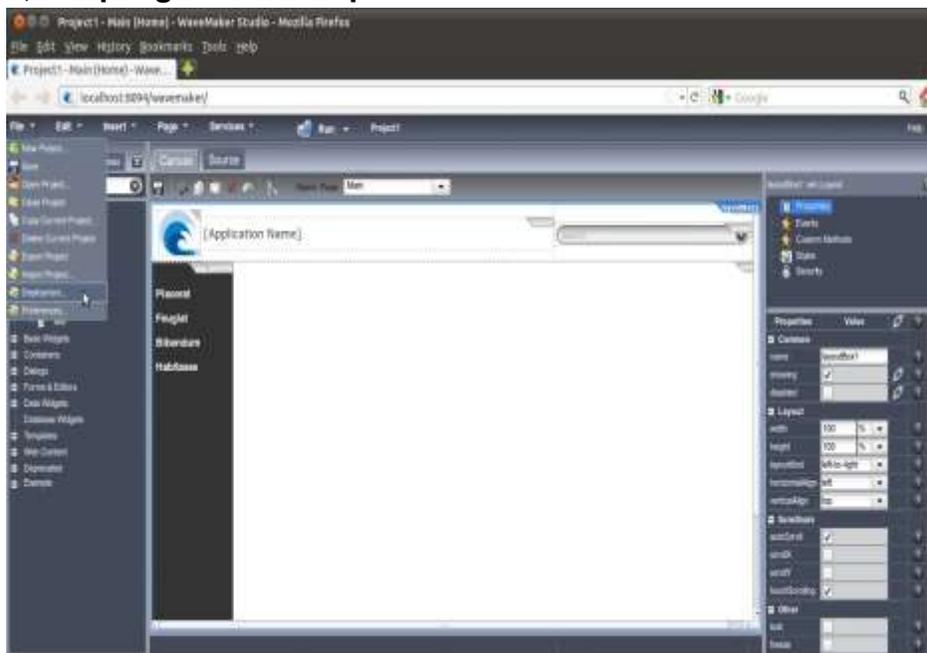
Figura 117, Aplicación corriendo en una nueva ventana del navegador.



Fuente, Elaboración propia.

- Despliegue de wavemaker con infraestructura como servicio privada. Para desplegar la aplicación utilizando una infraestructura como servicio privada, se abre el proyecto a ejecutar, luego se va al menú File se escoge al opción Deployment.

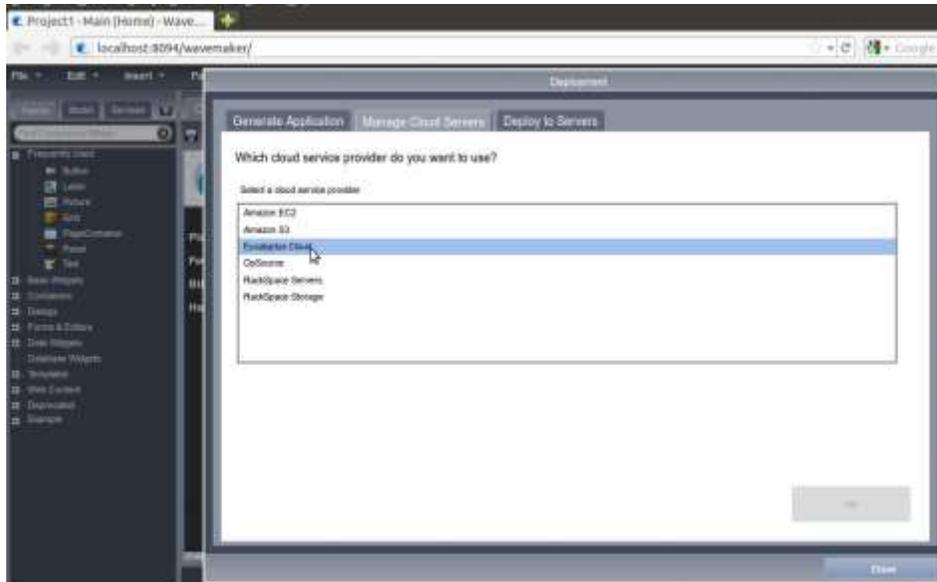
Figura 118, Despliegue de una aplicación Wavemaker con IaaS



Fuente, Elaboración propia.

Se abre el cuadro de opciones Deployment, se escoge la pestaña Manage Cloud Servers.

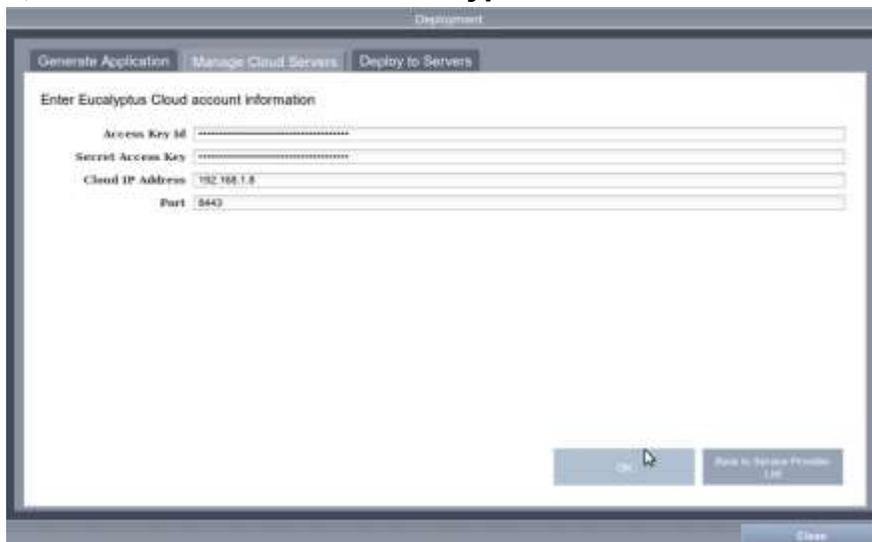
Figura 119, Administración de los servidores en al nube con wavemaker



Fuente, Elaboración propia.

Se selecciona Eucalyptus Cloud y se hace clic en el botón OK, Se introducen los parámetros de información de la cuenta de Eucalyptus Access Key Id, Secret Access Key, Cloud IP Address y Port

Figura 120, Acceso a la cuenta de Eucalyptus con wavemaker



Fuente, Elaboración propia.

Si no se conocen las llaves de autenticación de la cuenta de Eucalyptus, es necesario ingresar al administrador de la nube de Eucalyptus para obtener las credenciales de autenticación.

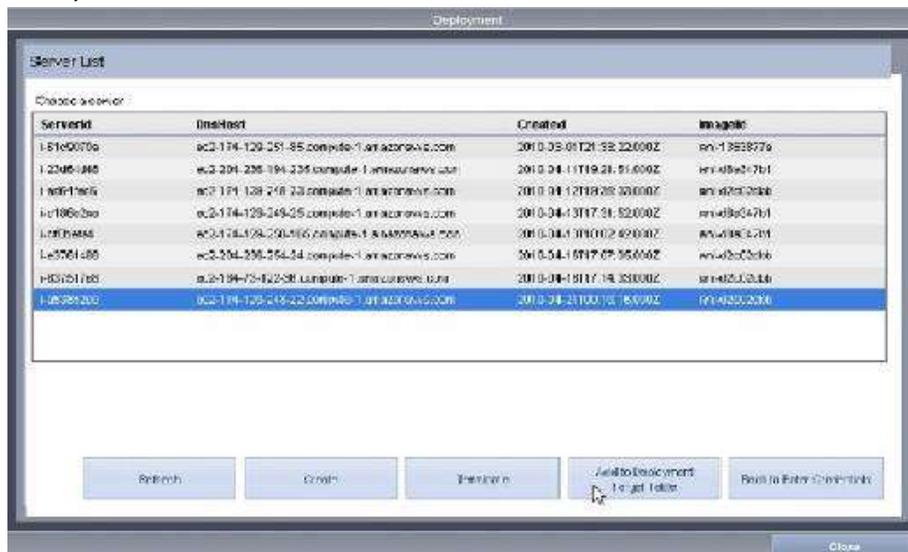
Figura 121, Credenciales de la cuenta de Eucalyptus



Fuente, Elaboración propia.

Luego se selecciona la instancia en la infraestructura como servicio se desea correr la aplicación y se hace clic en el botón Add deployment Teget table. Luego se muestra una lista de todas las aplicaciones que se ejecutan en la instancia, se escoge la desea de y pulsa el botón deploy

Figura 122, Lista de instancias en IaaS desde Wavemaker



Fuente, Elaboración propia.

ANÁLISIS DE LAS PRUEBAS PROTOTIPOS PAAS

Tabla 10, Análisis de pruebas Plataforma como servicio en una nube privada

Análisis PaaS	Cloud Foundry	WaveMaker	AppScale	SO2 Estratos
% de Pruebas.	100% nube privada.	6.3. 100% en nube privada.	50% error al conectar con la Imagen de Eucalyptus	40% error al cargar servicios en la MV VMware
Nombre de la Herramienta	Micro Cloud Foundry	Wavemaker Studio 6.3.3	AppScale 1.5 Eucalyptus Image	Wso2 Stratos 1.5.2.
Despliegue utilizado.	Infraestructura local, por medio de la Máquina virtual VMware	Instalación en servidor local y despliegue en IaaS privada con Eucalyptus	IaaS privada con Eucalyptus.	Infraestructura local, por medio de la Máquina virtual VMware
Otros métodos de Despliegue.	Implementación de la máquina virtual, con herramientas de virtualización de VMware para grandes centros de datos.	En servidor local, interacción con IaaS privada o pública con Amazon EC2	Por medio de Imagen virtual para KVM y XEN o integrada a la IaaS de Eucalyptus.	Instalación en servidor local, basados en Ubuntu Enterprise Cloud.
Instalación y configuración	6.4. Cargar la máquina virtual. 6.5. Se presentaron errores al realizar la configuración inicial de Micro Cloud Foundry, debido a que la MV no cargó la interfaz de red, se configuró manualmente.	6.6. Fácil instalación y configuración una herramienta muy amigable.	Hay que registrar la imagen en Eucalyptus manualmente y luego revisar la configuración. Dificultades para conectar con la instancia en la nube.	Solo es correr la máquina virtual sobre VMware, Dificultad al correr los servicios
Interface de usuario	Se instaló el cliente VMC de VMware, por línea de	Interface Web gráfica muy fácil de usar.	Por línea de comandos con AppScale Tools	Interface Web gráfica.

	comandos, se integrar al prompt del sistema operativo.			
Rendimiento	Al configurar la maquina virtual esta toma sus propios recursos, la respuesta es rápida por consola.	Lenta, debido a su interface grafica se torna pesada al cargar por red en otros equipos.	N/A	N/A
Plataformas de Uso	Multiplataforma por medio de la Maquina virtual, Herramienta desplegada en Ubuntu	Multiplataforma.	Solo Linux	Multiplataforma por medio de la Maquina virtual, instalación independiente Solo Ubuntu Enterprise Cloud
Documentación y soporte	Muy completo.	Faltan más detalles e la documentación, algo enredada.	Documentación simple falta especificar más.	Muy completa pero compleja.
Valoración De 1 a 5	4,5 Herramienta muy completa y fácil de implementar dentro de una Nube privada. Soporta, varios servicios y bases de datos la maquina virtual ya tiene todo integrado, solo es configurar las herramientas en el cliente.	3,8 Muy amigable pero limitada, solo permite desarrollo en AJAX, se necesita buenos recursos de Hardware si se desea correr localmente.	3,5 Permite el despliegue de Google AppEngine en una nube Híbrida, compleja la hora de desplegar e interactuar.	3,8 herramienta muy completa pero compleja se requiere tiempo para su estudio e implementación.

Fuente, Elaboración propia.

7. Referencias Bibliográficas.

AGUILERA, Sergio Omar. Desarrollo de Software Libre aplicado a la enseñanza de Sistemas Operativos sobre plataforma Cloud Computing. Universidad de Belgrano, Argentina 2011.

AJITH, Ranabahu, The Cloud Agnostic e-Science Analysis Platform. *Wright State University*.

AVETISYAN, Arutyun. Open Cirrus: A global cloud computing testbed. Institute for System Programming of the Russian Academy of Sciences.2010.

BACON, Jean. Enforcing End-to-End Application Security in the cloud. University of Cambridge. 2010.

BARRIE SOSINSKY, Cloud Computing Bible, Wiley Publishing, Indiana 2011

BUYYA, Rajkumar. InterCloud: Utility-Oriented Federation of Cloud Computing Environments for Scaling of Application Services. University of Melbourne, Australia. 2010.

CHAITAN Baru, Performance Evaluation of On-Demand Provisioning Strategies for Data Intensive Applications, Universidad de California San Diego.

CHENG. Ge. Building dynamic and transparent integrity measurement and protection for virtualized platform in cloud computing, Huazhong University of Science and Technology, People's Republic of China. 2010.

DIAZ, Marcos. A cost-benefit analysis of using cloud computing to extend the capacity of clusters. Springer Science+Business Media, 2010

FANG, Haifeng. TRIOB: A trusted virtual computing Environment based on Remote I/O binding mechanism. Institute of Computing Technology, Chinese Academy of Sciences. 2010

FREE SOFTWARE FOUNDATION en español, sitio web [En línea], [Diciembre 2011]. Disponible en: <http://www.gnu.org/home.es.html>

FRICKE Florian, CLOVR, A Genomics tool for automated and Portable Sequence Analysis using virtual machines and cloud computing. University Maryland and Baltimore.

Globus Toolking, Sitio Web [En línea], [Julio 2011]. Disponible en: <http://www.globus.org/toolkit/>

Grid Café, Sitio Web [En línea], [Julio 2011]. Disponible en http://www.gridcafe.org/que-es-la-computacion-grid_ES.html

Grid Way, Sitio Web [En línea], [Julio 2011]. Disponible en: <http://www.gridway.org/doku.php>

Gridtalk, pdf [En línea], [Julio 2011]. Disponible en: <http://www.gridtalk.org/briefings/gridsandclouds.pdf>

Harold Castro, Ph. D. Grid computing: promesa de los sistemas distribuidos, pdf [En línea], [Julio 2011]. Disponible en: http://www.acis.org.co/fileadmin/Revista_98/4_Uno.pdf

HOFFERT, Joe. Adapting Distributed Real-Time and Embedded Pub/Sub Middelware for cloud computing environments. Vanderbilt University. 2010.

Ian Foster, Yong Zhao, Ioan Raicu Cloud Computing and Grid Computing 360-Degree Compared, IEEE 06 enero 2009 pdf. http://ieeexplore.ieee.org/xpl/freeabs_all.jsp?tp=&isnumber=4738437&arnumber=4738445

JUNWEI, Ge. Research on the Resource Monitoring Model Under Cloud Computing Environment. Springer-Verlag Berlin Heidelberg 2010

KARIMI, Hassan Ali, Exploring Real-Time Geoprocessing in Cloud Computing: Navigation Services Case Study. TRANSACTIONS IN GIS . 2011

KARITA LUOKKANEN, Thomas Wozniak, Santi Ristol "Grid and Cloud Computing A Business Perspective on Technology and Applications" Springer Dec 2010.

KIM, Kyong Hoon. Power-aware provisioning of virtual machines for real-time Cloud services. CONCURRENCY AND COMPUTATION-PRACTICE & EXPERIENCE . 2011.

KIM, Svetlana. Smart Learning Services Based on Smart Cloud Computing. SENSORS . 2011.

KSHETRI, Nir, Cloud Computing in Developing Economies. University of North Carolina-Greensboro. 2010

LIN, Jimmy, Research and Education with MapReduce/Hadoop: Data Intensive Text Processing and Beyond. University of Maryland. 2009.

MADDEM, Sam, A Comparison of Approaches to Large-Scale Data Analysis, MIT - University of Wisconsin, University of Yale. 2009.

MARASSO, Lanfranco. Allowing Citizens to Self-compose Personalized Services: A Cloud Computing Model. Springer-Verlag Berlin Heidelberg 2010

Open Grid Forum, Sitio Web [En línea], [Julio 2011]. Disponible en:
http://www.ogf.org/About/abt_overview.php

Open Grid Services Arquitectura, Sitio Web [En línea], [Agosto 2011].
Disponible en: http://en.wikipedia.org/wiki/Open_Grid_Services_Architecture

Open Grid Services Infrastructure, Sitio Web [En línea], [Agosto 2011].
Disponible en: http://en.wikipedia.org/wiki/Open_Grid_Services_Infrastructure

Oracle Grid engine, Sitio Web [En línea], [Julio 2011]. Disponible en:
http://en.wikipedia.org/wiki/Oracle_Grid_Engine,

QUING, Li, Application-Services Integration among Multi-clouds for Small and Middle Scale Enterprises. Tsinghua University Beijing. 2010.

ROSALES FORERO, Edgar. UNA-CLOUD: Infraestructura como servicio para Cloud Computing Oportunista. Universidad de los Andes, Julio 2011. Bogotá.

Vmware Micro Cloud Foundry,, Sitio Web [En línea], [Octubre 2011]. Disponible en: <https://my.cloudfoundry.com/micro>,

Vmware Soporte Sitio Web [En línea], [Noviembre 2011]. Disponible en: <http://support.cloudfoundry.com/entries/20320253-introduction-to-usingmicro-cloud-foundry>

WEI Guiyi. A game theoretic method of fair resource allocation for cloud computing services. Springer Science+Business Media, LLC 2009

Weimar Díaz Garzón, realidad de la computación Grid en américa latina, pdf [En línea] ,[Julio2011]. Disponible en: http://www.sociedadelainformacion.com/12/Paper_Grid.pdf

WEIMING, Zheng. Design a cloud storage platform for pervasive computing environments.

WOOD, Timothy. CloudNet: Dynamic Pooling of Cloud Resources by Live WAN Migration of Virtual Machines. ACM SIGPLAN NOTICES . 2011.