

Universidad de las Ciencias Informáticas

Facultad 4



Título: Propuesta de métricas de calidad para el desarrollo de aplicaciones en la UCI con una Arquitectura Orientada a Servicios

Trabajo de Diploma para optar por el título de Ingeniero Informático

Autores:

- Frankis Peña Valera
- Lisandra Rodríguez Ferrer

Tutor:

- Ing. Irina Napal Torres

Junio 2008

DECLARACIÓN DE AUTORÍA

Declaramos ser autores de la presente tesis y reconocemos a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo.

Para que así conste firmo la presente a los ____ días del mes de _____ del año _____.

Para que así conste firmo la presente a los ____ días del mes de _____ del año _____.

Frankis Peña Valera

Firma del Autor

Lisandra Rodríguez Ferrer

Firma del Autor

Ing. Irina Napal Torres

Firma del Tutor

Cuando pueda medir lo que está diciendo y expresarlo con números, ya conoce algo sobre ello; cuando no pueda medir, cuando no pueda expresar lo que dice con números, su conocimiento es precario y deficiente: puede ser el comienzo del conocimiento, pero en sus pensamientos, apenas está avanzando hacia el escenario de la ciencia.

Lord Kelvin

DATOS DE CONTACTO

Ing. Irina Napal Torres.

- **Especialidad de graduación:** Ingeniería Informática.
- **Categoría docente:** Instructor.
- **Años de experiencia en el tema de Calidad:** 4.
- **Años de graduado:** 4.

AGRADECIMIENTOS COMPARTIDOS

Agradecerle a nuestro Comandante en Jefe Fidel Castro Ruz por haber tenido la grandiosa idea de crear la Universidad de las Ciencias Informáticas, la cual es una excelente casa de estudios que nos ha hecho ser mejores personas tanto en lo profesional como en lo personal.

Al profesor Manuel Alejandro Gil Martín por sus consejos, su orientación, su gran ayuda y su paciencia.

A nuestra tutora Irina Napal Torres por su ayuda y atención.

A todos los profesores de nuestra universidad ya que directa o indirectamente siempre nos dan lo mejor de cada uno, para que seamos mejores hombres y mujeres en el futuro.

A nuestros amigos porque gracias a ellos sabemos lo que es la amistad verdadera, valor importante en nuestras vidas. Gracias por estar siempre, por aconsejarnos, regañarnos, guiarnos, compartir alegrías y tristezas en todo este tiempo.

A todos nuestros familiares y vecinos que cada día están presentes y nos dan su ayuda incondicional.

A nuestros amigos de la universidad por permitirnos conocerlos y ser parte de su vida. Por ayudarnos y estar con nosotros a lo largo de la carrera.

A todas aquellas personas que ya no están con nosotros y que nos ayudaron tanto.

AGRADECIMIENTOS

A mis padres y mis hermanas por confiar en mí por todo su amor y dedicación. A Alberto y José Manuel por querer tanto a mis hermanas que con eso yo soy feliz también. A Yerandi por todo el cariño en estos 5 años, por toda su preocupación y su ayuda. A Yunet por ser mi amiga, por preocuparse por mis cosas y por ayudarme tanto.

A todos Gracias

Lisandra

A mi familia por estar siempre a mi lado, a mi madre la persona más especial y que más quiero, gracias por darme la posibilidad de venir a este mundo y vivirlo intensamente, a mis amigos que siempre han sabido aconsejarme y desearme lo mejor...

Frankis

DEDICATORIA LISANDRA

A mi mamá Aracelis Ferrer y mi papá José Antonio Rodríguez por ser las personas más maravillosas del mundo, por ser el principal motivo por el cual cada mañana me levanto con deseos de ser mejor, por todo su amor, cariño, dedicación y por quererme tanto.

A mis hermanas Lien y Liset por ser las hermanas más lindas del mundo, por quererme tanto, por todo su amor, por comprenderme y ayudarme a ser mejor.

A mis abuelos que me han dado siempre su amor y cariño.

A todos mis amigos que siempre han creído en mí y me han querido incondicionalmente.

Y a la Revolución por darme la posibilidad de cumplir mis sueños.

Lisandra

DEDICATORIA FRANKIS

A mi mamá Margarita por ser una madre especial que ha sabido enfrentar derrotas y victorias a mi lado, apoyándome en todo momento y queriéndome siempre, a mi abuelo Pedro Rolando quien cada día me ha demostrado el camino correcto a seguir, a mi abuela Paula por consentirme y darme su cariño incondicional , a mi abuelo Romarico por apoyarme siempre, a mis tíos Alberto y Yoleisi hombres de bien y buena voluntad que han inculcado en mi valores imprescindibles, a mi tía Magalis por apoyarme, a Alejandro por ser alguien con una mentalidad muy positiva, a mi tía Mercedes por ser además mi amiga, gracias por tu confianza y respeto, a mi padrastro Juan Mario quien me ha enseñado muchas cosas en la vida basándose en el respeto que me ha tenido y enseñado, a mi primo Geosvani por ser merecedor de mi confianza, a Odelta quien nunca podrá leer estas líneas pero la llevo muy presente en mi alma y corazón, gracias por enseñarme tu madurez adquirida en la vida, a mi bisabuela Eladia por todo su amor y cariño entregado a mí, a todos mis amigos, en fin a todos aquellos que han ayudado a lograr que sea la persona que soy, gracias a todos por ser tan especiales para mí.

Frankis

RESUMEN

La investigación propone un sistema de métricas dirigidas a proporcionar indicadores que brinden una visión de la calidad del proceso de desarrollo de la Arquitectura Orientada a Servicios en la UCI. Para definir las métricas propuestas se realiza un estudio de las métricas de calidad, de los antecedentes y las tendencias actuales de la Arquitectura Orientada a Servicios, de las métricas de calidad que existen actualmente para medir dicha arquitectura y del estado en el que se encuentra el desarrollo de la Arquitectura Orientada a Servicios en la Universidad. Se realiza un análisis de las métricas que proponen QoS, SLA, KPI y BAM. Para realizar la validación de las métricas propuestas se utiliza la herramienta SoapUI, y dicha validación fue realizada a los servicios que existen actualmente en la Universidad y que conforman la arquitectura hasta el momento y en específico al servicio Registro de Personal. Además se propone una guía para la aplicación de las mismas.

PALABRAS CLAVE:

Métricas de calidad, Arquitectura Orientada a Servicios.

Índice

AGRADECIMIENTOS COMPARTIDOS	II
AGRADECIMIENTOS	V
DEDICATORIA LISANDRA	VI
DEDICATORIA FRANKIS	VII
RESUMEN	VIII
INTRODUCCIÓN	1
CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA	6
1.1 Introducción del Capítulo	6
1.2 Calidad del Software	6
1.2.1 Factores de calidad.....	8
1.2.2 Aseguramiento de la calidad del software.....	9
1.2.3 CMMI.....	11
1.3 Métricas de Software	13
1.3.1 Definiciones	14
1.3.2 Tipos de métricas de software.....	15
1.4 Métricas de Calidad	17
1.5 Arquitectura Orientada a Servicios	18
1.5.1 Conceptos básicos.....	19
1.5.1.1 XML	19
1.5.1.2 Esquemas.....	19
1.5.1.3 Servicios Web.....	20
1.5.1.4 UDDI.....	20
1.5.1.5 ESB	21
1.5.1.6 BPM.....	21
1.5.1.7 Modelo de Madurez SOA.....	22
1.5.1.8 QoS	23
1.5.1.9 SLA.....	23
1.5.1.10 KPI.....	24
1.5.1.11BAM.....	25
1.6 Estado del arte	26
1.7 Conclusiones del capítulo	29
CAPÍTULO 2: SOLUCIÓN PROPUESTA	30
2.1 Introducción del capítulo	30
2.2 Estudio de las métricas de calidad existentes actualmente	30

2.2.1 Métricas para los servicios Web.....	32
2.2.2 Métricas de QoS	34
2.2.3 Métricas de SLA.....	35
2.2.4 Métricas de KPI.....	36
2.2.5 Métricas de BAM.....	36
2.3 Análisis de la Arquitectura Orientada a Servicios en la UCI.....	36
2.4 Definición de las Métricas	39
2.4.1 Estructura del sistema de Métricas	40
2.4.2 Guía de aplicación de la propuesta.....	51
2.5 Conclusiones del capítulo	54
CAPÍTULO 3: VALIDACIÓN DE LA SOLUCIÓN PROPUESTA	55
3.1 Introducción del capítulo	55
3.2 Validación de métricas	55
3.3 Herramienta SoapUI	56
3.4 Las métricas de calidad en los servicios de la UCI	57
3.5 Servicio Registro de Personal.....	57
3.6 Calidad en el servicio Registro de Personal (RP)	58
3.7 Conclusiones del capítulo	65
CONCLUSIONES	66
RECOMENDACIONES.....	67
BIBLIOGRAFÍA.....	68
ANEXOS.....	71
GLOSARIO	75

INTRODUCCIÓN

En enero del 2000 se crea el Ministerio de Informática y las Comunicaciones (MIC) con el objetivo de crear un organismo que una las tecnologías: la Informática y las Comunicaciones, dado precisamente por la convergencia tecnológica que éstas tienen y el impacto de la integración de ambas en gran parte de toda la actividad humana y como elemento de suma importancia en el proceso de informatización de la sociedad cubana y su repercusión en todos los sectores de la economía y la sociedad.

En el año 2002 fue creada la Universidad de las Ciencias Informáticas (UCI) en un inicio como una unidad presupuestada adscripta al MIC hoy en día adscripta al Consejo de Estado (CE), surge como idea del Comandante en Jefe Fidel Castro Ruz, en sus inicios denominada "Proyecto Futuro". Es la primera universidad cubana creada bajo el calor de la Batalla de Ideas, con dos objetivos fundamentales: informatizar el país y exportar software para el desarrollo económico del mismo.

La Industria Cubana del Software (ICSW) está llamada a convertirse en una significativa fuente de ingresos para el país, como resultado del correcto aprovechamiento de las ventajas del alto capital humano disponible.

En la década de los 90 y con la aparición de Internet y su posterior masificación a niveles jamás pensados, ha existido siempre la necesidad e inquietud por parte de las empresas desarrolladoras de software de buscar o contar con la manera de lograr la integración entre sistemas heterogéneos. Para tal efecto muchas compañías fueron creando de forma individual la mejor manera de lograr esta integración. Muchas empresas comenzaron una agitada carrera para generar la mejor tecnología integradora de sistemas, pero a medida que la competencia se hacía cada vez más fuerte, la integración se hacía cada vez más difícil.

Las empresas se percataron que era imposible crear una plataforma integrada de forma individual, así que decidieron que era mejor buscar un lenguaje común de intercambio de información aprovechando los estándares existentes en el mercado.

Introducción

Bajo este contexto nacen los Servicios Web basados en XML.

En el actual entorno económico, la cooperación electrónica entre empresas supone una ventaja competitiva. Los servicios Web ayudan a resolver los problemas de integración por varios motivos. Primero y más importante, porque no es una tecnología invasiva y, por tanto, su uso no implica tener que desterrar tecnologías anteriores, sino ofrecerles nuevas posibilidades de cooperación. Segundo, porque los servicios Web se apoyan en un conjunto de lenguajes y protocolos estándares en Internet aceptados universalmente. Esto garantiza, que sea cual sea la plataforma o el fabricante que la proporciona, los objetos de negocio pueden trabajar conjuntamente. Tercero, porque se trata de una tecnología abierta a la incorporación de nuevos avances en materia de seguridad y confidencialidad que ayuda a integrar los sistemas actuales y venideros.

La denominada tecnología de servicios Web aparece como un nuevo paradigma que se difunde de manera increíblemente rápida. Quiere decir que la llamada programación orientada a objetos se ha ido desplazando hacia la programación orientada a servicios. Donde los objetos son componentes fundamentales, pero al final, el diseño de la aplicación tiene que ver con el diseño del servicio. Significa que no se diseña una página Web, sino un sitio que permite un número determinado de servicios de acuerdo con las necesidades de los usuarios de Internet. Según este nuevo paradigma, se desarrollan componentes modulares de software, que son reusables de manera ubicua en Internet.

Con el desarrollo de la red se ha hablado mucho de la ubicuidad en el sentido que un mismo servicio o una misma persona pueden estar «virtualmente» al mismo tiempo en muchos lugares. Por ejemplo, la información diaria de Cuba que ofrece Cubasí. Este servicio puede ofrecerse, al mismo tiempo, en una página Web; o utilizando otros servicios Web, o con una «Palm», o a través de un teléfono móvil. Y todo ese conjunto es un mismo servicio, diversificado por un sistema de aplicaciones. Esto implica una nueva perspectiva del servicio, una universalización que técnicamente ya es posible gracias al servicio Web, que estandariza, que supone una dirección universal en Internet, invocable por otro programa.

Los servicios Web son los componentes principales que se utilizan para construir una Arquitectura Orientada a Servicios (SOA por sus siglas en inglés).

SOA surge debido a que las grandes empresas actuales se enfrentan a problemas cuando su crecimiento en términos de procesos de negocio y de sistemas informáticos que los soportan crece exponencialmente, ante la necesidad de dar respuesta a las necesidades del mercado, y la premura

Introducción

con que estas respuestas deben de ser dadas. Esto obliga a las empresas a que constantemente estén cambiando dichos procesos de negocio, y por consiguiente las aplicaciones que los soportan.

El estado actual de la Tecnología de la Información (TI) en cualquier empresa mediana o grande es de caos en cuanto a hardware, Sistema Operativo (SO), plataformas, y lenguajes de programación que soportan las aplicaciones, debido a la evolución de los mismos y a la demanda de nuevas soluciones a nuevos procesos. Esto trae consigo que dichas aplicaciones no se comuniquen entre sí ni que evolucionen en conjunto. Se puede encontrar una aplicación en Cobol junto a una en C# invisibles una de la otra.

Es en este escenario donde SOA hace su entrada como una propuesta pensada desde hace muchos años atrás, pero imposibilitada desde el punto de vista tecnológico.

Aunque los Servicios Web no necesariamente significan SOA, y no todas las SOA están basadas en Servicios Web , la relación entre las dos tendencias es importante, y se potencian mutuamente: el interés por los servicios Web lleva hacia SOA, y las ventajas de SOA ayudan a que las iniciativas de los servicios Web tengan éxito. (Gartner, 2007)

SOA es el nuevo estándar que permite soportar los requerimientos tecnológicos actuales y futuros de las empresas que estén decididas a prosperar en la nueva economía. Proporciona un puente entre TI y procesos de negocio ayudando a incrementar la flexibilidad de los negocios y la capacidad de las empresas para cumplir más rápida, fácil y económicamente las metas de negocio. Dependiendo de las prioridades de su industria, una SOA puede ayudar a reducir el tiempo y los costos de entrega de nuevos servicios.

Cuba principalmente mediante la UCI quiere tener factores tecnológicos relacionados con la maduración de una Arquitectura Orientada a Servicios, abierta; la maduración de los sistemas llamados de software libre, software abierto y la propia consolidación tecnológica de Internet. Se trata de interoperabilidad, de arquitectura basada en servicios, más descentralizado. Es decir, infinidad de personas comunicándose de uno a muchos, como es el caso de los blogs, que permiten que las propias personas actualicen y produzcan la información.

La UCI como universidad-empresa se ha insertado en este proceso de cambio de Internet enfocados a los servicios Web, uno de los que se benefician con el estilo arquitectónico **SOA**, pero aún sin la

Introducción

experiencia previa que tiene nuestro país, por lo que necesita para asegurarse de que sus aplicaciones sean eficientes, seguras e íntegras, definir métricas para medir la calidad de las aplicaciones desarrolladas en una Arquitectura Orientada a Servicios.

En la mayoría de los desafíos técnicos, las métricas ayudan a entender tanto el proceso técnico que se utiliza para desarrollar un producto, como el propio producto.

Es por ello que surge esta investigación donde se plantea el **problema**:

¿Cómo medir la calidad de las aplicaciones desarrolladas en una Arquitectura Orientada a Servicios en la UCI?

Como solución se debe investigar y valorar si se pueden usar las métricas existentes o reformularlas para utilizarlas en las aplicaciones desarrolladas con Arquitectura Orientada a Servicios, por lo que se define como **objeto de estudio** *las aplicaciones de la UCI desarrolladas en una Arquitectura Orientada a Servicios.*

Como **campo de acción** se define: *las métricas de calidad para el desarrollo de aplicaciones desarrolladas en una Arquitectura Orientada a Servicios en la UCI.*

El **objetivo general** de esta investigación es: *proponer métricas de calidad para el desarrollo de aplicaciones desarrolladas con una Arquitectura Orientada a Servicios en la UCI.*

Objetivos específicos:

- Realizar un análisis detallado de las métricas de calidad.
- Realizar un análisis detallado de la Arquitectura Orientada a Servicios.
- Realizar un análisis de uso de las métricas de calidad en una Arquitectura Orientada a Servicios.
- Definir métricas de calidad para el desarrollo de las aplicaciones desarrolladas con una Arquitectura Orientada a Servicios en la UCI
- Definir la estructura de validación de las métricas propuestas.
- Valorar los impactos de la aplicación de las métricas propuestas y emitir recomendaciones al respecto.

Introducción

Para dar cumplimiento a los objetivos mostrados anteriormente se establecen las siguientes **tareas**:

- Analizar las métricas de calidad que existen actualmente.
- Analizar los antecedentes y las tendencias actuales de la Arquitectura Orientada a Servicios en el mundo.
- Realizar un estudio de las métricas de calidad que existen en la actualidad para las aplicaciones desarrolladas con Arquitectura Orientada a Servicios.
- Describir el proceso de desarrollo de aplicaciones con Arquitectura Orientada a Servicios en la UCI.
- Definir la estructura de las métricas.
- Definir una guía de aplicación de las métricas propuestas.
- Estudiar las vías de validación de las métricas propuestas.

Este trabajo consta de introducción, tres capítulos y conclusiones.

El Capítulo 1: **Fundamentación Teórica** aborda los temas relacionados con la calidad de software, métricas, conceptos, tipos y reflexiones que ayudan al entendimiento de la importancia de su aplicación en los productos de la UCI desarrollados con una Arquitectura Orientada a Servicios.

El Capítulo 2: **Solución Propuesta**, en este capítulo se realiza un estudio de las métricas que existen actualmente para medir SOA y del proceso de desarrollo de SOA en la Universidad donde se definirán las que serán aplicadas haciendo las adaptaciones necesarias y se desarrollará una guía para la aplicación de las métricas propuestas.

El Capítulo 3: **Validación de la Solución Propuesta**, en este capítulo se le aplican las métricas propuestas a un Servicio Web desarrollado en la Universidad y se determina según los resultados obtenidos si se corresponden o no con lo esperado.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

1.1 Introducción del Capítulo

En este capítulo se fundamenta el uso de las métricas de calidad para el desarrollo de aplicaciones en la UCI con una Arquitectura Orientada a Servicios como una de las metas más importantes para obtener software de alta calidad. Se realiza un estudio de las métricas existentes en el mundo con el objetivo de lograr mayor calidad en el software, dentro de ellas las métricas de calidad las que se analizan para adaptarlas a una Arquitectura Orientada a Servicios que se quiere desarrollar en la Universidad de las Ciencias Informáticas.

1.2 Calidad del Software

La calidad del software es el conjunto de cualidades que lo caracterizan y que determinan su utilidad y existencia. La calidad es sinónimo de eficiencia, flexibilidad, corrección, confiabilidad, mantenibilidad, portabilidad, usabilidad, seguridad e integridad. (Heidi González Doria, 2001)

Es la concordancia con los requisitos funcionales y de rendimiento explícitamente establecidos, con los estándares de desarrollo explícitamente documentados y con las características implícitas que se espera de todo software desarrollado profesionalmente. (Pressman, 2005)

Un software de alta calidad debe tener:

➤ Utilidad:

Determina si el sistema hace lo que sus usuarios esperan que haga

El sistema es útil si:

Su funcionalidad ayuda al usuario en la realización de sus actividades

➤ Confiabilidad:

Baja probabilidad de falla o bajo número de errores

Capítulo 1: Fundamentación Teórica

➤ Mantenibilidad:

¿Es flexible, es modificable y es internamente comprensible?

➤ Eficiencia:

Medida en relación al uso de los recursos del computador

Ejemplo: bajo tiempo de ejecución y mínimo uso del espacio de almacenamiento

➤ Facilidad de prueba:

¿Se puede probar con facilidad?

En qué medida el diseño y codificación facilitan las pruebas.

A la hora de definir la calidad del software se debe diferenciar entre la calidad del producto de software y la calidad del proceso de desarrollo. No obstante, las metas que se establezcan para la calidad del producto van a determinar las metas a establecer para la calidad del proceso de desarrollo, ya que la calidad del producto va a estar en función de la calidad del proceso de desarrollo. Sin un buen proceso de desarrollo es casi imposible obtener un buen producto.

La calidad del software debe ser considerada en todos sus estados de evolución (especificaciones, diseño, código, etc.). No basta con tener en cuenta la calidad del producto una vez finalizado, cuando los problemas de mala calidad ya no tienen solución o la solución es muy costosa. (Angel Cervera Paz)

Uno de los principales problemas a los que nos enfrentamos a la hora de hablar de la calidad del software es el siguiente: ¿Es realmente posible encontrar un conjunto de propiedades en un software que nos den una indicación de su calidad? Para dar respuesta a esta pregunta aparecen los Modelos de Calidad. En los Modelos de Calidad, la calidad se define de forma jerárquica y tienen como objetivo resolver la complejidad mediante la descomposición. Por lo general en el nivel más alto de esta jerarquía se encuentran los factores de calidad.

Capítulo 1: Fundamentación Teórica

Los Factores de calidad representan la calidad desde el punto de vista del usuario y son las características que componen la calidad. También se denominan Atributos de Calidad Externos.

1.2.1 Factores de calidad

A pesar del avance en el desarrollo de software y las tecnologías, con el paso de los años los atributos que proporcionan una indicación de la calidad del software siguen siendo los mismos, en este sentido es inevitable mencionar el trabajo desarrollado por McCall y Cavano en cuanto a la definición de factores de calidad, pues a pesar del tiempo, sus estudios han sido una guía para otros modelos y normas de calidad. La norma ISO 9126 es un ejemplo de ello, muchas características y subcaracterísticas definidas en la misma hacen referencia a la operación, transición y revisión del software y señalan entre otras cosas la necesidad de lograr que el software opere correctamente y con el grado de exactitud requerido, que los usuarios sean capaces de entenderlo y usarlo, es decir que sea amigable con quienes interactúen con él, que sea capaz de responder correctamente ante fallos o cambios del entorno y que proporcione una ejecución o desempeño apropiado, teniendo en cuenta los recursos utilizados. (Ludisley la Torre , Mariela Cepero)

El estándar identifica seis atributos clave de calidad: (Pressman, 2005)

Funcionalidad. El grado en que el software satisface las necesidades indicadas por los siguientes subatributos: idoneidad, corrección, interoperatividad, conformidad y seguridad.

Confiabilidad. Cantidad de tiempo que el software esta disponible para su uso. Este referido por los siguientes subatributos: madurez, tolerancia a fallos y facilidad de recuperación.

Usabilidad. Grado en que el software es fácil de usar. Viene reflejado por los siguientes subatributos: facilidad de comprensión, facilidad de aprendizaje y operatividad.

Eficiencia. Grado en que el software hace óptimo el uso de los recursos del sistema. Está indicado por los siguientes subatributos: tiempo de uso y recursos utilizados.

Capítulo 1: Fundamentación Teórica

Facilidad de mantenimiento. La facilidad con que una modificación puede ser realizada. Está indicada por los siguientes subatributos: facilidad de análisis, facilidad de cambio, estabilidad y facilidad de prueba.

Portabilidad. La facilidad con que el software puede ser llevado de un entorno a otro. Está referido por los siguientes subatributos: facilidad de instalación, facilidad de ajuste, facilidad de adaptación al cambio.

Estos factores no necesariamente son utilizados para medidas directas. En cualquier caso, facilitan una valiosa base para medidas indirectas y una excelente lista para determinar la calidad de un sistema.

El objetivo primordial de la ingeniería del software es producir un sistema, aplicación o producto de alta calidad. Para lograr este objetivo, los ingenieros de software deben emplear métodos efectivos junto con herramientas modernas dentro del contexto de un proceso maduro de desarrollo del software. Partiendo entonces de la necesidad de tener una alta calidad del software se asume que es más rentable prevenir los fallos de calidad que corregirlos o lamentarlos, y se incorpora el concepto de la "prevención" a la Gestión de la Calidad, que se desarrolla sobre esta nueva idea en las empresas industriales, bajo la denominación de Aseguramiento de la Calidad.

1.2.2 Aseguramiento de la calidad del software

El Aseguramiento de la Calidad consiste en tener y seguir un conjunto de acciones planificadas y sistemáticas. Estas acciones deben ser demostrables para proporcionar la confianza adecuada de que se cumplen los requisitos del Sistema de la Calidad. Este aseguramiento se diseña para cada aplicación antes de comenzar a desarrollarla y no después.

El Aseguramiento de la Calidad aborda principalmente tres áreas o técnicas:

- Métricas del software: para el control del proyecto
- Verificación y validación: a lo largo del ciclo de vida del software, incluyendo pruebas y procesos de revisión.

Capítulo 1: Fundamentación Teórica

- Gestión de la configuración del software

Este Aseguramiento de la Calidad se realiza a través de modelos. Entre los que se encuentran:

- Modelo de Boehm: modelo de descomposición de características de calidad del software en tres niveles antes de aplicar las métricas: usos principales, componentes intermedios y componentes primitivos.
- Modelo factores/criterios/métricas: similar a Boehm, en el que se ha introducido mayor grado de descomposición en cada nivel.
- Marco ISO 9126: denominado Evaluación de Productos Software: Características de calidad y guías para su uso, la calidad se descompone en seis factores.
- Paradigma GQM (Goal-Question-Metric): enfoque de medición para evaluar la calidad del software basado en la identificación de objetivos a lograr.
- Modelo de Gilb: creación de una especificación de requisitos de calidad para cada proyecto que deben escribir conjuntamente el usuario y el analista.
- Modelo CMM (Capability Maturity Model): modelo de capacidad de madurez del SEI (Software Engineering Institute).
- Modelo SPICE: (Software Process Improvement and Capability Determination): modelo de valoración de la arquitectura que define los procesos y prácticas aconsejables.

El Aseguramiento de la Calidad se refiere a validar los procesos usados para crear los productos. Es una herramienta especialmente útil para administradores y patrocinadores, ya que permite discutir los procesos usados para crear los productos para determinar si son razonables. Este aseguramiento tiene asociado dos constitutivos diferentes: los Ingenieros de Software que realizan el trabajo técnico y un grupo de SQA (Software Quality Assurance) que tiene la responsabilidad de la planificación de aseguramiento de la calidad, supervisión, mantenimiento de registros, análisis e informes.

Las Actividades del SQA son:

- Establecimiento de un plan de SQA para un proyecto.
- Participación en el desarrollo de la descripción del proceso de software del proyecto.

Capítulo 1: Fundamentación Teórica

- Revisión de las actividades de Ingeniería del Software para verificar su ajuste al proceso de software definido.
- Auditoria de los productos de software designados para verificar el ajuste con los definidos como parte del proceso del software.
- Asegurar que las desviaciones del trabajo y los productos del software se documentan y se manejan de acuerdo con un procedimiento establecido.
- Registrar lo que no se ajuste a los requisitos e informar a sus superiores.

Además de estas actividades, el grupo de SQA coordina el control y la gestión de cambios y ayuda a recopilar y analizar las métricas del software.

Las métricas son escalas de unidades sobre las cuales puede medirse un atributo cuantificable. Cuando se habla de software nos referimos a la disciplina de recopilar y analizar datos basándonos en mediciones reales de software, así como a las escalas de medición. Los atributos son características observables del producto o del proceso de software, que proporciona alguna información útil sobre el estado del producto o sobre el progreso del proyecto. El término producto se utiliza para referirse a las especificaciones, a los diseños y a los listados del código. Los valores de las métricas no se obtienen sólo por mediciones. Algunos valores de métricas se derivan de los requisitos del cliente o de los usuarios y, por lo tanto, actúan como restricciones dentro del proyecto.

Para adaptar las directrices marcadas por los sistemas de calidad a cada proyecto particular hay que generar un plan específico de calidad, es decir un plan de aseguramiento de la calidad.

El mundo del software ha creado sus propias líneas de trabajo en la gestión de la calidad, trabaja sobre los procesos de producción como medio para asegurar la calidad del software. Por ejemplo, el SEI (Software Engineering Institute) propone un modelo de clasificación y mejora de los procesos empleados por las organizaciones de software denominado CMMI (Capability Maturity Model Integration).

1.2.3 CMMI

Capability Maturity Model Integration (CMMI) es un modelo para la mejora de procesos que proporciona a las organizaciones los elementos esenciales para procesos eficaces. Las mejores

Capítulo 1: Fundamentación Teórica

prácticas CMMI se publican en los documentos llamados modelos. En la actualidad hay dos áreas de interés cubiertas por los modelos de CMMI: Desarrollo y Adquisición.

Independientemente del modelo que opta una organización, las practicas CMMI deben adaptarse a cada organización en función de sus objetivos de negocio.

Los 6 niveles definidos en CMMI para medir la capacidad de los procesos son:

Nivel 0: Incompleto: El proceso no se realiza, o no se consiguen sus objetivos.

Nivel 1: Ejecutado: El proceso se ejecuta y se logra su objetivo.

Nivel 2: Gestionado: Además de ejecutarse, el proceso se planifica, se revisa y se evalúa para comprobar que cumple los requisitos.

Nivel 3: Definido: Además de ser un proceso gestionado se ajusta a la política de procesos que existe en la organización, alineada con las directivas de la empresa.

Nivel 4: Cuantitativamente gestionado: Además de ser un proceso definido se controla utilizando técnicas cuantitativas.

Nivel 5: Optimizante: Además de ser un proceso cuantitativamente gestionado, de forma sistemática se revisa y modifica o cambia para adaptarlo a los objetivos del negocio. Mejora continua.

CMMI Incorpora un área de proceso denominada "Medición y Análisis", cuyo objetivo es desarrollar y establecer una capacidad de medición que se pueda usar para dar soporte a las necesidades de información de la organización y que proporcionen resultados objetivos que sean útiles para la toma de decisiones y acciones correctivas.

Actualmente CMMI resulta muy útil a la hora de definir métricas, ya que a pesar de que este modelo no define exactamente métricas, es muy útil para determinar que es lo que interesa medir, al final las métricas concretas dependen de la empresa.

1.3 Métricas de Software

“... Empleamos medidas para entender mejor los atributos de los modelos que creamos. Pero, fundamentalmente, empleamos las medidas para valorar la calidad de los productos de ingeniería o de los sistemas que construimos”. (Pressman, 2005)

Todas las organizaciones de software exitosas implementan mediciones como parte de sus actividades cotidianas pues estas brindan la información objetiva necesaria para la toma de decisiones y que tendrá un impacto efectivo en el negocio y desempeño en la ingeniería.

Para poder asegurar que un proceso o sus productos resultantes son de calidad o poder compararlos, es necesario asignar valores, descriptores, indicadores o algún otro mecanismo mediante el cual se pueda llevar a cabo dicha comparación.

Se definen las métricas de software como la aplicación continua de mediciones basadas en técnicas para el proceso de desarrollo del software y sus productos, para suministrar información relevante a tiempo, así el administrador junto con el empleo de estas técnicas mejorará el proceso y sus productos (Heidi González Doria, 2001).

Para una definición más completa debe incluirse los servicios relacionados al software como la respuesta a los resultados del cliente. Ver Figura 1.1.

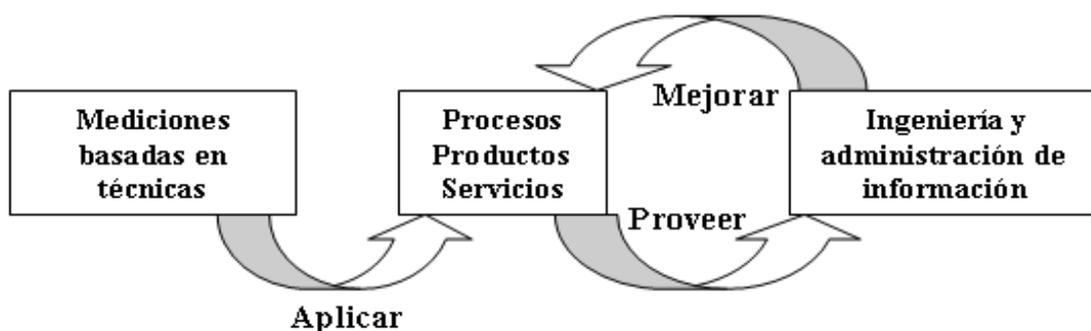


Figura 1.1. Concepto de Métricas.

Capítulo 1: Fundamentación Teórica

Las métricas de software pueden desempeñar varias funciones:

- Las métricas pueden ayudarnos a entender más acerca de nuestros productos, procesos y servicios de software.
- Las métricas pueden ser empleadas para evaluar el software de nuestros productos, procesos y servicios con respecto a los estándares y metas establecidas.
- Las métricas pueden proveer la información que nosotros necesitamos para controlar recursos y procesos utilizados en la producción de nuestro software.
- Las métricas pueden ser usadas para predecir los atributos de las entidades de software en el futuro.

La medición es esencial, si es que se desea realmente conseguir la calidad del software, para entender mejor las métricas del software es necesario tener claro algunas definiciones como que es una medida, medición, métrica e indicadores.

1.3.1 Definiciones

Medida. Proporciona una indicación cuantitativa de extensión, cantidad, dimensiones, capacidad y tamaño de algunos atributos de un proceso o producto. Pueden ser directas, por ejemplo número de líneas de código, número de errores encontrados, etc., o pueden ser indirectas, como funcionalidad, calidad y complejidad. (Navarro, 2003)

Medición. Acto de determinar una medida. (Navarro, 2003)

Métrica. Es una medida cuantitativa del grado en que un sistema o proceso posee un atributo dado. (Navarro, 2003)

Indicador. Es una métrica o combinación de métricas que proporcionan una visión del proceso, del proyecto o del software en sí, y poder hacer ajustes para que las cosas mejoren. (Navarro, 2003)

La clasificación de las métricas de software refleja o describe la conducta del software. Estas clasificaciones de métricas fortalecen la idea, de que más de una métrica puede ser deseable para

Capítulo 1: Fundamentación Teórica

valorar la complejidad y la calidad del software. A continuación se muestra una breve clasificación de métricas de Software.

1.3.2 Tipos de métricas de software

Existen diferentes **tipos de métricas de software** entre las que se encuentran:

Métricas técnicas: Se centra en las características del software no en cómo se obtiene.

Métricas de productividad: Se centran en el rendimiento del proceso de la ingeniería del software.

Métricas de calidad: Proporcionan una indicación de cómo se ajusta el software a los requisitos implícitos y explícitos del cliente. Se centran en la funcionalidad o utilidad del programa.

Métricas orientadas a la persona: Proporcionan medidas e información sobre la forma que se desarrolla el software de computadoras y sobre todo el punto de vista humano de la efectividad de las herramientas y métodos.

Métricas orientadas al tamaño: Indican en que tiempo se termina el software y cuantas personas se necesitan. Son medidas directas al software y el proceso por el cual se desarrolla.

Métricas orientadas a la función. Son medidas indirectas del software y del proceso por el cual se desarrolla.

La medición es algo común en el mundo de la ingeniería no siendo así en el caso de la ingeniería de software, puesto que a pesar de no ser un tema novedoso no se ha tratado en los últimos tiempos correctamente. (Pressman, 2005)

Las organizaciones no habían comprendido la necesidad de realizar medidas para obtener métricas hasta la aparición del mercado competitivo y demás, aún así, existen organizaciones que la utilizan incorrectamente. Esto se debe a que existen problemas a la hora de ponerse de acuerdo en aspectos como qué se debe medir y como hacerlo.

Capítulo 1: Fundamentación Teórica

Las medidas se pueden clasificar en:

Medidas Directas. En el proceso de ingeniería se encuentran el coste, el esfuerzo aplicado, las líneas de código producidas, velocidad de ejecución, el tamaño de memoria y los defectos observados en un determinado periodo de tiempo.

Medidas Indirectas. Se encuentran la funcionalidad, calidad, complejidad, eficiencia, fiabilidad, facilidad de mantenimiento, etc.

El primer paso de la medición es identificar los atributos o entidades a medir. Estos pueden ser de tres tipos:

Procesos: atributos de actividades relacionadas con el software.

Productos: componentes, entregas o documentos resultantes de una actividad de proceso.

Recursos: entidades requeridas por una actividad de proceso. Dentro de cada clase anterior se puede distinguir:

Atributos internos: Son aquellos que pueden ser medidos examinando el proceso, producto o recurso mismo.

Atributos externos: Se miden en cuanto a como el proceso, producto o recurso se relaciona con su entorno.

Las métricas del proceso se obtienen a partir de las métricas del proyecto. Las métricas no se pueden usar para comparar personas o equipos. Para la mejora del proceso las métricas deben aplicarse desde el nivel individual. Las métricas de un producto que son privadas para un individuo del equipo de desarrollo, se combinan para desarrollar métricas del proyecto que sean públicas para dicho equipo y estas a su vez se consolidan para crear métricas del proceso.

Para desarrollar software eficientes, seguros e íntegros, es necesario llevar a cabo una serie de pasos que determinan el proceso a seguir, una parte importante para lograr aplicaciones con alta calidad es aplicando las métricas de calidad.

Las métricas de calidad son usadas para demostrar niveles existentes de calidad y para monitorizar los esfuerzos continuos de mejoramiento en el proceso de desarrollo del software.

1.4 Métricas de Calidad

Métricas de calidad: Son todas las métricas de software que definen de una u otra forma la calidad del software; Tales como exactitud, estructuración o modularidad, pruebas, mantenimiento, reusabilidad, cohesión del módulo, acoplamiento del módulo, etc. Estas son los puntos críticos en el diseño, codificación, pruebas y mantenimiento. (Heidi González Doria, 2001)

Aunque hay muchas medidas de la calidad de software, la corrección, facilidad de mantenimiento, integridad, y facilidad de uso proporcionan indicadores útiles para el equipo del proyecto. (Heidi González Doria, 2001)

Se han determinado definiciones y medidas para cada uno de ellos.

Corrección. Un programa debe operar correctamente o proporcionará poco valor a sus usuarios. La corrección es el grado en el que el software lleva a cabo su función requerida.

Facilidad de mantenimiento. El mantenimiento del software cuenta con más esfuerzo que cualquier otra actividad de ingeniería del software. La facilidad de mantenimiento es la facilidad con la que se puede corregir un programa si se encuentra un error, se puede adaptar si su entorno cambia, o mejorar si el cliente desea un cambio de requisitos.

Integridad. Este atributo mide la capacidad de un sistema para resistir ataques (tanto accidentales como intencionados) contra su seguridad. El ataque se puede realizar en cualquiera de los tres componentes del software: programas, datos y documentos.

Facilidad de uso. Si un programa no es amigable con el usuario, frecuentemente esta abocado al fracaso, incluso aunque las funciones que realice sean valiosas. La facilidad de uso es un intento de cuantificar lo amigable que puede ser con el usuario.

Últimamente estamos escuchando palabras esotéricas y misteriosas como SOA, BPM, UDDI o ESB. Aunque no lo parezca se esta en una pequeña revolución, cambio de paradigma, o salto cualitativo. Junto a ello métricas para evaluar la eficiencia y calidad en una SOA, en esta investigación se realizará un análisis de los distintos componentes de la Arquitectura Orientada a Servicios, con el

Capítulo 1: Fundamentación Teórica

objetivo de incrementar los conocimientos y experiencia para analizar y escoger las distintas métricas a proponer.

1.5 Arquitectura Orientada a Servicios

La Arquitectura Orientada a Servicios es una tendencia creciente que intenta reconciliar la visión técnica y de negocios, basándose en estándares abiertos y promoviendo la interoperabilidad entre diversas organizaciones y plataformas de manera eficiente y flexible a los cambios.

Compartir los procesos y la información es importante para las instituciones, pues permite agilizar procesos y trámites, evita duplicar la información y tareas; además de que propicia la innovación en el desarrollo de aplicaciones, incluso en la creación de nuevas herramientas que no fueron previstas o esperadas.

Las arquitecturas SOA utilizan la información existente y que un sistema gestiona de manera aceptable; los desarrollos se centran en la atención de necesidades específicas: consumir información que otros sistemas expongan y, a su vez, exponer y publicar servicios para ser consumidos por otras aplicaciones.

La arquitectura de sistemas SOA surge ante la búsqueda de estrategias de interoperabilidad de sistemas, donde la problemática puede resumirse en los siguientes puntos: (Mario García, Saidel López, 2007)

- Distintos sistemas casi siempre se desarrollan en diversos lenguajes de programación. La interoperabilidad se ve limitada a sistemas implementados en un mismo lenguaje, y/o largos tiempos de desarrollo de sistemas para lograr la comunicación.
- La interoperabilidad se ve afectada por las medidas de seguridad impuestas por los firewalls que limitan el paso de información hacia diferentes puertos informáticos.
- Es imposible que todos los equipos de cómputo compartan las mismas especificaciones.

En una Arquitectura Orientada a Servicios se tratan conceptos como XML, Esquemas, Servicios Web, UDDI, ESB, BPM, Modelo de Madurez, QoS, SLA, KPI y BAM.

1.5.1 Conceptos básicos

1.5.1.1 XML

XML, siglas en inglés de **Extensible Markup Language** (lenguaje de marcas extensible), es un metalenguaje extensible de etiquetas que permite definir la gramática de lenguajes específicos. XML no es realmente un lenguaje en particular, sino una manera de definir lenguajes para diferentes necesidades. Algunos de estos lenguajes que usan XML para su definición son XHTML, SVG, MathML.

XML se propone como un estándar para el intercambio de información estructurada entre diferentes plataformas. Se puede usar en bases de datos, editores de texto, hojas de cálculo y casi cualquier cosa imaginable.

Es una tecnología sencilla que tiene a su alrededor otras que la complementan y la hacen mucho más grande y con posibilidades mucho mayores. Tiene un papel muy importante en la actualidad ya que permite la compatibilidad entre sistemas para compartir la información de una manera segura, fiable y fácil.

Un documento XML puede no tener un DTD, cada documento define sus propias etiquetas de manera informal.

Un documento XML puede ser válido y bien formado, el estar bien formado no implica que sea válido, no ocurre lo contrario. Para que un documento XML sea bien formado debe seguir reglas básicas de sintaxis para XML. Para que sea válido, debe estar bien formado, hacer referencia a un DTD o un Schéma XML y ser conforme a la gramática definida en dicho esquema.

1.5.1.2 Esquemas

XML Schema es un lenguaje de esquema utilizado para describir la estructura y las restricciones de los contenidos de los documentos XML de una forma precisa, más allá de las normas sintácticas impuestas por el propio lenguaje XML. Se logra así, una percepción del tipo de documento con un nivel alto de abstracción. Fue desarrollado por el World Wide Web Consortium (W3C) y alcanzó el nivel de recomendación en mayo del 2001.

1.5.1.3 Servicios Web

Un **Servicio Web** (en inglés Web service) es una colección de protocolos y estándares que sirven para intercambiar datos entre aplicaciones. Aplicaciones de software desarrolladas en lenguajes de programación diferentes, y ejecutadas sobre cualquier plataforma, pueden utilizar los servicios Web para intercambiar datos.

La interoperabilidad se consigue mediante la adopción de estándares abiertos.

Ventajas de los servicios Web

- Aportan interoperabilidad entre aplicaciones de software independientemente de sus propiedades o de las plataformas sobre las que se encuentran instaladas.
- Los servicios Web fomentan los estándares y protocolos basados en texto, que hacen más fácil acceder a su contenido y entender su funcionamiento.
- Al apoyarse en HTTP, los servicios Web pueden aprovecharse de los sistemas de seguridad firewall sin necesidad de cambiar las reglas de filtrado.
- Permiten que servicios y software de diferentes compañías ubicadas en diferentes lugares geográficos puedan ser combinados fácilmente para proveer servicios integrados.
- Permiten la interoperabilidad entre plataformas de distintos fabricantes por medio de protocolos estándar.

1.5.1.4 UDDI

UDDI son las siglas del catálogo de negocios de Internet denominado Universal Description, Discovery and Integration. UDDI es una iniciativa industrial abierta centrada en el contexto de los servicios Web. El registro en el catálogo se hace en XML y consta de tres partes:

- Páginas blancas: dirección, contacto y otros identificadores conocidos.
- Páginas amarillas: categorización industrial basada en taxonomías.
- Páginas verdes: información técnica sobre los servicios que aportan las propias empresas.

UDDI es uno de los estándares básicos de los servicios Web cuyo objetivo es ser accedido por los mensajes SOAP y dar paso a documentos SERVICIO WEBDL, en los que se describen los requisitos

Capítulo 1: Fundamentación Teórica

del protocolo y los formatos del mensaje solicitado para interactuar con los servicios Web del catálogo de registros.

1.5.1.5 ESB

Un bus de servicios empresariales (**ESB**) es una solución de integración distribuida, basada en los mensajes y en estándares abiertos. La función de un ESB es proporcionar una comunicación fiable entre los distintos recursos tecnológicos tales como aplicaciones, plataformas y servicios, que están distribuidos en múltiples sistemas por toda la empresa.

Los ESB se están convirtiendo en un paso clave para el establecimiento de una SOA empresarial. Constituyen los cimientos de una SOA y pueden complementarse con capacidades de productividad adicionales, como la orquestación de servicios y los registros.

1.5.1.6 BPM

Se llama **Business Process Management** a la metodología empresarial cuyo objetivo es mejorar la eficiencia a través de la gestión sistemática de los procesos de negocio (BPR), que se deben modelar, automatizar, integrar, monitorear y optimizar de forma continua.

Como su nombre sugiere, Business Process Management (BPM) se enfoca en la administración de los procesos del negocio.

A través del modelado de las actividades y procesos puede lograrse un mejor entendimiento del negocio y muchas veces esto presenta la oportunidad de mejorarlos.

La automatización de los procesos reduce errores, asegurando que los mismos se comporten siempre de la misma manera y dando elementos que permitan visualizar el estado de los mismos. La administración de los procesos permite asegurar que los mismos se ejecuten eficientemente, y la obtención de información que luego puede ser usada para mejorarlos. Es a través de la información que se obtiene de la ejecución diaria de los procesos, que se puede identificar posibles ineficiencias en los mismos, y actuar sobre las mismas para optimizarlos.

Capítulo 1: Fundamentación Teórica

Para soportar esta estrategia es necesario contar con un conjunto de herramientas que den el soporte necesario para cumplir con el ciclo de vida de BPM. Este conjunto de herramientas son llamadas Business Process Management System y con ellas se construyen aplicaciones BPM.

Existen diversos motivos que mueven la gestión de Procesos de Negocio (BPM), entre los cuales se encuentran:

- Extensión del programa institucional de calidad
- Cumplimiento de legislaciones
- Crear nuevos y mejores procesos
- Entender qué se está haciendo bien o mal a través de la comprensión de los procesos
- Documentar procesos para outsourcing y definición de SLA (Service Level Agreement)
- Automatización de procesos
- Crear y mantener las cadenas de valor

1.5.1.7 Modelo de Madurez SOA

Permite medir el estado actual de una empresa con respecto a la utilización de SOA: Proyectos, Arquitectura, Infraestructura, Gobierno y define una ruta de evolución. El modelo esta compuesto por 5 niveles que tienen una relación directa con los niveles de CMM donde el principal objetivo es medir la calidad del desarrollo del software.

Nivel 1: Servicios SOA iniciales.

Propósito: Introducir funcionalidad y nueva tecnología.

Nivel 2: Servicios SOA arquitecturados.

Propósito: Reducción de costos y control de TI.

Nivel 3 (a): Servicios de negocio SOA.

Propósito: Agilidad del negocio.

Nivel 3 (b): Servicios de colaboración SOA.

Propósito: Colaboración con asociados de negocio y mercado en general.

Capítulo 1: Fundamentación Teórica

Nivel 4: Servicios SOA de métricas del negocio.

Propósito: Encontrar medidas de desempeño del negocio.

Nivel 5: Servicios de Negocio Optimizados SOA.

Propósito: Optimización del negocio, reaccionar y responder automáticamente.

1.5.1.8 QoS

Calidad de Servicio (**Quality of Service, QoS**) son las tecnologías que garantizan la transmisión de cierta cantidad de datos en un tiempo dado, es la capacidad de dar un buen servicio.

QoS para aplicaciones que deben comunicarse en tiempo real, es el conjunto de características cualitativas y cuantitativas de un sistema multimedia distribuido, que son necesarias para lograr la funcionalidad requerida en una aplicación. En este sentido, QoS determina la usabilidad y utilidad del servicio, influenciando en la popularidad del mismo.

Con la extensa propagación de los servicios Web, QoS se ha convertido en un factor significativo para determinar el éxito que pueda alcanzar un proveedor de servicio sobre otros. Cubre un rango de técnicas que armoniza las necesidades de quien requiere el servicio con quien lo provee, basándose en los recursos de red disponibles.

A través de QoS se puede hacer referencia a las propiedades no funcionales del servicio Web, tales como rendimiento, fiabilidad, disponibilidad y seguridad, las cuales representan los requerimientos de calidad necesarios para alcanzar la funcionalidad esperada del mismo.

1.5.1.9 SLA

EL modelo de Acuerdo de Nivel de Servicios (Service Level Agreement, SLA) consiste en un contrato en el que se estipulan los niveles de un servicio en función de una serie de parámetros y objetivos, establecidos de mutuo acuerdo entre ambas partes, así, refleja contractualmente el nivel operativo de funcionamiento, penalizaciones por caída de servicio y limitación de responsabilidad por no servicio.

Este modelo no ha de estar relacionado necesariamente con la contratación de servicios a terceras partes, sino que puede implantarse a nivel interno, transformando una determinada unidad de negocio en centro de servicios que provea a la propia compañía.

Capítulo 1: Fundamentación Teórica

Para implantar con éxito un SLA han de tenerse en cuenta una serie de factores claves, de los que va a depender en gran medida la obtención de los resultados deseados:

Aspectos críticos

La definición de procedimientos estándares y los mecanismos de evaluación y seguimiento.

En la implantación de un SLA se deben seguir una serie de puntos

- Definición de Objetivos: mejora de la eficacia, reducción de costos, formalización de la relación
- Identificar expectativas: qué es lo que espera la organización de este acuerdo
- Adecuada planificación temporal
- Optimización/rediseño de procesos: revisar los procesos si el SLA no asegura ningún cambio o como mínimo formalizarlos.

Errores más frecuentes en la implantación

- Definir niveles de servicio inalcanzables
- Regulación excesiva
- Error en la definición de prioridades
- Complejidad técnica
- Irrelevancia (si un SLA no tiene ningún efecto sobre el cliente, el objetivo no tiene sentido).

1.5.1.10 KPI

Indicadores clave de rendimiento (Key Performance Indicators **KPI**), miden el nivel del desempeño de un proceso, enfocándose e indicando a los procesos, de forma que se pueda alcanzar el objetivo fijado.

Es necesario para una organización al menos identificar sus KPI. El ambiente clave para identificar los KPI son:

Capítulo 1: Fundamentación Teórica

- Tener un proceso del negocio pre -definido.
- Tener claros los requerimientos de las metas y desempeño del negocio.
- Tener mediciones cualitativas y cuantitativas de los resultados y compararlos con el conjunto de metas
- Investigar la variabilidad y debilidades del proceso o los recursos para lograr las metas en el corto plazo.

Los KPI difieren según la naturaleza de una organización y de su estrategia. Ayudan a la organización a medir el progreso hacia el logro de sus metas, especialmente hacia la dificultad de cuantificar el conocimiento basado en actividades.

1.5.1.11BAM

BAM (**Business Activity Monitoring**) es una estrategia de negocios que permite integrar diferentes fuentes de información, para generar alarmas en tiempo real dirigidas a personas que toman las decisiones adecuadas sobre procesos claves del negocio con la información suficiente y relevante para tomar acción sobre ellos. BAM le da la capacidad de acceder a sus procesos críticos de negocio y mediante indicadores controlar la eficiencia y la velocidad de sus operaciones. (Mind, 2006)

Beneficios obtenidos con las soluciones BAM

El uso de soluciones BAM en una entidad permite:

- Obtener información en tiempo real acerca del desarrollo de la actividad del negocio, que posibilita la rápida detección de problemas, permitiendo la identificación de su causa y el análisis de su impacto.
- Automatización de los procesos de control manual, dando lugar a la disminución de los costes/riesgos operacionales y el aumento de la eficiencia.
- Monitorización de los procesos/actividades empresariales en relación a los compromisos adquiridos en el negocio y expectativas del cliente.
- Aumento de la satisfacción y fidelidad del cliente a través de la gestión pro-activa.
- Permite la integración de una variedad de sistemas.

1.6 Estado del arte

SOA es un estilo arquitectónico que ha evolucionado en los últimos años. Existen 5 niveles de madurez que definen el grado de desarrollo de una SOA en una institución, los conceptos más actuales, están relacionados con los niveles 4 y 5.

Así, tenemos términos como SERVICIO WEB-BPEL, SERVICIO WEB-CDL, BPEL4PEOPLE, BAM y otros, que definen tecnologías de procesos y cuestiones de un nivel de abstracción mayor, por ejemplo, actualmente se profundizan los conocimientos en la aplicación de conceptos de la web semántica a SOA.

Es evidente que para que las empresas sean ágiles tienen que disponer de una tecnología fundamental, capaz de tomar decisiones. Este es la motivación esencial para introducir el “procesamiento semántico” a las TI.

El procesamiento semántico, expresa las relaciones entre conceptos representadas en sentencias, es tema favorito de investigación entre académicos y en el sector tecnológico. El constante deseo de que el software pueda inferir información a partir de un contexto es uno de los objetivos de la inteligencia artificial. El análisis semántico se ha estudiado en el contexto de procesamiento de lenguajes naturales. Reconocimiento de texto y voz (traducción de lenguajes, por ejemplo). Correlación y extracción de datos. Análisis de tendencias (para el almacenamiento de datos, la detección de amenazas, etc.). Búsquedas temáticas. Búsquedas refinadas y consultas que incentivan la consciencia del contexto empresarial.

De igual modo, se han propuesto Arquitecturas Semánticas Orientadas a Servicios (SSOA, por su acrónimo en inglés). SSOA introduce mejoras semánticas a servicios, de forma que un agente consciente del modelo semántico puede combinar servicios de forma dinámica para satisfacer los objetivos empresariales, utilizando **enterprise service-oriented architecture** (SOA empresarial), se pueden crear nuevas aplicaciones que se sumen a las soluciones existentes, lo que permitirá incrementar el valor de los sistemas actuales y automatizar nuevos procesos.

SOA empresarial es un plan adaptable, flexible y abierto a las Tecnologías de la Información (TI), que permite soluciones del negocio a nivel de empresa.

Capítulo 1: Fundamentación Teórica

SERVICIO WEB-BPEL (**Web Services-Business Process Execution Language**) o especificación del lenguaje de ejecución de procesos de negocio en servicios Web, pertenece a la capa de componentes establecida en SOA y es, en conjunto con SERVICIO WEB-CDL, una alternativa para la implementación y manejo de Servicios Web. SERVICIO WEB-BPEL surge como necesidad de ser el integrador o el engranaje para las diversas tecnologías que funcionan bajo SOA, pero que no logran una interoperabilidad al 100%, lo que restringe su funcionamiento y adaptabilidad.

SERVICIO WEB-BPEL nace tomando propiedades de dos lenguajes propietarios con características diferentes pero complementarias, estos lenguajes son: SERVICIO WEBFL (Web Services Flow Language), el cual se basa en gráficos de actividades y conectores de control para definir un modelo global describiendo las interacciones entre los servicios Web existentes y permitiendo la definición de nuevos servicios a partir de los existentes. El otro lenguaje es XLANG desarrollado por Microsoft, el cual basa su funcionamiento en una notación orientada al comportamiento de intercambio de mensajes a través de los servicios Web participantes, de esta manera automatiza el proceso de negocios. SERVICIO WEB-BPEL combina el estilo orientado en gráficos de SERVICIO WEBFL y el estilo algebraico propuesto por XLANG.

SERVICIO WEB-CDL (**Web Services-Choreography Description Language**) Este es un lenguaje utilizado para la definición de servicios dentro de la plataforma SOA, basado en XML y cuyo objetivo es la descripción del comportamiento de cada uno de los servicios establecidos para lograr un objetivo común.

Lo que permite en esencia SERVICIO WEB-CDL es la definición de las restricciones de ordenamiento (secuencias, dependencias, etc.), y el establecimiento de reglas y comportamientos que permitan la colaboración adecuada entre los servicios. Así, este lenguaje permite la descripción sin ambigüedades de las colaboraciones establecidas entre servicios, determinando un protocolo de negociación. De esta manera, cada organización puede desarrollar de manera independiente su propio papel en la colaboración siempre y cuando se respete el contrato global para que de esta manera se garantice la interoperabilidad.

SERVICIO WEB-CDL es importante dentro de SOA porque es una tecnología escalable, garantiza la interoperabilidad efectiva y segura de servicios, permite tener servicios más robustos reduciendo el tiempo de implementación de los mismos.

Capítulo 1: Fundamentación Teórica

BPEL4People es una especificación BPEL que fue creada para modelar a las personas dentro de la definición de procesos BPEL. Específicamente, podrá visualizarse una actividad denominada "Actividad de Persona"; igual que una estándar de invocación de Servicio Web; pero con un contexto adicional asociado. Describe como el lenguaje BPEL puede ser extendido para involucrar interacción de usuarios dentro de procesos de negocio.

A pesar de todo lo que se esta haciendo para el desarrollo de SOA todavía existen algunas dificultades en su desarrollo entre las que se encuentran la falta de habilidad o entrenamiento y la definición de métricas. SLA, KPI, y BAM definen algunas métricas que se estudiarán para su posible aplicación en la Universidad.

El Service Level Agreement (SLA) define métricas exactas de servicio, calidad y tiempos de respuesta. Un SLA debe contener un nivel de servicio específico, opciones de soporte, penalidades por servicios no brindados, garantía del nivel de rendimiento, nivel específico del nivel de soporte del cliente y que software o hardware va a ser incluido como parte del servicio.

Los KPI constituyen un conjunto de métricas enfocadas en medir aquellos aspectos de rendimiento organizacional que impactan en mayor forma en el éxito actual y futuro de la empresa. En cualquier organización sólo podemos hallar pocos KPI (generalmente no más de 10) que cumplan con ciertas características. KPI son generalmente métricas de acción hacia el futuro, en contraposición de métricas que expresan el resultado ya pasado, es decir históricas.

Monitoreo de las Actividades del Negocio (BAM) brinda acceso en tiempo real a los indicadores de desempeño críticos de la empresa, con el objetivo de mejorar la velocidad y efectividad de sus operaciones y procesos. Se trata de una solución que apunta específicamente a que los encargados de gestión dispongan de información para tomar mejores decisiones y puedan ser alertados automáticamente ante cualquier problemática que ponga en riesgo al negocio. (CIENTEC)

Una de los pasos para implementar BAM es desarrollando métricas e indicadores donde se trata de crear formas que se puedan utilizar para señalar problemas y cómo los datos serán recopilados. Se debe especificar las reglas para validar, correlacionar y analizar estas métricas. También hay que determinar dónde y cómo se consultarán a otros recursos para adquirir información de contexto adicional.

Capítulo 1: Fundamentación Teórica

Existe la necesidad de emplear métricas para evaluar la eficiencia y calidad en una SOA en la UCI, este trabajo tiene el objetivo de definir las métricas para lograr una Arquitectura Orientada a Servicios de alta calidad en la Universidad.

1.7 Conclusiones del capítulo

Este capítulo de manera general mostró la existencia de las mediciones que se aplican en el proceso del software con el intento de mejorarlo sobre una base continua. Las métricas constituyen un elemento fundamental para llevar a cabo dicho proceso organizado y controlado.

SOA es una arquitectura de software que permite la creación y/o cambios de los procesos de negocio desde la perspectiva de TI de forma ágil, a través de la composición de nuevos procesos utilizando las funcionalidades de negocio que están contenidas en la infraestructura de aplicaciones actuales o futuras, pero como todo proceso, se necesita aplicarle métricas para medir su calidad, lo que se dificulta debido a ser SOA tan novedosa.

QoS, SLA, KPI y BAM representan algo de lo que se está aplicando para medir calidad en SOA que se estudiará más profundamente en el próximo capítulo para su posible aplicación en la Universidad.

CAPÍTULO 2: SOLUCIÓN PROPUESTA

2.1 Introducción del capítulo

En este capítulo se realizará un análisis de las métricas de calidad existentes de las que se seleccionarán las que se ajusten al desarrollo de SOA en la Universidad. Se realizará un estudio de QoS, SLA, KPI y BAM para su posible aplicación de acuerdo a las especificaciones de SOA en la UCI y se desarrollará una guía para la aplicación de las métricas propuestas.

2.2 Estudio de las métricas de calidad existentes actualmente

Hoy día es cada vez más frecuente la consideración de métricas de software, es por eso que se están implantando en la actualidad. El uso de éstas se ha adoptado con éxito en el amplio mercado de desarrollo de software introduciendo reconocimientos y consideraciones por parte de administradores y usuarios, y estableciendo la necesidad de un enfoque más disciplinado y de una alta calidad. (Heidi González Doria, 2001)

A medida que el proyecto progresa siempre se debe valorar la calidad. Aunque se pueden recopilar muchas medidas de calidad, el primer objetivo es medir errores y defectos. Las métricas que provienen de éstas medidas proporcionan una indicación de la efectividad de las actividades de control y de la garantía de calidad en grupos o en particulares.

Existen muchas métricas de calidad orientadas a las líneas de código, y es un método muy utilizado por muchas organizaciones a nivel mundial, sin embargo resulta un poco engorroso, pues contar cada línea de código en las organizaciones que produzcan software, tiende a ser mal interpretada como pérdida de tiempo, o los programadores se sentirían amenazados con el control de su trabajo, calculando que cantidad de líneas de código realizan en determinado período de tiempo.

Además las líneas de código dependen del lenguaje de programación que se utilice, su aplicación no permite realizar estimaciones. (Giraldo)

Capítulo 2: Solución Propuesta

Además de estas, existen muchas más que no dependen de las líneas de código, más bien del funcionamiento, que no fueron pensadas para ser aplicadas en una Arquitectura Orientada a Servicios, pero en dependencia de lo que queramos medir se pueden o no utilizar.

En la Universidad se han definido métricas de calidad de las que en dependencia de lo que se quiere medir en la Arquitectura Orientada a Servicios se utilizarán. Se han desglosado en 6 categorías fundamentales de acuerdo a los atributos a tener en cuenta a la hora de medir la calidad de un producto, funcionalidad, confiabilidad, usabilidad, eficiencia, mantenibilidad y portabilidad definiendo para cada una de ellas un conjunto de subcaracterísticas, y de cada uno de ellas un conjunto de métricas.

Funcionalidad

- Idoneidad
- Precisión
- Interoperabilidad
- Conformidad de la funcionalidad

Confiabilidad

- Madurez
- Tolerancia ante fallos
- Recuperabilidad
- Conformidad con la confiabilidad

Usabilidad

- Comprensibilidad
- Atracción
- Conformidad con la usabilidad
- Afectividad
- Eficiencia
- Periodo productivo

Eficiencia

- Rendimiento

Capítulo 2: Solución Propuesta

- Utilización de recursos
- Conformidad de la eficiencia

Mantenibilidad

- Diagnósticabilidad
- Capacidad de análisis de fallos
- Flexibilidad
- Conformidad de la mantenibilidad

Portabilidad

- Instalabilidad
- Conformidad de la portabilidad

Cada una de estas categorías se centra en diferentes aspectos de calidad a tener en cuenta a la hora de definir las métricas.

La mayoría de las definiciones de SOA identifican la utilización de servicios Web en su implementación, no obstante se puede implementar una SOA utilizando cualquier tecnología basada en servicios, en teoría una SOA se podría implementar sin servicios Web, pero siendo realistas cualquier proyecto SOA realizado en los últimos años está compuesto por ellos. Es más, la mayoría de los clientes empiezan a practicar con SOA creando un Servicio Web, muchas veces sobre lógica de negocio ya existente. (Gartner, 2007)

Actualmente se han definido métricas para medir la calidad de estos servicios Web.

2.2.1 Métricas para los servicios Web

Poseer un Servicio Web de calidad es de importancia para una organización ya que se benefician las relaciones de negocio contribuyendo tanto a la integración interna como la inter-empresarial; al asegurar una infraestructura integrada, segura, escalable y disponible se disminuye el costo de propiedad y permite compartir información de manera confiable.

Capítulo 2: Solución Propuesta

La calidad del servicio se puede entender como una combinación entre el nivel de servicio (probabilidad de satisfacer un requerimiento en un cierto período de tiempo) y ciertos atributos del servicio entregado. (MORENO, 2006)

Estimar el nivel de servicio de un servicio Web, dado un tiempo exigido de respuesta, es una labor bastante sencilla. Sin embargo, controlar el nivel de servicio puede ser una tarea bastante complicada debido a la gran cantidad de factores que lo determinan. (MORENO, 2006)

Factores que pueden afectar el nivel de servicio:

- Una implementación subóptima de uno o más servicios Web.
 - ❖ Un algoritmo ineficiente puede restar rendimiento a un servicio.
 - ❖ Malas prácticas de programación podrían incidir en un uso excesivo de tiempo y otros recursos de un sistema.
- La sobrecarga del servidor, a causa de una gran demanda de sus servicios Web, podría reducir el rendimiento de cada servicio.
 - ❖ Hay servicios que son demasiado demandados.
 - ❖ Dependencias ineficientes; un servicio Web podría estar siendo solicitado de manera redundante.
- Hay servicios que son cruciales; si su rendimiento decae, gran parte de la red también lo hará.
- Vaivenes de la conexión.
- Un servidor ha dejado de funcionar, y así sus servicios.
 - ❖ El servidor se ha dañado debido a una falla externa, por ejemplo, debido a la subida de la tensión eléctrica o a un corte de suministro eléctrico.
 - ❖ Sabotaje (ataque por parte de sombreros negros).
- Un servicio Web podría tornarse lento a causa de algún servicio del cual dependa.

Capítulo 2: Solución Propuesta

- Un servidor tiene una potencia insuficiente; está obsoleto.

Si un servicio Web no está siendo lo suficientemente rápido o confiable, entonces se hace necesario determinar las medidas a tomar. Existen métricas destinadas a determinar sobre qué servicios Web destinar los esfuerzos para mejorar el nivel de servicio.

Existen también métricas estructurales que se basan en la estructura de la red de dependencias para medir distintos aspectos de los servicios Web (o de la red misma). Estas métricas son algo estáticas debido a su naturaleza: se basan en la forma de la red. Por ende, no necesitan ser actualizadas frecuentemente a menos que la estructura de la red cambie muy seguido. Tampoco consideran llamadas condicionales o múltiples entre servicios Web. Sólo consideran la forma de la red.

Las métricas de funcionamiento son otras de las métricas aplicadas para determinar la calidad de los servicios Web, estas miden aspectos de funcionamiento. Este tipo de métricas debería ser actualizado regularmente.

QoS, Calidad de servicio se refiere a la calidad del servicio, es un nombre genérico para un conjunto de algoritmos que tratan de proporcionar diferentes niveles de calidad para los diferentes tipos de tráfico de la red, con la implantación de este es posible ofrecer más garantía y seguridad para las aplicaciones avanzadas, una vez que el tráfico de estas aplicaciones pasa a tener prioridad en relación con aplicaciones tradicionales. QoS propone un sistema de evaluación de la calidad a través de métricas.

2.2.2 Métricas de QoS

En QoS la utilización de métricas adecuadas para diversos tipos de servicios pueden mejorar el rendimiento y la disponibilidad de los mismos.

QoS en su forma más sencilla se define como el mecanismo que satisface los requerimientos de las aplicaciones en una red, es decir, la habilidad que tienen unos elementos de la red para garantizar en cierto nivel la satisfacción de las necesidades de tráfico, y en el que entran en juego todas las capas de

Capítulo 2: Solución Propuesta

la red. QoS trabaja asignando prioridades según las necesidades del tráfico en la red, administrando de esta manera el ancho de banda de la red.

QoS tiene la habilidad de distinguir entre diferentes tipos de tráfico y asignar recursos de acuerdo con parámetros como ancho de banda, retardo, jitter (variación del retardo) y pérdida de paquetes. Agrupar el tráfico por tipo de aplicaciones con características comunes dentro de una categoría permite usar esquemas más amplios de políticas de QoS.

QoS es la medida de satisfacción de un cliente al recibir sus servicios de información en las redes. Debe asegurar que se transmitirá cierta cantidad de datos en un tiempo dado. El contrato que especifica los parámetros de QoS acordados entre el proveedor y el usuario se denomina SLA cuyas métricas típicas son: disponibilidad, ancho de banda, packet loss, round trip delay, jitter, entre los relevantes.

2.2.3 Métricas de SLA

En SLA es importante definir un conjunto de datos que se puedan medir para verificar la ejecución del servicio. En algunos casos será necesaria la utilización de herramientas para medir este servicio, y en otros casos se tratará de métricas que se puedan chequear manualmente. Este conjunto de métricas es imprescindible para definir correctamente el grado de cumplimiento o incumplimiento del contrato, sin dar lugar a una posible subjetividad. Las métricas deben definirse junto con sus puntos de control. Es importante asignar a estas métricas alarmas automáticas, de tal forma que informen al cliente cuanto antes de una incidencia.

Un SLA puede tener diferentes métricas de rendimiento KPI para distintos objetivos de QoS y distintos servicios. Estas métricas permiten segmentar la oferta de servicios, lo que beneficia tanto al cliente como al proveedor. Una correcta definición de las métricas evita la compilación redundante de datos y optimiza los sistemas.

La información de los SLA no ofrece una perspectiva completa por sí sola. Los parámetros KPI evalúan el rendimiento en función de los objetivos cuantificables del SLA para determinar qué procesos tienen un mayor impacto de negocio. Los proveedores de servicios pueden establecer rápidamente nuevos

Capítulo 2: Solución Propuesta

umbrales y desviaciones aceptables, mientras que los usuarios pueden definir nuevas reglas de KPI para detectar eventos como usos anormales o nuevos suscriptores semanales.

2.2.4 Métricas de KPI

Los KPI son métricas financieras y no financieras para cuantificar los objetivos que reflejan el desempeño estratégico de una compañía. Se utilizan en Inteligencia de Negocios para evaluar el estado actual del negocio y sugerir cursos de acción. (Camilo, 2007)

El acto de monitorear los KPI en tiempo real se conoce como el monitoreo de la actividad del negocio BAM y se usan con frecuencia para medir o darle valor a actividades difíciles de medir, como los beneficios del desarrollo del liderazgo, proveeduría de un buen servicio o la satisfacción del cliente. (Camilo, 2007)

2.2.5 Métricas de BAM

BAM es utilizado para detectar problemas y oportunidades. Para obtener el máximo beneficio de éste, las organizaciones deben experimentar un importante cambio cultural, así como cambios en sus flujos de trabajo.

La monitorización de la actividad es un elemento clave en la mejora de procesos y servicios: las soluciones de BAM aumentan las posibilidades del negocio a través de una estrategia de control y seguimiento proactivo y en tiempo real de la actividad dentro de los sistemas de información.

Uno de los principales beneficios que proporciona contar con BAM es disponer de métricas que permitan establecer el rendimiento del negocio y la calidad del servicio ofrecido, facilitando la optimización de procesos y sistemas.

2.3 Análisis de la Arquitectura Orientada a Servicios en la UCI

Una aplicación SOA está formada por un conjunto de servicios interconectados cuyo objetivo es automatizar uno o varios procesos de negocio.

Capítulo 2: Solución Propuesta

SOA propone como primer paso reutilizar las aplicaciones legacy existentes separándolas en unidades de lógica de negocio expuesta para ser utilizadas y reutilizadas dentro de la organización, llamadas servicios. Por lo que se comienza a llamar servicio el cual deberá contener lógica de negocio estructurada en varios niveles de acuerdo a las necesidades.

En la UCI se definieron 6 niveles de servicios, comenzando por el nivel más bajo:

- **Nivel 1: Servicios de infraestructura** se concentran en la preparación y en la implementación de una infraestructura de TI que dé soporte a la integración de las aplicaciones en un entorno de SOA. Los servicios conectan y reutilizan los activos existentes en su infraestructura, y crean control y vigilancia para reducir los riesgos de las iniciativas de SOA.
- **Nivel 2: Servicios utilitarios** son aquellos servicios que se caracterizan por representar una tarea altamente reutilizable. Existen dos tipos, los servicios orientados al negocio que representan una tarea de negocio altamente reutilizable entre aplicaciones y los servicios tecnológicos encargados de encapsular una determinada tecnología y por tanto altamente reutilizables.
- **Nivel 3: Servicios núcleos del negocio** estos son los servicios que tienen las funciones de más alta responsabilidad en el núcleo del negocio de la empresa (UCI).
- **Nivel 4: Servicios de capacidades de negocio** la primera capacidad de una Arquitectura de Referencia SOA es la facilidad para creación de servicios de negocios, construidos a partir de los sistemas existentes en una red empresarial extendida o servicios completamente nuevos. Son los servicios con la capacidad de creación y exposición de servicios a partir de componentes o programas existentes o creación de nuevos servicios.
- **Nivel 5: Servicios de procesos** servicios de negocio que encapsulan la lógica de proceso. Suelen conservar estados y pueden residir en herramientas BPM.

Capítulo 2: Solución Propuesta

- **Nivel 6: Soluciones** son los encargados de recibir las peticiones de los clientes y realizar las llamadas necesarias a otros servicios (en la secuencia adecuada) para devolver una respuesta. Es decir, son los sistemas que interactúan con el usuario.

La UCI se encuentra actualmente definiendo servicios. Los servicios definidos hasta el momento se encuentran en los niveles de servicios 1, 2 y 3 mostrados a continuación:

- Registro de Identidad UCI
- ASSETS - Sistema de Capital Humano
- Akademos - Sistema de Gestión Académica
- Registro de Residentes
- Registro de Personal
- Reservación de Transporte Estudiantil

No todos los servicios cumplirán la misma función ni al mismo nivel, habrá algunos que solo se comuniquen con los repositorios de información, otros que implementen lógica de negocio y otros que engloben completamente procesos complejos como orquestación de servicios más básicos.

Una vez definidos los servicios Web, se necesita darlos a conocer a la comunidad para que conozcan de su existencia. UDDI será el encargado de ello, como directorio global de servicios y será el punto de partida para la construcción de los mismos. Cada vez que se vaya a desarrollar una nueva aplicación, los desarrolladores deben consultar la UDDI para informarse de los servicios disponibles y de cómo aprovechar esta información para la reutilización de funcionalidades ya establecidas por otros servicios.

Debido a que la Arquitectura Orientada a Servicios es novedosa en la Universidad todavía no se ha desarrollado la UDDI, ESB, BPM, Gobernabilidad y otros componentes que conforman esta arquitectura, hasta el momento solo se han desarrollado los servicios anteriormente mencionados; el objetivo de esta investigación es precisamente definir las métricas de calidad, para refinar la calidad de estos servicios.

2.4 Definición de las Métricas

No existe una métrica que mida lo que se desea a la perfección solo se definen las que más cercanas estén al resultado que se desea obtener. Hay que aplicar la mayor cantidad de métricas posibles para obtener el resultado lo más exacto posible. Para la aplicación de métricas en las organizaciones, lo primero es reconocer la necesidad del uso de métricas, seguido a esto se debe definir cuáles son los objetivos que se persiguen con la aplicación de las mismas en la organización, según los objetivos se determina el tipo de métrica a utilizar.

Teniendo en cuenta el estudio realizado a la Arquitectura Orientada a Servicios en la UCI y conociendo que uno de los problemas que existe actualmente es la no existencia de un procedimiento estándar que controle la calidad y que aún no se le han aplicado métricas a la arquitectura, esta investigación propone definir una métrica o sistema de métricas que de solución a dicho problema.

A la hora de definir la métrica o sistema de métricas se debe tener en cuenta que las mismas:

- Cubran todo el ciclo de vida.
- Estén relacionadas directamente con los problemas que pretenden atacar.
- Brinden visibilidad del rendimiento de los procesos críticos seleccionados.
- Estén relacionadas directamente con los objetivos de negocio.
- Cubran tantas facetas y usuarios de los procesos como sea posible.
- Reflejen el grado en el cual un proceso logra objetivos importantes.
- Facilite la recopilación consistente y bien definida.
- Muestre una variación medible.
- Ayuden a determinar la causa del problema y no solamente que el problema ha ocurrido.

Para definir dicha métrica o sistema de métricas es necesario saber hasta que punto deben ajustarse al proceso, para ello se hace imprescindible que las métricas definidas sean comunes para todos los servicios ya que así sería más efectivo controlar la calidad y que las métricas sean fáciles de aplicar teniendo en cuenta que hasta el momento no se han aplicado, y no se tiene la experiencia necesaria.

2.4.1 Estructura del sistema de Métricas

Partiendo por la convicción de que en la UCI hasta el momento solo se han definido y desarrollado servicios, el sistema de métricas propuestas estará dirigido a proporcionar un indicador que facilite una visión del estado de éstos servicios para poder hacer los ajustes necesarios para mejorar los mismos.

Las métricas se definirán teniendo en cuenta los atributos de calidad.

Funcionalidad: Es la capacidad del servicio Web para proveer las funciones que cumplan con las necesidades específicas o implícitas, cuando es utilizado bajo ciertas condiciones.

Para determinar si un servicio Web cumple con un determinado nivel de servicio, es decir determinar cuales son los servicios más importantes y utilizados, para así si estos servicios prioritarios tienen un nivel de servicio insuficiente, un estudio de las causas de la ineficiencia debiera partir del análisis de estos servicios. Se ha definido la métrica Nivel de Servicio.

➤ **Métrica: Nivel de Servicio**

Esta métrica mide la probabilidad de que un servicio Web cumpla un tiempo exigido T_0 . Esta se debe mantener alta, igual o casi igual a 1.

$$NS(w) = \frac{N(T(w) < T_0)}{N(w)}$$

Donde:

NS (w): Nivel de servicio de un servicio determinado w.

N (w): Número de llamadas a ese servicio en un periodo determinado.

T₀: Tiempo exigido de respuesta de ese servicio.

T (w): Tiempo de respuesta del servicio w, que es el tiempo tomado entre la petición al servicio hasta que el servicio entrega una respuesta satisfactoria.

Capítulo 2: Solución Propuesta

Si un servicio Web genera mucha carga en una red, está teniendo muchas dependencias. Este es un potencial riesgo de una arquitectura SOA, la dependencia excesiva. ¿Cuánta dependencia hay en la red? (MORENO, 2006)

La métrica Grado de Conexión determina esta dependencia.

➤ Métrica: Grado de Conexión

Esta métrica indica cuán conectada está la red. A mayor número de conexiones, más cercano a 1 será su valor. Una conexión total vale 1. Un valor alto, cercano a 1, indicaría una red demasiado dependiente. Por el contrario, un valor muy cercano a cero indicaría un escaso uso de la red.

$$GC(G) = \frac{N(L)}{N(W) \times (N(W) - 1)}$$

Donde:

GC (G): Grado de Conexión de una red G.

N (W): Número de servicios Web W. Por ejemplo $W = \{A, B, C, D\}$ siendo A, B, C y D servicios Web.

N (L): Número del conjunto de llamadas L. Por ejemplo $L = \{(A, B)(B, C)\}$ siendo B un servicio llamado por A y C un servicio llamado por B.

Grado de Conexión es insuficiente para indicar aspectos de la red relacionados con la orientación de las dependencias; de hecho, si se define $L' = \{(A, B) \in W \times W : (B, A) \in L\}$, se ve que la red de dependencias $G = (W, L')$ tiene el mismo grado de conexión que $G = (W, L)$, pero todos los arcos están invertidos. La **figura 2.1** muestra que la dirección de los arcos es muy importante. En la red desconcentrada (a), no parece haber un nodo más importante que otro. Sin embargo, si se cae cualquier servicio Web, C se caería también. Además, si C se ejecuta, toda la red se ejecuta también. En la red concentrada (b), hay un nodo que es crítico para el funcionamiento de la red: C. Si C se cae, toda la red queda inutilizada. Pero si se cae A, B, D o E, el resto de la red sigue funcional. (MORENO, 2006)

Capítulo 2: Solución Propuesta



Figura 2.1. Concentración y desconcentración.

La noción informal de red concentración de la red, es que si algunos nodos determinan el funcionamiento de gran parte de la red, la red está concentrada en esos nodos es decir depende de esos nodos. En cambio, si los nodos dependen de muchos otros, la red está desconcentrada. Ambos problemas desaparecen si la red está desconectada, o sea, tiene un Grado de Conexión nulo (o pequeño). Y si Grado de Conexión es alto (cercano a uno), estos problemas se agravan.

➤ **Métrica: Grado de Concentración**

Mide el grado de concentración de la red. Una red concentrada es muy sensible a la caída de ciertos servicios Web.

$$GC(G) = \sqrt{6 \frac{\sum_{w \in W} Entrantes^2(w)}{2N^3(W)3N^2(W) + N(W)}}$$

Donde:

GC (G): Grado de Concentración

Entrantes (w): es una métrica definida más adelante.

➤ **Métrica: Grado de Desconcentración**

Mide el grado de desconcentración de la red. Una red desconcentrada tiene servicios que generan mucha carga al resto de la red.

Capítulo 2: Solución Propuesta

$$GD(G) = \sqrt{6 \frac{\sum_{w \in W} Salientes^2(w)}{2N^3(W)3N^2(W) + N(W)}}$$

Donde:

GD (G): Grado de Desconcentración

Salientes (w): es una métrica definida más adelante.

Otro aspecto importante es el número de servicios Web que se deben ejecutar. Esto es diferente al número de llamadas que implica la ejecución de un servicio. La razón para que ocurra esto es la redundancia existente. **La figura 2.2** muestra un servicio Web W que implica la ejecución de cinco servicios Web (A, B, C, D, E), pero ocurren siete llamadas. Los servicios D y E son ejecutados dos veces. Esto puede ser una potencial redundancia.

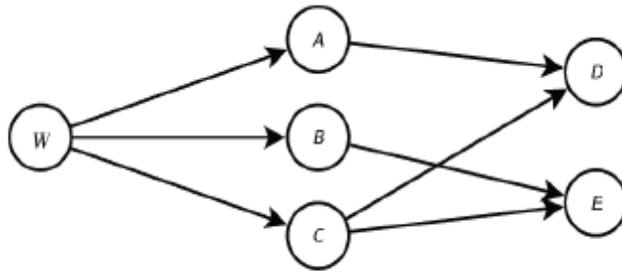


Figura 2.2. Potenciales llamadas redundantes.

Cada vez que un servicio Web se ejecuta, sus dependencias directas son todas ejecutadas también, y al ejecutarse, las llamadas de cada una también son ejecutadas, recursivamente. Por ejemplo, si a llama a b y b genera 10 llamadas, entonces a genera la llamada a b y las 10 llamadas que produce b. ¿Cuántas llamadas implica la ejecución de un servicio Web? Las métricas definidas para determinar estas llamadas son las nombradas entrantes y salientes. (MORENO, 2006)

➤ Métrica: Entrantes

Esta métrica mide la carga que genera un servicio Web al ser ejecutado, como el número de servicios Web que acceden directamente a dicho servicio.

$$Entrantes(w) = N(v \in W: (v, w) \in L)$$

Capítulo 2: Solución Propuesta

$$N(v \in W: Directa(v, w))$$

➤ Métrica: Salientes

Esta métrica mide la carga que genera un servicio Web al ser ejecutado, como el número de servicios Web llamados por dicho servicio.

$$Salientes(w) = N(v \in W: (w, v) \in L)$$

$$N(v \in W: Directa(w, v))$$

Confiabilidad: La confiabilidad es la capacidad del servicio Web para mantener un nivel especificado de rendimiento cuando es utilizado bajo condiciones especificadas.

➤ Métrica: Carga Subestimada

Mide la cantidad de servicios que dependen de un determinado servicio.

$$CS(w) = N(v \in W: Dependencia(v, w))$$

Donde:

CS (w): Carga Subestimada de un servicio w

➤ Métrica: Recarga

Esta métrica indica la cantidad potencial de ejecuciones redundantes que son consecuencia de la ejecución de un servicio Web. Esta métrica debe ser usada con cuidado pues no indica realmente si las potenciales redundancias son redundancias reales. En este sentido, Recarga sólo sirve para iniciar una inspección.

$$RC(w) = CEG(w) \times CS(w)$$

Donde:

RC (w): Recarga de un servicio w

CEG (w): Métrica definida más adelante.

Capítulo 2: Solución Propuesta

La disponibilidad del servicio también es muy importante, se define normalmente como la proporción del tiempo que un bien o servicio puede ser utilizado. El resto del tiempo, el servicio está dado de baja o estropeado. Considerando el caso de los servicios Web, la demanda diaria de un servicio puede variar regularmente, por lo que la disponibilidad deja de tener sentido. Lo que realmente interesa es la probabilidad de que un servicio Web responda correctamente una petición.

➤ **Métrica: Probabilidad de Servicio Disponible**

Mide la probabilidad de que un servicio Web responda correctamente. La métrica Probabilidad de Servicio Disponible se debe mantener alta igual o casi igual a 1. Valores como 0,95 son muy bajos para Probabilidad de Servicio Disponible; considerando los objetivos del gobierno electrónico, el propósito del servicio Web es estar disponible en todo momento.

$$PSD(w) = \frac{NC}{N}$$

Donde:

PSD (w): Probabilidad de Servicio Disponible de en servicio w.

NC: Número de veces que el servicio w ha respondido correctamente en la últimas N ejecuciones.

Probabilidad de Servicio Disponible debe ser muestreada por un período largo de tiempo debido a que un servicio Web no queda inutilizado con frecuencia. Los sistemas informáticos actuales son bastante resistentes a las caídas es por eso que esta métrica debe ser muestreada durante un período largo de tiempo, para incorporar varias caídas dentro de las observaciones. De todas maneras, se debe cumplir que: $PSD(w) > NS(w)$ (pues toda caída reduce el nivel de servicio).

Usabilidad: Esta categoría se refiere a la capacidad del servicio Web para ser atractivo, entendido, aprendido y utilizado por el usuario bajo condiciones específicas.

¿Cuánta carga genera un servicio Web al ejecutarse? La métrica Carga Estructural Generada se define como el número de llamadas implicadas por la ejecución de un servicio Web. Sumando una llamada por dependencia directa y la carga generada por éstas, se define recursivamente.

Capítulo 2: Solución Propuesta

➤ Métrica: Carga Estructural Generada

Esta métrica mide la carga generada por un servicio Web al ejecutarse.

$$CEG(w) = \sum_{v \in W: \text{Directa}(w,v)} (1 + CEG(v))$$

Donde:

CEG (w): Carga Estructural generada de un servicio w.

CEG (v): Carga Estructural generada por la llamada de w a un servicio v.

Observaciones: (MORENO, 2006)

- La fórmula tiene en cuenta que llamada se ejecuta una vez. La carga producida por una llamada es producto de ejecutar el servicio Web llamado y de la carga que genera ese servicio al resto de la red (recursividad).
- Si un servicio Web no depende directamente de otro nodo, entonces no ocurre la suma. Luego, la métrica Carga Estructural Generada será simplemente 0.

Eficiencia: Es la capacidad del servicio Web para proveer un rendimiento apropiado, relativo a la cantidad de recursos utilizados, bajo condiciones específicas.

El tiempo de respuesta de un servicio Web está fuertemente relacionado al nivel de servicio y a la aleatoriedad de las dependencias.

➤ Métrica: Tiempo Esperado de Respuesta

Permite conocer cuánto se demora realmente el servicio Web sin considerar la carga generada (descontando el tiempo que significa usar las dependencias).

$$TER(w) = \frac{\sum_{k \in \{\text{Últimas } N \text{ ejecuciones}\}} T_{S_k}^w}{N}$$

Capítulo 2: Solución Propuesta

Donde:

TER (w): Tiempo Esperado de Respuesta de un servicio w

Ts: Tiempo de respuesta del servicio w en su k-ésima ejecución.

Asumiendo que un servicio Web debe esperar la respuesta de los servicios que solicita, descontar el tiempo de espera revela el propio tiempo de proceso del servicio.

➤ Métrica: Tiempo Propio de Respuesta

Mide el tiempo “propio” de respuesta, aquel no explicado por las llamadas realizadas por el servicio Web.

$$TPR(w) = TER(w) \times \sum_{(w,v) \in L} NL(w,v) \times TER(v)$$

Donde:

TPR (w): Tiempo Propio de Respuesta de un servicio w.

NL (w, v): Métrica definida más adelante.

La demanda de un servicio Web representa cuántas solicitudes (llamadas) se realizan a un servicio en un determinado período de tiempo. En este sentido, la demanda es una frecuencia.

➤ Métrica: Demanda Total

Indica la frecuencia de accesos al servicio Web (sea o no proveniente de la red) en un intervalo de tiempo I.

$$D(w) = NA$$

Donde:

D (w): Demanda Total de un servicio w

NA: Número de accesos a w en un intervalo I

Capítulo 2: Solución Propuesta

La métrica Demanda Total es acumulativa en el número de accesos dentro de un intervalo de tiempo. La demanda de un servicio Web es información relevante. Y es un buen criterio para definir servicios Web cuyo nivel de servicio es relevante. Esta información se puede dividir, es posible determinando si el servicio Web tiene alta demanda interna a la red o alta demanda externa a la red.

La demanda interna es aquella originada por otros servicios Web. Un servicio con alta demanda interna es crucial en la red.

➤ **Métrica: Demanda Interna**

Mide la demanda explicada por las llamadas realizadas desde el interior de la red.

$$DI(w) = \sum_{v \in W: Directa(v,w)} D(v) \times NL(v, w)$$

Donde:

DI (w): Demanda Interna de un servicio w.

NL (v, w): Métrica definida más adelante.

La demanda externa es aquella que no proviene de otros servicios Web (de la red G (W, L))

➤ **Métrica: Demanda Externa**

Mide la demanda que no es explicada por las llamadas realizadas al interior de la red

$$DE(w) = D(w) \times DI(w)$$

Donde:

DE (w): Demanda Externa de un servicio w

La carga de un servicio Web (y de su servidor) proviene principalmente de su demanda. Pero un servicio Web también puede generar demanda a otros servicios. Esta demanda representa la carga de trabajo, sobre la red, que provoca la ejecución de ese servicio Web. La métrica Carga Real Generada sobre la red, considera aspectos de la ejecución.

Capítulo 2: Solución Propuesta

➤ Métrica: Carga Real Generada

Indica la demanda total generada sobre la red, por una ejecución del servicio. Mide el número real de llamadas generadas por la simple ejecución de un servicio Web. Si se multiplica por la demanda de ese servicio, el resultado será toda la demanda ocasionada por el servicio Web, ya sea a través de dependencias directas o indirectas.

$$CRG(w) = \sum_{(w,v) \in L} NL(w,v) \times (1 + CRG(v))$$

Donde:

CRG (w): Carga Real Generada por un servicio w.

NL (w, v): Métrica definida más adelante.

Si un servicio Web llama a otro servicio Web varias veces durante una ejecución, puede ser síntoma de llamadas redundantes. Un análisis del motivo de las múltiples llamadas podría revelar una oportunidad de mejora de un servicio, debido a la menor carga generada durante su ejecución y al menor tiempo de respuesta. Por tal motivo se proponen las métricas para la probabilidad de realizar una llamada y el número esperado de llamadas que realiza un servicio ante otro, por ejecución.

➤ Métrica: Probabilidad de Llamada

Mide la probabilidad que la llamada del servicio Web w al servicio Web v sea realizada durante una ejecución del servicio Web w.

$$PLL(w,v) = \frac{N \times Nn}{N}$$

Donde:

PLL (w, v): Probabilidad que la llamada del servicio w al servicio v sea realizada en la ejecución del servicio w.

Nn: Veces que el servicio w no llama al v, en las últimas N ejecuciones.

Capítulo 2: Solución Propuesta

➤ Métrica: Número de Llamadas

Mide el número esperado de veces que un servicio w realiza una llamada al servicio v por ejecución.

$$NL(w, v) = \frac{Na}{N}$$

Donde:

NL: Número de llamadas.

Na: Llamadas del servicio w al v en las últimas N ejecuciones de w .

Observación. La fórmula para calcular NL es simplemente una medida, un promedio.

Es posible notar que una llamada I que ocurre a lo más una vez por ejecución, tiene la propiedad: $PLL(I) = NL(I)$, $I \in L$. De lo contrario, $PLL(I) < NL(I)$. De hecho, NL se puede relacionar a PLL de la siguiente forma: $NL(a, b) = E(\text{Veces que } a \text{ llama a } b \mid a \text{ necesita de } b) \times PLL(a, b)$.

➤ Métrica: Número Relativo de Llamadas

Estima la cantidad esperada de ejecuciones de I que un servicio hace, dado que va a usar la dependencia I . La filosofía es que, si va a hacer una llamada, hacerla más de una vez puede significar redundancia. Para completar, se resta uno, dejando sólo la potencial redundancia esperada del uso de la llamada I . Si $NRLL$ es igual a cero, entonces la llamada no ocurre de manera redundante. Si es mayor que cero, puede significar que hay redundancia (depende de la naturaleza de las llamadas).

$$NRLL(I) = \frac{NL(I)}{PLL(I)} - 1$$

Donde:

NRLL (I): Número Relativo de Llamadas de I .

Si $NRLL$ es positivo, entonces hay una posibilidad de reducir el número de llamadas que realiza un servicio, mejorando su tiempo de respuesta.

El sistema de métricas se plantea como propuesta de solución al problema de la investigación. Se concibe desde la perspectiva de que en la Universidad hasta el momento en la Arquitectura Orientada a Servicios solo se han desarrollado servicios Web. Las métricas están definidas con el objetivo de

Capítulo 2: Solución Propuesta

obtener indicadores que muestren el estado actual de dichos servicios y poder trazar acciones con el propósito de mejorarlos. Estas métricas proporcionan indicadores de funcionalidad, confiabilidad, usabilidad y eficiencia.

2.4.2 Guía de aplicación de la propuesta

La aplicación de métricas de calidad en una Arquitectura Orientada a Servicios puede realizarse persiguiendo varios objetivos:

- Calidad de los servicios en un tiempo determinado.
- Calidad para demostrar que se sigue un procedimiento estándar en los servicios.
- Calidad de servicios para compararlo con la calidad de la arquitectura en general.

El valor que se obtiene al evaluar los valores en una métrica, no aportan información convincente, solo ayuda a que se realicen las comparaciones para obtener la información requerida y así tomar decisiones sobre el proceso, se puede obtener información por ejemplo: si existe mucha diferencia entre los tiempos de respuesta de los servicios puede ser un indicador de que no se sigue un procedimiento estándar, si se compara los tiempos de respuesta del proceso general con los de etapas anteriores, se puede concluir el estado de la calidad en una Arquitectura Orientada a Servicios, si ha avanzado, o no. Después de analizar estos elementos cabría preguntar:

¿Cómo se lleva a cabo el proceso de aplicación de las métricas propuestas?

El primer paso para llevar a cabo el proceso de aplicación de las métricas es determinar los objetivos que se persiguen con la aplicación de las mismas.

Una vez definido hacia que va estar orientado el resultado obtenido, o sea que información brindará.

La propuesta consiste en un sistema de métricas que permiten obtener la calidad de los servicios de una Arquitectura Orientada a Servicios de la Universidad de las Ciencias Informáticas.

Las primeras métricas definidas son las que proporcionan indicadores de funcionalidad de los servicios Web.

Capítulo 2: Solución Propuesta

Para obtener este resultado se debe partir de obtener cuales son los servicios más utilizados, más importantes y así en caso de ineficiencias partir por el estudio de estos servicios prioritarios, la métrica definida **Nivel de Servicio**: $NS(w) = \frac{N(T(w) < T0)}{N(w)}$ determina cuales son dichos servicios.

Otra métrica que mide funcionamiento de estos servicios es **Grado de Conexión**: $GCo(G) = \frac{N(L)}{N(W) \times (N(W) - 1)}$ que proporciona una visión de la conexión de la red. Pero debido a que esta métrica es

insuficiente para indicar aspectos de la red relacionados con la orientación de las dependencias, es decir si la red está concentrada o desconcentrada. La métrica **Grado de Concentración** mide el grado de concentración de la red para poder obtener esta medida hace falta calcular primero la métrica **Entrantes**: $Entrantes(w) = N(v \in W: (v, w) \in L)$. Después de obtener el valor de la métrica

Entrantes podemos calcular el **Grado de Concentración**: $GC(G) = \sqrt{6 \frac{\sum_{w \in W} Entrantes^2(w)}{2N^3(W)3N^2(W) + N(W)}}$. Para

obtener el valor del grado de Desconcentración hay que calcular **Salientes** que proporciona una visión de de la carga que generan los servicios Web al ser ejecutados. $Salientes(w) = N(v \in W: (w, v) \in L)$.

Ahora podemos obtener el valor de la métrica **Grado de Desconcentración**:

$GD(G) = \sqrt{6 \frac{\sum_{w \in W} Salientes^2(w)}{2N^3(W)3N^2(W) + N(W)}}$. Evidentemente para obtener el grado de Concentración y de

Desconcentración se tienen que haber calculado primero las métricas Entrantes y Salientes.

Para medir la confiabilidad de los servicios Web se definieron métricas específicas para mantener un nivel de rendimiento cuando es utilizado bajo condiciones especificadas, la métrica **Carga**

Subestimada: $CS(w) = N(v \in W: Dependencia(v, w))$ determina cuantos servicios dependen de un servicio determinado. **Carga Estructural Generada**: $CEG(w) = \sum_{v \in W: Directa(w, v)} (1 + CEG(v))$ mide

la carga generada por un servicio al ser ejecutado y constituye una métrica que define la usabilidad de los servicios. Después de haber calculado las métricas **Carga Subestimada** y **Carga Estructural**

Generada podemos calcular entonces la métrica **Recarga**: $RC(w) = CEG(w) \times CS(w)$ que indica la cantidad potencial de ejecuciones redundantes que son consecuencia de la ejecución de un servicio

Web. **Probabilidad de Servicio Disponible**: $PSD(w) = \frac{NC}{N}$ es otra métrica que indica la recuperabilidad y probabilidad de que un servicio Web responda correctamente.

Capítulo 2: Solución Propuesta

Para determinar la eficiencia de los servicios se definieron un conjunto de métricas que definen a los mismos, bajo condiciones específicas. **Tiempo Esperado de Respuesta:**

$TER(w) = \frac{\sum_{k \in \{\text{últimas } N \text{ ejecuciones}\}} T_s^w}{N}$ permite estimar niveles de servicio para distintas exigencias de

tiempo. La métrica **Número de Llamadas:** $NL(w, v) = \frac{Na}{N}$ mide el número esperado de veces que un

servicio realiza una llamada a otro por ejecución la cual es utilizada para determinar el **Tiempo Propio**

de Respuesta: $TPR(w) = TER(w) \times \sum_{(w,v) \in L} NL(w, v) \times TER(v)$ que mide el tiempo de respuesta no explicado por las llamadas realizadas por el servicio Web.

Otro atributo medible muy importante es la demanda de un servicio Web la métrica **Demanda Total:**

$D(w) = NA$ determina lo solicitado que puede ser un servicio. La demanda puede ser interna o

externa de acuerdo a lo que queramos medir entonces se definen las métricas **Demanda Interna:**

$DI(w) = \sum_{v \in W: Directa(v,w)} D(v) \times NL(v, w)$ y **Demanda Externa:** $DE(w) = D(w) \times DI(w)$. La

demanda de un servicio puede provocar la carga de ese servicio. Esta demanda representa la carga

de trabajo, sobre la red, que provoca la ejecución de ese servicio. **Probabilidad de Llamada:**

$PLL(w, v) = \frac{N \times Nn}{N}$ mide la probabilidad que la llamada de un servicio Web a otro sea realizada

durante una ejecución de uno de los servicios. Después de haber obtenido el valor de la métrica

Probabilidad de Llamada y Número de Llamadas podemos calcular la métrica **Número Relativo de**

Llamadas: $NRLL(I) = \frac{NL(I)}{PLL(I)} - 1$ que estima la cantidad esperada de ejecuciones de I que un

servicio hace, dado que va a usar la dependencia, I y la métrica **Carga Real Generada:** $CRG(w) =$

$\sum_{(w,v) \in L} NL(w, v) \times (1 + CRG(v))$ que indica la demanda total generada sobre la red.

Es importante destacar que para llevar a cabo el proceso de aplicación de las métricas se debe designar una persona o grupo de personas que ejecuten los pasos a seguir. Estas personas deben ser capaces de realizar conclusiones según los resultados obtenidos para tomar decisiones sobre el proceso.

2.5 Conclusiones del capítulo

En este capítulo se presentaron una serie de tecnologías que proponen métricas para medir la calidad de una Arquitectura Orientada a Servicios, de las que se definieron las que se ajustaban al desarrollo de dicha arquitectura en la Universidad.

Uno de los factores más importantes que se observa es la no existencia de métricas para evaluar la calidad de las aplicaciones desarrolladas en una SOA en la UCI, donde se aprecia que el trabajo que se presenta es de vital importancia ya que trata de mejorar la calidad del producto y por tanto la calidad de los servicios.

Para lograr un sentido de pertenencia de las métricas en un programa de medición, se tiene que lograr la participación de todos. Las personas que trabajan con un proceso en su quehacer diario tendrán un conocimiento íntimo de ese proceso, esto les da una perspectiva valiosa de cómo el proceso puede ser bien medido para asegurar confiabilidad y validez, y cómo interpretar mejor las mediciones resultantes para maximizar la utilidad

CAPÍTULO 3: VALIDACIÓN DE LA SOLUCIÓN PROPUESTA

3.1 Introducción del capítulo

La solución propuesta se aplica a uno de los servicios desarrollado como parte de la Arquitectura Orientada a Servicios en la Universidad. La validación se realiza mediante la aplicación de la guía propuesta en el capítulo anterior. Se utiliza la herramienta SoapUI para obtener los datos necesarios para el cálculo de las métricas.

3.2 Validación de métricas

La validación de las métricas es un proceso de suma importancia en la disciplina de evaluación del software. Los resultados de la investigación cobran su valor siempre que sea posible validar sus resultados. La validación puede asegurar que la métrica fija un determinado factor de calidad.

No resulta significativa contar con una diversidad de métricas, si no se sabe si son válidas o no. Es decir que en verdad miden lo que realmente debieran medir. De esta manera, la validación constituye un aspecto crucial en la medición del software, ya que asegura que las medidas representen con precisión la eficiencia de los servicios que se pretenden cuantificar.

Actualmente se reconocen dos tipos de validación: la interna y la externa, comúnmente referidas como teórica y empírica respectivamente.

- **Validación interna:** es un ejercicio teórico que asegura que la métrica tiene una caracterización numérica apropiada de la propiedad que pretende medir. Esto es por supuesto un pre-requisito para demostrar la utilidad de dicha métrica.
- **Validación externa:** se lleva a cabo para demostrar con evidencia real que una métrica es útil en el sentido de que está asociada con alguna característica externa del software tal como la usabilidad, la mantenibilidad, etc.

Capítulo 3: Validación de la Solución Propuesta

Para demostrar que la propuesta de las métricas definidas es realmente válida y resulta de utilidad para la UCI, se valida la propuesta mediante la utilización de la herramienta SoapUI, la cual se utiliza en estos momentos en la Universidad para realizar las pruebas a los servicios desarrollados en una SOA. En la investigación se utiliza con el objetivo de obtener un valor que muestre la información necesaria que ayude a determinar la calidad de los servicios.

3.3 Herramienta SoapUI

SoapUI es la principal herramienta para las prueba de Servicios Web y una de la más utilizada para las pruebas de SOA en el mundo. SoapUI es libre y de código abierto fue desarrollada en Java y permite hacer pruebas funcionales y de carga para servicios Web. Es muy útil para probar y debuguear servicios Web en cualquier plataforma.

SoapUI es muy sencilla de utilizar solo es necesario cargar la WSDL del servicio y ya se tiene un generador de peticiones.

SoapUI tiene las siguientes áreas de funcionalidad

- Inspección e invocación de servicios Web
- Pruebas funcionales de servicios Web
- Pruebas de carga de servicios Web

La interfaz de usuario cumple con la arquitectura de la IDE típica: un panel de navegación a la izquierda, un panel de contenido a la derecha, y paneles de propiedades adicionales cerca de la parte inferior. SoapUI acomoda el trabajo en proyectos. Cada proyecto es identificado principalmente por las interfaces que éste construye para las pruebas.

El trabajo real de esta herramienta es el envío de peticiones, recepción de respuestas, analizar resultados, y alterar el flujo de la ejecución de la prueba que ocurre al nivel del paso de prueba. SoapUI determina una falla o éxito de una prueba.

Un paso de prueba es análogo a una línea de código en un programa. Actualmente, SoapUI define seis tipos de pasos de prueba, siendo el más común una petición, el cual envía un requerimiento HTTP

Capítulo 3: Validación de la Solución Propuesta

a un destino y acepta una respuesta. Controla el flujo a través del cual los pasos de prueba pueden ser modificados insertando un paso de prueba, esencialmente una colección de una o más expresiones Xpath que examinan la respuesta más reciente. La primera expresión en la colección que tiene éxito causa una ramificación a un paso de prueba de objetivo asociado.

SoapUI se utiliza en la investigación fundamentalmente para la prueba de funcionamiento de cada uno de los métodos de los servicios Web y para determinar el tiempo de respuesta de dichos métodos.

3.4 Las métricas de calidad en los servicios de la UCI

En vista a validar la propuesta de métricas de calidad definidas en el capítulo anterior, se impone aplicar dichas métricas a uno de los servicios desarrollados en la Universidad como parte de la Arquitectura Orientada a Servicios. Los resultados que se obtengan servirán de base para analizar el conjunto de métricas propuestas.

Reconocer que para este tipo de aplicaciones es necesario aplicar un conjunto de métricas, es un paso importante para llevar a cabo el proceso de medición.

La validación de la propuesta de solución se aplica al servicio Registro de Personal, por ser éste el único servicio de la Arquitectura Orientada a Servicios en la Universidad que se relaciona con otro, actualmente es el único servicio que se encuentra conectado en la arquitectura.

3.5 Servicio Registro de Personal

El servicio Registro de Personal es el servicio que contiene los datos de cada uno de los trabajadores de la UCI que pertenecen a las áreas llamadas tercerizadas entre las que se encuentran ETECSA, MININT, SEPCOM, COPEXTEL, GASTRONOMÍA, y otras. Actualmente el servicio cuenta con 12 funcionalidades que permiten conocer las áreas que existen en la universidad, las categorías de cada una de las personas tercerizadas, las personas que cumplen año dado un día y un mes, la persona dado el carné de identidad, la persona dado el Id del expediente, los datos de todas las personas terceras o eventuales que son altas activas en la UCI, los datos de las personas de alta desde una

Capítulo 3: Validación de la Solución Propuesta

fecha dada, todas las personas tercerizadas o eventuales que son bajas en la UCI, los datos de las personas de baja desde una fecha dada, todas las personas que tienen una categoría dada, las personas que tienen un cargo dado y todas las personas dado el id de una categoría.

Los métodos del servicio Registro de Personal se relacionan con el servicio Assets Sistema de Capital Humano que es el servicio que contiene los datos de los trabajadores que son plantilla de la UCI, y la información de las áreas pertenecientes a la UCI. **(Ver Figura 3.1)**

Las métricas serán aplicadas a este servicio con el objetivo de obtener indicadores que **brinden** una visión de la calidad del servicio en la arquitectura de la Universidad, y determinar si resulta válido aplicar las métricas propuestas.

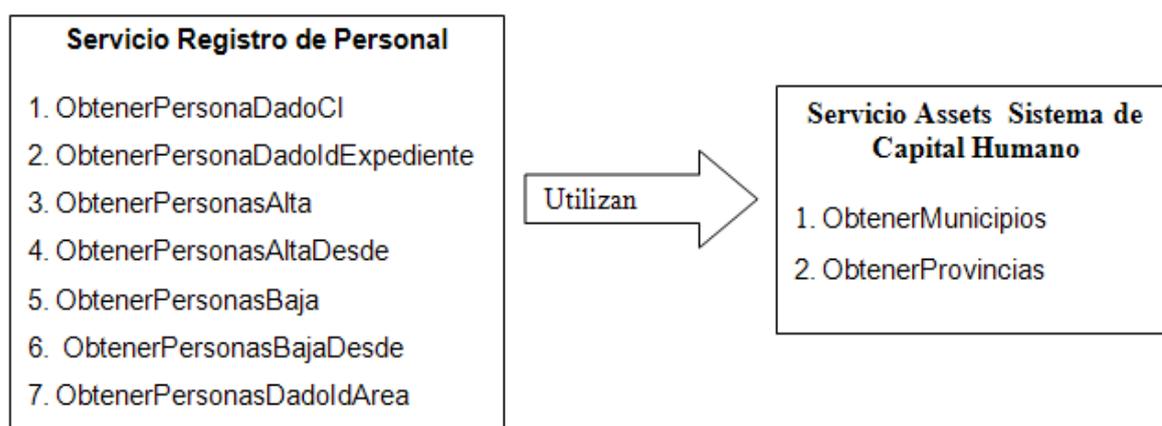


Figura 3.1 Relación del servicio Registro de Personal y Assets Sistema de Capital Humano.

3.6 Calidad en el servicio Registro de Personal (RP)

Según el sistema de métricas propuestas **(Ver anexos Tabla 1)** y siguiendo los pasos de la guía de aplicación propuesta, se comienza aplicando la métrica Nivel de Servicio, para ello se define el tiempo exigido de respuesta. El hecho de elegir un valor u otro puede ser demasiado arbitrario, depende de los requisitos del servicio, en este caso sería para cada uno de los métodos. Por lo que se define el tiempo exigido **(Ver Tabla 1)** de 1000 ms y 2000 ms en dependencia de la complejidad del método, es decir para los métodos de menos complejidad 1000 ms y para los demás que requieren de mayor

Capítulo 3: Validación de la Solución Propuesta

complejidad 2000 ms, por ejemplo el método ObtenerAreasTercerizadas va a devolver las áreas tercerizadas de la UCI, es una búsqueda sencilla por lo que se define un tiempo exigido de 1000 ms, mientras que el método ObtenerPersonaDadoCI, el cual devolverá la persona con el número de CI dado, es una búsqueda más compleja por lo que se define un tiempo exigido de 2000 ms.

Métodos	T0 ms
ObtenerAreasTercerizadas	1000
ObtenerCategorias	1000
ObtenerCumpleTercerosDadoDiayMes	2000
ObtenerPersonaDadoCI	2000
ObtenerPersonaDadoldExpediente	1000
ObtenerPersonasAlta	2000
ObtenerPersonasAltaDesde	2000
ObtenerPersonasBaja	2000
ObtenerPersonasBajaDesde	2000
ObtenerPersonasDadoldArea	1000
ObtenerPersonasDadoldCargo	2000
ObtenerPersonasDadoldCategoria	2000

Tabla 1. Tiempo exigido para cada uno de los métodos.

De acuerdo al tiempo exigido y al tiempo de respuesta de cada uno de los métodos (Ver anexo Tabla 2) el Nivel de Servicio para cada uno de los métodos sería (Ver Tabla 2):

Métodos	$N(T(w) < T0)$	N(RP)	NS(RP)
ObtenerAreasTercerizadas	10	10	1
ObtenerCategorias	10	10	1
ObtenerCumpleTercerosDadoDiayMes	1	10	0.1
ObtenerPersonaDadoCI	3	10	0.3
ObtenerPersonaDadoldExpediente	10	10	1
ObtenerPersonasAlta	0	10	0
ObtenerPersonasAltaDesde	5	10	0.5
ObtenerPersonasBaja	0	10	0
ObtenerPersonasBajaDesde	8	10	0.8
ObtenerPersonasDadoldArea	10	10	1
ObtenerPersonasDadoldCargo	7	10	0.7
ObtenerPersonasDadoldCategoria	8	10	0.8

Tabla 2. Nivel de Servicio de cada uno de los métodos.

Capítulo 3: Validación de la Solución Propuesta

Del resultado de esta métrica se puede inferir que en el servicio Registro de Personal existen métodos que no cumplen con el tiempo exigido.

Para aplicar la métrica Grado de Conexión se tiene en cuenta que la misma determinará el grado de conexión de la red en la Arquitectura Orientada a Servicios por lo que se tienen en cuenta todos los servicios.

$$GC(G) = \frac{N(L)}{N(W) \times (N(W) - 1)}$$

Sustituyendo:

$$GC(G) = \frac{1}{6 \times (6 - 1)}$$

$$GC(G) = 0.033$$

El Grado de Conexión de la red en la Arquitectura Orientada a Servicios en la UCI es bastante escaso debido al desarrollo de la misma actualmente en la universidad. Solo existen dos servicios conectados.

La métrica Entrantes para el caso del servicio Registro de Personal sería 0, ya que actualmente ninguno de los servicios desarrollados realiza ninguna llamada a éste.

Para calcular Grado de Concentración que es la métrica que proporciona una visión del grado de concentración de la red se tendría que obtener primero Entrantes de cada uno de los 6 servicios con que cuenta la arquitectura en la Universidad pero debido a que actualmente el único servicio en el que la métrica Entrantes obtiene un resultado distinto de 0 es Assets Sistema de Capital Humano, por lo tanto el Grado de Concentración de la red sería:

$$Entrantes(w) = N(v \in W: (v, w) \in L)$$

Sustituyendo:

$$Entrantes(A) = 1$$

Capítulo 3: Validación de la Solución Propuesta

El Grado de Concentración de la red sería:

$$GC(G) = \sqrt{6 \frac{\sum_{w \in W} Entrantes^2(w)}{2N^3(W)3N^2(W) + N(W)}}$$

Sustituyendo:

$$GC(G) = \sqrt{6 \frac{1}{46662}}$$

$$GC(G) = 0.011$$

La métrica Salientes para el servicio Registro de Personal sería:

$$Salientes(w) = N(v \in W: (w, v) \in L)$$

Sustituyendo:

$$Salientes(RP) = 1$$

El Grado de Desconcentración de la red sería:

$$GD(G) = \sqrt{6 \frac{\sum_{w \in W} Salientes^2(w)}{2N^3(W)3N^2(W) + N(W)}}$$

Sustituyendo:

$$GD(G) = \sqrt{6 \frac{1}{46662}}$$

$$GD(G) = 0.011$$

Capítulo 3: Validación de la Solución Propuesta

El valor del grado de Concentración y de Desconcentración es bastante pequeño con un valor de 0.011 en ambas métricas, estos valores se deben fundamentalmente a la pobre conexión que existe entre los servicios actualmente.

La carga subestimada de Registro de Personal sería:

$$CS(w) = N(v \in W: Dependencia(v, w))$$

Sustituyendo:

$$CS(RP) = 1$$

La Carga Estructural Generada sería:

$$CEG(w) = \sum_{v \in W: Directa(w, v)} (1 + CEG(v))$$

Sustituyendo:

$$CEG(RP) = \sum_{v \in W: Directa(w, v)} (1 + 1)$$

$$CEG(RP) = 2$$

La métrica Recarga sería:

$$RC(w) = CEG(w) \times CS(w)$$

Sustituyendo:

$$RC(RP) = CEG(RP) \times CS(RP)$$

$$RC(RP) = 2 \times 1$$

$$RC(RP) = 2$$

Para obtener el valor de la métrica Probabilidad de Servicio Disponible se tienen en cuenta todos los métodos del servicio Registro de Personal. **(Ver tabla 3)**

Capítulo 3: Validación de la Solución Propuesta

Métodos	NC	N	PSD(RP)
ObtenerAreasTercerizadas	10	10	1
ObtenerCategorias	10	10	1
ObtenerCumpleTercerosDadoDiayMes	10	10	1
ObtenerPersonaDadoCI	10	10	1
ObtenerPersonaDadoldExpediente	10	10	1
ObtenerPersonasAlta	10	10	1
ObtenerPersonasAltaDesde	10	10	1
ObtenerPersonasBaja	10	10	1
ObtenerPersonasBajaDesde	10	10	1
ObtenerPersonasDadoldArea	10	10	1
ObtenerPersonasDadoldCargo	10	10	1
ObtenerPersonasDadoldCategoria	10	10	1

Tabla 3. Probabilidad de Servicio Disponible de cada uno de los métodos.

La métrica Tiempo Esperado de Respuesta para cada uno de los métodos sería (Ver Tabla 4):

Métodos	$\sum_{k \in \{\text{Últimas } N \text{ ejecuciones}\}} T_{S_k}^w$	N	TER(RP) (ms)
ObtenerAreasTercerizadas	2831	10	283.1
ObtenerCategorias	1220	10	120
ObtenerCumpleTercerosDadoDiayMes	23001	10	2300.1
ObtenerPersonaDadoCI	22495	10	2249.5
ObtenerPersonaDadoldExpediente	4146	10	414.6
ObtenerPersonasAlta	54570	10	5457
ObtenerPersonasAltaDesde	9329	10	932.9
ObtenerPersonasBaja	54902	10	5490.2
ObtenerPersonasBajaDesde	7646	10	764.6
ObtenerPersonasDadoldArea	5617	10	561.7
ObtenerPersonasDadoldCargo	20575	10	2057.5
ObtenerPersonasDadoldCategoria	14178	10	1417.8

Tabla 4. Tiempo Esperado de Respuesta de cada uno de los métodos.

La métrica Número de Llamadas será aplicada al método ObtenerPersonaDadoCI del servicio Registro de Personal que realiza llamadas al método ObtenerProvincias del servicio Assets Sistema de Capital Humano (A).

$$NL(RP, A) = \frac{Na}{N}$$

Capítulo 3: Validación de la Solución Propuesta

Sustituyendo:

$$NL(RP, A) = \frac{10}{10}$$
$$NL(RP, A) = 1$$

Para aplicar la métrica **Tiempo Propio de Respuesta** se calculó primero el Tiempo Esperado de Respuesta del método ObtenerProvincias del servicio Assets Sistema de Capital Humano de acuerdo a los valores (**Ver anexo Tabla 3**).

$$TPR(w) = TER(w) \times \sum_{(w,v) \in L} NL(w, v) \times TER(v)$$

Sustituyendo:

$$TPR(RP) = TER(RP) \times \sum_{(w,v) \in L} NL(RP, A) \times TER(A)$$
$$TPR(RP) = 2249.5 \times \sum_{(w,v) \in L} 1 \times 134$$
$$TPR(RP) = 301433ms$$

Las métricas Demanda Total, Demanda Interna, Demanda Externa, Probabilidad de Llamada y Número Relativo de Llamadas no pueden ser aplicadas debido al actual desarrollo de la arquitectura en la universidad.

La métrica Carga Real Generada

$$CRG(RP) = \sum_{(w,v) \in L} NL(w, v) \times (1 + CRG(v))$$

Sustituyendo:

$$CRG(RP) = 1$$

Los resultados obtenidos en cada una de las métricas se deben fundamentalmente a la escasa conexión que existe hoy entre los servicios de la arquitectura en la Universidad. Esto es ocasionado

Capítulo 3: Validación de la Solución Propuesta

producto a que la Arquitectura Orientada a Servicios es algo novedoso y actualmente se desarrolla en la UCI. Se propone aplicar las métricas cuando la arquitectura alcance un nivel de desarrollo mayor, es decir cuando todos sus servicios estén conectados.

3.7 Conclusiones del capítulo

En este capítulo se propone, de alguna manera validar la propuesta de solución planteada en el capítulo anterior, se llegó al consenso de aplicar las métricas al servicio Registro de Personal de la Arquitectura Orientada a Servicios en la Universidad de las Ciencias Informáticas siendo el servicio mencionado anteriormente el único que actualmente llama a otro servicio Assets Sistema de Capital Humano, con ellas se aporta un resultado válido, desde el punto de vista matemático, por lo que se infiere que es factible aplicar la propuesta a las aplicaciones desarrolladas con una SOA y a su vez a los servicios que la componen.

CONCLUSIONES

La interoperabilidad de la información es un requisito para una mayor eficiencia en la gestión de los servicios y negocios contribuyendo a elevar su calidad, actualmente las instituciones trabajan de forma mutua apoyándose en las tecnologías de la información y las comunicaciones. De ahí la necesidad de medir estos aspectos.

La investigación cumpliendo con los objetivos trazados aporta elementos que posibilitan el seguimiento de esta temática.

Se realizó un estudio de las métricas de calidad vigentes en el mundo para determinar cuales se podrían adaptar para medir una Arquitectura Orientada a Servicios, además de otras lecturas e investigaciones para definir y proponer un sistema de métricas que permita medir la calidad de los servicios desarrollados con una Arquitectura Orientada a Servicios en la Universidad de las Ciencias Informáticas.

La validación de las métricas definidas se realizó al servicio Registro de Personal obteniéndose resultados que permiten una visión del desarrollo de la arquitectura en la Universidad, y continuar con el uso de las mismas como uno de los objetivos fundamentales para obtener una Arquitectura Orientada a Servicios de alta calidad.

Se considera que el tema debe profundizarse para perfeccionar la evolución de aplicaciones desarrolladas en una SOA en la Universidad de las Ciencias Informáticas, en aras de mejorar la calidad de servicios y procesos de negocios.

RECOMENDACIONES

Después de haber realizado un estudio detallado de la Arquitectura Orientada a Servicios en la UCI se emiten recomendaciones dirigidas a mejorar el proceso de desarrollo de la arquitectura en la Universidad.

- Crear un ambiente propicio para el uso de métricas en la Arquitectura Orientada a Servicios en la UCI, capacitando al personal sobre el tema.
- Aplicar las métricas al resto de los servicios existentes con el fin de lograr una arquitectura de alta calidad.
- Continuar con el estudio de las métricas de calidad para medir otros aspectos de la arquitectura.
- Automatizar el proceso de aplicación de métricas para que se cuente con un sistema automatizado para la eficiencia del control de la calidad de la arquitectura.

BIBLIOGRAFÍA

Trabajos citados

Angel Cervera Paz. [Online]

<http://www.willydev.net/InsiteCreation/v1.0/descargas/articulos/general/mcall.aspx>.

Camilo, José. 2007. GestioPolis.com. [Online] 2007. <http://www.gestiopolis.com/delta/term/TER433>.

CIENTEC . CIENTEC Respuestas Rápidas, Soluciones Permanentes. [Online]

<http://www.cientec.com/Management/Management12>.

Gartner. 2007. Blog de Jaime Cid. [Online] 4 7, 2007.

http://blogs.sun.com/jaimecid/entry/gartner_webservices_soa.

Giraldo. Métricas, Estimación y Planificación en Proyectos de Software. [Online] www.WilyDev.Net.

Heidi González Doria. Las Métricas de Software y su Uso en la Región. [Online]

http://catarina.udlap.mx/u_dl_a/tales/documentos/lis/gonzalez_d_h/capitulo1.pdf .

Ludisley la Torre , Mariela Cepero . monografías.com. [Online]

<http://www.monografias.com/trabajos55/proceso-de-desarrollo-software/proceso-de-desarrollo-software2.shtml>.

María Pérez, Luis E. Mendoza, Anna C. Grimán, . Modelo para estimación de la calidad de un Web Service. [Online]

http://www.willydev.net/InsiteCreation/v1.0/descargas/soa/willydev_estimaciondecalidadws.pdf

Mario García, Saidel López. 2007. Entérate en línea. [Online] 9 2007.

<http://www.enterate.unam.mx/artic/2007/noviembre/art4.html>.

Mind. 2006. [Online] 2006. <http://www.mind.com.co/saleslogix/pdfs/FolletoBAM.pdf>.

MORENO, MAURICIO MONSALVE. 2006. Métricas para la Interoperabilidad de la Información en el Gobierno Electrónico. [Online] 2006. <http://www.dcc.uchile.cl/~mnmonsal/memoria/memoria.pdf>.

Navarro, Antonio. 2003. [Online] 2003.

http://www.fdi.ucm.es/profesor/anavarro/4._Proceso_de_software_y_metricas_de_proyectos.pdf..

Pressman, Roger. 2005. Ingeniería de Software. Un enfoque práctico. La Habana : Feliz Varela, 2005.

Tibco. El papel de un bus de servicios empresariales (ESB) en una SOA. [Online]

http://www.tibco.com/international/spain/resources/es_esb_for_soa.pdf.

Wikipedia la enciclopedia libre. [Online] <http://es.wikipedia.org/wiki/>.

Trabajos consultados

ACUERDO DE NIVEL DE SERVICIO EN CONTRATACIÓN (SLA). [Online]

<http://www.contratosinformaticos.com/sla/>.

Angel Cervera Paz. [Online]

<http://www.willydev.net/InsiteCreation/v1.0/descargas/articulos/general/mcall.aspx>.

CALIDAD DEL SOFTWARE. [Online] [Cited:] <http://usuarios.lycos.es/ronaldy/siguiente%201.htm>.

Camilo, José. 2007. GestioPolis.com. [Online] 2007. <http://www.gestiopolis.com/delta/term/TER433>.

CIENTEC . CIENTEC Respuestas Rápidas, Soluciones Permanentes. [Online]

<http://www.cientec.com/Management/Management12>.

2006. COMPUTERWORLD. [Online] 3 24, 2006. <http://www.idg.es/computerworld/articulo>.

CONTROL DE CALIDAD DEL SOFTWARE. [Online]

<http://maestros.its.mx/jcabrera/Calidad/Unidad4.pdf>.

Gartner. 2007. Blog de Jaime Cid. [Online] 4 7, 2007.

http://blogs.sun.com/jaimecid/entry/gartner_webservices_soa.

Giraldo. Métricas, Estimación y Planificación en Proyectos de Software. [Online] www.WilyDev.Net.

Heidi González Doria. Las Métricas de Software y su Uso en la Región. [Online]

http://catarina.udlap.mx/u_dl_a/tales/documentos/lis/gonzalez_d_h/capitulo1.pdf .

2007. Infoworld. [Online] 6 27, 2007. http://www.iworld.com.mx/iw_TestCenter_read.asp.

2008. Introducción a la Creación de Servicios Web Con .NET. [Online] 3 17, 2008.

http://www.procodigo.com/art10_Introduccion_a_la_Creacion_de_Servicios_Web_Con_NET.htm .

Ludisley la Torre , Mariela Cepero . monografías.com. [Online]

<http://www.monografias.com/trabajos55/proceso-de-desarrollo-software/proceso-de-desarrollo-software2.shtml>.

María Pérez, Luis E. Mendoza, Anna C. Grimán, . Modelo para estimación de la calidad de un Web Service. [Online]

http://www.willydev.net/InsiteCreation/v1.0/descargas/soa/willydev_estimaciondecalidadws.pdf.

Mario García, Saidel López. 2007. Entérate en línea. [Online] 9 2007.

<http://www.enterate.unam.mx/artic/2007/noviembre/art4.html>.

Mind. 2006. [Online] 2006. <http://www.mind.com.co/saleslogix/pdfs/FolletoBAM.pdf>.

MORENO, MAURICIO MONSALVE. 2006. Métricas para la Interoperabilidad de la Información en el Gobierno Electrónico. [Online] 2006. <http://www.dcc.uchile.cl/~mnmonsal/memoria/memoria.pdf>.

Bibliografía

Navarro, Antonio. 2003. [Online] 2003.

http://www.fdi.ucm.es/profesor/anavarro/4._Proceso_de_software_y_metricas_de_proyectos.pdf..

Pedro Urra. 2008. Cubahora . [Online] 2 1, 2008. http://cubahora.co.cu/index.php?tpl=principal/ver-noticias/ver-not_ptda.tpl.html&newsid_obj_id=1024020.

Pressman, Roger. 2005. Ingeniería de Software. Un enfoque práctico. La Habana : Feliz Varela, 2005.

SOA agenda. [Online] <http://soaagenda.com/journal/articulos/orquestadores-y-soa>.

2007. SOA con capacidades semánticas. [Online] 9 8, 2007. <http://www.mkmpi.com/mkmpi.php?article913>.

soapUI; the Web Services Testing tool. [Online] <http://www.soapui.org>..

Spri. [Online] <http://www.spri.es/ddweb/inicio/cursos/DD/ct/UNIDAD%201.pdf>.

2006. Tecnologías de redes LAN y WAN con Calidad de Servicios. [Online] 9 28, 2006.

<http://www.ieee.org.ar/noticiasdetalle.asp?IDNoticia=161> .

Tibco. El papel de un bus de servicios empresariales (ESB) en una SOA. [Online]

http://www.tibco.com/international/spain/resources/es_esb_for_soa.pdf.

Wikipedia la enciclopedia libre. [Online] <http://es.wikipedia.org/wiki/>.

ANEXOS

Tabla1. Sistema de métricas propuestas

Nombre de la métrica	Propósito	Fórmula
Nivel de Servicio	Medir la probabilidad de que un servicio Web cumpla un tiempo exigido T0.	$NS(w) = \frac{N(T(w) < T0)}{N(w)}$
Grado de Conexión	Indicar cuán conectada está la red.	$GCo(G) = \frac{N(L)}{N(W) \times (N(W) - 1)}$
Entrantes	Esta métrica mide la carga que genera un servicio Web al ser ejecutado.	$Entrantes(w) = N(v \in W: (v, w) \in L) \\ N(v \in W: Directa(v, w))$
Grado de Concentración	Mide el grado de concentración de la red	$GC = \sqrt{6 \frac{\sum_{w \in W} Entrantes^2(w)}{2N^3(W)3N^2(W) + N(W)}}$
Salientes	Esta métrica mide la carga que genera un servicio Web al ser ejecutado	$Salientes(w) = N(v \in W: (w, v) \in L) \\ N(v \in W: Directa(w, v))$
Grado de Desconcentración	Mide el grado de desconcentración de la red.	$GD(G) = \sqrt{6 \frac{\sum_{w \in W} Salientes^2(w)}{2N^3(W)3N^2(W) + N(W)}}$
Carga Subestimada	Mide la cantidad de servicios que dependen de un determinado servicio.	$CS(w) = N(v \in W: Dependencia(v, w))$
Carga Estructural	Esta métrica mide la carga generada por un	$CEG(w) = \sum_{v \in W: Directa(w, v)} (1 + CEG(v))$

Anexos

Generada	servicio Web al ejecutarse.	
Recarga	Indica la cantidad potencial de ejecuciones redundantes que son consecuencia de la ejecución de un servicio Web.	$RC(w) = CEG(w) \times CS(w)$
Probabilidad de Servicio Disponible	Mide la probabilidad de que un servicio Web responda correctamente	$PSD(w) = \frac{NC}{N}$
Tiempo Esperado de Respuesta	Permite conocer cuánto se demora realmente el servicio Web sin considerar la carga generada	$TER(w) = \frac{\sum_{k \in \{\text{Últimas } N \text{ ejecuciones}\}} T_{S_k}^w}{N}$
Número de Llamadas	Mide el número esperado de veces que un servicio w realiza una llamada al servicio v por ejecución.	$NL(w, v) = \frac{Na}{N}$
Tiempo Propio de Respuesta	Mide el tiempo "propio" de respuesta, aquel no explicado por las llamadas realizadas por el servicio Web.	$TPR(w) = TER(w) \times \sum_{(w,v) \in L} NL(w, v) \times TER(v)$
Demanda Total	Indica la frecuencia de accesos al servicio Web en un intervalo de tiempo I .	$D(w) = NA$

Anexos

Demanda Interna	Mide la demanda explicada por las llamadas realizadas desde el interior de la red.	$DI(w) = \sum_{v \in W: Directa(v,w)} D(v) \times NL(v,w)$
Demanda Externa	Mide la demanda que no es explicada por las llamadas realizadas al interior de la red.	$DE(w) = D(w) \times DI(w)$
Probabilidad de Llamada	Mide la probabilidad que la llamada del servicio Web w al servicio Web v sea realizada durante una ejecución del servicio Web w.	$PLL(w,v) = \frac{N \times Nn}{N}$
Número Relativo de Llamadas	Estima la cantidad esperada de ejecuciones de l que un servicio hace, dado que va a usar la dependencia l.	$NRLL(l) = \frac{NL(l)}{PLL(l)} - 1$
Carga Real Generada	Indica la demanda total generada sobre la red, por una ejecución del servicio.	$CRG(w) = \sum_{(w,v) \in L} NL(w,v) \times (1 + CRG(v))$

Anexos

Tabla 2. Tiempo de Respuesta de cada uno de los métodos en 10 llamadas realizadas

# Llamadas	1	2	3	4	5	6	7	8	9	10
Métodos										
ObtenerAreasTerce rizadas	488	47	487	77	391	110	470	191	534	36
ObtenerCategorias	108	55	97	74	83	117	91	80	400	115
ObtenerCumpleTer cerosDadoDiayMes	2985	2185	2159	1869	2294	2077	2120	2171	2358	2790
ObtenerPersonaDa doCI	2076	2014	2474	1841	1852	2022	2491	1847	2863	3015
ObtenerPersonaDa doldExpediente	573	344	435	333	395	349	364	371	495	487
ObtenerPersonasAl ta	2682	7111	2960	2710	3175	13006	8253	6380	2744	5549
ObtenerPersonasAl taDesde	852	289	1162	694	813	670	1150	1260	1219	1220
ObtenerPersonasB aja	2130	8849	3287	6330	3879	8623	3633	5026	4740	8405
ObtenerPersonasB ajaDesde	1048	575	660	601	1448	809	598	725	581	601
ObtenerPersonasD adoldArea	486	417	623	548	358	386	739	591	713	756
ObtenerPersonasD adoldCargo	2887	1864	1857	1978	2127	1871	1852	2398	1884	1857
ObtenerPersonasD adoldCategoria	545	483	780	460	967	285	822	1225	1702	649

Tabla3. Tiempo de Respuesta del método ObtenerProvincia de Assets Sistema de Capital Humano en 10 llamadas realizadas

Método	1	2	3	4	5	6	7	8	9	10
ObtenerProvincia	135	38	55	85	53	61	89	458	89	150

GLOSARIO

A continuación se presenta por orden alfabético algunos de los términos utilizados en la realización de este documento y que puedan crear confusión en la comprensión del mismo.

D

Debuggear Corrección de errores

G

Gobernabilidad SOA suele ser un grupo de herramientas en conjunto con diversas metodologías, guías y buenas prácticas.

L

Legacy es una aplicación basada en tecnologías y hardware más viejo, tales como mainframes, la cual continúa brindando servicios esenciales para una organización.

M

MathML o Mathematical Markup Language es un lenguaje de marcado basado en XML, cuyo objetivo es expresar notación matemática de forma que distintas máquinas puedan entenderla, para su uso en combinación con XHTML en páginas Web, y para intercambio de información entre programas de tipo matemático en general.

S

SEI Software Engineering Institute (SEI) es un instituto para desarrollar modelos de evaluación y mejora en el desarrollo de software.

SOAP (siglas de Simple Object Access Protocol) es un protocolo estándar que define cómo dos objetos en diferentes procesos pueden comunicarse por medio de intercambio de datos XML. SOAP es uno de los protocolos utilizados en los servicios Web.

SVG (Scalable Vector Graphics) es un lenguaje para describir gráficos vectoriales bidimensionales, tanto estáticos como animados en XML.

!

ISO (en inglés, International Organization for Standardization) (**Organización Internacional para la Estandarización**), es el organismo encargado de promover el desarrollo de normas internacionales de fabricación, comercio y comunicación para todas las ramas industriales a excepción de la eléctrica y la electrónica. Su función principal es la de buscar la estandarización de normas de productos y seguridad para las empresas u organizaciones a nivel internacional.

X

XHTML, acrónimo inglés de **Extensible Hypertext Markup Language** (lenguaje extensible de marcado de hipertexto), es el lenguaje de marcado pensado para sustituir a HTML como estándar para las páginas Web. XHTML es la versión XML de HTML, por lo que tiene, básicamente, las mismas funcionalidades, pero cumple las especificaciones, más estrictas, de XML.

XPath (XML Path Language) es un lenguaje que permite construir expresiones que recorren y procesan un documento XML además permite buscar y seleccionar teniendo en cuenta la estructura jerárquica del XM.