

Análisis Semántico de los Aspectos Técnicos y Operaciones de Cloud Computing

Ana Sofía Zalazar¹, Luciana Ballejos², Sebastian Rodriguez¹

¹GITIA (UTN-FRT), Rivadavia 1050, 4000, Tucumán, Argentina
{ana.zalazar, sebastian.rodriguez}@gitia.org

²CIDISI (UTN-FRSF), Lavaisse 610, 3000, Santa Fe, Argentina
lballejo@frsf.utn.edu.ar

Abstract

Cloud Computing es un paradigma de negocio que suministra dinámicamente recursos de computación conectados a Internet, en el cual un proveedor ofrece estos recursos como servicios y los consumidores contratan el uso de servicios, aceptando un acuerdo de nivel de servicio. A través de este modelo de negocio el proveedor optimiza la utilización de su infraestructura tecnológica, mediante mecanismos de virtualización y distribución de procesos. Además, el consumidor adquiere las funcionalidades necesarias, en estos servicios de Cloud Computing, disminuyendo su inversión en tecnología y adquiriendo soluciones que se adaptan rápidamente a los cambios de sus requerimientos. El objetivo de este trabajo es proponer un modelo semántico para favorecer la comprensión de los aspectos técnicos y operacionales de los servicios de software ofrecidos en Cloud Computing. El modelo permite el análisis de estos aspectos dentro de los contratos de servicios, ya que son claves para adoptar este paradigma de negocio. Finalmente se valida la estructura del modelo utilizando unas métricas de ontologías y dos casos prácticos de contratos de servicios.

1. Introducción

Cloud Computing o computación en la nube es un paradigma de negocio que optimiza la utilización de infraestructura tecnológica, a través de técnicas de deslocalización, portabilidad, compartición de recursos y virtualización [1]. El proveedor ofrece servicios desplegados en un mismo servidor físico a un variado número de consumidores, y estos consumidores pagan por el uso de estos servicios. Las técnicas de virtualización permiten abstraer las capas de software, escalar y disminuir la capacidad reservada de los recursos, y aislar adecuadamente la infraestructura subyacente de estos servicios.

Se considera a Cloud Computing como una nueva manera de prestación de servicios, cuyo fin es el máximo aprovechamiento de la infraestructura informática, para evitar el sobre costo por recursos no utilizados y mal administrados [2].

La selección de servicios de Cloud Computing depende generalmente de los acuerdos de nivel de servicio (*Service Level Agreements*, SLAs) que son un tipo de contrato entre proveedores y consumidores, donde se identifican las características funcionales y de calidad de los servicios. Generalmente, un SLA incluye la descripción del servicio, los parámetros de nivel de servicio, las garantías ofrecidas por el proveedor, las obligaciones de los consumidores, los procedimientos de provisión del servicio, y el marco legal de los términos de uso. Los acuerdos son la base para comparar diferentes servicios, contratar proveedores, monitorear los indicadores de nivel de servicio y disparar una acción cuando el nivel de servicio acordado no está siendo alcanzado [3]. Por esto, es imprescindible que estos acuerdos tengan una semántica compartida por las partes que participan en la contratación del servicio, es decir deben poseer un lenguaje y vocabulario común, donde los términos de uso de los servicios sean claramente comprendidos por el consumidor y el proveedor.

Existen dos tipos principales de contratos en Cloud Computing: negociado o de adhesión. En un contrato negociado, el proveedor y el consumidor fijan conjuntamente las condiciones y los términos del acuerdo de servicio. En un contrato de adhesión de servicio, las cláusulas contractuales del proveedor son cerradas y los consumidores no pueden negociar estos acuerdos. La mayoría de los acuerdos de nivel de servicio para Cloud Computing no son negociables y son presentados en la página web del proveedor de servicio [4].

En los contratos de adhesión, el proveedor generalmente se reserva el derecho de actualizar de manera no predictiva los acuerdos de servicio, según los cambios dinámicos de sus políticas de negocio, y el consumidor debe aceptar chequear periódicamente los cambios y actualizaciones de estos contratos [5]. La

mayoría de estos acuerdos son genéricos, ambiguos y poco claros, y no representan las necesidades y requerimientos individuales de cada consumidor. Cuando el contrato es de adhesión, los consumidores no tienen forma de especificar sus requerimientos y restricciones a los grandes proveedores de servicios, por lo que se encuentra en desequilibrio con respecto a estos. Además, comparar acuerdos y contratos de servicios de Cloud Computing representa un gran desafío para los consumidores, por la cantidad de términos y vocabulario utilizado en estos documentos [5].

El objetivo de esta investigación es presentar los aspectos técnicos y operacionales de los servicios de Cloud Computing, para evaluar y analizar las características de los servicios. Para esto se propone un modelo semántico, en especial para el modelo de servicio “software como servicio” (*Software as a Service*, SaaS). El modelo presentado es una herramienta útil para capturar los términos más comunes de los servicios, sus relaciones y sus atributos, y además permite evaluar los contratos existentes. Para validar el modelo se utilizó un razonador y algunos ejemplos de SLA que fueron extraídos de la página web de dos grandes proveedores.

A continuación, se presentan las características de Cloud Computing que fundamentan la propuesta desarrollada. En la Sección 3, se definen los aspectos técnicos y operativos de los acuerdos de servicios, dando más detalles sobre la motivación de este trabajo. La Sección 4 introduce las características de las ontologías y se presentan los trabajos relacionados sobre las ontologías en Cloud Computing. En la sección 5 se presenta el modelo semántico desarrollado para presentar los aspectos de técnicos y operacionales. Las clases presentadas en el modelo permiten la capturar los datos necesarios para la subcontratación, la localización de los datos, la portabilidad y los atributos para medir los servicios. La aplicación de este modelo se describe en la sección 6. Finalmente, en la Sección 7 se presenta algunas recomendaciones sobre los servicios y en la Sección 8 se concluye el trabajo.

2. Características de Cloud Computing

Cloud Computing es un modelo de negocio basado en Internet, donde un proveedor ofrece sus recursos de computación como servicio (por ejemplo: memoria de almacenamiento, capacidad de procesamiento, conectividad de sus redes, etc.) y el consumidor utiliza el servicio después de aceptar el acuerdo asociado. Cuando un consumidor contrata un servicio, este puede desconocer totalmente la localización precisa de sus datos y no tiene el control directo sobre ellos [5].

Debido a los diferentes enfoques y la falta de un estándar, existen diferentes definiciones para Cloud

Computing [1]. Esta contribución considera la definición del *National Institute of Standards and Technology* (NIST), la cual introduce los aspectos más representativos de los entornos de Cloud Computing: “*Cloud Computing es un modelo que permite acceso a redes bajo demanda, para compartir un conjunto de recursos de computación configurable (es decir, redes, servidores, almacenamiento, aplicaciones y servicios) que pueden ser rápidamente provistos o liberados con un mínimo esfuerzo de administración o interacción con los proveedores de servicio*” [2].

El NIST considera cinco características esenciales de este modelo (autoservicio a demanda, amplio acceso a redes, puesta en común de los recursos, rápida elasticidad, y medición de servicios), cinco actores o roles principales (consumidor, proveedor, portador, bróker y auditor), tres modelos de servicio (infraestructura como un servicio, plataforma como un servicio, y software como un servicio) y cuatro modelos de despliegue (nube privada, nube pública, nube comunitaria, y nube híbrida) [2]. En la Figura 1, se muestra un mapa conceptual sobre las clases principales de Cloud Computing.

Los modelos de despliegue son divididos de acuerdo a quiénes acceden a los servicios de Cloud Computing y cómo el servicio es desplegado de forma física y virtual. En cada modelo de despliegue se debe analizar las garantías de seguridad que proporcionan la integridad de los datos y la continuidad de servicios, antes de aceptar un contrato de servicio..

- **Nube Privada:** los recursos de computación (físicos y virtuales) son accedidos exclusivamente por una organización, bajo el rol de consumidor de servicio, que probablemente tiene múltiples usuarios. Las responsabilidades de gestión de los recursos y servicios son compartidos entre el proveedor (dueño del servicio) y el consumidor, que paga por la prestación de servicio.
- **Nube Pública:** la infraestructura (recurso físico) que sostiene el entorno de Cloud Computing es compartida por consumidores independientes y heterogéneos. Estos consumidores acceden a través de redes públicas, de forma dinámica, y abonando el servicio utilizado bajo demanda. Los servicios son generalmente localizados en los centros de datos del proveedor o en la infraestructura externa de terceras partes (subcontratistas), por lo que los acuerdos deben asegurar que cada instancia del servicio ofrecido y cada cuenta de los consumidores (recurso virtual) esté correctamente aislada de las demás sin representar riesgos de seguridad.

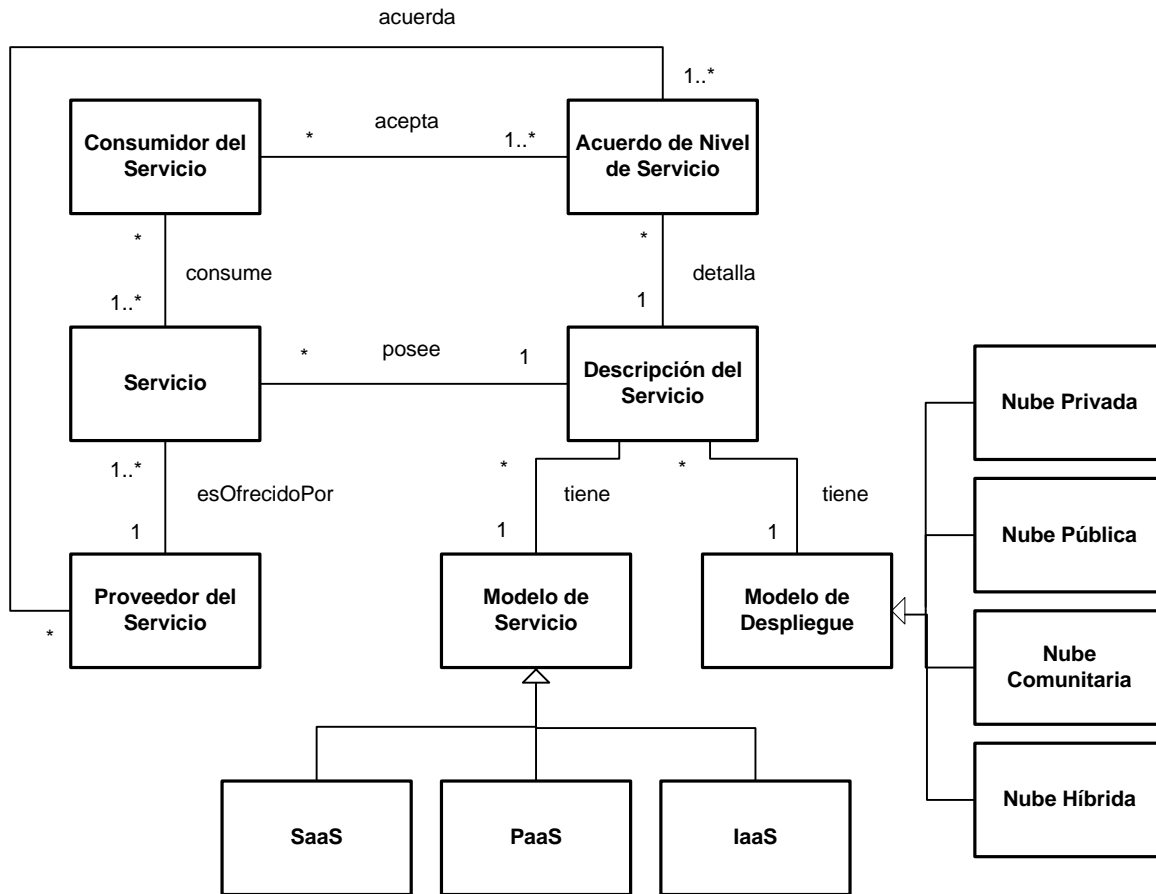


Figura 1. Visión General de Cloud Computing.

- **Nube Comunitaria:** los recursos de computación (físicos y virtuales) son compartidos y accedidos por una comunidad de consumidores y usuarios de Cloud Computing, quienes poseen características especiales y objetivos comunes que los hace pertenecer a la misma comunidad. El proveedor implementa políticas y reglas específicas para la administración de este tipo de nube, que son iguales para todos los usuarios de la misma comunidad.
- **Nube Híbrida:** Es la combinación de nubes privadas, nubes públicas, o nubes comunitarias. Esta combinación permite que se aprovechen las ventajas de rápido suministro y puesta en común de recursos físicos, sin poner en riesgo los datos sensibles y procesos de negocio, que son conservados bajo un modelo más restringido.

Los modelos de servicio representan el nivel de abstracción y virtualización del hardware, el nivel de responsabilidades de cada actor, y los derechos de control que tiene el consumidor. Los modelos presentados por el

NIST [2] forman parte del modelo “SPI” (*Software, Plataforma e Infraestructura*) que son genéricos. Sin embargo, otros modelos pueden derivarse de éstos, por ejemplo: comunicación como un servicio y base de datos como un servicio [6].

- **Software como un Servicio (SaaS):** El proveedor de servicio ofrece aplicaciones y componentes web basados en Internet, los cuales puede ser accedidos a través de navegadores y otras interfaces específicas. Estos tipos de servicios pueden implementar procesos específicos del consumidor, y este no tiene ninguna influencia ni control de la infraestructura tecnológica (servidores, sistemas operativos, memoria, etc.). El consumidor puede tener algunos permisos y derechos para configurar y personalizar externamente el servicio.
- **Plataforma como un Servicio (PaaS):** El proveedor de servicio ofrece unidades para desarrollar aplicación, es decir un contenedor con máquinas virtuales (VM), bases de datos, herramientas de

programación y entornos de desarrollo de sistemas. Este contenedor es aislado de los otros sistemas que pueden estar desplegados en el mismo recurso físico. El hardware es sólo controlado por el proveedor del servicio, quien se encarga de mantener actualizadas las librerías y de instalar las aplicaciones y las herramientas para la programación de sistemas.

- **Infraestructura como un Servicio (IaaS):** El proveedor de servicio administra todos los recursos físicos, las máquinas virtuales y las redes de comunicación, mientras que el consumidor tiene control limitado sobre el sistema operativo, aplicaciones de software, memoria, base de datos y recursos de redes. El proveedor proporciona las capacidades de almacenamiento y procesamiento. Mediante los mecanismos de virtualización, los consumidores manejan las instancias del servicio, como si tuvieran control absoluto de la infraestructura subyacente.

Cuando una organización debe asegurar altos niveles de seguridad de datos y procesos, seguramente contratará una nube privada y desplegará parte del servicio “*on-premise*”, es decir que la información sensible se mantendrá dentro de su propio servidor físico.

La solución más recomendada es la combinación de las nubes públicas y privadas, de manera que el consumidor pueda controlar sus datos sensibles y procesos críticos dentro de sus servidores, y al mismo tiempo aprovechar los beneficios de tercerización de mantenimiento de infraestructura para obtener una rápida escalabilidad de los servicios.

Antes de aceptar cualquier contrato o acuerdo de nivel de servicio, los consumidores deben analizar los modelos presentados, para comprender las características y el riesgo de seguridad asociados a cada modelo. Mientras menor control tenga un consumidor sobre la administración de los servicios, mayor será el nivel de riesgo de seguridad y pérdidas de datos, por lo tanto debe considerar qué datos sensibles son almacenados en un entorno compartido y qué procesos de terceras partes manipulan estos datos.

3. Aspectos Técnicos y Operativos de los Acuerdos de Servicio

Existen diversos documentos que definen condiciones y términos de negociación entre el proveedor y el consumidor de servicios de Cloud Computing, por ejemplo los “*términos de usos*” (ToU), “*términos de servicios*” (ToS), y “*acuerdo de licencia del usuario*” (ULA). Estos términos contractuales muchas veces reemplazan al SLA, pero todos estos documentos también cuentan con tres partes principales: promesas,

limitaciones y obligaciones [7]. Las promesas describen las responsabilidades de los proveedores que ofrecen los servicios de Cloud Computing, como proporcionar cierto porcentaje de disponibilidad del servicio, encargarse de monitorear los servicios, y cumplir las políticas de privacidad y seguridad. Las limitaciones indican las restricciones de responsabilidad del proveedor cuando suceden ciertos eventos fuera de su control, como son las catástrofes naturales, los cambios en la jurisdicción de un país, y/o las leyes impuestas por un gobierno. Las obligaciones en un contrato se aplican al rol del consumidor que asume compromisos al adquirir un servicio, como por ejemplo pagar la utilización del servicio, no realizar ningún acto ilícito utilizando el servicio, y poseer software autorizado en los entornos de Cloud Computing. Adicionalmente, el proveedor se reserva el derecho de cambiar los acuerdos o las características del servicio en cualquier momento, mientras el consumidor tiene la obligación de chequear periódicamente los acuerdos y mantenerse informado de estos cambios en la adhesión a los términos del contrato.

El proveedor de un servicio puede a la vez subcontratar a terceros y, a su vez, estos subcontratistas pueden tener sus propios proveedores indefinidamente. Por lo tanto, los datos pueden estar en cualquier lugar en cualquier momento, pero los proveedores deben garantizar la integridad, la seguridad y la disponibilidad de los mismos. Además, la portabilidad también es uno de los factores más importantes, ya que un consumidor debe tener la facilidad de transferir todos sus datos y procesos desde cualquier proveedor a otro. Se debe garantizar siempre que los derechos y obligaciones se cumplen, independientemente del lugar físico de los datos.

Los aspectos técnicos y operacionales de un acuerdo en Cloud Computing, son los aspectos más importantes durante la contratación de un servicio. Estos aspectos comprenden las funcionalidades y las competencias de un servicio, y aquellas operaciones que aseguran que el servicio estará disponible para su utilización. Los aspectos técnicos abarcan indicadores, valores, unidades, funciones, métricas, y métodos de medición que definen el nivel de un servicio de Cloud Computing, y los aspectos operacionales indican las actividades de mantenimiento, soporte, actualización, configuración del servicio, mecanismo de copia de resguardo, y monitoreo del servicio. La finalidad de estos aspectos es asegurar que el servicio estará disponible para ser utilizado y para alcanzar los niveles acordados.

Los aspectos técnicos son directamente medibles y calculables, y el nivel del servicio puede ser examinado por sus parámetros. Los aspectos operacionales tienen una repercusión directa sobre los aspectos técnicos ya que se encargan de mantener el servicio operativo y agregar recursos según la demanda. La elasticidad de los

servicios de Cloud Computing se logra a partir a la evaluación y el control de estos aspectos.

4. Diseño de la Ontología

Una ontología es un modelo conceptual que representa la semántica en un determinado dominio. Es decir, la ontología es una especificación explícita de una conceptualización de un dominio, y la misma comprende: (a) *Términos*: palabras y grupo de palabras que identifican las entidades del dominio, por ejemplo: proveedor, consumidor, etc.; (b) *Propiedades*: características o atributos de una entidad, por ejemplo: nombre del proveedor, identificación del consumidor, etc.; (c) *Relaciones*: elementos que conectan entidades del dominio en una ontología, por ejemplo la clase consumidor del servicio se conecta a la clase contrato a través de la relación tiene contrato o “*hasContract*”; (d) *Instancias*: valores individuales de una entidad del dominio o de la característica, por ejemplo “*Google*”, “*Amazon*”, y “*Microsoft*” que son instancias de la entidad proveedor; y (e) *Axiomas*: representan sentencias que son verdaderas para todo el dominio y, por lo tanto, son formalizadas en un lenguaje lógico, por ejemplo “ $\forall Contract, \exists ServiceProvider \wedge \exists ServiceConsumer$ ” que significa que para todo contrato existe un proveedor y un consumidor de servicio [8,9].

La web semántica explota los beneficios de las ontologías con el propósito de tener toda la información interconectada, de forma que sea fácil de manipular por máquinas y algoritmos. De esta manera, se pueden procesar datos, descubrir nuevas relaciones y transmitir nuevo conocimiento sobre un dominio [10].

Existen varios trabajos que proponen modelos semánticos y ontologías al área de Cloud Computing, pero no se encontró un enfoque bien definido que se tome como punto de partida para el análisis de acuerdos y contratos de servicios. Tampoco se encontró hasta el momento sobre los aspectos operacionales y técnicos que se tratan en este trabajo. Sin embargo, se destacan los siguientes autores en esta área.

Youseff y colaboradores [11] son unos de los primeros en publicar una ontología para Cloud Computing en un intento de establecer el dominio de conocimiento de esta área y sus componentes pertinentes. Los autores definen una estructura de cinco capas, sus interrelaciones y sus interdependencias, para lograr una mejor comprensión de este paradigma. La propuesta no entra en muchos detalles en los componentes, y tampoco hace la separación que se considera en este trabajo.

Androcec y colaboradores [12] realizan un análisis bibliográfico donde identifican 24 estudios sobre ontologías para Cloud Computing, en estos estudios los enfoques principales son los recursos, la descripción de los servicios, la seguridad, la interoperabilidad, y la

selección de servicios. A través de esta revisión bibliográfica se detectó que los mayores desafíos son la seguridad y la interoperabilidad en las ofertas de Cloud Computing, y los mayores obstáculos son referidos a la privacidad de la información. En este estudio, no se han identificado parámetros y atributos para manejar los requerimientos y acuerdos de servicios referidos a los aspectos operacionales y técnicos de Cloud Computing.

Kang y Sim [13] presentan “Cloudle” que es un motor de búsqueda para sistemas de Cloud Computing, el mismo es basado en dos ontologías que contienen un conjunto de conceptos, individuos de estos conceptos y las relaciones entre los individuos. Los autores afirman que a través de estos conceptos se pueden especificar los requerimientos funcionales, los requerimientos técnicos y los requerimientos de costos. Pero, los autores no analizan los acuerdos como se propone en esta contribución.

Ma y colaboradores [14] proponen una ontología basada en los sistemas de administración de recursos, para resolver problemas de los algoritmos y procesos en los ambientes de Cloud Computing. Si bien, esta ontología se utiliza para ubicar tareas y procedimientos en los recursos disponibles, también define conceptos y describe relaciones que son útiles para comprender acuerdos.

Moscato y colaboradores [15] proponen en el proyecto mOSAIC la definición de una ontología capaz de describir servicios y sus interfaces, y los conceptos utilizados son extraídos de los estándares y de la literatura existente. Esta ontología es construida para mejorar la interoperabilidad de los servicios en Cloud Computing.

5. Ontología Propuesta

La herramienta de edición de ontología que se eligió para esta contribución fue Protégé Desktop 5.1, un editor de código abierto desarrollado por la Universidad de Stanford. Esta herramienta soporta las últimas versiones del lenguaje de ontologías web (OWL) y se puede conectar con clasificadores deductivos para validar la consistencia de los modelos e inferir nueva información basada en el análisis de las clases de la ontología. La consistencia de la ontología propuesta fue validada con el razonador Pellet que se puede ejecutar como un plug-in en Protégé.

En Protégé el nombre del espacio, “*namespace*”, fue definido con el prefijo “*csla*”, en referencia a los contratos y acuerdos de servicios de Cloud Computing. Todos los términos identificados fueron organizados en una estructura jerárquica, que permitió detectar las

¹ Protégé Desktop 5: <http://protege.stanford.edu/products.php#desktop-protege> (Mayo 2016)

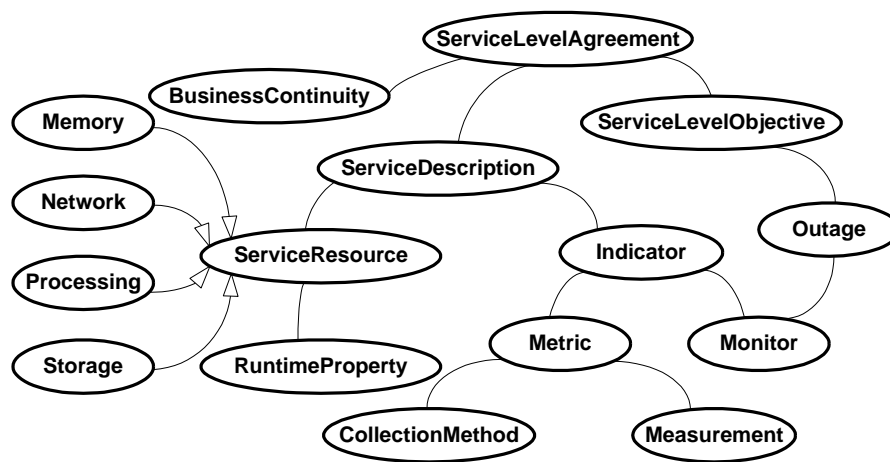


Figura 2. Ontología Propuesta.

primeras relaciones de términos. Luego, se identificaron las clases de la ontología (*Class*), los atributos (*Data Type Property*) y las relaciones (*Object Property*). Cada clase tiene una lista de valores de rango y dominio, y cada relación tiene una cardinalidad.

En esta contribución, se construyó un modelo semántico para representar los aspectos técnicos y operacionales de los servicios de software. Esta ontología es extensible y tiene el fin de facilitar la comunicación entre diferentes actores, proponer un vocabulario común y una estructura de términos relacionados. La ontología fue creada utilizando referencias científicas (artículos de revistas, conferencias, y contribuciones para definir estándares), experiencias referidas a la industria de software (foros, blogs y redes sociales) y algunos acuerdos de servicios de proveedores de Cloud Computing. La representación gráfica de la ontología se puede visualizar en la Figura 2.

En la Tabla 1 se presentan las clases, los atributos y las relaciones identificadas sobre los aspectos técnicos y operacionales de los servicios desplegados en Cloud Computing. La cardinalidad de las relaciones se presenta entre corchetes, indicando con asterisco la cardinalidad múltiple.

En la ontología propuesta, el acuerdo de nivel de servicio (*ServiceLevelAgreement* en Tabla 1) contiene la descripción de un servicio (*ServiceDescription*) que especifica varios recursos (*ServiceResource*). Un servicio en Cloud Computing se configura a través de un archivo de parámetros (*ConfigurationSettingFile*), que contiene todos aquellos valores necesarios para ejecutar el servicio.

La descripción del servicio (*ServiceDescription*) es la entidad más relevante de la ontología, ya que especifica las funcionalidades del servicio y sus características más importantes como el modelo de

servicio (*ServiceModel*), el modelo de despliegue (*DeploymentModel*) y la manera de aprovisionamiento de los recursos del servicio (*ProvisionModel*), que puede ser de forma estática, dinámica o a demanda.

Las propiedades de ejecución (*RuntimeProperty*) contienen la información referida a los recursos cuando el servicio de Cloud Computing se encuentra disponible y en estado operativo. Estas propiedades también indican los límites mínimos de capacidad reservados para el servicio (*LowerLimitCapacity*) y la máxima capacidad de recursos que el servicio puede utilizar para su ejecución (*UpperLimitCapacity*). Cuando se supera el límite máximo, es probable que el consumidor del servicio deba realizar una petición especial para subir los márgenes de capacidad. Las reglas de elasticidad y el tiempo de escalabilidad de un servicio (*ElasticityRuler* y *ScalabilityTime* en Tabla 1, respectivamente) limitan el rápido aprovisionamiento de los recursos para la ejecución de un servicio. De esta manera, cuando existe un pico de demanda, las reglas de elasticidad dispararán las tareas para agregar más capacidad a los servicios, y cuando exista un nivel muy bajo de demanda del servicio, estas reglas indicarán que se pueden liberar recursos.

La clase objetivo de nivel de servicio (*ServiceLevelObjective*) indica el nivel esperado del servicio y se relaciona con los indicadores (*Indicator*), que son índices de calidad y performance. Estos indicadores utilizan métricas, método de recolección de medidas y medición del servicio (*Metric*, *CollectionMethod* y *Measurement*, respectivamente). Cada servicio tiene asociado una o más métricas que se relacionan a los indicadores y miden el nivel del servicio. Existe una frecuencia de recolección de métricas (*CollectionFrequency*) y también reglas de recolección (*CollectionRuler*) y la frecuencia de colección de

medidas varía de ciclos de segundos a semanas, o bien cuando aparezca algún evento que se haya configurado en la regla de recolección de medidas.

Para que el servicio siempre se encuentre operativo durante el acuerdo de nivel de servicio, se considera un plan de continuidad (“*BusinessContinuity*”), que consiste en políticas de copias de resguardo que se almacenan en un repositorio (“*ServiceBackupPath*”). También se utilizan reglas de recuperación del servicio (“*ServiceRestoreRuler*”) y archivos del plan de contingencia (“*ContingencyPlan*”). La clase monitor (“*Monitor*”) controla los indicadores y realiza un seguimiento de los cortes del servicio (“*Outage*”) cuando el servicio no se encuentra disponible o está disponible por debajo del nivel objetivo. Cuando existe un corte, una penalidad puede ser aplicada (“*OutagePenalty*”).

Los indicadores y parámetros de nivel de servicio no fueron estandarizados todavía para Cloud Computing, por lo tanto cada contrato y acuerdo de servicio presenta una manera diferente de definirlos y calcularlos. Por ejemplo, no es lo mismo comparar una disponibilidad considerada al 99.99% mensual, que una disponibilidad considerada al 99.99% diaria, ya que mensualmente existe más probabilidad que el servicio falle. De la misma manera, los indicadores de performance muchas veces hacen referencia a métricas y medidas diferentes, como por ejemplo, la performance calculada como cantidad de registros procesados por minuto y la performance calculada como tiempo de respuestas a una petición de servicio. Por consiguiente, compartir las clases de la ontología propuesta permite desambiguar algunos términos.

Para validar la consistencia de la ontología, se ejecutó el razonador *Pellet* que indicó que no existía ninguna inconsistencia en la estructura de la misma. Adicionalmente, se calculó la calidad y complejidad de la ontología según el ancho y la profundidad de su estructura [16,17]:

- **Ancho** (*width*): número de clases visibles [17] equivalente al número promedio de subclases en una clase [16]. En la ontología propuesta tenemos sólo 4 subclases de la clase recurso del servicio (“*ServiceResource*”) que son: recursos de memoria (“*Memory*”), recursos de red (“*Network*”), recursos de procesamiento (“*Processing*”) y recursos de almacenamiento (“*Storage*”) y el número total de clases en la ontología propuesta que es 16. Por lo tanto, el ancho es igual a 0,5.
- **Profundidad** (*depth*): número de niveles de la jerarquía de una ontología, desde el término raíz hasta el término hoja [16]. Se tomó la clase acuerdo de nivel de servicio (“*ServiceLevelAgreement*”) como término raíz de esta ontología, y corte de servicio (“*Outage*”)

como término hoja. Finalmente, la profundidad es igual a 5.

A través del ancho de la ontología propuesta se puede concluir que es de naturaleza vertical y demuestra conocimiento detallado, y por la profundidad se considera de baja complejidad [16].

6. Aplicación en los Servicios de Cloud

En esta sección se presenta la aplicación de la ontología para comparar dos servicios de software de herramientas ofimáticas ofrecidos en Cloud Computing: *Google Apps for Work*² y *Office 365 Business*³. Sin lugar a dudas, se recomienda comparar las funcionalidades ofrecidas por varios proveedores, no sólo en términos económicos sino también diferentes aspectos como cumplimiento legal, garantías de calidad, y la forma de prestación del servicio.

Las herramientas ofimáticas son programas que dan soporte al trabajo en el ámbito de oficinas. Las más comunes de este tipo son los procesadores de texto, hoja de cálculos, editores de presentación, gestor de agendas y gestor de archivos compartidos. La ventaja de estos servicios en línea es que permiten crear, compartir, modificar, e intercambiar archivos entre varios usuarios en tiempo real. A continuación se comparan estas herramientas:

- **Google Apps for Work:** Es un servicio de aplicaciones web de Google Inc. con funcionalidades similares a las herramientas ofimáticas. Los costos son por cuenta de usuario de un consumidor al mes, y proporciona a partir de 30 GB por cuenta y un soporte de 24/7 telefónico y por correo electrónico. Al evaluar el acuerdo de nivel de servicio (“*ServiceReference*” en Tabla 1) “*Google Apps SLA*”, se concluyó que se trata de una nube pública y un aprovisionamiento de cuentas de usuario a demanda. La especificación de servicio (“*ServiceSpecification*”) consiste en las siguientes aplicaciones: “Gmail, Google Calendar, Google Talk, Google Docs and Drive, Google Groups, Google Sites and Google Apps Vault components of the Service”. También se presentan algunos indicadores (“*Indicator*”) para medir el nivel objetivo del servicio (“*ServiceLevelObjective*”). Por ejemplo, el indicador de disponibilidad (“*Availability*” en *Google Apps SLA*) se mide mensualmente y su nivel objetivo es 99.9% de disponibilidad. También se tiene en cuenta en los aspectos técnicos el tiempo de

²Google Apps: <https://www.google.com/apps/intl/en-GB/terms/sla.html> (Mayo 2016)

³Office 365: <https://technet.microsoft.com/en-us/library/office-365-service-descriptions.aspx> (Mayo 2016)

inactividad del servicio (“*Downtime*” en Google Apps SLA), que se calcula como el porcentaje de error por parte del servidor, y su nivel objetivo es 5% de error. El porcentaje de tiempo de actividad mensual (“*Monthly Uptime Percentage*” en Google Apps SLA) es uno de los indicadores más importantes y se mide considerando los minutos del calendario de un mes, menos el número de minutos en el cual el servicio no estuvo disponible, dividido por el total de números de minutos del calendario de un mes. Los incidentes (“*Outage*” en Tabla 1 y Google Apps SLA) también se consideran un indicador, y cuando esto sucede la penalidad (“*Penalty*” en Tabla 1) es medida por créditos de servicios a favor del consumidor.

- **Office 365 Business:** Es una alternativa en Cloud Computing de las herramientas de Microsoft Office de Microsoft Corporation, para la comunicación y el trabajo colaborativo. El servicio se paga mensualmente por cantidad de cuentas de usuario del consumidor del servicio y la capacidad de memoria es a partir de los 50 GB por cuenta. Al evaluar el acuerdo de nivel de servicio (“*ServiceReference*” en Tabla 1) “SLA Microsoft Online Service”, se concluyó que se trata de una nube pública y un aprovisionamiento de cuentas de usuario a demanda, proporcionando cuentas a partir de 50 GB disponibles de memoria (“*ResourceCapacity*”). Para medir el nivel objetivo de los servicios (“*ServiceLevelObjective*”) se consideraron los mismos parámetros que para el análisis de Google Apps for Work. El indicador (“*Indicator*”) tiempo de inactividad (“*Downtime*” en SLA Microsoft Online Service) se refiere al período de tiempo en que las aplicaciones de Office Online no está disponible o su disponibilidad es reducida por algún incidente en los ambientes del proveedor, y se mide sumando la duración en minutos de cada incidente en ese mes, multiplicado por el número de usuarios afectados por tal incidente. El porcentaje de tiempo de actividad mensual (“*Monthly Uptime Percentage*” en SLA Microsoft Online Service) se calcula como minutos por usuario al mes, menos minutos del tiempo de inactividad, dividido por minutos por usuarios por 100. Los incidentes (“*Outage*” en Tabla 1 y SLA Microsoft Online Service) hacen referencia dentro de este servicio a eventos únicos o conjunto de eventos que tengan como resultado un tiempo de inactividad, y son diferentes al tiempo de inactividad programados que se relacionan con actualizaciones o mantenimientos en el servicio, siendo notificados al menos 5 días antes del que este incidente se lleve a cabo.

Como se analiza en estos contratos, la representación semántica puede ayudar a los consumidores de servicio a

extraer información de los acuerdos de nivel de servicio, y a partir de la ontología propuesta se pueden interpretar y comparar estos acuerdos. Los proveedores y productos analizados en esta sección básicamente ofrecen el mismo tipo de servicio, pero algunos términos y parámetros difieren mínimamente.

7. Algunas Recomendaciones sobre la Contratación de Servicios

La utilización de servicios en Cloud Computing ofrece muchas ventajas, pero presenta también muchos riesgos, como la falta de transparencias y la falta de control, y esto se debe principalmente por las dificultades para conocer en todo momento la ubicación de los datos y los procesos. El consumidor del servicio debe evaluar la tipología de sus datos e identificar los datos sensibles, que no desea poner bajo estos riesgos [10].

La aceptación del empleo de servicios ofrecidos bajo la arquitectura de Cloud Computing depende ampliamente de la manera que los mismos cumplan con los requerimientos. Sin embargo, la seguridad y los contratos de servicios son los aspectos más criticados de las soluciones de Cloud Computing. Generalmente la brecha de seguridad se crea porque los centros de datos de los proveedores de servicio no están ubicados en la misma geografía que el consumidor del servicio, y por lo tanto el proveedor no está obligado a cumplir los mismos aspectos legales. Consecuentemente, el consumidor de servicio debe evaluar las políticas de seguridad ofrecidas en el SLA y asegurarse que se ajustan a las necesidades de su organización.

En el siguiente listado se encuentran los aspectos de seguridad más importantes considerados por la *European Network and Information Security Agency* (ENISA) [7] y algunas recomendaciones sobre los mismos:

- **Protección, Seguridad de Datos y Propiedad Intelectual:** La localización de los datos y de los procesos es importante ya que las garantías y obligaciones legales pueden ser distintas según los países donde estos se encuentre. Por lo tanto, en este punto se debe analizar aspectos de seguridad [18] como la localización, la autenticidad, la integridad de los datos y los servicios, la operatividad durante un periodo de tiempo, y la confidencialidad de la información. Antes de adquirir un servicio, donde se delegue la manipulación de datos sensibles de la organización, se debe analizar que el contrato estipule mecanismos de protección, garantías de tratamiento lícito y compensaciones ante violación de las cláusulas de seguridad. Además, el proveedor debe notificar cuando existan amenazas, riesgos o incidentes que

afecten la integridad, la confidencialidad y la disponibilidad de la información del consumidor.

- **Transferencia de Información:** La protección de la información almacenada y en tránsito es fundamental en este paradigma. Se debe Garantizar la protección adecuada de los datos, aun cuando el origen/destino de la transferencia sea de diferente jurisdicción.
- **Confidencialidad y no divulgación:** El consumidor del servicio debería analizar las políticas de confidencialidad y no divulgación de sus datos, además debe saber qué información circulará en los entornos de Cloud Computing, ya que estos datos pueden almacenarse y procesarse sin su consentimiento.
- **Limitación de la responsabilidad:** Se debe tener en cuenta las políticas de subcontratación a terceros (subcontratistas), para limitar el control y las responsabilidades de los servicios. Además, se deberá considerar los riesgos y los límites de responsabilidad cuando no se cumpla un contrato. Las compensaciones asociadas deben ser de la dimensión de los riesgos.
- **Análisis de impacto al cambio de control:** Los subcontratistas deben ofrecer las mismas garantías y cumplir las mismas obligaciones legales que los proveedores de servicio han asumido en el acuerdo. Por lo tanto, se debe derivar las responsabilidades y las obligaciones contractuales cuando el proveedor realice cambio de control o subcontrataciones sin consentimiento del consumidor.
- **Portabilidad de datos y funcionalidad:** El procedimiento de salida al cambiar de proveedor debe asegurar que no se disminuyen la integridad de los datos. Para esto, se debe exigir en los contratos de negocio que la transferencia de datos y documentos entre proveedores del servicio pueda llevarse a cabo sin complicaciones, así el consumidor pueda migrar sus servicios de Cloud Computing cuando lo requiera necesario. La portabilidad de datos indica que, al resolverse el contrato o la terminación del servicio, el proveedor da la garantía de que se entregará toda la información a un formato apropiado.

Los aspectos operacionales y técnicos se complementan con estos puntos de seguridad. Antes de contratar, el proveedor y el consumidor del servicio deben realizar el análisis de riesgos, en función a la sensibilidad de los datos, la localización de los mismos, las amenazas, y los niveles de control.

Los riesgos de seguridad, no sólo se pueden presentar en los sistemas del proveedor del servicio, sino que también pueden ser riesgos internos de la infraestructura del consumidor y de las redes utilizadas en la contratación de servicios [10].

8. Conclusiones

Inicialmente este trabajo presenta una introducción a la temática de Cloud Computing, presentando sus características y modelos de servicios más importantes. Luego, se definieron algunos aspectos técnicos y operacionales de Cloud Computing que deben tenerse en cuenta antes de contratar un servicio de software.

Los servicios desplegados en Internet, sin dudas están cambiando la manera de hacer ingeniería de software y desarrollar sistemas de información. Para contribuir en esta área se propuso una ontología, donde pueden aplicarse herramientas de web semántica para extraer información y tomar decisiones referentes a la adquisición de servicios en Cloud Computing. La ontología presentada en esta contribución mantiene la compatibilidad con trabajos anteriores sobre ontologías para Cloud Computing.

Usualmente, cuando un consumidor decide contratar servicios de Cloud Computing, él selecciona los servicios candidatos según sus requerimientos funcionales y el grado de seguridad que le garantizan los acuerdos de nivel de servicio. Por lo tanto, esta contribución es un aporte a la automatización de esta tarea, a las futuras negociaciones de contratos, a la creación de acuerdos de servicios, y al control de los servicios desplegados en Cloud Computing. En este trabajo, se asume que la seguridad de los servicios que es una cuestión transversal en este campo de conocimiento.

La ontología propuesta en esta contribución es una herramienta útil para los especialistas que necesitan considerar características de los servicios, analizar riesgos, y elegir el servicio que mejor se ajuste a sus requerimientos funcionales. Generalmente, esta comparación de servicios y decisiones de adopción de herramientas en Cloud Computing es realizada por un especialista técnico. Por lo tanto, como trabajo futuro se trabajará en el análisis y ajuste de los algoritmos de análisis de lenguaje natural para mapear las instancias de los acuerdos con los conceptos claves de esta propuesta.

9. Referencias

- [1] L.M. Vaquero, L. Rodero-Merino, J. Caceres, y M. Lindner, "A break in the clouds: towards a cloud definition", *SIGCOMM Computer Communication Review*, 39, 1, 2008, pp. 50-55.
- [2] P. Mell y T. Grance, "The NIST Definition of Cloud Computing", *National Institute of Standards and Technology*, NIST Special Publication 800-145, 2011.
- [3] R. Buyya, C. S. Yeo, S. Venugopal, J. Broberg, e I. Brandic, "Cloud Computing and emerging IT platforms: Vision, hype, and reality for delivering computing as the 5th utility", *Future Generation computer systems*, 25, 6, 2009, pp 599-616.

- [4] J. Repschlaeger, R. Zarnekow, S. Wind, y T. Klaus, "Cloud requirement framework: Requirements and evaluation criteria to adopt cloud solutions", *20th European Conference on Information Systems*, 2012.
- [5] A. S. Zalazar, S. Gonnet, y H. Leone, "Un Modelo para Contratos de Cloud Computing", *42JAIIO – 14th Argentine Symposium on Software Engineering*, 2013, pp. 303-317.
- [6] V. J. Winkler, "*Securing the Cloud: Cloud Computing security techniques and tactics*". Boston: Syngress, 2011.
- [7] D. Catteddu y G. Hogben, "Cloud Computing: Benefits, risks and recommendations for information security". *European Network and Information Security Agency*, 2009.
- [8] T. Gruber, "A translation approach to portable ontology specifications", *Knowledge acquisition*, 5, 2, 1993, pp. 199-220.
- [9] A. Gomez-Perez, M. Fernandez-Lopez, y O. Corcho-Garcia, "*Ontological engineering: Advanced Information and Knowledge Processing*", Springer, 2004.
- [10] A. S. Zalazar, S. Rodriguez, y L. Ballejos, "Handling Dynamic Requirements in Cloud Computing", *44JAIIO – 16th Argentine Symposium on Software Engineering*, 2015, pp. 212-225.
- [11] L. Youseff, M. Butrico, y D. Da Silva, "Toward a unified ontology of cloud computing", *Grid Computing Environments Workshop*, IEEE, 2008, pp. 1-10.
- [12] D. Androcec, N. Vrcek, y J. Seva, "Cloud computing ontologies: A systematic review", *Third international conference on models and ontology-based design of protocols, architectures and services*, 2012, pp. 9-14.
- [13] J. Kang y K. M. Sim, "Ontology and search engine for cloud computing system", *International Conference on System Science and Engineering*, IEEE, 2011, pp. 276-281.
- [14] Y. B. Ma, S. H. Jang, y J. S. Lee, "Ontology-based resource management for cloud computing", *Asian Conference on Intelligent Information and Database Systems*, Springer, 2012, pp. 343-352.
- [15] F. Moscato, R. Aversa, B. Di Martino, T. F. Fortiș, y V. Munteanu, "An analysis of mosaic ontology for cloud resources annotation", *Federated Conference on Computer Science and Information Systems (FedCSIS)*, IEEE, 2011, pp. 973-980.
- [16] M. Rico, "*Soporte para enriquecer la representación de entidades en una ontología*", Tecnológica Nacional, Fac. Reg. Santa Fe, Tesis doctoral, Universidad, 2011.
- [17] R. M. Colomb, "*Quality of ontologies in interoperating information systems*", Technical report 18/02 ISIB-CNR, 2002.
- [18] Y. Cherdantseva, y J. Hilton, "Reference Model of Information Assurance & Security", *Eighth International Conference on Availability, Reliability and Security (ARES)*, IEEE, 2013, pp. 546-555.

Tabla 1. Elementos de la Ontología.

Class	Data Type Property	Object Property / Cardinality
ServiceLevel Agreement	ServiceReference: String LastUpdate: Date ServiceSpecification: String	specifiesService(ServiceDescription)[1,*] hasSLO(ServiceLevelObjective)[*,*]
ServiceDescription	ServiceReference: String ServiceModel: String ProvisioningModel: String DeploymentModel: String	isSpecified (ServiceLevelAgreement)[*,1] involvesResource(ServiceResource) [1,*]
Indicator	IndicatorName: String IndicatorReference: String IndicatorDescription: String Mandatory: Boolean Validity: Boolean	hasMetric(Metric) [1,*] referesTo(ServiceDescription) [1,*] isControlledBy(Monitor) [*,1]
ServiceResource	ResourceDescription: String ResourceIdentification: String IPAddress: String ConfigurationSetting: File ResourceCapacity: Float CapacityUnit: String	isRelatedTo(ServiceDescription) [*,1] hasRunProperty(RuntimeProperty) [1,1]
RuntimeProperty	UpperLimitCapacity: Float LowerLimitCapacity: Float CapacityUnit: String ElasticityRuler: String ScalabilityTime: Float TimeUnit: String	describesResource(ServiceResource) [1,1]
Memory/Network/ Processing/ Storage		isA(ServiceResource) [1,1]
BusinessContinuity	ServiceBackupPath: String ServiceRestoreRuler: String IncidentReport: String ContingencyPlan: File OperationalAuditing: String	isRelatedTo(ServiceLevelAgreement) [1,1]
ServiceLevel Objective	ServiceLevelTarget: String	hasIndicator(Indicator) [1,*]
Metric	MetricName: String MetricReference: String MetricDescription: String MetricScale: String MetricUnit: String	hasMethod(CollectionMethod) [1,*] hasMeasurement(Measurement) [1,*] hasMetric(Metric) [1,*]
CollectionMethod	CollectionFrecuency: Float CollectionRuler: String CollectionTimeUnit: String	supportsMetric(Metric) [1,*] influencesMeasurement(Measurement) [1,1]
Measurement	ActualValue: Float ValueUnit: String LastColletedTime: Date	hasMeasurement(Measurement) [1,*] isUsedBy(Metric)[*,1]
Monitor	MonitoringMethod: String MetricTracking: String MonitorLogPath: String	tracesOutage(Outage) [1,*] controls(Indicator) [1,*]
Outage	OutageDescription: String OutageDuration: Float OutageUnit: String OutageDate: Date OutagePlan: File OutagePenalty: Boolean	refersToSLO(ServiceLevelObjective) [*,1]