

Incidencia de atributos de calidad de software en el diseño, construcción y despliegue de ambientes arquitectónicos Cloud

Incidence of software quality attributes in the design, construction and deployment of Cloud architectural environments

Armando Cabrera S., Marco Abad E., Danilo Jaramillo H., Ana Poma G.
Universidad Técnica Particular de Loja
Loja, Ecuador
{aacabrera, mpabad, djaramillo, alpomax}@utpl.edu.ec

José Carrillo Verdúm
Universidad Politécnica de Madrid
Madrid, España
jcarrillo@fi.upm.es

Resumen — Cloud Computing (CC) es un nuevo paradigma en el mundo de la computación, incluye varios modelos de implementación y de servicio. Este proyecto forma parte de un estudio general que inicia con el proceso de ingeniería y el ciclo de vida de desarrollo de software (SDLC), y se sustenta en la norma ISO / IEC / IEEE 12207 para evaluar la implementación del proceso de software, luego se revisan algunos aspectos de calidad aplicando la norma ISO / IEC 9126; finalmente se revisan algunos términos clave, características, modelos, arquitectura, taxonomía, escenarios de implementación y las declaraciones de alcance de Cloud Computing, con lo cual se procede a identificar las características claves de entornos SaaS públicas y privadas. Se obtuvo un modelo de calidad de servicio (QoS) utilizando atributos de calidad y sus correspondientes métricas derivadas de la dicha norma ISO / IEC 9126.

Palabras Clave -. Computación en la nube, Atributos de calidad, Arquitectura de Software, SDLC

Abstract - Cloud Computing (CC) is a new paradigm in the world of computing, it includes several service and deployment models. This project is part of a general study that starts from the software engineering process and the software development life cycle (SDLC), this study is based on the ISO / IEC / IEEE 12207 standard, to evaluate the software implementation process and then we review some software quality aspects by applying the standard ISO / IEC 9126; finally we review some key terms, features, models, architecture, taxonomy, deployment scenarios and scope statements of Cloud Computing. After that we proceed to identify the key characteristics of private and public SaaS environments and obtained a quality model of service (QoS) using quality attributes and their corresponding metrics derived from the ISO / IEC 9126 Standard.

Keywords - Cloud Computing, Quality Attributes, Software Architecture, SDLC

I. INTRODUCCIÓN

El diseño de una arquitectura de software es un proceso iterativo basado en decisiones tomadas en etapas tempranas del ciclo de vida de desarrollo del software (SDLC –Software Development Life Cycle). Los conductores clave del negocio definen los atributos de calidad, e influyen significativamente en las decisiones del diseño de la arquitectura. En primera iteración del diseño arquitectónico se desarrolla un esqueleto que se ocupa del comportamiento del sistema y proporciona las bases para el perfeccionamiento de los requisitos. El proceso iterativo continúa hasta que se definen y documentan los sub-componentes del sistema en el que se puede producir un diseño detallado. La arquitectura define la estructura general del software a través de componentes y de las interacciones entre éstos, en donde un componente puede identificarse como una unidad que lleva a cabo alguna función. Es importante destacar que los escenarios operativos han cambiado significativamente el enfoque en el que se desarrollan los sistemas, pensar en arquitecturas orientadas a servicios (SOA –Service Oriented Architecture) o computación en la nube (CC –Cloud Computing) hacen que la captura inicial de requisitos se convierta en factor clave de la identificación de atributos de calidad para el diseño arquitectónico del sistema.

La computación en nube comprende tres modelos de servicio: Software como Servicio (SaaS), Plataforma como Servicio (PaaS), e Infraestructura como Servicio (IaaS), así como cuatro modelos de despliegue: privado, comunitario, público e híbrido. A pesar de la importancia y difusión de los servicios que ofrece, no se han evidenciado investigaciones

asociadas a los parámetros de calidad que merezcan ser tomados en cuenta para su implementación.

En el presente trabajo se identifican atributos de calidad tomando como referencia la norma ISO/IEC 9126 para el establecimiento de un modelo de calidad de servicios (QoS – Quality of Service) de modo que incida en los criterios de selección para el diseño arquitectónico, construcción y despliegue de entornos SaaS privados y públicos.

II. MARCO DE REFERENCIA

A. Ciclo de vida de desarrollo del Software (SDLC)

El ciclo de vida de desarrollo del software es “un marco de referencia que contiene procesos, actividades y tareas involucradas en el desarrollo, explotación y mantenimiento de un producto software, abarcado la vida del sistema desde la definición de los requisitos hasta la finalización de su uso” [1], [2]. Este paradigma propone la división de un proyecto en etapas, con entradas, procesos y salidas definidas.

La norma ISO 12207:2008, procesos del ciclo de vida del software [3] organiza el proceso de adquisición, suministro, desarrollo, operación y mantenimiento de productos y servicios de software. Su objetivo fundamental es proporcionar una guía para el proceso de obtención de requerimientos, diseño, desarrollo, implementación y pruebas. Existen dos grandes grupos de procesos en esta norma, el primero es un conjunto de procesos de cara al contexto del sistema, es decir a la gestión del proyecto mismo, y el segundo es un grupo de procesos específicos de software. En conjunto forman siete grupos de procesos dentro del SDLC [4]. El presente proyecto se enfoca en el quinto grupo del SDLC – procesos de implementación del software, cuyo propósito es producir un elemento especificado del sistema implementado como un producto o servicio software, éste a su vez contiene procesos de nivel inferior que se detallan a continuación.

Proceso de análisis de requisitos del software, su propósito es establecer los requisitos de los elementos software del sistema. Implica el análisis y descripción exacta de los requisitos para su validación; la verificación de la implementación y la estimación de costes; y el desarrollo de modelos para entender de mejor manera un problema del mundo real [5]. En [6] se menciona que dichos modelos se deben crear de acuerdo a diferentes niveles de abstracción, es decir desde el punto de vista del cliente y después en base a una representación de nivel técnico.

Proceso de diseño de la arquitectura del software, su propósito es proporcionar un diseño para el software que lo implementa y que pueda ser verificado con los requisitos [3]. La arquitectura del software es “una descripción de los subsistemas y de los componentes de un sistema de software y de las relaciones entre ellos, procurando así definir la estructura interna” [7]. El diseño de la arquitectura del software produce una descripción de la estructura interna del software que servirá como base para su construcción y descomposición, la organización de los componentes y sus interfaces, y los componentes en un nivel de detalle que permita su construcción, produciendo diversos modelos para la solución que se pondrá en desarrollo [5].

Proceso de diseño detallado del software, su propósito es proporcionar el diseño para el software que lo implementa, que pueda ser verificado con los requisitos y la arquitectura de software, y que contenga suficientes detalles para permitir la codificación y pruebas [3]. Los estilos y patrones arquitectónicos detallan el diseño del software. Los estilos arquitectónicos definen la arquitectura a nivel macro y los patrones arquitectónicos se utilizan para describir los detalles del software a bajo nivel [8]. En el presente trabajo se considera a SOA como base para los esquemas de computación en la nube.

Proceso de construcción del software, su propósito del es producir unidades de software ejecutables que reflejen adecuadamente el diseño de software [3].

Proceso de integración del software, combina las unidades y componentes de software, produciendo elementos software integrados, coherentes con el diseño de software, que demuestren que los requisitos de software funcionales y no funcionales se satisfacen en una plataforma operativa equivalente o completa [3].

Proceso de pruebas de calificación del software, permite confirmar que el producto software integrado logra sus requisitos definidos [3].

B. Calidad del software

La calidad final del producto software está directamente relacionada con la forma en que este se desarrolla y mantiene. Existen herramientas y modelos de evaluación, mejora y estandarización de procesos que han adquirido un papel muy importante en la identificación, integración, medición y optimización de buenas prácticas para el desarrollo y mantenimiento del software, estas permiten evaluar y tomar decisiones sobre la calidad de los procesos que están sujetos a análisis, con el objetivo de establecer una estrategia de mejora. [9]

En el software la calidad se define como “Proceso de software eficaz que se aplica de manera que crea un producto útil que proporciona valor medible a quienes lo producen y a quienes lo utilizan” [6]. Existen modelos desarrollados para evaluar la calidad de los productos software, los que se dividen en dos grupos: 1) Los que se miden directamente (p.e., defectos descubiertos durante una prueba), y 2) Los que se miden indirectamente (p.e., facilidad de uso o de mantenimiento). En cada caso debe presentarse una medición, y se debe comparar el software con algún conjunto de datos y obtener así algún indicio sobre la calidad [6]. En el presente proyecto se analiza el segundo grupo, ya que se pretende evaluar la calidad del producto software en base a sus características intrínsecas.

El Grupo de Trabajo 6 (WG6) evaluación y métricas: es el encargado de la evaluación de productos software a través del establecimiento de métricas para productos y procesos software; propone varias normas enfocadas a la calidad del producto software:

- ISO/IEC 14598: evaluación del producto software.
- ISO/IEC 9126:2001: calidad del producto software.
- ISO/IEC 25000: requisitos y evaluación de calidad del producto software (SQuARE).

Para efectos del presente proyecto se utiliza la norma ISO/IEC 9126:2001, la cual especifica y evalúa características de un producto software de acuerdo al modelo de calidad propuesto y a las métricas externas, internas y en uso, las cuales son respectivamente validadas y aceptadas [10]. Se descompone jerárquicamente en una serie de características y subcaracterísticas que pueden usarse como una lista de comprobación de aspectos relacionados con la calidad [9], y que pueden ser usadas para especificar requisitos de usuario y cliente tanto funcional como no funcional [10]. Se organiza en cuatro partes:

1. *Modelo de calidad*. Define los términos de las características de calidad de software y de cómo estas características son descompuestas en subcaracterísticas
2. *Métricas externas*. Miden el cumplimiento de los sistemas basados en computador que incluye el software.
3. *Métricas internas*. Miden el software en sí.
4. *Métricas de calidad en uso*. Miden los efectos de usar el software en un contexto de uso específico.

A partir del modelo de calidad del producto software definido en [8], se desarrolla un modelo para medir los atributos de calidad en el diseño de ambientes arquitectónicos CC. Para esto se seleccionaron características y subcaracterísticas acordes con entornos SaaS privados y públicos, las cuales deben ser medidas, verificadas y validadas a partir de la aplicación de métricas definidas en [11], [12] y [13] correspondientes a métricas externas, internas y de calidad en uso respectivamente.

C. Cloud Computing

Cloud Computing es definido por NIST [14] como “Un modelo para habilitar el acceso a la red de forma ubicua, conveniente y bajo demanda a un conjunto compartido de recursos informáticos configurables (p.e., redes, servidores, almacenamiento, aplicaciones y servicios) que pueden ser rápidamente suministrados y liberados con un esfuerzo mínimo de gestión o interacción con el proveedor de servicios” [2]. CC está compuesto por cinco características esenciales, cuatro modelos de despliegue y tres modelos de servicio, que juntos categorizan las formas de ofrecer servicios en la nube [15].

c1. Características esenciales

- Auto-servicio bajo demanda: un consumidor puede proveer capacidades de computación, tales como el tiempo del servidor y almacenamiento en red, según sea necesario y de forma automática.
- Acceso amplio a la red: las capacidades están disponibles en la red y son accedidas a través de mecanismos estándar que promueven el uso de plataformas heterogéneas de cliente.
- Puesta en común de recursos: los recursos informáticos del proveedor se combinan para servir a múltiples clientes mediante el modelo multi-inquilinos.
- Rápida elasticidad: las capacidades pueden ser aprovisionadas rápidamente y elásticamente, en algunos casos de forma automática.
- Servicio medido: los sistemas cloud controlan y optimizan automáticamente el uso de recursos

mediante la influencia de una capacidad de medida (pay-per-use) en algún nivel de abstracción adecuado para el tipo de servicio.

c2. Modelos de despliegue

- Cloud privado: la infraestructura cloud es operada exclusivamente por una organización individual compuesta por múltiples clientes
- Cloud comunitario: la infraestructura cloud es compartida por varias organizaciones y es compatible con una comunidad específica que tienen intereses compartidos
- Cloud público: la infraestructura cloud se pone a disposición del público en general o de un grupo industrial grande y es propiedad de una organización que vende servicios en la nube.
- Cloud híbrido: la infraestructura cloud es una composición de dos o más clouds (privado, comunitario o público) que permanecen como entidades únicas, pero son enlazadas a la vez por tecnologías estandarizadas o propietarias que permiten la portabilidad de datos y aplicación

c3. Modelos de servicio

- SaaS (Software como Servicio): provee al consumidor la capacidad para utilizar las aplicaciones cloud proporcionadas por el proveedor, alojadas en una infraestructura y accedidas a través de Internet.
- PaaS (Plataforma como Servicio): provee al consumidor la capacidad de desplegar en la infraestructura cloud sus aplicaciones,
- IaaS (Infraestructura como Servicio): provee al consumidor la capacidad de utilizar procesamiento físico, almacenamiento, redes y otros recursos fundamentales de computación.

c4. Arquitectura de referencia y taxonomía de CC

Para entender los principios de CC es importante desarrollar una Arquitectura de Referencia (RA –por sus siglas en inglés) la cual es definida por NIST como un “Modelo conceptual genérico de alto nivel, considerado como una poderosa herramienta para la discusión de los requisitos, estructuras y operaciones dentro de un ambiente cloud; define un conjunto de actores, actividades y funciones que se pueden utilizar en el proceso de desarrollo arquitectónico, acompañado de una taxonomía para CC. Contiene un conjunto de puntos de vista y descripciones que son la base para la discusión de las características, usos y normas para CC” [15]. La taxonomía asociada establece los elementos centrales de CC y los describe de forma inequívoca, está conformada por cuatro niveles fundamentales: *Nivel 1- roles*: conjunto de obligaciones y comportamientos conceptualizados por los actores asociados en el contexto de CC. [16]. *Nivel 2 -actividades*: comportamientos generales o tareas asociadas a un rol específico. *Nivel 3 - componentes*: procesos específicos, acciones o tareas que se deben realizar para alcanzar el objetivo de una actividad específica. *Nivel 4 - sub componentes*: presenta la parte modular de un componente [15].

A continuación se describe las actividades, componentes y subcomponentes de CC asociados al proveedor cloud [16]:

- Despliegue de servicios: implica todas las actividades y la organización necesaria para hacer que un servicio cloud esté disponible. Una infraestructura cloud puede ser operada en uno de los siguientes modelos de despliegue: cloud público, cloud privado (en el sitio o subcontratado), cloud comunitario (en el sitio o subcontratado) y cloud híbrido
- Orquestación de servicios: disposición, coordinación y gestión de la infraestructura cloud para proporcionar diferentes servicios cloud para satisfacer los requerimientos de negocio y de TI.
 - Gestión de servicios cloud: incluye todas las funciones relacionadas con los servicios que son necesarios para la gestión y operación de los servicios requeridos o propuestos a los consumidores de servicios cloud.
 - Apoyo empresarial: conlleva un conjunto de servicios relacionados con el negocio, clientes y procesos de apoyo.
 - Portabilidad/interoperabilidad: factores cruciales que enfrenta el consumidor en la adopción de servicios cloud relacionados con la portabilidad de datos, interoperabilidad de servicios y portabilidad de sistemas
 - Seguridad: se refiere a la “seguridad de la información” implica protección de la información que incluye accesos no autorizados, uso, divulgación, alteración, modificación o destrucción, con el fin de proporcionar: integridad, confidencialidad y disponibilidad.
 - Privacidad: se refiere a la adecuada, segura y coherente colección, tratamiento, comunicación, uso y disposición de la Información Personal (PI) y la Información de Identificación Personal (PII) a lo largo de su ciclo de vida.

c5. Seguridad perimetral

La seguridad perimetral sirve para organizar y caracterizar los diferentes modelos de despliegue, la cual se basa en los límites de acceso y control a través de un perímetro de seguridad (PS) el cual permite o deniega el acceso a los recursos de computación a través de un controlador de límite que impone una política sobre el acceso. El PS incluye firewalls, guardias y redes privadas virtuales (VPN).

c6. Declaraciones de alcance en el despliegue de servicios.

Existen ciertas declaraciones de alcance en el despliegue de servicios para escenarios cloud privados y públicos [17] como se describe a continuación:

- Dependencia de la red: los consumidores necesitan acceder a los servicios cloud a través de una red segura.
- Ubicaciones de carga de trabajo ocultas a los clientes: para gestionar los recursos de hardware de una nube de manera eficiente.

- Riesgos de multi-alquiler: las cargas de trabajo de los diferentes clientes pueden residir simultáneamente en el mismo sistema y red local,
- Importación/exportación de datos, y limitaciones de rendimiento: debido a que los consumidores accedan a un cloud en una red.
- Seguridad frente a amenazas externas: de acuerdo a las características de cada uno de los modelos de despliegue se implementa un nivel de control de recursos que depende de la implementación de los perímetros de seguridad (PS).
- Costos iniciales para migrar a cloud: dependiendo si los recursos hardware y software para cloud computing están en las instalaciones del proveedor o en las instalaciones del consumidor.
- Elasticidad: si el servicio cloud es lo suficientemente grande y compatible con una diversidad suficiente de cargas de trabajo, puede ser capaz de proporcionar elasticidad a sus clientes.
- Control de recursos y límite de acceso: el control y visibilidad de datos y programas, la posesión física y la capacidad de configurar mecanismos de protección de acceso alrededor de los recursos informáticos.

III. METODOLOGIA

El presente trabajo inicia con el análisis de la norma ISO/IEC 12207:2008 la cual establece un proceso que describe el ciclo de vida del software y las actividades que se ejecutan en la configuración de servicios de un sistema, proceso en el cual se debe considerar diferentes criterios no funcionales descritos en la norma ISO/IEC 9126:2001, que garanticen la calidad del producto y permitan la consolidación del estilo arquitectónico a implementar basados en la norma ISO/IEC/IEEE 42010. Al mismo tiempo se analizan los modelos de servicio (SaaS, PaaS e IaaS) y despliegue (Públicos, Privados, Comunitarios e Híbridos) de una arquitectura Cloud, con la finalidad de determinar las características de cada modelo de servicios y la incidencia de los atributos de calidad en el despliegue de los mismos. Basados en el enfoque de la investigación se desarrolla un modelo de calidad de servicios (QoS) para SaaS en la cual se toma en consideración a la arquitectura de referencia y taxonomía para Cloud Computing definida por NIST [2].

IV. DESCRIPCION DEL MODELO DE CALIDAD DE SERVICIOS

Luego del análisis realizado a los modelos de servicio y despliegue para CC, y a las declaraciones de alcance en el despliegue de servicios para escenarios cloud privados y públicos se determinaron las principales obligaciones entre el proveedor y el consumidor cloud, así como las características clave para entornos SaaS privados y públicos definidos en NIST [2].

De estas características se derivaron los atributos de calidad en base a la Norma ISO/IEC 9126 y se definieron sus métricas correspondientes, obteniendo un modelo de calidad de servicios (QoS) para entornos SaaS privados y públicos (Fig. 1).

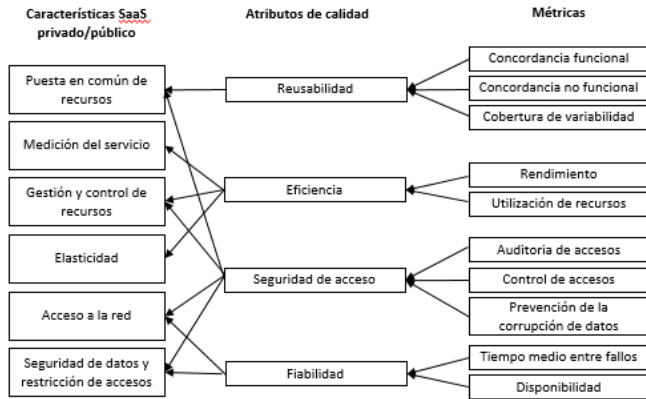


Figura 1. Modelo de calidad de servicio (QoS) para entornos SaaS privados y públicos.

La reusabilidad es un atributo de calidad adicional según la norma ISO/IEC 9126, y está relacionado al diseño. Los atributos de calidad restantes, como eficiencia, fiabilidad y seguridad de acceso están relacionadas al tiempo de ejecución según se menciona en [16].

A continuación se describe cada atributo de calidad y las métricas asociadas para el modelo de calidad (QoS) para entornos SaaS privados y públicos.

Reusabilidad: mide las funcionalidades proporcionadas por los servicios y valida si estas son comunes a los requisitos definidos por los consumidores, se basa en la reutilización de diversos tipos de aplicaciones (servicios) alojadas en la nube y accedidas por múltiples consumidores a través de internet. Las métricas asociadas a la reusabilidad se definen como:

- Concordancia Funcional: mide un promedio de similitud de cada característica funcional definida en un objetivo del servicio cloud. La concordancia de cada característica funcional puede medirse mediante el cálculo del grado de los miembros que utilizan cada característica funcional.

$$CF = \left(\sum_{i=1}^n \frac{nf}{nd} \right) / n \quad (1)$$

CF = Concordancia Funcional

nf = número de requisitos aplicables con característica funcional

nd = número total de requerimientos analizados en el dominio

n = número total de características funcionales

En donde $0 \leq CF \leq 1$, y el valor 1 indica todas las funciones aplicadas en SaaS son comunes en el mismo dominio.

- Concordancia no funcional: mide el promedio de similitud de cada característica funcional. Esta métrica es similar a

la métrica de CF.

$$CNF = \left(\sum_{i=1}^m \frac{nn_f}{nd} \right) / m \quad (2)$$

CNF: Concordancia no funcional

nn_f: número de requisitos aplicables con característica no funcional

nd = número total de requerimientos analizados en el dominio

m = número total de características no funcionales en el dominio.

En donde $0 \leq CNF \leq 1$, y el valor 1 indica que todas las características funcionales son comunes en el mismo dominio.

- Cobertura de variabilidad: mide el número de los puntos de variación incluidos en el dominio que están efectivamente realizadas en SaaS.

$$CV = \left(\frac{nv}{nd} \right) / m \quad (3)$$

CV = Cobertura de Variabilidad

nv = número de puntos de variación realizados en SaaS.

nd = número de puntos de variación en el dominio.

En donde $0 \leq CV \leq 1$, y un valor más alto indica que la cobertura del servicio Cloud es lo suficientemente amplio para un mayor número de puntos de variación.

Eficiencia: mide la cantidad de recursos utilizados para proporcionar la funcionalidad del servicio cloud requerido y el nivel de rendimiento bajo condiciones establecidas. Un modelo de negocio del servicio cloud depende de la cantidad de recursos utilizados o de la cantidad de uso del servicio. Luego se describe métricas asociadas a la eficiencia.

- Comportamiento temporal: debe ser capaz de medir atributos como el comportamiento de un sistema informático incluyendo el software durante las pruebas u operaciones.

$$CT = \frac{t}{ti} \quad (4)$$

CT = Comportamiento temporal

t = tiempo de ejecución, que es (tiempo de espera - tiempo de entrada de comando finalizada)

ti = tiempo total de invocación de servicio.

En donde $0 < t$, y el más cercano a 1 indica que el SaaS tiene una mayor eficiencia de tiempo.

- Rendimiento: mide la cantidad de tareas que puede realizar con éxito durante un período determinado de tiempo.

$$X = \frac{A}{T} \quad (5)$$

A = número de tareas completadas

T = período de tiempo de observación

En donde $0 < X$, y el mayor es el mejor.

Utilización de recursos: mide atributos como el comportamiento de utilización de recursos del servicio cloud que incluye el software durante la fase de pruebas u operación.

$$UR = \frac{ra}{rp} \quad (6)$$

ra = cantidad real de recursos asignados

rp = cantidad de recurso pre-definidos para cada tipo de recursos.

En donde $0 \leq UR$, y el valor más pequeño indica que el servicio Cloud tiene acciones de recursos superiores.

Seguridad de acceso: Mide el número de funciones u ocurrencias con problemas de seguridad en el servicio cloud, tales como: a) fugas en la salida de información, b) pérdida de datos importantes; c) acciones contra operaciones ilegales.

Métricas asociadas a la seguridad de acceso.

- Auditoria de datos: evalúa la cantidad de accesos que el sistema registra en la base de datos históricos de acceso.

$$X = \frac{A}{B} \quad (7)$$

A = Número de usuarios que acceden al sistema y los datos, grabado en la base de datos del historial de acceso.

B = Número de usuarios que acceden al sistema y los datos realizado durante la evaluación.

En donde $0 \leq X \leq 1$, el más cercano a 1 es el mejor.

- Control de acceso: se debe contar el número de operaciones ilegales detectadas en comparación con el número de operaciones ilegales especificadas.

$$X = \frac{A}{B} \quad (8)$$

A = Número de diferentes tipos de operaciones ilegales detectadas.

B = Número de tipos de operaciones ilegales especificadas.

En donde $0 \leq X \leq 1$, el más cercano a 1 es el mejor.

- Prevención de corrupción de datos: se debe contar las apariciones de eventos mayores y menores de corrupción de datos.

$$a) X = 1 - \frac{A}{N} \quad (9)$$

A = Número de veces que se ha producido un evento mayor de corrupción de datos

N = Número de casos de prueba probados que causan eventos de corrupción de datos.

$$b) Y = 1 - \frac{B}{N} \quad (10)$$

B = Número de veces que un evento menor de corrupción de datos ha ocurrido.

$$c) Z = \frac{A}{T} \quad Z = \frac{B}{T} \quad (11)$$

T = período de tiempo de operación (durante las pruebas de funcionamiento).

En donde: $0 \leq X \leq 1$, el más cercano a 1 es el mejor. $0 \leq Y \leq 1$, el más cercano a 1 es el mejor. $0 \leq Z$, el más se acerca a 0 es el mejor.

Fiabilidad: Debe tener la capacidad de medir los atributos relacionados con el comportamiento del sistema y del software de gestión cloud durante las pruebas de ejecución para indicar el grado de fiabilidad del servicio Cloud durante su funcionamiento.

Métricas asociadas a la fiabilidad.

- Disponibilidad: capacidad del servicio Cloud para que esté disponible para su uso durante el período de tiempo especificado para realizar todas las operaciones de los usuarios.

Medir el período de tiempo de reparación cada vez que el sistema no esté disponible durante una prueba.

$$a) X = \frac{T_o}{T_o + T_r} \quad (12)$$

Calcular el tiempo medio de reparación.

$$b) Y = \frac{A1}{A2} \quad (13)$$

To = tiempo de operación

Tr = tiempo de reparación

A1 = total de casos de uso disponibles exitosos de software cuando el usuario utiliza el software.

A2 = número total de casos de intento del usuario para utilizar el servicio Cloud durante el tiempo de observación. En donde: $0 \leq X \leq 1$, el mayor y más cercano a 1 es el mejor, ya que el usuario puede utilizar el servicio Cloud para obtener más tiempo. $0 \leq Y \leq 1$, el mayor y más cercano a 1 es el mejor.

Tiempo medio entre fallos: con qué frecuencia el servicio Cloud falla en la operación. Contar el número de fallos ocurridos durante un periodo definido de operación y calcular el intervalo promedio entre fallos.

$$a) X = T1 / A \quad (14)$$

$$b) Y = T2 / A \quad (15)$$

T1 = tiempo de operación

T2 = suma de intervalos de tiempo entre ocurrencias de fallo consecutivo.

A = número total de fallos detectados actualmente (fallos ocurridos durante el tiempo de operación observado).

En donde $0 < X, Y$, el más grande es el mejor. El tiempo más largo puede ser previsto entre fallos.

V. CONCLUSIONES

La computación en la nube obliga a cambios en las normas, las capacidades de infraestructura; la definición de una adecuada arquitectura de software basada en la utilización de un SDLC específico, es de gran importancia al momento de evaluar el despliegue de servicios alojados en una nube. Esto nos permite cerrar la brecha entre la aplicación y la infraestructura al proporcionar los medios para que tanto proveedores como consumidores de servicios cloud puedan evaluar la calidad de los mismos.

SaaS provee un esquema eficaz de reutilización de servicios, con beneficios para los consumidores ya que los servicios de actualización son libres de costos de compra y mantenimiento/actualizaciones, lo que implica el cumplimiento de compromisos de calidad que los proveedores deben garantizar en la provisión del servicio. La evaluación de la calidad de SaaS se convierte en la actividad más importante para una gestión exitosa de este modelo de servicios.

En base al modelo de calidad SaaS propuesto, los proveedores de servicios podrán evaluar sus servicios y el retorno de la inversión, y, los consumidores de servicios pueden consultar los resultados de evaluación de calidad para descubrir, suscribirse y utilizar SaaS. Además, los resultados de la evaluación se podrán utilizar como un indicador para la gestión de la calidad SaaS

REFERENCIAS

- [1] 19011:2002b ISO, "Guidelines for quality and/or environmental management systems auditing.," no. 2, 2002.
- [2] P. Mell and T. Grance, "The NIST Definition of Cloud Computing. NIST Special Publication 800-145. Available in:," 2011, http://csrc.nist.gov/publications/drafts/800-145/Draft-SP-800-145_cloud-definition.pdf.
- [3] 12207:2008(E) ISO/IEC/IEEE, "Systems and software engineering—Software life cycle processes. ".
- [4] ISO, "International Organization for Standardization," 2008.
- [5] IEEE Computer Society, "SWEBOK: Guide to the Software Engineering Body of Knowledge.," 2004.
- [6] R. S. Pressman, *Ingeniería de Software: un enfoque práctico* , 7th ed.: México: McGraw-Hill/Interamericana., 2010.
- [7] F. Buschmann, "Pattern-Oriented Software Architecture: A System of Patterns," 1996.
- [8] M., García, F., & Caballero, I. Piattini, "Calidad de sistemas informáticos. ," 2010.
- [9] Jae Yoo Lee, Jung Woo Lee, Du Wan Cheun, and Soo Dong Kim, "A Quality Model for Evaluating Software-as-a-Service," in *7th. ACIS International Conference Engineering Research, Management and Applications*, Hikou, 2009, pp. 261-266.
- [10] L., Grance, T., Patt-Corner, R., & Voas, J. Badger, "Cloud Computing Synopsis and Recommendations. ," *NIST Special Publication 800-146.*, 2012, Available in: <http://csrc.nist.gov/publications/drafts/800-146/Draft-NIST-SP800-146.pdf>.
- [11] L., Clements, P., & Kazman, R. Bass, "Software Architecture in Practice ," 2003.
- [12] R., Messina, J., Liu, F., Tong, J., & Mao, J. Bohn, "NIST Cloud Computing Reference Architecture," *EEE World Congress on Services. Maryland, USA.*, 2011.
- [13] M. & Sokol, A. Hogan, "NIST Cloud Computing Standards Roadmap. ," *NIST Special Publication 500-291 (v2).* , 2013, Available in: http://collaborate.nist.gov/twiki-cloud-computing/pub/CloudComputing/StandardsRoadmap/NIST_SP_500-291_Version_2_2013_June18_FINAL.pdf.
- [14] 9126-1:2000 ISO/IEC, "Information technology—Software product quality—Part 1: Quality model."
- [15] 9126-2:2002 ISO/IEC, "Software Engineering—Product Quality—Part 2: External metrics."
- [16] 9126-3:2002 ISO/IEC, "Software Engineering—Product Quality—Part 3: Internal metrics."
- [17] 9126-4:2001 ISO/IEC, "Software Engineering—Software product quality—Part 4: Quality in use metrics. ".
- [18] NIST, "National Institute of Standards and Technology.," 2010, Available in: <http://www.nist.gov/>.