



ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL

Facultad de Ingeniería en Electricidad y Computación

“SIMULADOR PARA ASIGNACIÓN DE RECURSOS EN
NUBES BASADAS EN TEORÍA DE JUEGOS Y SISTEMAS
DE REPUTACIÓN”

TESIS DE GRADO

PREVIO A LA OBTENCIÓN DEL TÍTULO DE
INGENIERO EN CIENCIAS COMPUTACIONALES
ORIENTACIÓN SISTEMAS DE INFORMACIÓN

PRESENTADO POR:

JUAN MANUEL ROMERO SANTAMARÍA
JEFFERSON WLADIMIR VALDEZ COELLO

GUAYAQUIL – ECUADOR

AÑO: 2015

AGRADECIMIENTO

A Dios, a mi director de tesis Carlos Mera, a mis padres y a mi amigo Juan Romero por todo el apoyo y a aquellas personas que estuvieron conmigo en este largo camino.

Jefferson Wladimir Valdez Coello.

AGRADECIMIENTO

A Dios, a mis padres que me dieron su apoyo en todo momento, a mi director de tesis Carlos Mera por transmitirme sus conocimientos, experiencia y motivarme a siempre dar lo mejor de mí.

Juan Manuel Romero Santamaría

DEDICATORIA

Este proyecto lo dedico a Dios , a mis padres por el apoyo que me dieron en toda mi carrera profesional, y especialmente a Sara Zambrano .

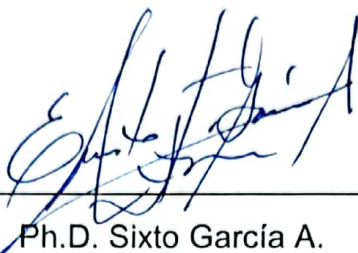
Jefferson Wladimir Valdez Coello.

DEDICATORIA

Dedico este trabajo a mi querida tía la Srta. Bárbara Santamaría por ser como una madre para mí y aconsejarme y brindarme todo su apoyo a lo largo de las diferentes etapas de mi vida.

Juan Manuel Romero Santamaría.

TRIBUNAL DE SUSTENTACIÓN



Ph.D. Sixto García A.
SUBDECANO SUBROGANTE



MSc. Carlos Mera G.
DIRECTOR DEL PROYECTO DE GRADUACIÓN




Ph.D. Carlos Monsalve A.
MIEMBRO DEL TRIBUNAL

DECLARACIÓN EXPRESA

“La responsabilidad del contenido de esta Tesis de Grado nos corresponde exclusivamente; y el patrimonio intelectual de la misma, a la Escuela Superior Politécnica del Litoral”

(Reglamento de exámenes y títulos profesionales de la ESPOL)



Jefferson Wladimir Valdez Coello



Juan Manuel Romero Santamaría

RESUMEN

El presente trabajo tiene como objetivo diseñar un modelo basado en teoría de juegos no cooperativos y mecanismos de reputación, que ayude a mejorar la asignación de recursos en Cloud Computing y permita minimizar las violaciones SLA.

El modelo propuesto trata de minimizar el número de violaciones SLA, incluyendo la reputación de un proveedor Cloud como un nuevo parámetro en el proceso de negociación entre cliente y proveedor, para ello se modela este proceso como un juego no cooperativo, en el cual los proveedores Cloud compiten entre sí por la asignación de recursos de una tarea requerida por un usuario Cloud, y postulan sus ofertas en orden de menor a mayor reputación, las mismas que serán procesadas por un agente intermediario, el cual utilizará el mecanismo de reputación beta para seleccionar al proveedor que más se ajuste a los criterios de selección del usuario Cloud.

Para realizar la etapa de experimentación, se implementará un simulador sobre el framework CloudSim, el cual se desarrollará en un ambiente web, utilizando el framework JSF y la librería de componentes PrimeFaces para

mostrar los resultados gráficamente y facilitar la creación de interfaces de usuario.

ÍNDICE GENERAL

AGRADECIMIENTO	II
AGRADECIMIENTO	III
DEDICATORIA	IV
DEDICATORIA	V
TRIBUNAL DE SUSTENTACIÓN	VI
DECLARACIÓN EXPRESA	VII
RESUMEN	VIII
ÍNDICE GENERAL	X
ABREVIATURAS	XIII
GLOSARIO	XIV
ÍNDICE DE FIGURAS	XVI
ÍNDICE DE TABLAS	XVIII
INTRODUCCIÓN	XIX
CAPÍTULO 1	1
1 ANTECEDENTES Y JUSTIFICACIÓN	1
1.1 DESCRIPCION DEL PROBLEMA	1
1.2 ANTECEDENTES	2
1.3 JUSTIFICACIÓN	8
CAPÍTULO 2	13
2 DEFINICIONES Y TRABAJOS RELACIONADOS	13
2.1 CLOUD COMPUTING	13
2.1.1 CARACTERÍSTICAS DE CLOUD COMPUTING	15
2.1.2 TIPOS DE ESQUEMAS CLOUD, SEGÚN EL MODELO DE IMPLEMENTACIÓN	18
2.1.3 TIPOS DE ESQUEMA CLOUD COMPUTING, SEGÚN EL MODELO DE SERVICIO ...	20
2.1.4 INFRAESTRUCTURA COMO SERVICIO (IaaS)	22
2.1.5 PLATAFORMA COMO SERVICIO (PaaS)	24
2.1.6 SOFTWARE COMO SERVICIO (SaaS)	25
2.2 SISTEMAS DE REPUTACIÓN	26
2.2.1 MÉTRICAS EN SISTEMA DE REPUTACIÓN	27

2.3	ECONOMÍA EXPERIMENTAL	30
2.3.1	TEORÍA DE JUEGOS.....	32
2.3.2	REPRESENTACIÓN DE JUEGOS.....	34
2.4	SERVICE LEVEL AGREEMENT (SLA).....	38
2.5	SOFTWARE DE SIMULACIÓN CLOUDSIM	39
2.5.1	ARQUITECTURA CLOUDSIM.....	41
2.6	TRABAJOS REALIZADOS	43
CAPÍTULO 3		45
3	DISEÑO DEL MODELO.....	45
3.1	DISEÑO DE LA EXPERIMENTACIÓN.....	45
3.1.1	DESCRIPCIÓN DEL MODELO	45
3.1.2	MECANISMO DE REPUTACIÓN BETA	48
3.2	VARIABLES MANIPULADAS.....	49
3.3	VARIABLES CONTROLADAS.....	50
3.4	DESCRIPCIÓN DEL ENTORNO.....	51
CAPÍTULO 4		55
4	DISEÑO DEL SIMULADOR.....	55
4.1	DIAGRAMA DE CASOS DE USO	55
4.1.1	ESPECIFICACIONES DE CASOS DE USO.....	56
4.2	DIAGRAMA DE COMPONENTES.....	59
4.3	DIAGRAMA DE CLASES.....	60
4.4	DIAGRAMA DE SECUENCIA	63
CAPÍTULO 5		64
5	IMPLEMENTACIONES Y PRUEBAS.....	64
5.1	ALGORITMO DE ASIGNACIÓN DE RECURSOS	64
5.2	CARACTERÍSTICAS DEL SIMULADOR	67
5.3	CONFIGURACIÓN DEL JUEGO	68
5.4	CONFIGURACIÓN DE LOS PARÁMETROS DEL JUEGO	71
5.5	EXPORTANDO E IMPORTANDO ESCENARIOS	72
5.6	REPORTE DE SIMULACIÓN.....	74
5.7	PRUEBAS APLICADAS A LA HERRAMIENTA DE SIMULACIÓN.....	76

CAPÍTULO 6	77
6 ANÁLISIS DE RESULTADOS.....	77
6.1 ANÁLISIS DE RESULTADOS.....	77
CONCLUSIONES.....	85
RECOMENDACIONES.....	87
ANEXO A	88
ANEXO B	90
ANEXO C.....	92
ANEXO D	93
BIBLIOGRAFÍA	96

ABREVIATURAS

AMD	Advanced Micro Devices
API	Application Programming Interface
CRGAG	Cloud Resource Allocation Game
IAAS	Infrastructure as a Service
ITIL	Information Technology Infrastructure Library
JSF	Java Server Faces
MIPS	Millions of Instructions per Second
MIT	Massachusetts Institute of Technology
NIST	National Institute of Standards and Technology
PAAS	Platform as a Service
SAAS	Software as a Service
SLA	Service Level Agreement
TI	Information Technology
XML	eXtensible Markup Language

GLOSARIO

Calidad De Servicio (QoS): Es el rendimiento promedio de una red de telefonía o de computadoras, particularmente el rendimiento visto por los usuarios de la red

Cloud Computing: Cloud Computing es un modelo para habilitar acceso conveniente por demanda a un conjunto compartido de recursos computacionales configurables, por ejemplo redes, servidores, almacenamiento, aplicaciones y servicios.

Datacenter: Se denomina datacenter o centro de procesamiento de datos (CPD) a aquella ubicación donde se encuentran alojados los recursos necesarios para el procesamiento de información de una organización.

Proveedor Cloud (Cloud Provider): Proveedor Cloud es una compañía que ofrece servicios de almacenamiento o de servicios de software disponibles a través de una nube privada o pública.

Reputación: Es una opinión sobre una organización dada por los resultados de una evaluación social.

Startup: Es un término utilizado actualmente en el mundo empresarial para denotar una compañía que busca arrancar, emprender o montar un nuevo negocio y hace referencia a ideas de negocios que están empezando o están en construcción, es decir son empresas emergentes apoyadas en la tecnología

TI: Es la aplicación de ordenadores y equipos de telecomunicación para almacenar, recuperar, transmitir y manipular datos, con frecuencia utilizado en el contexto de los negocios u otras empresas.

Virtualización: Es la creación a través de software de alguna versión virtual de algún recurso tecnológico, como puede ser una plataforma de hardware o software o un dispositivo de almacenamiento.

ÍNDICE DE FIGURAS

Figura 2.1 Modelo de servicios de Cloud Computing junto al cliente objetivo por nivel	22
Figura 2.2 Arquitectura de un Sistema de Reputación.....	27
Figura 2.3 Representación de un juego no cooperativo.....	33
Figura 2.4 Representación de un juego cooperativo.....	34
Figura 2.5 Dilema del prisionero en representación extensiva	38
Figura 2.6 Representación de un SLA	39
Figura 2.7 Arquitectura de Cloudsim	43
Figura 3.1 Variables Manipuladas CRAG	50
Figura 3.2 Representación gráfica del modelo.....	54
Figura 4.1 Diagrama de Casos de uso del simulador	55
Figura 4.2 Diagrama de Componentes del Simulador	59
Figura 4.3 Diagrama de clases del simulador	61
Figura 5.1 Configuración de proveedores Cloud en el simulador	70
Figura 5.2 Configuración de trabajos planificados en el simulador.....	71
Figura 5.3 Parámetros de configuración del simulador.....	72
Figura 5.4 Exportación de escenario.....	73
Figura 5.5 Importación de escenario.....	74
Figura 5.6 Carga de escenario.....	74
Figura 5.7 Violaciones SLA por Proveedor	75
Figura 5.8 Violaciones SLA vs Importancia de reputación.....	75
Figura 6.1 Violaciones SLA en simulación 1	79
Figura 6.2 Violaciones SLA en simulación 2	80

Figura 6.3 Violaciones SLA en simulación 3	81
Figura 6.4 Violaciones SLA en simulación 4	82
Figura 6.5 Violaciones SLA en simulación 5	83
Figura 6.6 Importancia de reputación vs violaciones SLA	83

ÍNDICE DE TABLAS

Tabla 1 Razones para migrar a la nube	14
Tabla 2 Dilema del prisionero en forma normal	31
Tabla 3 Representación General de un juego con 2 acciones y 2 jugadores en forma normal	36
Tabla 4 Elementos de un juego en representación extensiva.....	37
Tabla 5 Escenarios de simulación.....	78
Tabla 6 Resultados para simulación con importancia de reputación: 2	79
Tabla 7 Resultados para simulación con importancia de reputación: 4	80
Tabla 8 Resultados para simulación con importancia de reputación: 6	80
Tabla 9 Resultados para simulación con importancia de reputación: 8	81
Tabla 10 Resultados para simulación con importancia de reputación: 10	82

INTRODUCCIÓN

Cloud Computing es un modelo computacional que propone una nueva forma de gestionar servicios de tecnologías de información y aplicaciones. Este modelo se caracteriza por ofrecer a los usuarios finales un conjunto de servicios bien definido, para responder a las necesidades de una empresa o negocio, reduciendo así costos en tiempo y contratación de personal especializado para realizar tareas cotidianas que a menudo se presentan en el mundo de las tecnologías de información.

Una de las características mas importantes de esta tecnología es la escalabilidad y el pago por el uso del servicio , ya que esta diseñada para ajustarse a medida que las necesidades de una organización aumentan o requieren de mas poder computacional, así pues el usuario final solamente paga por el uso de servicios y se olvida de otros costos que normalmente serian considerados si no se contaran con estos beneficios y se trabajara de la forma tradicional.

CAPÍTULO 1

1 ANTECEDENTES Y JUSTIFICACIÓN

1.1 DESCRIPCION DEL PROBLEMA

El objetivo principal de esta investigación se centra en diseñar un nuevo modelo basado en teoría de juegos, el cual haciendo uso de un mecanismo de reputación que lleva un registro histórico de eventos, prediga futuros comportamientos que ayuden a mejorar la asignación de recursos en la nube y reduzcan las violaciones de parámetros de calidad de servicio (QoS), así como también la gran brecha que puede haber entre lo que el usuario necesita y lo que el proveedor está en capacidad de ofrecer.

De esta manera, se consigue que el precio del servicio requerido por el usuario no sea la única variable que influya en el proceso de

toma de decisión, y se considere la reputación de un vendedor como parte influyente de la decisión final.

1.2 ANTECEDENTES

El mundo actual en el que vivimos se encuentra bajo constante evolución. La llegada del Internet y la globalización permitieron a las empresas expandir sus fronteras y aumentar su alcance para ofertar sus productos, lo que generó una gran competitividad en el mercado. Para no estar un paso atrás, las instituciones de hoy en día se ven obligadas a actualizarse continuamente para sostener su rentabilidad.

Las empresas pasaron de sus registros manuales a complejas infraestructuras de manejo de información llamados datacenters. Generalmente se invierten considerables sumas de dinero en adquirir recursos informáticos (tanto de hardware como de software) para poder contar con un mejor soporte en la administración de la información y facilitar los procesos de toma de decisiones y planificación estratégica, que serán el pilar fundamental para

optimizar los procesos del giro de negocio y obtener una ventaja competitiva.

A medida que aparecen nuevos productos y servicios, algunas empresas se ven obligadas a crecer tanto en infraestructura como en personal, para satisfacer las necesidades del mercado vigente, lo que trae consigo costos adicionales para mantener constante la calidad del servicio ofrecido. Estos costos adicionales pueden ser: compras de licencia de software, actualización de equipos informáticos, contratación y capacitación de personal y mantenimiento en general.

En el año de 1961, durante un discurso para celebrar el centenario del instituto tecnológico de Massachusetts (M.I.T.), John McCarthy (1927-2011), inventor del lenguaje de programación LISP y responsable de crear el término “inteligencia artificial”, sugirió que llegaría el día en que la computación estaría organizada como un servicio público gracias a la tecnología de tiempo compartido de las computadoras [1]. Se sugirió que el sistema de tiempo compartido proporcionaría un ambiente operacional completo: editores de texto, aplicaciones informáticas, almacenamiento de archivos, impresión

masiva, entre otros. Lastimosamente, este concepto se desvaneció a mediados de los años 70's debido a que el hardware y el software de la época no estaban debidamente preparados para atender la demanda requerida [1].

En la actualidad, las comunicaciones han unido al mundo de tal forma que ya no hay necesidad de viajar largas distancias para realizar una reunión de negocios, basta tan solo con realizar una videoconferencia en la cual de manera coordinada se pueden tratar cada uno de los puntos de discusión de manera interactiva y sin la necesidad de que físicamente todos los participantes estén el mismo lugar. Avances tecnológicos como estos han permitido aumentar la productividad y dar igual número de oportunidades a las organizaciones sin importar su tamaño. Bajo esta premisa los gerentes de TI (Tecnologías de la información) han encontrado en Cloud Computing la solución a la gran problemática que afrontan las empresas de tecnología actualmente. El problema radica en que a medida que las empresas crecen significativamente, se vuelve imprescindible contar con equipos informáticos que puedan soportar tal crecimiento, manteniendo así la calidad del servicio ofrecido y previendo futuros inconvenientes. Esto trae consigo grandes

inversiones de dinero y además requiere de un periodo de tiempo para llevarse a cabo, tiempo en el cual la empresa podría estar planificando una nueva estrategia de ventas o diseñando un nuevo producto. Desde un punto de vista empresarial, realizar todo este proceso hace que se descuide el giro de negocios de la empresa y se invierta tiempo y esfuerzo en algo que si bien es cierto, va a mantener la calidad del servicio, no generará ninguna innovación ni hará que las ventas crezcan.

Cloud Computing es una tecnología que emerge principalmente porque se reconoce que los grandes centros de datos cuentan con miles de servidores que en general poseen un excedente de capacidad de cómputo, que en ciertos momentos no es utilizado, generando con esto un gasto innecesario del cual podría prescindirse si se concibe la idea de que exista un agente tercero que pueda proveer estos servicios en forma dinámica, escalable y bajo un esquema de pago por uso [2]. Este nuevo paradigma representa una gran ventaja para las pequeñas y medianas empresas que cuentan con una gran idea para emprender en Internet, pero no con el presupuesto necesario para adquirir y mantener todo un ambiente de TI que los respalde, así pues,

adquieren los servicios de un proveedor Cloud, que hace que la administración y escalabilidad se vuelvan transparentes para el usuario y el poder cambiar de un esquema a otro este solo a un click de distancia.

Una de las empresas pioneras en este nuevo paradigma fue *SalesForces.com* que en 1999 cambió de manera radical la forma en la cual los negocios adquirirían el software, permitiendo la distribución de aplicaciones empresariales en su página web [1]. En el 2002, *Amazon* aprovechó esta revolución al lanzar Amazon Web Services al mercado. *Google* hizo lo propio en el 2006 con Google Docs (ahora conocido como *Google Drive*). Mientras que *Microsoft* y *Apple* esperaron hasta el 2009 y 2011 con Windows Azure e iCloud respectivamente [1].

Los proveedores Cloud en algunas ocasiones no cumplen con los parámetros de calidad de servicio acordados con el cliente, ya sea por una sobrecarga de recursos, falta de procesos de gestión de incidentes, etc. Los sistemas de reputación ayudan a los clientes a escoger un proveedor Cloud que asegure la calidad de servicio esperada, incrementando así su utilidad y reduciendo al mismo

tiempo, la probabilidad de escoger un proveedor con antecedentes malos, el cual podría ocultarse bajo un precio atractivo.

Los sistemas de reputación han existido a lo largo de la historia, las personas los han usado a diario, incluso sin darse cuenta, debido a que la reputación ayuda a las personas a tomar mejores decisiones cuando la información disponible es escasa o carece de credibilidad [3]. La reputación es algo que usualmente no es tomado en consideración en los modelos tradicionales de negociación entre clientes y proveedores Cloud, simplemente se escoge la mejor oferta en función del precio.

Normalmente, los clientes y los proveedores Cloud tienen objetivos distintos, mientras los primeros buscan obtener el mejor rendimiento al menor costo posible, los otros desean incrementar sus utilidades empleando el menor número de recursos posibles, por lo que se genera un conflicto de intereses a nivel de oferta y demanda.

Debido a que las empresas buscan tomar decisiones que les permitan obtener una ventaja competitiva, muchas de sus estrategias están basadas en teorías económicas y modelos

matemáticos. Así, pueden entender de una mejor manera las situaciones que se presentan en un determinado instante, elaborar un modelo que lo represente, predecir los cambios que se pueden presentar a futuro, y tomar decisiones en base a ellos.

Una de las ramas de las matemáticas aplicadas, que estudia de manera abstracta y formal las decisiones óptimas que deben tomar diversos competidores en un conflicto, es la Teoría de Juegos. Pese a que la palabra juego tiene implicaciones lúdicas y se asocia con el azar, la teoría de juegos puede definirse como “el estudio de modelos matemáticos que describen el conflicto y la cooperación entre entes inteligentes que toman decisiones” [4].

1.3 JUSTIFICACIÓN

A nivel global, se está viviendo un cambio de paradigma en el mundo TI. La explosión de la información que trajo consigo Internet y la gran variedad de dispositivos móviles conectados a la red, permiten que las personas accedan a todo tipo de información de manera casi instantánea y mejoren su productividad en todos los aspectos. Esto ha producido que las empresas tengan que adaptar

sus servicios a las cambiantes necesidades de sus clientes, lo que conllevó al descubrimiento de muchas limitaciones en su infraestructura de información y manejo de datos.

Las aplicaciones han evolucionado, ahora es común llevar a cabo negocios en línea. Internet ya no es sólo un medio de comunicación para las personas, sino también una plataforma de trabajo que acelera el crecimiento de las empresas, por lo cual, contar con una plataforma que administre los procesos empresariales en línea ayudará a obtener mejores resultados a futuro.

La mayoría de los servidores de cómputo son grandes computadoras cuyo mantenimiento es costoso, y en algunas ocasiones no se ajustan del todo a las necesidades de procesamiento que los clientes requieren a lo largo del tiempo. Por ejemplo, una plataforma de banca virtual recibe habitualmente un número promedio de usuarios al día, que realizan diferentes transacciones. Suponiendo que existen momentos en los cuales los usuarios ingresan con mayor frecuencia a la plataforma para consultar su saldo actualizado, se esperaría que este comportamiento no afecte en nada al rendimiento de la plataforma y

el funcionamiento continúe sin inconvenientes, El no poder soportar la gran concurrencia de usuarios conllevaría a congestionamientos que se reflejarían en largos tiempos de respuesta y disminuirán considerablemente la experiencia de usuario. Invertir en un datacenter, que soporte grandes cargas de trabajo únicamente durante periodos cortos de tiempo, es algo que no es viable para todas las empresas, ya que durante los periodos restantes, el poder de procesamiento seguiría disponible aun cuando no representa una necesidad. Evidentemente, si se desea mantener el nivel de satisfacción de los usuarios, los ingenieros de TI deben encontrar una solución que involucre escalabilidad dinámica de recursos en tiempo real y bajo demanda.

Cloud Computing plantea una solución a este problema, haciendo la escalabilidad transparente para el usuario distribuyéndola en forma de servicio, dejando así que el trabajo pesado sea algo de lo cual ya no haya que preocuparse y se concentre en los verdaderos problemas de negocio.

Según un estudio llevado en el 2011 por AMD (Advanced Micro Devices) a nivel global, sobre “sobre la adopción, actitudes y

estrategias en *Cloud Computing*", el 70% de los encuestados indicaron que se encuentran investigando o haciendo uso de esta tecnología, ya sea para almacenar datos o alojar aplicaciones en la nube. El 60% de los encuestados, que han implementado soluciones de Cloud Computing, admitieron que han encontrado una clara ventaja de negocio frente al modelo tradicional de computación [5].

En el futuro, la tendencia sugiere que la mayoría de infraestructuras de TI empezaran a migrar a la nube. Por ello, las empresas necesitan diseñar e implementar una variedad de servicios que garanticen el uso óptimo de los recursos involucrados. Comúnmente, los mecanismos de negociación que se dan entre proveedores y clientes, se centran en una variable, el precio. Un cliente desea cierta cantidad de procesamiento y diversos proveedores ofrecen brindarlo fijando un precio. El inconveniente del seleccionar el menor de los precios ofertados se presenta debido a que no existe una forma de cuantificar el riesgo que se adquiere al momento de contratar el servicio, ni tampoco se establecen acuerdos mediante los cuales se garantizaran los niveles acordados de servicio.

El cliente siempre buscará que sus trabajos sean terminados en la menor cantidad de tiempo posible para minimizar sus costos, mientras que el proveedor querrá maximizar la utilización de sus recursos para incrementar sus utilidades, lo cual podría ir en contra de los intereses de los clientes. Este comportamiento en particular se puede representar, bajo un modelo basado en *Teoría de Juegos*, como un *juego no cooperativo* en el cual cada uno de los proveedores dan a conocer sus ofertas en orden de menor a mayor reputación para que así, siempre el proveedor que tenga mayor reputación pueda ver cuáles han sido los precios que han sido ofrecidos al cliente antes de ofertar su precio final, de esta manera podría ajustar su margen de ganancia para ofrecer el mejor precio, ganar el juego y el usuario obtendría un precio más conveniente. La reputación de cada uno de los proveedores estará basada en información histórica que determinará la probabilidad de que el servicio se ejecute sin inconvenientes.

CAPÍTULO 2

2 DEFINICIONES Y TRABAJOS RELACIONADOS

2.1 CLOUD COMPUTING

De acuerdo con [2] Cloud Computing se define como “un modelo que permite, el acceso ubicuo de manera conveniente y bajo demanda, a una red compartida de recursos computacionales configurables”. Cloud Computing, ayuda a reducir los gastos en activos fijos que incurrirían las empresas en la construcción de un datacenter propio, ofreciendo poder computacional bajo demanda y en modalidades de pago por uso. Este esquema es ideal para las startups ya que facilita el proceso de arrancar el negocio, y en caso de que éste no resulte, reduce considerablemente el impacto económico que esto traería consigo.

Según [6], las razones más importantes por las cuales las empresas deberían migrar a la nube se pueden resumir en la tabla 1:

Tabla 1 Razones para migrar a la nube

Escalabilidad	Cloud Computing es altamente escalable, los recursos pueden ser añadidos y removidos a la red sin necesidad de grandes inversiones.
Seguridad	Los proveedores Cloud que están el cumplimiento con SOC2 Y SSAE16 deben cumplir con ciertos requisitos rigurosos para vigilancia y seguridad.
Libertad	Con un proveedor Cloud corporativo debidamente organizado, ya no hay necesidad de preocuparse por el hardware, seguridad de la información, respaldos, recuperación de desastres, actualización de sistemas operativos, etc.
Costo	Los gastos de capital de TI migrarán a gastos de funcionamiento, en vez de una enorme ola de gastos de capital cada 3 o 5 años para actualizaciones de hardware, los pagos son mensuales, lo que conlleva a tener un presupuesto más previsible y un mejor flujo de caja.
Continuidad de negocio	Si se hace apropiadamente el proceso de migración, Cloud Computing ofrece redundancia completa de conmutación por error, se ocupa de las copias de seguridad y recuperación de desastres.

Movilidad	Con una solución adecuada de Cloud Computing, el escritorio corporativo puede ser utilizado por los usuarios a través de cualquier dispositivo que pueda conectarse a Internet
-----------	--

2.1.1 CARACTERÍSTICAS DE CLOUD COMPUTING

El instituto NIST (*National Institute of Standards and Technology*) especifica cinco características esenciales que una infraestructura de *Cloud Computing* debe tener para ser considerada como tal [2]:

- **Gestionada por el Proveedor**

Los servicios de computación en la nube son gestionados directamente por el proveedor Cloud. Una vez que las aplicaciones y los servicios se han trasladado a la nube, las empresas ya no tienen que preocuparse por los problemas que los centros de datos locales puedan tener con respecto a gastos de energía, el espacio físico ni la refrigeración; mientras que los desarrolladores sólo necesitan saber si sus aplicaciones se ejecutan correctamente, sin tener en cuenta donde se ubicarán los servicios o recursos de su software.

- **Asignación flexible de recursos**

El número de recursos disponibles en *Cloud Computing* pueden ser aumentados o disminuidos, ajustando sus costos en función del consumo real. Esto permite que una organización comience a satisfacer la demanda de procesamiento en los altos “picos” de trabajo, pagando sólo por el uso necesario. Con esto, se evita la problemática de los *datacenters* tradicionales donde se adquieren recursos “sobrantes” de antemano como planificación a algún tipo de crecimiento que pueda acaparar rápidamente la capacidad de la red.

- **Accesible desde la Red**

Los servicios en la nube están disponibles a través de cualquier dispositivo y tecnología que pueda conectarse a Internet, lo que facilita su rápido acceso para los clientes móviles y oficinas remotas. Esto proporciona un nuevo modelo “cualquier lugar, cualquier momento” que no es posible en *datacenters* tradicionales. Debido a que los proveedores *Cloud* pueden estar ubicados en cualquier parte del mundo, estos alojan los servicios de las organizaciones en áreas lejos de complicaciones

ambientales y conflictos geopolíticos. Por ejemplo, antes de la llegada de un desastre natural (como un huracán o un tifón), un proveedor *Cloud* podría transferir sus operaciones desde EEUU a Europa de manera transparente para el consumidor del servicio, llegando este a ni siquiera tener el conocimiento de lo ocurrido.

- **Sostenible y Elástico**

Debido a que los proveedores *Cloud* pueden suministrar recursos según la necesidad, las empresas reducen las obligaciones por refrigeración y uso de energía de los *datacenter*. La flexibilidad en el alojamiento físico de los servidores en Cloud Computing permite a los proveedores *Cloud* cambiar la locación del centro de operaciones sin interrumpir los recursos de los consumidores. Pueden mover toda la actividad del centro de datos a países fríos durante los meses de verano para ahorrar en costos de refrigeración, o pueden trabajar en países donde la producción de energía eléctrica es más amigable para el medio ambiente y más económica en cuestiones monetarias.

- **Gestión a través del Autoservicio a demanda**

Los límites de disponibilidad de recursos se configuran dentro de los sistemas por parte del proveedor, haciendo este tópico transparente para el usuario. El aumento de la capacidad del servicio puede ser definido de manera automática o gestionado por el usuario Cloud sin mucho esfuerzo. Si una empresa desea levantar un servidor de prueba, ya no requiere el acceso al sistema físico, cargar el software o configurar la red manualmente, por el contrario, el cliente sólo necesita acceder a la plataforma de su proveedor *Cloud* y solicitar una nueva asignación de recursos mediante una interfaz de autoservicio, sin necesidad de asistencia técnica.

2.1.2 TIPOS DE ESQUEMAS CLOUD, SEGÚN EL MODELO DE IMPLEMENTACIÓN

Adicionalmente a definir *Cloud Computing* según sus características esenciales, *NIST* propuso también cuatro modelos de implementación [2]. Esto responde a que no todos los proveedores *Cloud*, y los servicios que ofrecen, son iguales. Dichos modelos se pueden clasificar en:

- **Nube Pública**

Es una infraestructura *Cloud* cuyo propietario es un proveedor que proporciona sus recursos de manera abierta al público, sin tener algún tipo de relación entre sí.

- **Nube Privada**

Opuesto al modelo anterior, es una infraestructura diseñada e implementada para una sola empresa que realiza una administración directa de sus servicios en la nube, sin que participen entidades externas. Esto logra que se mantenga un total control sobre sus recursos. No necesariamente esto es implementado por la misma organización que lo utiliza, sino que contratan a proveedores *Cloud* que actuarán bajo supervisión. Las entidades que se inclinan por este modelo suelen ser complejas y, a su vez, necesitan centralizar todos sus recursos mientras ofrecen una gran flexibilidad de los mismos.

- **Nube Comunitaria**

En este modelo, muchas organizaciones tienen un propósito en común, por lo que implementan una

infraestructura compartida en la nube. Un claro ejemplo de esto es **OpenCirrus**, que es un banco de pruebas de experimentación sobre *Cloud Computing* que puede ser usado por universidades e instituciones de investigación [7].

- **Nube Híbrida**

Finalmente, las empresas pueden optar por combinaciones de las diferentes implementaciones. Podrían optar por centralizar su información que es crítica y sensible para sus operaciones de negocios en una nube privada, manteniendo el control de la misma; mientras que, pueden adquirir más recursos de nubes públicas para desarrollar una mayor capacidad de respuesta a demandas de trabajo imprevistas y picos de carga.

2.1.3 TIPOS DE ESQUEMA CLOUD COMPUTING, SEGÚN EL MODELO DE SERVICIO

Debido a que existen diversos proveedores *Cloud* con diferentes productos de cómputo en la nube, NSIT también estableció una clasificación de *Cloud Computing* dependiendo de los servicios ofrecidos, que consiste en tres

categorías primarias asociados con el término en inglés “*as a Service*” (*aaS*), de acorde a su nivel de abstracción [2]. Esta clasificación, se la conoce como Cloud Computing Ontology, la cual define dependiendo de la existencia o ausencia de uno o más servicios de cómputo. Cada nivel superior de esta categorización, incluye las características de los niveles inferiores [2].

Existen proveedores que ofrecen una infraestructura de su centro de cómputo para hospedar aplicaciones y datos. En otra categoría, se puede encontrar adicionalmente las herramientas para poder desarrollar aplicaciones que puedan ser hospedadas y de esta forma aprovechar de manera eficiente esta infraestructura. Finalmente, se hallan aplicaciones listas para ser utilizadas y la posibilidad de almacenamiento para los datos.

Como se representa en la figura 2.1, los proveedores comienzan con el nivel más fundamental llamado *Infraestructura como Servicio (IaaS)*, que incluye elementos de entornos de computación tradicionales como las redes,

almacenamiento y otros componentes arquitectónicos que son de interés para los administradores de sistemas, los ingenieros de TI y los gerentes de la empresa. Mientras que, los desarrolladores de software aprovechan los servicios prestados por los proveedores del nivel *Plataforma como Servicio (PaaS)*, que también soporta la infraestructura de alojamiento. Los usuarios consumen directamente las aplicaciones proporcionadas por el nivel de *Software como Servicio (SaaS)*, que a su vez incluye componentes tanto de la plataforma como de la infraestructura.



Figura 2.1 Modelo de servicios de Cloud Computing junto al cliente objetivo por nivel [8]

2.1.4 INFRAESTRUCTURA COMO SERVICIO (IaaS)

Infraestructura como Servicio (*IaaS*) es llamada también hardware como servicio (*HaaS*) para expresar su finalidad en

el suministro de recursos de hardware sobre demanda. *IaaS* puede definirse como el servicio tradicional de hospedaje de un datacenter. Los proveedores rentan tiempo de cómputo y espacio de almacenamiento, permitiéndole al cliente manejar la carga de trabajo y tener un control casi total sobre las aplicaciones y recursos fundamentales que soportan a la organización.

Una de las características más importantes y que forman parte del núcleo de Cloud Computing es la virtualización, que se define como “dividir o particionar los recursos de un servidor en múltiples máquinas virtuales aisladas” [8], permite compartir las características de hardware de una manera más fácil. Cada equipo físico se divide en diferentes e independientes servidores virtuales, cada uno con su propio sistema operativo, bases de datos, manejo de archivos y conexión a la red. Como ejemplos de proveedores de infraestructura se puede nombrar a *Amazon Web Services*, *Google Compute Engine*, *Windows Azure*, *Flexiscale*, *GoGrid*, entre otros.

2.1.5 PLATAFORMA COMO SERVICIO (PaaS)

Proporciona un entorno de trabajo sobre el cual se pueden desarrollar aplicaciones para Cloud. Incluye un conjunto de herramientas de programación, lenguajes, librerías e interfaces de programación de aplicaciones, que amplían la capacidad de una organización para desarrollar dichas aplicaciones. Es descrito frecuentemente como CloudWare o como una especie de sistema operativo para Cloud Computing. Los proveedores suelen ofrecer además: almacenamiento de datos, caché, seguridad y monitoreo.

Algunos proveedores de *PaaS* han creado sus propios lenguajes de programación, muchas veces análogos y similares a los ya existentes para disminuir la curva de aprendizaje de los programadores e incentivar el desarrollo de aplicaciones para su plataforma. *Salesforce.com* es ejemplo de esto con los lenguajes Apex (similar a *Java*) y *Visualforce* (similar a *XML*). Los proveedores con servicios más robustos, utilizan lenguajes estandarizados y de código abierto para facilitar la adopción de la plataforma y simplificar la migración de aplicaciones existentes en una organización.

Otros ejemplos de *PaaS* son Microsoft Windows Azure, Amazon EC2, Google App Engine, y Force.com.

2.1.6 SOFTWARE COMO SERVICIO (SaaS)

Permite que una aplicación esté disponible en Internet, frecuentemente mediante la suscripción de pago por el uso del servicio. Los clientes *SaaS* utilizan las aplicaciones existentes y los datos de las mismas, todo sobre Internet, sin que busquen desarrollar aplicaciones ni que deseen encontrar un ambiente para alojarlas. Los usuarios utilizan los programas directamente de la nube, a través de un navegador web o un cliente ligero, en lugar de tener su propia infraestructura para instalarlas y ejecutarlas.

Al igual que en los productos de software tradicionales, las alternativas *SaaS* están prediseñadas y no se pueden modificar más allá de la configuración de personalización que se le permite al consumidor. El proveedor es quien se encarga de la infraestructura de hardware y software, así como del trabajo de la instalación, configuración, licenciamiento, actualización y mantenimiento de las

aplicaciones. Entre los proveedores de software como servicio, se resaltan *Microsoft Office 365*, *SalesForce.com*, *LotusLive*, *Google apps*, entre otros.

2.2 SISTEMAS DE REPUTACIÓN

Los sistemas de reputación, como sugiere [9], ayudan a construir confianza en las comunidades y generalmente son usados cuando los usuarios que interactúan realizando transacciones, no tienen conocimiento uno del otro, así pues basándose en conocimiento previo forman una opinión que podría ayudar a otros usuarios a tomar una decisión.

La reputación, como define [9] es “la información recolectada y procesada sobre el comportamiento de una entidad según la experiencia de los demás”, los sistemas de reputación precisamente van llevando registros históricos sobre el comportamiento de las entidades involucradas, para luego procesarlos y transformarlos en información útil para el usuario. Acorde con [9] la arquitectura de un sistema de reputación está constituida por varios componentes y actores, tal como se muestra en la figura 2.2.

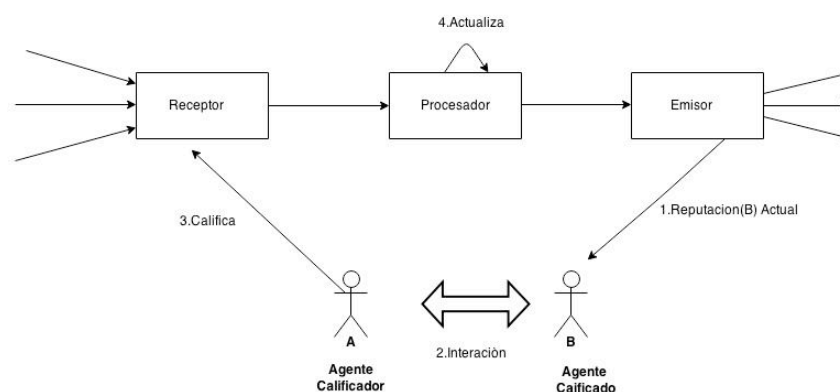


Figura 2.2 Arquitectura de un Sistema de Reputación [9]

Como se observa en la figura 2.2 existen 2 agentes que interactúan entre sí. El agente encargado de emitir una calificación es denominado agente calificador, mientras que el que recibirá la calificación es denominado agente calificado. El receptor por otra parte se encarga de recibir las calificaciones de los agentes calificadores, procesarlas y añadirlas, para que finalmente el emisor publique los resultados y estos estén disponibles para los demás agentes.

2.2.1 MÉTRICAS EN SISTEMA DE REPUTACIÓN

Un sistema de reputación puede ser descrito por sus métricas, que representan la forma en la cual se comparan diferentes agentes para medir su reputación. En [9] se clasifica a los sistemas de reputación según sus métricas de la siguiente manera:

- **Sistemas Acumulativos**

El sistema recibe y acumula todas las calificaciones en orden para calcular la reputación global de un agente.

- **Sistemas Promedios**

El sistema calcula la reputación de un agente como el promedio de todas las calificaciones que éste ha obtenido, lo cual se fundamenta en el hecho de que un agente se comportará de la misma manera la mayor parte del tiempo.

- **Sistemas Blurred**

El sistema calcula la reputación de un agente como la suma de medias ponderadas de todas las calificaciones, con la diferencia de que las nuevas calificaciones tienen más influencia para la reputación actual, esto se basa en

el hecho de que los agentes pueden cambiar su comportamiento durante su existencia.

- **Sistemas Only Last**

El sistema calcula la reputación de un agente a partir de la última calificación que éste ha obtenido, fundamentado en el hecho de que un agente se comporte de la misma forma que la última vez.

- **Sistemas Eigen Trust**

El sistema combina los valores de reputación local de un agente de manera iterativa para obtener una reputación global, la reputación local se define por un conjunto de agentes que pueden tener diferentes reputaciones en cada sistema de reputación local, pero solo un único valor en un sistema de reputación global.

- **Sistemas Adaptativos**

El sistema se comporta de manera distinta en función de la reputación actual del agente; por ejemplo, si la reputación de un agente es baja, esto hace que una calificación positiva cause un alto incremento en la reputación, mientras que si la reputación es alta, la misma

calificación produce únicamente un incremento pequeño en la reputación.

- **Sistemas Beta**

El sistema trata de predecir el comportamiento futuro de un agente usando estadística. Esta predicción toma en consideración el comportamiento pasado de un agente y la probabilidad de comportarse bien o mal en la siguiente transacción.

2.3 ECONOMÍA EXPERIMENTAL

Según Vernon Smith, premio Nobel de Economía en 2002, “la economía experimental aplica métodos de laboratorio para estudiar las interacciones de los seres humanos en los contextos sociales gobernados por reglas explícitas o implícitas” [10]. En el presente trabajo las reglas explícitas representarán las variables que se pueden controlar en el modelo, y las implícitas son las que se escapan del control como por ejemplo, los criterios de selección de un proveedor de servicios Cloud considerando únicamente o en mayor grado su reputación sin importar el precio, preferencias personales, etc.

La economía experimental resulta muy útil cuando se desea hacer predicciones sobre los comportamientos o patrones que pueden tener los individuos cuando están en un entorno particular sujeto a ciertas condiciones establecidas inicialmente. Un ejemplo clásico que aportó al desarrollo de la economía experimental es un tipo de juego estático¹ llamado: El dilema del prisionero. En el juego existen 2 sospechosos de haber cometido un crimen y existe suficiente evidencia para que los 2 se queden en la cárcel por un largo tiempo a no ser que uno de ellos confiese. Para el experimento se separó a los reos en celdas diferentes y se les explicó las consecuencias de las acciones que podían tomar, la tabla 2 muestra la representación del juego en forma normal [11].

Tabla 2 Dilema del prisionero en forma normal

Acciones	Confesar	No confesar
Confesar	-6,-6	0,-9
No Confesar	-9,0	-1,-1

1 Juego en el cual los jugadores realizan sus movimientos de manera aislada y sin conocer lo que los otros jugadores han hecho.

2.3.1 TEORÍA DE JUEGOS

La teoría de juegos estudia “como los individuos racionales toman decisiones cuando estos son mutuamente interdependientes” [11], es decir que las acciones que tomen repercutirán en los demás individuos. En [11] también se menciona que para poder comprender mejor esta definición se requiere tener en claro tres conceptos claves, que son: individualismo, racionalidad e interdependencia mutua.

El individualismo se refiere al hecho de que los participantes de un juego siempre buscarán su beneficio personal como principal objetivo sin importar lo que suceda. La racionalidad como define [11] asume que los individuos siempre actuarán a favor de sus intereses personales, y no tomarán decisiones que vayan en contra de los mismos. Finalmente, la interdependencia mutua significa que en cualquier momento las acciones de un individuo en un juego, pueden estar por lo menos parcialmente determinadas o condicionadas por las acciones de otros jugadores.

Acorde con [11] los juegos pueden ser cooperativos y no cooperativos. Como se muestra en la figura 2.3, en un juego no cooperativo cada individuo actúa de manera aislada para buscar el máximo beneficio, lo cual no significa que no existirán acuerdos entre jugadores, esto se dará solamente si favorece los intereses personales de los individuos.



Figura 2.3 Representación de un juego no cooperativo

Por otra parte, en los juegos cooperativos pueden existir acuerdos entre los individuos, pero eso no conlleva a que dejen de buscar su beneficio personal, actuarán de manera cooperativa siempre y cuando esto beneficie sus intereses. (Ver figura 2.4)

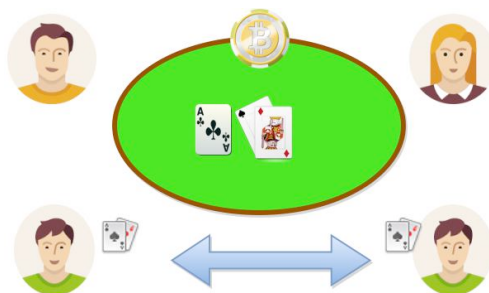


Figura 2.4 Representación de un juego cooperativo

2.3.2 REPRESENTACIÓN DE JUEGOS

De acuerdo a [11], en la teoría de juegos no cooperativa existen 2 formas en las cuales un juego puede representarse, la primera forma es la llamada forma normal, y la segunda es la forma extensiva. Existen ocasiones en la cual una representación ayuda a entender mejor el juego que otra, pero en general ambas son válidas.

- **Juegos de Forma Normal**

Acorde a [11], en un juego de forma normal se deben poder identificar de manera clara 3 componentes:

1. Jugadores
2. Estrategias

3. Pagos

Los jugadores son individuos que tomarán decisiones racionales buscando siempre su beneficio en el juego, un juego de forma normal requiere un mínimo de 2 jugadores para que exista interdependencia.

Adicionalmente [11], define estrategia como “una descripción completa de como un jugador podría jugar un juego”; es decir, una estrategia representa la forma o el conjunto de acciones que un jugador realizará para lograr su objetivo final; por ejemplo, en un juego de *póker* las acciones que un jugador puede realizar en cada turno son pagar la apuesta, aumentar la apuesta, pasar el turno (en caso de que no haya habido apuestas anteriores) o retirarse, este conjunto de acciones no representan estrategias, el cuándo hacerlo y el orden en el cual se hagan sí.

Los pagos representan lo que un jugador recibirá al final del juego, teniendo en consideración las acciones de los demás jugadores. Los pagos no necesariamente deben

ser cantidades monetarias, pueden definirse en diferentes formas como por ejemplo en el dilema del prisionero [11] los pagos son los años que los reos estarán en prisión. La tabla 3 muestra la representación en forma normal de un juego con 2 posibles acciones.

Tabla 3 Representación General de un Juego con 2 acciones y 2 jugadores en forma Normal [11]

		Jugador 1	
		Acción 1	Acción 2
Jugador 2	Acción 1	pago 1	pago 2
	Acción 2	pago 3	pago 4

- **Juegos de Forma Extensiva**

En una representación extensiva se destacan 2 aspectos importantes que no son considerados en una representación normal: el tiempo en el cual los jugadores toman las decisiones y la información que dispone cada jugador cuando debe tomar una decisión.

En [11] se definen los elementos que componen una representación extensiva de un juego, las cuales se muestran en la tabla 4.

Tabla 4 Elementos de un juego en representación extensiva

Elementos de un juego en representación extensiva	
Nodos	Posición en el juego en el cual un jugador debe tomar una decisión
Ramas	Representan las diferentes alternativas que pueden escoger los jugadores
Vectores	Representan los pagos para jugador listados en orden de los jugadores respectivamente, cuando un vector es alcanzado el juego habrá terminado
Sets de información	Ocurre cuando dos o más nodos se juntan con una línea entrecortada representando que el jugador no sabe en qué posición se encuentra cuando va a tomar la decisión

La figura 2.5, muestra la representación del juego del dilema del prisionero [11] en forma extensiva:

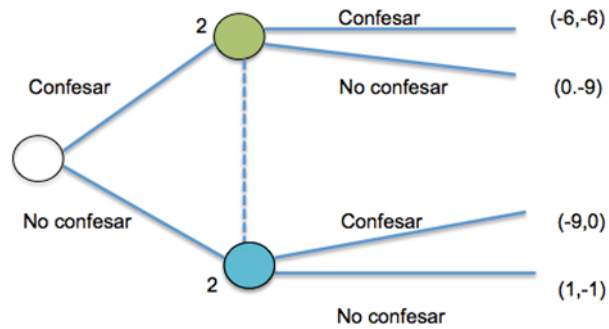


Figura 2.5 Dilema del prisionero en representación extensiva [11]

2.4 SERVICE LEVEL AGREEMENT (SLA)

Un acuerdo de nivel de servicio (*SLA*), tal como sugiere [12], es un “documento que describe el nivel de servicio esperado por un cliente de un proveedor, definiendo las métricas en las que el servicio se medirá, y las soluciones o sanciones, en caso de que estos niveles no se puedan lograr”.

El *SLA* cubre por lo general, una lista de los servicios que se entregarán, informes de seguimiento y rendimiento de recursos, la gestión de estrategias contra fallos, los deberes y responsabilidades de los clientes, la manera como se manejará la información confidencial, el tiempo de vigencia del contrato y los términos para el finiquito del mismo.

Dependiendo de la jerarquía de servicio (*SaaS*, *PaaS*, *IaaS*) a la que pertenezca, el proveedor deberá definir un "nivel objetivo de servicio", o a su vez un "nivel mínimo de servicio" que especifique las garantías del sistema, velocidad de conectividad, la disponibilidad, mantenimiento, funcionamiento y operación de los recursos; aceptando las diferentes penalizaciones a las que se debe someter en caso de incumplimiento. La figura 2.6 ilustra gráficamente lo descrito anteriormente:

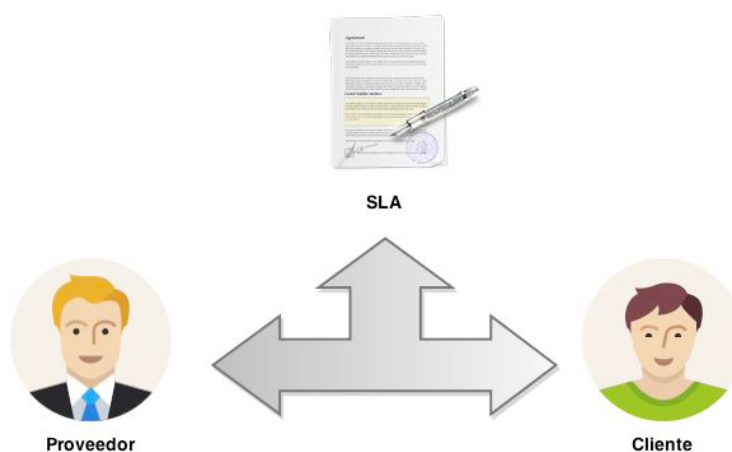


Figura 2.6 Representación de un SLA

2.5 SOFTWARE DE SIMULACIÓN CLOUDSIM

Si bien *Cloud Computing* permite tomar considerables ventajas, como el ahorro de costos de infraestructura, la movilidad de sus servicios, la abstracción de estrategias de mantenimiento, la

decisión por parte de una empresa para trasladar todos sus productos hacia la nube es bastante delicada. El poco control de recursos, obliga a depender fuertemente del proveedor *Cloud*. Muchas organizaciones pueden sentirse arriesgadas por la imposibilidad de manejar directamente su privacidad y por no estar seguros sobre la confidencialidad de su información. La preocupación de los gerentes de TI, en cambio, radica más en escoger al proveedor de servicios más idóneo y que vaya a satisfacer adecuadamente las necesidades del negocio y que tenga más garantías de cumplimiento.

Es entonces cuando realizar una simulación de las diferentes situaciones que pueden presentarse en la nube, se vuelve imperioso. La posibilidad de experimentar con variadas exigencias de carga de trabajo en un entorno simulado y controlable, permitirá realizar pruebas de eficiencia y rendimiento que ayudaran a tomar decisiones antes de llevar a cabo la implementación en un ambiente real.

CloudSim [13] es una herramienta de código abierto implementada en Java que permite simular diferentes escenarios que ocurren en

la nube, a través de la simulación del hardware y software de un proveedor Cloud, el modelamiento de centros de datos, establecimiento de políticas de asignación de recursos, administración de máquinas virtuales y control de ancho de banda, son unas de las tantas configuraciones que pueden simularse, con la ventaja de que las simulaciones se pueden volver a replicar de manera sencilla, así, los ingenieros TI pueden realizar pruebas robustas de la futura implementación sin que esto represente un costo adicional y controlando todas las variables necesarias. [13]. Entre las características que *CloudSim* ofrece, se pueden encontrar clases para el modelamiento de infraestructuras de centro de datos, administración de máquinas virtuales, políticas de asignación de recursos, etc. Esto hace que sea más fácil realizar un análisis cruzado de costo y beneficio tanto para cliente que busca la mejor oferta como para el proveedor que desea aumentar sus utilidades.

2.5.1 ARQUITECTURA CLOUDSIM

En la figura 2.7, se muestra la arquitectura de CloudSim [13] y sus componentes. En versiones iniciales de Cloudsim la capa de *SimJava* era usada como motor de simulación de eventos, la cual soportaba muchas funcionalidades

esenciales para creación de entidades Cloud tales como Servicios, Hosts, Datacenters, Brokers, máquinas virtuales, comunicación entre componentes, simulación del reloj, etc. Sin embargo, en la versión utilizada en este trabajo, esta capa ha sido removida para poder permitir operaciones avanzadas que no eran soportadas anteriormente.

La capa de simulación de CloudSim provee soporte para modelar y simular entornos de Datacenters basados en Cloud Computing, incluyendo interfaces de administración dedicadas para máquinas virtuales, almacenamiento y ancho de banda, esto podría ser de gran utilidad para los Proveedores Cloud que deseen probar nuevas estrategias de asignación de recursos y decidir así hasta qué punto son viables de implementar en un ambiente real.

Finalmente, la capa superior de *CloudSim* es la del código de usuario que permite a los desarrolladores extender de las entidades principales del simulador para poder generar diferentes cargas de trabajo, configuración de las aplicaciones, modelar la disponibilidad de recursos, y realizar

pruebas robustas basadas en configuraciones personalizadas.

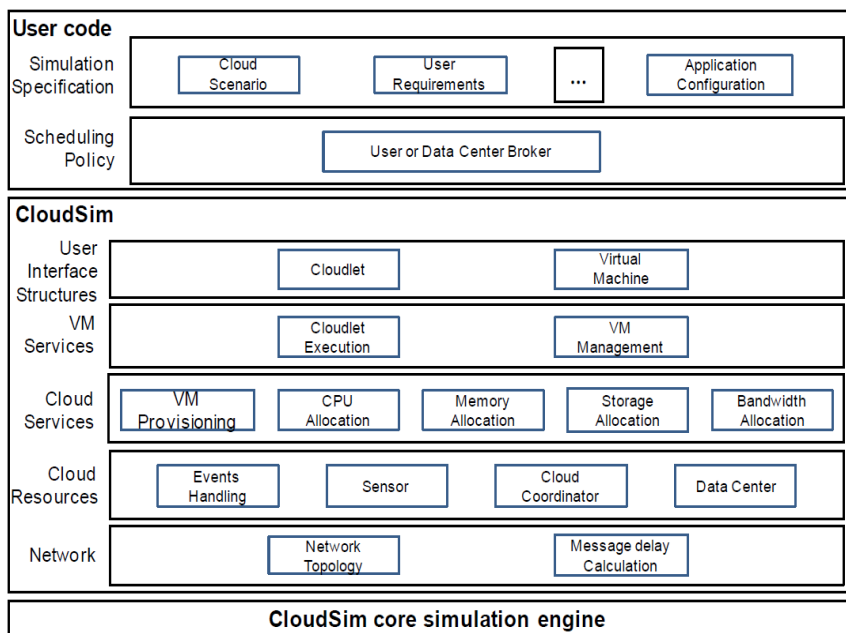


Figura 2.7 Arquitectura de Cloudsim [13]

2.6 TRABAJOS REALIZADOS

Los siguientes estudios han servido de referencia para el diseño y modelo del simulador Cloud.

[14] Realiza un simulador Cloud en el cual se gestionan los entornos de la simulación y se realizan pruebas en escenarios Cloud, permitiendo recoger resultados por cada prueba y dándole al

usuario la facilidad de realizar los análisis de los resultados para así proceder a tomar una conclusión. Al realizar la implementación de un simulador Cloud se puede considerar el uso de parámetros o métodos basados en de teoría de juego como lo propone [15], donde se realizó una adecuada asignación de recursos que servirá para un servicio Cloud .Por otra parte, la confianza es uno de los factores más importantes. En [16] se propone un modelo cross-doup que se basa en investigaciones anteriores en el cual cada cliente almacena los datos de una tabla, llamada tabla de confianza, que servirá para la recomendación de confianza en los sistemas para mejorar la seguridad de tomar la decisión. Por otra parte [17] propone un mecanismo para la asignación dinámica de recursos en la nube aplicando una combinación de conceptos entre sistemas de mercados y sistemas de reputación en el cual se obtienen resultados de un simulador desarrollado en Cloudsim.

CAPÍTULO 3

3 DISEÑO DEL MODELO

3.1 DISEÑO DE LA EXPERIMENTACIÓN

La experimentación tiene como objetivo diseñar un modelo basado en teoría de juegos, el cual haciendo uso de un sistema de reputación, ayude a mejorar la asignación de recursos en Cloud Computing y reduzca las violaciones SLA por parte de los proveedores de Servicio.

3.1.1 DESCRIPCIÓN DEL MODELO

El modelo consiste en un juego no cooperativo en el cual los proveedores Cloud compiten por la asignación de recursos y envían sus ofertas en orden de menor a mayor reputación

hacia un agente intermediario, el cual usará un sistema de reputación Beta [9] que predecirá el futuro comportamiento del proveedor en la siguiente transacción, para que así el precio no sea el único factor que haga que éste sea escogido. Los actores y componentes que han sido considerados para el modelo se muestran a continuación:

- **CRAG:** Cloud Resource Allocation Game es un juego no cooperativo en el cual los proveedores Cloud compiten por la asignación de recursos, a partir de un trabajo enviado por el Usuario Cloud.
- **Usuario Cloud:** El Usuario Cloud envía un trabajo al CRAG, en el cual especifica requerimientos específicos de recursos, el precio máximo que está dispuesto a pagar, y los parámetros de nivel de servicio en un SLA.
- **Trabajo:** Representa una tarea que será enviada al juego, y está definida en millones de instrucciones por segundo requeridas para completar la misma.

- **SLA:** Componente que contiene especificaciones formales sobre los parámetros de nivel de servicio tales como: disponibilidad, confiabilidad, etc.
- **Proveedor Cloud:** Es el dueño de los recursos Cloud y compite por la asignación de recursos en su infraestructura.
- **Agente Intermediario:** Es el intermediario entre el Usuario Cloud y el Proveedor Cloud, el cual se encarga de encontrar el precio más económico con el menor riesgo de violaciones SLA para el Usuario Cloud, y recibe una comisión por parte del Proveedor por cada trabajo ejecutado exitosamente.
- **Recurso Cloud:** Son las infraestructuras de hardware que poseen cada uno de los proveedores Cloud, en este modelo los recursos serán los hosts que están dentro de un datacenter y poseen máquinas virtuales con una capacidad específica de computo medida en millones de instrucciones por segundo.

- **Mecanismo de Reputación Beta:** Recopila las calificaciones que ha recibido un Proveedor Cloud para predecir de manera estadística el comportamiento del mismo en la siguiente transacción.

3.1.2 MECANISMO DE REPUTACIÓN BETA

El mecanismo de reputación Beta, como menciona [9], está basado en la función de densidad de probabilidad Beta, que puede ser usada para representar eventos binarios.

La distribución de probabilidad beta $f(p|\alpha, \beta)$ puede ser expresada mediante la ecuación 3.1 usando la función gamma de la siguiente manera:

$$f(p|\alpha, \beta) = \frac{\Gamma(\alpha + \beta)}{\Gamma(\alpha) \Gamma(\beta)} p^{\alpha-1} (1 - p)^{\beta-1} \quad (3.1)$$

$$; 0 \leq p \leq 1, \alpha > 0, \beta > 0$$

La ecuación 3.1 tiene las siguientes restricciones:

- $p \neq 0$ si $\alpha < 1$
- $p \neq 1$ si $\beta < 1$

El valor esperado de la distribución Beta está dado por la ecuación 3.2:

$$E(p) = \alpha / (\alpha + \beta) \quad (3.2)$$

En el modelo de experimentación α y β representan la cantidad de trabajos realizados con éxito, y la cantidad de trabajos fallidos respectivamente por parte de un proveedor cloud. Estos valores servirán para poder calcular el valor esperado, que representa la probabilidad de que un proveedor ejecute un trabajo con éxito.

3.2 VARIABLES MANIPULADAS

Las variables manipuladas, son las que se modifican o manipulan para provocar cambios en las variables controladas, con el objetivo de comprobar la validez de una hipótesis. En este modelo de experimentación, la variable manipulada será la importancia de reputación, la cual podrá tomar valores en una escala del 1 al 10, el valor más bajo significa que no se tomará en cuenta la reputación de un proveedor cloud al momento de competir por la asignación de recursos, y a medida que este valor va incrementando, la reputación se vuelve un factor determinante en la competencia, y el

precio ya no es el único criterio para escoger a un ganador. La figura 3.1 muestra como un usuario Cloud puede ajustar la importancia de reputación para la simulación.

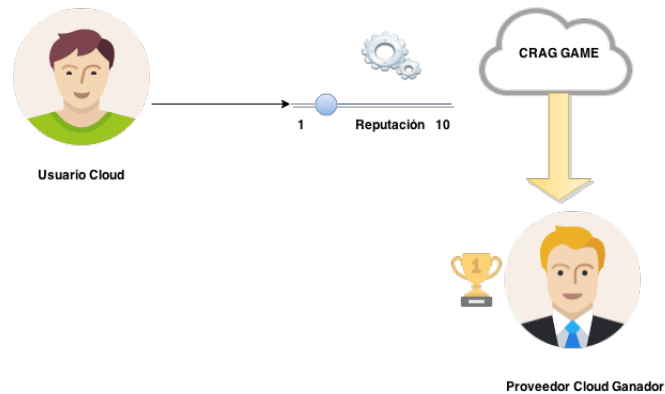


Figura 3.1 Variables Manipuladas CRAG

3.3 VARIABLES CONTROLADAS

Las variables controladas son de vital importancia, ya que son las que controlarán que el cambio de las variables dependientes sea a causa de la manipulación de las variables independientes o manipuladas y no debido a factores externos que no hayan sido tomados en consideración. A continuación se listan las variables controladas del modelo de experimentación y se detalla cómo se efectuará el control:

- **Precio:** El precio que define cada Proveedor Cloud siempre será mayor que su costo de producción.
- **Disponibilidad:** La disponibilidad de recursos en el juego es suficiente para satisfacer cualquier trabajo requerido por el usuario.
- **Cooperación:** No existirá cooperación entre Proveedores Cloud de ningún tipo.

3.4 DESCRIPCIÓN DEL ENTORNO

El juego inicia cuando el *Usuario Cloud* envía un *trabajo* al *CRAG*, el mismo que tiene adjunto un *SLA* y un valor máximo a pagar por la asignación de recursos, estos 2 parámetros son llamados de inicialización.

Adicionalmente, el *Usuario Cloud* define un nivel de importancia para la reputación, este parámetro junto con los parámetros de inicialización son procesados por el *Agente Intermediario*, de los cuales solamente el trabajo expresado en MIPS (Millones de

Instrucciones Por Segundos) es dado a conocer a los *Proveedores Cloud* que estén compitiendo.

Los *Proveedores Cloud* harán sus ofertas en orden de menor a mayor reputación, de esta forma los que tienen mayor reputación pueden ajustar su nivel de ganancia acorde a las ofertas que hayan sido realizadas anteriormente, para así aumentar las posibilidades de ganar el juego. Este ajuste de ganancias que realizan los Proveedores Cloud mientras el juego se desarrolla, forma parte del modelo de teoría de juegos propuesto, el mismo que se clasifica dentro de teoría de juegos dinámica.

El *CRAG* es un juego dinámico debido a que los jugadores observan las acciones de otros jugadores, antes de tomar una decisión, con el objetivo de jugar su estrategia más óptima acorde al entorno de ese momento en particular. Los Proveedores Cloud establecen inicialmente un costo de producción por MIPS, el mismo que será incrementado en un porcentaje para convertirse en el precio de mercado. Los porcentajes de ganancia mínima y deseada, son definidos inicialmente pero estos pueden variar en función de la mejor oferta que se haya postulado en el juego.

Se dirá que un proveedor postulará su precio mínimo si aplica el porcentaje de ganancia mínimo, y que postulará su precio deseado si aplica el porcentaje de ganancia deseado.

Las estrategias que jugarán todos los proveedores Cloud se listan a continuación:

- Aumentar el porcentaje de ganancia deseada en relación a la mejor oferta, solo si al hacerlo, el precio deseado se mantiene por debajo de la mejor oferta.
- Aumentar el porcentaje de ganancia mínimo en relación a la mejor oferta solo si el precio deseado es mayor a la mejor oferta y si al hacerlo, el precio mínimo se mantiene menor a la mejor oferta.
- Mantener el porcentaje de ganancia mínimo si no hay forma de ofrecer un precio menor al de la mejor oferta.

Una vez que todos los proveedores han enviado sus ofertas al *CRAG*, este se encarga de procesarlas y ajustarlas acorde a los parámetros de configuración recibidos inicialmente, para ello utilizará el mecanismo de reputación Beta [9] para predecir el comportamiento del proveedor en la siguiente transacción.

Finalmente, el agente intermediario selecciona al Proveedor Cloud que resulte ganador del *CRAG*. La asignación de recursos se realiza en la infraestructura del ganador y el agente intermediario recibe una comisión por parte del Proveedor Cloud, el cual consiste en el 15% del precio que el usuario cloud ha pagado por la asignación de recursos. El modelo se representa gráficamente en la figura 3.2

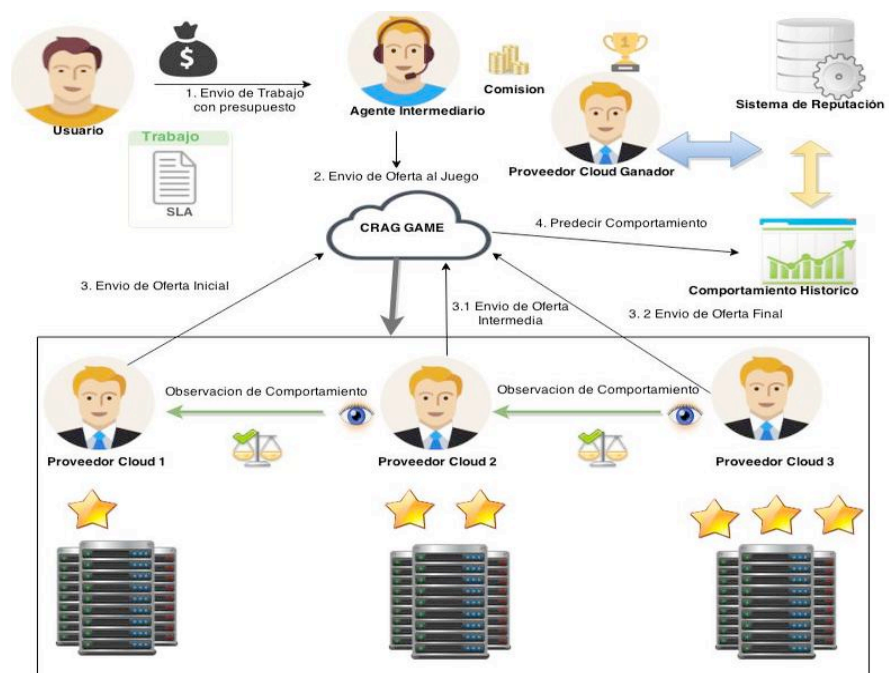


Figura 3.2 Representación gráfica del modelo

CAPÍTULO 4

4 DISEÑO DEL SIMULADOR

4.1 DIAGRAMA DE CASOS DE USO

Los casos de uso y sus relaciones se muestran en la figura 4.1.

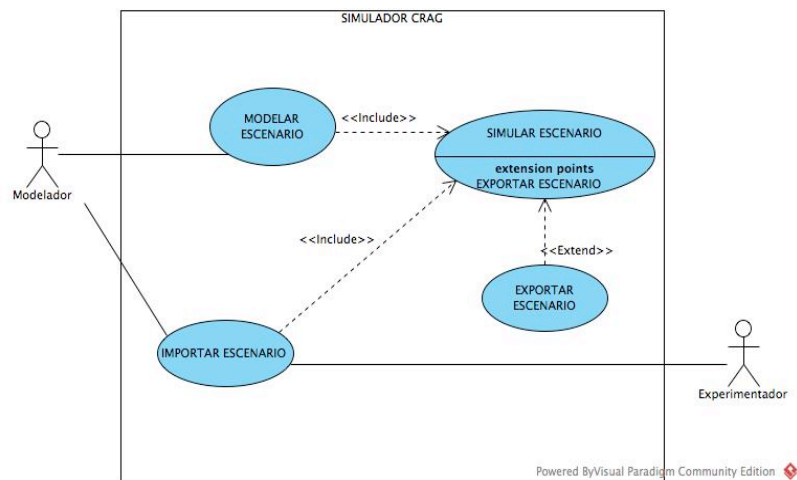


Figura 4.1 Diagrama de Casos de uso del simulador

4.1.1 ESPECIFICACIONES DE CASOS DE USO

I. **Nombre del Caso de Uso:** Modelar Escenario

Actor Principal: Modelador

Descripción: El actor define los parámetros necesarios para realizar una simulación

Precondiciones: Ninguna

Flujo Principal:

- a) El actor ingresa la información de los proveedores Cloud.
- b) El actor ingresa la información de tareas Cloud.
- c) El actor selecciona las métricas para la reputación.

Flujo Alternativo: Ninguno

Postcondiciones: La simulación del escenario puede ser ejecutada.

II. **Nombre del Caso de Uso:** Simular Escenario

Actor Principal: Experimentador, Modelador

Descripción: Realizar la simulación de asignación de recursos en Cloud, utilizando el mecanismo de reputación beta.

Precondiciones: El actor debería haber configurado el escenario de simulación con la información necesaria descrita en el caso I.

Flujo Principal:

- a) El actor inicia la simulación pulsando el botón ejecutar simulación.
- b) El actor continua con las rondas de simulación pulsando el botón continuar simulación.
- c) El actor selecciona las métricas para la reputación.

Flujo Alternativo: Ninguno

Postcondiciones: Se generan reportes que muestran el proceso de simulación y los resultados obtenidos.

III. Nombre del Caso de Uso: Exportar Escenario

Actor Principal: Modelador

Descripción: El actor exporta a un archivo XML el escenario de simulación configurado.

Precondiciones: Ninguna

Flujo Principal:

- a) El actor pulsa el botón exportar escenario.
- b) El actor ingresa el nombre y la ubicación en donde desea guardar el archivo.

Flujo Alternativo: Ninguno

Postcondiciones: El archivo descargado permitirá cargar un escenario de simulación configurado anteriormente.

IV. Nombre del Caso de Uso: Importar Escenario

Actor Principal: Experimentador, Modelador

Descripción: El actor exporta a un archivo XML el escenario de simulación configurado.

Precondiciones: Debe existir un archivo XML válido que corresponda a un escenario para poder realizar la importación.

Flujo Principal:

- a) El actor despliega el panel de importación de escenarios.
- b) El actor pulsa el botón seleccionar y sube el archivo XML que desee importar.

Flujo Alternativo: Ninguno

Postcondiciones: El escenario de simulación es recreado con los parámetros del archivo XML importado.

4.2 DIAGRAMA DE COMPONENTES

Los principales componentes utilizados en el simulador se muestran en la figura 4.2.

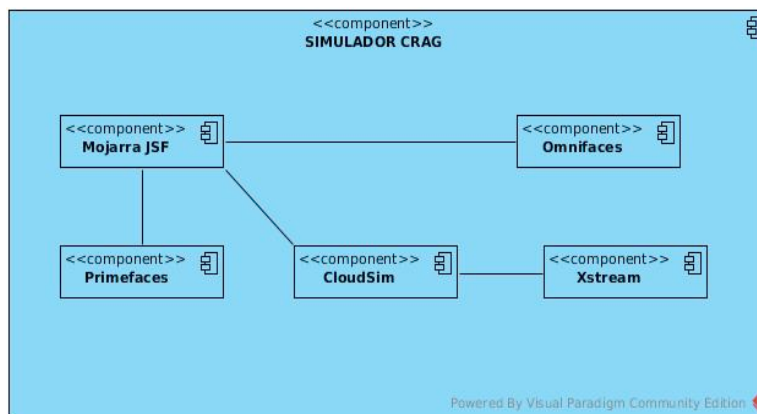


Figura 4.2 Diagrama de Componentes del Simulador

La funcionalidad principal de los componentes se describe a continuación:

- **Mojarra JSF:** Implementación Open Source de Oracle del estándar Java Server Faces (JSF) que simplifica el desarrollo de interfaces de usuario en aplicaciones Java EE.
- **Primefaces:** librería de componentes para Java Server Faces (JSF) de código abierto que cuenta con un conjunto de

componentes enriquecidos que facilitan la creación de las aplicaciones web.

- **Omnifaces:** Librería de utilidades para Java Server Faces (JSF) que se enfoca en utilidades que faciliten las tareas cotidianas del API estándar de JSF.
- **Xstream:** Librería de Java que permite serializar objetos a XML y viceversa.
- **CloudSim:** Framework que permite modelar y simular infraestructuras y servicios de Cloud Computing.

4.3 DIAGRAMA DE CLASES

La figura 4.3 muestra las clases principales del simulador y sus relaciones utilizadas en la implementación del mismo, las clases que fueron implementadas para el simulador se muestran de color celeste, mientras que las incorporadas en *CloudSim* se muestran de color verde.

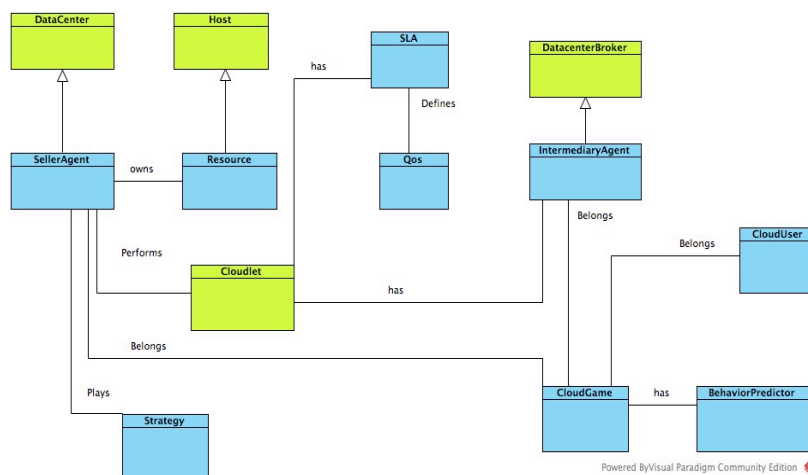


Figura 4.3 Diagrama de clases del simulador

Las clases más significativas del simulador son descritas a continuación:

- **DataCenter:** Esta clase modela los principales niveles de servicio a nivel de infraestructura, ofrecidos por un servidor de servicios en un ambiente de Cloud Computing.
- **Host:** Esta clase modela un servicio físico en un datacenter Cloud, contiene una cantidad de memoria, almacenamiento y una lista de elementos de procesamiento.

- **Cloudlet:** Esta clase modela los servicios de aplicaciones basadas en Cloud Computing, que son comúnmente desplegadas en los datacenters.
- **DataCenterBroker:** Esta clase modela un agente intermediario, que es responsable de mediar entre los usuarios cloud y los proveedores de servicios, para poder determinar los proveedores más convenientes que cumplan con los parámetros de calidad de servicio definidos por el usuario.
- **SellerAgent:** Esta clase hereda las funcionalidades de la clase Datacenter e incluye nuevos atributos y métodos relacionados al simulador.
- **IntermediaryAgent:** Esta clase hereda de la clase DatacenterBroker e incluye operaciones para regular el proceso de negociación entre el usuario y el proveedor de servicios cloud.
- **BehaviorPredictor:** Esta clase utiliza el mecanismo de reputación Beta para predecir el comportamiento futuro de un proveedor Cloud.

- **CloudGame:** Esta clase modela la funcionalidad principal del simulador, y procesa todos los parámetros previos a la simulación de un escenario.

4.4 DIAGRAMA DE SECUENCIA

En el (anexo c) se muestra el diagrama de secuencia desde el momento en el cual el usuario del simulador modela un escenario, estableciendo parámetros iniciales, hasta que un proveedor cloud es escogido como ganador, después de participar en la competencia de asignación de recursos para ejecutar las tareas requeridas por el usuario.

CAPÍTULO 5

5 IMPLEMENTACIONES Y PRUEBAS

5.1 ALGORITMO DE ASIGNACIÓN DE RECURSOS

El algoritmo de asignación de recursos implementado, utiliza el sistema de reputación Beta [9] para seleccionar el proveedor Cloud con menor probabilidad de cometer una violación SLA a un parámetro de servicio especificado por el usuario.

La razón principal por la cual el algoritmo utiliza un sistema de reputación, es para poder establecer un criterio de selección en base al comportamiento pasado que los proveedores han tenido, de esta manera el usuario cuenta con un parámetro adicional al precio, que ayudara a filtrar a proveedores con precios atractivos pero que no necesariamente representan una buena alternativa.

El algoritmo está estructurado de la siguiente manera:

Dada una colección de trabajos en la cual, cada uno de ellos:

- a) Requiere un número de instrucciones de cómputo que se ejecutarán en un entorno Cloud, medidas en MIPS.
- b) Pertenecen a un usuario Cloud, el cual especifica el valor máximo que pagará por el servicio de ejecutar este trabajo en un entorno Cloud.
- c) Requieren un espacio de memoria a ser asignado, medido en GB.
- d) Contienen un SLA adjunto en el cual se especifican parámetros de confiabilidad y disponibilidad, los cuales pueden tomar el valor de: BAJA, MEDIA, ALTA.

Dada una colección de n Proveedores Cloud, de los cuales cada uno es dueño de recursos con:

- a) Una capacidad de cómputo medida en MIPS.
- b) Un espacio de memoria en GB.

- c) Una probabilidad de éxito p y una probabilidad de fracaso $1-p$ para violar un parámetro de nivel de servicio (en este modelo el único parámetro es disponibilidad).

Los Proveedores Cloud harán sus ofertas en orden de menor a mayor reputación, por medio de la ecuación 5.1.

$$\text{Precio (Trabajo)} = \text{Costo producción} + \text{Ganancia deseada} \quad (5.1)$$

La ganancia deseada es un porcentaje del costo de producción e inicialmente es fijada ente un rango de valores mínimo y máximo, los mismos que serán ajustados por los proveedores de manera dinámica, a medida que los demás vayan postulando sus ofertas, con el objetivo de obtener una mayor ganancia. Un Agente Intermediario que controla todo el proceso de negociación ajusta todas las ofertas postuladas de los proveedores Cloud con reputación R por medio de la ecuación 5.2.

$$\text{Precio Ajustado} = \text{Precio Original} * (1 + \text{Penalidad}(R)) \quad (5.2)$$

Donde la penalidad se la calcula mediante la ecuación 5.3.

$$Penalidad (R) = \frac{1}{probabilidad\ éxito} \quad (5.3)$$

** importancia de reputación * confiabilidad (SLA)*

Se puede notar que la penalidad que tendrá cada uno de los proveedores Cloud será inversamente proporcional a su reputación, que esta medida por la probabilidad de éxito en sus transacciones. El factor importancia de reputación define que tan grande será el incremento del precio, así de esta forma se podría controlar el grado de afectación de la reputación en la penalidad que un proveedor Cloud recibirá. El pseudocódigo del algoritmo se muestra en (anexo B).

5.2 CARACTERÍSTICAS DEL SIMULADOR

El simulador permite a través de una interfaz web, realizar las siguientes acciones:

1. Configurar un escenario de simulación.
2. Simular un escenario, y continuar la simulación a partir de la última ejecutada.

3. Visualizar reportes después de ejecutar la simulación.
4. Exportar e importar escenarios de simulación a un archivo XML.

El simulador está implementado como una extensión del framework CloudSim. La interfaz Web se implementó utilizando el framework Java Server Faces con los componentes que se muestran en la figura 4.2.

5.3 CONFIGURACIÓN DEL JUEGO

El juego es representado como un escenario, en el cual pueden participar hasta un máximo de 8 proveedores Cloud, debido a que si el número es mayor, los reportes no se visualizarían claramente y costaría más trabajo adaptar para una adecuada visualización. Los proveedores competirán por la asignación de una lista de trabajos planificados que serán ejecutados en Cloud.

Cada proveedor Cloud es implementado como un Datacenter, el cual contiene un único Host, el mismo que está compuesto por 4 núcleos.

Los atributos que se muestran para cada uno de los proveedores son los siguientes:

- **Nombre:** La identificación del proveedor Cloud.
- **Reputación Inicial:** La reputación inicial del proveedor es representada como un número del 1 al 10, siendo 1 la reputación más baja y 10 la más alta. Estos valores fueron escogidos para que el usuario se le facilite diferenciar los niveles de reputación que un proveedor puede tomar, y para que no existan grandes varianzas entre estos valores. Esta reputación inicial es mapeada por el simulador para obtener la tasa de trabajos fallidos y trabajos exitosos inicialmente. estos valores serán procesados posteriormente por el sistema de reputación beta para predecir el comportamiento futuro de un proveedor.
- **Costo:** El costo representa la cantidad que el proveedor cobrará por MIPS, de forma lineal.

- **Ganancia Mínima:** Representa el porcentaje mínimo del costo de cada proveedor que será recargado al precio final de mercado.
- **Ganancia Deseada:** Representa el porcentaje mínimo del costo de cada proveedor que será recargado al precio final de mercado.

La figura 5.1 muestra cómo establecer las características de los proveedores Cloud que participaran en la simulación. Desde el simulador, la interfaz web permite añadir hasta 8 participantes y requiere un mínimo de 2 para que pueda realizarse la simulación.

Nombre	Reputacion Inicial	Costo(MIPS)	Ganancia Minima	Ganancia Deseada
Proveedor_A	★★★★★☆☆☆☆	\$1,00	0.3 %	0.6 %
Proveedor_B	★★★★★☆☆☆☆	\$1.20	0.3 %	0.5 %
Proveedor_C	★★★★★☆☆☆☆	\$1,50	0.2 %	0.4 %
Proveedor_D	★★★★★☆☆☆☆	\$1,60	0.2 %	0.5 %

La simulacion se realizara con 4 Proveedores Cloud

+ Añadir Proveedor Eliminar Proveedor

Figura 5.1 Configuración de proveedores Cloud en el simulador

La figura 5.2 muestra como añadir y establecer las características de los trabajos que serán ejecutados por el proveedor Cloud que resulte ganador de la simulación.

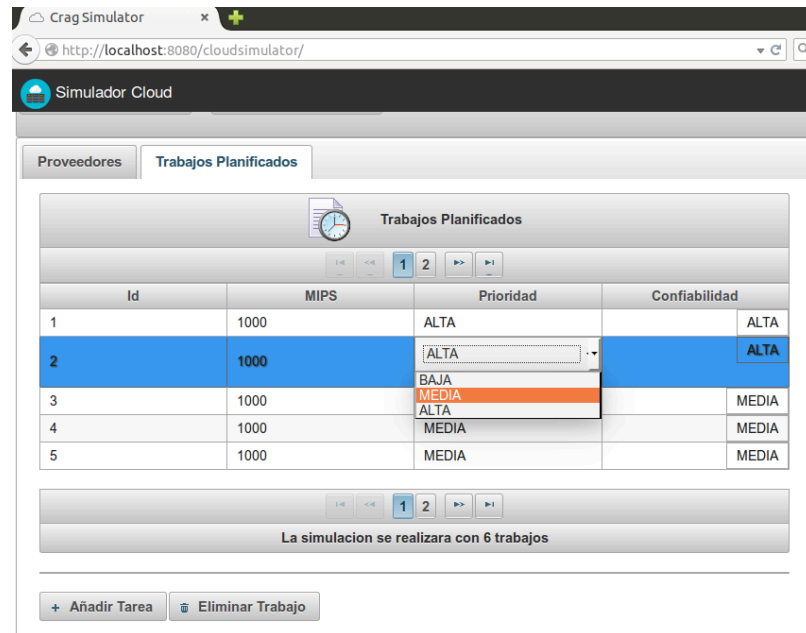


Figura 5.2 Configuración de trabajos planificados en el simulador

5.4 CONFIGURACIÓN DE LOS PARÁMETROS DEL JUEGO

Los parámetros del juego son las preferencias que el simulador permite establecer, para acorde a esto seleccionar al proveedor Cloud que mejor se ajuste a los criterios del usuario que requiere ejecutar un trabajo en la Nube. La figura 5.3 muestra como un

usuario puede ajustar los parámetros importancia de reputación y factor de olvido.

Mientras más alto sea el valor de la importancia de reputación, mayor será la penalidad que se aplicará para cada uno de los proveedores.



Figura 5.3 Parámetros de configuración del simulador

5.5 EXPORTANDO E IMPORTANDO ESCENARIOS

Una característica importante del simulador es la posibilidad de importar y exportar escenarios previamente configurados por el usuario. El escenario configurado se podrá exportar a un archivo

XML, el mismo que permitirá recrear la configuración establecida en un principio, y con ello corroborar resultados.

La figura 5.4 muestra el proceso de exportar un escenario, para hacerlo, se debe hacer click sobre el botón azul que dice “Exportar Escenario”, y la ventana emergente predeterminada del navegador sobre el cual se esté ejecutando el simulador, aparecerá permitiendo guardar el archivo.

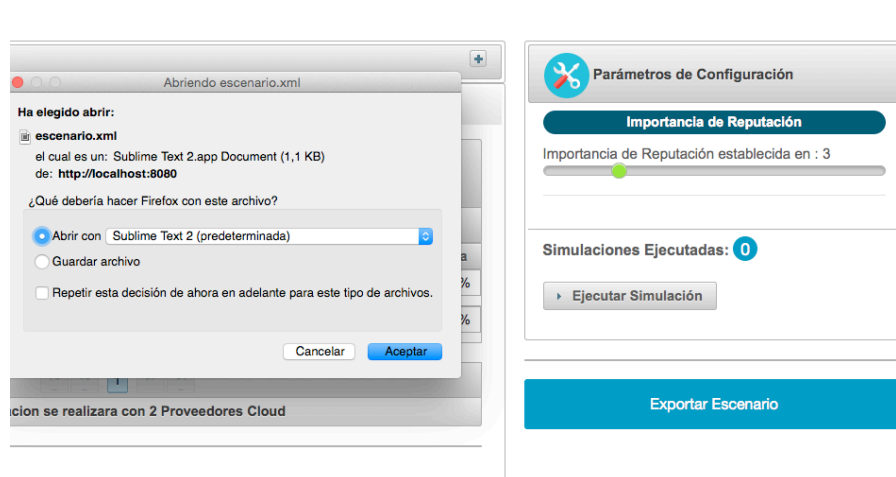


Figura 5.4 Exportación de escenario

Las figuras 5.5 y 5.6 muestran el proceso de importar un escenario al simulador, a partir de un archivo XML, para hacerlo se debe abrir el panel que dice importar escenario y luego dar click en el botón

subir, una vez que el archivo ha sido subido, el escenario de simulación se cargara tal cual fue exportado.

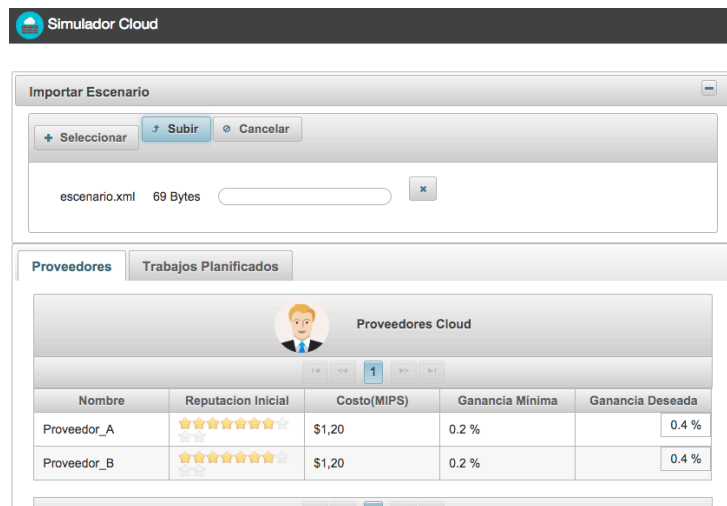


Figura 5.5 Importación de escenario

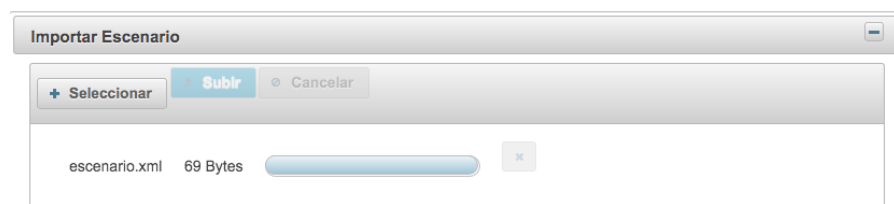


Figura 5.6 Carga de escenario

5.6 REPORTE DE SIMULACIÓN

Una vez que el escenario ha sido ejecutado, se presentan 2 reportes al usuario, el primero es un gráfico de barras que muestra

el número de violaciones que ha cometido cada uno de los proveedores participantes en ese escenario, y el segundo es un gráfico de líneas que muestra el número de violaciones que ha ido ocurriendo por cada simulación ejecutada. Las figuras 5.7 y 5.8 muestran los diagramas de barras y de líneas después de haber realizado 10 simulaciones.



Figura 5.7 Violaciones SLA por Proveedor

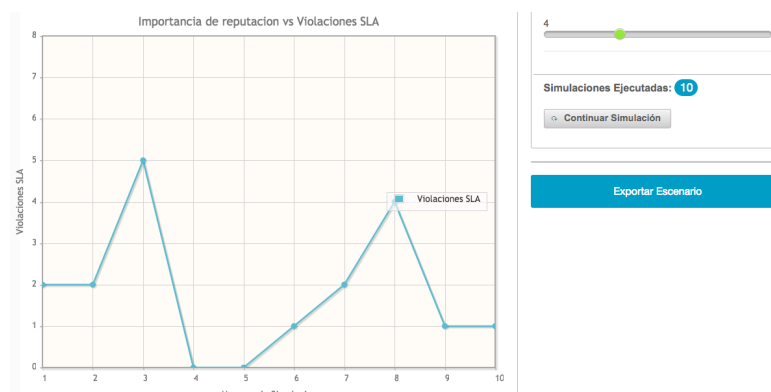


Figura 5.8 Violaciones SLA vs Importancia de reputación

5.7 PRUEBAS APLICADAS A LA HERRAMIENTA DE SIMULACIÓN

Las pruebas aplicadas a la herramienta de simulación se encuentran en el (anexo D).

CAPÍTULO 6

6 ANÁLISIS DE RESULTADOS

6.1 ANALISIS DE RESULTADOS

Para efectuar las simulaciones se estableció un escenario inicial, sobre el cual se ejecutaron 5 rondas de simulaciones, las cuales estuvieron compuestas de 10 iteraciones cada una, cabe recalcar que las iteraciones no son repeticiones, debido a que los eventos que ocurran en cada una de ellas influirán en la siguiente iteración. La razón por la cual esto sucede es debido a que cada vez que un proveedor realiza un trabajo, ya sea exitoso o fallido, la probabilidad de que realice un trabajo con éxito en la siguiente simulación es recalculada utilizando el mecanismo de reputación Beta [9], esto puede hacer que si un proveedor ha resultado ganador y ha cometido violaciones SLA de manera seguida, su probabilidad de éxito se vea afectada y disminuya a medida que las iteraciones

aumentan. Por otra parte, si el proveedor tenía una baja probabilidad de éxito y ha realizado trabajos exitosos subsecuentes, su probabilidad de éxito podría aumentar acorde al número de iteraciones. El escenario que se estableció al inicio de todas las rondas fue presentado en la tabla 5.

Tabla 5 Escenario de simulación

Proveedor	Reputación Inicial	Costo(MIPS)	Ganancia mínima(%)	Ganancia Deseada(%)
A	3	\$1	0,2	0,4
B	5	\$1,3	0,3	0,5
C	7	\$1,6	0,3	0,5
D	9	\$1,9	0,4	0,5

Para cada una de las rondas, la importancia de reputación fue aumentándose progresivamente de 2 en 2, comenzando en 2 y terminando en 10, con esto se esperaba que a medida que la importancia de reputación aumente, el número de violaciones SLA disminuyan.

Al realizar la experimentación se obtuvieron los siguientes resultados:

Tabla 6 Resultados para simulación con importancia de reputación: 2

SIMULACION 1											IMPORTANCIA DE REPUTACION: 2										
VIOLACIONES SLA																					
Proveedor A	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	18
Proveedor B	2	2	2	1	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	0	17
Proveedor C	2	2	1	2	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	11
Proveedor D	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	5
TOTAL	7	7	6	6	6	5	4	5	5	5	3	3	3	3	3	3	3	3	3	3	51

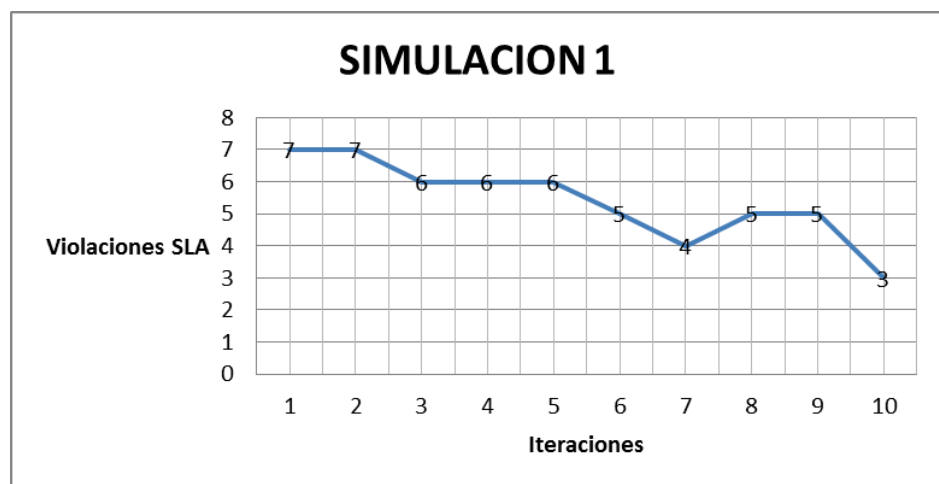


Figura 6.1 Violaciones SLA en simulación 1

Tabla 7 Resultados para simulación con importancia de reputación: 4

SIMULACION 2		IMPORTANCIA DE REPUTACION: 4										
VIOLACIONES SLA												
Proveedor A		2	2	1	2	1	0	1	2	1	1	12
Proveedor B		1	2	2	1	2	1	1	1	1	0	12
Proveedor C		2	1	2	1	0	1	1	1	0	0	9
Proveedor D		1	1	1	1	1	1	1	0	0	0	7
TOTAL		6	6	6	5	4	3	4	4	2	1	40



Figura 6.2 Violaciones SLA en simulación 2

Tabla 8 Resultados para simulación con importancia de reputación: 6

SIMULACION 3		IMPORTANCIA DE REPUTACION: 6										
VIOLACIONES SLA												
Proveedor A		1	1	1	1	1	0	1	1	0	1	8
Proveedor B		1	1	1	0	1	1	1	1	1	0	8
Proveedor C		1	1	1	1	0	1	1	0	1	0	7
Proveedor D		0	1	0	1	0	1	0	0	0	0	3
TOTAL		3	4	3	3	2	3	3	2	2	1	26



Figura 6.3 Violaciones SLA en simulación 3

Tabla 9 Resultados para simulación con importancia de reputación: 8

SIMULACION 4		IMPORTANCIA DE REPUTACION: 8										
VIOLACIONES SLA												
Proveedor A		1	1	1	1	1	1	1	1	0	0	8
Proveedor B		1	1	1	0	1	1	0	0	0	0	5
Proveedor C		0	1	1	1	0	0	0	0	0	0	3
Proveedor D		1	0	0	0	0	0	0	0	0	0	1
TOTAL		3	3	3	2	2	2	1	1	0	0	17

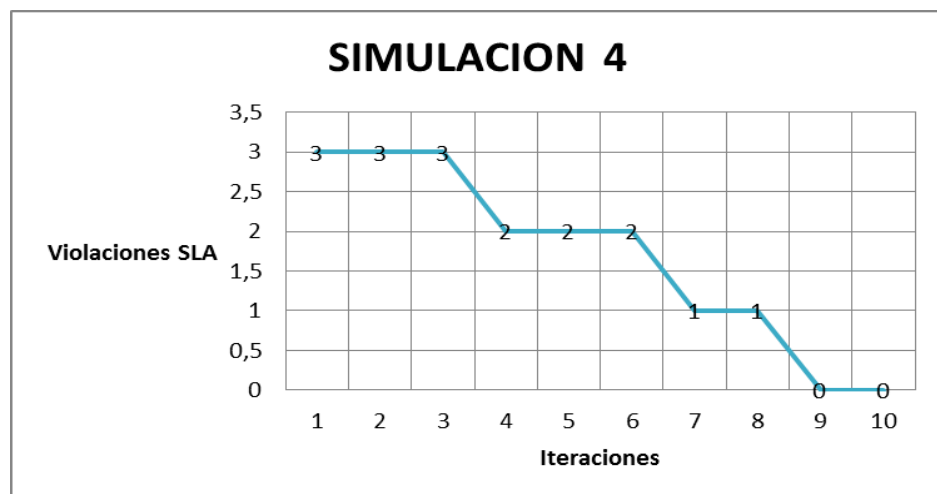


Figura 6.4 Violaciones SLA en simulación 4

Tabla 10 Resultados para simulación con importancia de reputación: 10

SIMULACION 5		IMPORTANCIA DE REPUTACION: 10										
VIOLACIONES SLA												
Proveedor A		1	1	0	1	1	1	1	1	0	0	7
Proveedor B		1	0	1	0	0	1	0	0	0	0	3
Proveedor C		0	0	1	1	0	0	0	0	0	0	2
Proveedor D		0	1	0	0	0	0	0	0	0	0	1
TOTAL		2	2	2	2	1	2	1	1	0	0	13

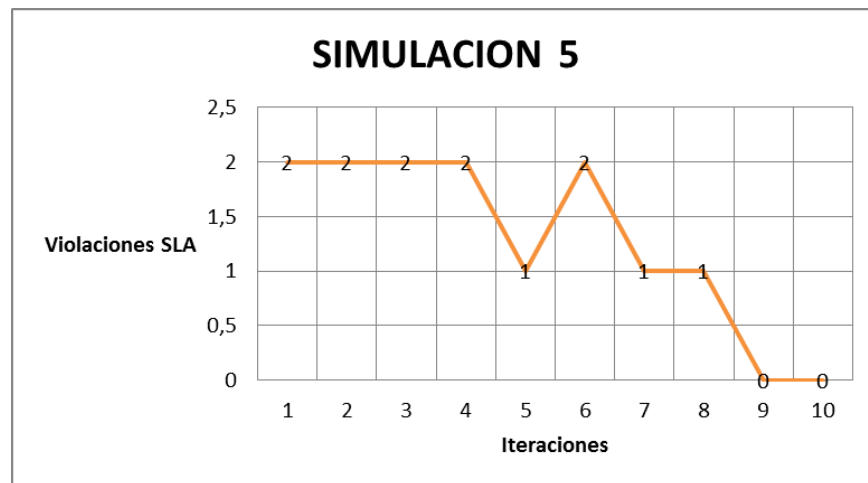


Figura 6.5 Violaciones SLA en simulación 5

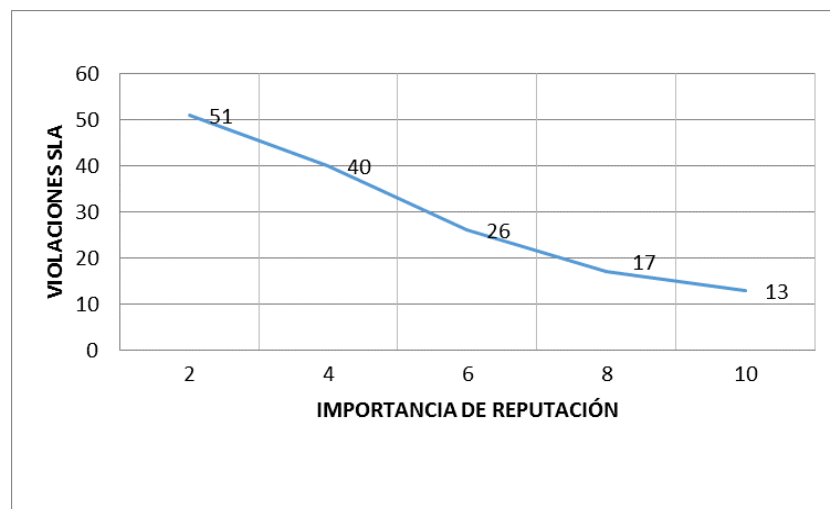


Figura 6.6 Importancia de reputación vs violaciones SLA

Para cada una de las rondas de simulaciones realizadas, se obtuvo el número de violaciones SLA por cada proveedor en cada iteración. En todas las simulaciones, se pudo observar que el mayor número de violaciones es cometido por los proveedores con más baja reputación.

Para cada una de las gráficas de las simulaciones, a medida que la importancia de reputación va aumentando, el número de violaciones SLA, se va reduciendo gradualmente y se puede observar que hay una tendencia a que estos se reduzcan en mayor cantidad si la importancia de reputación se establece en el valor máximo de 10.

Finalmente en la figura 6.6 se muestra el número total de violaciones acumuladas que han cometido los proveedores Cloud por cada ronda de simulación, y se puede notar claramente que al aumentar el nivel de importancia de reputación, el número de violaciones SLA tiende a reducirse considerablemente.

CONCLUSIONES

1. Finalmente se puede concluir que un modelo que toma en cuenta la reputación de los proveedores para establecer un mecanismo de negociación entre cliente y proveedor, siempre será de beneficio para el usuario cloud ya que, no solamente se puede guiar por el precio, sino también por el nivel de calidad de servicio.
2. Se puede concluir también que el uso de la teoría de juegos, que cada proveedor efectúa cada vez que varía o ajusta su oferta para tratar de ganar la competencia por los recursos a asignar, es de beneficio para el usuario final y para los proveedores ya que permite establecer un precio más justo para el usuario y hace que las probabilidades de que un proveedor con más reputación gane aumenten debido a que las ofertas se realizan en orden de menor a mayor reputación.

3. Al incluir un modelo que toma en cuenta la reputación de los proveedores Cloud como criterio de selección para realizar tareas requeridas por un usuario Cloud, el número de violaciones SLA se reduce en promedio, a medida que el nivel del parámetro importancia de reputación es incrementado.

RECOMENDACIONES

1. Para reducir la curva de aprendizaje del framework CloudSim, se recomienda, además de descargar los ejemplos incluidos en la documentación, hacer depuración al código de los ejemplos para que se pueda entender de una mejor manera cual es el flujo de eventos que se dan cuando una simulación se lleva a cabo.
2. Para mejorar la implementación del sistema de reputación Beta, se recomienda añadir al modelo propuesto, un factor de olvido, en el cual, las calificaciones mas antiguas que reciban los proveedores Cloud, tengan menos peso que las mas recientes, de esta manera se deja abierta la posibilidad de que el comportamiento de un proveedor pueda cambiar a lo largo del tiempo.

ANEXO A

DATOS DE PROVEEDORES Y PARÁMETROS UTILIZADOS PARA LA SIMULACIÓN

```

<?xml version="1.0" encoding="utf-16"?>
<scenario>
  <sellers>
    <seller>
      <name>Proveedor_A</name>
      <initialReputation>3</initialReputation>
      <cost>1</cost>
      <minimunProfit>0.2</minimunProfit>
      <preferredProfit>0.4</preferredProfit>
    </seller>
    <seller>
      <name>Proveedor_B</name>
      <initialReputation>5</initialReputation>
      <cost>1.3</cost>
      <minimunProfit>0.3</minimunProfit>
      <preferredProfit>0.5</preferredProfit>
    </seller>
    <seller>
      <name>Proveedor_C</name>
      <initialReputation>7</initialReputation>
      <cost>1.6</cost>
      <minimunProfit>0.3</minimunProfit>
      <preferredProfit>0.5</preferredProfit>
    </seller>
    <seller>
      <name>Proveedor_D</name>
      <initialReputation>9</initialReputation>
      <cost>1.9</cost>
      <minimunProfit>0.4</minimunProfit>
      <preferredProfit>0.5</preferredProfit>
    </seller>
  </sellers>
  <preferences>
    <reputationWeight>2</reputationWeight>
    <reputationWeight>4</reputationWeight>
    <reputationWeight>6</reputationWeight>
    <reputationWeight>8</reputationWeight>
  </preferences>
  <jobs>
    <job>
      <id>1</id>
      <mips>1000</mips>
      <priority>MEDIA</priority>
      <reliability>ALTA</reliability>
    </job>
    <job>
      <id>1</id>
      <mips>1000</mips>
    </job>
  </jobs>

```

```
        <priority>MEDIA</priority>  
        <reliability>ALTA</reliability>  
    </job>  
</jobs>  
</scenario>
```

ANEXO B

PSEUDOCÓDIGO DEL ALGORITMO PARA ASIGNACIÓN DE RECURSOS CLOUD

Input: *reputationImportance, forgettingFactor, job, sellerAgents*
Output: Best seller agent for perform *job*

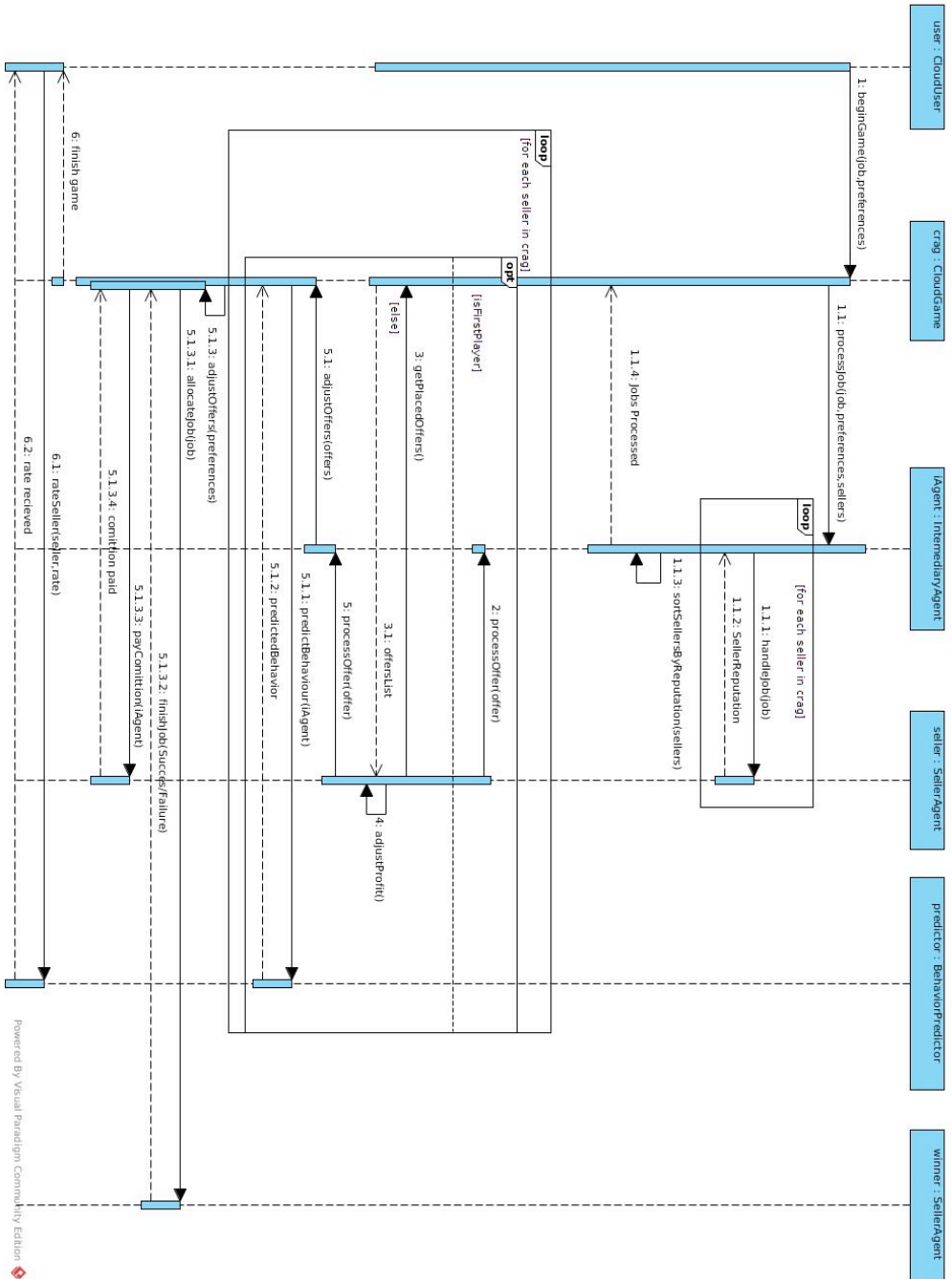
- 1 Initialize CloudSim enviroment
- 2 Initialize Game enviroment
- 3 *competitors* \leftarrow sellerAgents
- 4 *simulations* \leftarrow 0
- 5 *competitionJob* \leftarrow job
- 6 Sort *competitors* by Reputation
- 7 Begin game with
competitors, competitionJob, reputationImportance
- 8 Order seller agents by Reputation
- 9 **foreach** *competitor* in *competitors* **do**
- 10 | *minimunProfit* \leftarrow competitor minimunprofit
- 11 | *preferredProfit* \leftarrow competitor preferred profit
- 12 | *productionCost* \leftarrow competitor production cost of *job*
- 13 | *probabilityOfSuccess* \leftarrow competior probability of success
- 14 | *minunOffer* \leftarrow productionCost * (1+*minimunProfit*)
- 15 | *preferredOffer* \leftarrow productionCost * (1+*preferredProfit*)
- 16 | **if** Game Offers list is empty **then**
- 17 | | add *preferredOffer* to game' s offer list
- 18 | | *ponderedOffer* \leftarrow PonderOffer (*preferredOffer*)
- 19 | | add *ponderedOffer* to Intermediary Agent' s offer list
- 20 | **else**
- 21 | | *bestOffer* \leftarrow GetLowerOffer(*game*)
- 22 | | **if** *bestOffer* > *preferredOffer* **then**
- 23 | | | *delta* \leftarrow bestOffer -preferredOffer
- 24 | | | *adjustedOffer* \leftarrow AdjustOffer(*bestOffer*, *delta*,
| | | *preferredProfi*
| | | t)
- 25 | | | add *adjustedOffer* to game' s offer list
- 26 | | | *ponderedOffer* \leftarrow PonderOffer(*adjustedOffer*)
- 27 | | | add *ponderedOffer* to Intermediary Agent' s offer list
- 28 | | **else**
- 29 | | | **if** *bestOffer* > *minimunOffer* **then**
- 30 | | | | *delta* \leftarrow bestOffer -minimunOffer
- 31 | | | | *adjustedOffer* \leftarrow AdjustOffer (*bestOffer*, *delta*, *minimunProfit*)
- 32 | | | | add *adjustedOffer* to game' s offer list
- 33 | | | | *ponderedOffer* \leftarrow PonderOffer (*adjustedOffer*)
- 34 | | | | add *ponderedOffer* to Intermediary Agent' s offer list

```
35     else
36         add minunOffer to game' s offer list
37         ponderedOffer ← PonderOffer(minunOffer)
38         add ponderedOffer to Intermediary Agent' s offer list

39 sort IntermediaryAgent' s pondered offers list
40 winnerAgent ← GetWinnerAgent(game);
41 allocate job in winnerAgent infrastructure
42 rate winnerAgent after transaction ends
43 PayBrokerCommission ()
44 UpdateReputation(winnerAgent)
45 simulations++
```

ANEXO C

DIAGRAMA DE SECUENCIA



ANEXO D

TABLA DE PRUEBAS

Código de caso prueba	Entradas	Salidas esperadas	Salidas reales	Observaciones
TC-001	En la pestaña de proveedores, hacer clic en el botón Añadir Proveedor	Un nuevo proveedor por defecto debería ser añadido al final de la tabla de proveedores Cloud	Un nuevo proveedor por defecto fue añadido al final de la tabla de proveedores Cloud	Pasa la prueba
TC-002	En la pestaña de proveedores, seleccionar una fila de la tabla de Proveedores Cloud y modificar el porcentaje de ganancia mínima del proveedor seleccionado a 0.1	No se debería poder establecer el valor del porcentaje de ganancia mínima, y debería permanecer en el valor de 0.2	El porcentaje de ganancia mínima toma el valor de 0.2	Pasa la prueba
TC-003	En la pestaña de proveedores, seleccionar una fila y modificar el porcentaje de ganancia deseada a 1.3	No se debería poder establecer el valor de porcentaje de ganancia deseada y debería permanecer en el valor de 1.0	El porcentaje de ganancia deseada toma el valor de 1.0	Pasa la prueba
TC-004	En la pestaña de proveedores, seleccionar una fila de la tabla de Proveedores Cloud y hacer clic en el botón eliminar proveedor.	El proveedor seleccionado, debería ser removido de la tabla de Proveedores Cloud	El proveedor seleccionado es removido de la tabla de Proveedores Cloud	Pasa la prueba
TC-005	En la pestaña de proveedores, seleccionar una fila de la tabla de Proveedores Cloud y hacer clic en el botón eliminar proveedor cuando únicamente hayan 2 proveedores	Un mensaje de error debería aparecer advirtiéndolo siguiente: "Debe haber al menos 2 proveedores para la simulación"	El mensaje de error se muestra advirtiéndolo: "Debe haber al menos 2 proveedores para la simulación"	Pasa la prueba

TC-006	En la pestaña de Trabajos planificados, hacer clic en el botón Añadir Tarea	Una nueva tarea por defecto debería ser añadida al final de la tabla de Trabajos Planificados	Una nueva tarea por defecto fue añadida al final de la tabla de Trabajos Planificados	Pasa la prueba
TC-007	En la pestaña de Trabajos Planificados, seleccionar una fila de la tabla de Trabajos Planificados , y modificar la columna MIPS a 0	El valor mínimo de 400 debería mostrarse en la columna MIPS	El valor de 400 se muestra en la columna MIPS	Pasa la prueba
TC-008	En la pestaña de Trabajos Planificados, seleccionar una fila de la tabla de Trabajos Planificados y hacer clic en el botón eliminar tarea cuando únicamente hayan 2 tareas.	Un mensaje de error debería aparecer advirtiendo lo siguiente: "Debe haber al menos 2 proveedores para la simulación"	El mensaje de error se muestra advirtiendo: "Debe haber al menos un trabajo para la simulación"	Pasa la prueba
TC-009	En la pantalla principal, dentro del panel :parámetros de configuración, hacer clic en el botón Ejecutar simulación	El sistema debería ejecutar la simulación, mostrar el botón Continuar Simulación y crear 2 nuevas pestañas llamadas Resultados y Violaciones SLA, conteniendo gráficos y resultados de la simulación	El sistema ejecuta la simulación, muestra el botón Continuar Simulación y crea 2 nuevas pestañas llamadas Resultados y Violaciones SLA conteniendo gráficos y resultados de la simulación	Pasa la prueba
TC-010	En la pantalla principal, dentro del panel: parámetros de simulación, hacer clic en el botón Continuar Simulación	El sistema debería ejecutar la simulación, tomando los datos de la simulación anterior y actualizar los resultados y gráficos de la nueva simulación	El sistema ejecuta la simulación, con los datos de la simulación anterior y actualiza los resultados y gráficos de la nueva simulación	Pasa la prueba

TC-011	En la pantalla principal, hacer clic en el botón Exportar Escenario	El sistema debería, generar un archivo XML, con el escenario que este configurado.	El sistema genera un archivo XML con el escenario configurado	Pasa la prueba
TC-012	En el panel importar escenario, hacer clic en el botón seleccionar	La aplicación debería abrir el archivo seleccionado, cargar el escenario en el simulador y mostrar el mensaje: "Escenario importado con éxito"	La aplicación abre el archivo seleccionado, carga el escenario correspondiente, y muestra el mensaje: "Escenario importado con éxito"	Pasa la prueba

BIBLIOGRAFÍA

- [1] Tecayehuatl, E. «El origen de: El Cómputo en la Nube,» [En línea]. Disponible: <https://www.fayerwayer.com/2012/01/el-origen-de-el-computo-en-la-nube/>. [Último acceso: Junio 2014].
- [2] National Institute of Standards and Technology, «The NIST Definition of Cloud,» [En línea]. Disponible: <http://csrc.nist.gov/publications/nistpubs/800-145/SP800-145.pdf>. [Último acceso: Septiembre 2014].
- [3] Farmer, R., y Glass, B., «Building Web reputation systems», O' Reilly, 2010.
- [4] Fernández, F., «Teoría de juegos: análisis matemático de conflictos,» [En línea]. Disponible: <http://imarrero.webs.ull.es/sctm05/modulo1lp/5/ffernandez.pdf>. [Último acceso: Julio 2014].
- [5] CIOAL, «Estudio Global revela crecimiento de la confianza en el Cloud Computing,» [En línea]. Disponible: <http://www.cioal.com/2011/06/01/estudio-global-revela-crecimiento-de-la-confianza-en-el-cloud-computing/>. [Último acceso: Junio 2014].
- [6] Landa, H., «Six reasons why your company should move to the cloud,» [En línea]. Disponible: <http://www.bizjournals.com/bizjournals/how-to/technology/2014/04/6-reasons-why-your-company-should-move-to-the.html>. [Último acceso: Julio 2014].

- [7] Campbell, R., Kozuch, M., Kunze, M., Kwan, T., Lyons, M., y Chai, Y., «Open Cirrus Cloud Computing Testbed,» [En línea]. Disponible: <http://opencirrus.intel-research.net/doc/droh-opencirrus-whitepaper-hotcloud09.pdf>. [Último acceso: Agosto 2014].
- [8] Antonopoulos, N., y Gillam, L., «Cloud Computing: Principles, Systems and Applications», Springer, 2010.
- [9] Schlosser, A., Voss, M., y Bruckner, L., «Comparing and Evaluating Metrics for Reputation Systems by Simulation,» [En línea]. Disponible: <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.142.2347&rep=rep1&type=pdf>. [Último acceso: Julio 2014].
- [10] Smith, V., «¿Qué es economía Experimental?,» [En línea]. Disponible: http://www.academia.edu/1129890/Qu%C3%A9_es_econom%C3%ADa_Experimental. [Último acceso: Agosto 2014].
- [11] Romp, G., «Game Theory: Introduction and Application», Oxford University Press, 1997.
- [12] Lynn, G., y Lauren, P., «SLA Definitions and Solutions,» [En línea]. Disponible: <http://www.cio.com/article/2438284/outsourcing/sla-definitions-and-solutions.html>. [Último acceso: Agosto 2014].
- [13] Calheiros, R., Ranjan, R., y De Rose, C., «CloudSim: a toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms,» [En línea]. Disponible:

- <http://www.buyya.com/papers/CloudSim2010.pdf>. [Último acceso: Agosto 2014].
- [14] Paniagua, F., «Gestor de entornos de simulación de cloud computing,» [En línea]. Disponible: http://e-archivo.uc3m.es/bitstream/handle/10016/15533/MemoriaPFC_FcoJavier_Paniagua_Correa.pdf?sequence=1. [Último acceso: Agosto 2014].
- [15] Guiyi, W., Athanasios, V., y Yao, Z., «A game-theoretic method of fair resource allocation for cloud computing services,» [En línea]. Disponible: http://140.123.102.14:8080/reportSys/file/paper/cktsung/cktsung_39_paper.pdf. [Último acceso: Marzo 2015].
- [16] Wenjuan, L., y Lingdi, P., «Trust Model to Enhance Security and Interoperability of Cloud Environment,» [En línea]. Disponible: <http://barbie.uta.edu/~hdfeng/CloudComputing/cc/cc02.pdf>. [Último acceso: Marzo 2015].
- [17] Mera, C., «Simulation Tool for Market-Based Cloud Resource Allocation,» [En línea]. Disponible: <http://repositorio.educacionsuperior.gob.ec/bitstream/28000/277/1/T-SENESCYT-0047.pdf>. [Último acceso: Marzo 2015].