

**Universidad de las Ciencias Informáticas
Facultad 2**



Título:

**Propuesta de métricas para medir y evaluar la reusabilidad
de un producto Web Services.**

**Trabajo de Diploma para optar por el título de
Ingeniero en Ciencias Informáticas**

Autor(es):

**Marilyn Amador Medero
Lisbet Bertoli Ramos**

Tutor(es):

Lic. Elizabeth Betancourt Herrera

**Ciudad de la Habana, Junio, 2008
“Año 50 de la Revolución”**

“ Si alguna vez nuestro trabajo nos pareciera bueno, debemos luchar por hacerlo mejor, luchar por hacerlo perfecto, sabiendo de antemano que no hay obra humana totalmente perfecta. ”

Fidel Castro Ruz.

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 _____.

Lisbet Bertoli Ramos

Firma del Autor

Marilyn Amador Medero

Firma del Autor

Lic. Elizabeth Betancourt Herrera

Firma del Tutor

DATOS DE CONTACTO

Email: elizabeth@softel.cu

La Lic. Elizabeth Betancourt Herrera es especialista de la dirección de Desarrollo de la Empresa SOFTEL, graduada de Licenciatura en Cibernética-Matemática en el año 1994. Ha desarrollado diferentes proyectos de gestión en la esfera de la salud y empresarial. Se ha desempeñado como jefe de proyecto en estas áreas. Ha participado en diferentes eventos científicos-técnicos como salud para todos e informática en diferentes ediciones. Ha recibido diferentes cursos de post-gradados y actualmente cursa la maestría de gestión de proyectos. Se ha desempeñado durante 4 años como profesor adjunto de la Sede Universitaria de 10 de Octubre.

AGRADECIMIENTOS

A mis padres, especialmente a mi mamá, por su apoyo incondicional y confianza en mí, sin esto no hubiese podido llegar hasta donde estoy hoy.

A mi hermana por todo su cariño y estar siempre pendiente de mí.

A mi tío Pipo por ser una guía para mí, por sus consejos y por ayudarme tanto.

A mi novio por su cariño, comprensión y por apoyarme siempre.

A todos mis tios, primos y demás familiares por todo el cariño que me han dado, por preocuparse siempre por mí, por guiarme en cada etapa de mi vida.

A Elizabeth, nuestra tutora por su dedicación y paciencia.

A Lisbet, mi compañera de tesis por su dedicación y toda su ayuda para hacer posible la realización de este trabajo.

A la Revolución por darnos los medios para convertirnos en profesionales.

A todos mis compañeros durante estos 5 años, y a todos los que de una forma u otra han contribuido a que este sueño se haga realidad.

Marifyn

AGRADECIMIENTOS

A mi mamá, mi papá y mi abuela por su ayuda, amor y comprensión. Por estar siempre a mi lado apoyándome y dándome aliento para salir adelante.

A mis hermanas y mi hermanito por formar parte de mi corazón y ser tan lindos conmigo, los quiero mucho.

A toda mi familia y en especial a mis tías y primas por ayudarme tanto y por estar siempre al pendiente de mí.

A mi novio Lisbet por su cariño, su amor, su comprensión, por todo el apoyo que me ha brindado y por hacerme feliz.

A mi mejor amiga Yelíne por estar siempre a mi lado en las buenas y en las malas a pesar de la distancia.

A nuestra tutora Elizabet por todo el apoyo y la ayuda que nos ha brindado, por su paciencia y profesionalismo, muchas gracias.

A mi compañera de tesis Marilyn pues sin ella no hubiese sido posible la realización de este trabajo, le agradezco su paciencia y dedicación.

A la Revolución por darme la oportunidad de formarme como profesional en esta maravillosa universidad donde han transcurrido los mejores años de mi vida.

A mis compañeros de aula durante estos 5 años, especialmente a Yisi, Mila, Sheila y Yuri, por ser mis amigas y confidentes, por apoyarme y aconsejarme, por compartir tantos momentos lindos.

A todos les doy las gracias...

...porque sin ustedes no hubiese podido ser lo que soy.

Lisbet

DEDICATORIA

*A mi mamá por todo su amor y dedicación,
A mi sobrinito y a todos los que han confiado en mí.*

Marifyn

A mis padres, mi abuela linda, mis hermanas y a mi hermanito, por ser los seres más importantes en mi vida.

A todas aquellas personas que de una forma u otra han hecho posible que este sueño se haga realidad.

Lisbet

RESUMEN

La medición de la reusabilidad es un aspecto importante en la producción de software y especialmente cuando se ponen en práctica nuevas tecnologías. La reusabilidad siempre ha sido perseguida por los diferentes paradigmas dentro de la Ingeniería de Software como es el caso de la Programación Orientada a Objetos, la Programación Orientada a Componentes, la Arquitectura Orientada a Servicios, entre otros. Este último paradigma junto a los Servicios Web tiene entre sus beneficios una mejora de eficiencia en el desarrollo de aplicaciones a través de la reusabilidad.

Muchas empresas cubanas han ganado cierta experiencia en el desarrollo de aplicaciones y componentes bajo una Arquitectura Orientada a Servicios, específicamente utilizando la tecnología de servicios Web, la cual se ha convertido en punto de partida para muchas investigaciones, de ahí el surgimiento de este trabajo de diploma que tiene como objetivo, proponer un conjunto de métricas para medir y evaluar la reusabilidad de un producto web services en aras de contar con componentes robustos y con un aval de calidad en la conformación de soluciones de negocio.

En este trabajo se proponen un conjunto de métricas para medir la consistencia, legibilidad, autodescriptividad, tamaño, comprensibilidad, criterios que en conjunto dan una medida de la reusabilidad de un producto web service. Estas métricas propuestas se aplican a dos casos de estudio.

PALABRAS CLAVES

Reusabilidad, métricas, web services, legibilidad, comprensibilidad, consistencia, autodescriptividad, tamaño.

TABLA DE CONTENIDOS

AGRADECIMIENTOS	I
AGRADECIMIENTOS	II
DEDICATORIA	III
RESUMEN	IV
INTRODUCCIÓN	1
CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA.....	7
1.1. Introducción	7
1.2. Reseña histórica de la reutilización de software	7
1.3. Reutilización	8
1.4. La reutilización en los paradigmas de programación.....	11
1.5. Métricas de reutilización.....	14
1.5.1. Métricas Orientadas a Objeto.....	16
1.5.2. Métricas Orientadas a Componentes.....	19
1.5.3. Métricas adaptadas a Servicios Web.....	22
1.6. Reutilización de software en Cuba.....	25
1.7. Conclusiones	26
CAPÍTULO 2: DIAGNÓSTICO DE LA SITUACIÓN ACTUAL.....	27
2.1. Introducción	27
2.2. Análisis estadístico.....	27
2.3. Conclusiones	40
2.4. Recomendaciones.....	40
CAPÍTULO 3: PROPUESTA.....	41
3.1. Introducción	41
3.2. Estructura del documento WSDL. Propuesta de métricas	41
3.2.1. Documento WSDL.....	41
3.2.2. Métricas.....	43
3.3. Conclusiones	47
3.4. Recomendaciones.....	47
CAPÍTULO 4: ANÁLISIS DE LOS RESULTADOS.....	48
4.1. Introducción	48
4.2. Aplicación a los Casos de Estudio	48
4.2.1. Caso 1: Servicio Web Catalogo_ECService	48
4.2.2. Caso 2: Servicio Web LabOnlineservice	54
4.3. Evaluación de los resultados obtenidos	64
4.3.1. Caso 1: Servicio Web Catalogo_ECService	65

TABLA DE CONTENIDOS

4.3.2. Caso 2: Servicio Web LabOnlineservice	67
4.4. Conclusiones	69
4.5. Recomendaciones.....	69
CONCLUSIONES	70
RECOMENDACIONES.....	71
REFERENCIAS BIBLIOGRAFICAS.....	72
BIBLIOGRAFÍA	74
ANEXOS	75
GLOSARIO DE TÉRMINOS.....	106

INTRODUCCIÓN

La informática en los últimos tiempos se ha convertido en parte del sustrato tecnológico del proceso de globalización en el cual está inmerso todo el mundo, lo que implica la necesidad de preparar a las nuevas generaciones para la asimilación y utilización de dicha tecnología.

El modelo cubano, como parte de la concepción revolucionaria del desarrollo económico y social, a pesar del bloqueo y las limitaciones de todo tipo que éste representa, ha logrado el conocimiento y uso de la informática en los objetivos sociales de desarrollo e inclusión humana con una altísima eficiencia económica y social, contribuyendo de manera importante a que el país alcance los objetivos del milenio en este campo con desempeño destacado en comparación con los demás países. Al mismo tiempo ha posibilitado que se creen las condiciones para que el capital humano asociado a la rama tenga y represente un importante potencial aportador de recursos para el país y para la sociedad.

Uno de los importantes pasos que se han dado con el objetivo de informatizar el país es la creación de la Universidad de las Ciencias Informáticas (UCI), la que revolucionó la educación y se ha convertido en la aspirante incansable de un desarrollo mejor para Cuba. Esta universidad no solo tiene fines académicos sino que también, en el empeño por hacer crecer la rama de la informática en el país, está vinculada a la producción en colaboración con otras empresas que desarrollan software en el país.

Las empresas de software en el mundo han orientado su desarrollo hacia componentes de negocio, específicamente con la tecnología de Web Services que le permitan ser más eficientes y eficaces en su gestión. Una empresa cubana productora de software ha dado sus primeros pasos en esta dirección y ya cuenta con un repositorio de Servicios Web desarrollados. No obstante para incrementar el uso y reuso, así como el desarrollo de nuevos Servicios Web es importante medir y evaluar el grado de reusabilidad de los mismos, por el impacto que ello tiene en la calidad del producto final y en la eficiencia del proceso de desarrollo. De esta forma además, los Servicios Web que se desarrollen contarán con un aval de calidad dado por su grado de reusabilidad y por consiguiente la empresa puede contar con Servicios Web cada vez más robustos en la conformación de soluciones de negocio. Esta situación conlleva a plantear como **problema científico** ¿Cómo medir las características o propiedades de reusabilidad de los componentes Web Services para una empresa cubana productora de software? y basar el **objeto de estudio** en las métricas de reusabilidad.

Estudiando el tema se tiene enmarcado como **campo de acción** métricas de reusabilidad para Web Services en una empresa cubana productora de software.

Este trabajo tiene como **objetivo general** proponer un grupo de métricas para medir y evaluar la reusabilidad de un Servicio Web.

Para encaminar la investigación en vista a resolver el problema planteado se propone la siguiente **hipótesis**: Si se contara con métricas para medir la reusabilidad, se tendría una evaluación de los componentes Web Services en una empresa cubana productora de software.

Variable	Dimensiones	Indicadores	Subindicadores	Índice de evaluación
Evaluación de los componentes Web Services	Reusabilidad	Experiencia	Años de experiencia	0-2 (poca) 3-8 (alguna) Más de 8 (mucha)
			Número de entidades a las que ha desarrollado Software	1 - 4 (pocas) 5 – 9 (algunas) mayor de 10 (muchas)
		Conocimiento	Conceptos de reusabilidad	Si Mencione brevemente el concepto. No
			Métricas de reusabilidad	Si ¿Cuáles? No
			Modelos de reusabilidad	Si ¿Cuáles? No
			Otros temas de reusabilidad	¿Cuál o Cuáles?

		Aplicación de Métricas	Orientada a Producto	A nivel de documentación
				A nivel de código
				A nivel de modelo
				A nivel de componente
				A nivel de clases
				A nivel de Dominio o Negocio
				Otros niveles
		Orientada a Proceso	Si No	
		Orientada a Negocio	Si No	
		Otros niveles	¿Cuáles?	
		Criterios a tener en cuenta	Información que produce	
			Ayuda en la toma de decisiones	
			Simplicidad de la métrica	
			Facilidad para recolectar y evaluar los datos.	
			Herramientas que ayuden a recolectar y evaluar los datos.	
		Otros criterios	¿Cuáles?	

		Necesidad de contar con métricas	Evaluación cuantitativa o cualitativa de cómo y cuánto se aplica la reutilización	Si No
			Métrica o procedimiento que aplica	¿Cuáles?
			Métricas para evaluar cuantitativa o cualitativamente la capacidad de reutilización de los componentes de software disponibles en la empresa	¿Cuáles?
			Datos sobre la reusabilidad de los componentes disponibles en la empresa	Si No
			Componentes adecuados para desarrollar nuevas aplicaciones	Si No

			Construcción de componentes para reutilizar	Si No
			Construcción de nuevas aplicaciones utilizando lo que ya está disponible	Si No
			Otros	¿Cuáles?
		Aplicabilidad	Herramientas para recolectar datos	Si ¿Cuáles? No
			Herramientas para evaluar datos	Si ¿Cuáles? No
			Definición del proceso	Si No
			Planificación del proceso	Si No
			Infraestructura para guardar los resultados	Si No

Se plantean entonces un grupo de **tareas de investigación** que permitirán satisfacer el objetivo planteado, las cuales se pueden resumir en las siguientes:

1. Estudio de conceptos e indicadores de reusabilidad.
2. Estudio de paradigmas dentro de la Ingeniería de Software que han tenido entre sus objetivos la reutilización.
3. Estudio de las métricas existentes para medir reusabilidad.
4. Diagnóstico de la situación actual en cuanto a la reutilización en una empresa cubana productora de software.

5. Evaluación y selección de las métricas a proponer.
6. Aplicación de las métricas a dos casos de estudio.
7. Evaluar resultados.

Para la realización de la investigación se utilizan los métodos teóricos que a través de sus procesos de análisis y síntesis permiten una representación abstracta del fenómeno y crean las condiciones para ir más allá de las características superficiales, esto es evidenciado mediante las consultas a sitios web, publicaciones, catálogos, libros y luego el análisis profundo del objeto de estudio para sintetizar las citas, apuntes, datos y resúmenes tomados.

Además se emplea la inducción-deducción para el planteamiento del objetivo y como método particular la encuesta, para conocer la situación actual de la empresa.

La tesis cuenta con cuatro capítulos. En el primer capítulo se recoge el resultado de la investigación sobre el estado del arte en los temas relacionados: reusabilidad, Programación Orientada a Objetos, Programación Orientada a Componentes, SOA, Web Services y métricas. En el segundo capítulo se realiza un análisis estadístico de los resultados arrojados por la aplicación de una encuesta, para conocer la situación en que se encuentra una empresa productora de software cubana con respecto a la reusabilidad. En el tercer capítulo se hace una propuesta de las métricas para dar solución al problema. Se realiza un análisis de cada una, adaptándolas a los servicios web. En el cuarto y último capítulo, se aplican las métricas propuestas a dos casos de estudio con el objetivo de evaluar la factibilidad de su aplicación y analizar los resultados.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

1.1. Introducción

En el presente capítulo se brinda una visión general de algunos aspectos relacionados con la reusabilidad en la industria del software, así como de paradigmas que han surgido dentro de la Ingeniería de Software con la meta de lograrla. Además se muestra una breve panorámica de la reutilización de componentes en Cuba y en el mundo.

1.2. Reseña histórica de la reutilización de software

Desde los inicios de la era de la información la reutilización ha supuesto una práctica habitual para el desarrollo de aplicaciones software. Actualmente, la reutilización del software continúa siendo un aspecto esencial en los sistemas de información actuales por el uso de componentes software, vistos como colecciones de código reutilizables que facilitan el desarrollo de estas aplicaciones. (1)

La reutilización del software no es un concepto nuevo, de hecho se lleva reutilizando software durante varias décadas. En 1968, Dough McIlroy, en la Conferencia de Ingeniería del Software de la NATO, introduce el concepto de reusabilidad. McIlroy veía este concepto como la construcción de sistemas software a partir de bloques pequeños (componentes código fuente) y utilización de técnicas automatizadas para tratar los componentes. (2)

Inicialmente el concepto de reutilización se concibe como la combinación de componentes de código almacenados en una biblioteca.

Actualmente este concepto ha evolucionado hacia la idea de que todo el conocimiento y productos derivados de la producción de software son susceptibles de ser reutilizados en la construcción de nuevos sistemas, surgiendo de esta forma el concepto de asset o de componente software reutilizable. Así, se va a entender por elemento reutilizable cualquier producto software obtenido en el ciclo de vida del software, con independencia de su nivel de abstracción. (3)

1.3. Reutilización

1.3.1. ¿En qué consiste la reutilización?

Consiste en construir nuevos sistemas informáticos a partir de cualquier información, artefacto o producto software existente.

1.3.2. Objetivos

Los objetivos que se persiguen con la reutilización son:

- ❖ Reducir el tiempo y el coste de desarrollo de nuevos productos (o mejoras).
- ❖ Incrementar la diversidad de los productos disponibles para el usuario.
- ❖ Mejorar su estandarización e interoperabilidad.

1.3.3. Escenarios

Existen varios escenarios en los que se puede hacer uso de la reutilización para lograr un mejor funcionamiento de la empresa tales como:

- ❖ Optimizar el desarrollo, evolución y mantenimiento de una familia de productos para hacerlos tan consistentes como sea posible.
- ❖ Una empresa que desarrolla aplicaciones (o partes) para muchos clientes distintos y quiere evitar desarrollar las mismas soluciones una y otra vez.
- ❖ Una empresa que tiene sistemas antiguos con conocimiento embebido importante, que quiere aprovechar ese conocimiento para crear una nueva generación de sistemas.
- ❖ Se tienen componentes potencialmente reutilizables y se querría sacar provecho de ellos en el desarrollo de nuevos sistemas.

1.3.4. Impacto en la calidad

La reutilización tiene gran impacto en la calidad.

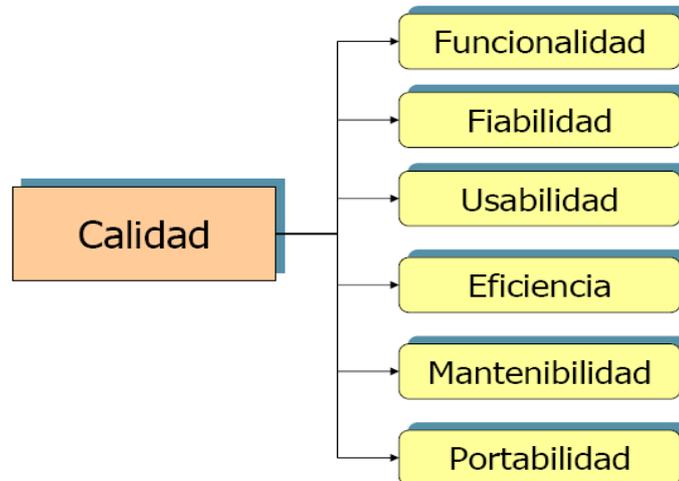


Fig.1.1. Impacto de la reutilización en la calidad

Funcionalidad

- ❖ Reusar sistemas similares como prototipos para capturar requisitos.
- ❖ En el desarrollo con reutilización se pueden descubrir nuevos requisitos durante la evaluación de componentes potenciales.
- ✓ Se corre el riesgo de sacrificar la funcionalidad correcta en favor del reuso.

Fiabilidad

- ❖ La correcta reutilización de un componente bien probado incrementa la fiabilidad del sistema.
- ❖ Usar varias veces el mismo componente aumenta la confianza en el correcto funcionamiento de dicho componente.
- ✓ Riesgo de reutilización incorrecta o fuera del alcance del componente.

Usabilidad

- ✓ Componentes con características ligeramente diferentes pueden ser inaceptables desde el punto de vista de la usabilidad.

Eficiencia

- ❖ Es más justificable un mayor esfuerzo por hacer eficiente un componente que se va a usar más de una vez.

- ✓ Los componentes específicos, hechos a medida suelen ser más eficientes que los más generales.

Mantenibilidad

- ❖ Muchas de las modificaciones a hacer en el componente ya se han incorporado o planeado.
- ✓ Las modificaciones fuera del plan previsto son más difíciles ya que la funcionalidad es más compleja.
- ✓ A veces no se dispone del código fuente en el diseño basado en componentes.

Portabilidad

- ❖ Está correlacionada con la reusabilidad del componente. (18)

1.3.5. Reutilización de componentes

1.3.5.1. Actividades de la reutilización

Cuando la reutilización está presente en un proceso de desarrollo software, este debe integrar todas las actividades necesarias para producir y reutilizar componentes software. Así se distinguen dos actividades principales dentro de la reutilización: el desarrollo para la reutilización y el desarrollo con reutilización.

1.3.5.2. Desarrollo para la reutilización

El desarrollo para la reutilización consiste en la realización de componentes que cumplen un conjunto de restricciones sobre su reutilización y su calidad. A la hora de desarrollar componentes reutilizables es fundamental centrarse en criterios de calidad, en detrimento de los costes de producción de los componentes.

Cuando se va a crear un componente reutilizable debe analizarse la variabilidad de los requisitos que satisface dicho componente, de forma que se construya como un componente genérico que pueda ser especializado en el momento de su reutilización para ajustarse a unos requisitos específicos.

1.3.5.3. Desarrollo con reutilización

Las técnicas utilizadas en el proceso de desarrollo con reutilización dependen en gran medida de los componentes que se hayan preparado en el proceso de desarrollo para la reutilización.

El desarrollo con reutilización consiste en la generación de nuevos productos software integrando elementos existentes, de forma directa o pasando por un proceso de adaptación.

Aparecen así cuatro problemas fundamentales:

- ❖ La selección y recuperación de los componentes.
- ❖ La comprensión y evaluación de los componentes.
- ❖ La adaptación de los componentes.
- ❖ La integración de los componentes. (4)

Las ventajas más importantes de la reutilización de componentes de aplicaciones son: un desarrollo más rápido, mayor calidad del software, y ahorro de costos. La reutilización de codificación en la propia empresa o la compra de componentes que ya existen reducen el coste de desarrollo a la vez que evita a los programadores tener que repetir continuamente pasos estándar en el desarrollo de software. (5)

1.4. La reutilización en los paradigmas de programación

1.4.1. Programación Orientada a Objeto (POO)

Con la meta de lograr la reutilización han surgido varios paradigmas dentro de la Ingeniería de Software, por ejemplo podemos mencionar la Programación Orientada a Objetos.

La Programación Orientada a Objetos proporciona el marco idóneo para la reutilización de las clases. Los conceptos de encapsulamiento y herencia son las bases que facilitan la reutilización. Un programador puede utilizar una clase existente y sin modificarla, añadirle nuevas características y datos. Esta operación se consigue derivando una clase a partir de la clase base existente, la nueva clase hereda las propiedades de la antigua, pero se pueden añadir nuevas propiedades.

La facilidad para reutilizar clases y en consecuencia objetos es una de las propiedades fundamentales que justifican el uso de la Programación Orientada a Objetos. Por esta razón los sistemas y en particular los lenguajes orientados a objetos suelen venir provistos de un conjunto (biblioteca) de clases predefinidas que permiten ahorrar tiempo y esfuerzo en el desarrollo de cualquier aplicación. (6)

1.4.2. Programación Orientada a Componentes (POC)

En muchas ocasiones las tecnologías tradicionales como las Orientadas a Objetos no son suficientes para solucionar los problemas planteados por el desarrollo de sistemas distribuidos y abiertos.

Comienzan a aparecer por tanto nuevos paradigmas de programación, como pueden ser la Programación Orientada a Componentes, que persigue una mejora en los procesos de construcción de aplicaciones software.

La Programación Orientada a Componentes aparece como una variante natural de la Programación Orientada a Objetos para los sistemas abiertos, en donde la POO presenta algunas limitaciones, (7) dentro de las cuales podemos destacar la dificultad a la hora de reutilizar objetos.

La POC tiene como objetivo construir un mercado global de componentes software, cuyos usuarios son los propios desarrolladores de aplicaciones que necesitan reutilizar componentes ya hechos y probados para construir sus aplicaciones de forma más rápida y robusta. (7). Las entidades básicas de la POC son los componentes, unidad de composición de aplicaciones software que posee un conjunto de requisitos y que ha de poder ser desarrollado, adquirido, incorporado al sistema y compuesto con otros componentes, de forma independiente en tiempo y espacio. La reutilización en POC se consigue mediante composición. (8)

1.4.3. Arquitectura Orientada a Servicios (SOA)

A pesar de los beneficios que aporta la Programación Orientada a Componentes para los sistemas abiertos y distribuidos aún persisten algunos problemas como es la interoperabilidad de componentes y es para darle solución a estos problemas que surge SOA.

Al contrario de las arquitecturas orientadas a objetos, SOA está formada por servicios de aplicación con acoplamiento débil y altamente interoperable.

Si hubiese que reducir a dos las virtudes de SOA en el ámbito de las Tecnologías de la Información (TI), la primera de ellas sería, sin duda, la capacidad de reutilización a gran escala del software, lo cual tiene consecuentes ventajas en calidad, costos y facilidad de despliegue de aplicaciones. Por otro lado, un segundo factor está dado por los múltiples beneficios asociados a la independencia que permite SOA entre las aplicaciones, la infraestructura y plataforma tecnológica.

SOA permite una separación clara entre diferentes niveles: presentación, lógica de negocio, acceso a datos; lo que facilita su reutilización y rápida adaptación al cambio. (9)

SOA es una forma de organizar soluciones que promueve la reutilización, el crecimiento y la interoperabilidad. No es por si sola una solución a los problemas del dominio sino un paradigma de organización y distribución que permite obtener más valor del uso de capacidades internas y aquellos bajo el control de otros. También permite expresar las soluciones en un modo que las hace más fáciles de modificar, evolucionar o probar soluciones alternativas.

En general, SOA no impone la elección de una tecnología específica en la construcción de los servicios. Aunque actualmente, los Servicios Web son la forma más ampliamente adoptada y soportada de construir servicios. (10)

1.4.4. Servicios Web

Los Servicios Web, se basan en estándares abiertos, formando sistemas débilmente acoplados o altamente independientes, consiguiendo alta reusabilidad e interoperabilidad. (11)

Adentrándonos en el mundo de los Servicios Web podemos decir que estos son una aplicación modular, contenida en sí misma y accesible a través de protocolos web. Los Web Services combinan lo mejor de los desarrollos basados en componentes y la web. Desde el punto de vista de componentes, un Web Service no tiene que revelar cómo está implementado. Sólo es necesario saber cómo accederlo y utilizar los servicios provistos por él. Desde el punto de vista de la web, son accesibles a través de protocolos standard (HTTP) completamente independientes de proveedores, plataformas y lenguajes, utilizando formatos de datos universalmente aceptados (XML). (12)

No están ligados a ningún sistema operativo o lenguaje de programación, además no necesitan usar browsers (Explorer) ni el lenguaje de especificación HTML.

El término Web Services describe una forma estandarizada de integrar aplicaciones WEB mediante el uso de XML, SOAP, WSDL y UDDI sobre los protocolos de la Internet. XML es usado para describir los datos, SOAP se utiliza para la transferencia de los datos, WSDL se emplea para describir los servicios disponibles y UDDI se utiliza para conocer cuales son los servicios disponibles. (13)

La aparición de los Servicios Web ha proporcionado un modelo diferente para el desarrollo de aplicaciones, ofreciendo la capacidad de acceder a servicios heterogéneos de forma unificada e interoperable a través de Internet. Nuevas aplicaciones web son creadas componiendo o combinando Servicios Web heterogéneos. (14)

1.5. Métricas de reutilización

Varias métricas económicas han sido propuestas para la medición del costo-beneficio de la reutilización. La métrica de Gaffney and Durek es útil para medir el costo de reuso. La reutilización es económicamente umbral de rentabilidad cuando el costo de la reutilización es igual al beneficio de la reutilización.

La métrica de reutilización económica de Gaffney y Durek es representada por

$$C = (b + E/n - 1)R + 1$$

Donde:

C= el costo relativo de desarrollo de software (Generalmente C será inferior a 1, menos es mejor).

R= la proporción de código de reuso en el proyecto ($0 \leq R \leq 1$).

b = el costo relativo a código nuevo dentro del código de reutilización existente en el proyecto (Por ejemplo, la búsqueda, la adaptación y la integración de costos).

E=el costo relativo a código nuevo dentro del desarrollo de un componente para reuso (Por ejemplo el costo para hacer código reusable).

n = el número de reusos esperados.

Existen diferentes interpretaciones acerca de como medir la reutilización. Poulin and Kain; solo tienen en cuenta el reuso externo (a través de la reutilización de ámbitos de proyectos). Por ejemplo, no cuentan la copia y pega, o características de lenguajes de programación, tales como sub-classes, y polimorfismo, como reuso. Ellos no dan crédito adicional para cuántas veces un componente es reutilizado. Por otro lado, tanto Henderson-Sellers como Frakes disponen de otra perspectiva acerca de la reutilización. En su opinión la herencia es una vía de reutilización, cuentan la reutilización interna y dan crédito por el número de veces que un componente es reutilizado. Poulin y Kain enfatizan "los beneficios industriales" del reuso, Mientras que Henderson-Sellers y Frakes hacen hincapié en la definición académica del reuso.

Dos métricas de reutilización comúnmente citadas son las de Banker y Frakes.

Banker desarrolló una métrica en un repositorio basado en el entorno CASE llamada Sistemas de Productividad Alta (HPS). Su métrica mide el nivel de reuso utilizando un porcentaje de reuso:

Porcentaje de Reuso = $(1 - \text{Nuevos objetos construidos} / \text{Total de objetos utilizados}) * 100\%$.

Métrica de nivel de Reuso

Frakes introduce la métrica de nivel de reuso. Esta métrica utiliza los niveles umbrales para filtrar los ítems que no son reutilizados con la frecuencia necesaria, por ejemplo si en el nivel umbral 3, un ítem tendría que ser llamado al menos 4 veces para ser contado como reusable. Esta métrica establece diferencias entre los niveles de reuso interno y los niveles de reuso externo.

Nivel de reuso interno = IU/T

Nivel de reuso externo = EU/T

Total de niveles de reuso = Nivel de reuso interno + Nivel de reuso externo

Donde:

IU = número de ítems externos reusados, es mayor que el nivel umbral externo.

EU = número de ítems internos reusados, es mayor que el nivel umbral interno.

T = número total de ítems en el sistema. Ambos internos y externos.

Cada ítems reusado (tales como IU y EU) solo tienen un valor entre 0 y 1. Si el ítem es reusado más que el nivel umbral, siempre será 1, de otro modo será 0. Esto significa que esta métrica no tiene en cuenta el número de veces que un ítem es usado.

Esta métrica de nivel de reuso usa los ítems en lugar de las líneas de código (LOC), ya que cada ítem sería diferente, algunos ítems podrían ser muy grandes y otros muy pequeños. Otra versión de esta métrica asigna un peso a cada uno de los ítems dependiendo del tamaño. También permite diferentes niveles umbral para el reuso interno y externo. Esto significa que, a fin de comparar resultados entre los proyectos, los niveles umbrales deben ser los mismos. De lo contrario, proyectos con valores umbrales pequeños aparecerán mejor. (15)

1.5.1. Métricas Orientadas a Objeto

La familia de métricas de Chidamber y Kemerer.

Métrica	Definición	Propiedades
Weighted methods per class (WMC)	<p>Considérese una clase C_1 con los métodos M_1, M_2, \dots, M_n. Sea C_1, C_2, \dots, C_n la complejidad estática de los métodos. Entonces:</p> $WMC = \sum_{i=1}^n C_i$ <p>La complejidad estática se puede medir de muchas maneras, siendo una de ellas CC(McCabe, 1976).</p>	Complejidad
Depth of Inheritance tree (DIT)	<p>La métrica DIT de una clase A es su profundidad en el árbol de herencia. Si A se encuentra en situación de herencia múltiple la longitud máxima hasta la raíz será el DIT.</p>	Herencia
Number of children (NOC)	<p>NOC de una clase es el número de subclases inmediatamente subordinadas a una clase en la jerarquía.</p>	Herencia
Coupling between object classes (CBO)	<p>CBO de una clase es el número de clases con las que está acoplada. Una clase está acoplada a otra si utiliza sus métodos o variables de instancia, excluyendo acoplamiento por herencia.</p>	Acoplamiento
Response for a class (RFC)	<p>$RFC = RS$ donde RS es el conjunto respuesta de la clase, dado que $RS = \{M\} \cup_{all\ i} \{R_i\}$ donde $\{R_i\}$ = conjunto de los métodos invocados por el método i y $\{M\}$ es el conjunto de todos los métodos de la clase. El conjunto respuesta de una clase es el conjunto de métodos que se pueden ejecutar como respuesta a un mensaje recibido por un objeto de la clase.</p>	Comunicación
Lack of cohesion in methods (LCOM)	<p>Considérese una clase C_i con n métodos M_1, M_2, \dots, M_n. Sea $\{I_j\}$ el conjunto de variables de instancia usados por el método M_j. Existen n de esos conjuntos $\{I_1\}, \dots, \{I_n\}$. Sea $P = \left\{ \left(I_i, I_j \right) \mid I_i \cap I_j = \emptyset \right\}$ y $Q = \left\{ \left(I_i, I_j \right) \mid I_i \cap I_j \neq \emptyset \right\}$. Si todos los n conjuntos $\{I_1\}, \dots, \{I_n\}$ son \emptyset entonces sea $P = \emptyset$. $LCOM = P - Q$, si $P > Q$ ó 0 en otro caso.</p>	Cohesión

Tabla 1.1. Métricas de Chidamber y Kemerer

S.R. Chidamber y C.F. Kemerer (C&K) fueron los primeros en establecer un conjunto de 6 métricas de productos específica para código OO; todas menos una se aplicaban a las clases y trataban de medir la complejidad, acoplamiento, cohesión, herencia y comunicación inter-clases. Se puede ver en la Tabla 1.1 la definición de dichas métricas y la propiedad OO con la que se relacionan.

Métricas de Li y Henry

A partir de las métricas de C&K se van proponiendo modificaciones y ampliaciones sobre ellas, éste es el caso de Li y Henry que proponen varias modificaciones y añaden cuatro nuevas métricas. La modificación más resaltable es la que se hace sobre la LCOM. Por otra parte esta nueva suite de métricas tiene un objetivo claro, que es la medición y mejora de la mantenibilidad del software OO.

Métrica	Definición	Propiedades
Message passing coupling (MPC)	MPC= número de métodos invocados en un clase.	Acoplamiento/Comunicación
Data abstraction coupling (DAC)	El número de atributos en una clase que tienen como tipo otra clase.	Acoplamiento/Abstracción
SIZE1	Es una variación de la tradicional LOC (Lineas de Código) definida específicamente para el lenguaje Ada. Obviamos la definición	Tamaño del diseño
SIZE2	SIZE2 = número de atributos + número de métodos locales.	Tamaño del diseño

Tabla 1.2. Métricas de Li y Henry

Métricas de Bansiya y Davis

Métrica	Definición	Propiedades
Design size of classes (DSC)	Número total de clases en el diseño	Tamaño del diseño
Number of Hierarchies (NOH)	Número de jerarquías de clases	Jerarquías
Average number of ancestors (ANA)	Número medio de ancestros	Abstracción
Data access Metric (DAM)	Relación entre el número de atributos privados (protegidos) y el número total de atributos de la clase.	Encapsulamiento
Direct class coupling (DCC)	Número de clases diferentes con las que una clase está relacionada. Esta métrica incluye clases que están relacionadas directamente mediante la declaración de atributos y el paso de parámetros de los métodos.	Acoplamiento
Cohesion among methods of class (CAM)	Esta métrica calcula la relación entre métodos de una clase basándose en la lista de parámetros de los métodos. Se calcula utilizando la suma de la intersección de parámetros de un método con el conjunto máximo independiente de todos los tipos de parámetros de la clase.	Cohesión
Measure of aggregation (MOA)	Mide la extensión de la relación parte/todo, mediante el uso de atributos. Es el número de declaraciones de variables cuyo tipo es definido por el usuario.	Composición
Measure of functional abstraction (MFA)	Relación del número de métodos heredados por una clase y el número total de métodos accesibles por un método miembro.	Herencia
Number of polymorphic methods (NPM)	Número de métodos que pueden mostrar comportamiento polimórfico (virtual en C++ y no final en Java)	Polimorfismo
Class interface size (CIS)	Número de métodos públicos en una clase	Comunicación
Number of methods (NOM)	Número de métodos definidos en una clase	Complejidad

(16)

Tabla 1.3 Métricas de Bansiya y Davis

1.5.2. Métricas Orientadas a Componentes

El desarrollo de software basado en componentes se caracteriza por dos actividades: el desarrollo de componentes para su reutilización, y el desarrollo de sistemas de software con la reutilización de componentes.

La actividad de desarrollo con reutilización se realiza mediante el desarrollo de software con los componentes existentes. Es necesario medir el grado de reutilización de los componentes a fin de hacer realidad la reutilización de componentes de manera eficaz.

En el departamento de Ciencias de la computación en la Universidad Waseda, Japón se define un modelo de reusabilidad para componentes. Este Modelo se estructura en tres niveles de arriba hacia abajo. Los tres términos (factor, criterio y métricas) se organizan en un análisis y evaluación de jerarquía. Los factores se utilizan a nivel de dirección, como los requisitos no funcionales de un componente. Los criterios se utilizan en el diseño (aplicación) y las métricas se utilizan en el nivel de producto.

Las métricas se utilizan para determinar los factores de calidad que afectan la reusabilidad. Un componente por sí solo tiene ciertas características que tienden a afectar su reutilización. En este modelo de reutilización, la reutilización de componentes puede descomponerse en tres factores: la claridad, la capacidad de adaptación, y la portabilidad.

La elección de los tres factores que afectan la reutilización se hace en este modelo sobre la base de un análisis de las actividades llevadas a cabo cuando se realiza la reutilización de los componentes cajas negras.

Estas actividades son las siguientes:

- ❖ La comprensión de la funcionalidad del componente, decidir si cumple con las nuevas necesidades funcionales. Las necesidades de los usuarios de alta comprensibilidad para hacer esta actividad. La comprensibilidad se define sobre la base de la estimación de esfuerzo que necesita un usuario para reconocer el concepto detrás de uno de los componentes y su aplicabilidad.
- ❖ La adaptación de los componentes a las necesidades funcionales del nuevo sistema. Las necesidades de los usuarios de alta capacidad de adaptación para

hacer esta actividad. Adaptabilidad es la facilidad con la que un componente puede ser adaptado para cumplir las exigencias que difieren de las que tenía cuando fue originalmente desarrollado.

- ❖ Portar el componente a un nuevo entorno. Las necesidades de los usuarios de alta portabilidad para hacer esta actividad. Portabilidad es la facilidad con la que el software puede ser transferido de un entorno a otro.

Definición de métricas de reusabilidad

De acuerdo con el modelo de reutilización mencionado, se definen cinco métricas, EMI, RCO, RCC, SCCr, y SCCp, para la medición de la existencia de meta-información, observabilidad, personalización y la dependencia externa de un componente JavaBeans caja negra.

Definición (EMI: Existencia de Meta-Información)

EMI(c) toma los siguientes valores según la existencia de la clase BeanInfo correspondiente al componente c:

$$EMI(c) = \begin{cases} 1 & (\text{BeanInfo class exists}) \\ 0 & (\text{otherwise}) \end{cases}$$

Intervalo de confianza: [0.5, 1.0]

Consideración: Si el valor de EMI(c) es 1, los usuarios de c pueden entender fácilmente como usar c, y que es lo que el desarrollador de c asume.

Definición (RCO: Tasa de observabilidad de componentes)

RCO(c) es el porcentaje de las propiedades legibles en todos los campos implementados en la clase fachada de un componente c.

$$RCO(c) = \begin{cases} \frac{Pr(c)}{A(c)} & (A(c) > 0) \\ 0 & (\text{otherwise}) \end{cases}$$

Donde:

Pr(c): número de propiedades legibles en c.

A(c): número de campos en las clases fachadas de c.

Intervalo de confianza: [0.17, 0.42]

Consideración: RCO indica el grado de observabilidad del componente para los usuarios del mismo. Para entender el comportamiento de un componente desde fuera de él, su observabilidad debe ser alta. Sin embargo, existe la posibilidad de que sea difícil a los usuarios la búsqueda de una propiedad de lectura importante entre todas las propiedades de lectura cuando la observabilidad es demasiado alta.

Definición (RCC: Grado de personalización de componente)

RCC(c) Es un porcentaje de propiedades de escritura en todos los campos implementados en una clase fachada de un componente c.

$$RCC(c) = \begin{cases} \frac{P_w(c)}{A(c)} & (A(c) > 0) \\ 0 & (\text{otherwise}) \end{cases}$$

Donde:

Pw(c): número de las propiedades de escritura en c.

Intervalo de confianza: [0.17, 0.34]

Consideración: RCC indica el grado de personalización de un componente c para los usuarios del componente. Para adaptar la configuración de un componente a los requerimientos del usuario, la capacidad de personalización del mismo debe ser alta. Sin embargo una alta personalización viola la encapsulación del componente y conduce a las más grandes oportunidades de mal uso.

Definición (SCCr: libre-integridad del valor de retorno del componente).

SCCr(c): Es un porcentaje de métodos de negocio sin ningún valor de retorno en todos los métodos de negocio implementados en el componente c.

$$SCCr(c) = \begin{cases} \frac{B_v(c)}{B(c)} & (B(c) > 0) \\ 1 & (\text{otherwise}) \end{cases}$$

Donde:

Bv(c): número de métodos de negocio sin valor de retorno en c.

B(c): número de métodos de negocio en c.

Intervalo de Confianza: [0.61, 1.0]

Se obtuvo 0.96 como límite superior de confianza de SCCr por análisis estadístico. Sin embargo se encontró que los componentes tendían a tener métodos de negocio sin valor de retorno independientemente de la reutilización. Por lo tanto se usó 1.0 como límite superior de confianza.

Consideración: SCCr indica el grado de libre-integridad, y el bajo grado de dependencia externa para los usuarios del componente.

Una alta libre-integridad de un componente (baja dependencia externa) conduce a la alta portabilidad del componente. Si el valor de SCCr(c) está en el intervalo de confianza, la dependencia externa es pequeña y la portabilidad de c es alta.

Definición (SCCp: libre integridad de parámetros de un componente)

SCCp(c): Es el porcentaje de los métodos de negocio sin ningún parámetro en todos los métodos de negocio implementados en un componente c.

$$SCCp(c) = \begin{cases} \frac{B_p(c)}{B(c)} & (B(c) > 0) \\ 1 & (\text{otherwise}) \end{cases}$$

Donde:

Bp(c): número de métodos de negocio sin parámetros en un componente c.

Intervalo de Confianza: [0.42, 0.77]

Consideración: SCCp indica el grado de libre-integridad de un componente y el pequeño grado de dependencia para usuarios de un componente. (17)

1.5.3. Métricas adaptadas a Servicios Web

Las siguientes métricas fueron obtenidas de un estudio realizado basándose en métricas de interfaz de componentes las cuales fueron adaptadas a servicios web. El resultado de este estudio fue expuesto en la Conferencia Internacional de Web Services del año 2007. Estas métricas miden la consistencia, legibilidad, autodescriptividad, tamaño, comprensibilidad, criterios que en conjunto dan una medida de la reusabilidad de un producto web service.

Definición 1:

APO: argumento por operación

$$APO = \frac{n_a}{n_o}$$

Donde:

Na: número total de argumentos.

No: número total de operaciones.

Consideración: La cantidad de argumentos y de operaciones, provee una forma de medir el tamaño de un web service, mostrando su tamaño funcional. Cuanto más pequeño es el número de operaciones y los argumentos, mejor será la comprensibilidad. Por otro lado un número grande de argumentos y operaciones permite que el servicio sea más fácil de invocar y de usar.

Definición 2:

(DAC, DAR): Contador de distintos argumentos (DAC)

DAR se define como:

$$DAR = \frac{DAC}{n_a}$$

Donde:

Na: es el número total de argumentos.

Consideración: Cuanto mayor sea el valor de DAR, mayor independencia entre los servicios, menos similitudes entre los argumentos y más consistencia de las declaraciones del argumento. Las declaraciones de argumentos consistentes hacen más fácil el entendimiento y reuso de componentes.

Definición 3:

ARS: escala de repetición del argumento

Dada una secuencia A de pares (nombre del argumento, tipo del argumento) en el servicio web, ARS se define como:

$$ARS(A) = \frac{\sum_{a \in A} |a|^2}{|A|}$$

Donde |a| es el número de repeticiones de a en A.

Consideración: Un bajo valor de ARS indica más funcionalidad especializada. Teniendo un DAR alto y un ARS bajo conlleva a que las interfaces WSDL sean más reusables en general, pero también más difícil de comprender la funcionalidad de cada operación.

Cuando los valores de DAR en diferentes servicios web son iguales, se aplica la métrica ARS para medir cual es más consistente y por tanto, tiene mayor transferibilidad de datos.

Definición 4:

MIL: longitud media de identificadores

Es el tamaño medio de la longitud de los identificadores argumentos y operaciones que aparecen en el documento WSDL.

Definición 5:

MEIL: es la mediana de la longitud de los identificadores argumentos y operaciones que aparecen en el documento WSDL.

Consideración: Cuanto mayor sea el valor de las métricas anteriores mayor será el grado de legibilidad y entendimiento del documento WSDL, lo cual es un factor que ayuda a la reusabilidad del servicio.

Definición 6:

MSC: media de cadena común de un conjunto de identificadores A que se define como:

$$MSC(A) = \frac{\sum_{(x,y) \in comb(A)} \frac{|lcs(x, y)|}{\max(|x|, |y|)}}{n * (n - 1) / 2}$$

Donde:

$$n = |A|$$

lcs(x, y): es la cadena común mas grande de x e y.

Comb(A): es el conjunto de todas las combinaciones distintas entre los identificadores en A.

MSC: media de cadena común de un conjunto de operaciones O que se define como:

$$MSC(O) = \frac{\sum_{(x,y) \in comb(O)} \frac{|lcs(x, y)|}{\max(|x|, |y|)}}{n * (n - 1) / 2}$$

Donde:

$$n = |O|$$

lcs (x, y): es la cadena común mas grande de x e y.

Comb (O): es el conjunto de todas las combinaciones distintas entre las operaciones en O.

Consideración: Las descripciones WSDL tienden a tener una longitud larga de identificadores que hacen que el WSDL sea más auto-descriptivo. (19)

1.6. Reutilización de software en Cuba

Debido a que la industria cubana del software se está ordenando y va madurando gradualmente, algunos temas como los relacionados con la reutilización de componentes de software, aunque se conocen, han sido pensados y se ha comprendido su importancia en muchas empresas cubanas, no es de los temas que se han llevado a la práctica con más fuerza.

DESOFTEC, una de las empresas que más experiencia tiene en el desarrollo de software en Cuba, no tiene mucha experiencia en reutilizar componentes de software, a pesar de que es algo que desean llevar a la práctica y para lo cual comenzaron a dar los primeros pasos. En estos momentos están inmersos en mantener la metodología del proceso de desarrollo, que todos hagan las cosas de la misma manera a la hora de producir software. Algo que les interesa es que cuando se termine un proyecto, puedan quedar cosas para reutilizar en los próximos proyectos que deban realizar.

En el caso de TECNOMÁTICA, que cuenta con más de 25 años de experiencia y se dedican generalmente a la Gestión Empresarial, principalmente el área de la Contabilidad y Finanzas, desde hace más de dos décadas emplean la reutilización en la producción de software. Señalan que ahorra mucho tiempo, permite dar soluciones estandarizadas y más seguras, pues se utiliza código que ha sido probado en la práctica. En sus inicios se hacía de forma individual y no organizada, pero desde hace algunos años se aplica una política más efectiva y generalizada al respecto, lo que permite aumentar considerablemente la productividad del proceso de desarrollo. Los elementos que más ha reutilizado TECNOMÁTICA son: el código, los componentes, datos, entre otros en menor grado.

La falta de estándares prefijados de documentación técnica de los componentes, estructuras de datos y código que se reutilizan, son las principales dificultades que han enfrentado. Apuestan por la reutilización, a pesar de los problemas que aún puedan presentarse, porque consideran que es más factible crear los nuevos productos aplicándola, pues permite a los técnicos concentrarse en los problemas de la nueva funcionalidad solicitada, al aprovechar programas y datos ya elaborados y probados con anterioridad. Además reduce el tiempo de desarrollo y aumenta la seguridad y confiabilidad del producto final.

No solo se deben mencionar las empresas cubanas productoras de software. Se debe tener en cuenta el aporte que pueden hacer al tema las universidades, en las que se estudia por supuesto la carrera Ingeniería Informática y que también construyen software. La Universidad Central de Las Villas “Martha Abreu” (UCLV) ha trabajado esencialmente en las esferas turismo y educación y hace ya algún tiempo que emplean la reutilización de software, como un factor importante para el ahorro de tiempo y esfuerzo. Lo que reutilizan en mayor medida son componentes del tipo de código, arquitectura, diseño, entre otros. Como principal dificultad enfrentada está el hecho de que todo lo que desarrollan no lo conciben como un elemento que puede ser reutilizado posteriormente, por lo que tienen que dedicar tiempo y esfuerzo extras.

En la UCLV existe un grupo investigativo, Laboratorio de Investigación de Informática Educativa del Centro de Estudios de Informática, que posee más de 17 años de experiencia en el tema de la producción de software, esencialmente para la educación y quienes consideran a la reutilización como un factor importante dentro del proceso de desarrollo de un producto, debido a que contribuye al perfeccionamiento de los resultados y permite ahorrar tiempo. El problema principal que han afrontado es la documentación y la comprensión del código.

1.7. Conclusiones

Después de la investigación realizada se puede concluir que se van a proponer un conjunto de métricas ya adaptadas a servicios web, las cuales no cubren todos los criterios para medir la reusabilidad por lo que no constituyen un modelo de reusabilidad. Es necesario destacar que no se pueden adaptar algunas métricas de componentes a los servicios web por las diferencias que existen entre ambos, dentro de las que se pueden mencionar la ausencia de estado en los servicios web, lo que provoca que las operaciones en el WSDL sean diseñadas para proveer servicios de estilo-transacción y deban obtener la información necesaria de la lista de argumentos, lo que constituye una limitante para aplicar a los servicios web las métricas de reusabilidad orientadas a componentes que se basan en los métodos sin parámetros y sin valor de retorno.

CAPÍTULO 2: DIAGNÓSTICO DE LA SITUACIÓN ACTUAL

2.1. Introducción

El objetivo de este capítulo es evaluar la necesidad y condiciones que existen actualmente en una empresa cubana productora de software para introducir la aplicación de métricas de reusabilidad, a los servicios Web que se desarrollan en la misma. Para ello las estudiantes Lisbet Bertoli Ramos y Marilyn Amador Medero tutoradas por la Lic. Elizabeth Betancourt Herrera realizan una encuesta, la cual se aplica a una muestra de 14 especialistas de mayor experiencia y 13 programadores de una empresa cubana productora de software.

Para emitir una valoración de la situación actual de la empresa se hizo un análisis estadístico por cada una de las preguntas teniendo en cuenta el rol que desempeña el trabajador encuestado y sus años de experiencia desarrollando software.

2.2. Análisis estadístico

Preg: ¿La empresa desarrolla únicamente software interno?

Valoración: Se puede decir que los servicios web que se desarrollan en la empresa pueden ser reutilizados tanto por los desarrolladores de la empresa como por otras empresas.

Preg: Tiene algún conocimiento de:

- Conceptos de reusabilidad
- Métricas de reusabilidad
- Modelos de reusabilidad

CAPÍTULO 2: DIAGNÓSTICO DE LA SITUACIÓN ACTUAL

Conocimiento	Cant Si	Cant No
Concepto	22	5
Métricas	9	18
Modelos	3	24

Tabla 2.1.

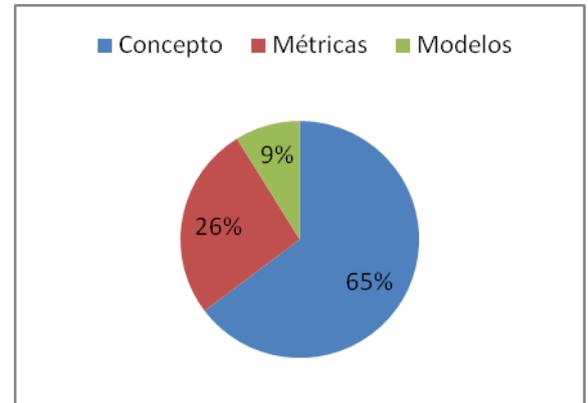


Fig.2.1.

Valoración: En la empresa se tiene conocimiento del concepto de reusabilidad, de métricas se tiene algún conocimiento, pero de modelos no se conoce prácticamente nada.

Preg: ¿Se aplica alguna métrica relacionada con reusabilidad en su ámbito de trabajo?

Roles	Cant Si	Cant No
Director de desarrollo		
Jefe de proyecto	1	
Gestor de proyecto	2	
Analista		2
Arquitecto	2	2
Programador	3	10
Administrador de red		
Gestor de producto		3
Jefe Unidad de Servicios		1
Total	8	18

Tabla 2.2.

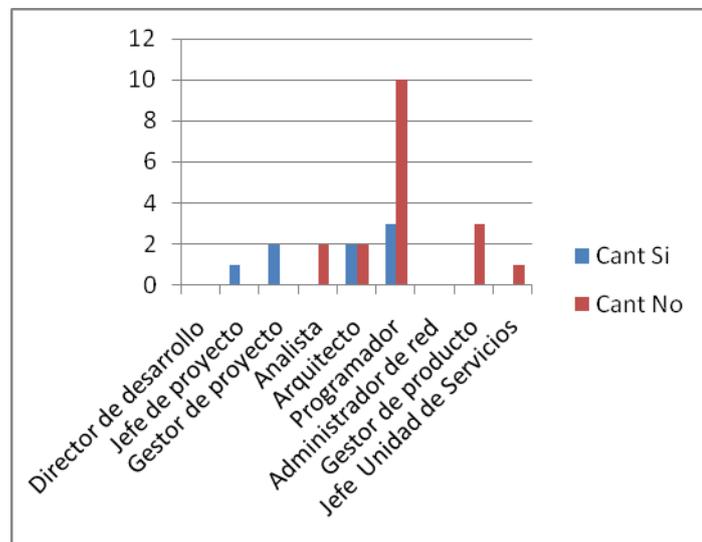


Fig.2.2.

CAPÍTULO 2: DIAGNÓSTICO DE LA SITUACIÓN ACTUAL

Experiencia	Si	No
(0-2)		6
(3-8)	4	8
(Más de 8)	4	4

Tabla 2.3.

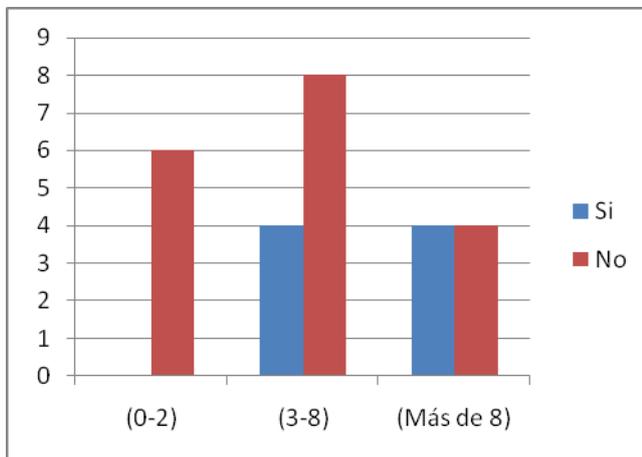


Fig.2.3.

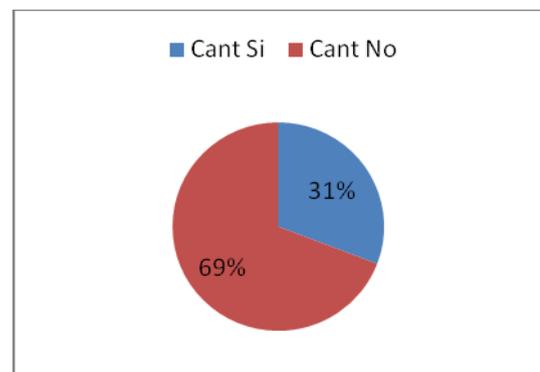


Fig.2.4

Valoración: Según la opinión de los trabajadores de mayor experiencia se tiene que hay la misma cantidad de respuestas afirmativas que negativas. Atendiendo a los roles la cantidad de respuestas afirmativas por parte de los especialistas está por encima de las respuestas afirmativas de los programadores. De forma general el porcentaje de respuestas negativas supera al porcentaje de respuestas positivas. Se puede decir que aunque en la empresa se aplican métricas relacionadas con la reusabilidad, no se aplica en todos los roles ni en todos los ámbitos de trabajo. Aparentemente, los años de experiencia influyen en la cantidad de respuestas positivas.

CAPÍTULO 2: DIAGNÓSTICO DE LA SITUACIÓN ACTUAL

Preg: ¿Qué métricas utiliza?

Métricas	Cant
Orientada a Producto	5
Orientada a Proceso	2
Orientada a Negocio	4

Tabla 2.4.

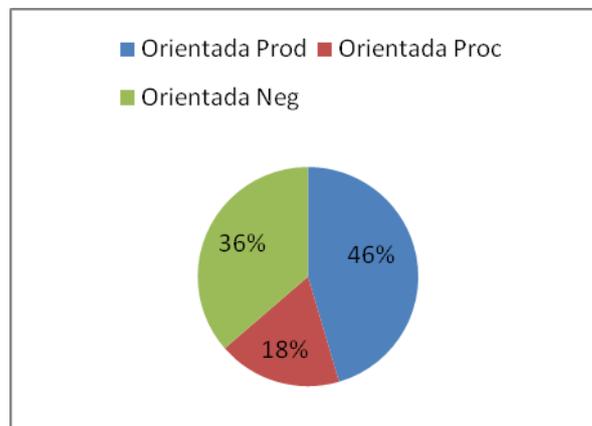


Fig.2.5.

Valoración: En los roles y ámbitos de trabajo en que se aplica alguna métrica relacionada con la reusabilidad, se utiliza principalmente orientada a producto.

Preg: ¿Considera que es importante reutilizar?

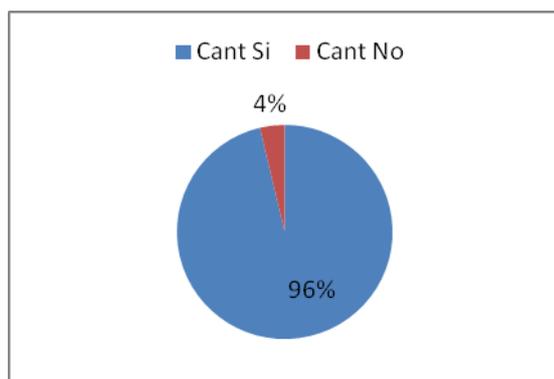


Fig.2.6.

Valoración: Teniendo en cuenta el porcentaje elevado de las respuestas afirmativas, se puede decir que los trabajadores de la empresa valoran la importancia de la reutilización.

CAPÍTULO 2: DIAGNÓSTICO DE LA SITUACIÓN ACTUAL

Preg: ¿En su ámbito de trabajo se aplica la reutilización?

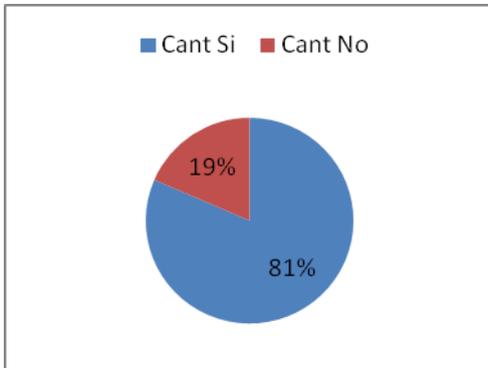


Fig.2.7.

Valoración: Por ser mayor el porcentaje de respuestas afirmativas se puede decir que en la empresa se aplica la reutilización aunque no en todos los ámbitos de trabajo.

Preg: ¿En qué nivel se aplica la reutilización?

Niveles	Cant
Código	19
Componentes	17
Clases	12
Documentación	11
Modelos	6
Dom. o Neg.	7

Tabla 2.5.

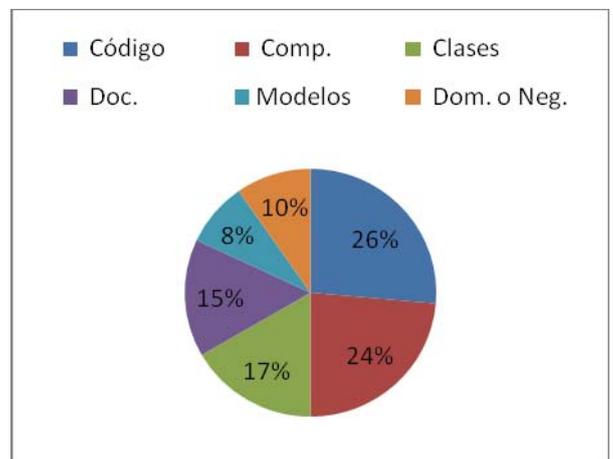


Fig.2.8.

Valoración: En la empresa se aplica la reutilización a los niveles más bajos de abstracción(nivel de código,componentes, clases y documentación).

Preg: ¿Conoce o tiene para su ámbito de trabajo una evaluación cuantitativa o cualitativa de cómo y cuánto se aplica la reutilización?



Fig.2.9.

Valoración: Aunque en la empresa se aplique la reutilización, no se conoce la medida en que se aplica.

Preg: ¿Conoce o aplica alguna métrica para evaluar cuantitativa o cualitativamente la capacidad de reutilización de los componentes de software disponibles en la empresa?

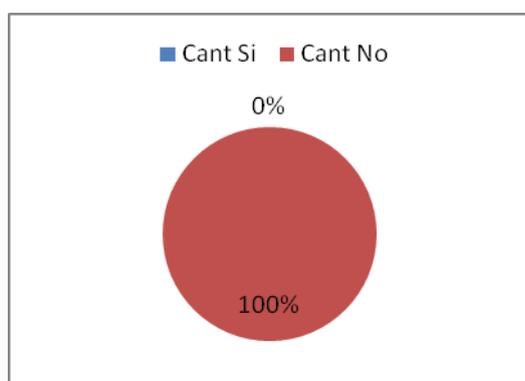


Fig.2.10.

Valoración: Aunque en la empresa se aplique la reutilización, no se tiene como evaluar la reusabilidad.

Preg: ¿Ayudaría a su trabajo contar con datos sobre la reusabilidad de los componentes disponibles en la empresa?

CAPÍTULO 2: DIAGNÓSTICO DE LA SITUACIÓN ACTUAL

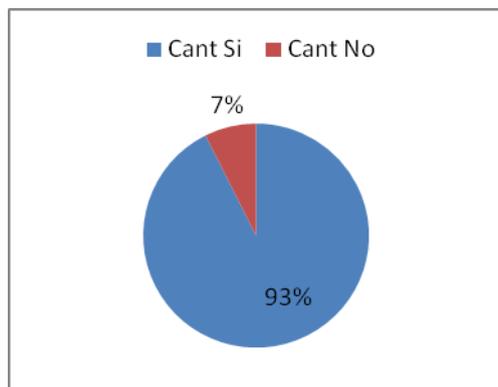


Fig.2.11.

Valoración: Teniendo en cuenta que el porcentaje de respuestas afirmativas es bastante elevado, se puede decir que la empresa considera que sería de gran ayuda contar con datos sobre la reusabilidad de los componentes disponibles en la misma. Los años de experiencia y los roles no influyen.

Preg: ¿Cómo le ayudaría a su trabajo contar con datos sobre reusabilidad de componentes?

Cómo ayudaría la reusabilidad	Cant
A	19
B	22
C	25

Tabla 2.6.

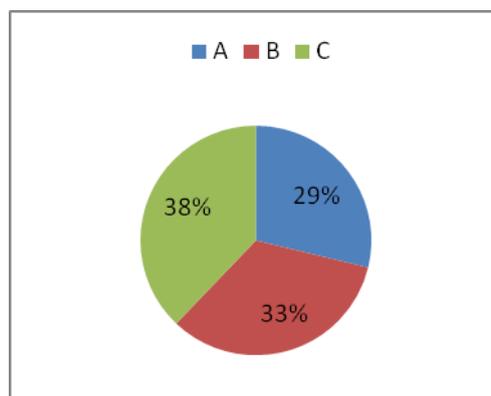


Fig.2.12.

A: Ayudaría a seleccionar los componentes adecuados para desarrollar nuevas aplicaciones.

B: Ayudaría a construir componentes para reutilizar.

C: Ayudaría a construir nuevas aplicaciones utilizando lo que ya está.

Valoración: En la empresa consideran que contar con métricas en su ámbito de trabajo les ayudaría a construir nuevas aplicaciones utilizando los componentes ya existentes.

CAPÍTULO 2: DIAGNÓSTICO DE LA SITUACIÓN ACTUAL

Preg: A la hora de proponer métricas que se puedan utilizar para medir la reusabilidad ¿Qué criterios valoraría?

Criterios	Cant
A	18
B	14
C	13
D	16
E	19

Tabla 2.7.

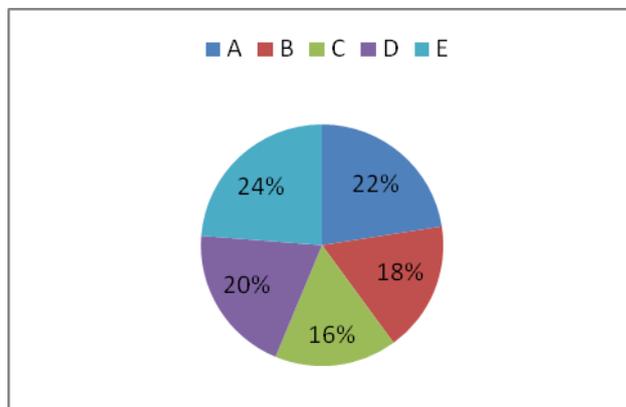


Fig.2.13.

A: Información que produce.

B: Ayuda en la toma de decisiones.

C: Simplicidad de la métrica.

D: Facilidad para recolectar y evaluar los datos.

E: Herramientas que ayuden a evaluar y recolectar los datos.

Valoración: A la hora de proponer métricas para medir la reusabilidad los criterios que más valorara la empresa sería contar con herramientas que ayuden a recolectar y evaluar los datos de los servicios web, la información que produce la métrica, la facilidad para recolectar y evaluar los datos, la ayuda en la toma de decisiones y la simplicidad de la métrica.

Preg: ¿Cuentan con herramientas para recolectar datos sobre los Servicios Web disponibles?

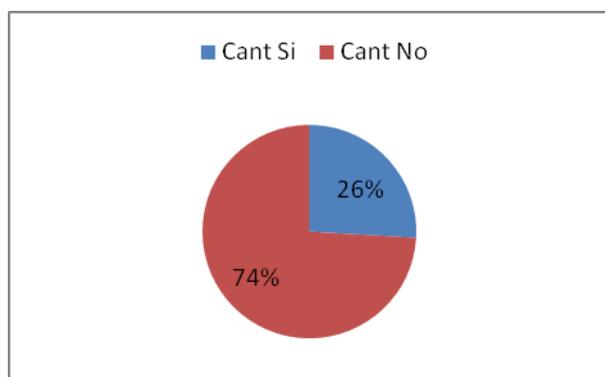


Fig.2.14.

CAPÍTULO 2: DIAGNÓSTICO DE LA SITUACIÓN ACTUAL

Valoración: Aunque se utilizan herramientas para recolectar datos, no están generalizadas en todos los ámbitos de trabajo, hay un porcentaje pequeño que las conoce y las utiliza.

Preg: ¿Cuentan con herramientas para evaluar los datos recolectados de los Servicios Web disponibles?

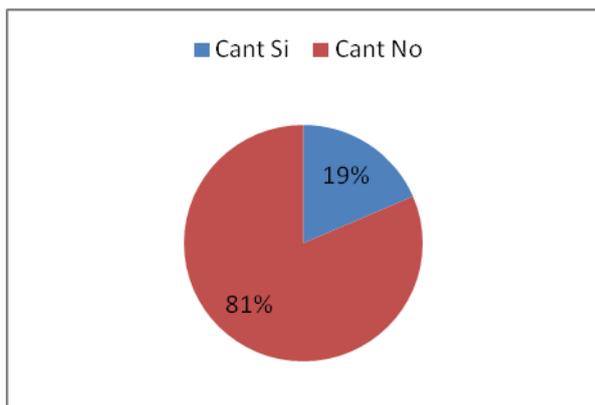


Fig.2.15.

Valoración: Aunque se utilizan herramientas para evaluar datos, no están generalizadas en todos los ámbitos de trabajo, hay un porcentaje pequeño que las conoce y las utiliza.

Preg: ¿Existe un proceso o mecanismo definido para recolectar datos sobre los Servicios Web disponibles?

Roles	Cant Si	Cant No
Director de desarrollo		
Jefe de proyecto		1
Gestor de proyecto		2
Analista		2
Arquitecto	4	
Programador	5	8
Administrador de red		1
Gestor de producto		3
Jefe Unidad de Servicios		1
Total	9	18

Tabla 2.9.

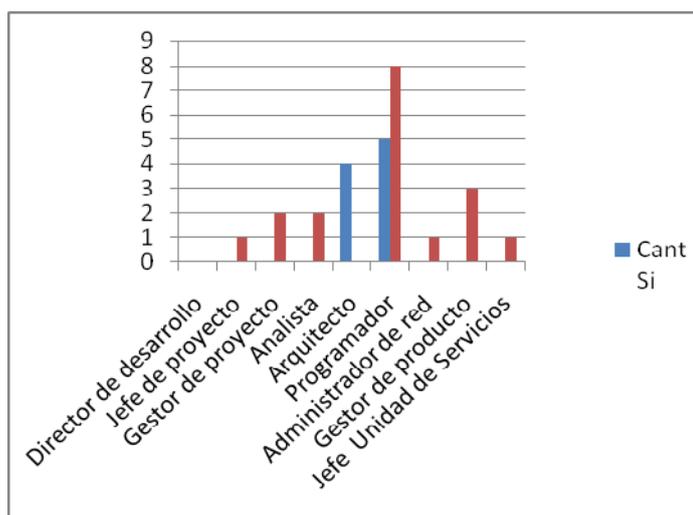


Fig.2.16

CAPÍTULO 2: DIAGNÓSTICO DE LA SITUACIÓN ACTUAL

Experiencia	Si	No
(0-2)	3	4
(3-8)	3	7
(Más de 8)	3	7

Tabla 2.10.

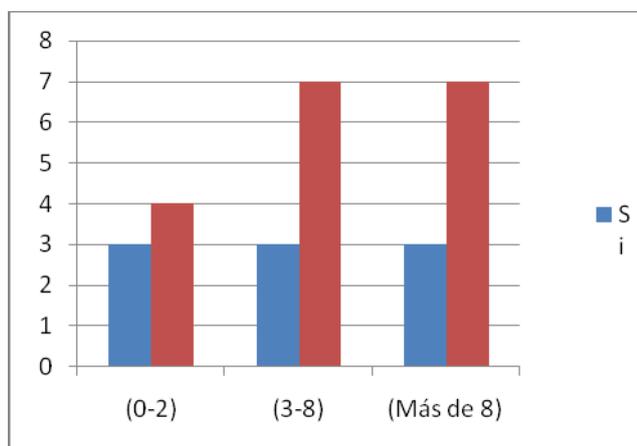


Fig.2.17.

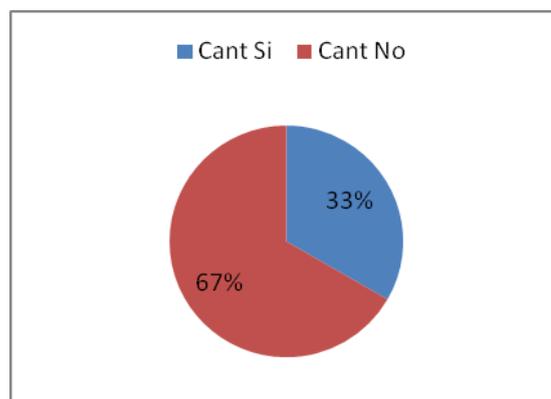


Fig.2.18.

Valoración: Según el criterio de los trabajadores de mayor experiencia se tiene que es mayor la cantidad de respuestas negativas. Atendiendo a los roles tanto especialistas como programadores dan respuestas afirmativas pero las respuestas negativas de ambos las superan en número. De forma general el porcentaje de respuestas negativas supera al porcentaje de respuestas positivas. Se puede decir que en la empresa existe un proceso o mecanismo definido para recolectar datos sobre los Servicios Web disponibles, pero no generalizado a todos los ámbitos de trabajo y roles.

Preg: ¿Existe una planificación (tiempo y recursos asignados) para el proceso o mecanismo definido para recolectar datos sobre los servicios web disponibles?

CAPÍTULO 2: DIAGNÓSTICO DE LA SITUACIÓN ACTUAL

Roles	Cant Si	Cant No
Director de desarrollo		
Jefe de proyecto		1
Gestor de proyecto		2
Analista		2
Arquitecto	2	2
Programador	5	8
Administrador de red		1
Gestor de producto		3
Jefe Unidad de Servicios		1

Tabla.2.11.

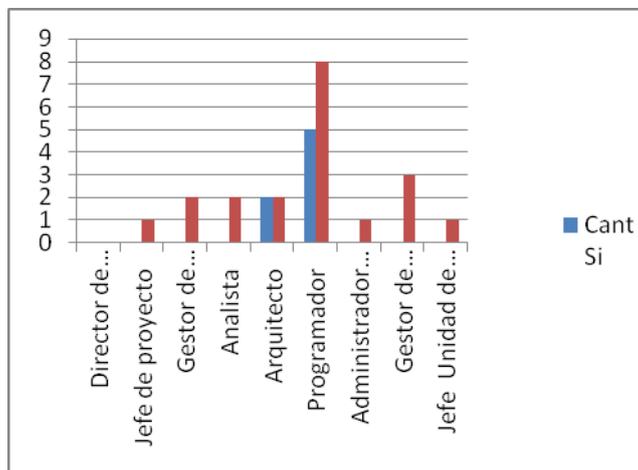


Fig.2.19.

Experiencia	Si	No
(0-2)	3	4
(3-8)	3	9
(Más de 8)	1	7

Tabla.2.12.

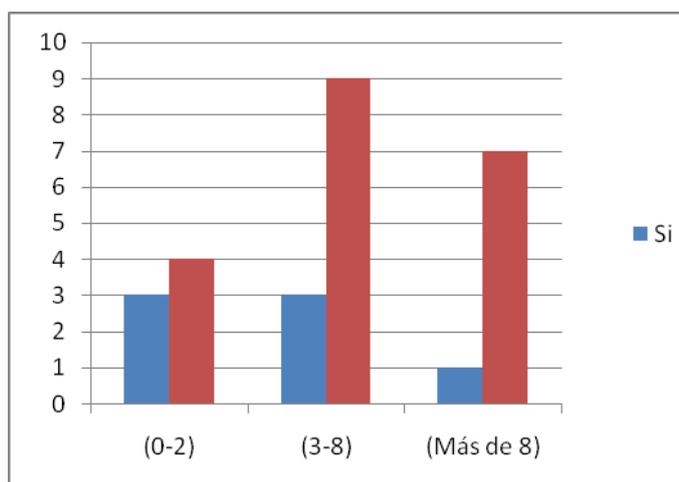


Fig.2.20.

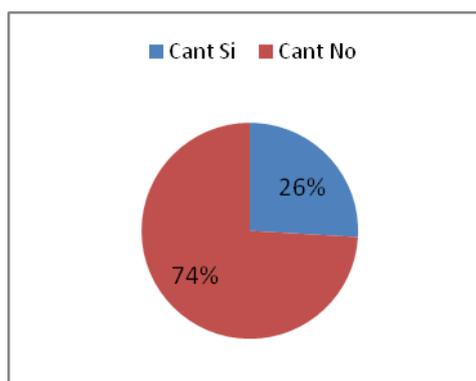


Fig.2.21.

Valoración: Según el criterio de los trabajadores de mayor experiencia se tiene que es mayor la cantidad de respuestas negativas. Atendiendo a los roles tanto especialistas como programadores dan

CAPÍTULO 2: DIAGNÓSTICO DE LA SITUACIÓN ACTUAL

respuestas afirmativas pero las respuestas negativas de ambos las superan en número. De forma general el porcentaje de respuestas negativas supera al porcentaje de respuestas positivas. Se puede decir que en los ámbitos de trabajo y roles en donde existe el proceso o mecanismo para recolectar datos sobre web services en la empresa, existe una planificación para el mismo.

Preg: ¿Cuentan con un repositorio, base de datos, mecanismo o infraestructura para guardar los datos recolectados sobre los Servicios Web disponibles?

Roles	Cant Si	Cant No
Director de desarrollo		
Jefe de proyecto		1
Gestor de proyecto	1	1
Analista		2
Arquitecto	3	1
Programador	9	1
Administrador de red	1	
Gestor de producto	1	
Jefe Unidad de Servicios	1	
Total	16	6

Tabla 2.13.

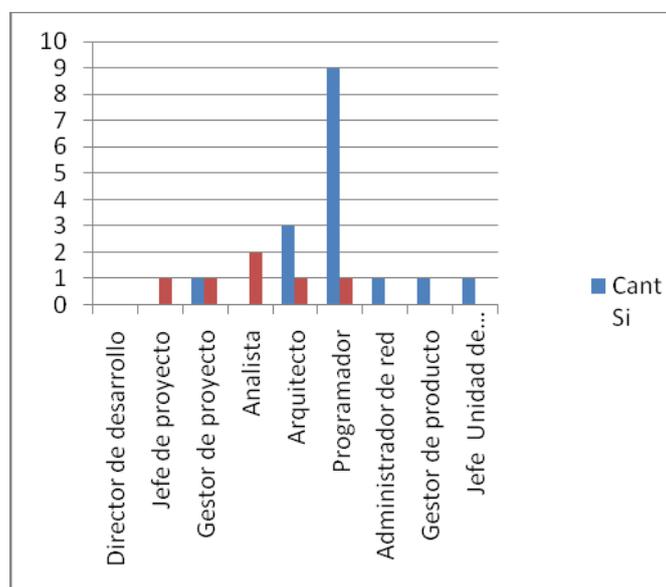


Fig.2.22

Experiencia	Si	No
(0-2)	5	
(3-8)	7	3
(Más de 8)	4	3

Tabla 2.14.

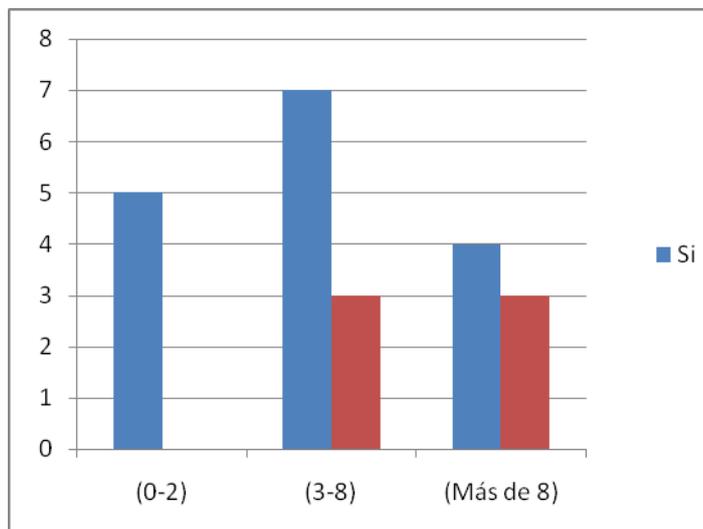


Fig.2.23.

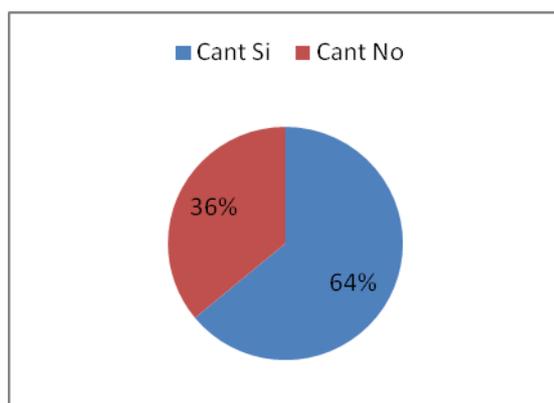


Fig.2.24.

Valoración: Según el criterio de los trabajadores de mayor experiencia se tiene que es mayor la cantidad de respuestas afirmativas. Atendiendo a los roles tanto especialistas como programadores dan respuestas negativas pero las respuestas afirmativas de ambos las superan en número. De forma general el porcentaje de respuestas positivas supera al porcentaje de respuestas negativas. Se puede decir que en la empresa se cuenta con un repositorio, base de datos, mecanismo o infraestructura para guardar los datos recolectados sobre los Servicios Web disponibles, pero no generalizado a todos los ámbitos de trabajo y roles.

2.3. Conclusiones

Como resultado de la encuesta se puede concluir que la empresa no cuenta con métricas definidas para medir la reusabilidad de los web services que desarrolla. Existen pocas herramientas e información sobre reusabilidad lo cual puede constituir una limitante a la hora de introducir la aplicación de las métricas. No obstante, se conoce y valora la importancia de aplicar la reutilización y contar con información sobre la reusabilidad de los servicios web que desarrollan, en aras de obtener una mayor calidad en el desarrollo de soluciones y productos software.

2.4. Recomendaciones

Para comenzar a introducir las métricas, se propone que se escojan métricas sencillas que no necesiten de un volumen grande de datos ni de herramientas complejas para poder aplicarse y que a medida que se introducen las mismas se mejore el conocimiento sobre los criterios de reusabilidad en la empresa. Además es necesario generalizar e introducir nuevas herramientas que permitan recolectar y evaluar datos de reusabilidad.

CAPÍTULO 3: PROPUESTA.

3.1. Introducción

El desarrollo de software basado en componentes se basa en componentes reutilizables con el fin de mejorar la calidad y flexibilidad de los productos, así como el aumento de la productividad del desarrollo.

El objetivo de este capítulo es proponer un conjunto de métricas para medir y evaluar la reusabilidad de un producto Web Services.

Las métricas a proponer fueron obtenidas de un estudio realizado basándose en métricas de interfaz de componentes las cuales fueron adaptadas a servicios web. El resultado de este estudio fue expuesto en la Conferencia Internacional de Web Services del año 2007.

3.2. Estructura del documento WSDL. Propuesta de métricas

3.2.1. Documento WSDL

Para entender las métricas a proponer hay que conocer algunos aspectos del documento WSDL el cual es un fichero XML que define un conjunto de operaciones, que es similar a definiciones de interfaz en los componentes. Este documento es usado para describir la interfaz de un servicio web como un conjunto de puntos finales de comunicación (métodos) capaces de intercambiar mensajes (es decir recibir llamadas con sus parámetros correspondientes y generar respuesta con el resultado que le corresponda). WSDL se considera parte integral de UDDI, que es un registro de servicios Web XML

WSDL define un servicio web utilizando a tal fin elementos XML, como:

- <portType> para las operaciones que proporciona el servicio web
- <message> para los mensajes utilizados por el servicio web
- <types> para los tipos de datos que utiliza el servicio web
- <binding> para los protocolos de comunicaciones que utiliza el servicio web

Estructura de un documento WSDL:

```
<definitions>
<types>
los tipos de datos...
</types>
<message>
las definiciones del mensaje...
</message>
<portType>
las definiciones de operación...
</portType>
<binding>
las definiciones de protocolo...
</binding>
</definitions>
```

Los puertos de WSDL

`<portType>` es el elemento XML de WSDL que define el servicio web, así como las operaciones posibles mediante dicho servicio y los mensajes vinculados. `<portType>` cumple una función análoga a la de una función de biblioteca en programación clásica o a la de una clase en programación orientada a objetos.

Los mensajes WSDL

El elemento `message` define los datos que participan en una operación. Cada mensaje puede tener una o varias partes, y cada parte puede considerarse como si fuera los parámetros que se pasan en la llamada a una función en programación clásica o un método en programación orientada a objetos.

Los tipos de datos en WSDL

Mediante el elemento `<types>` se definen los tipos de datos utilizados en el servicio web. Para ello, WSDL utiliza XML Schema.

Los vínculos en WSDL

<binding> define el formato del mensaje y el protocolo para cada uno de los puertos. El elemento <binding> cuenta con 2 atributos: "name" y "type". Con "name" (se puede utilizar cualquier nombre, sin que coincida necesariamente con el utilizado en la definición de port) se define el nombre del vínculo, y el atributo "type" apunta al puerto del vínculo.

Operación

“Operation” define las operaciones que proporciona el puerto. Para cada una se debe especificar la acción SOAP que corresponda. También se debe especificar el método para codificar la entrada (o "input") y la salida (u "output").

Tipos de operación

Existen varios tipos de operación en WSDL. El tipo más frecuente es el denominado "de petición-respuesta". Se dispone además, de:

- Unidireccional: la operación recibe mensajes, sin retornar respuestas.
- Petición-respuesta: la operación recibe una petición y devuelve una respuesta.
- Solicitud-respuesta: la operación puede enviar una petición y permanecerá a la espera de una respuesta.

Notificación: la operación puede enviar un mensaje sin esperar respuesta.

3.2.2. Métricas

La propuesta de las siguientes métricas se basa en diferentes criterios como consistencia, legibilidad, autodescriptividad, tamaño y comprensibilidad, que en conjunto dan una medida de la reusabilidad de un producto web service.

Como primera métrica se tiene:

Definición 1:

EMI: Existencia de Meta-Información.

EMI (sw) Es una adaptación para servicios web de la métrica EMI(c) para componentes la que es utilizada para medir si se tiene una descripción con la que se pueda entender mejor el componente.

EMI (sw) =1 Si se cuenta con el documento WSDL.

EMI (sw)=0 En otro caso.

Consideración: Si el valor de EMI (sw) es 1, se puede entender fácilmente como usar el servicio web y que es lo que el desarrollador del servicio asume.

Definición 2:

APO: argumento por operación

$$APO = \frac{n_a}{n_o}$$

Donde:

Na: número total de argumentos, que serían los part name de un mensaje en el documento WSDL.

No: número total de operaciones definidas en cada mensaje en el documento WSDL.

Consideración: La cantidad de argumentos y de operaciones, provee una forma de medir el tamaño de un web service, mostrando su tamaño funcional. Cuanto más pequeño es el número de operaciones y los argumentos, mejor será la comprensibilidad. Por otro lado un número grande de argumentos y operaciones permite que el servicio sea más fácil de invocar y de usar.

Definición 3:

(DAC, DAR): Contador de distintos argumentos (DAC) será la cantidad de argumentos distintos del conjunto de pares (argumento, tipo) que aparecen en la descripción de interfaz WSDL. DAR se define como:

$$DAR = \frac{DAC}{n_a}$$

Donde:

Na: es el número total de argumentos que aparecen en el documento WSDL.

Consideración: Cuanto mayor sea el valor de DAR, mayor independencia entre los servicios, menos similitudes entre los argumentos y más consistencia de las declaraciones del argumento. Las declaraciones de argumentos consistentes hacen más fácil el entendimiento y reuso de componentes.

Definición 4:

ARS: escala de repetición del argumento

Dada una secuencia A de pares (nombre del argumento, tipo del argumento) en el servicio web, ARS se define como:

$$ARS(A) = \frac{\sum_{a \in A} |a|^2}{|A|}$$

Donde $|a|$ es el número de repeticiones de a en A .

Consideración: Un bajo valor de ARS indica más funcionalidad especializada. Teniendo un DAR alto y un ARS bajo conlleva a que las interfaces WSDL sean más reusables en general, pero también más difícil de comprender la funcionalidad de cada operación.

Cuando los valores de DAR en diferentes servicios web son iguales, se aplica la métrica ARS para medir cual es más consistente y por tanto, tiene mayor transferibilidad de datos.

Definición 5:

MIL: longitud media de identificadores

$$MIL = \frac{PCA + PCO}{n_{ad} + n_{od}}$$

Donde:

PCA: cantidad de palabras completas en los identificadores de los argumentos.

PCO: cantidad de palabras completas en los identificadores de las operaciones.

n_{ad} : número de argumentos distintos.

n_{od} : número de operaciones distintas.

Es el tamaño medio de la cantidad de palabras completas de los nombres de argumentos y operaciones que aparecen en el documento WSDL.

Definición 6:

MEIL: es la mediana de la cantidad de palabras completas de los identificadores que aparecen en el documento WSDL.

$$MEIL = \frac{PCA + PCO}{2}$$

Donde:

PCA: cantidad de palabras completas en los identificadores de los argumentos.

PCO: cantidad de palabras completas en los identificadores de las operaciones.

Consideración: Cuanto mayor sea el valor de las métricas anteriores mayor será el grado de legibilidad y entendimiento del documento WSDL, lo cual es un factor que ayuda a la reusabilidad del servicio.

Definición 7:

MSC: media de cadena común de un conjunto de identificadores A que se define como:

$$MSC(A) = \frac{\sum_{(x,y) \in comb(A)} \frac{|lcs(x,y)|}{\max(|x|, |y|)}}{n * (n-1) / 2}$$

Donde:

$n = |A|$

$lcs(x, y)$: es la cadena común mas grande de x e y.

$Comb(A)$: es el conjunto de todas las combinaciones distintas entre los identificadores en A.

MSC: media de cadena común de un conjunto de operaciones O que se define como:

$$MSC(O) = \frac{\sum_{(x,y) \in comb(O)} \frac{|lcs(x,y)|}{\max(|x|, |y|)}}{n * (n-1) / 2}$$

Donde:

$n = |O|$

$lcs(x, y)$: es la cadena común mas grande de x e y.

$Comb(O)$: es el conjunto de todas las combinaciones distintas entre las operaciones en O.

Consideración: Las descripciones WSDL tienden a tener una longitud larga de identificadores que hacen que el WSDL sea más auto-descriptivo.

3.3. Conclusiones

Dándole cumplimiento al objetivo del capítulo, quedaron propuestas siete métricas para medir la reusabilidad de un producto web services. Estas métricas fueron escogidas por su sencillez, porque no se necesitan herramientas complejas para poder recolectar los datos ni para evaluarlos.

3.4. Recomendaciones

Aunque las métricas sean sencillas, para facilitar su cálculo se recomienda la creación de una herramienta.

CAPÍTULO 4: ANÁLISIS DE LOS RESULTADOS.

4.1. Introducción

En este capítulo se aplican las métricas propuestas al documento WSDL de dos servicios web desarrollados en una empresa cubana productora de software que se ha estado analizando en esta investigación. Además, se hace una evaluación de la reusabilidad de los mismos a partir del análisis y valoración de las medidas obtenidas al aplicar dichas métricas. El texto de estos WSDL se puede consultar en los Anexos.

4.2. Aplicación a los Casos de Estudio

4.2.1. Caso 1: Servicio Web Catalogo_ECService

Métrica 1:

EMI (sw) tiene valor 1, pues se cuenta con el documento WSDL.

Métrica 2:

$$APO = \frac{n_a}{n_o}$$

Selección de argumentos

1. Catalogo
2. estructura: es de un tipo de dato compuesto array_CatalogoStruct, que a su vez es también de tipo compuesto CatalogoStruct, el cual se compone por los elementos: (idCamp, Nombre, Tipo, Tamano, Orden, Llave, Visible, Editable, Sugerible, Vacio, Enlace, Origen, Filtro, Grupo).
3. codigo
4. mensaje
5. InfoCatalogos: es de un tipo de dato compuesto array_CatalogoInfo, que a su vez es de tipo compuesto CatalogoInfo que esta compuesto por los elementos (Nombre, Alias y Array_Grupo de tipo compuesto Array_GrupoType (Grupo que es de tipo compuesto GrupoType (Nombre y Cardinal))).
6. codigo
7. mensaje
8. Catalogo

9. Datos: es de tipo de dato compuesto array_CatValue (que tiene dos argumentos de tipo compuesto CampoType (id_Campo) y Array_FilaType (que tiene un argumento de tipo de dato compuesto FilaType (Valor))).

10. ok

11. codigo

12. mensaje

13. Catalogo

14. Datos: es de tipo de dato compuesto array_CatValue (que tiene dos argumentos de tipo compuesto CampoType (id_Campo) y Array_FilaType (que tiene un argumento de tipo de dato compuesto FilaType (Valor))).

15. ok

16. codigo

17. mensaje

18. Catalogo

19. Datos: es de tipo de dato compuesto array_CatValue (que tiene dos argumentos de tipo compuesto CampoType (id_Campo) y Array_FilaType (que tiene un argumento de tipo de dato compuesto FilaType (Valor))).

20. ok

21. codigo

22. mensaje

23. Catalogo

24. paginado: es de un tipo de dato compuesto Paginado (offset, cantidad, cantidad_real, total).

25. Criterio: es de tipo de dato compuesto array_CatValue (que tiene dos argumentos de tipo compuesto CampoType (id_Campo) y Array_FilaType (que tiene un argumento de tipo de dato compuesto FilaType (Valor))).

26. Datos: es de tipo de dato compuesto array_CatValue (que tiene dos argumentos de tipo compuesto CampoType (id_Campo) y Array_FilaType (que tiene un argumento de tipo de dato compuesto FilaType (Valor))).

27. paginado: es de un tipo de dato compuesto Paginado (offset, cantidad, cantidad_real, total)

28. codigo

29. mensaje

30. Catalogo

31. Consecutivo

32. codigo

33. mensaje

Selección de operaciones

1. DameStructCatalogo

2. DameInfoCatalogo

3. Adicionar

4. Modificar

5. Eliminar

6. UltimoConsecutivo

7. Listar

Se obtuvo un total de 60 argumentos y 7 operaciones, para un valor de APO de 8.57.

Métrica 3:

$$DAR = \frac{DAC}{n_a}$$

Pares (argumento, tipo) distintos.

(Catalogo,byte),(idCampo,string),(Nombre,string),(Tipo,string),(Tamano,byte),(Orden,byte),(Llave,boolean),(Visible,boolean),(Editable,boolean),(Sugerable,boolean),(Vacio,boolean),(Enlace,string),(Origen,string),(Filtro,boolean),(Grupo,string),(codigo,int),(mensaje,string),(Alias,byte),(Cardinal,byte),(Valor,string),(ok,boolean),(offset,int),(Cantidad,int),(cantidad_real,int),(total,int),(Consecutivo,string).

Se obtuvo un valor de DAC de 26 pares distintos, para un valor de DAR de 0.43.

Métrica 4:

Pares (argumento,tipo)	Cant de Repeticiones
(Catalogo,byte)	6
(idCampo,string)	4
(Nombre,string)	3

Pares (argumento,tipo)	Cant de Repeticiones
(Tipo,string)	1
(Tamano,byte)	1
(Orden,byte)	1

Pares (argumento,tipo)	Cant de Repeticiones	Pares (argumento,tipo)	Cant de Repeticiones
(Llave, boolean)	1	(mensaje,string)	7
(Visible, boolean)	1	(Alias,byte)	1
(Editable, boolean)	1	(Cardinal,byte)	1
(Sugerible, boolean)	1	(Valor,string)	1
(Vacio, boolean)	1	(ok,boolean)	3
(Enlace,string)	1	(offset,int)	2
(Origen,string)	1	(Cantidad,int)	2
(Filtro,boolean)	1	(cantidad_real,int)	2
(Grupo,string)	1	(total,int)	2
(codigo,int)	7	(Consecutivo,string)	1

Tabla 4.1. Pares distintos y repeticiones

$$ARS(A) = \frac{\sum_{a \in A} |a|^2}{|A|}$$

$$ARS(A) = 3.33$$

El valor que se obtuvo al calcular el ARS fue de 3.33.

Métrica 5:

$$MIL = \frac{PCA + PCO}{n_{ad} + n_{od}}$$

$$MIL = 0.85$$

Se obtuvo una cantidad de 23 palabras completas en los identificadores de argumentos y 5 palabras completas en los identificadores de operaciones, para un valor de MIL de 0.85.

Métrica 6:

$$MEIL = \frac{PCA + PCO}{2}$$

$$MEIL = 14.$$

El valor que se obtuvo de la media de la cantidad de palabras completas de identificadores en argumentos y operaciones (MEIL) es de 14.

Métrica 7:

Para la aplicación de esta métrica se toma una muestra de cuatro argumentos y cuatro operaciones, pues hacer la combinatoria de 26 argumentos y 7 operaciones daría como resultado 325 y 21 pares distintos respectivamente y para estas cantidades sería muy difícil aplicar esta métrica de forma manual.

MSC: media de cadena común de un conjunto de identificadores A.

A (Cantidad, cantidad_real, idCampo, Catalogo)

Comb A	Lcs(x,y)	Max(x , y)
(Cantidad, cantidad_real)	8	13
(Cantidad, idCampo)	2	8
(Cantidad, Catalogo)	2	8
(cantidad_real, idCampo)	2	13
(cantidad_real, Catalogo)	2	13
(idCampo, Catalogo)	2	8

Tabla 4.2. Combinatoria de argumentos

$$MSC(A) = \frac{\sum_{(x,y) \in comb(A)} \frac{|lcs(x,y)|}{\max(|x|,|y|)}}{n*(n-1)/2}$$

$$MSC(A) = 0.28$$

Se obtuvo un valor para MSC(A) de 0.28.

MSC: media de cadena común de un conjunto de operaciones O que se define como:

O (DameStructCatalogo, DameInfoCatalogo, Adicionar, Modificar)

Comb O	Lcs(x,y)	Max(x , y)
(DameStructCatalogo, DameInfoCatalogo)	8	18
(DameStructCatalogo, Adicionar)	0	18
(DameStructCatalogo, Modificar)	2	18
(DameInfoCatalogo, Adicionar)	0	16
(DameInfoCatalogo, Modificar)	2	16
(Adicionar, Modificar)	2	9

Tabla 4.3. Combinatoria de operaciones

$$MSC(O) = \frac{\sum_{(x,y) \in comb(O)} \frac{|lcs(x,y)|}{\max(|x|,|y|)}}{n * (n - 1) / 2}$$

$$MSC(O) = 0.15$$

Se obtuvo un valor para MSC(O) de 0.15.

4.2.2. Caso 2: Servicio Web LabOnlineservice

Métrica 1:

EMI (sw) tiene valor 1, pues se cuenta con el documento WSDL.

Métrica 2:

$$APO = \frac{n_a}{n_o}$$

Selección de argumentos

1. listaPaciente: es de un tipo de dato compuesto array_PacienteLab(que tiene un argumento PacienteLab que es de tipo de dato compuesto PacienteLabType(esta formado por 4 argumentos de tipos de datos compuestos DatosPersonalesType(IdPersona, Identificacion, Nombre, ApellidoPaterno, ApellidoMaterno, Fecha_Nac, Edad, Sexo, Provincia_Nac, Pais_Nac, Grupo_Sanguineo, IdRaza), DatosDemograficosType(Calle, Barrio, CP, Telefono, Nu_Int, Nu_Ext, Provincia, Municipio, Pais), DatosSeguroType(NSS, Calidad y un argumento de tipo compuesto array_AseguradoraType(con un argumento de tipo compuesto AseguradoraType(Nu_Cuenta, Empresa_Aseguradora, Cobertura, TipoSeguro))) y DatosLaboralesType(IdOcupacion, Ocupacion) respectivamente)).

2. resultado

3. codigo

4. mensaje

5. listaSolicitudLab: es de un tipo de dato compuesto array_SolicitudLab(que esta formado por un argumento de tipo de dato compuesto SolicitudLabType(esta formado por 6 argumentos de tipos de datos compuestos DatosminPacienteType(IdPaciente, Nombre, Sexo, Edad, Peso, IdRaza,

Identificacion),DatosUnidadType(Delegacion,Unidad,Servicio),DatosUnidadType(Delegacion,Unidad,Servicio),array_DatosSolicitudType(que tiene un atributo de tipo de dato compuesto DatosSolicitudType(Folio, Fecha, Fecha_Cita, Observaciones, Id_Prioridad, Fecha_Entrega_Resultados, Estado, Id_Diagnostico_Presuntivo, Diagnostico_Presuntivo, Folio_Ref y 2 atributos de tipos de datos compuestos DeclaracionMedicaType(Area_Salud, Direccion, Sintomatologia, IdRiesgo, Otros_Datos) y array_ExamenesType(que tiene un atributo de tipo de dato compuesto ExamenType(Estudio, Importe, Unidad_Monetaria, Nu_Toma, Observaciones_Cita))),DatosMedicoType(Matricula,IdRol),DatosPacienteUnidadType(Piso, Cama, Nu_Afiliacion, Nu_Consultorio, Prox_Consulta) respectivamente)).

6. resultado

7. listaResultadosLab: es de un tipo de dato compuesto array_ResultadosLab(que tiene un atributo con tipo de dato compuesto ResultadosLabType(esta compuesto por dos atributos de tipos de datos compuestos DatosminPacienteType(IdPaciente, Nombre,Sexo,Edad, Peso, IdRaza, Identificacion) y array_DatosResultadosType(que esta formado por un atributo de tipo de dato compuesto DatosResultadosType(Folio, Estudio, Componente, Fecha, Fecha_Resultado, Fecha_Validacion, Matricula_Med_Valida, Id_Equipo, Nu_Equipo, Observaciones, Resultado)) respectivamente)).

8. resultado

9. codigo

10. mensaje

11. paginado: es de un tipo de dato compuesto Paginado (offset, cantidad, cantidad_real, total).

12. criterioResultado: es de un tipo de dato compuesto ResultadosLabType (esta compuesto por dos atributos de tipos de datos compuestos DatosminPacienteType (IdPaciente, Nombre, Sexo, Edad, Peso, IdRaza, Identificacion) y array_DatosResultadosType (que esta formado por un atributo de tipo de dato compuesto DatosResultadosType (Folio, Estudio, Componente, Fecha, Fecha_Resultado, Fecha_Validacion, Matricula_Med_Valida, Id_Equipo, Nu_Equipo, Observaciones, Resultado).

13. unidad: es de un tipo de dato compuesto DatosUnidadType (Delegacion, Unidad, Servicio).

14. paginado: es de un tipo de dato compuesto Paginado (offset, cantidad, cantidad_real, total).

15. listaResultadosLab: es de un tipo de dato compuesto array_ResultadosLab(que tiene un atributo con tipo de dato compuesto ResultadosLabType(esta compuesto por dos atributos de tipos de datos compuestos DatosminPacienteType(IdPaciente, Nombre,Sexo,Edad, Peso, IdRaza, Identificacion) y array_DatosResultadosType(que esta formado por un atributo de tipo de dato compuesto DatosResultadosType(Folio, Estudio, Componente, Fecha, Fecha_Resultado, Fecha_Validacion, Matricula_Med_Valida, Id_Equipo, Nu_Equipo, Observaciones, Resultado)) respectivamente)).

16. codigo

17. mensaje

18. codigo

19. mensaje

20. paginado: es de un tipo de dato compuesto Paginado (offset, cantidad, cantidad_real, total).

21. criterio: es de un tipo de dato compuesto criterioSolicitud(esta compuesto por una extensión base que es de tipo de dato compuesto SolicitudLabType(esta formado por 6 argumentos de tipos de datos compuestos DatosminPacienteType(IdPaciente, Nombre,Sexo,Edad, Peso, IdRaza, Identificacion),DatosUnidadType(Delegacion,Unidad,Servicio),DatosUnidadType(Delegacion,Unidad, Servicio),array_DatosSolicitudType(que tiene un atributo de tipo de dato compuesto DatosSolicitudType(Folio, Fecha, Fecha_Cita, Observaciones, Id_Prioridad, Fecha_Entrega_Resultados, Estado, Id_Diagnostico_Presuntivo, Diagnostico_Presuntivo, Folio_Ref y 2 atributos de tipos de datos compuestos DeclaracionMedicaType(Area_Salud, Direccion, Sintomatologia, IdRiesgo, Otros_Datos) y array_ExamenesType(que tiene un atributo de tipo de dato compuesto ExamenType(Estudio, Importe, Unidad_Monetaria, Nu_Toma, Observaciones_Cita))),DatosMedicoType(Matricula,IdRol),DatosPacienteUnidadType(Piso, Cama, Nu_Afiliacion, Nu_Consultorio, Prox_Consulta) respectivamente)).

22. paginado: es de un tipo de dato compuesto Paginado (offset, cantidad, cantidad_real, total).

23. listaSolicitudLab: es de un tipo de dato compuesto array_SolicitudLab(que esta formado por un argumento de tipo de dato compuesto SolicitudLabType(esta formado por 6 argumentos de tipos de datos compuestos DatosminPacienteType(IdPaciente, Nombre,Sexo,Edad, Peso, IdRaza, Identificacion),DatosUnidadType(Delegacion,Unidad,Servicio),DatosUnidadType(Delegacion,Unidad, Servicio),array_DatosSolicitudType(que tiene un atributo de tipo de dato compuesto DatosSolicitudType(Folio, Fecha, Fecha_Cita, Observaciones, Id_Prioridad, Fecha_Entrega_Resultados, Estado, Id_Diagnostico_Presuntivo, Diagnostico_Presuntivo, Folio_Ref y 2 atributos de tipos de datos compuestos DeclaracionMedicaType(Area_Salud, Direccion, Sintomatologia, IdRiesgo, Otros_Datos) y array_ExamenesType(que tiene un atributo de tipo de dato compuesto ExamenType(Estudio, Importe, Unidad_Monetaria, Nu_Toma, Observaciones_Cita))),DatosMedicoType(Matricula,IdRol),DatosPacienteUnidadType(Piso, Cama, Nu_Afiliacion, Nu_Consultorio, Prox_Consulta) respectivamente)).

24. codigo

25. mensaje

26. paginado: es de un tipo de dato compuesto Paginado (offset, cantidad, cantidad_real, total).

27. criterio: es de un tipo de dato compuesto criterioPaciente(esta compuesto por una extensión base que es de tipo de dato compuesto PacienteLabType(esta formado por 4 argumentos de tipos de datos compuestos DatosPersonalesType(IdPersona, Identificacion, Nombre, ApellidoPaterno, ApellidoMaterno, Fecha_Nac, Edad, Sexo, Provincia_Nac, Pais_Nac, Grupo_Sanguineo, IdRaza),DatosDemograficosType(Calle, Barrio,CP, Telefono, Nu_Int, Nu_Ext, Provincia,Municipio,Pais),DatosSeguroType(NSS, Calidad y un argumento de tipo compuesto array_AseguradoraType(con un argumento de tipo compuesto AseguradoraType(Nu_Cuenta, Empresa_Aseguradora, Cobertura,TipoSeguro))) y DatosLaboralesType(IdOcupacion, Ocupacion) respectivamente)).

28. paginado: es de un tipo de dato compuesto Paginado (offset, cantidad, cantidad_real, total).

29. resultado: es de un tipo de dato compuesto array_PacienteLab(que tiene un argumento PacienteLab que es de tipo de dato compuesto PacienteLabType(esta formado por 4 argumentos de tipos de datos compuestos DatosPersonalesType(IdPersona, Identificacion, Nombre, ApellidoPaterno, ApellidoMaterno, Fecha_Nac, Edad, Sexo, Provincia_Nac, Pais_Nac, Grupo_Sanguineo, IdRaza),DatosDemograficosType(Calle, Barrio,CP, Telefono, Nu_Int, Nu_Ext, Provincia,Municipio,Pais),DatosSeguroType(NSS, Calidad y un argumento de tipo compuesto array_AseguradoraType(con un argumento de tipo compuesto AseguradoraType(Nu_Cuenta, Empresa_Aseguradora, Cobertura,TipoSeguro))) y DatosLaboralesType(IdOcupacion, Ocupacion) respectivamente)).

30. codigo

31. mensaje

32. listaSolicitud: es de un tipo de dato compuesto array_DatosMinSolicitud (que tiene un argumento de tipo de dato compuesto DatosMinSolicitudType (IdPaciente, IdFolio, Fecha, IdFolio_Ref)).

33. resultado

34. codigo

35. mensaje

36. unidad: es de un tipo de dato compuesto DatosUnidadType (Delegacion, Unidad, Servicio).

37. folio

38. codigo

39. mensaje

40. paginado: es de un tipo de dato compuesto Paginado (offset, cantidad, cantidad_real, total).

41. criterio: es de un tipo de dato compuesto criterioSolicitud(esta compuesto por una extensión base que es de tipo de dato compuesto SolicitudLabType(esta formado por 6 argumentos de tipos de

datos compuestos DatosminPacienteType(IdPaciente, Nombre, Sexo, Edad, Peso, IdRaza, Identificacion), DatosUnidadType(Delegacion, Unidad, Servicio), DatosUnidadType(Delegacion, Unidad, Servicio), array_DatosSolicitudType(que tiene un atributo de tipo de dato compuesto DatosSolicitudType(Folio, Fecha, Fecha_Cita, Observaciones, Id_Prioridad, Fecha_Entrega_Resultados, Estado, Id_Diagnostico_Presuntivo, Diagnostico_Presuntivo, Folio_Ref y 2 atributos de tipos de datos compuestos DeclaracionMedicaType(Area_Salud, Direccion, Sintomatologia, IdRiesgo, Otros_Datos) y array_ExamenesType(que tiene un atributo de tipo de dato compuesto ExamenType(Estudio, Importe, Unidad_Monetaria, Nu_Toma, Observaciones_Cita))), DatosMedicoType(Matricula, IdRol), DatosPacienteUnidadType(Piso, Cama, Nu_Afiliacion, Nu_Consultorio, Prox_Consulta) respectivamente)).

42. paginado: es de un tipo de dato compuesto Paginado (offset, cantidad, cantidad_real, total).

43. listaSolicitudLab: es de un tipo de dato compuesto array_SolicitudLab(que esta formado por un argumento de tipo de dato compuesto SolicitudLabType(esta formado por 6 argumentos de tipos de datos compuestos DatosminPacienteType(IdPaciente, Nombre, Sexo, Edad, Peso, IdRaza, Identificacion), DatosUnidadType(Delegacion, Unidad, Servicio), DatosUnidadType(Delegacion, Unidad, Servicio), array_DatosSolicitudType(que tiene un atributo de tipo de dato compuesto DatosSolicitudType(Folio, Fecha, Fecha_Cita, Observaciones, Id_Prioridad, Fecha_Entrega_Resultados, Estado, Id_Diagnostico_Presuntivo, Diagnostico_Presuntivo, Folio_Ref y 2 atributos de tipos de datos compuestos DeclaracionMedicaType(Area_Salud, Direccion, Sintomatologia, IdRiesgo, Otros_Datos) y array_ExamenesType(que tiene un atributo de tipo de dato compuesto ExamenType(Estudio, Importe, Unidad_Monetaria, Nu_Toma, Observaciones_Cita))), DatosMedicoType(Matricula, IdRol), DatosPacienteUnidadType(Piso, Cama, Nu_Afiliacion, Nu_Consultorio, Prox_Consulta) respectivamente)).

44. codigo

45. mensaje

Selección de operaciones

1. RegistroPaciente
2. BuscarRegistroPaciente
3. RegistroSolicitud
4. SolicitarEstudios
5. RegistroResultados
6. SolicitarResultados

7. ConfirmarReciboSolicitud

8. FolioConsecutivoSolicitud

9. BuscarPendienteResultados

Se obtuvo un total de 414 argumentos y 9 operaciones, para un valor de APO de 46.

Métrica 3:

$$DAR = \frac{DAC}{n_a}$$

Pares (argumento, tipo) distintos

(IdPersona,string) ,(Identificacion, string), (Nombre, string),(ApellidoPaterno, string) ,(ApellidoMaterno, string),(Fecha_Nac,date),(Edad, string),(Sexo,byte),(Provincia_Nac, string),(Pais_Nac,byte),(Grupo_Sanguineo,byte),(IdRaza,byte),(Calle, string),(Barrio, string),(CP, string),(Telefono, string),(Nu_Int, string),(Nu_Ext, string), (Provincia, string),(Municipio, string),(Pais,byte),(NSS, string),(Calidad,byte),(Nu_Cuenta, string),(Empresa_Aseguradora, string), (Cobertura, string),(TipoSeguro, string),(IdOcupacion, string),(Ocupacion, string),(resultado,boolean),(codigo,integer),(mensaje,string),(IdPaciente,string),(Peso,float), (Identificacion,string),(Delegacion,string),(Unidad,string),(Servicio,string),(Folio,string),(Fecha,date),(Fecha_Cita,date),(Observaciones,string),(Id_Prioridad,byte),(Fecha_Entrega_Resultados,date),(Estado,byte),(Id_Diagnostico_Presuntivo,string),(Diagnostico_Presuntivo,string),(Folio_Ref,string),(Area_Salud, string),(Direccion,string),(Sintomatologia, string),(IdRiesgo, integer),(Otros_Datos, string),(Estudio, string),(Importe,float),(Unidad_Monetaria,string),(Nu_Toma,byte),(Observaciones_Cita,string),(Matricula, string),(IdRol,byte),(Piso, string), (Cama, string),(Nu_Afiliacion, string), (Nu_Consultorio, string), (Prox_Consulta,date),(Componente,string),(Fecha_Resultado,date),(Fecha_Validacion,date), (Matricula_Med_Valida,string),(Id_Equipo,string),(Nu_Equipo,string),(Resultado,string),(offset,integer), (cantidad, integer),(cantidad_real, integer),(total, integer),(IdFolio,string),(IdFolio_Ref,string).

Se obtuvo un valor de DAC de 78 pares distintos, para un valor de DAR de 0.19.

CAPÍTULO 4: ANÁLISIS DE LOS RESULTADOS

Métrica 4:

Pares (argumento,tipo)	Cant de Rep.
(IdPersona,string)	3
(Identificacion, string)	3
(Nombre, string)	13
(ApellidoPaterno, string)	3
(ApellidoMaterno, string)	3
(Fecha_Nac,date)	3
(Edad, string)	13
(Sexo,byte)	13
(Provincia_Nac, string)	3
(Pais_Nac,byte)	3
(Grupo_Sanguineo,byte)	3
(IdRaza,byte)	13
(Calle, string)	3
(Barrio, string)	3
(CP, string)	3
(Telefono, string)	3
(Nu_Int, string)	3
(Nu_Ext, string)	3
(Provincia, string)	3
(Municipio, string)	3
(Pais,byte)	3
(NSS, string)	3

Pares (argumento,tipo)	Cant de Rep.
(Calidad,byte)	3
(Nu_Cuenta, string)	3
(Empresa_Aseguradora, string)	3
(Cobertura, string)	3
(TipoSeguro, string)	3
(IdOcupacion, string)	3
(Ocupacion, string)	3
(resultado,boolean)	4
(codigo,integer)	9
(mensaje,string)	9
(IdPaciente,string)	11
(Peso,float)	7
(Identificacion,string)	7
(Delegacion,string)	9
(Unidad,string)	9
(Servicio,string)	9

Pares (argumento,tipo)	Cant de Rep.
(Folio,string)	9
(Fecha,date)	9
(Fecha_Cita,date)	5
(Observaciones,string)	8

CAPÍTULO 4: ANÁLISIS DE LOS RESULTADOS

Pares (argumento,tipo)	Cant de Rep.
(Id_Prioridad,byte)	5
(Fecha_Entrega_Resultados, date)	5
(Estado,byte)	5
(Id_Diagnostico_Presuntivo, string)	5
(Diagnostico_Presuntivo, string)	5
(Folio_Ref,string)	5
(Area_Salud,string)	5
(Direccion,string)	5
(Sintomatologia, string)	5
(IdRiesgo, integer)	5
(Otros_Datos, string)	5
(Estudio, string)	8
(Importe,float)	5
(Unidad_Monetaria,string)	5
(Nu_Toma,byte)	5
(Observaciones_Cita,string)	5
(Matricula,string)	5

Pares (argumento,tipo)	Cant de Rep.
(IdRol,byte)	5
(Piso, string)	5
(Cama, string)	5
(Nu_Afiliacion, string)	5
(Nu_Consultorio, string)	5
(Prox_Consulta,date)	5
(Componente,string)	3
(Fecha_Resultado,date)	3
(Fecha_Validacion,date)	3
(Matricula_Med_Valida, string)	3
(Id_Equipo,string)	3
(Nu_Equipo,string)	3
(Resultado,string)	3
(offset,integer)	4
(cantidad, integer)	4
(cantidad_real, integer)	4
(total, integer)	4
(IdFolio,string)	1
(IdFolio_Ref,string)	1

Tabla 4.4. Pares distintos y repeticiones

$$ARS(A) = \frac{\sum_{a \in A} |a|^2}{|A|}$$

$$ARS(A) = 6.08$$

El valor que se obtuvo al calcular el ARS fue de 6.08.

Métrica 5:

$$MIL = \frac{PCA + PCO}{n_{ad} + n_{od}}$$

$$MIL = 0.86$$

Se obtuvo una cantidad de 66 palabras completas en los identificadores de argumentos y 9 palabras completas en los identificadores de operaciones, para un valor de MIL de 0.86.

Métrica 6:

$$MEIL = \frac{PCA + PCO}{2}$$

$$MEIL = 37.5$$

El valor que se obtuvo de la media de la cantidad de palabras completas de identificadores en argumentos y operaciones (MEIL) es de 37.5.

Métrica 7:

Para la aplicación de esta métrica se toma una muestra de cuatro argumentos y cuatro operaciones, pues hacer la combinatoria de 78 argumentos y 9 operaciones daría como resultado 3003 y 36 pares distintos respectivamente, y para estas cantidades sería muy difícil aplicar esta métrica de forma manual.

MSC: media de cadena común de un conjunto de argumentos A.

A (ApellidoPaterno, ApellidoMaterno, Fecha_Nac, Pais_Nac)

Comb A	Lcs(x,y)	Max(x , y)
(ApellidoPaterno, ApellidoMaterno)	8	15
(ApellidoPaterno, Fecha_Nac)	0	15

Comb A	Lcs(x,y)	Max(x , y)
(ApellidoPaterno, Pais_Nac)	2	15
(ApellidoMaterno, Fecha_Nac)	0	15
(ApellidoMaterno, Pais_Nac)	0	15
(Fecha_Nac, Pais_Nac)	4	9

Tabla 4.5. Combinatoria de argumentos

$$MSC(A) = \frac{\sum_{(x,y) \in comb(A)} \frac{|lcs(x,y)|}{\max(|x|,|y|)}}{n*(n-1)/2}$$

$$MSC(A) = 0.19$$

Se obtuvo un valor para MSC(A) de 0.19.

MSC: media de cadena común de un conjunto de operaciones O que se define como:

Comb (O)	Lcs(x,y)	Max(x , y)
(RegistroPaciente, BuscarRegistroPaciente)	16	22
(RegistroPaciente, RegistroSolicitud)	8	17
(RegistroPaciente, SolicitarEstudios)	2	17

CAPÍTULO 4: ANÁLISIS DE LOS RESULTADOS

Comb (O)	Lcs(x,y)	Max(x , y)
(BuscarRegistroPaciente,RegistroSolicitud)	8	22
(BuscarRegistroPaciente,SolicitarEstudios)	2	22
(RegistroSolicitud, SolicitarEstudios)	7	17

Tabla 4.6. Combinatoria de operaciones

$$MSC(O) = \frac{\sum_{(O_1, O_2) \in comb(O)} \frac{Lcs(O_1, O_2)}{\max(|x|, |y|)}}{n \cdot (n-1) / 2}$$

$$MSC(O) = 0.36$$

Se obtuvo un valor para MSC(O) de 0.36.

4.3. Evaluación de los resultados obtenidos

Del estudio de las métricas propuestas, aplicadas en varios componentes, se cuenta con un valor promedio por cada una de ellas, el cual aparece en las siguientes tablas. Estos valores se toman como referencia para evaluar la reusabilidad en los dos casos de estudio anteriores.

Component	Arguments	Operations	APO	Median APO
WSDL	199208	36187	5.50	3.07
CoreAudio[5]	42	134	3.19	N/A
DB [5]	210	404	1.92	N/A
MFC[5]	5907	7735	1.31	N/A
Eclipse	144106	136762	1.05	0.67

Tabla. 4.7. Lista de valores promedios para la métrica APO en varios componentes.

Libraries	DAC	DAR
WSDL	17	0.54
DB [5]	129	0.32
CoreAudio [5]	33	0.25
MFC [5]	2363	0.31

Tabla. 4.8. Lista de valores promedios para la métrica DAR en varios componentes.

Libraries	DAC	ARS
WSDL	17	1.9
DB [5]	129	12.55
CoreAudio [5]	33	9.06
MFC [5]	2363	21.42

Tabla. 4.9. Lista de valores promedios para la métrica ARS en varios componentes.

Language	MIL	MeIL	MSC(A)	MSC(O)
WSDL	12.0	11	9.00	0.19
CoreAudio [5]	15.04	12	0.27	0.46
DB [5]	6.23	6	0.14	0.18
MFC [5]	7.69	7	0.17	0.21

Tabla. 4.10. Listado de MIL, MEIL, MSC por argumentos, denotados como MSC(A) y por operaciones, denotado como MSC(O) por el documento WSDL.

4.3.1. Caso 1: Servicio Web Catalogo_ECService

Para dar una evaluación de los resultados, en la siguiente tabla se muestra un resumen de las métricas aplicadas en cuanto al (los) criterio(s) de reusabilidad que miden y el valor que se obtuvo al ser aplicada al servicio web.

Métrica	Criterio	Valor
EMI	comprensibilidad	1
APO	tamaño	8.57
DAR	consistencia, comprensibilidad	0.43
ARS	consistencia	3.33
MIL	legibilidad, comprensibilidad, autodescriptividad	0.85
MEIL	legibilidad, comprensibilidad, autodescriptividad	14
MSC(A)	autodescriptividad	0.28
MSC(O)	autodescriptividad	0.15

Tabla. 4.11. Resumen de métricas

Evaluación por métricas

EMI:

Como el resultado es 1 se puede decir que el web service cuenta con el documento WSDL, lo que hace que sea comprensible el servicio.

APO:

El valor obtenido para el Servicio Web es mayor que el valor promedio que aparece en las tablas, de aquí que puede decirse que este es un Servicio Web de grano grueso por su tamaño. No obstante ello permite que pueda ser invocado con más frecuencia.

DAR:

Se puede decir, teniendo en cuenta la cantidad de argumentos distintos que tiene el servicio web y haciendo una comparación con los valores promedios de la Tabla 4.8, que el valor de DAR obtenido es aceptable, el cual evidencia consistencia en las declaraciones de los argumentos del servicio, lo que facilita la comprensibilidad y reuso del mismo.

ARS:

Según el valor obtenido para esta métrica se puede decir que la consistencia de las declaraciones de los argumentos no es muy buena, pues el mismo es muy elevado con respecto al valor promedio para esta métrica, obtenido del análisis de varios componentes.

MIL:

Para evaluar el resultado de esta métrica no se puede tomar como guía el promedio obtenido del análisis de varios componentes, porque en este caso se cuenta la cantidad de palabras completas y no la cantidad de caracteres en argumentos y operaciones.

Se toma como criterio que un valor pegado a 1 es el mejor valor, ya que la cantidad de palabras completas en argumentos y operaciones no va a ser mayor que la cantidad total de los mismos. Por tanto se puede decir que el servicio posee un alto grado de legibilidad, comprensibilidad y autodescriptividad.

MEIL: Comparando el valor de esta métrica con el promedio obtenido del análisis de varios componentes, se puede decir que el servicio cuenta con un buen grado de legibilidad, comprensibilidad y autodescriptividad, ya que el valor de la métrica es mayor.

MSC(A) y MSC(O):

En el caso de éstas métricas no se puede dar una evaluación pues se tomó solo una muestra para aplicarla, ya que sería muy difícil de aplicar de forma manual.

4.3.2. Caso 2: Servicio Web LabOnlineservice

Para dar una evaluación de los resultados, en la siguiente tabla se muestra un resumen de las métricas aplicadas en cuanto al (los) criterio(s) de reusabilidad que miden y el valor que se obtuvo al ser aplicada al servicio web.

Métrica	Criterio	Valor
EMI	comprensibilidad	1
APO	tamaño	46
DAR	consistencia , comprensibilidad	0.19
ARS	consistencia	6.08
MIL	legibilidad, comprensibilidad, autodescriptividad	0.86
MEIL	legibilidad, comprensibilidad, autodescriptividad	37.5
MSC(A)	autodescriptividad	0.19
MSC(O)	autodescriptividad	0.36

Tabla. 4.12. Resumen de métricas

Evaluación por métricas

EMI:

Como el resultado es 1 se puede decir que el web service cuenta con el documento WSDL, lo que hace que sea comprensible el servicio.

APO:

El valor obtenido para el Servicio Web es mayor que el valor promedio que aparece en las tablas, de aquí que puede decirse que este es un Servicio Web de grano grueso por su tamaño. No obstante ello permite que pueda ser invocado con más frecuencia.

DAR:

Se puede decir, teniendo en cuenta la cantidad de argumentos distintos que tiene el servicio web y haciendo una comparación con los valores promedios de la Tabla 4.8, que el valor de DAR obtenido es mucho menor, lo cual puede indicar que el servicio web tiene muy poca consistencia en las declaraciones de sus argumentos, lo que limita su comprensibilidad y reutilización.

ARS:

Según el valor obtenido para esta métrica se puede decir que la consistencia de las declaraciones de los argumentos no es buena, pues el mismo es muy elevado con respecto al valor promedio para esta métrica, obtenido del análisis de varios componentes.

MIL:

Para evaluar el resultado de esta métrica no se puede tomar como guía el promedio obtenido del análisis de varios componentes, porque en este caso se cuenta la cantidad de palabras completas y no la cantidad de caracteres en argumentos y operaciones.

Se toma como criterio que un valor pegado a 1 es el mejor valor, ya que la cantidad de palabras completas en argumentos y operaciones no va a ser mayor que la cantidad total de los mismos. Por tanto se puede decir que el servicio posee un alto grado de legibilidad, comprensibilidad y autodescriptividad.

MEIL:

Comparando el valor de esta métrica con el promedio obtenido del análisis de varios componentes, se puede decir que el servicio cuenta con un alto grado de legibilidad, comprensibilidad y autodescriptividad, ya que el valor de la métrica es mucho mayor.

MSC(A) y MSC(O):

En el caso de éstas métricas no se puede dar una evaluación pues se tomó solo una muestra para aplicarla, ya que sería muy difícil de aplicar de forma manual.

4.4. Conclusiones

En este capítulo se aplicó el conjunto de métricas propuestas a dos servicios web desarrollados en una empresa productora de software, las cuales evalúan los criterios de comprensibilidad, tamaño, legibilidad, consistencia y autodescriptividad para tener una medida de la reusabilidad.

4.5. Recomendaciones

Se recomienda que se apliquen éstas métricas a todos los servicios web desarrollados en la empresa y que se incluya el cálculo de las mismas como parte de la información en el repositorio actual que tienen en la empresa.

CONCLUSIONES

Para cumplir con el objetivo de este trabajo de diploma se le aplicó una encuesta a especialistas y programadores de una empresa cubana productora de software, para corroborar su situación actual en cuanto a la necesidad de tener una medida de la reusabilidad de los servicios web que se desarrollan en la misma. Como solución a esta situación, quedaron propuestas un conjunto de métricas que miden criterios como la consistencia, legibilidad, comprensibilidad, tamaño y autodescriptividad de los servicios web utilizando el documento WSDL de los mismos. Las mismas fueron aplicadas a dos casos de estudio dando una valoración del comportamiento de cada criterio en ambos casos.

RECOMENDACIONES

Tomando como base la investigación realizada y la experiencia acumulada durante la realización de este trabajo, se proponen las siguientes recomendaciones:

- ❖ Ampliar las métricas propuestas con métricas que cubran otros criterios, y de ésta forma poder dar una valoración general de la reusabilidad de un producto web service.
- ❖ Crear una herramienta que facilite el cálculo de las métricas propuestas ya que de forma manual el procedimiento es engorroso.
- ❖ Incluir los resultados de la aplicación de las métricas propuestas y su interpretación en el repositorio de componentes de la empresa.

REFERENCIAS BIBLIOGRAFICAS

1. **Canut, Carlos Granell.** Reutilización de servicios web mediante componentes integrados. [Online] 2006. http://biblioteca.universia.net/html_bura/ficha/params/id/9443010.html.
2. **Alvarez, Juan Carlos Granja.** Un horizonte en el ámbito de la Calidad del Software. [Online] 2000. <http://www.ati.es/novatica/2000/145/juagra-145.pdf>.
3. **Francisco José García Peñalvo, José Manuel Marqués Corral, Jesús Manuel Maudes Raedo.** Mecano: Una Propuesta De Componente Software Reutilizable. [Online] 1997. <http://www.giro.infor.uva.es/oldsite/docpub/jis97.html>.
4. —. *Análisis y Diseño Orientado al Objeto para*. España : s.n., 1997.
5. COMPUTERWORLD. [Online] <http://www.idg.es/computerworld/articulo.asp?id=129033>).
6. El rincon del vago. [Online] http://html.rincondelvago.com/poo_9.html.
7. PROGRAMACIÓN ORIENTADA A COMPONENTES . [Online] <http://martileo.pcc.uasnet.mx/computacion/node27.html>.
8. Ingeniería del software basada en. [Online] <http://www.escet.urjc.es/~aisw/docs/isiiisbc.pdf>.
9. **Navarro, Manuel.** La revolución SOA. [Online] febrero 2008. <http://www.mkmpi.com/mkmpi.php?article1600>.
10. **Betancourt, Elizabeth.** *Tesina*. 2008.
11. **Martínez, Pedro Espina.** EXTENSIBILIDAD DE UDDI. [Online] sept 2007. [http://www.lsi.us.es/docs/doctorado/memorias/Pedro%20Espina%20\(Periodo%20Investigacion\).pdf](http://www.lsi.us.es/docs/doctorado/memorias/Pedro%20Espina%20(Periodo%20Investigacion).pdf).

12. **Fabio Di Donato, Grupo ASSA.** Tecnología: Web Services: la nueva tecnología para integrar organizaciones . [Online] 2007.
<http://www.grupoassa.com/Espanol/general.asp?N1=3&N2=4&N3=1&N4=0&cat=mi&reIID=284>.
13. **Saffirio, Mario.** ¿Qué son los Web Services? . [Online] febrero 2006.
<http://msaffirio.wordpress.com/2006/02/05/%C2%BFque-son-los-web-services/>.
14. **Granell Canut, Carlos.** Reutilización de servicios web mediante componentes integrados . [Online] julio 2006. http://www.tesisenred.net/TDX-0913106-133147/index_cs.html.
15. **Kelvin H.T. Choi, Ewan Tempero.** Dynamic Measurement of Polymorphism. [Online]
<http://crpit.com/confpapers/CRPITV62Choi.pdf>.
16. **Arregui, Juan José Olmedilla.** Revisión Sistemática de Métricas de. [Online] sept 2005.
<http://is.ls.fi.upm.es/doctorado/Trabajos20042005/Olmedilla.pdf>.
17. **Washizaki, H. Yamamoto, H. Fukazawa, Y.** A metrics suite for measuring reusability of software components. [Online] Sept. 3-5 , 2003.
<http://dspace.wul.waseda.ac.jp/dspace/bitstream/2065/822/1/01232469.pdf>.
18. **EPS, Informática.** *Reutilización*. España : s.n.
19. **Yijun Yu Jianguo Lu Fernandez-Ramil, J. Yuan, P.** Comparing Web Services with other Software Components. [Online] julio 30, 2007. <http://cs.uwindsor.ca/~jlu/wsdl-oo-comparisonSubmit.pdf>.

BIBLIOGRAFÍA

F, Luis Armando Canchala. Conceptos Fundamentales de la POO. [Online] 2004.

http://www.elguille.info/colabora/puntoNET/canchala_FundamentosPOO.htm.

Harrison, Daniel Rodríguez y Rachel. MEDICIÓN EN LA ORIENTACIÓN A OBJETOS. [Online]

<http://www.cc.uah.es/drg/b/RodHarRama00.pdf>.

Maria de los Angeles Martin. Sistema de Catalogacion de Metricas e indicadores con Potencia de

Web semantica. [Online] sept 2004. <http://gidis.ing.unlpam.edu.ar/home/downloads/Tesis-Martin.pdf>.

Arregui, Juan José Olmedilla. Revisión Sistemática de Métricas de Diseño Orientado a Objetos .

[Online] sept 2005. <http://is.ls.fi.upm.es/doctorado/Trabajos20042005/Olmedilla.pdf>.

M. Esperanza Manso Martínez, Francisco J. García Peñalvo, Juan José Rodríguez Díez, Miguel

A. Laguna Serrano. Modelo de Cualificación de Assets del repositorio GIRO. [Online] 1998.

<http://www.giro.infor.uva.es/Publications/1998/MGRL98/menhir-3-98-cualificacion.pdf>.

Alvarez, Juan Carlos Granja. Un horizonte en el ámbito de la Calidad del Software. [Online] 2000.

<http://www.ati.es/novatica/2000/145/juagra-145.pdf>.

Washizaki, H. Yamamoto, H. Fukazawa, Y. A metrics suite for measuring reusability of software components. [Online] Sept. 3-5 , 2003.

<http://dspace.wul.waseda.ac.jp/dspace/bitstream/2065/822/1/01232469.pdf>.

Yijun Yu Jianguo Lu Fernandez-Ramil, J. Yuan, P. Comparing Web Services with other Software

Components. [Online] julio 30, 2007. <http://cs.uwindsor.ca/~jlu/wsd1-oo-comparisonSubmit.pdf>.

Kelvin H.T. Choi, Ewan Tempero. Dynamic Measurement of Polymorphism. [Online]

<http://crpit.com/confpapers/CRPITV62Choi.pdf>.

ANEXOS**Anexo #1****Encuesta realizada.****Datos de la Empresa.**

1. ¿Cuál es el rol que desempeña en la empresa?

___ Director de desarrollo

___ Jefe de proyecto

___ Gestor de proyecto

___ Analista

___ Arquitecto

___ Programador

___ Otro ¿Cuál? _____

2. ¿Cuántos años lleva usted desarrollando software?

3. ¿La empresa desarrolla únicamente software interno? Si___ No___

4. ¿Aproximadamente cuál es el número de entidades a las que ha desarrollado software? _____.

Datos uso de Reusabilidad en la Empresa.

5. Tiene algún conocimiento de:

___ Conceptos de reusabilidad. Mencione brevemente el concepto.

___ Métricas de reusabilidad. ¿Cuáles? _____

___ Modelos de reusabilidad. ¿Cuáles? _____

___ Otros temas de reusabilidad. ¿Cuáles? _____

6. ¿Se aplica alguna métrica relacionada con reusabilidad en su ámbito de trabajo? Si__ No__

7. Si la respuesta anterior es afirmativa ¿Qué métricas utilizan?

___ Orientada a Producto.

___ Orientada a Proceso.

___ Orientada a Negocio.

___ Otros niveles. ¿Cuáles?_____

8. ¿Considera que es importante reutilizar? Si__ No__

9. ¿En su ámbito de trabajo se aplica la reutilización? Si__ No__

10. Si la respuesta anterior es afirmativa, diga a que nivel aplica la reutilización.

___ Nivel de código.

___ Nivel de componentes.

___ Nivel de clases.

___ Nivel de documentación.

___ Nivel de modelos.

___ Nivel de Dominio o Negocio.

___ Otros niveles. ¿Cuáles?_____

11. ¿Conoce o tiene para su ámbito de trabajo una evaluación cuantitativa o cualitativa de cómo y cuanto se aplica la reutilización? Si__ No__

12. Si la respuesta anterior es afirmativa, diga qué métrica o procedimiento aplica.

13.- ¿Conoce o aplica alguna métrica para evaluar cuantitativa o cualitativamente la capacidad de reutilización de los componentes de software disponibles en la empresa?

Si__ No__ ¿Cuáles?_____

14.- ¿Ayudaría a su trabajo contar con datos sobre la reusabilidad de los componentes disponibles en la empresa? Si__ No__

15.- Si la respuesta anterior es afirmativa, explique cómo le ayudaría

__ Ayudaría a seleccionar los componentes adecuados para desarrollar nuevas aplicaciones.

__ Ayudaría a construir componentes para reutilizar.

__ Ayudaría a construir nuevas aplicaciones utilizando lo que ya está disponible.

__ Otros _____

16. A la hora de proponer métricas que se puedan utilizar para medir la reusabilidad ¿Qué criterios valoraría?

__ Información que produce.

__ Ayuda en la toma de decisiones.

__ Simplicidad de la métrica.

__ Facilidad para recolectar y evaluar los datos.

__ Herramientas que ayuden a recolectar y evaluar los datos.

__ Otros criterios. ¿Cuáles?_____

17. ¿Cuentan con herramientas para recolectar datos sobre los Servicios Web disponibles? Si__
No__

Menciónelas. _____

18. ¿Cuentan con herramientas para evaluar los datos recolectados de los Servicios Web
disponibles? Si__ No__

Menciónelas. _____

19. ¿Existe un proceso o mecanismo definido para recolectar datos sobre los Servicios Web
disponibles? Si__ No__

20. Si la respuesta anterior es afirmativa ¿Existe una planificación (tiempo y recursos asignados) para
el proceso o mecanismo definido para recolectar datos sobre los Servicios Web disponibles?
Si__ No__

21. ¿Cuentan con un repositorio, base de datos, mecanismo o infraestructura para guardar los datos
recolectados sobre los Servicios Web disponibles? Si__ No__

Anexo #2**Documento WSDL del servicio web Catalogo_ECEservice**

```
<!-- edited with XMLSPY v2004 rel. 2 U (http://www.xmlspy.com) by Crack (Crack) -->
<wsdl:definitions xmlns:typens="urn:Catalogo_ECE"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"
xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/" xmlns="http://schemas.xmlsoap.org/wsdl/"
targetNamespace="urn:Catalogo_ECE" name="urn:Catalogo_ECE">
  <wsdl:types>
    <schema targetNamespace="urn:Catalogo_ECE"
xmlns="http://www.w3.org/2001/XMLSchema">
      <complexType name="CatalogoInfo">
        <sequence>
          <element name="Nombre" type="string"/>
          <element name="Alias" type="byte"/>
          <element name="Array_Grupo" type="typens:Array_GrupoType"/>
        </sequence>
      </complexType>
      <complexType name="CatalogoStruct">
        <sequence>
          <element name="idCampo" type="string"/>
          <element name="Nombre" type="string"/>
          <element name="Tipo" type="string"/>
          <element name="Tamano" type="byte"/>
          <element name="Orden" type="byte"/>
          <element name="Llave" type="boolean"/>
          <element name="Visible" type="boolean"/>
          <element name="Editable" type="boolean"/>
          <element name="Sugerible" type="boolean"/>
          <element name="Vacio" type="boolean"/>
          <element name="Enlace" type="string"/>
        </sequence>
      </complexType>
    </schema>
  </wsdl:types>
</wsdl:definitions>
```

```
        <element name="Origen" type="string"/>
        <element name="Filtro" type="boolean"/>
        <element name="Grupo" type="string"/>
    </sequence>
</complexType>
<complexType name="array_CatalogoStruct">
    <sequence maxOccurs="unbounded">
        <element name="CatalogoStruct" type="typens:CatalogoStruct"/>
    </sequence>
</complexType>
<complexType name="array_CatalogoInfo">
    <sequence maxOccurs="unbounded">
        <element name="CatalogoInfo" type="typens:CatalogoInfo"
nillable="false"/>
    </sequence>
</complexType>
<complexType name="array_CatValue">
    <sequence>
        <element name="Campo" type="typens:CampoType"/>
        <element name="Array_Fila" type="typens:Array_FilaType"/>
    </sequence>
</complexType>
<complexType name="CampoType">
    <sequence maxOccurs="unbounded">
        <element name="id_Campo" type="string"/>
    </sequence>
</complexType>
<complexType name="FilaType">
    <sequence maxOccurs="unbounded">
        <element name="Valor" type="string"/>
    </sequence>
</complexType>
<complexType name="Paginado">
```

```
<sequence>
  <element name="offset" type="int"/>
  <element name="cantidad" type="int"/>
  <element name="cantidad_real" type="int"/>
  <element name="total" type="int"/>
</sequence>
</complexType>
<complexType name="Array_FilaType">
  <sequence maxOccurs="unbounded">
    <element name="Fila" type="typens:FilaType"/>
  </sequence>
</complexType>
<complexType name="GrupoType">
  <sequence>
    <element name="Nombre" type="string"/>
    <element name="Cardinal" type="byte"/>
  </sequence>
</complexType>
<complexType name="Array_GruposType">
  <sequence maxOccurs="unbounded">
    <element name="Grupo" type="typens:GrupoType"/>
  </sequence>
</complexType>
</schema>
</wsdl:types>
<wsdl:message name="DameStructCatalogoRequest">
  <wsdl:part name="Catalogo" type="xsd:byte"/>
</wsdl:message>
<wsdl:message name="DameStructCatalogoResponse">
  <wsdl:part name="estructura" type="typens:array_CatalogoStruct"/>
</wsdl:message>
<wsdl:message name="DameStructCatalogoFault">
  <wsdl:part name="codigo" type="xsd:int"/>
</wsdl:message>
```

```
<wsdl:part name="mensaje" type="xsd:string"/>
</wsdl:message>
<wsdl:message name="DameInfoCatalogoRequest"/>
<wsdl:message name="DameInfoCatalogoResponse">
  <wsdl:part name="InfoCatalogos" type="typens:array_CatalogoInfo"/>
</wsdl:message>
<wsdl:message name="DameInfoCatalogoFault">
  <wsdl:part name="codigo" type="xsd:int"/>
  <wsdl:part name="mensaje" type="xsd:string"/>
</wsdl:message>
<wsdl:message name="AdicionarRequest">
  <wsdl:part name="Catalogo" type="xsd:byte"/>
  <wsdl:part name="Datos" type="typens:array_CatValue"/>
</wsdl:message>
<wsdl:message name="AdicionarResponse">
  <wsdl:part name="ok" type="xsd:boolean"/>
</wsdl:message>
<wsdl:message name="AdicionarFault">
  <wsdl:part name="codigo" type="xsd:int"/>
  <wsdl:part name="mensaje" type="xsd:string"/>
</wsdl:message>
<wsdl:message name="ModificarRequest">
  <wsdl:part name="Catalogo" type="xsd:byte"/>
  <wsdl:part name="Datos" type="typens:array_CatValue"/>
</wsdl:message>
<wsdl:message name="ModificarResponse">
  <wsdl:part name="ok" type="xsd:boolean"/>
</wsdl:message>
<wsdl:message name="ModificarFault">
  <wsdl:part name="codigo" type="xsd:int"/>
  <wsdl:part name="mensaje" type="xsd:string"/>
</wsdl:message>
<wsdl:message name="EliminarRequest">
```

```
<wsdl:part name="Catalogo" type="xsd:byte"/>
  <wsdl:part name="Datos" type="typens:array_CatValue"/>
</wsdl:message>
<wsdl:message name="EliminarResponse">
  <wsdl:part name="ok" type="xsd:boolean"/>
</wsdl:message>
<wsdl:message name="EliminarFault">
  <wsdl:part name="codigo" type="xsd:int"/>
  <wsdl:part name="mensaje" type="xsd:string"/>
</wsdl:message>
<wsdl:message name="ListarRequest">
  <wsdl:part name="Catalogo" type="xsd:byte"/>
  <wsdl:part name="paginado" type="typens:Paginado"/>
  <wsdl:part name="criterio" type="typens:array_CatValue"/>
</wsdl:message>
<wsdl:message name="ListarResponse">
  <wsdl:part name="Datos" type="typens:array_CatValue"/>
  <wsdl:part name="paginado" type="typens:Paginado"/>
</wsdl:message>
<wsdl:message name="ListarFault">
  <wsdl:part name="codigo" type="xsd:int"/>
  <wsdl:part name="mensaje" type="xsd:string"/>
</wsdl:message>
<wsdl:message name="UltimoConsecutivoRequest">
  <wsdl:part name="Catalogo" type="xsd:byte"/>
</wsdl:message>
<wsdl:message name="UltimoConsecutivoResponse">
  <wsdl:part name="Consecutivo" type="xsd:string"/>
</wsdl:message>
<wsdl:message name="UltimoConsecutivoFault">
  <wsdl:part name="codigo" type="xsd:int"/>
  <wsdl:part name="mensaje" type="xsd:string"/>
</wsdl:message>
```

```
<portType name="Catalogo_ECEPort">
  <operation name="DameStructCatalogo">
    <input message="typens:DameStructCatalogoRequest"/>
    <output message="typens:DameStructCatalogoResponse"/>
    <wsdl:fault name="FaultName" message="typens:DameStructCatalogoFault"/>
  </operation>
  <operation name="DameInfoCatalogo">
    <input message="typens:DameInfoCatalogoRequest"/>
    <output message="typens:DameInfoCatalogoResponse"/>
    <wsdl:fault name="FaultName" message="typens:DameInfoCatalogoFault"/>
  </operation>
  <operation name="Adicionar">
    <input message="typens:AdicionarRequest"/>
    <output message="typens:AdicionarResponse"/>
    <wsdl:fault name="FaultName" message="typens:AdicionarFault"/>
  </operation>
  <operation name="Modificar">
    <input message="typens:ModificarRequest"/>
    <output message="typens:ModificarResponse"/>
    <wsdl:fault name="FaultName" message="typens:ModificarFault"/>
  </operation>
  <operation name="Eliminar">
    <input message="typens:EliminarRequest"/>
    <output message="typens:EliminarResponse"/>
    <wsdl:fault name="FaultName" message="typens:EliminarFault"/>
  </operation>
  <operation name="UltimoConsecutivo">
    <input message="typens:UltimoConsecutivoRequest"/>
    <output message="typens:UltimoConsecutivoResponse"/>
    <wsdl:fault name="FaultName" message="typens:UltimoConsecutivoFault"/>
  </operation>
  <operation name="Listar">
    <input message="typens:ListarRequest"/>
```

```
        <output message="typens:ListarResponse"/>
        <wsdl:fault name="FaultName" message="typens:ListarFault"/>
    </operation>
</portType>
<binding name="Catalogo_ECEBinding" type="typens:Catalogo_ECEPort">
    <soap:binding style="rpc" transport="http://schemas.xmlsoap.org/soap/http"/>
    <operation name="DameStructCatalogo">
        <soap:operation soapAction="urn:Catalogo_ECEAction" style="rpc"/>
        <input>
            <soap:body use="encoded"
encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" namespace="urn:Catalogo_ECE"/>
        </input>
        <output>
            <soap:body use="encoded"
encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" namespace="urn:Catalogo_ECE"/>
        </output>
        <fault name="FaultName">
            <soap:fault name="DameStructCatalogoFault" use="encoded"
encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" namespace="urn:Catalogo_ECE"/>
        </fault>
    </operation>
    <operation name="DameInfoCatalogo">
        <soap:operation soapAction="urn:Catalogo_ECEAction" style="rpc"/>
        <input>
            <soap:body use="encoded"
encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" namespace="urn:Catalogo_ECE"/>
        </input>
        <output>
            <soap:body use="encoded"
encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" namespace="urn:Catalogo_ECE"/>
        </output>
        <fault name="FaultName">
```

```

                <soap:fault          name="DameInfoCatalogoFault"          use="encoded"
encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" namespace="urn:Catalogo_ECE"/>
                </fault>
        </operation>
        <operation name="Adicionar">
                <soap:operation soapAction="urn:Catalogo_ECEAction" style="rpc"/>
                <input>
                        <soap:body          use="encoded"
encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" namespace="urn:Catalogo_ECE"/>
                </input>
                <output>
                        <soap:body          use="encoded"
encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" namespace="urn:Catalogo_ECE"/>
                </output>
                <fault name="FaultName">
                        <soap:fault          name="AdicionarFault"          use="encoded"
encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" namespace="urn:Catalogo_ECE"/>
                </fault>
        </operation>
        <operation name="Modificar">
                <soap:operation soapAction="urn:Catalogo_ECEAction" style="rpc"/>
                <input>
                        <soap:body          use="encoded"
encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" namespace="urn:Catalogo_ECE"/>
                </input>
                <output>
                        <soap:body          use="encoded"
encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" namespace="urn:Catalogo_ECE"/>
                </output>
                <fault name="FaultName">
                        <soap:fault          name="ModificarFault"          use="encoded"
encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" namespace="urn:Catalogo_ECE"/>
                </fault>

```

```
</operation>
<operation name="Eliminar">
  <soap:operation soapAction="urn:Catalogo_ECEAction" style="rpc"/>
  <input>
    <soap:body use="encoded"
encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" namespace="urn:Catalogo_ECE"/>
  </input>
  <output>
    <soap:body use="encoded"
encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" namespace="urn:Catalogo_ECE"/>
  </output>
  <fault name="FaultName">
    <soap:fault name="EliminarFault" use="encoded"
encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" namespace="urn:Catalogo_ECE"/>
  </fault>
</operation>
<operation name="UltimoConsecutivo">
  <soap:operation soapAction="urn:Catalogo_ECEAction" style="rpc"/>
  <input>
    <soap:body use="encoded"
encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" namespace="urn:Catalogo_ECE"/>
  </input>
  <output>
    <soap:body use="encoded"
encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" namespace="urn:Catalogo_ECE"/>
  </output>
  <fault name="FaultName">
    <soap:fault name="UltimoConsecutivoFault" use="encoded"
encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" namespace="urn:Catalogo_ECE"/>
  </fault>
</operation>
<operation name="Listar">
  <soap:operation soapAction="urn:Catalogo_ECEAction" style="rpc"/>
```

```
<input>
  <soap:body use="encoded"
encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" namespace="urn:Catalogo_ECE"/>
</input>
<output>
  <soap:body use="encoded"
encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" namespace="urn:Catalogo_ECE"/>
</output>
<fault name="FaultName">
  <soap:fault name="ListarFault" use="encoded"
encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" namespace="urn:Catalogo_ECE"/>
</fault>
</operation>
</binding>
<service name="Catalogo_ECEservice">
  <documentation>Catalogo_ECE</documentation>
  <port name="Catalogo_ECEPort" binding="typens:Catalogo_ECEBinding">
    <soap:address
location="http://localhost/laboratorio_online/WebServices/WS_ECE/Laboratorio_Online_Server1.ph
p"/>
  </port>
</service>
</wsdl:definitions>
```

Anexo #3**Documento WSDL del servicio web LabOnlineservice.**

```

<!-- edited with XMLSPY v2004 rel. 2 U (http://www.xmlspy.com) by Crack (Crack) -->
<wsdl:definitions xmlns:typens="urn:LabOnline" xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"
xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/" xmlns="http://schemas.xmlsoap.org/wsdl/"
targetNamespace="urn:LabOnline" name="urn:LabOnline">
  <wsdl:types>
    <schema targetNamespace="urn:LabOnline"
xmlns="http://www.w3.org/2001/XMLSchema">
      <complexType name="array_PacienteLab">
        <sequence maxOccurs="unbounded">
          <element name="PacienteLab" type="typens:PacienteLabType"/>
        </sequence>
      </complexType>
      <complexType name="array_ResultadosLab">
        <sequence maxOccurs="unbounded">
          <element name="ResultadosLab"
type="typens:ResultadosLabType"/>
        </sequence>
      </complexType>
      <complexType name="criterioSolicitud">
        <complexContent>
          <extension base="typens:SolicitudLabType"/>
        </complexContent>
      </complexType>
      <complexType name="criterioPaciente">
        <complexContent>
          <extension base="typens:PacienteLabType"/>
        </complexContent>
      </complexType>

```

```
<complexType name="array_SolicitudLab">
  <sequence maxOccurs="unbounded">
    <element name="SolicitudLab" type="typens:SolicitudLabType"/>
  </sequence>
</complexType>
<complexType name="PacienteLabType">
  <sequence>
    <element name="DatosPersonales"
type="typens:DatosPersonalesType"/>
    <element name="DatosDemograficos"
type="typens:DatosDemograficosType"/>
    <element name="DatosSeguro" type="typens:DatosSeguroType"
nillable="true"/>
    <element name="DatosLaborales"
type="typens:DatosLaboralesType" nillable="true"/>
  </sequence>
</complexType>
<complexType name="DatosminPacienteType">
  <all>
    <element name="IdPaciente" type="string"/>
    <element name="Nombre" type="string"/>
    <element name="Sexo" type="byte"/>
    <element name="Edad" type="string" fixed="9"/>
    <element name="Peso" type="float"/>
    <element name="IdRaza" type="byte"/>
    <element name="Identificacion" type="string"/>
  </all>
</complexType>
<complexType name="DatosPacienteUnidadType">
  <all>
    <element name="Piso" type="string"/>
    <element name="Cama" type="string"/>
    <element name="Nu_Afiliacion" type="string"/>
  </all>
</complexType>
```

```
        <element name="Nu_Conultorio" type="string"/>
        <element name="Prox_Consulta" type="date"/>
    </all>
</complexType>
<complexType name="DatosMedicoType">
    <all>
        <element name="Matricula" type="string"/>
        <element name="IdRol" type="byte"/>
    </all>
</complexType>
<complexType name="DatosSolicitudType">
    <sequence>
        <element name="Folio" type="string"/>
        <element name="Fecha" type="date"/>
        <element name="Fecha_Cita" type="date"/>
        <element name="Observaciones" type="string"/>
        <element name="Id_Prioridad" type="byte"/>
        <element name="Fecha_Entrega_Resultados" type="date"/>
        <element name="Estado" type="byte"/>
        <element name="Id_Diagnostico_Presuntivo" type="string"/>
        <element name="Diagnostico_Presuntivo" type="string"/>
        <element name="Folio_Ref" type="string"/>
        <element name="DeclaracionMedica"
type="typens:DeclaracionMedicaType"/>
        <element name="array_Examenes"
type="typens:array_ExamenesType"/>
    </sequence>
</complexType>
<complexType name="DatosUnidadType">
    <all>
        <element name="Delegacion" type="string"/>
        <element name="Unidad" type="string"/>
        <element name="Servicio" type="string"/>
```

```
        </all>
    </complexType>
    <complexType name="ResultadosLabType">
        <sequence>
            <element name="DatosminPaciente"
type="typens:DatosminPacienteType"/>
            <element name="array_DatosResultados"
type="typens:array_DatosResultadosType"/>
        </sequence>
    </complexType>
    <complexType name="SolicitudLabType">
        <sequence>
            <element name="DatosMinPaciente"
type="typens:DatosminPacienteType"/>
            <element name="DatosUnidadTratante"
type="typens:DatosUnidadType"/>
            <element name="DatosUnidadProcedencia"
type="typens:DatosUnidadType"/>
            <element name="array_DatosSolicitud"
type="typens:array_DatosSolicitudType"/>
            <element name="DatosMedico" type="typens:DatosMedicoType"/>
            <element name="DatosPacienteUnidad"
type="typens:DatosPacienteUnidadType"/>
        </sequence>
    </complexType>
    <complexType name="array_DatosMinSolicitud">
        <sequence maxOccurs="unbounded">
            <element name="DatosMinSolicitud"
type="typens:DatosMinSolicitudType"/>
        </sequence>
    </complexType>
    <complexType name="Paginado">
        <sequence>
```

```
<element name="offset" type="integer"/>
<element name="cantidad" type="integer"/>
<element name="cantidad_real" type="integer"/>
<element name="total" type="integer"/>
</sequence>
</complexType>
<complexType name="DatosResultadosType">
  <sequence>
    <element name="Folio" type="string"/>
    <element name="Estudio" type="string"/>
    <element name="Componente" type="string"/>
    <element name="Fecha" type="date"/>
    <element name="Fecha_Resultado" type="date"/>
    <element name="Fecha_Validacion" type="date"/>
    <element name="Matricula_Med_Valida" type="string"/>
    <element name="Id_Equipo" type="string"/>
    <element name="Nu_Equipo" type="string"/>
    <element name="Observaciones" type="string"/>
    <element name="Resultado" type="string"/>
  </sequence>
</complexType>
<complexType name="AseguradoraType">
  <all>
    <element name="Nu_Cuenta" type="string"/>
    <element name="Empresa_Aseguradora" type="string"/>
    <element name="Cobertura" type="string"/>
    <element name="TipoSeguro" type="string"/>
  </all>
</complexType>
<complexType name="ExamenType">
  <sequence>
    <element name="Estudio" type="string"/>
    <element name="Importe" type="float"/>
  </sequence>
</complexType>
```

```
<element name="Unidad_Monetaria" type="string"/>
<element name="Nu_Toma" type="byte"/>
<element name="Observaciones_Cita" type="string"/>
</sequence>
</complexType>
<complexType name="DatosPersonalesType">
  <all>
    <element name="IdPersona" type="string" fixed="18"/>
    <element name="Identificacion" type="string"/>
    <element name="Nombre" type="string" fixed="50"/>
    <element name="ApellidoPaterno" type="string" fixed="50"/>
    <element name="ApellidoMaterno" type="string" fixed="50"/>
    <element name="Fecha_Nac" type="date"/>
    <element name="Edad" type="string" fixed="9"/>
    <element name="Sexo" type="byte"/>
    <element name="Provincia_Nac" type="string" fixed="2"/>
    <element name="Pais_Nac" type="byte"/>
    <element name="Grupo_Sanguineo" type="byte"/>
    <element name="IdRaza" type="byte"/>
  </all>
</complexType>
<complexType name="DatosDemograficosType">
  <all>
    <element name="Calle" type="string"/>
    <element name="Barrio" type="string"/>
    <element name="CP" type="string"/>
    <element name="Telefono" type="string"/>
    <element name="Nu_Int" type="string"/>
    <element name="Nu_Ext" type="string"/>
    <element name="Provincia" type="string"/>
    <element name="Municipio" type="string"/>
    <element name="Pais" type="byte"/>
  </all>
</complexType>
```

```
</complexType>
<complexType name="array_AseguradoraType">
  <sequence maxOccurs="unbounded">
    <element name="Aseguradora" type="typens:AseguradoraType"/>
  </sequence>
</complexType>
<complexType name="DatosSeguroType">
  <all>
    <element name="NSS" type="string"/>
    <element name="Calidad" type="byte"/>
    <element name="array_Aseguradora"
type="typens:array_AseguradoraType"/>
  </all>
</complexType>
<complexType name="DatosLaboralesType">
  <all>
    <element name="IdOcupacion" type="string"/>
    <element name="Ocupacion" type="string"/>
  </all>
</complexType>
<complexType name="array_DatosResultadosType">
  <sequence maxOccurs="unbounded">
    <element name="DatosResultados"
type="typens:DatosResultadosType"/>
  </sequence>
</complexType>
<complexType name="DeclaracionMedicaType">
  <sequence>
    <element name="Area_Salud" type="string" fixed="15"/>
    <element name="Direccion" type="string" fixed="60"/>
    <element name="Sintomatologia" type="string" fixed="150"/>
    <element name="IdRiesgo" type="int"/>
    <element name="Otros_Datos" type="string" fixed="500"/>
  </sequence>
</complexType>
```

```
        </sequence>
    </complexType>
    <complexType name="array_ExamenesType">
        <sequence maxOccurs="unbounded">
            <element name="Examen" type="typens:ExamenType"/>
        </sequence>
    </complexType>
    <complexType name="array_DatosSolicitudType">
        <sequence maxOccurs="unbounded">
            <element name="DatosSolicitud"
type="typens:DatosSolicitudType"/>
        </sequence>
    </complexType>
    <complexType name="DatosMinSolicitudType">
        <sequence>
            <element name="IdPaciente" type="string"/>
            <element name="IdFolio" type="string"/>
            <element name="Fecha" type="date"/>
            <element name="IdFolio_Ref" type="string"/>
        </sequence>
    </complexType>
</schema>
</wsdl:types>
<wsdl:message name="RegistroPacienteRequest">
    <wsdl:part name="listaPaciente" type="typens:array_PacienteLab"/>
</wsdl:message>
<wsdl:message name="RegistroPacienteResponse">
    <wsdl:part name="resultado" type="xsd:boolean"/>
</wsdl:message>
<wsdl:message name="RegistroPacienteFault">
    <wsdl:part name="codigo" type="xsd:int"/>
    <wsdl:part name="mensaje" type="xsd:string"/>
</wsdl:message>
```

```
<wsdl:message name="RegistroSolicitudRequest">
    <wsdl:part name="listaSolicitudLab" type="typens:array_SolicitudLab"/>
</wsdl:message>
<wsdl:message name="RegistroSolicitudResponse">
    <wsdl:part name="resultado" type="xsd:boolean"/>
</wsdl:message>
<wsdl:message name="RegistroResultadosRequest">
    <wsdl:part name="listaResultadosLab" type="typens:array_ResultadosLab"/>
</wsdl:message>
<wsdl:message name="RegistroResultadosResponse">
    <wsdl:part name="resultado" type="xsd:boolean"/>
</wsdl:message>
<wsdl:message name="RegistroResultadosFault">
    <wsdl:part name="codigo" type="xsd:int"/>
    <wsdl:part name="mensaje" type="xsd:string"/>
</wsdl:message>
<wsdl:message name="SolicitarResultadosRequest">
    <wsdl:part name="paginado" type="typens:Paginado"/>
    <wsdl:part name="criterioResultado" type="typens:ResultadosLabType"/>
    <wsdl:part name="unidad" type="typens:DatosUnidadType"/>
</wsdl:message>
<wsdl:message name="SolicitarResultadosResponse">
    <wsdl:part name="paginado" type="typens:Paginado"/>
    <wsdl:part name="listaResultadosLab" type="typens:array_ResultadosLab"/>
</wsdl:message>
<wsdl:message name="SolicitarResultadosFault">
    <wsdl:part name="codigo" type="xsd:int"/>
    <wsdl:part name="mensaje" type="xsd:string"/>
</wsdl:message>
<wsdl:message name="RegistroSolicitudFault">
    <wsdl:part name="codigo" type="xsd:int"/>
    <wsdl:part name="mensaje" type="xsd:string"/>
</wsdl:message>
```

```
<wsdl:message name="SolicitarEstudiosRequest">
  <wsdl:part name="paginado" type="typens:Paginado"/>
  <wsdl:part name="criterio" type="typens:criterioSolicitud"/>
</wsdl:message>
<wsdl:message name="SolicitarEstudiosResponse">
  <wsdl:part name="paginado" type="typens:Paginado"/>
  <wsdl:part name="listaSolicitudLab" type="typens:array_SolicitudLab"/>
</wsdl:message>
<wsdl:message name="SolicitarEstudiosFault">
  <wsdl:part name="codigo" type="xsd:int"/>
  <wsdl:part name="mensaje" type="xsd:string"/>
</wsdl:message>
<wsdl:message name="BuscarRegistroPacienteRequest">
  <wsdl:part name="paginado" type="typens:Paginado"/>
  <wsdl:part name="criterio" type="typens:criterioPaciente"/>
</wsdl:message>
<wsdl:message name="BuscarRegistroPacienteResponse">
  <wsdl:part name="paginado" type="typens:Paginado"/>
  <wsdl:part name="resultado" type="typens:array_PacienteLab"/>
</wsdl:message>
<wsdl:message name="BuscarRegistroPacienteFault">
  <wsdl:part name="codigo" type="xsd:int"/>
  <wsdl:part name="mensaje" type="xsd:string"/>
</wsdl:message>
<wsdl:message name="ConfirmarReciboSolicitudRequest">
  <wsdl:part name="listaSolicitud" type="typens:array_DatosMinSolicitud"/>
</wsdl:message>
<wsdl:message name="ConfirmarReciboSolicitudResponse">
  <wsdl:part name="resultado" type="xsd:boolean"/>
</wsdl:message>
<wsdl:message name="ConfirmarReciboSolicitudFault">
  <wsdl:part name="codigo" type="xsd:int"/>
  <wsdl:part name="mensaje" type="xsd:string"/>
</wsdl:message>
```

```
</wsdl:message>
<wsdl:message name="FolioConsecutivoSolicitudRequest">
  <wsdl:part name="unidad" type="typens:DatosUnidadType"/>
</wsdl:message>
<wsdl:message name="FolioConsecutivoSolicitudResponse">
  <wsdl:part name="folio" type="xsd:string"/>
</wsdl:message>
<wsdl:message name="FolioConsecutivoSolicitudFault">
  <wsdl:part name="codigo" type="xsd:int"/>
  <wsdl:part name="mensaje" type="xsd:string"/>
</wsdl:message>
<wsdl:message name="BuscarPendienteResultadosRequest">
  <wsdl:part name="paginado" type="typens:Paginado"/>
  <wsdl:part name="criterio" type="typens:criterioSolicitud"/>
</wsdl:message>
<wsdl:message name="BuscarPendienteResultadosResponse">
  <wsdl:part name="paginado" type="typens:Paginado"/>
  <wsdl:part name="listaSolicitudLab" type="typens:array_SolicitudLab"/>
</wsdl:message>
<wsdl:message name="BuscarPendienteResultadosFault">
  <wsdl:part name="codigo" type="xsd:int"/>
  <wsdl:part name="mensaje" type="xsd:string"/>
</wsdl:message>
<portType name="LabOnlinePort">
  <operation name="RegistroPaciente">
    <input message="typens:RegistroPacienteRequest"/>
    <output message="typens:RegistroPacienteResponse"/>
    <wsdl:fault name="FaultName" message="typens:RegistroPacienteFault"/>
  </operation>
  <operation name="BuscarRegistroPaciente">
    <input message="typens:BuscarRegistroPacienteRequest"/>
    <output message="typens:BuscarRegistroPacienteResponse"/>
  </operation>
</portType>
</wsdl:binding>
</wsdl:service>
</wsdl:definitions>
```

```
        <wsdl:fault name="FaultName"
message="typens:BuscarRegistroPacienteFault"/>
    </operation>
    <operation name="RegistroSolicitud">
        <input message="typens:RegistroSolicitudRequest"/>
        <output message="typens:RegistroSolicitudResponse"/>
        <wsdl:fault name="FaultName" message="typens:RegistroSolicitudFault"/>
    </operation>
    <operation name="SolicitarEstudios">
        <input message="typens:SolicitarEstudiosRequest"/>
        <output message="typens:SolicitarEstudiosResponse"/>
        <wsdl:fault name="FaultName" message="typens:SolicitarEstudiosFault"/>
    </operation>
    <operation name="RegistroResultados">
        <input message="typens:RegistroResultadosRequest"/>
        <output message="typens:RegistroResultadosResponse"/>
        <wsdl:fault name="FaultName" message="typens:RegistroResultadosFault"/>
    </operation>
    <operation name="SolicitarResultados">
        <input message="typens:SolicitarResultadosRequest"/>
        <output message="typens:SolicitarResultadosResponse"/>
        <wsdl:fault name="FaultName" message="typens:SolicitarResultadosFault"/>
    </operation>
    <operation name="ConfirmarReciboSolicitud">
        <wsdl:input message="typens:ConfirmarReciboSolicitudRequest"/>
        <wsdl:output message="typens:ConfirmarReciboSolicitudResponse"/>
        <wsdl:fault name="FaultName"
message="typens:ConfirmarReciboSolicitudFault"/>
    </operation>
    <operation name="FolioConsecutivoSolicitud">
        <wsdl:input message="typens:FolioConsecutivoSolicitudRequest"/>
        <wsdl:output message="typens:FolioConsecutivoSolicitudResponse"/>
```

```
        <wsdl:fault name="FaultName"
message="typens:FolioConsecutivoSolicitudFault"/>
    </operation>
    <operation name="BuscarPendienteResultados">
        <wsdl:input message="typens:BuscarPendienteResultadosRequest"/>
        <wsdl:output message="typens:BuscarPendienteResultadosResponse"/>
        <wsdl:fault name="FaultName"
message="typens:BuscarPendienteResultadosFault"/>
    </operation>
</portType>
<binding name="LabOnlineBinding" type="typens:LabOnlinePort">
    <soap:binding style="rpc" transport="http://schemas.xmlsoap.org/soap/http"/>
    <operation name="RegistroPaciente">
        <soap:operation soapAction="urn:LabOnlineAction" style="rpc"/>
        <input>
            <soap:body use="encoded"
encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" namespace="urn:LabOnline"/>
        </input>
        <output>
            <soap:body use="encoded"
encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" namespace="urn:LabOnline"/>
        </output>
        <fault name="FaultName">
            <soap:fault name="RegistroPacienteFault" use="encoded"
encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" namespace="urn:LabOnline"/>
        </fault>
    </operation>
    <operation name="BuscarRegistroPaciente">
        <soap:operation soapAction="urn:LabOnlineAction" style="rpc"/>
        <input>
            <soap:body use="encoded"
encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" namespace="urn:LabOnline"/>
        </input>
```

```
<output>
  <soap:body use="encoded"
encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" namespace="urn:LabOnline"/>
</output>
<fault name="FaultName">
  <soap:fault name="BuscarRegistroPacienteFault" use="encoded"
encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" namespace="urn:LabOnline"/>
</fault>
</operation>
<operation name="RegistroSolicitud">
  <soap:operation soapAction="urn:LabOnlineAction" style="rpc"/>
  <input>
    <soap:body use="encoded"
encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" namespace="urn:LabOnline"/>
  </input>
  <output>
    <soap:body use="encoded"
encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" namespace="urn:LabOnline"/>
  </output>
  <fault name="FaultName">
    <soap:fault name="RegistroSolicitudFault" use="encoded"
encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" namespace="urn:LabOnline"/>
  </fault>
</operation>
<operation name="SolicitarEstudios">
  <soap:operation soapAction="urn:LabOnline" style="rpc"/>
  <input>
    <soap:body use="encoded"
encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" namespace="urn:LabOnline"/>
  </input>
  <output>
    <soap:body use="encoded"
encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" namespace="urn:LabOnline"/>
```

```
</output>
  <fault name="FaultName">
    <soap:fault name="SolicitarEstudiosFault" use="encoded"
encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" namespace="urn:LabOnline"/>
  </fault>
</operation>
<operation name="RegistroResultados">
  <soap:operation soapAction="urn:LabOnline" style="rpc"/>
  <input>
    <soap:body use="encoded"
encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" namespace="urn:LabOnline"/>
  </input>
  <output>
    <soap:body use="encoded"
encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" namespace="urn:LabOnline"/>
  </output>
  <fault name="FaultName">
    <soap:fault name="RegistroResultadosFault" use="encoded"
encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" namespace="urn:LabOnline"/>
  </fault>
</operation>
<operation name="SolicitarResultados">
  <soap:operation soapAction="urn:LabOnline" style="rpc"/>
  <input>
    <soap:body use="encoded"
encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" namespace="urn:LabOnline"/>
  </input>
  <output>
    <soap:body use="encoded"
encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" namespace="urn:LabOnline"/>
  </output>
  <fault name="FaultName">
```

```
        <soap:fault name="SolicitarResultadosFault" use="encoded"
encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" namespace="urn:LabOnline"/>
        </fault>
    </operation>
    <operation name="ConfirmarReciboSolicitud">
        <soap:operation soapAction="urn:LabOnline" style="rpc"/>
        <input>
            <soap:body use="encoded"
encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" namespace="urn:LabOnline"/>
        </input>
        <output>
            <soap:body use="encoded"
encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" namespace="urn:LabOnline"/>
        </output>
        <fault name="FaultName">
            <soap:fault name="ConfirmarReciboSolicitudFault" use="encoded"
encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" namespace="urn:LabOnline"/>
        </fault>
    </operation>
    <operation name="FolioConsecutivoSolicitud">
        <soap:operation soapAction="urn:LabOnline" style="rpc"/>
        <input>
            <soap:body use="encoded"
encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" namespace="urn:LabOnline"/>
        </input>
        <output>
            <soap:body use="encoded"
encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" namespace="urn:LabOnline"/>
        </output>
        <fault name="FaultName">
            <soap:fault name="FolioConsecutivoSolicitudFault" use="encoded"
encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" namespace="urn:LabOnline"/>
        </fault>
```

```
</operation>
  <operation name="BuscarPendienteResultados">
    <soap:operation soapAction="urn:LabOnline" style="rpc"/>
    <input>
      <soap:body use="encoded"
encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" namespace="urn:LabOnline"/>
    </input>
    <output>
      <soap:body use="encoded"
encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" namespace="urn:LabOnline"/>
    </output>
    <fault name="FaultName">
      <soap:fault name="BuscarPendienteResultadosFault" use="encoded"
encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" namespace="urn:LabOnline"/>
    </fault>
  </operation>
</binding>
<service name="LabOnlineservice">
  <documentation>LabOnline</documentation>
  <port name="LabOnlinePort" binding="typens:LabOnlineBinding">
    <soap:address
location="http://localhost/laboratorio_online/WebServices/WS_ECE/Laboratorio_Online_Server.php
"/>
  </port>
</service>
</wsdl:definitions>
```

GLOSARIO DE TÉRMINOS

Métrica de reusabilidad

Es una medida cuantitativa del grado de reusabilidad de un componente.

Reutilización de software

Es el proceso de crear sistemas de software a partir de software existente, en lugar de desarrollarlo desde el comienzo.

Item

Unidad de almacenamiento.

Herencia

En programación orientada a objetos, la capacidad de crear nuevas clases (o interfaces) que contienen todos los métodos y propiedades de otra clase (o interfaz), más métodos adicionales. Si una clase (o interfaz).

Interfaz

Declaración de un conjunto de métodos sin información sobre su implementación.

HTML

El Lenguaje de Etiquetas de Hipertexto (HyperText Markup Language) (HTML) es un lenguaje simple de etiquetas usado para crear documentos de hipertexto que son portables de una plataforma a otra.

Interoperabilidad

Habilidad que tiene un sistema o producto para trabajar con otros sistemas o productos sin un esfuerzo especial por parte del cliente.

Criterio

Es una condición/regla que permite realizar una elección, lo que implica que sobre un criterio se pueda basar una decisión o un juicio de valor.

Legibilidad

La cualidad que posee un texto para ser fácilmente leído, es decir, fácilmente percibido en cuanto a su forma, presentación y disposición.

Comprensibilidad

El grado o nivel de comprensibilidad de un determinado texto. En general, como resultado de la evaluación de un texto se obtiene el nivel de instrucción mínimo necesario por parte de un usuario para comprenderlo.

Consistencia

Propiedad de lo que es duradero, estable.

Autodescriptividad

Medida en que se hace comprensible un servicio.