

## Proceso dirigido por objetivos para análisis de dominio bajo estándares de calidad<sup>1</sup>

*Francisca Losavio*<sup>2</sup>  
*Alfredo Matteo*  
*Irma Pacilli*

### Resumen

Una de las preocupaciones actuales de la Ingeniería de Software, es reducir la brecha entre la ingeniería de requisitos y la ingeniería del sistema de software; el creciente interés en la disciplina denominada *Ingeniería del Dominio* trata de llenar esta brecha. En particular, este trabajo se enmarca en el contexto de la identificación temprana de requisitos no funcionales (RNF). El enfoque propuesto por Chung y otros, integra requisitos funcionales (RF) y RNF en el modelo de casos de uso y llega a una configuración arquitectónica inicial utilizando el enfoque dirigido por objetivos de Yu y otros. Nuestro aporte consiste en incorporar al enfoque de Chung, un primer paso de análisis del dominio basado en los estándares de calidad ISO/IEC 9126-1, para la especificación temprana de los RNF, mediante un modelo de calidad que representa una vista de calidad del conocimiento del dominio. Este nuevo paso de análisis permite justificar con precisión los requisitos globales y los límites del sistema, los cuales no son justificados por Chung, y reutilizar este conocimiento sobre los objetivos de calidad de estilos arquitectónicos y funcionalidades principales, para la obtención de la arquitectura inicial de la aplicación. Nuestra contribución principal y resultado es el paso de análisis del dominio como extensión al proceso de Chung, el cual puede ser aplicado en el contexto de métodos de diseño de líneas de producto y de desarrollo de software centrados en la arquitectura, ofreciendo también un lenguaje unificado sobre la calidad del producto software, del cual se carece en general.

**Palabras clave:** ingeniería de software, análisis del dominio, diseño dirigido por objetivos, modelo de calidad, estándares de calidad, ISO/IEC 9126-1, RNF

Recibido: 21-01-09 Aceptado: 23-09-09

<sup>1</sup> Este trabajo es parcialmente financiado por el Consejo de Desarrollo Científico y Humanístico (CDCH) de la Universidad Central de Venezuela, proyecto ADIRE, No. PG-03-7310-2008/1

<sup>2</sup> Autor de correspondencia. Laboratorio MoST, Centro ISYS, Escuela de Computación, Facultad de Ciencias, Universidad Central de Venezuela; Caracas, Venezuela

Correo electrónico: francisosavio@gmail.com, almatteo@cantv.net, pacilli\_irma@hotmail.com.

# Goal-oriented Process for Domain Analysis Using Quality Standards

## Abstract

One of the present concerns of Software Engineering is to reduce the gap between the stages of requirements engineering and software system engineering; the growing interest in the discipline of *Domain Engineering* is justly to try to fill in this gap. This work is framed in the context of the early identification of non functional requirements (NFR). Functional requirements (FR) and NFR are integrated into the use case model, according to the approach of Chung et al., where an initial architectural configuration is achieved according to the goal-oriented approach of Yu et al. Our main contribution consists in adding to the Chung et al. a domain analysis step based on the ISO/IEC 9126-1 quality standards, for the early specification of NFR, using a quality model representing a quality view of the domain knowledge. This domain analysis allows a precise justification of the system's global requirements and boundaries, which are not at all justified by Chung, reusing this knowledge on the quality goals of architectural styles and main functionality to obtain the initial architecture for the application. The main contribution and result is this extension step of domain analysis to the Chung et al. process; it can be applied in the context of software product lines design methods and in early stages of architecture centric software development methods. A unified language of software product quality which is generally missing is also provided by our approach.

**Key words:** Software Engineering, Domain Analysis, Goal-oriented Design, Quality Model, Quality Standards, ISO/IEC 9126-1, NFR

## Introducción

Para la Ingeniería de Software, la calidad de un producto debe ser considerada en etapas tempranas del proceso de desarrollo. En este contexto, la noción de *calidad de un producto* puede definirse como el conjunto de características o propiedades deseadas que deben estar presentes en el producto, considerando el punto de vista del usuario y del producto en si (ISO/IEC 9126-1, 2001). Por lo tanto, todo proyecto de software debe considerar satisfacer estas características deseadas y una manera de validar gran parte de ellas

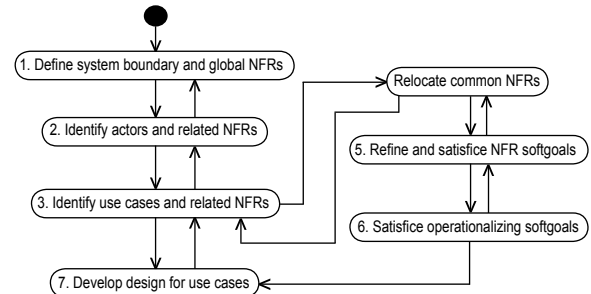
es considerar una *arquitectura documentada* del sistema como elemento central, en vista de que ésta debe responder a los requisitos exigidos por el sistema, tanto funcionales porque algunos componentes son introducidos para responder a estos requisitos, como no funcionales porque otros componentes son introducidos para responder a requisitos de calidad precisos. Esto sólo se puede lograr siguiendo procesos que permitan verificar la satisfacción de los requisitos iniciales y reutilizar soluciones ya probadas. Una arquitectura documentada se refiere a que cada decisión arquitectónica está justificada sobre las bases de haber

verificado que se cumplen los requisitos de calidad exigidos. Por *requisito de calidad* nos referimos a las propiedades que caracterizan una solución arquitectónica, por ejemplo interoperabilidad de los componentes. Los requisitos no funcionales a los cuales denotaremos RNF, tradicionalmente se han asociado a los requisitos de calidad pero no se ha desarrollado un marco en el cual se puedan representar e integrar fácilmente al proceso de diseño de las aplicaciones tal y como ocurre con los requisitos funcionales, a los cuales denotaremos RF, que se representan comúnmente a través del modelo de casos de uso de Booch, Rumbaugh & Jacobson (1999).

#### • RF, RNF: el proceso de Chung y Supakkul (2004)

Hay enfoques (Lamsweerde, 2001; Yu y Mylopoulos, 1998) en el proceso de desarrollo que consideran relacionar los RF con los requisitos de calidad u objetivos de calidad a los cuales éstos responden. Existen propuestas donde se han logrado integrar RF y RNF en un modelo de casos de uso extendido, como es el caso de Moreira, Brito y Araújo (2002); Sousa, Soares, Borba y Castro (2004) y Chung y Supakkul. (2004); pero sin llegar a ser un estándar específico y comúnmente aceptado. En la práctica, dentro del contexto de un desarrollo basado en aspectos tempranos ó “early aspects” (Brito y Moreira, 2004; Sousa, Soares, Borba y Castro, 2004), el enfoque de Chung resulta prometedor y parece ser de fácil aplicación; es claro que esta afirmación se podrá comprobar en el curso de investigaciones futuras en el área. El **Gráfico 1** muestra el diagrama de actividades de este proceso.

**Gráfico 1**  
**Diagrama de Actividad describiendo el proceso de Integración.**  
Chung y Supakkul (2004)



#### • Objetivo y enfoque del trabajo

En este trabajo se extiende un proceso propuesto por Chung y Supakkul (2004), para integrar RF y RNF en el modelo de casos de uso y bajo un enfoque orientado a objetivo ó “goal-oriented approach”; Chung utiliza para esta integración el framework NFR ó “Non Functional Requirements” definido por Chung, Nixon, Yu y Mylopoulos (2000).

Nuestra proposición especifica los requisitos de calidad asociados a RF y RNF de acuerdo al estándar ISO/IEC 9126-1, que considera la calidad interna, externa y en uso del producto de software, tomando en consideración la importancia de seguir estándares que permitan asegurar un lenguaje común para el entendimiento de todos los involucrados en el proyecto de software. En particular, sólo se tomará en cuenta el modelo de

calidad interna/externa, ya que se trata del producto de software durante su proceso de diseño y no del software como producto final en uso. En nuestro enfoque, la extensión al proceso de Chung consiste en incluir como etapa inicial, el análisis del dominio para la reutilización del conocimiento sobre la arquitectura y la identificación de restricciones o propiedades globales sobre las familias de aplicaciones del dominio. Un *dominio*, según Bernard (1992), es el conjunto mínimo de propiedades que definen una familia de problemas para los cuales se requieren soluciones computacionales. El resultado de este análisis es el punto de partida para la justificación de los requisitos globales de la aplicación o sistema a ser construido; Chung no justifica la obtención de estos requisitos globales. El proceso de Chung es entonces complementado por nuestro enfoque con esta etapa inicial y con el uso de estándares (ISO/IEC 9126-1, 2001; Losavio F., Chirinos L., Matteo A. y Ramdane-Cherif A., 2004), para la especificación de los RNF. El paso inicial de análisis del dominio con estándares de calidad para determinar los requisitos globales del sistema, constituye el aporte principal de este trabajo.

## Estructura del trabajo

Además de esta introducción y de las conclusiones, el presente documento contiene una segunda sección donde se reseñan los estándares de calidad y la elección del modelo ISO/IEC 9126-1. En la tercera sección, se presenta el enfoque orientado a objetivos. En la cuarta sección se describe el framework NFR de análisis y diseño orientado a objetivos propuesto en el proceso de Chung y

Supakkul (2004), extendido con el análisis del dominio como etapa inicial. Finalmente, en la quinta sección se ilustra la aplicación de la propuesta con un caso de estudio específico al dominio de las aplicaciones basadas en Servicios Web del World Wide Web Consortium (2004).

## Estándares de calidad del producto de software

Normalmente las características de calidad inherentes al producto a construir, son directamente definidas como RNF, pero es bueno señalar que algunas de las funcionalidades del sistema llevan implícitas la aplicación o cumplimiento de un requisito de calidad, que corresponde a un requisito no funcional (funcionalidad implícita), como por ejemplo la seguridad exigida por la funcionalidad *control de acceso*. Existen varios modelos de calidad en la literatura como son los propuestos por Dromey (1994), ISO/IEC 9126-1 (2001) y GQM (Berghout E., Solingen R., 1999) entre otros, para especificar la calidad de productos software; en este documento utilizaremos el estándar ISO/IEC 9126-1, representado en el **Cuadro 1**, donde puede notarse la representación jerárquica de cada una de las seis características principales de calidad interna y externa presentadas. El modelo puede ser refinado para incluir nuevas sub-sub características. Preferimos este estándar por su amplia difusión y aceptación en la comunidad internacional. Cabe resaltar que el comité JTC 1/SC7 WG6, encargado de la redacción de los estándares ISO/IEC sobre la calidad del producto de software, debe decidir la aprobación oficial del nuevo estándar ISO/IEC 25010 después de un proceso

complejo de votación que sustituiría el ISO/IEC 9126-1; esta votación parece estar prevista para julio 2009, pero la aprobación definitiva puede tardar aún más. El nuevo estándar aportaría modificaciones menores al actual estándar ISO/IEC 9126-1 básicamente en referencia a la característica *funcionalidad*, respecto a sus sub-características *seguridad* e *interoperabilidad* que pasarían a ser características de alto nivel, llevando de 6 a 8

las características del ISO/IEC 9126-1. Pensamos utilizar el estándar 25010 en el desarrollo de los trabajos futuros en esta línea de investigación en cuanto esté oficialmente aprobado. Por los momentos, como el estándar 9126-1 es el vigente y representa el consenso de la comunidad científica en un dominio particular, nos regimos con la última versión oficial de este estándar, para proporcionar así una mayor confiabilidad a la investigación.

**Cuadro 1**  
**Características y sub-características**  
**interna/externa del modelo de Calidad ISO/IEC 9126-1**

Funcionalidad	Confiabilidad	Usabilidad	Eficiencia	Facilidad de Mantenimiento (Mantenibilidad)	Portabilidad
Adecuación	Madurez	Entendimiento	Comportamiento	Facilidad de ser	Adaptabilidad
Precisión	Tolerancia a	Aprendizaje	en tiempo	analizado	Facilidad de ser
Interoperabi-	fallas	Operabilidad	Uso de recursos	Facilidadde	instalado.
lidad	Facilidad de	Atractivo	Conformidad	cambio	Co-existencia
Seguridad	recuperación	Conformidad		Estabilidad	Fácil. de ser
Conformidad	Conformidad			Facilidad de ser	reemplazado
				probado	Conformidad
				Conformidad	

Establecer el modelo de calidad para el dominio de la aplicación ayuda a definir los *requisitos no funcionales globales* del sistema, así como algunos requisitos de calidad para las funcionalidades del usuario comunes a una familia de aplicaciones del dominio y representa una vista de calidad del conocimiento del dominio.

### Enfoque orientado a objetivos

El enfoque orientado a objetivos de software, denominados en la literatura *objetivos blandos* del inglés “softgoals” (Dardenne y Lamsweerde, 1993; Lamsweerde, 2001; Yu y Mylopoulos, 1998), se traduce en la satisfacción de objetivos no fun-

cionales, a través de las contribuciones positivas de las *operacionalizaciones*, que son actividades de bajo nivel o mecanismos introducidos para la satisfacción de un objetivo requerido por los mismos. Es decir, los softgoals, (Chung, Nixon, Yu, y Mylopoulos, 2000), son objetivos difíciles de expresar ya que no son funcionalidades requeridas directamente por la interacción del usuario con el sistema; un ejemplo es la seguridad, ilustrada en el **Gráfico 2**. Estos objetivos se descomponen y refinan en una estructura de árbol –llamado el *Grafo de Interdependencia de Softgoals*, denominado en la literatura “Softgoal Interdependency Graph” (SIG) (Chung, Nixon, Yu y Mylopoulos, 2000), el cual es un grafo que recoge el criterio del desarrollador sobre estos objetivos y que muestra la interdependencia entre ellos– dividiéndolos en *subobjetivos* hasta alcanzar la operacionalización, que indica un componente de software que se introduce para satisfacer el objetivo, en un enfoque descendente o “topdown”. Luego, se comienza desde el componente que operacionaliza el objetivo, hacia arriba ó “bottom up”, evaluando las *contribuciones* de cada rama del árbol; aquellas que tengan mayor peso positivo serán las que se deben ir logrando para satisfacer el objetivo principal de más alto nivel. En el **Gráfico 2** tomado directamente de Sousa, Soares, Borba y Castro (2004), se observa un ejemplo de un SIG donde se descompone el RNF *seguridad* ó “security”, indicando que se deben satisfacer los RNFs *integridad* ó “integrity”, *confidencialidad* ó “confidentiality” y *disponibilidad* ó “availability” para poder satisfacerlo.

## Framework NFR para el Análisis y Diseño Orientado a Objetivos

En el enfoque presentado en Chung y Supakkul (2004) se propone utilizar el framework NFR, definido por Yu y Mylopoulos (1998) junto con el enfoque basado en escenarios o casos de uso, sobre las bases de dos premisas principales:

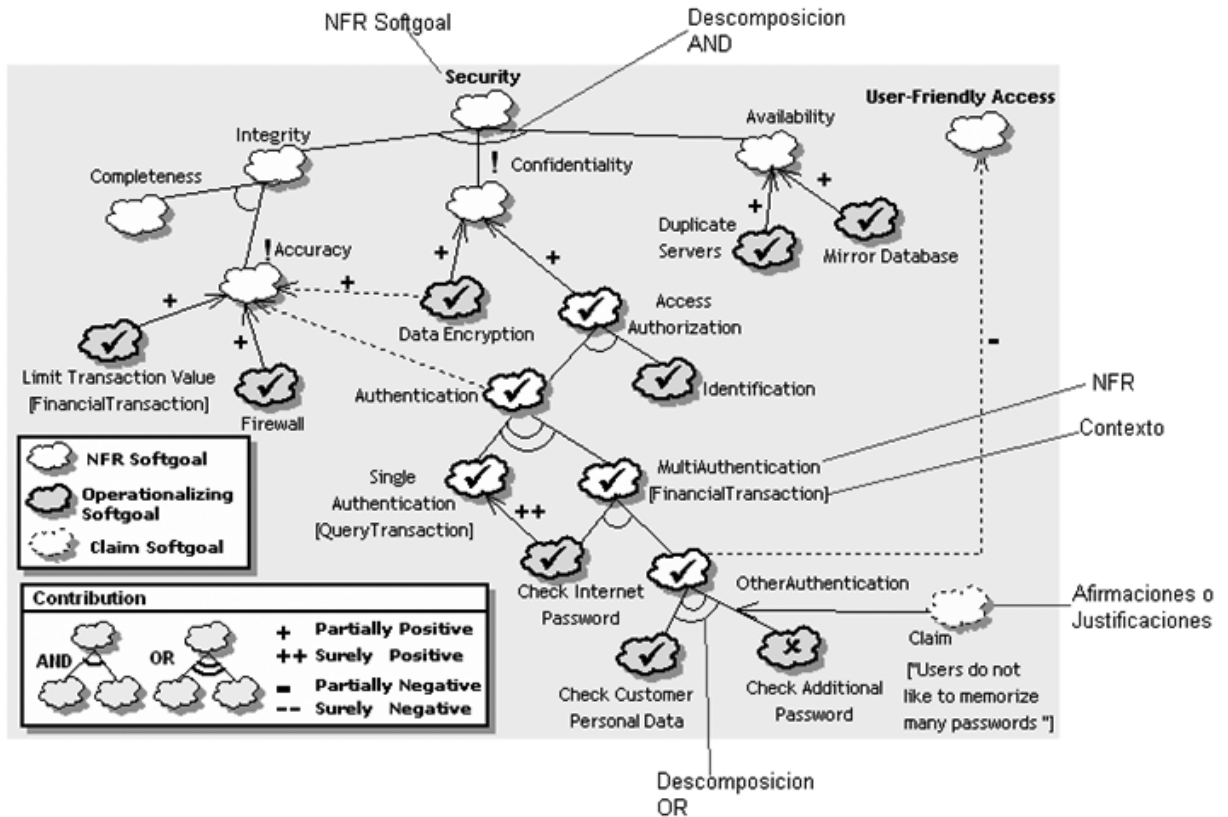
- 1) integrar los RNFs a los diagramas de casos de uso, que expresan los requisitos funcionales principales respecto a los usuarios a través de los denominados *puntos de asociación* para clasificar los RNF y
- 2) establecer el alcance de reglas de propagación para relacionar las asociaciones propuestas con el modelo de casos de uso.

Los puntos de asociación se utilizan para realizar una taxonomía de los RNF y las relaciones de generalización/especialización son utilizadas para “propagar” el RNF, es decir incluirlo en el diagrama de casos de uso como una especialización del caso de uso base. Para los casos de uso por inclusión, denotados como <<include>> en el Lenguaje de Modelación Unificado, abreviado “UML” del inglés “Unified Modeling Language (OMG, del inglés Object Management Group; Booch, Rumbaugh & Jacobson, 1999), el RNF es propagado también al caso de uso incluido. Esto es debido al hecho de que UML no dispone aún de elementos de modelación aceptados para los RNF. Sin embargo, el enfoque de Brito y Moreira (2004) propone considerar los RNF como casos de uso de inclusión ó <<include>>, lo cual nos parece atractivo, en el sentido de que el RNF se expresa como

un mecanismo que siempre deberá estar presente en la ejecución del caso de uso. Además, a partir de *tablas de composición* (Brito y Moreira, 2004)

es más fácil identificar las posibles incumbencias transversales en un contexto de diseño temprano de aspectos.

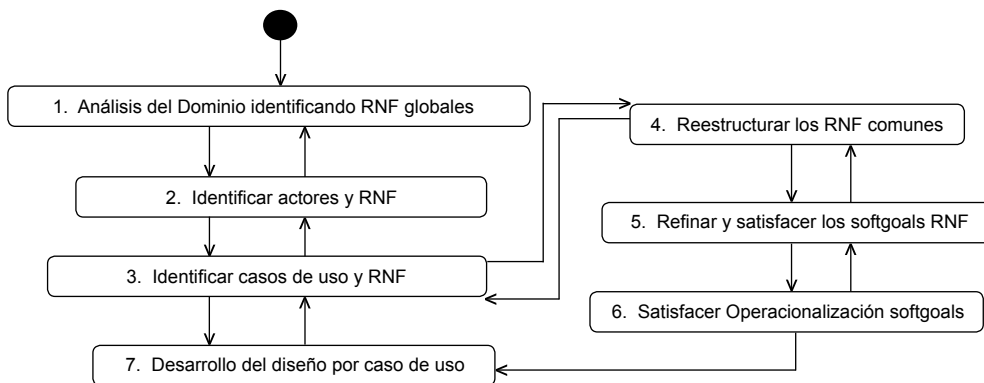
**Gráfico 2**  
**Grafo de Interdependencia del RNF**  
**seguridad ó “security”.**  
 Sousa et al. (2004)



**Puntos de Asociación:** se proponen cuatro asociaciones de los RNF relacionados con: el proceso de desarrollo del sistema denominados *globales*, entidades externas ó *actor*, de acceso, comunicación o intercambio de información denominados asociación *Actor-Caso de uso* y finalmente los requisitos funcionales clásicos ó *caso de uso*; los tipos de RNF mencionados se aplican respectivamente en cuatro puntos del diagrama

de casos de uso, específicamente en: la frontera del sistema, en el actor, en el caso de uso y en el enlace entre actor y el caso de uso. Estas asociaciones permiten una primera clasificación de requisitos no funcionales, a un alto nivel. Se plantea un diagrama de actividades donde se esboza un proceso iterativo e incremental a través del cual se integran los RF y RNFs, bajo las premisas planteadas, como se muestra en el **Gráfico 3**.

**Gráfico 3**  
**Diagrama de Actividad del Proceso de Integración de RF y RNF, con análisis del dominio**



Un aporte específico del presente trabajo es integrar, en la primera fase de ese proceso denominada en Chung y Supakkul (2004) “*definir límites del sistema y requerimientos globales*” que se mostró en el **Gráfico 1**, la actividad 1 en el diagrama del **Gráfico 3**, un análisis del dominio de la aplicación, utilizando el estándar de calidad ISO/

IEC 9126-1 para la especificación de los requisitos de calidad asociados a los RNF, de tal manera de proporcionar una primera especificación y justificación razonada ó documentación, de los requisitos que afectan al dominio y a cualquier familia de aplicaciones que se encuentre dentro de éste. Nuestro enfoque se centra en especificar las pro-



iedades de calidad utilizando el modelo estándar ISO/IEC 9126-1 para unificar la terminología respecto a los requisitos de calidad.

*Análisis del dominio* en este contexto implica identificar aquellos requisitos funcionales o no, relativos a las familias de aplicaciones del dominio del problema a resolver. Para esta etapa, Losavio, Matteo y Rahamut (2008), proponen un proceso para una caracterización del dominio de aplicaciones basadas en servicios Web, a los cuales denominaremos WS, que será aplicado en la siguiente sección para el caso de estudio; este proceso será generalizado aquí para cualquier dominio y constituye el principal aporte de este trabajo.

### **Proceso de análisis del dominio del problema**

Entrada: la *descripción textual del problema*, para el cual se requiere como solución un sistema o aplicación de software; *taxonomía de familias de aplicaciones del dominio y estilos o soluciones arquitecturales de alto nivel* para estas familias.

- a) Definir la funcionalidad del dominio: listar las funcionalidades principales por cada tipo de familia
- b) Definir el modelo de calidad, utilizando el estándar ISO/IEC 9126-1 (nótese que otro estándar podría ser utilizado).
  - b.1) se especifica la *calidad arquitectural del dominio*, utilizando los objetivos de calidad críticos de una solución arquitectural genérica o estilo para cada familia de ese dominio

- b.2) se especifica la *calidad funcional del dominio*, donde por cada familia se identifican las funcionalidades propias a la familia y a cada funcionalidad se asocian los requisitos de calidad, como objetivos que deben cumplirse para lograr las funcionalidades. Este paso se repite para cada familia del dominio

Salida: *modelo de calidad* para cada familia del dominio

### **Caso de estudio: fijación de precios para el suministro de servicios en líneas aéreas**

#### ***Paso 1. Análisis del dominio***

Este paso es una propuesta original del presente trabajo, constituyendo uno de sus principales aportes y no es considerado en el proceso de Chung.

*Entrada:*

–*Descripción del problema*

El problema del caso de estudio es propuesto por Chung & Supakkul (2004).

El Sistema de Fijación de Precios permitirá a las aerolíneas colaborar con sus proveedores a través de internet para establecer los precios de los ítems de servicio que se ofrecen en los vuelos, tales como: leche, bebidas, comidas y artículos de limpieza. La aerolínea se encarga de mantener actualizado el sistema con los servicios que requiere. Cuando se crea o actualiza un servicio se le envía esa información a los proveedores, vía internet, para que ellos envíen su propuesta de precios. Si la propuesta es aceptada o rechazada, se le informa

al proveedor y éste puede reenviar su propuesta hasta llegar a un acuerdo con respecto al precio del servicio.

- *Taxonomía de familias:*

Aplicación basada en WS de tipo transaccional (contratación de servicios vía Internet); una transacción se define como un conjunto de actividades agrupadas en una unidad; una transacción es cumplida como un todo, si una actividad falla, toda la transacción falla, World Wide Web Consortium (2004).

- *Estilo arquitectural para la familia* de aplicaciones basadas en WS transaccionales: según Carnegie Mellon-Software Engineering Institute (2003), Arquitecturas Orientadas a Servicios denotadas por “SOA” del inglés “Service Oriented Architecture”; estilo capas, siguiendo un modelo de comunicación cliente/servidor (Losavio F., Matteo A. y Rahamut R., 2008).

a) *Identificar funcionalidades propias a la familia*

Para el tipo de aplicaciones basadas en WS transaccionales, se identifican las siguientes funcionalidades básicas: intercambio de datos, control de acceso (Losavio F., Matteo A. y Rahamut R., 2008).

b) *Definir modelo de calidad*

b.1) Especificar *la calidad arquitectural para la familia de aplicaciones basadas en WS transaccionales*: la arquitectura SOA para estas aplicaciones debe responder a los objetivos de calidad. Losavio F., Matteo A. y Rahamut R.(2008), proponen: **interoperabilidad**, **confiabilidad** (*disponibilidad*), **mantenibilidad** (*extensión*),

**portabilidad** (*escalabilidad*). Nótese que las sub-características se han colocado entre paréntesis a continuación de la característica principal de calidad. Debe observarse que SOA es parte del conocimiento del dominio; pueden existir otras soluciones arquitectónicas para ese dominio en particular. La elección de una solución particular debe obedecer a un proceso de selección basado en la experticia, es decir es una solución probada y aparece en un catálogo y en la satisfacción de sus requisitos de calidad, respecto a los requisitos de la aplicación particular. Este proceso de selección no es obvio y pertenece al área de investigación abierta de la evaluación de arquitecturas, donde se han realizados numerosos trabajos previos (Losavio F., Chirinos L., Matteo A. y Ramdane-Cherif A., 2004).

b.2) Especificar *la calidad funcional para la familia de aplicaciones basadas en WS transaccionales*: Para realizar **intercambio de datos** se requiere *seguridad, precisión y eficiencia (tiempo de respuesta)* en la transmisión de datos, además de la *interoperabilidad y confiabilidad (disponibilidad)* por parte del servidor, que son aseguradas por la SOA. Para **control de acceso** se requiere *seguridad*.

*Salida:*

El modelo de calidad para el dominio de aplicaciones basadas en WS transaccionales, está constituido por las características y sub-características de calidad encontradas en b.1 y b.2, según el **Cuadro 1: mantenibilidad (extensión), eficiencia (tiempo de respuesta), confiabilidad (disponibilidad), portabilidad (escalabilidad)**, además de las sub-características de funcionalidad, que son: *interoperabilidad, seguridad y precisión*.

Los *RNF globales*, que deben ser cumplidos en general por todo el sistema, se derivan directamente del modelo de calidad del dominio, considerando los requisitos de calidad arquitecturales obtenidos en la actividad **b.1**, es decir: *interoperabilidad, confiabilidad (disponibilidad), mantenibilidad (extensión), portabilidad (escalabilidad)*. La calidad funcional, actividades **a** y **b.2**, será utilizada en el paso 3, para justificar los objetivos de calidad de los RF.

### **Paso 2. Identificar actores y RNF**

Los actores humanos que se pueden detectar de la descripción del problema son: *Proveedor, Gerente de Item de Servicio, Gerente de Aprobación de Propuesta*. El *Sistema de Facturación de Materiales* es también un actor, pero no humano. Se identifica un solo RNF, la *usabilidad*, asociado a cada uno de los actores humanos, como se muestra en el **Cuadro 2**.

**Cuadro 2**  
**Lista de funcionalidades del Sistema de Fijación de Precios**

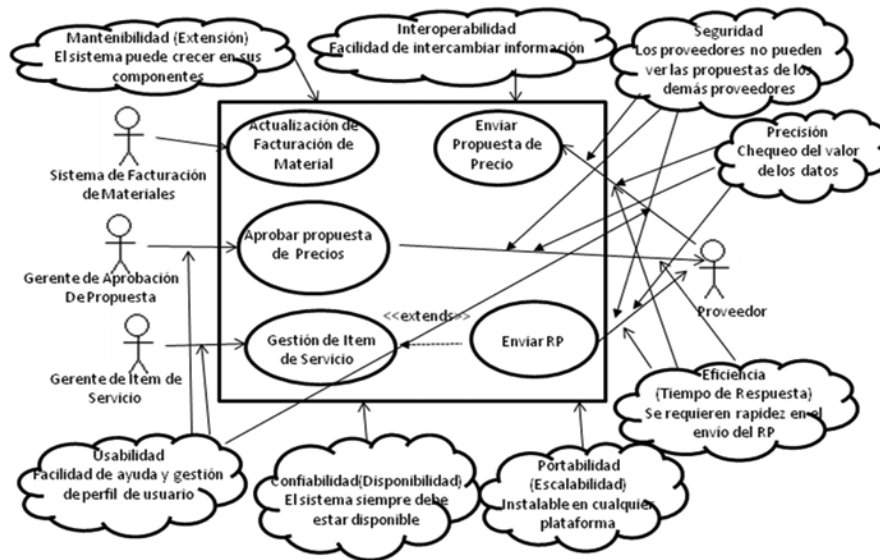
<b>Requisito Funcional (RF)</b>	<b>Calidad asociada (RNF)</b>
a. Gestión de item de servicio	Usabilidad
b. Enviar requerimiento de propuesta (RP) (intercambio de datos)	Eficiencia (Tiempo de respuesta), seguridad, precisión
c. Enviar propuesta de precios (PP) (intercambio de datos)	Usabilidad, eficiencia (tiempo de respuesta), seguridad, precisión
d) Aprobar propuesta de precios (PP) (intercambio de datos)	Usabilidad, eficiencia (tiempo de respuesta), seguridad, precisión

### **Paso 3. Identificar casos de uso (RF) y RNF**

Con la ayuda del marco NFR de Chung, Nixon, Yu, y Mylopoulos (2000), de la descripción del problema y considerando las funcionalidades y sus propiedades de calidad, obtenidas en las actividades **a** y **b.2** del **paso 1**, se obtienen todas las

funcionalidades del Sistema de Fijación de Precios, a las cuales corresponderán los casos de uso y los objetivos de calidad asociados, en el **Gráfico 4**. La mayoría de los requisitos de calidad (RNF) son obtenidos observando la correspondencia con las funcionalidades principales detectadas en el **paso 1**, como se muestra en el **Cuadro 2**.

**Gráfico 4**  
**Modelo de Caso de uso y los RNF Asociados**



El **Gráfico 4** muestra el modelo de casos de uso, todos los RNF asociados, además de los RNF globales del Sistema de Fijación de Precios.

#### **Paso 4. Reestructurar los RNF comunes:**

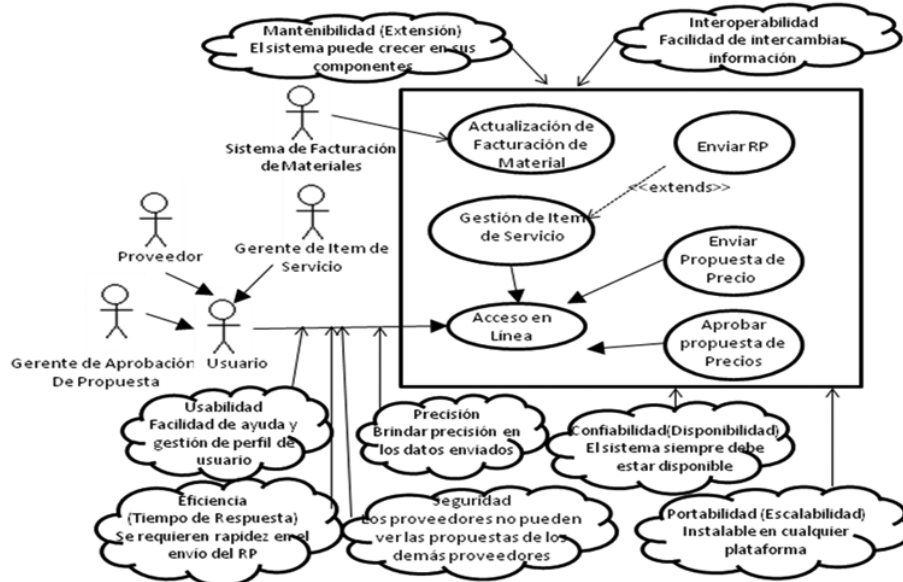
De acuerdo a Chung, significa hacer uso de las relaciones de generalización/especialización para reacomodar los RNF. Del **Cuadro 2**, se observa que el RNF usabilidad, se aplica a los tres casos de uso en donde intervienen actores humanos. En este caso, se decide crear el caso de

uso Acceso en Línea que generaliza Gestión de Item de Servicio, Enviar PP y Aprobar PP. Nótese que el caso de uso *Acceso en Línea* implica tener control de acceso, funcionalidad detectada en la actividad **a)** del **paso 1** y requiere seguridad en **b.1**. Esta generalización afecta a los actores porque son diferentes para cada caso de uso, por lo tanto también se pueden generalizar los actores Proveedor, Gerente de Item de Servicio y Gerente de Aprobación, en un sólo actor llamado Usuario y asociar el RNF usabilidad a la asociación entre el actor Usuario y el caso de uso Acceso en Línea;

cabe señalar, que por las reglas de propagación, esto implica que el RNF usabilidad se propaga a los casos de usos incluidos, en el **Gráfico 5**. Por otra parte, también los RNF tiempo de respuesta, seguridad y precisión intervienen en todas las

transmisiones de datos: Enviar RP, Enviar PP y Aprobar PP; y por lo tanto se asocian ahora a la comunicación entre Usuario y Acceso en Línea, que es el punto de entrada de todos los actores al sistema.

**Gráfico 5**  
**Reestructuración de los RNF comunes en el Modelo de Casos de Uso**

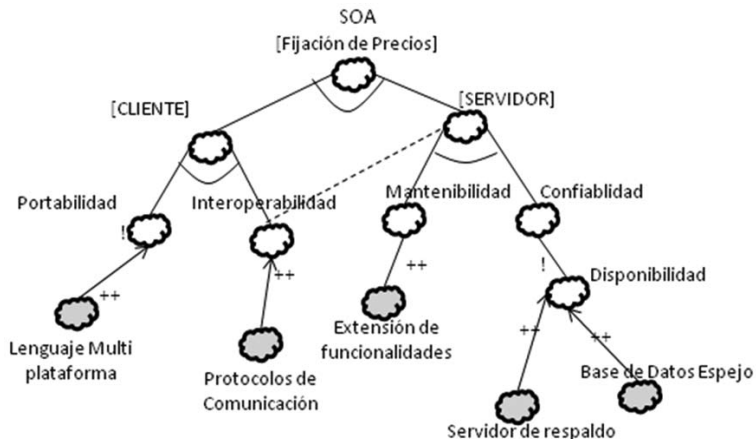


### **Paso 5. Refinar y satisfacer los softgoals RNF**

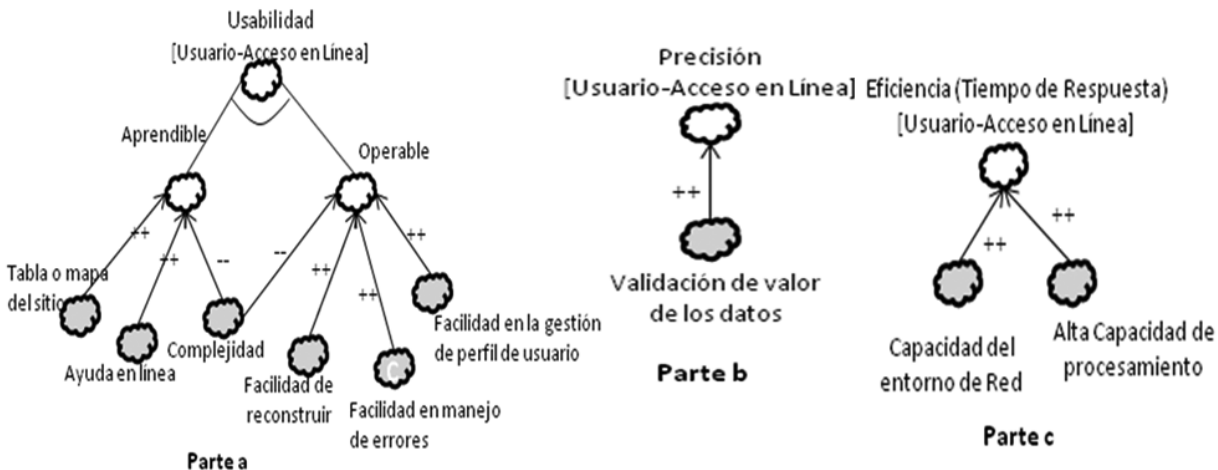
Se realiza un SIG por cada RNF asociado al Modelo de casos de uso. Los RNF globales se expresan en un SIG para la arquitectura; el

**Gráfico 6**, muestra el SIG para SOA determinada también en el análisis del dominio. Las partes a, b, c del **Gráfico 7** y la parte d.1 del **Gráfico 8** muestran los SIG para los RNF usabilidad, precisión, eficiencia (tiempo de respuesta) y seguridad respectivamente.

**Gráfico 6**  
**Gráfico de interdependencia de objetivos (SIG) de SOA**



**Gráfico 7**  
**SIG de los RNFs asociados a las funcionalidades del sistema (partes a, b, c)**

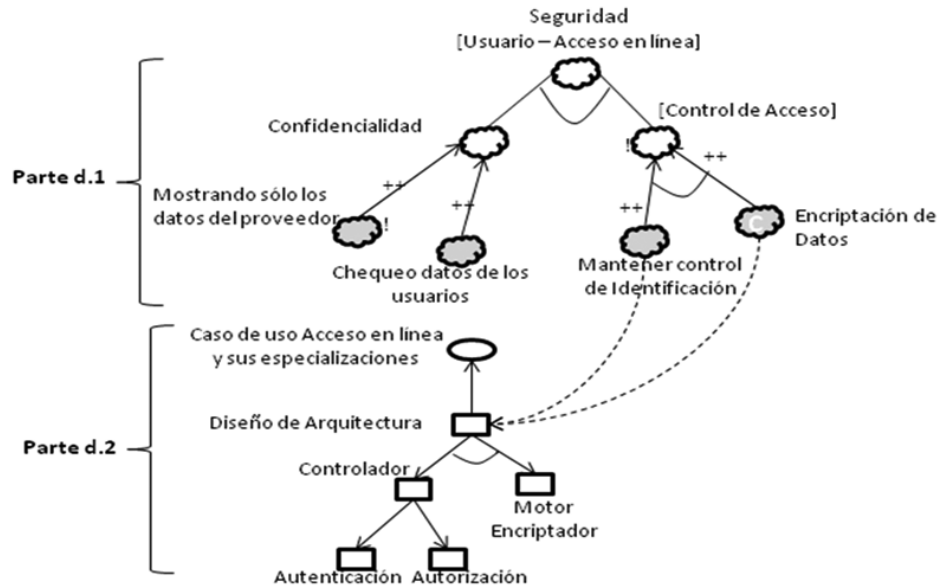


### **Paso 6. Satisfacer las operacionalizaciones de softgoals**

En el SIG de seguridad se seleccionaron como ejemplo las operacionalizaciones “Mantener control de Identificación” y “Encriptación de datos” en la parte d.2 del **Gráfico 8**, para obtener componentes de las soluciones arquitecturales para el RNF seguridad; se construyen de esta manera componentes para cada SIG, para así obtener una

arquitectura inicial o baseline para el Sistemas de Fijación de precios, dentro de un enfoque de línea de producto. Nótese que el uso del modelo de calidad ISO/IEC 9126-1 ha modificado algunos SIG del enfoque de Chung L., Supakkul S. (2004), en particular la seguridad en este estándar no incluye ni confiabilidad ni precisión, que son tratadas aquí como SIGs separados. En el **Gráfico 6** se muestra el SIG que corresponde a la arquitectura orientada a servicios (SOA) de la aplicación.

**Gráfico 8**  
**SIG de los RNFs asociados a las funcionalidades del sistema (partes d.1 y d.2)**



**Paso 7.** Diseño de casos de uso. Luego de determinar la satisfacción de los softgoals y las decisiones de diseño, se elaboran los diagramas de colaboración y secuencia para la implementación de cada caso de uso, los cuales no se muestran aquí para facilitar la lectura del documento y consideraciones de espacio.

## **Conclusión**

Se propone un proceso de análisis del dominio a ser incluido en el proceso de Chung como primer paso, para la integración de los RNF y los FR en los puntos de asociación con los elementos del diagrama de casos de uso; se hace especial énfasis en el uso del estándar de calidad ISO/IEC 9126-1 para especificar los RNF. La mayor contribución de este trabajo consiste en reutilizar en el proceso de Chung, los resultados del análisis del dominio del Paso 1, los cuales se reutilizan también en el paso 2, para definir los RNF globales del sistema, no justificados en el proceso de Chung y en el paso 3, para justificar los requisitos de calidad que deben ser satisfechos para las distintas funcionalidades de la aplicación. La especificación de los requisitos de calidad por estándares internacionales (ISO/IEC 9126-1, 2001), comúnmente aceptados por la comunidad, facilitan el entendimiento entre los grupos que desarrollan el proyecto de software, ofreciendo un lenguaje unificado sobre la calidad de software, del cual se carece. Nuestra contribución principal y resultado es el paso de análisis del dominio como extensión al proceso de Chung, el cual parece ser de fácil aplicabilidad en el contexto de métodos de diseño de líneas de productos y de desarrollo de software centrados en la arquitec-

tura, ofreciendo también un lenguaje unificado sobre la calidad del producto software, del cual se carece en general. En particular, pensamos utilizar el nuevo estándar ISO/IEC 25010 en el desarrollo de los trabajos futuros en esta línea de investigación, en cuanto esté oficialmente aprobado. Por otra parte, definir un proceso general y reutilizable de análisis del dominio que pueda ser incorporado a métodos centrados en la arquitectura o de líneas de producción, en un contexto de desarrollo de software orientado a aspectos, es un área abierta de investigación. Incorporar el proceso con análisis de dominios basado en estándares de calidad para construir una arquitectura inicial, por ejemplo en las fases de Inicio y Elaboración del Proceso Unificado, abreviado UP del inglés Unified Process (Booch, Rumbaugh y Jacobson, 1999), es una investigación en curso. Se han hecho trabajos que contemplan procesos con estándares de calidad y UP (Losavio, Chirinos, Matteo, Lévy y Ramdane-Cherif, 2004; Cooper, Abraham, Unnithan, Chung y Courtney, 2006), sin embargo no se considera allí un enfoque de análisis del dominio. Otros trabajos en curso son, por una parte la comparación de enfoques similares, por ejemplo en Brito I., Moreira A. (2004) de integración de RNF y RF con escenarios con el enfoque orientado a objetivos y aplicar nuestro análisis de dominio con estándares a los enfoques que resulten más prometedores de la comparación; debe también experimentarse con un mayor número de casos de estudios realistas. Así mismo se está estudiando la especificación de la vista de calidad del conocimiento del dominio mediante ontologías (Sancho P. P., Juiz C., Puigjaner R., Chung L. y Subramanian N., 2007), para facilitar la integración de diferentes estándar-



res, los cuales son utilizados en la práctica y la recuperación de las métricas correspondientes a los atributos de calidad.

## Bibliografía

- Berard, E. (1992). *Essays on Object-Oriented Software Engineering*, Prentice Hal, New Jersey, U.S.A
- Berghout, E., Solingen, R. (1999). *The Goal/Question/Metric Method (GQM) A practical method for quality improvement of software development*. London, McGraw Hill
- Booch, G., Rumbaugh, J. y Jacobson, I. (1999). *The Unified Modeling Language User Guide*, Addison-Wesley, Boston, U.S.A.
- Brito, I., Moreira, A. (2004). *Integrating the NFR Framework in a RE Model. Early Aspects: Aspect-Oriented Requirements Engineering and Architecture Design*, Lancaster, UK. 2004
- Carnegie Mellon. Software Engineering Institute (2003). *Client/Server Software Architectures-An Overview*. Disponible en <http://web.simmons.edu/~benoit/LIS455/Client-Server1.doc>
- Chung, L., Nixon B. A., Yu E., y Mylopoulos, J. (2000). *Non-Functional Requirements in Software Engineering*. Kluwer Academic Publishers
- Chung, L., Supakkul, S. (2004). *Integrating FRs and NFRs: A Use Case and Goal Driven Approach*. 2<sup>nd</sup> Inter. Conference on Software Engineering (SERA'04), pp 30-37
- Cooper, K., Abraham S. P., Unnithan R. S., Chung L. y Courtney, S.(2006). *Integrating Goal Models in the Rational Unified Process*. *Journal of Visual Languages & Computing*: 17(6), Dec. 2006, pp. 551-583
- Dardenne, A., Lamsweerde A. Von (1993). *Goal-Directed Requirements Acquisition*. *Science of Computer Programming*, vol. 20, 179-190
- Dromey (1994). *A Model for Software Product Quality*. Disponible en [www.sqi.gu.edu.au/docs/sqi/technical/Model\\_For\\_S\\_W\\_Prod\\_Qual.pdf](http://www.sqi.gu.edu.au/docs/sqi/technical/Model_For_S_W_Prod_Qual.pdf)
- ISO/IEC 9126-1 (2001). *International Organization for Standardization (ISO)*. *Software engineering-Product quality - Part 1: Quality model*.  
[http://www.iso.org/iso/iso\\_catalogue/catalogue\\_tc/catalogue\\_detail.htm?csnumber=22749](http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=22749)
- Lamsweerde A. Von (2001). *Goal-Oriented Requirements Engineering: A Guided Tour*. 5<sup>th</sup> Int'l Symp. On RE, IEEE CS Press, pp. 249-261
- Losavio, F., Matteo, A. y Rahamut, R. (2008). *Web Services Domain Analysis Based on Quality Standards 2<sup>nd</sup> European Conference on Software Architecture*, Cyprus, 2008, R. Morrison, D. Balasubramaniam, and K. Falkner (Eds.): *ECSA 2008*, LNCS Vol. 5292, 354-358 © Springer-Verlag Berlin Heidelberg
- Losavio, F., Chirinos, L., Matteo, A. y Ramdane-Cherif, A. (2004). *ISO quality standards for measuring architectures*. *The Journal of Systems and Software (JSS)*. Vol. 72 (2), 209-223
- Losavio, F., Chirinos, L., Matteo, A., Lévy, N. y Ramdane-Cherif, A. (2004). *Designing Quality Architecture: Incorporating ISO Standards into the Unified Process*. *Information Systems Management (ISYM)*, Auerbach Publications, Vol. 21(1), 27-44
- Moreira, A., Brito, I. y Araújo, J. (2002). *Crosscutting quality attributes for requirements engineering*. The 14<sup>th</sup>. *International Conference on Software Engineering and Knowledge Engineering (SE-KE'02)*, Ischia, Italy, ACM Press, pp 167-174

- OMG. Object Management Group. Unified Modeling Language (UML). [www.uml.org/](http://www.uml.org/)
- Sancho, P. P., Juiz, C., Puigjaner, R., Chung, L. y Subramanian, N. (2007). An Approach to Ontology-aided Performance Engineering through NFR Framework. Proc. WOSP'07, Buenos Aires, Argentina. ACM (Order No. 488073). Feb. 5-8, 2007. pp. 125-128
- Sousa, G., Soares, S., Borba, P. y Castro, J. (2004). Separation of Crosscutting Concerns from Requirements to Design: Adapting a Use Case Driven Approach. Early Aspects 2004: Aspect-Oriented Requirements Engineering and Architecture Design. Workshop at International Conference on Aspect-Oriented Software Development (AOSD), Lancaster UK. pp 93-102
- World Wide Web Consortium (2004). Web Services Architecture Requirements. W3C Working Group Note 11 February. Copyright © 2004 W3C® (MIT, ERCIM, Keio). Disponible en <http://www.w3.org/TR/wsa-reqs>
- Yu, E., Mylopoulos, J. (1998). Why goal-oriented requirements engineering, Proceedings of the Fourth International Workshop on Requirements Engineering: Foundations of Software Quality, Pisa, Italy, pp. 15-22

## **Agradecimiento**

Agradecemos altamente a los árbitros por la cuidadosa lectura del documento y las observaciones formuladas, las cuales redundan en beneficio de la calidad de este trabajo. Así mismo, queremos agradecer al Sr. Jorgen Boegh, miembro activo del JTC 1/SC7 WG6 de la organización ISO/IEC por corroborar la información sobre la vigencia del estándar ISO/IEC 9126-1.