



**sQMA - HERRAMIENTA PARA AUDITORIA DE
NIVELES DE SERVICIO DE SISTEMAS DE
INFORMACIÓN**

TRABAJO DE GRADO

**Ing. Jaime Andrés Montoya Vásquez
Ing. Cristian Camilo Palacios Ruiz**

**Asesor
Hugo Arboleda PhD.**

**FACULTAD DE INGENIERÍA
MAESTRÍA EN GESTIÓN INFORMÁTICA Y TELECOMUNICACIONES
SANTIAGO DE CALI
2014**

AUDITORIA DE NIVELES DE SERVICIO DE SISTEMAS DE INFORMACIÓN: ITIL V3 – ISO 25000

**Ing. Jaime Andrés Montoya Vásquez
Ing. Cristian Camilo Palacios Ruiz**

**Trabajo de grado para optar al título de Magister en Gestión de Informática y
Telecomunicaciones
Concentración en Gobierno de TI**

**Asesor
Hugo Arboleda PhD.**



**FACULTAD DE INGENIERÍA
MAESTRÍA EN GESTIÓN INFORMÁTICA Y TELECOMUNICACIONES
SANTIAGO DE CALI
2014**

Nota de aceptación

Firma del Presidente del Jurado

Firma del Jurado

Firma del Jurado

Santiago de Cali, _____ de 2014

Tabla de contenido

Capítulo 1: Introducción	1
1.1 Planteamiento del problema	2
1.2 Objetivos.....	3
1.2.1 Objetivo general	3
1.2.2 Objetivos específicos.....	3
1.3 Resumen de resultados obtenidos	3
1.4 Organización del documento	4
Capítulo 2: Antecedentes	5
2.1 Serie de estándares ISO/IEC 25000	5
2.1.1 Conceptos asociados	6
2.1.2 Enfoques y modelos asociados	7
2.2 Gestión de Niveles de Servicio (Service Level Management - SLM)	12
2.2.1 Conceptos asociados [12]	12
2.2.2 Roles [14]	14
2.2.3 Proceso de gestión de niveles de servicio ITIL V3.....	14
2.3 Estado del arte.....	15
Capítulo 3: Modelo de auditoria de niveles de servicio de sistemas de información	19
3.1 Modelo de dominio.....	21
3.1.1 Subdominio de configuración de modelos de calidad SQuaRE	21
3.1.2 Subdominio de configuración de modelos de calidad SQuaRE	22
3.1.3 Escenarios de calidad: elemento de integración de subdominios	23
3.2 Modelo operacional.....	24
3.3 Validación del modelo de auditoría de niveles de servicio de sistemas de información	27
Capítulo 4: Herramienta de soporte de auditoria de niveles de servicio de sistemas de información	28
4.1 Descripción de arquitectura	28
4.1.1 Vista de casos de uso	28
4.1.2 Vista lógica.....	31
4.1.3 Vista de procesos.....	32
4.1.4 Vista de implementación.....	33
4.1.5 Vista de despliegue	34
4.2 Implementación de referencia	34

4.2.1	Decisiones de arquitectura	36
4.2.2	Resumen de la pila de tecnologías	37
4.3	Validación de la herramienta sQMA	37
4.3.1	Caso de prueba funcional 1: Eficiencia - Peor tiempo de respuesta	38
4.3.2	Caso de prueba funcional 2: Recuperabilidad - Disponibilidad	39
Capítulo 5:	Conclusiones y trabajo futuro	40
5.1	Conclusiones	40
5.1.1	Conclusiones del modelo de auditoria de niveles de servicio de sistemas de información	40
5.1.2	Conclusiones de la herramienta de soporte de auditoria de niveles de servicio de sistemas de información	41
5.1.3	Conclusiones de la validación del modelo de auditoría de niveles de servicio de sistemas de información	41
5.1.4	Conclusiones generales	41
5.2	Trabajo futuro	42
Bibliografía		43

Listado de figuras

Figura 2.1. Organización de la serie de estándares SQuaRE.....	6
Figura 2.2. Ciclo de vida de la calidad del software SQuaRE.....	7
Figura 2.3. Estructura de los modelos de calidad SQuaRE.....	8
Figura 2.4. Alcance de las medidas de calidad SQuaRE	9
Figura 2.5. Modelo de ciclo de vida de calidad de producto de software SQuaRE	10
Figura 2.6. Modelo de referencia de medición de calidad de producto de software SQuaRE	12
Figura 2.7. Estructura del catálogo de servicios.....	13
Figura 2.8. Procesos de gestión de niveles de servicio ITIL V3	15
Figura 2.9. Modelo de información de medición ISO 15939.....	16
Figura 2.10. Fases del modelo de medición ISO 15939.....	17
Figura 2.11. Proceso de gestión de niveles de servicio	18
Figura 3.1. Estructura del modelo de auditoría de niveles de servicio.....	20
Figura 3.2. Subdominio de configuración de modelos de calidad SQuaRE	22
Figura 3.3. Subdominio de auditoría de niveles de servicio.....	23
Figura 3.4. Modelo operacional de auditoría de niveles de servicio	25
Figura 4.1. Diagrama de casos de uso de auditoría de niveles de servicio sQMA	29
Figura 4.2. Diagrama de paquetes sQMA	31
Figura 4.3. Diagrama de clases de dominio sQMA.....	32
Figura 4.4. Diagrama de clases de componentes de auditoría sQMA.....	32
Figura 4.5. Diagrama de clases de objetos de transferencia para recolección sQMA	32
Figura 4.6. Diagrama de secuencia de auditoría de niveles de servicio sQMA	33
Figura 4.7. Diagrama de componentes de auditoría sQMA.....	34
Figura 4.8. Diagrama de despliegue sQMA	34
Figura 4.9. WLSD Servicio Colector de datos de calidad sQMA	35
Figura 4.10. Esquemas XSD de transfer objects del servicio de recolección de datos de calidad sQMA	35
Figura 4.11. Interfaz de usuario lista de escenarios de calidad.....	35
Figura 4.12. Interfaz de usuario de evaluaciones de escenario de calidad sQMA.....	36
Figura 4.13. Función Groovy de procesamiento para métrica de tiempo de respuesta	36

Listado de tablas

Tabla 1.1. Modelos y estándares de calidad de software	2
Tabla 3.1. Formato de especificación de escenario de calidad	24
Tabla 3.2. Especificación de actividad: Especificar escenario de calidad.....	25
Tabla 3.3. Especificación de actividad: Recolectar datos de nivel de servicio	26
Tabla 3.4. Especificación de actividad: Procesar datos de niveles de servicio	26
Tabla 3.5. Especificación de actividad: Evaluar escenario de calidad	26
Tabla 3.6. Especificación de actividad: Comunicar reporte de nivel de servicio	26
Tabla 4.1. Especificación de caso de uso UC1: Diseñar escenario de calidad.....	29
Tabla 4.2. Especificación de caso de uso UC2: Monitorear escenario de calidad	30
Tabla 4.3. Requerimientos de alto nivel para el modelo operacional.....	30
Tabla 4.4. Mapeo entre procesos del modelo operacional y requerimientos funcionales	30
Tabla 4.5. Descripción de paquetes sQMA	31
Tabla 4.6. Escenario de calidad para caso de prueba 1.....	38
Tabla 4.7. Configuración del cliente de simulación para caso de prueba 1	38
Tabla 4.8. Escenario de calidad para caso de prueba 2.....	39
Tabla 4.9. Configuración del cliente de simulación para caso de prueba 2	39

Capítulo 1

Introducción

Existe una tendencia actual de crecimiento de la oferta y demanda de servicios de información en las organizaciones. El software juega un papel cada vez más importante. Su correcta operación usualmente resulta crítica para el éxito del negocio, por tanto desarrollar o seleccionar productos de alta calidad es una necesidad que no se puede tomar a ligera. Una exhaustiva especificación y evaluación de calidad constituye un factor de éxito para asegurar la conveniencia de las soluciones. Este proceso inicia con la definición de las características de calidad adecuadas, con base en el propósito de uso del producto. Es importante que cada aspecto relevante sea especificado y evaluado.

El área de TI en una organización enfrenta desafíos como: identificar correctamente los requerimientos de negocio, seleccionar adecuadamente soluciones y proveedores, garantizar el cumplimiento del tiempo de salida al mercado, asegurar la interoperabilidad de las soluciones y contar con información oportuna de su comportamiento para promover acciones de mejora, en el marco de un modelo de interacción entre cliente y proveedor que pactan acuerdos para establecer los términos del intercambio de información incluyendo la calidad, a través de servicios clasificados en un catálogo que ofrece una visión operacional de la organización.

Toda relación de negocios se basa en expectativas que deben ser establecidas, gestionadas y cumplidas. En el contexto de la gestión de servicios de TI, las expectativas están directamente relacionadas con los servicios entregados y los acuerdos que se hacen en torno a estos. De forma general, la gestión de niveles de servicio es un proceso mediante el cual se establecen puntos de referencia, se mide el rendimiento y se asegura el cumplimiento de los acuerdos establecidos con los clientes. Mediante la implementación adecuada de esta práctica se puede mejorar la calidad de los servicios y en consecuencia la satisfacción del cliente; alcanzar metas internas asociadas a los servicios; gestionar, analizar y reportar sobre su comportamiento para proveer la retroalimentación necesaria que permita mantener el grado de servicio esperado.

Por su parte, la calidad del software es una disciplina de la ingeniería de software que trata la coherencia entre los requerimientos funcionales y no funcionales, los estándares de desarrollo adoptados y las características implícitas esperadas. Bajo esta definición, los requerimientos de software son la base de las medidas de calidad. La calidad del software debe ser considerada en todas las fases del ciclo de vida del producto: especificación, diseño, implementación, mantenimiento, etc. Dado que es un proceso incremental, se parte de la premisa de que la calidad del proceso asegura la calidad del producto.

Como se puede apreciar en la Tabla 1.1, existen numerosos modelos y estándares de gestión de calidad de software, cada uno desde su enfoque aporta elementos de valor, diferenciando principalmente dos grupos: el primero centrado en el proceso de construcción de las soluciones y el segundo destinado a evaluar la calidad desde la perspectiva del producto. No obstante, los objetivos que se establecen para la calidad del producto determinan los objetivos de la calidad del proceso de desarrollo, ya que la calidad del producto va a estar en función de la calidad del proceso de desarrollo.

Nivel	Modelo	Estándar
Proceso	CMMi	ISO 90003 ISO 20000
	TickIT	
	Bootstrap	ISO 15504 (SPICE)
	Personal SW Process (PSP)	IEEE / EIA 12207
	Team SW Process (TSP)	ISO 12207
	Practical SW Measurement (PSM)	ITIL Cobit 4.0
	Six Sigma for Software	
	Producto	Gilb
GQM		ISO 25000 (SQUARE)
Mc Call		IEEE Std 1061-1998
Furps		
Boehm		
SATC		
Dromey		
C-QM		
Metodología SQAE		
WebEQM		

Tabla 1.1. Modelos y estándares de calidad de software (Fuente: adaptado de [1])

Por un lado la gestión de niveles de servicio, por el otro la gestión de la calidad de software. Dos líneas de conocimiento ampliamente exploradas. Al hablar de gestión de niveles de servicio de sistemas de información se crea un punto de convergencia. Claramente tienen objetivos en común, se puede analizar su relación en función de la compatibilidad existente entre los modelos conceptuales y de proceso, para promover la construcción de herramientas que apoyen la gestión de oferta y demanda de servicios con el propósito de asegurar la satisfacción de los clientes y la optimización de recursos en la generación de valor al negocio.

1.1 Planteamiento del problema

La gestión de niveles de servicio puede ser considerado un factor crítico para el éxito de una organización que confíe parte de su estrategia u operación a servicios de información. El aseguramiento de la calidad de los servicios debe ser una prioridad, sin embargo, entre los principales desafíos se encuentran: identificar un responsable del servicio desde la perspectiva del cliente que exprese las necesidades reales del negocio; poca experiencia previa en esta práctica; diferentes perspectivas en el equipo de trabajo del lado del cliente frente a lo que se espera del servicio; datos de monitoreo previos no disponibles que imposibilitan asegurar que los niveles pactados sean apropiados [2].

Por otra parte algunos riesgos asociados a esta práctica son: falta de compromiso de parte del negocio y de los clientes; falta de herramientas y recursos necesarios para pactar, documentar, monitorear, reportar y revisar acuerdos y niveles de servicio; burocratización del proceso, se convierte en un proceso administrativo en vez de un proceso proactivo que le entrega beneficios medibles al negocio; las mediciones del negocio y del cliente son demasiado complejas de obtener y mejorar, porque lo que no se recolectan; altas expectativas y una baja percepción en el cliente, comunicación pobre e inadecuada con el negocio y los clientes [3].

Ya sea bajo un modelo de tercerización, como proveedores o clientes, las organizaciones deben ser conscientes de la creciente necesidad de gestionar su catálogo de servicios. En la etapa de concepción de las soluciones para identificar correctamente los requerimientos de negocio; en la recepción a satisfacción de los productos mediante la realización de pruebas de disponibilidad, escalabilidad, rendimiento, seguridad, usabilidad y demás atributos de calidad; durante la operación a través de la comprobación y actualización de acuerdos de nivel de servicio, partiendo del hecho que se requiere información precisa y oportuna para predecir el comportamiento de los sistemas.

Hoy, las organizaciones presentan escenarios de interacción cada vez más complejos y dinámicos, los servicios deben evolucionar y adaptarse rápidamente las necesidades de negocio. La gestión de niveles de servicio provee al negocio los objetivos de servicio pactados y la información de gestión requerida para asegurar su cumplimiento. Cuando existe una brecha provee la retroalimentación identificando efectivamente la causa y las acciones necesarias para mitigar la recurrencia del problema. La gestión de niveles de servicio constituye un canal de comunicación confiable con los clientes adecuados y con los representantes del negocio a un nivel táctico [4]. Algunos de los desafíos y riesgos hacen referencia a la especificación y comunicación de requerimientos, a la inexistencia o inmadurez de procesos en las áreas de TI, en este trabajo se presta especial atención al monitoreo del cumplimiento de los niveles de servicio. Se plantea la posibilidad de aportar a la siguiente problemática: *¿Cómo contribuir al aseguramiento de la calidad de los servicios de información en el marco de la gestión de niveles de servicio?*

1.2 Objetivos

1.2.1 Objetivo general

Construir una herramienta de auditoría de niveles de servicio de sistemas de información que permita monitorear de forma oportuna el cumplimiento de los acuerdos pactados entre clientes y proveedores.

1.2.2 Objetivos específicos

- Definir un modelo de auditoría de niveles de servicio de sistemas de información.
- Diseñar implementar y evaluar una herramienta para soportar el modelo.
- Validar el modelo y la herramienta propuestos.

1.3 Resumen de resultados obtenidos

- Modelo de auditoría de niveles de servicio de sistemas de información, compuesto por:
 - ✓ Modelo de dominio, vista conceptual que integra conceptos de (1) calidad de software seleccionadas de la serie de estándares ISO 25000 y (2) gestión de niveles de servicio seleccionadas de ITIL V3. Constituye la base para construir y evaluar escenarios de calidad estandarizados, los cuales son diseñados a partir de

acuerdos de nivel de servicio empleando modelos de calidad predefinidos que abarcan todo el ciclo de vida del producto de software, o modelos extendidos que permiten definir características y métricas a la medida del evaluador.

- ✓ Modelo operacional, vista de proceso especificada empleando notación BPMN, que integra las tareas requeridas para desarrollar actividades de evaluación de acuerdos de nivel de servicio tomando como referente la serie de estándares ISO 25000 e ITIL V3.
- Herramienta de soporte del modelo propuesto, denominada sQMA (Software Quality Measurement and Audit Automation Tool), que permite (1) la configuración de modelos de calidad extensibles en tiempo de ejecución, y (2) la definición de escenarios de calidad derivados de los acuerdos entre clientes y proveedores, para simplificar el proceso de auditoría de niveles de servicio.
- Casos de prueba empleados para validar la integralidad del modelo propuesto y la funcionalidad de la herramienta sQMA. Constituyen un referente para evaluar la adopción del modelo en entornos reales.

1.4 Organización del documento

Este documento ha sido dividido de la siguiente forma:

- El capítulo 2, denominado “Antecedentes”, contiene el marco de referencia con las definiciones y modelos seleccionados de ITIL V3 y de ISO 25000, que permiten entender el origen conceptual del modelo propuesto. Contiene también el estado del arte, compuesto por experiencias de otros autores relacionadas con la gestión niveles de servicio desde la perspectiva de la gestión de servicios de TI y de la gestión de la calidad del software. Este capítulo aporta definiciones, modelos y lineamientos a la propuesta.
- El capítulo 3, denominado “Modelo de auditoría de niveles de servicio de sistemas de información”, presenta un modelo de dominio y un modelo operacional para abordar la problemática de la auditoría de niveles de servicio de sistemas de información desde las perspectivas conceptual y de proceso respectivamente.
- El capítulo 4, denominado “Herramienta de soporte de auditoría de niveles de servicio de sistemas de información”, contiene la descripción de arquitectura y otros apartes de la construcción de la herramienta denominada sQMA (Software Quality Measurement and Analysis Tool).
- Finalmente el capítulo 5 presenta las conclusiones y el trabajo futuro identificado.

Capítulo 2

Antecedentes

Este capítulo contiene una recopilación los aspectos conceptuales y tecnológicos, base para la concepción y desarrollo del presente trabajo. Para una mejor comprensión ha sido dividido de la siguiente forma:

- Marco teórico: Estándares de la industria de gestión de calidad de software y de gestión de niveles de servicio seleccionados, que aportan definiciones, esquemas, lineamientos y buenas prácticas a la propuesta y que por el grado de madurez favorecen su aplicabilidad.
- Estado del arte: conjunto de experiencias seleccionadas de otros autores, que fueron empleadas para definir el alcance de la propuesta identificando enfoques comunes y aportes que puedan resultar de utilidad.

2.1 Serie de estándares ISO/IEC 25000

La serie de estándares ISO/IEC 25000 - ISO/IEC 25099, denominada SQuaRE (Software Quality Requirements and Evaluation) busca normalizar la definición de requerimientos y la evaluación de calidad de software. Las familias de estándares (divisiones) asociadas a esta serie son (Figura 2.1):

- ISO/IEC 2500n – División general de calidad del producto: Los estándares de esta división incluyen la definición de modelos y conceptos transversales a la serie SQuaRE.
- ISO/IEC 2501n – División de modelo de calidad: Los estándares de esta división incluyen la definición de modelos de calidad detallados.
- ISO/IEC 2502n – División de Medición de calidad: Está familia de estándares incluye la definición de un modelo de referencia de medición de calidad de productos de software, definiciones matemáticas de medidas de calidad y guías prácticas para su aplicación.
- ISO/IEC 2503n – División de requerimientos de calidad: El objetivo de los estándares que componen esta división es contribuir a la especificación de requerimientos de calidad. Estos requerimientos pueden ser usados en la elicitación de requerimientos de calidad para el desarrollo de un producto de software o como entradas para un proceso de evaluación.
- ISO/IEC 2504n – División de evaluación de calidad: Esta familia de estándares comprende requerimientos, recomendaciones, y guías para la evaluación de calidad de software, dirigida a evaluadores, compradores o desarrolladores.

- ISO/IEC 25050 - 25099 - Extensiones: Estándares de extensión para calidad de productos comerciales Off-The-Self y formatos comunes de industria para reportes de usabilidad.



Figura 2.1. Organización de la serie de estándares SQuaRE (Fuente: adaptado de [5])

2.1.1 Conceptos asociados

La serie SquaRE involucra un conjunto de conceptos que constituyen los fundamentos para la evaluación cuantitativa de atributos de calidad, los más relevantes se definen a continuación:

- *Atributo de calidad*: Propiedad inherente o característica de una entidad que puede ser identificada cualitativa o cuantitativamente por medios humanos o automatizados.
- *Medida*: Variable a la que se asigna un valor como resultado de una medición.
- *Medida base*: Medida definida de un atributo y el método para cuantificarlo, es independiente funcionalmente de otras medidas.
- *Medida derivada*: Medida definida en función de una (1) o más valores de medidas base.
- *Medición*: Conjunto de operaciones que tienen como objetivo determinar el valor de una medida.
- *Función de medición*: Algoritmo o cálculo realizado para combinar dos (2) o más medidas base.
- *Método de medición*: Secuencia lógica de operaciones, descrita de forma general, utilizada en la cuantificación de un atributo con respecto a una escala especificada.
- *Elemento de medición de calidad*: Medida que puede corresponder con una medida base o una medida derivada empleada para construir medidas de calidad de software.
- *Indicador*: Medida que provee una estimación o evaluación de los atributos especificados derivada de un modelo con respecto a las necesidades de información.

2.1.2 Enfoques y modelos asociados

La serie SQuaRE está compuesta por varias divisiones que a su vez agrupan estándares, a continuación se introducen los modelos y enfoques que permiten la comprensión de la propuesta.

2.1.2.1 Enfoques de Calidad de software

Las necesidades de calidad de software de los usuarios están asociadas a los requerimientos de calidad de software en uso en un contexto específico. A la vez, estas necesidades identificadas pueden ser usadas para especificar la *calidad interna* y *externa* a partir de *características* y *sub características* de calidad de software. La calidad de productos de software puede ser evaluada midiendo *atributos internos*, típicamente mediciones estáticas, o midiendo *atributos externos*, que usualmente reflejan el comportamiento del código en ejecución, o midiendo atributos de *calidad en uso*, cuando el producto es usado en ambientes reales. Como se puede observar en la Figura 2.2, mejorar la calidad del proceso, durante la fase de construcción, contribuye a mejorar la calidad del producto, en consecuencia impactando positivamente la calidad en uso. Recíprocamente la evaluación de calidad en uso provee la retroalimentación necesaria para mejorar la calidad del producto.

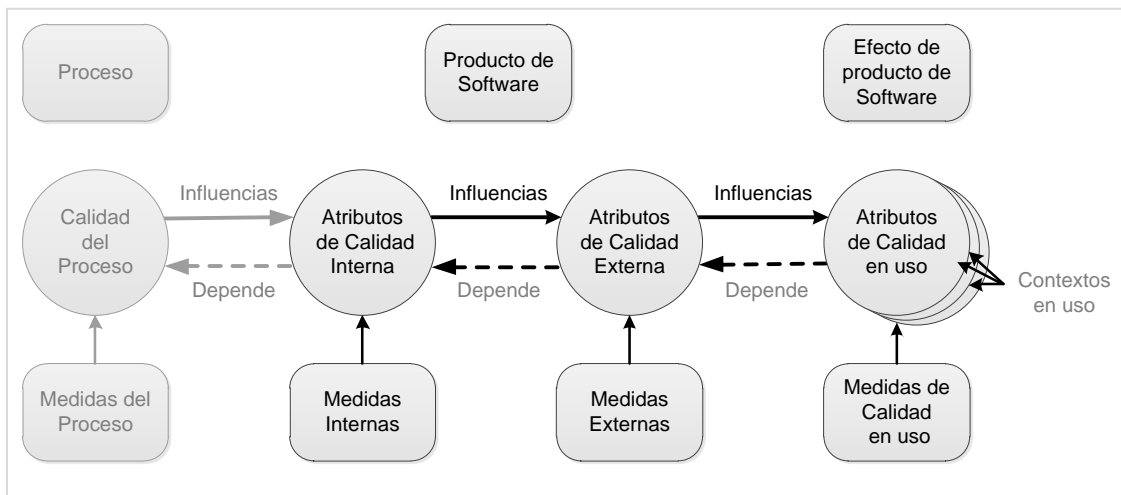


Figura 2.2. Ciclo de vida de la calidad del software SQuaRE (Fuente: adaptado de [6])

Los *atributos internos* apropiados son necesarios para lograr el comportamiento externo deseado, al tiempo que comportamientos externos apropiados son pre requisito para lograr la calidad en uso esperada.

2.1.2.2 Modelos de calidad de software

Las propiedades de software pueden ser inherentes al producto de software, depender del sistema o ser asignadas al producto. La calidad de un producto de software en un contexto particular de uso está determinada por sus *propiedades inherentes*, entendiendo por inherente, algo que existe de forma permanente como una característica o atributo. Así, las propiedades inherentes pueden ser clasificadas como *funcionales* o *de calidad*. Las *funcionales* determinan lo que está en capacidad de hacer el sistema, mientras las *de calidad* determinan qué también el software se desempeña, en otras palabras, muestran el grado en el que el software es capaz de proveer y mantener sus servicios. Las propiedades *de calidad* son inherentes al producto de software y al sistema asociado, por

tanto una propiedad asignada no es considerada *de calidad* dado que puede variar sin necesidad de cambiar el software. Los modelos de calidad especificados en la serie SQuaRE descomponen la calidad del producto en *características*, *sub características* y *atributos* (Figura 2.3), para conformar dos clasificaciones primarias, una aplicable a la calidad interna/externa y otra para calidad en uso. La relación de las características de uno y otro está determinada por el tipo de usuario, por ejemplo:

- Para un usuario final, la calidad en uso es principalmente un resultado de la idoneidad funcional, confiabilidad, operabilidad y eficiencia de rendimiento.
- Para quien mantiene el software, es un resultado de la mantenibilidad.
- Para quien migra el software es un resultado de la portabilidad.

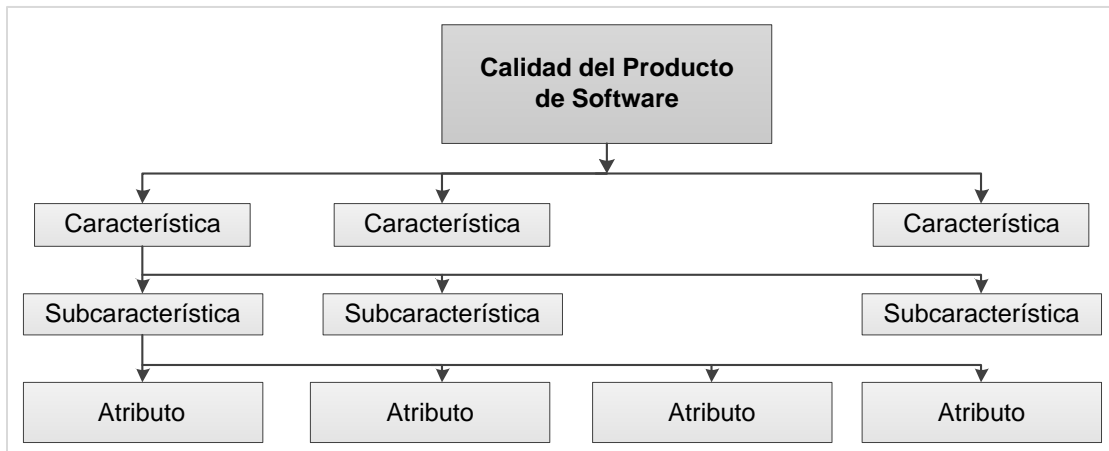


Figura 2.3. Estructura de los modelos de calidad SQuaRE (Fuente: adaptado de [7])

La calidad de un sistema es el resultado de la calidad de los componentes del sistema y su interacción, mientras que la calidad del software es el grado en que un producto de software satisface las necesidades establecidas al ser usado en condiciones específicas. En torno a estas definiciones, la serie SQuaRE establece inicialmente:

- Un *modelo de calidad interna/externa de software* compuesto por ocho (8) características: idoneidad funcional, confiabilidad, eficiencia de rendimiento, operabilidad, seguridad, mantenibilidad y transferibilidad, divididas en sub características que pueden ser medidas interna o externamente. Mientras la calidad externa del software provee una vista de caja negra del software y atiende propiedades relacionadas con su ejecución sobre el sistema operativo y el hardware, la calidad interna del software provee una vista de caja blanca a partir de propiedades que típicamente están disponibles en la etapa de construcción.
- Un *modelo de calidad en uso* compuesto por tres (3) características: usabilidad en uso, flexibilidad en uso y seguridad en uso, que son divididas en sub características que pueden ser medidas cuando el producto está en uso en un contexto específico. La calidad en uso es una medida de la calidad de todo el sistema en su contexto operacional para usuarios específicos desarrollando tareas específicas (Figura 2.4).

En la Figura 2.4 se puede apreciar cómo los 4 modelos de calidad obtienen medidas de sistemas de información operados en contextos de uso específicos, que a su vez pueden estar compuestos por subsistemas de hardware y software.

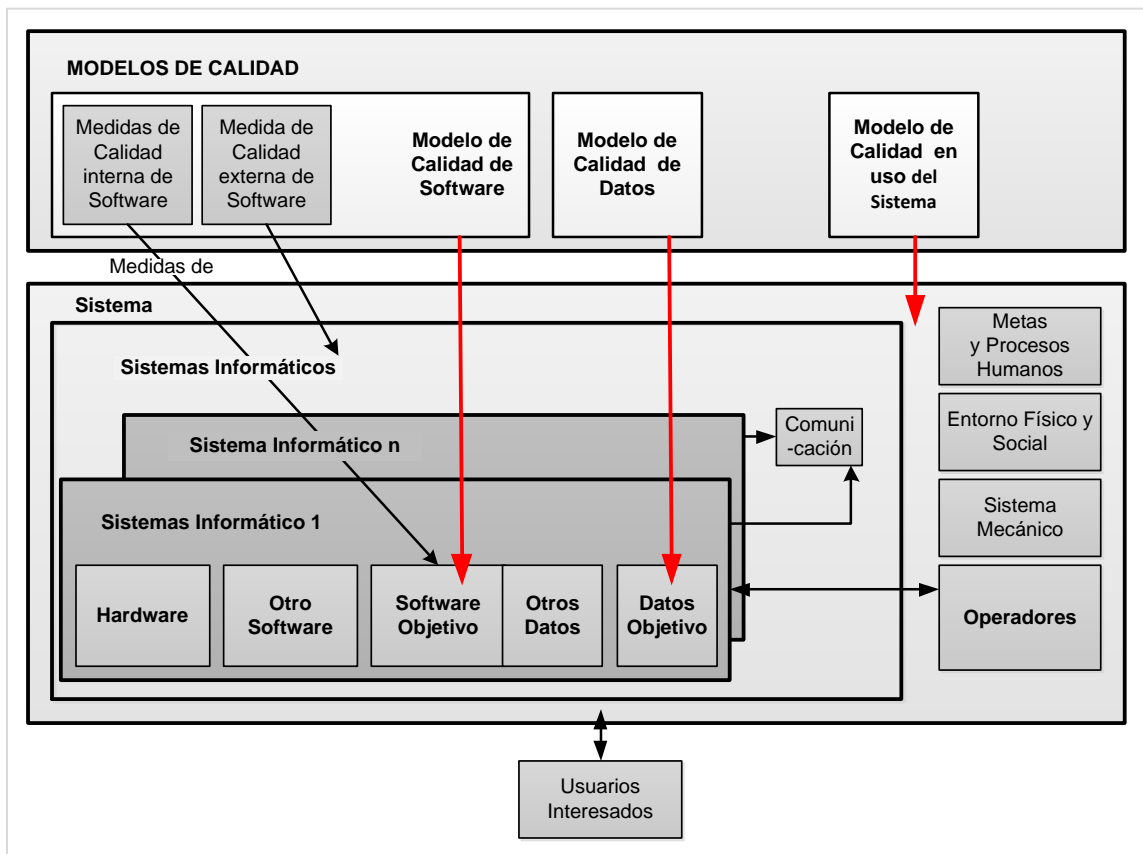


Figura 2.4. Alcance de las medidas de calidad SQuaRE (Fuente: adaptado de [7])

Las características y sub características definidas en los modelos son aplicables a cualquier tipo de software, proveen una terminología consistente y establecen un referente para validar la completitud de la definición requerimientos de calidad o pueden ser usados para soportar la especificación y evaluación de software desde diferentes perspectivas, relacionadas con la adquisición, definición de requerimientos, uso, evaluación, soporte, mantenimiento, aseguramiento de la calidad y auditoría.

Los modelos de calidad sirven como un marco de trabajo para garantizar que todos los aspectos de calidad son considerados desde las perspectivas de calidad interna, externa y en uso. Sin embargo, el estándar permite agregar sub características, definidas por el usuario en la jerarquía, que pueden ser empleadas en conjunto como lista de chequeo para asegurar la exhaustividad de la cobertura de calidad. En cuanto a conformidad, una definición de requerimientos o evaluación de calidad que se ajuste al estándar debe hacer uso de todo el conjunto de características y sub características del modelo seleccionado, justificando cualquier exclusión o describir su propio conjunto de características de calidad y proveer el mapeo con la línea base del estándar.

2.1.2.3 Modelo de ciclo de vida de calidad de productos de software

Este modelo aborda la calidad de productos de software en tres (3) fases principales del ciclo de vida del producto de software: *desarrollo*, *pruebas* y *uso*, estableciendo una relación directa con los enfoques de calidad de la siguiente forma (Figura 2.5):

- La fase de desarrollo es asunto de la calidad interna del software.

- La fase de pruebas de producto es asunto de la calidad externa del software.
- La fase de producto en uso es asunto de la calidad en uso del software.

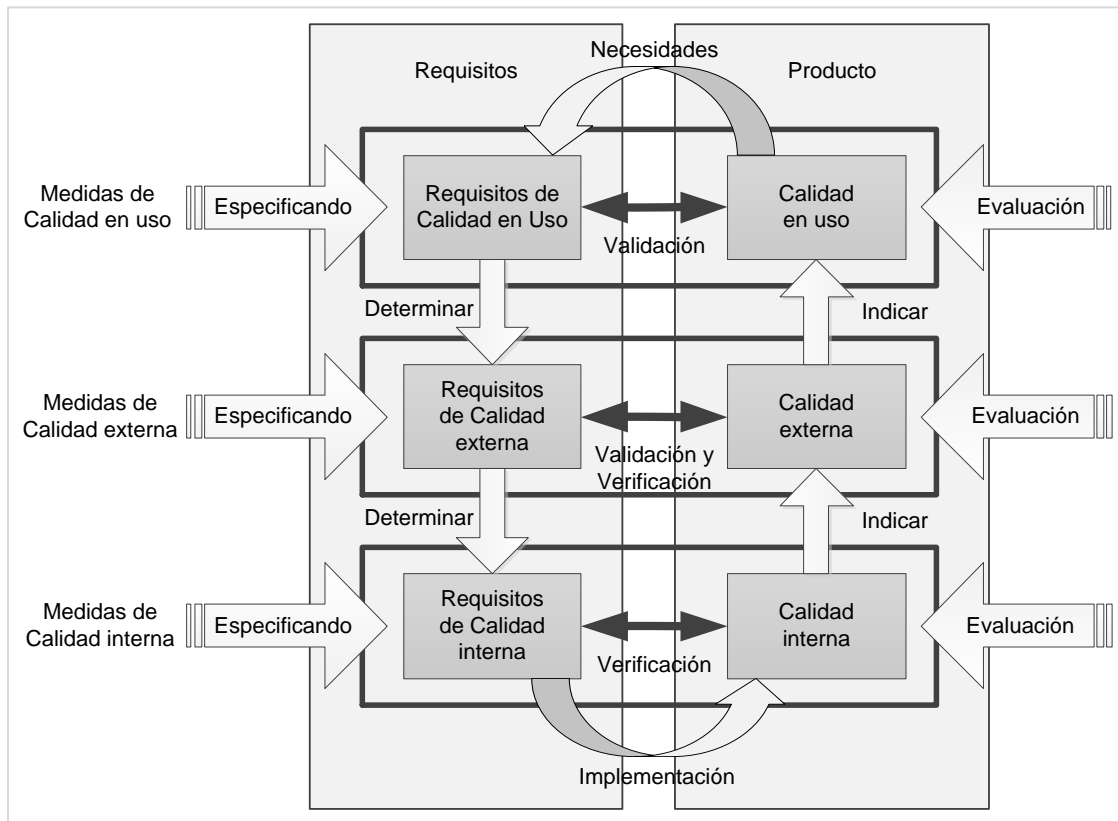


Figura 2.5. Modelo de ciclo de vida de calidad de producto de software SQuaRE (Fuente: adaptado de [8])

Este modelo también establece que lograr niveles aceptables de calidad de software debería ser parte integral del proceso de construcción para cada tipo de calidad incluyendo: requerimientos, implementación y validación de los resultados [9].

- Los requerimientos de calidad en uso especifican el nivel requerido de calidad desde el punto de vista de los usuarios finales, se derivan de sus necesidades y las de otros interesados. Los requerimientos de características de calidad en uso deben especificarse con base en medidas de calidad en uso estableciendo los criterios de evaluación del producto, y contribuyendo a identificar los requerimientos de calidad externa.
- Los requerimientos de calidad externa especifican el nivel de calidad requerido desde el punto de vista exterior, comprenden requerimientos derivados de las necesidades de calidad de los interesados incluyendo los requerimientos de calidad en uso. Son usados como objetivo para validaciones y verificaciones técnicas y se especifican cuantitativamente con base en medidas externas. Recíprocamente, la evaluación de calidad externa puede ser usada para predecir la calidad en uso.
- Los requerimientos de calidad interna de software especifican el nivel de calidad requerido desde el punto de vista interno del producto, incluyen requerimientos derivados de los requerimientos de calidad externa. Son empleados para especificar propiedades de productos intermedios de software como especificaciones y código fuente, o entregables no ejecutables como documentación y manuales. Pueden ser

empleados como objetivos de validación en diferentes fases de desarrollo, también pueden ser usados para definir estrategias de desarrollo y criterios para evaluación y verificación durante etapas de construcción. Deben ser especificados cuantitativamente en función de medidas internas y pueden ser usados para predecir la calidad externa del software.

Usualmente el software hace parte de sistemas más grandes constituidos por otros productos de software, en los que intervienen interfaces, hardware, operadores humanos y flujos de trabajo. Es relevante diferenciar la evaluación de calidad de un producto de software a la del sistema en el que se ejecuta, para el primer caso se podría tener únicamente en cuenta las originadas por requerimientos, diseño e implementación, mientras para el sistema pueden tenerse en cuenta fallas de hardware, errores humanos, etc. En etapas tempranas de la construcción del producto, únicamente se pueden evaluar recursos y procesos, cuando hay productos intermedios disponibles, pueden ser evaluados a partir de medidas de calidad interna, que pueden ser empeladas para predecir los valores de medidas de calidad externa. Un producto de software terminado puede ser evaluado a partir de medidas de calidad externa para describir su interacción con el entorno, observando el producto en operación [10]. Por su parte la calidad en uso puede ser medida a partir del grado en que un producto usado por usuarios específicos satisface sus necesidades para alcanzar objetivos específicos con efectividad, productividad, satisfacción, seguridad y flexibilidad.

2.1.2.4 Modelo de medición de calidad de software

Las características del software inherentes que pueden ser diferenciadas cualitativa o cuantitativamente se denominan *atributos*, clasificados en una o más *sub categorías* y medidos mediante la aplicación de un *método de medición*. Un método de medición, es una secuencia lógica de operaciones para cuantificar un atributo con respecto a una escala determinada, el resultado de su aplicación es un elemento de medición de calidad. Las características y sub características de calidad son cuantificadas mediante la aplicación de *funciones de medición*, algoritmos utilizados para combinar elementos de medición de calidad que generan medidas de calidad, donde puede emplearse más de una medida para la misma característica o sub característica (Figura 2.6).

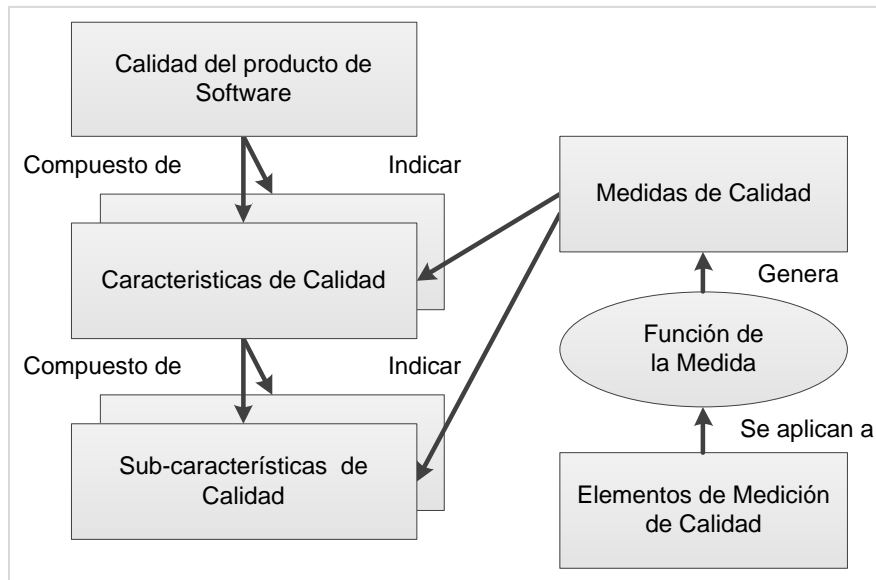


Figura 2.6. Modelo de referencia de medición de calidad de producto de software SQuaRE (Fuente: adaptado de [11])

2.2 Gestión de Niveles de Servicio (Service Level Management - SLM)

La Gestión de Niveles de Servicio es el proceso por el cual la calidad de los servicios de TI ofrecidos es definida, negociada y supervisada. Es responsable por encontrar un compromiso realista entre las necesidades y expectativas de los clientes, y el costo de los servicios asociados. Su propósito es poner la tecnología al servicio del cliente, siendo un medio para agregar valor a los usuarios y clientes, y no un fin. Con su implementación se busca asegurar la calidad de los servicios de TI alineando la tecnología con los procesos de negocio de una forma rentable. Para esto es esencial: conocer las necesidades del cliente, definir los servicios ofertados correctamente y monitorear la calidad con base en los objetivos fijados en los acuerdos de nivel de servicio.

Entre los objetivos de esta práctica se encuentran: documentar los servicios ofrecidos, presentar los servicios a los clientes en una forma comprensible, trabajar de la mano del cliente para proponer servicios realistas alineados a sus necesidades, establecer los acuerdos necesarios con clientes y proveedores para ofrecer los servicios correctos, definir los indicadores de rendimiento de los servicios, monitorear la calidad de los servicios para mejorarlos, entregar reportes de calidad de los servicios y de los planes de mejora de servicio.

2.2.1 Conceptos asociados [12]

Para comprender el proceso de gestión de niveles de servicio, se deben considerar al menos los siguientes conceptos:

- *Cliente:* Organización que contrata el servicio de TI.
- *Usuario:* Quien usa el servicio.
- *Proveedor:* Organización que provee el servicio requerido por el cliente.
- *Catálogo de servicios:* Es una herramienta esencial para simplificar la comunicación con el cliente. El catálogo debe describir los servicios ofrecidos en una forma no técnica accesible a los clientes y personal no especializado, ser usado como un guía

para orientar y dirigir a los clientes, incluir en términos generales, los niveles de servicio asociados con cada uno de los servicios ofrecidos, estar disponible a la mesa de servicio y todo el personal que tenga contacto directo con el cliente. Como se puede apreciar en la Figura 2.7, refleja la relación entre los servicios y los procesos de negocio, identificando los tipos de servicio y los acuerdos de nivel de servicio asociados para cada cliente.

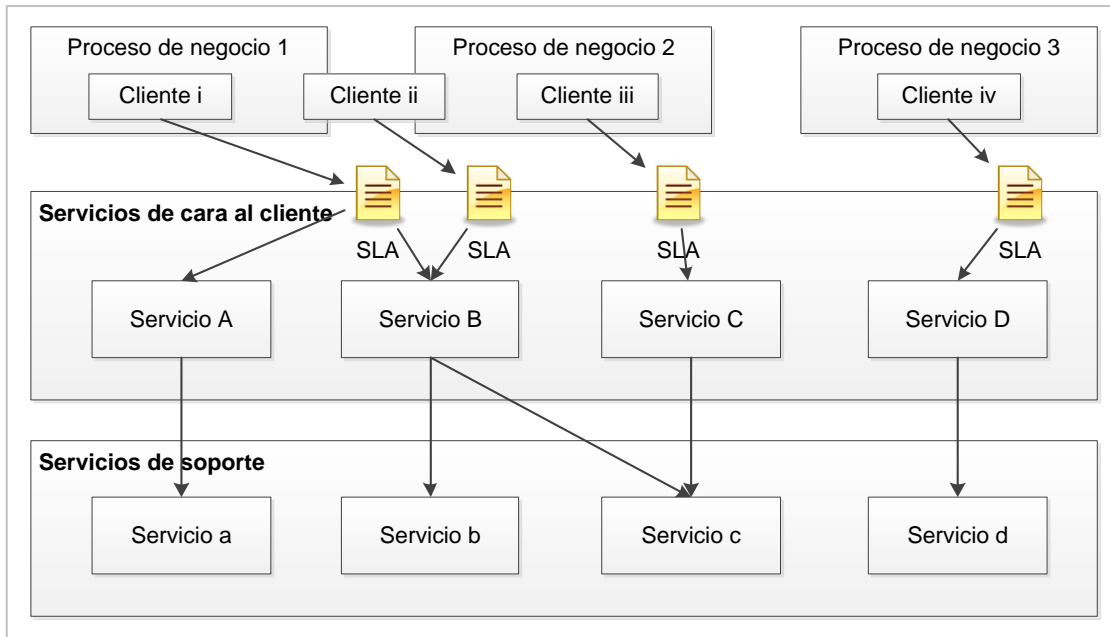


Figura 2.7. Estructura del catálogo de servicios (Fuente: Adaptado de [13])

- **Requerimientos de nivel de servicio (Service Level Requirements - SLR):** Debe incluir información detallada acerca de las necesidades del cliente y sus expectativas en términos del rendimiento y nivel de servicio. Es la base para la definición del acuerdo de nivel del servicio y los acuerdos de nivel operacional relacionados.
- **Hojas de especificación:** Son documentos técnicos de uso interno para delimitar y definir los servicios ofrecidos a los clientes. En ellas se debe evaluar los recursos necesarios para ofrecer los servicios requeridos con el nivel de calidad suficiente y determinar si es necesario tercerizar ciertos procesos; sirve como un documento base para especificar los acuerdos de nivel operacional y los contratos de apoyo.
- **Programa de calidad del servicio (Service Quality Programme - SQP):** Debe incorporar toda la información necesaria para permitir la gestión eficiente de los niveles de calidad del servicio, incluyendo objetivos para cada servicio, estimación de recursos, indicadores clave de rendimiento (Key Performance Indicator - KPI) y procedimientos de monitoreo al proveedor.
- **Acuerdo de nivel de servicio (Service Level Agreement – SLA):** Describe todos los detalles de los servicios ofrecidos en un lenguaje comprensible para el cliente. Después de firmado, el acuerdo debe ser tratado como el documento de referencia en las relaciones con el cliente en todos los aspectos de la entrega de los servicios pactados, por tanto, debe definir claramente los aspectos esenciales del servicio, como su descripción disponibilidad, niveles de calidad, tiempos de recuperación, etc.

- *Acuerdo de nivel operacional (Operation Level Agreement - OLA)*: Documento interno que especifica las responsabilidades y compromisos de varios departamentos de la organización de TI en la entrega de un servicio en particular.
- *Contrato de apoyo (Underpinning Contract - UC)*: Acuerdo con un proveedor externo para proveer servicios no cubiertos por la organización de TI.
- *Programa de mejora del servicio (Service Improvement Programme - SIP)*: Debe incluir las medidas de corrección para fallas detectadas en los niveles de servicio y propuestas de mejora basadas en desarrollos de la tecnología. Debe hacer parte de la documentación básica para la renovación del acuerdo de nivel de servicio y debe mantenerse internamente para que esté disponible a los administradores de otros procesos de TI.

2.2.2 Roles [14]

Según ITIL V3, los roles involucrados en la gestión de niveles de servicios son:

- *Gestor del Nivel de Servicio*: Propietario de Proceso, responsable de negociar Acuerdos de Nivel de Servicio y de velar que se cumplan. Debe asegurar que todos los procesos de Gestión de Servicios de TI, Acuerdos de Nivel Operacional (OLA) y Contratos de Apoyo (UC) sean adecuados para los niveles de servicio acordados. Debe monitorear e informar acerca de los niveles de servicio.
- *Arquitecto de Aplicaciones*: Responsable del diseño de aplicaciones necesarias para la prestación de un servicio, incluyendo la especificación de tecnologías, la aplicación de arquitecturas y de estructuras de datos como base para el desarrollo o la personalización de aplicaciones.
- *Analista / Arquitecto Técnico*: Se ocupa de diseñar componentes de infraestructura y sistemas necesarios para la prestación de un servicio, incluyendo la especificación de tecnologías y productos como base para su adquisición y personalización.
- *Gestor de Diseño del Servicio*: Responsable de producir diseños seguros, sólidos y de calidad para servicios nuevos o mejorados, incluyendo la producción y el mantenimiento de toda la documentación de diseños.
- *Propietario del Servicio*: Debe cumplir su responsabilidad dentro de los niveles de servicio acordado. Por lo general, actúa como contraparte del Gestor del Nivel de Servicio al negociar Acuerdos de Nivel Operacional (OLA). Líder de un equipo de especialistas técnicos o de una unidad de apoyo interno.

2.2.3 Proceso de gestión de niveles de servicio ITIL V3

La Figura 2.8 presenta las actividades desarrolladas como parte del proceso de gestión de niveles de servicio, y los artefactos afectados/generados como resultados de su ejecución. En la planeación se debe determinar, negociar, documentar y pactar requerimientos para servicios nuevos o modificados en SLRs. En la implementación se debe gestionar y revisar los requerimientos a través del ciclo de vida del servicio en base al SLA negociando acuerdos de nivel operacional y contratos de apoyo. Posteriormente se procede a monitorear y medir el rendimiento de todos los servicios operacionales respecto a los objetivos documentados en los SLAs, para generar reportes de nivel de servicio. Finalmente se dirigen revisiones de los servicios, identificando oportunidades de mejora y gestionando los SIPs adecuados [15].

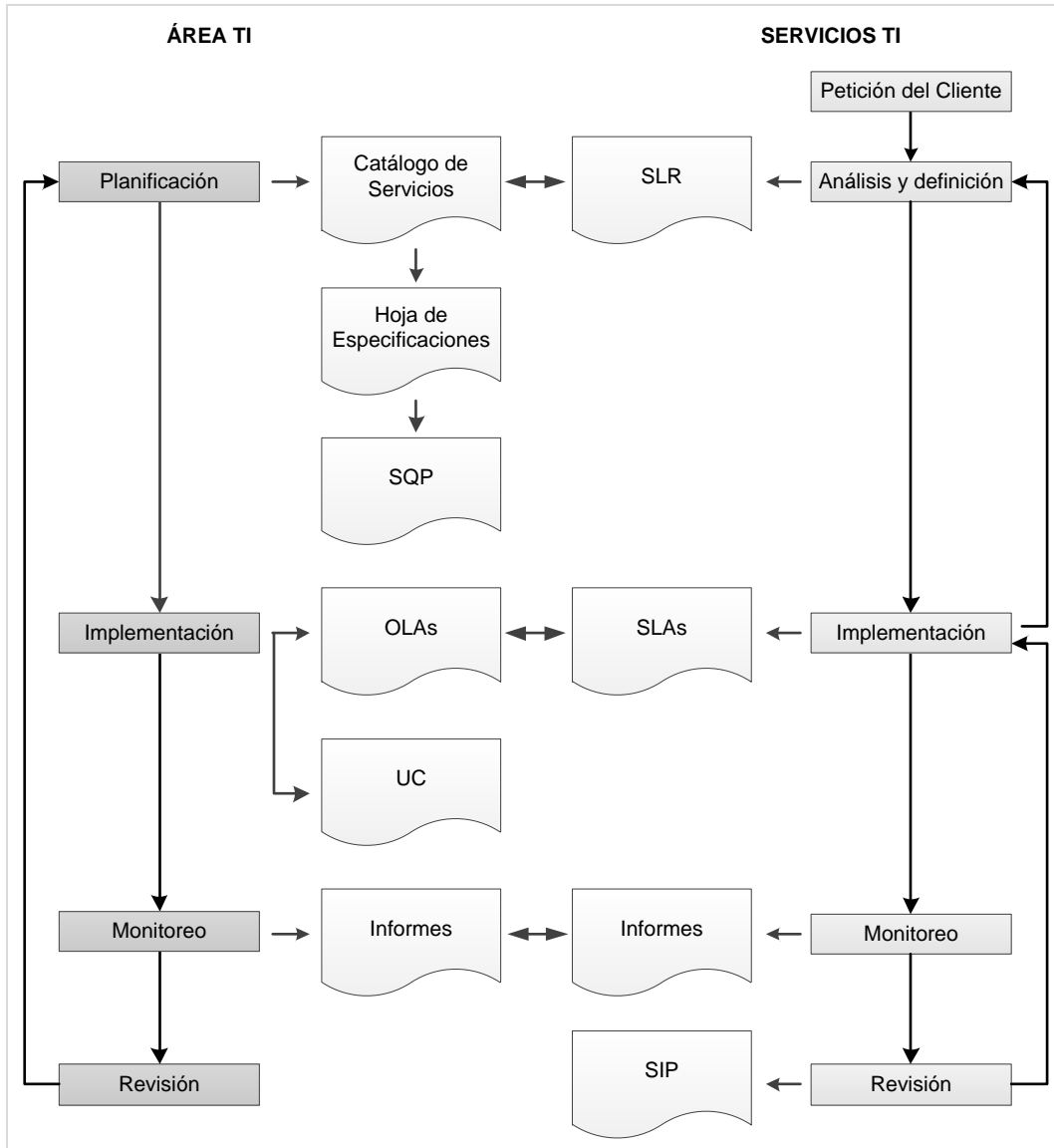


Figura 2.8. Procesos de gestión de niveles de servicio ITIL V3 (Fuente: adaptado de [12])

En el contexto de la gestión de servicios de TI, la gestión de niveles de servicio también ejecuta actividades relacionadas con las interfaces con otros procesos como: medir y mejorar la satisfacción del cliente en cooperación con el proceso de gestión de relaciones de negocio; asistir a la gestión de proveedores para revisar contratos de apoyo; asignar recursos, preparar el catálogo de servicios, desarrollar acuerdos de nivel de servicio estándares, herramientas para monitorear la calidad del servicio; analizar e identificar las necesidades del cliente; preparar requerimientos de nivel de servicio, hojas de especificación, y planes de calidad del servicio.

2.3 Estado del arte

Los marcos de trabajo introducidos aportan los conceptos para la construcción de un modelo de auditoría. De forma complementaria, a continuación se presenta un conjunto

de trabajos relacionados de otros autores, alineados conceptual o técnicamente con el enfoque seleccionado para el desarrollo de nuestra contribución.

Abram y sus colegas [16] definen un modelo de información para el modelo de medición de la serie SquaRE. La Figura 2.9 presenta el modelo de información para el modelo medición ISO 15939 [17], estándar que define un proceso de medición aplicable a sistemas de información e ingeniería de software. Por otra parte en la Figura 2.10 se presenta una abstracción de las fases del modelo de medición de la serie SQuaRE, que se mantuvo en la transición desde la versión anterior del estándar (ISO 9126). Ahí se divide el proceso de medición en recopilación, preparación y análisis de datos, transformando valores medidos de atributos de calidad en información que refleja el estado de los servicios, luego de aplicar funciones de combinación y transformación. Este trabajo aporta principalmente una visión secuencial de los conceptos de calidad de software, sin embargo, esta estrategia no considera la calidad parte de la gestión de niveles de servicio.

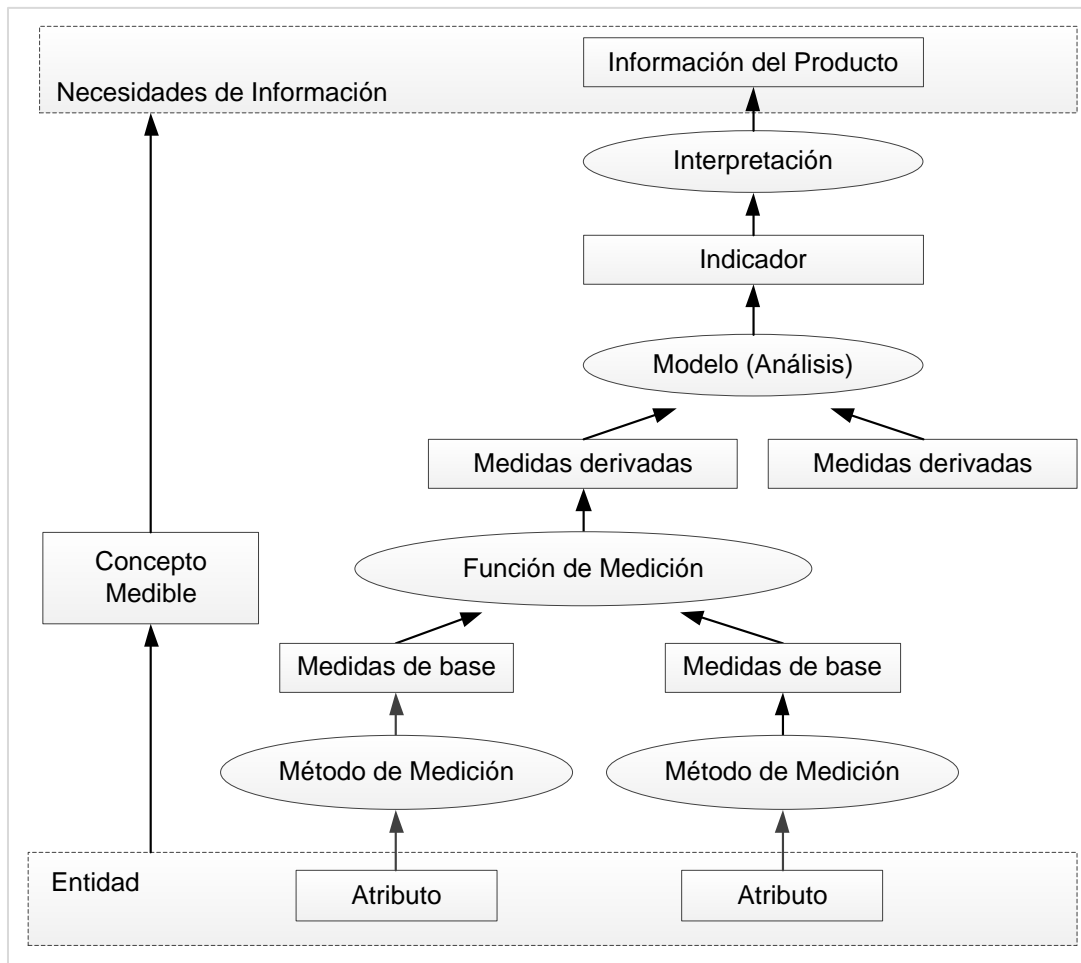


Figura 2.9. Modelo de información de medición ISO 15939 (Fuente: adaptado de [16])

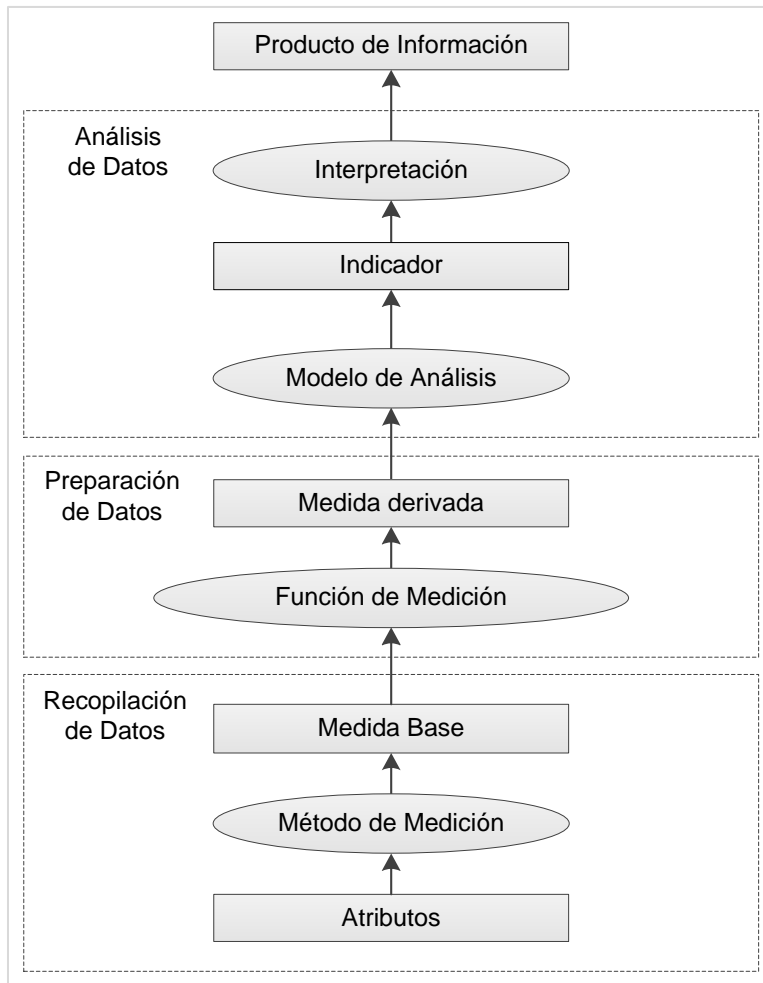


Figura 2.10. Fases del modelo de medición ISO 15939 (Fuente: adaptado de [16])

Por su parte, Wustenhoff [18] presenta la descripción del proceso de gestión de niveles de servicio en el contexto de un centro de datos, haciendo énfasis en la necesidad de herramientas de automatización de medición, y resaltando que uno de los beneficios a obtener es la mejora continua de la oferta de servicios de información. Para la definición del proceso de medición el autor emplea parcialmente las definiciones presentadas de ITIL V3 (Figura 2.11). Desde el punto de vista de la gestión de niveles de servicio, se definen acuerdos de nivel de servicios, indicadores, se capturan datos del estado de los servicios, se analizan y comunican a los interesados. La contribución de este trabajo es una visión procedimental de la gestión de niveles de servicio, sin embargo, no considera la utilización de estándares de calidad de software, al tiempo que está enfocada al monitoreo de recursos de un centro de datos.

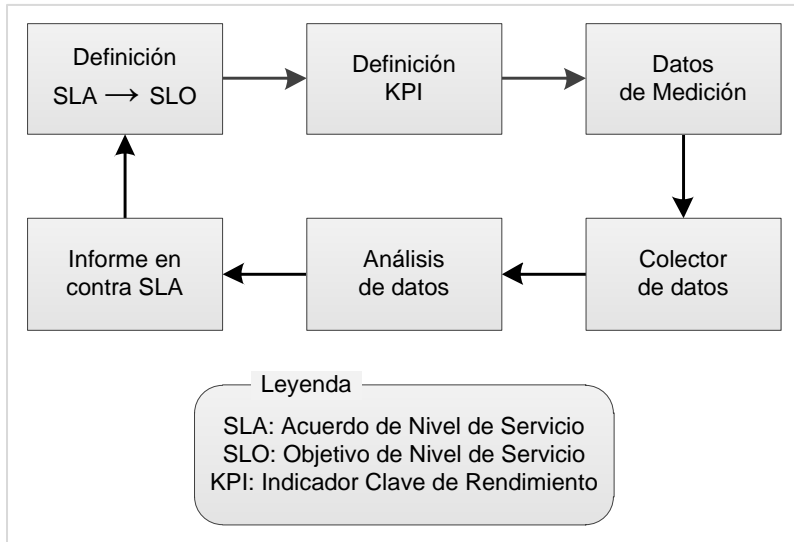


Figura 2.11. Proceso de gestión de niveles de servicio (Fuente: adaptado de [18])

Ameller y Franch [19] contribuyen al estado del arte con la propuesta de un sistema de monitoreo de acuerdos de nivel de servicio, en el contexto de aplicaciones bajo arquitectura SOA (Service Oriented Architecture) auto adaptativas, que buscan mantener los niveles de calidad especificados en los acuerdos de nivel de servicio. Las métricas empleadas en la evaluación de calidad son tomadas del estándar ISO 9126, predecesor de la serie ISO 25000. El aporte presenta limitada independencia de la tecnología dado que aborda la problemática desde la perspectiva de la gestión externa, presentando un caso de aplicación a servicios web en el contexto de arquitectura orientada a servicios. Entre las bondades de este trabajo se encuentra su visión de gestión y el uso de un estándar de calidad de software. Sin embargo, tampoco considera la visión de la gestión de niveles de servicio.

Los trabajos presentados, aunque consideran tanto la perspectivas de calidad de software y de gestión de niveles de servicio, no abordan de forma integral el problema de la gestión de niveles de servicio para sistemas de información, limitando el alcance de sus soluciones. En ocasiones también se proveen herramientas sesgadas a una tecnología, característica que limita su aplicabilidad en arquitecturas empresariales modernas, dada la heterogeneidad de los sistemas de información o carecen de una visión de gestión.

Capítulo 3

Modelo de auditoría de niveles de servicio de sistemas de información

Como parte de nuestra propuesta, integramos principios y conceptos en el marco de la gestión de los niveles de servicio y la gestión de la calidad del software. Para hacer efectivo este enfoque, seleccionamos un marco de trabajo de gestión de TI y un marco de gestión de calidad de software que contribuyen a la conceptualización de elementos del modelo. Basamos su concepción en tres conjuntos de principios: *generales, específicos de la gestión de servicios de TI y específicos de la gestión de calidad de software.*

Principios generales

- Línea base definida: Para que el modelo sea fácil de entender y de aplicar por los interesados, se requiere contar con una base conceptual sólida, derivada de marcos de trabajo con objetivos y alcance bien definidos y con flujos de información documentados que permitan obtener resultados de forma sistemática.
- Aplicabilidad probada: las organizaciones actuales presentan retos de implementación por la variabilidad de los entornos de negocio y tecnológicos, se requiere estándares de la industria depurados en el tiempo mediante su aplicación en un amplio espectro de entornos tecnológicos y de negocio.
- Soporte y actualización: La tecnología tiene una estrecha relación con el cambio constante. Para hacer perdurable la propuesta, se requiere la selección de marcos de trabajo que históricamente hayan evolucionado con las tendencias del sector para responder de manera ágil y efectiva a las nuevas necesidades.

Principios de la gestión de servicios de TI:

- Enfoque a gestión de niveles de servicio: Para diseñar un modelo con un bajo acoplamiento, que pueda ser fácilmente integrado en procesos de gestión de TI existentes, se requiere un marco de trabajo que considere la gestión de niveles de servicio una actividad esencial en el ciclo de vida de los servicios, identificando de forma precisa sus dependencias e interacciones con otros procesos o prácticas.
- Objetivos genéricos asociados a la gestión de niveles de servicio: Se requiere un marco de trabajo que tenga objetivos enfocados en obtener beneficios como eficiencia operativa, reducción de costos y satisfacción de los usuarios, indicando cómo las actividades, prácticas o procesos definidos contribuyen a alcanzarlos.
- Indicadores clave de rendimiento para servicios de información: El marco de trabajo seleccionado debe considerar indicadores genéricos de gestión de niveles de

servicio que puedan ser empleados para reflejar los beneficios obtenidos de la implementación del modelo a proponer en un modelo de causa-efecto.

Principios de la gestión de calidad de software:

- Cobertura del ciclo de vida del producto de software: La auditoría de acuerdos de nivel de servicio suele asociarse con sistemas de información en producción. En consecuencia, se requiere un marco de trabajo que esté definido de acuerdo con el ciclo de vida del producto de software, diferenciando al menos entre construcción, pruebas y producción, y ofreciendo modelos de calidad con atributos y métricas en cada caso.
- Base de conocimiento de medición de calidad: Para facilitar la implementación del modelo, se requiere un marco de trabajo que cuente con un modelo de medición establecido, con tablas de referencia para las métricas de calidad que faciliten y estandaricen la construcción de los escenarios de calidad, y la comunicación de los reportes de nivel de servicio.

La Figura 3.1 presenta la estructura general de nuestra propuesta de modelo de auditoría de sistemas de información. Tal como se ilustra, el *modelo de dominio* incluye los conceptos y relaciones para integrar las perspectivas de gestión de niveles de servicio y de gestión calidad de software. Sobre esta base conceptual, se define un *modelo operacional* que considera el conjunto de actividades requeridas para poner en producción el modelo de auditoría, definiendo responsables y automatizando las actividades siempre que es posible. El *modelo operacional* puede ser usado como referencia para evaluar la viabilidad de implantación en un ambiente tecnológico, dado que expone explícitamente las actividades y capacidades que debe tener un área de TI cuando se pretende adoptar esta propuesta. Como punto de partida y siguiendo los principios establecidos, se seleccionó ITIL V3 para todo lo referente a gestión de niveles de servicio y la serie SQuaRE para calidad de software. En la propuesta se abstraen e integran conceptos y modelos de estos dos estándares ampliamente difundidos y aceptados en la industria de TI.

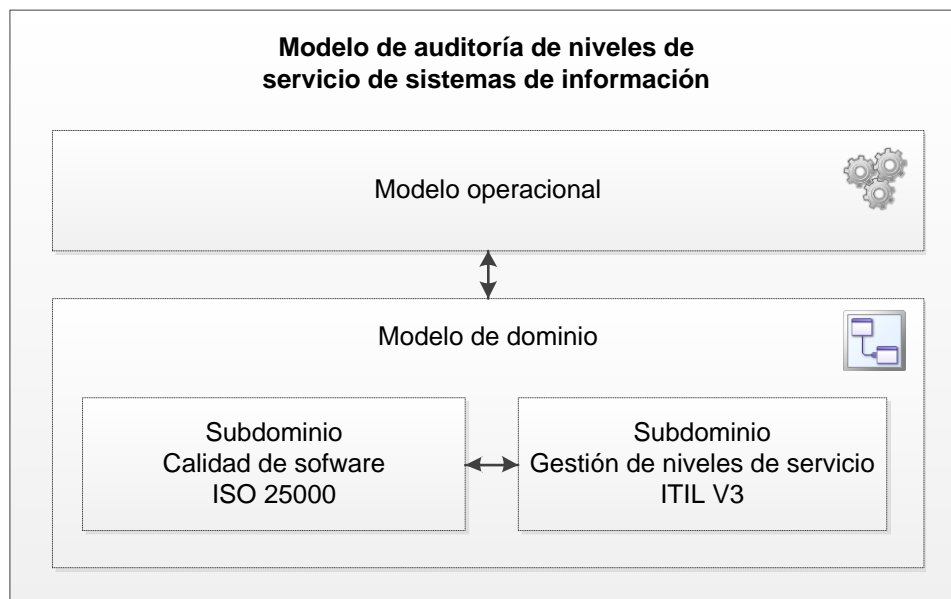


Figura 3.1. Estructura del modelo de auditoría de niveles de servicio (Fuente: propia)

3.1 Modelo de dominio

Como lo presentamos antes (ver Figura 3.1), nuestro modelo de dominio está dividido en dos subdominios, el primero concebido para soportar la configuración de modelos de calidad de software de acuerdo con la serie SQuaRE (ver Figura 3.2); el segundo concebido para soportar la auditoría de niveles de servicio de acuerdo con el subconjunto de definiciones seleccionadas de ITIL V3 (ver Figura 3.3). Los dos subdominios son integrados por medio de un elemento que permite acoplarlos de manera ordenada.

3.1.1 Subdominio de configuración de modelos de calidad SQuaRE

Como parte de nuestra propuesta, del marco de trabajo seleccionado para la gestión de calidad de software, tomamos como referencia los siguientes conceptos y sus respectivas definiciones (ver Sección 2.1.1): *atributo de calidad, medida base, medida derivada, método de medición, función de medición, elemento de medición de calidad e indicador.*

En la Figura 3.2 se puede ver cómo los modelos de calidad son clasificados de acuerdo a la fase del producto (ver sección 2.1.2.3) y cómo mediante una relación recursiva las características de calidad adquieren profundidad agrupando métricas basadas en medidas normalizadas (ver sección 2.1.2.2), que a su vez son generadas mediante la aplicación de funciones matemáticas consideradas en el modelo de medición (ver sección 2.1.2.4).

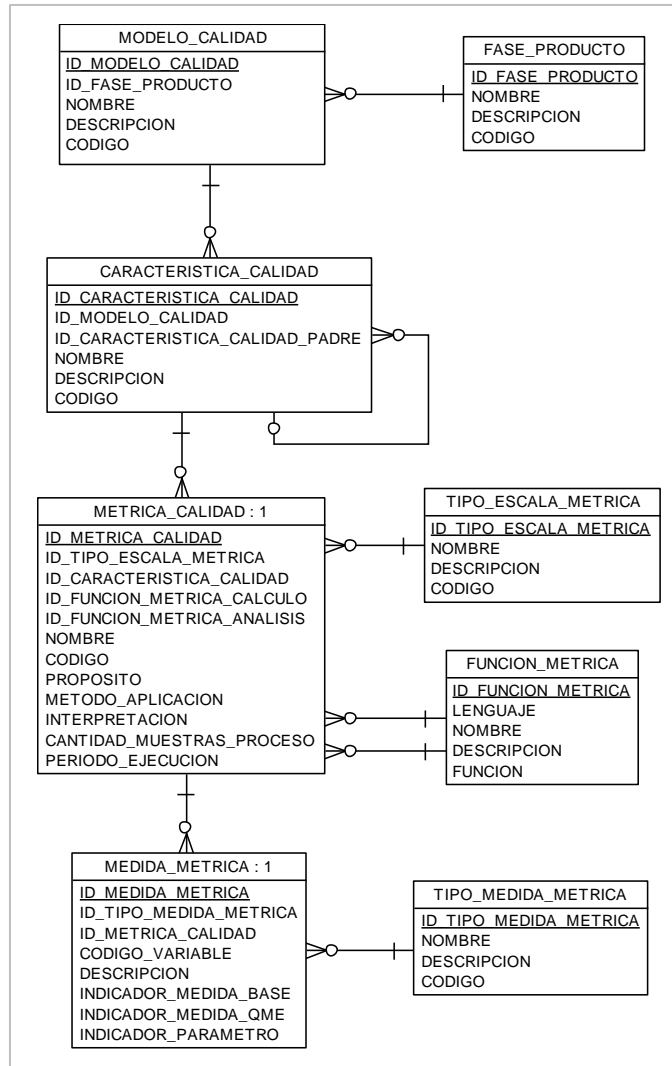


Figura 3.2. Subdominio de configuración de modelos de calidad SQuaRE (Fuente: propia)

3.1.2 Subdominio de configuración de modelos de calidad SQuaRE

Del marco de trabajo seleccionado para la gestión de servicios de TI tomamos como referencia los siguientes conceptos y sus respectivas definiciones (ver Sección 2.2.1): *cliente, proveedor, catálogo de servicios, requerimientos de nivel de servicio, hojas de especificación, programa de calidad del servicio, acuerdo de nivel de servicio, acuerdo de nivel operacional, contrato de apoyo, programa de mejora del servicio.*

En la Figura 3.3 se observa como partiendo de la base de calidad de software aportada por el subdominio anterior, aquí se agregan elementos enfocados a la gestión de niveles de servicio como la evaluación que agrupa un conjunto de datos de calidad procesados de acuerdo a la métrica para determinar el estado del servicio. Por su parte el escenario de calidad actúa como elemento integrador agrupando las evaluaciones y asociándolas al proceso de negocio y al servicio en cuestión, relación que se puede inferir de acuerdo a lo expuesto en las definiciones de *acuerdo de nivel de servicio* y de *catálogo de servicios* (ver sección 2.2.1)

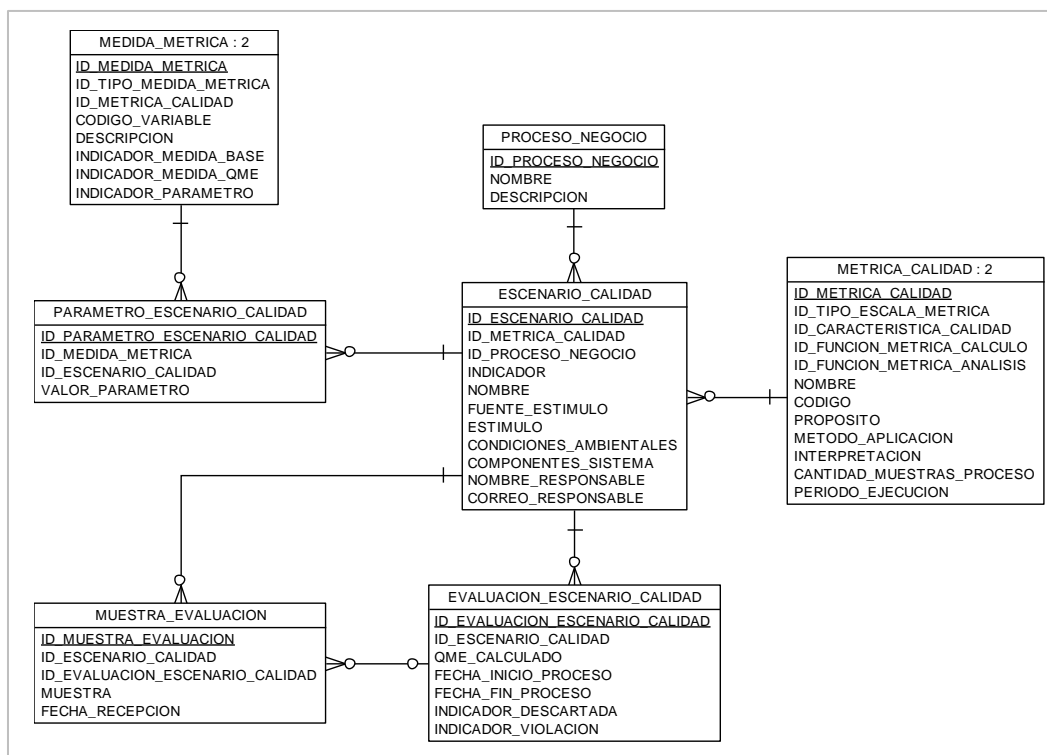


Figura 3.3. Subdominio de auditoría de niveles de servicio (Fuente: propia)

3.1.3 Escenarios de calidad: elemento de integración de subdominios

Como parte de nuestra propuesta consideramos el uso del *Escenario de Calidad* como elemento integrador. Definimos un escenario de calidad como un marco para la evaluación de calidad de un componente de un sistema con base en un *atributo de calidad*, una *métrica* y un *nivel esperado*, una vez que una fuente genera un *estímulo* que tiene efecto sobre un *componente* en un *contexto* particular. Tomamos como referencia el formato de especificación presentado en la Tabla 3.1.

Nombre	<Nombre descriptivo>	Identificador	<Código único>
Servicio	<Servicio de información afectado>		
Modelo de calidad	<Modelo de calidad de software seleccionado de acuerdo con estado del producto o según el criterio del diseñador de escenario>		
Atributo de calidad	<Atributo de calidad que se desea evaluar.>		
Métrica de calidad	<Métrica de calidad seleccionada para el atributo de calidad a evaluar>		
Cliente	<Descripción del cliente que consume el servicio.>		
Estímulo	<Fenómeno que desencadena la respuesta del servicio, puede ser una solicitud de usuario, un evento, una interrupción, un error, etc.>		
Condiciones ambientales	<Cualquier condición ambiental relevante que impacte la respuesta del servicio. Por ejemplo: carga normal, carga límite, operación degradada, en desarrollo>		
Indicador	<Valor de referencia para comparar las medidas derivadas calculadas a partir de la métrica>		

	<i>seleccionada></i>
Responsable	<i><Gestor del nivel del servicio, interesado a quien debe comunicarse el reporte de nivel de servicio y las violaciones></i>

Tabla 3.1. Formato de especificación de escenario de calidad (Fuente: propia)

Adicionalmente se proveen los siguientes conceptos unifican definiciones de los enfoques de calidad y gestión de niveles para facilitar la comprensión de nuestra propuesta:

- *Dato de nivel de servicio:* Conjunto de *medidas base* asociadas a un *escenario de calidad* entregadas por el proveedor del servicio que han sido generadas como resultado de la excitación del sistema de información en el marco de un acuerdo de nivel de servicio.
- *Información de nivel de servicio:* *Medida derivada*, resultado de aplicar la *función de medición* asociada a la *métrica de calidad* del *escenario*, a un conjunto de datos de nivel de servicio.
- *Función de análisis:* Procedimiento matemático que recibe como parámetro información de nivel de servicio para compararlo con *indicador* asociado al escenario de calidad.
- *Reporte de nivel de servicio:* Resultado de la aplicación de la función de análisis asociada al escenario de calidad, que muestra si se ha detectado una violación del nivel de servicio pactado entre cliente y proveedor.

3.2 Modelo operacional

Nuestra propuesta de modelo operacional considera el conjunto de actividades requeridas para poner en producción el modelo de auditoría; considera también las perspectivas de gestión de niveles de servicio y de gestión calidad de software, definiendo responsables y automatizando las actividades siempre que es posible. Este modelo expone explícitamente las actividades y capacidades que debe tener un área de TI cuando se pretende adoptar el modelo de auditoría propuesto.

La Figura 3.4 presenta el modelo operacional usando sintaxis BPMN2. En la Figura 3.4, la primera actividad consiste en especificar el escenario de calidad. Esta actividad es desarrollada por un usuario del sistema con habilidades para transformar especificaciones de nivel de servicio concertadas entre el cliente y el proveedor bajo un SLA en requerimientos de calidad de software bajo el modelo de escenario presentado en la Tabla 3.1. Luego de este proceso el modelo define actividades automáticas en las que se recibe y procesan los datos de calidad para generar información que se emplea para evaluar y comunicar el estado de los servicios. Las tablas Tabla 3.2, Tabla 3.3, Tabla 3.4, Tabla 3.5, Tabla 3.5 y Tabla 3.6 presentan el detalle de las actividades que involucra el modelo.

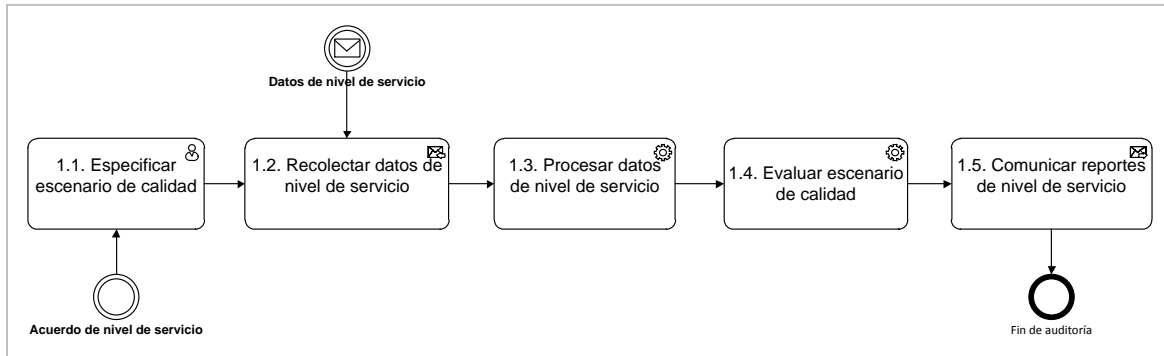


Figura 3.4. Modelo operacional de auditoría de niveles de servicio (Fuente: propia)

Actividad	Especificar escenario de calidad.	Tipo	Usuario
Entradas		Salidas	
<ul style="list-style-type: none"> ▪ Catálogo de servicios. ▪ Acuerdo de nivel de servicio nuevo o renegociado. 		<ul style="list-style-type: none"> ▪ Catálogo de servicios actualizado. 	
Tareas			
<ul style="list-style-type: none"> ▪ Documentar la información básica del escenario de calidad. Roles: Gestor de Diseño del Servicio. ▪ Seleccionar un modelo, un atributo y una métrica de calidad que se ajusten a los requerimientos de nivel del servicio, las hojas de especificación y el programa de calidad del servicio. Roles: Propietario del Servicio, Gestor de Diseño del Servicio. ▪ Relacionar el escenario de calidad en el catálogo de servicios. Comprobar la clasificación de los servicios por proceso de negocio. Roles: Gestor de Diseño del Servicio. ▪ Realizar una evaluación de viabilidad técnica del escenario de calidad para determinar si el servicio dispone de mecanismos para entregar datos de calidad que permitan monitorear su estado. Roles: Analista / Arquitecto Técnico. 			

Tabla 3.2. Especificación de actividad: Especificar escenario de calidad (Fuente propia)

Actividad	Recolectar datos de nivel de servicio.	Tipo	Servicio
Entradas		Salidas	
<ul style="list-style-type: none"> ▪ Datos de nivel de servicio entregados por el proveedor. 		<ul style="list-style-type: none"> ▪ Paquete de datos de niveles de servicio depurados. 	
Tareas			
<ul style="list-style-type: none"> ▪ Validar estructura y formato de los datos de nivel de servicio entregados por el proveedor, descartar las muestras inválidas y notificar al proveedor. Roles: No aplica (Tarea ejecutada por el sistema). ▪ Empaquetar los datos de nivel de servicio depurados y asociarlos al escenario de calidad. 			

Tabla 3.3. Especificación de actividad: Recolectar datos de nivel de servicio (Fuente propia)

Actividad	Procesar datos de nivel de servicio.	Tipo	Servicio
Entradas		Salidas	
<ul style="list-style-type: none"> ▪ Paquete de datos de nivel de servicio depurados. 		<ul style="list-style-type: none"> ▪ Evaluación de nivel de servicio. 	
Tareas			
<ul style="list-style-type: none"> ▪ Aplicar la función de medición de la métrica del escenario de calidad al conjunto de datos de nivel de servicio. Roles: No aplica (Tarea ejecutada por el sistema). ▪ Crear una evaluación de calidad con la información de nivel de servicio generada. ▪ Relacionar la evaluación de nivel de servicio al escenario de calidad en el catálogo de servicios. Roles: No aplica (Tarea ejecutada por el sistema). 			

Tabla 3.4. Especificación de actividad: Procesar datos de niveles de servicio (Fuente propia)

Actividad	Evaluar escenario de calidad.	Tipo	Servicio
Entradas		Salidas	
<ul style="list-style-type: none"> ▪ Evaluación de nivel de servicio. 		<ul style="list-style-type: none"> Reporte de niveles de servicio. 	
Tareas			
<ul style="list-style-type: none"> ▪ Aplicar la función de análisis a la información de nivel de servicio para determinar el estado del servicio en relación al indicador especificado en el escenario de calidad para generar el reporte de nivel de servicio. Roles: No aplica (Tarea ejecutada por el sistema). ▪ Relacionar el reporte de nivel de servicio al escenario de calidad en el catálogo de servicios. Roles: No aplica (Tarea ejecutada por el sistema). 			

Tabla 3.5. Especificación de actividad: Evaluar escenario de calidad (Fuente propia)

Actividad	Comunicar reporte de nivel de servicio.	Tipo	Servicio
Entradas		Salidas	
<ul style="list-style-type: none"> ▪ Reporte de nivel de servicio. 		<ul style="list-style-type: none"> ▪ Programa de mejora del servicio. ▪ Solicitudes de ajuste a contratos de apoyo. 	
Tareas			
<ul style="list-style-type: none"> ▪ Distribuir el reporte de nivel de servicio a los interesados. Roles: No aplica (Tarea ejecutada por el sistema). ▪ Analizar el reporte de nivel de servicio para identificar violaciones y generar el programa de mejora del servicio, analizar dependencias de contratos de apoyo para promover solicitudes de renegociación. Roles: Gestor del Nivel de Servicio, Arquitecto de Aplicaciones, Analista / Arquitecto Técnico, Gestor de Diseño del Servicio, Propietario del Servicio. 			

Tabla 3.6. Especificación de actividad: Comunicar reporte de nivel de servicio (Fuente propia)

3.3 Validación del modelo de auditoría de niveles de servicio de sistemas de información

Consideremos la siguiente situación: la organización A requiere varios servicios de información de la organización B, luego A actúa en calidad de *cliente* y B de *proveedor*. Siguiendo sus prácticas de gestión de niveles de servicio, para formalizar el acuerdo establecido, A y B pactan un *SLA*, donde se establece entre otras cosas que el servicio S en condiciones normales de carga, debe responder en un tiempo máximo de 4 segundos.

Para auditar la calidad del servicio S, El *gestor de diseño del servicio* documenta un *escenario de calidad* (entidad ESCENARIO_CALIDAD del subdominio de la Figura 3.3), identificando cada elemento del formato presentado en la Tabla 3.1. A selecciona la *métrica* (entidad METRICA_CALIDAD) “Peor tiempo de respuesta” del estándar ISO 25020 RRRREEEFF, que corresponde a la *característica de calidad* (entidad CARACTERISTICA_CALIDAD) *eficiencia* del *modelo de calidad* (entidad MODELO_CALIDAD) *calidad externa* en la *fase del producto* (entidad FASE_PRODUCTO) *operación*. La métrica seleccionada requiere la *medida base* (entidad MEDIDA_METRICA para el TIPO_MEDIDA *medida base*) *tiempo de respuesta*; también tendrá asignadas *funciones de procesamiento y análisis* (entidad FUNCION_METRICA) que tomarán paquetes de una sola muestra, en el primer caso será la función unitaria y en el segundo un algoritmo de comparación frente al valor máximo esperado (propiedad INDICADOR de la entidad ESCENARIO_CALIDAD) de 4 segundos.

Posteriormente el *escenario* es asociado al servicio en *catálogo de servicios* y en consecuencia al *SLA*. Luego se solicita al *proveedor* que notifique los *datos de calidad* que deben contener el *identificador de escenario* (propiedad ID_ESCENARIO_CALIDAD de la entidad MUESTRA_EVALUACION) y cada *medida base* asociada a la *métrica* seleccionada. Una vez en operación, se ejecutan de las actividades del *modelo operacional* (ver sección 3.2). En primera instancia, se alimenta el *repositorio de auditoría* con los datos entregados por el *proveedor* para armar *paquetes de evaluación* (EVALUACION_ESCENARIO_CALIDAD); luego se aplica la *función de procesamiento* de la *métrica* a cada paquete para calcular el *valor de referencia* (propiedad QME_CALCULADO de la entidad EVALUACION_ESCENARIO_CALIDAD); a continuación se toma la *función de análisis* y el *indicador* (propiedad INDICADOR de la entidad ESCENARIO_CALIDAD) para detectar *violaciones*; finalmente se comunica a los *interesados* el estado del servicio. Así concluye el proceso de auditoría, con lo que las áreas de TI de A y B, podrán promover las acciones necesarias, basadas en información precisa, oportuna y compartida, para corregir las *inconformidades* de acuerdo a las prácticas de gestión de servicios adoptadas por cada organización.

Capítulo 4

Herramienta de soporte de auditoría de niveles de servicio de sistemas de información

ITIL en fase de mejora continua del servicio considera el uso de herramientas de gestión de servicios de TI (ITSM Tool – Information Technology Service Management Tool Suite), que no solamente atienden requerimientos de gestión de niveles de servicio, sino de todo el ciclo de vida del servicio. Se dice que la mayoría de estas soluciones soportan la gestión del catálogo de servicios, el monitoreo automático de acuerdos de nivel de servicio, la identificación de violaciones y la generación de reportes [20]. Sin embargo consideran el software un elemento genérico en el contexto de las tecnologías de información y carecen de flexibilidad para adaptarse a necesidades de calidad específicas. La herramienta construida, sQMA (Software Quality Measurement and Analysis Automation), constituye el principal habilitador de la auditoría sistemática de niveles de servicio partiendo de un estándar de calidad de software que abarca todo el ciclo de vida del producto. Por su alcance, no pretende competir solución completa de herramientas de gestión de TI, busca aportar un enfoque y una herramienta de sistematización de auditoría que contribuya a los desafíos y mitigue los riesgos identificados en la práctica de la gestión de niveles de servicio.

4.1 Descripción de arquitectura

Para la descripción de arquitectura, se hace uso del modelo 4+1 vistas, de tal suerte que al emplear un lenguaje de modelado ampliamente difundido como UML, se logre un entendimiento común en los diferentes interesados.

4.1.1 Vista de casos de uso

El modelo propuesto, comprende actividades relacionadas a la configuración de modelos de calidad de software (subdominio 1) y de auditoría de niveles de servicio (subdominio 2), de acuerdo al alcance del proyecto, la especificación funcional, se centra en las interacciones del subdominio 2, de acuerdo al diagrama de casos de uso de la Figura 4.1.

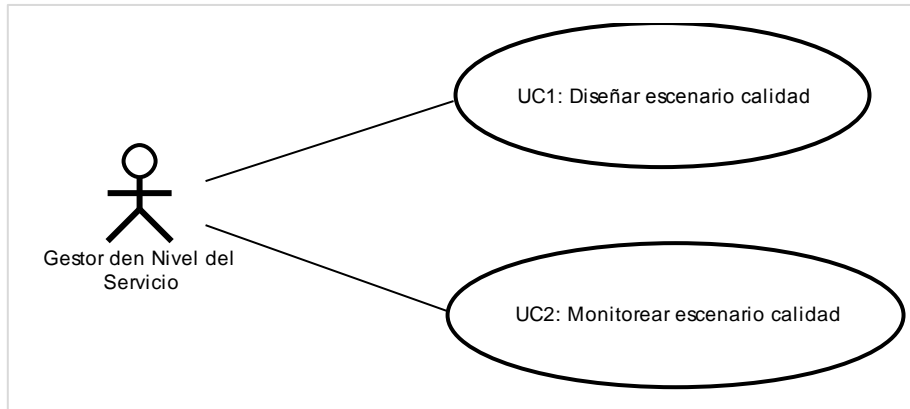


Figura 4.1. Diagrama de casos de uso de auditoría de niveles de servicio sQMA (Fuente: propia)

A continuación se presenta la especificación de los casos de uso.

Nombre	UC1: Diseñar escenario de calidad.
Actores	Gestor del Nivel de Servicio.
Requisitos	Modelo de calidad previamente configurado en el sistema.
Flujo de eventos	El Gestor del Nivel de Servicio recibe una notificación de ingreso o cambio de un acuerdo de nivel de servicio, ingresa al sistema y verifica de acuerdo al catálogo de servicios si el escenario de calidad ya existe y si debe crearse, ingresa a la interfaz de creación o edición, selecciona una característica de calidad que se adecue a su necesidad, selecciona una métrica que se ajuste a la especificación técnica del acuerdo de nivel de servicio, ingresa un valor de referencia para la evaluación de violaciones (indicador), documenta los demás parámetros del escenario, comprueba la clasificación en el catálogo de servicios y guarda la configuración.
Resultado	Se ha creado o actualizado el escenario de calidad y ha sido clasificado en el catálogo de servicios.
Flujos alternos	CA1: Asociar parámetro, el gestor del Nivel del Servicio puede agregar, modificar o eliminar parámetros al escenario de calidad, se hace necesaria esta funcionalidad porque las métricas suelen tener asociados valores de referencia necesarios al momento de ejecutar los scripts de procesamiento y análisis.

Tabla 4.1. Especificación de caso de uso UC1: Diseñar escenario de calidad (Fuente: propia)

Nombre	UC2: Monitorear escenario de calidad.
Actores	Gestor del Nivel de Servicio.
Requisitos	Modelo de calidad previamente configurado en el sistema.
Flujo de eventos	El Gestor del Nivel de Servicio ingresa a la interfaz de monitoreo, selecciona un escenario de calidad y puede ver en tiempo real las evaluaciones identificando cuales fueron violaciones.

Resultado	El Gestor del Nivel de Servicio accede al listado de evaluaciones asociadas al escenario de calidad.
Flujos alternos	NA.

Tabla 4.2. Especificación de caso de uso UC2: Monitorear escenario de calidad (Fuente: propia)

La Tabla 4.3 presenta un conjunto de requerimientos de alto nivel para el modelo operacional de auditoría de niveles de servicio.

ID	Requerimiento
RF01	El sistema permite asociar scripts de procesamiento/análisis externos y reutilizables que se cargan y ejecutan dinámicamente permitiendo la extensibilidad del sistema.
RF02	El sistema provee un servicio web para recolectar las muestras de los escenarios de calidad.
RF03	El sistema permite reportar información de diferentes escenarios de calidad en un mismo paquete de muestras y por varios clientes de forma simultánea empleando una estructura con formato estándar.
RF04	El sistema valida la estructura de la muestra, verificando que el identificador del escenario de calidad sea válido y que la muestra contenga las variables de medida base requeridas por la métrica.
RF05	El sistema descarta las muestras inválidas y encola las correctas por escenario hasta completar la cantidad necesaria para la evaluación.
RF06	El sistema almacena la evaluación y las muestras asociadas.
RF07	El sistema calcula y almacena los QME a partir de un conjunto de muestras y el script de procesamiento asociado a la métrica del escenario.
RF08	El sistema identifica violaciones a partir del QME de referencia, el QME calculado y la función de análisis asociada a la métrica.
RF09	El sistema notifica vía correo electrónico las violaciones detectadas al responsable del escenario.

Tabla 4.3. Requerimientos de alto nivel para el modelo operacional. (Fuente: propia)

En la Tabla 4.4 se puede apreciar el mapeo entre los procesos del modelo operacional, los casos de uso y los requerimientos de alto nivel.

Proceso	Requerimientos funcionales
1.1. Especificar escenario de calidad.	UC1.
1.2. Recolectar datos de nivel de servicio.	RF01, RF02, RF03, RF04, RF05, RF06.
1.3. Procesar datos de nivel de servicio.	RF07.
1.4. Evaluar escenario de calidad.	RF08.
1.5. Comunicar reporte de nivel de servicio.	UC2, RF09.

Tabla 4.4. Mapeo entre procesos del modelo operacional y requerimientos funcionales. (Fuente: propia)

Aun cuando en la definición del modelo de auditoría se consideró la gestión modelos de calidad, se toma como referencia la descomposición basada en el ciclo de vida del producto propuesta en ISO 25000, omitiendo la especificación de este tipo

requerimientos, dado que el estándar aporta una base de conocimiento de calidad bastante completa.

4.1.2 Vista lógica

La Figura 4.2 presenta el diagrama de paquetes, agrupando clases por finalidad y estableciendo las dependencias.

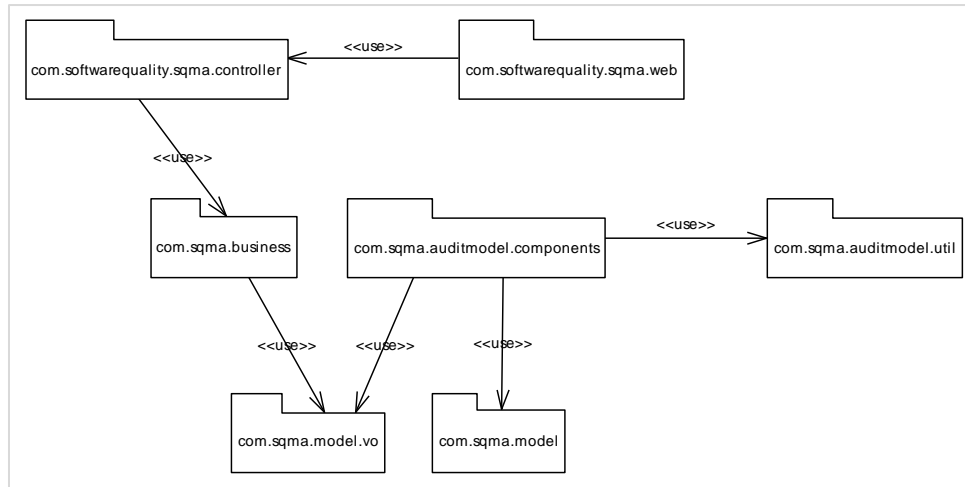


Figura 4.2. Diagrama de paquetes sQMA (Fuente: propia)

A continuación se presenta una descripción de cada uno de los paquetes mencionados.

Paquete	Descripción
com.sqma.model	Contiene las clases mapeadas del modelo relacional de los subdominios del modelo de auditoría.
com.sqma.model.vo	Contiene las clases creadas para empaquetar las muestras reportadas por el proveedor del servicio.
com.sqma.business	Contiene las interfaces e implementaciones de los servicios asociados a interacciones de usuario.
com.sqma.auditmodel.components	Contiene los componentes del modelo operacional que realizan procesos del sistema como el colector, el procesador y el evaluador.
com.sqma.auditmodel.util	Contiene clases de utilidad para diferentes capas, además de las excepciones.
com.softwarequality.sqma.controller	Contiene los controladores de presentación.
com.softwarequality.sqma.web	Contiene los componentes de interfaz de usuario, además de los recursos multimedia y de estilo requeridos.

Tabla 4.5. Descripción de paquetes sQMA (Fuente: propia)

La Figura 4.3 presenta las clases mapeadas de los subdominios del modelo de auditoría y sus relaciones.

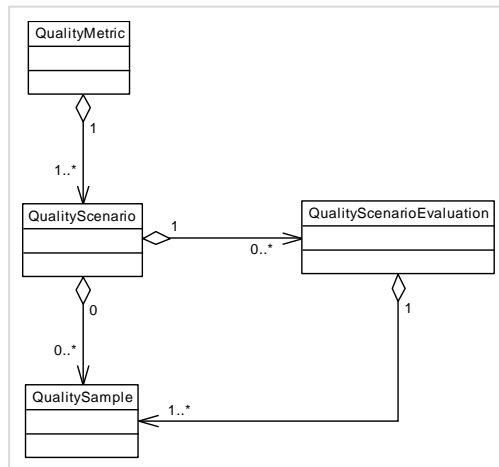


Figura 4.3. Diagrama de clases de dominio SQMA (Fuente: propia)

La Figura 4.4 presenta las clases e interfaces de los componentes del sistema que intervienen en los procesos de recolección, procesamiento y análisis de datos de calidad. Las funcionalidades fueron generalizadas mediante las interfaces Collector, Store, Processor y Evaluator. Sus implementaciones QualityEvaluationSampleCollector, QualityAuditStore, QualityEvaluationSampleProcessor y QualityScenarioEvaluator respectivamente, implementan los servicios empleando tecnologías JAVA EE.

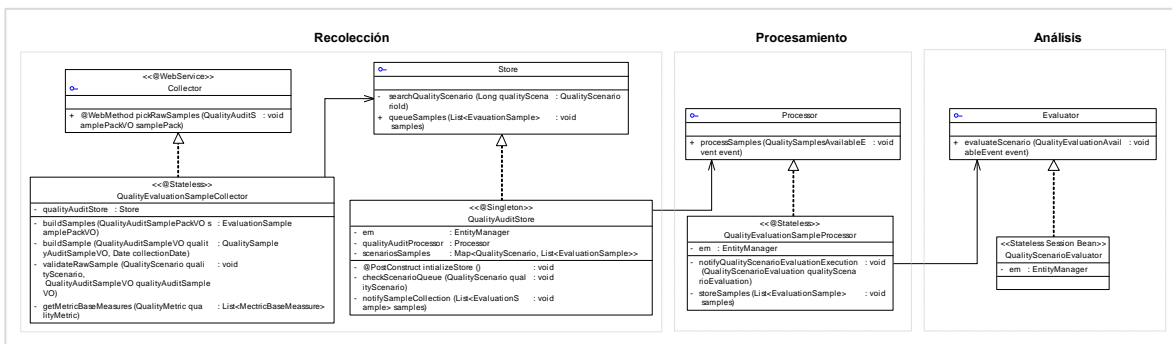


Figura 4.4. Diagrama de clases de componentes de auditoría sQMA (Fuente: propia)

La Figura 4.5 presenta las clases basadas en el patrón View Object empleadas para encapsular los datos de calidad entregados por el proveedor del servicio, que posteriormente serán extraídos para ser validados y procesados.

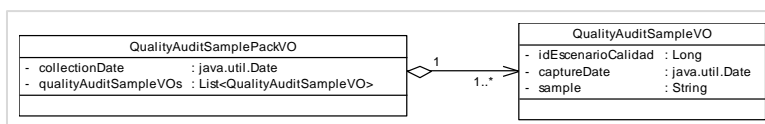


Figura 4.5. Diagrama de clases de objetos de transferencia para recolección sQMA (Fuente: propia)

4.1.3 Vista de procesos

La Figura 4.6 presenta el diagrama de secuencia para el CU: Evaluar escenario de calidad, donde se puede evidenciar como instancias de las clases de recolección, procesamiento y análisis interactúan para completar una evaluación, apoyándose en

utilidades de invocación de los scripts de las métricas y de notificación por correo electrónico.

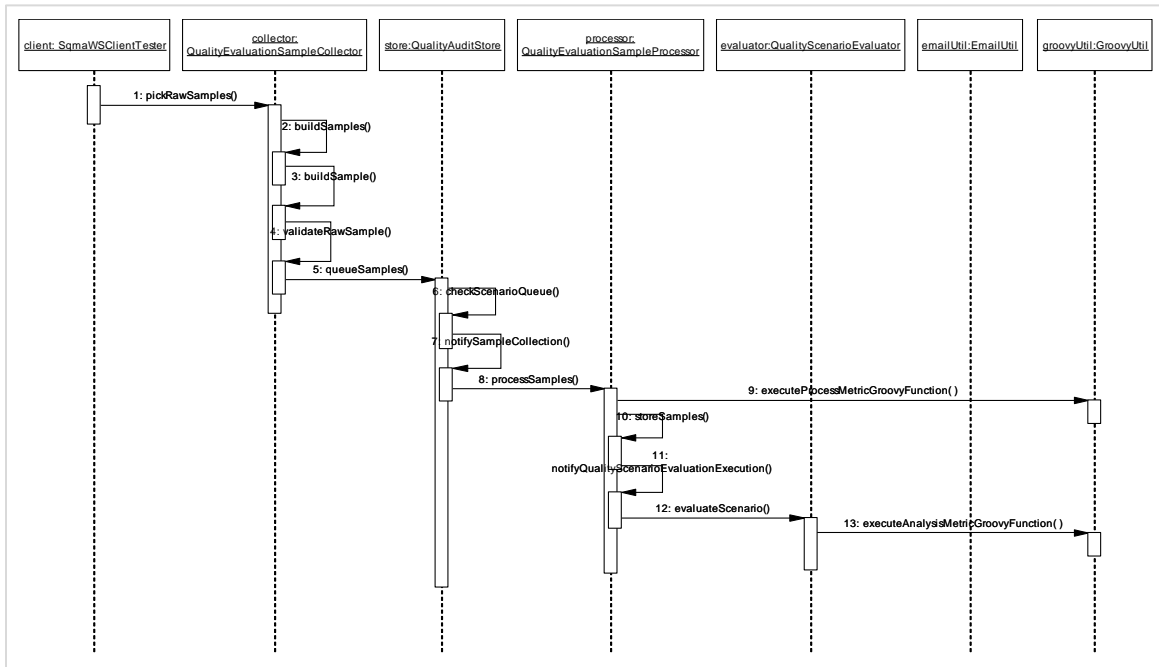


Figura 4.6. Diagrama de secuencia de auditoría de niveles de servicio sQMA (Fuente: propia)

4.1.4 Vista de implementación

La Figura 4.7 corresponde al diagrama de componentes de recolección, procesamiento y análisis del sistema desacoplados con interfaces que exponen funcionalidades. El componente QualityEvaluationSampleCollector implementa el patrón Web Service Broker para exponer su funcionalidad a quienes reportan los datos de calidad empleando un estándar de interoperabilidad.

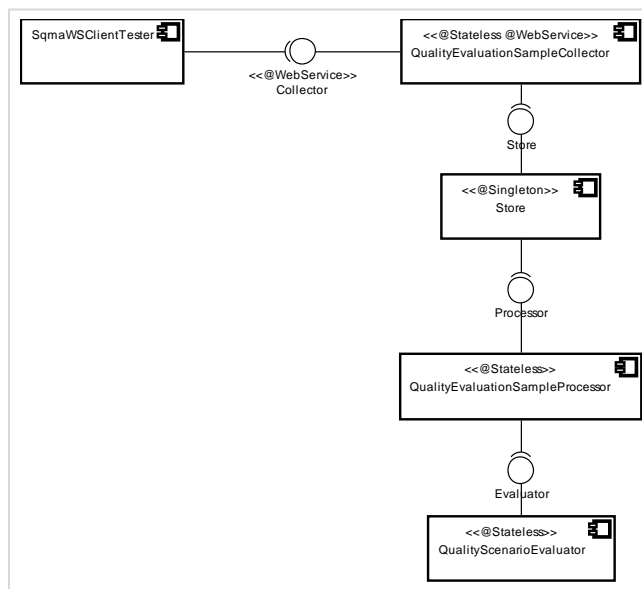


Figura 4.7. Diagrama de componentes de auditoría sQMA (Fuente: propia)

4.1.5 Vista de despliegue

La Figura 4.8 presenta el diagrama de despliegue, donde los diferentes artefactos implementados son mapeados sobre nodos de plataforma y hardware, interconectados por protocolos de comunicación. En un despliegue operativo el cliente `sqma-ws-client.jar`, empleado como artefacto de simulación es reemplazado por uno o más proveedores que notifican a la herramienta sQMA el estado de sus servicios empleando la interfaz de servicio web.

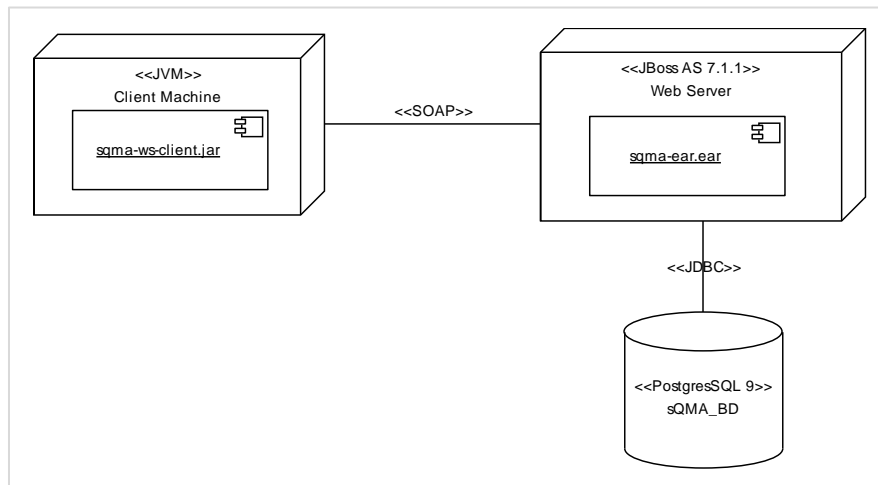


Figura 4.8. Diagrama de despliegue sQMA (Fuente: propia)

4.2 Implementación de referencia

Basada en especificación de la sección anterior, se construyó la herramienta sQMA, que satisface parcialmente los requerimientos expuestos en la sección 4.1.1, se seleccionaron los aspectos más relevantes para la validar tanto el modelo de dominio como el modelo operacional. Así, su objetivo es inicialmente soportar la configuración de modelos de calidad de software, la definición de escenarios de calidad clasificados en el catálogo de servicios y su evaluación en tiempo real siguiendo el modelo operacional (Requerimientos UC1, UC2, RF01 a RF09, algunos sin interfaz de usuario).

Una vez desplegada la herramienta, se habilita el servicio web para recibir los datos de calidad de los escenarios configurados, de acuerdo al WSDL (Web Service Description Language) de la Figura 4.9, apoyado en las definiciones XSD (XML Schema Definition) de la Figura 4.10. La estructura de datos utilizada incluye el identificador de escenario y el listado de muestras en formato JSON, con pares de medidas base de calidad de acuerdo a la métrica seleccionada. El sistema está en capacidad de recibir datos de calidad para diferentes escenarios de forma concurrente. Posteriormente cada conjunto de muestras recibido es serializado y analizado para determinar si el escenario de calidad referenciado existe y si la estructura de la muestra es correcta en función de la métrica. Las muestras inválidas son descartadas y las correctas encoladas. Cuando se completa la cantidad mínima de muestras, se crea una evaluación, se carga y ejecuta la función de procesamiento de la métrica para generar información de calidad que se compara con el valor de referencia empleando la función de análisis para detectar violaciones. Finalmente se comunica el reporte de niveles de servicio a los interesados, completando el ciclo de auditoría definido en el modelo propuesto.

```

Este fichero XML no parece tener ninguna información de estilo asociada. Se muestra debajo el árbol del documento.

<?xml version="1.0" encoding="UTF-8" standalone="no" ?>
<wsdl:definitions name="QualityAuditCollector" targetNamespace="http://components.auditmodel.sqma.com/">
  <wsdl:import location="http://127.0.0.1:8080/sqma-ejb/QualityAuditCollector/QualityAuditCollector?wsdl=Collector.wsdl" namespace="http://interfaces.auditmodel.sqma.com/"></wsdl:import>
  <wsdl:binding name="QualityAuditCollectorSoapBinding" type="ns1:Collector">
    <soap:binding style="document" transport="http://schemas.xmlsoap.org/soap/http"/>
    <wsdl:operation name="pickRawSamples">
      <soap:operation soapAction="" style="document"/>
      <wsdl:input name="pickRawSamples">
        <soap:body use="literal"/>
      </wsdl:input>
      <wsdl:output name="pickRawSamplesResponse">
        <soap:body use="literal"/>
      </wsdl:output>
    </wsdl:operation>
  </wsdl:binding>
  <wsdl:service name="QualityAuditCollector">
    <wsdl:port binding="ns1:QualityAuditCollectorSoapBinding" name="QualityAuditCollectorPort">
      <soap:address location="http://camilo-pc:8080/sqma-ejb/QualityAuditCollector/QualityAuditCollector"/>
    </wsdl:port>
  </wsdl:service>
</wsdl:definitions>

```

Figura 4.9. WSDL Servicio Colector de datos de calidad sQMA (Fuente: propia)

```

Este fichero XML no parece tener ninguna información de estilo asociada. Se muestra debajo el árbol del documento.

<?xml version="1.0" encoding="UTF-8" standalone="no" ?>
<xsd:definitions name="Collector" targetNamespace="http://interfaces.auditmodel.sqma.com/">
  <xsd:types>
    <xsd:schema elementFormDefault="unqualified" targetNamespace="http://interfaces.auditmodel.sqma.com/" version="1.0">
      <xsd:element name="pickRawSamples" type="tns:pickRawSamples"/>
      <xsd:element name="pickRawSamplesResponse" type="tns:pickRawSamplesResponse"/>
    </xsd:schema>
    <xsd:complexType name="pickRawSamples">
      <xsd:sequence>
        <xsd:element minOccurs="0" name="arg0" type="tns:qualityAuditSamplePackVO"/>
      </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="qualityAuditSamplePackVO">
      <xsd:sequence>
        <xsd:element minOccurs="0" name="collectionDate" type="xsd:dateTime"/>
        <xsd:element maxOccurs="unbounded" minOccurs="0" name="qualityAuditSampleVOs" nillable="true" type="tns:qualityAuditSampleVO"/>
      </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="qualityAuditSampleVO">
      <xsd:sequence>
        <xsd:element minOccurs="0" name="captureDate" type="xsd:dateTime"/>
        <xsd:element minOccurs="0" name="idEscenarioCalidad" type="xsd:long"/>
        <xsd:element minOccurs="0" name="sample" type="xsd:string"/>
      </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="pickRawSamplesResponse">
      <xsd:sequence>
        <xsd:complexType>
          <xsd:schema>
            <wsdl:types>
              <wsdl:message name="pickRawSamples">
                <wsdl:part element="ns1:pickRawSamples" name="parameters"/>
              </wsdl:message>
              <wsdl:message name="pickRawSamplesResponse">
                <wsdl:part element="ns1:pickRawSamplesResponse" name="parameters"/>
              </wsdl:message>
            </wsdl:types>
          </xsd:schema>
        </xsd:complexType>
      </xsd:sequence>
    </xsd:complexType>
  </xsd:types>
  <wsdl:portType name="Collector">
    <wsdl:operation name="pickRawSamples">
      <wsdl:input message="ns1:pickRawSamples" name="pickRawSamples"/>
      <wsdl:output message="ns1:pickRawSamplesResponse" name="pickRawSamplesResponse"/>
    </wsdl:operation>
  </wsdl:portType>
</xsd:definitions>

```

Figura 4.10. Esquemas XSD de transfer objects del servicio de recolección de datos de calidad sQMA (Fuente: propia)

La Figura 4.11 y la Figura 4.12 presentan las interfaces de usuario para monitorear los niveles de servicio del conjunto de escenarios de calidad identificando las violaciones respecto al indicador.

The screenshot shows a web browser window with the URL `127.0.0.1:8080/sqma-web/qualityScenarios.jdf`. The page title is "Quality Scenarios". Below the title is a table with the following data:

Name	Metric	Reference QME	Stimulus source	Stimulus	Evaluations
Sample quality scenario 1: Response time for a daily sales report	Response time	3.000000	Sample user 1	Sample user 1 generates the daily sales report	Evaluations

At the bottom of the page, there is a small footer that reads "sQMA - Software Quality Measurement and Analysis Automation".

Figura 4.11. Interfaz de usuario lista de escenarios de calidad (Fuente: propia)

#	Start time	End time	QME calculation span	Calculated QME
1	24/03/2014 7:43:29 AM	24/03/2014 7:43:30 AM	488	2.000000
2	24/03/2014 7:43:30 AM	24/03/2014 7:43:30 AM	30	2.000000
3	24/03/2014 7:43:30 AM	24/03/2014 7:43:30 AM	27	1.000000
4	24/03/2014 7:43:30 AM	24/03/2014 7:43:30 AM	28	4.000000
5	24/03/2014 7:43:30 AM	24/03/2014 7:43:30 AM	26	2.000000
6	24/03/2014 7:43:30 AM	24/03/2014 7:43:30 AM	25	3.000000
7	24/03/2014 7:43:30 AM	24/03/2014 7:43:30 AM	28	1.000000
8	24/03/2014 7:43:30 AM	24/03/2014 7:43:30 AM	25	3.000000
9	24/03/2014 7:43:30 AM	24/03/2014 7:43:30 AM	23	1.000000
10	24/03/2014 7:43:30 AM	24/03/2014 7:43:30 AM	22	4.000000
11	24/03/2014 7:43:30 AM	24/03/2014 7:43:30 AM	21	1.000000
12	24/03/2014 7:43:30 AM	24/03/2014 7:43:30 AM	22	4.000000
13	24/03/2014 7:43:30 AM	24/03/2014 7:43:30 AM	22	2.000000
14	24/03/2014 7:43:30 AM	24/03/2014 7:43:30 AM	26	4.000000
15	24/03/2014 7:43:31 AM	24/03/2014 7:43:31 AM	21	3.000000
16	24/03/2014 7:43:31 AM	24/03/2014 7:43:31 AM	23	4.000000
17	24/03/2014 7:43:31 AM	24/03/2014 7:43:31 AM	22	4.000000
18	24/03/2014 7:43:31 AM	24/03/2014 7:43:31 AM	28	1.000000
19	24/03/2014 7:43:31 AM	24/03/2014 7:43:31 AM	25	1.000000
20	24/03/2014 7:43:31 AM	24/03/2014 7:43:31 AM	23	2.000000
21	24/03/2014 7:43:31 AM	24/03/2014 7:43:31 AM	23	4.000000
22	24/03/2014 7:43:31 AM	24/03/2014 7:43:31 AM	20	2.000000
23	24/03/2014 7:43:31 AM	24/03/2014 7:43:31 AM	20	1.000000
24	24/03/2014 7:43:31 AM	24/03/2014 7:43:31 AM	20	1.000000
25	24/03/2014 7:43:31 AM	24/03/2014 7:43:31 AM	20	3.000000
26	24/03/2014 7:43:31 AM	24/03/2014 7:43:31 AM	19	2.000000
27	24/03/2014 7:43:31 AM	24/03/2014 7:43:31 AM	21	4.000000
28	24/03/2014 7:43:31 AM	24/03/2014 7:43:31 AM	20	3.000000
29	24/03/2014 7:43:31 AM	24/03/2014 7:43:31 AM	22	3.000000
30	24/03/2014 7:43:31 AM	24/03/2014 7:43:31 AM	21	1.000000
31	24/03/2014 7:43:31 AM	24/03/2014 7:43:31 AM	25	4.000000
32	24/03/2014 7:43:31 AM	24/03/2014 7:43:31 AM	23	2.000000

Figura 4.12. Interfaz de usuario de evaluaciones de escenario de calidad sQMA (Fuente: propia)

La Figura 4.13 presenta a manera de referencia, la implementación de la función de procesamiento para el tiempo de respuesta implementada en el lenguaje de programación Groovy.

```

1 class Sample{
2   def Ti
3 }
4
5 def samples = []
6 def slurper
7 def parseResult
8
9 for (rawSample in rawSamples) {
10  slurper = new groovy.json.JsonSlurper()
11  parseResult = slurper.parseText(rawSample)
12  samples.add(new Sample(Ti:parseResult.Ti))
13 }
14
15 return samples.size()==1 ? new java.math.BigDecimal(samples.get(0).Ti) : null

```

Figura 4.13. Función Groovy de procesamiento para métrica de tiempo de respuesta (Fuente: propia)

4.2.1 Decisiones de arquitectura

Con base en los requerimientos funcionales y no funcionales, se tomaron las siguientes decisiones de arquitectura enfocadas a disminuir el tiempo de construcción y a asegurar la calidad del producto:

- Se seleccionó el formato JSON para el intercambio de información de las muestras presenta una tasa de información mayor a XML, no se requieren tipos complejos, solo pares de variable valor para las medidas base de la métrica.
- Se seleccionó un servicio web SOAP () para la especificación de la función de recolección de muestras para favorecer la estandarización de los clientes que reportan muestras de calidad. Se simplifica la generación automática de clientes con base en el descriptor WSDL del servicio. El servicio web se construyó a partir de un componente @Stateless de EJB 3.1 anotado como @WebService.
- Se seleccionó un componente @Singleton de EJB 3.1 para la cola de muestras porque el contenedor gestiona su ciclo de vida de forma simplificada y es fácil de inyectar. Se sincronizan bloques de código, empleando como lock el escenario de calidad, para garantizar la consistencia de las colas en operaciones de lectura y escritura concurrente de múltiples clientes.
- Se seleccionó el lenguaje scripting Groovy para permitir la externalización de los scripts de procesamiento y análisis de las métricas debido a que es un lenguaje basado en Java que tiene diferentes mecanismos de integración, además de una sintaxis simplificada ampliamente documentada con librerías para cálculos matemáticos y procesamiento de archivos con formato JSON.
- Para la presentación se seleccionó la suite de componentes PrimeFaces 4 por eficiencia en el procesamiento de peticiones AJAX y facilidad de implementación.

4.2.2 Resumen de la pila de tecnologías

Conforme a las decisiones de arquitectura, fue seleccionado un conjunto de tecnologías para la implementación de referencia, entre estas, varias de las especificaciones de la plataforma empresarial JAVA para desarrollar componentes de persistencia, negocio, presentación e inyección de dependencias, desplegadas en un servidor de código abierto certificado para la versión JAVA EE 6. A continuación se presenta una lista de las principales tecnologías:

- Java EE 6
 - ✓ EJB 3.1
 - ✓ CDI
 - ✓ JPA 2: Hibernate 4
 - ✓ JSF 2: Primefaces 4
 - ✓ JAXWS
- JBoss AS 7.1.1
- PostgreSQL 9
- Groovy 2.2
- JSON

4.3 Validación de la herramienta sQMA

Para validar la herramienta, debido a restricciones de tiempo y alcance del proyecto, se decidió ejecutar un conjunto de casos de prueba funcionales en un ambiente controlado. Decisión que no tiene un impacto significativo en la cobertura de la validación respecto a un escenario real. El análisis y la evaluación de los casos de prueba se basan en tres (3) criterios, que buscan asegurar la integridad de la herramienta frente al modelo propuesto:

- La herramienta soporta el modelo de dominio para la parametrización parcial de al menos un modelo de calidad de la serie SQuaRE.

- La herramienta soporta el modelo de dominio para la especificación de escenarios de calidad derivados de los acuerdos de nivel de servicio.
- La herramienta soporta el modelo operacional de acuerdo al ciclo de auditoría de acuerdos de nivel de servicio.

Cada caso de prueba comprende las siguientes actividades:

- *Definición de escenario de calidad a partir del acuerdo de nivel de servicio:* Identificación del servicio; identificación de fuente del estímulo; identificación de estímulo; selección de modelo y métrica de calidad; definición de valor de referencia; definición de cantidad de muestras por evaluación; identificación de interesados; identificación de contexto de operación.
- *Configuración del ambiente:* Configuración de la métrica seleccionada para el escenario de calidad en la herramienta sQMA; registro del escenario de calidad en la herramienta sQMA; configuración del cliente de simulación con el formato de medidas base asociadas a la métrica seleccionada, el identificador generado para el escenario y los parámetros de carga: cantidad de muestras a generar y tamaño del paquete de muestras; puesta en operación de la herramienta sQMA
- *Ejecución de la prueba:* Ejecución del cliente de simulación.
- *Análisis de resultados:* Valoración de cada caso de prueba de acuerdo a los criterios establecidos.

4.3.1 Caso de prueba funcional 1: Eficiencia - Peor tiempo de respuesta

La Tabla 4.6 presenta la documentación del escenario de calidad para el caso de prueba 1.

Nombre	Peor tiempo de respuesta Servicio 1.	Id	CP-1
Servicio	Servicio 1.		
Modelo de calidad	Calidad externa ISO 25000.		
Atributo de calidad	Eficiencia.		
Métrica de calidad	Peor tiempo de respuesta.		
Cliente	Cliente 1.		
Estímulo	El cliente 1 hace una petición al Servicio 1		
Condiciones ambientales	Sistema desplegado y en condiciones normales de operación.		
Indicador	400 milisegundos.		
Responsable	camilo.palacios@hotmail.com		

Tabla 4.6. Escenario de calidad para caso de prueba 1 (Fuente: propia)

La Tabla 4.7 contiene los parámetros de configuración del cliente de simulación.

Parámetro	Valor
Cantidad de clientes concurrentes	5
Cantidad total de muestras por cliente	1500
Cantidad de muestras por paquete	100
Tiempo base	250ms
Rango de desplazamiento aleatorio	100ms - 250ms
Plantilla de la muestra formato JSON	{"Ti":\${tiempo_aleatorio}}

Tabla 4.7. Configuración del cliente de simulación para caso de prueba 1 (Fuente: propia)

El caso de prueba se ejecutó correctamente en un tiempo aproximado de 7 minutos y 30 segundos, con un total de 1348 violaciones que fueron notificadas al interesado registrado en el escenario de calidad.

4.3.2 Caso de prueba funcional 2: Recuperabilidad - Disponibilidad

La Tabla 4.8 presenta la documentación del escenario de calidad para el caso de prueba 2.

Nombre	Disponibilidad Servicio 1.	Id	CP-2
Servicio	Servicio 1.		
Modelo de calidad	Calidad externa ISO 25000.		
Atributo de calidad	Recuperabilidad.		
Métrica de calidad	Disponibilidad.		
Cliente	Cliente 1.		
Estímulo	El cliente 1 hace peticiones al Servicio 1 durante un periodo de tiempo de 1 hora.		
Condiciones ambientales	Sistema desplegado y en condiciones normales de operación.		
Indicador	Disponible.		
Responsable	camilo.palacios@hotmail.com		

Tabla 4.8. Escenario de calidad para caso de prueba 2 (Fuente: propia)

La Tabla 4.9 contiene los parámetros de configuración del cliente de simulación.

Parámetro	Valor
Cantidad de clientes concurrentes	10
Cantidad total de muestras por cliente	1000
Cantidad de muestras por paquete	1000
Estado base	Disponible
Rango de desplazamiento aleatorio	Disponible/No disponible.
Plantilla de la muestra formato JSON	{"Estado":\${estado_aleatorio}}

Tabla 4.9. Configuración del cliente de simulación para caso de prueba 2 (Fuente: propia)

El caso de prueba se ejecutó correctamente en un tiempo aproximado de 10 minutos y 15 segundos, con un total de 200 violaciones que fueron notificadas al interesado registrado en el escenario de calidad.

Capítulo 5

Conclusiones y trabajo futuro

5.1 Conclusiones

5.1.1 Conclusiones del modelo de auditoría de niveles de servicio de sistemas de información

El modelo de dominio, compuesto por un subdominio de calidad de software diseñado a partir del metamodelo de calidad de software ISO 25000 y por el subdominio de auditoría de acuerdos de nivel de servicio basado en la práctica de ITIL V3, constituye la base para soportar la evaluación de escenarios de calidad, diseñados a partir de acuerdos de nivel de servicio, de forma extensible y estandarizada, con una base de modelos de calidad que comprende todo el ciclo de vida del producto de software y que también puede extenderse para implementar características y métricas a la medida del evaluador.

Basado en el modelo de auditoría ideado, con base en los conceptos y definiciones de proceso tanto de ITIL V3 en el contexto de la gestión de niveles de servicio como de ISO 25000 en el contexto de la gestión de calidad software, se prueba que ambos pueden coexistir para aportar elementos de estandarización en las métricas empleadas en las evaluaciones. Por su parte, el enfoque que aporta ITIL V3 para la gestión de niveles de servicios, es complejo y requiere un área de TI especializada, consciente de los beneficios que se pueden alcanzar. El modelo de auditoría propuesto, surge como una alternativa de simplificación y automatización que ofrece beneficios en la disminución de las responsabilidades derivadas de procesos manuales, al tiempo ofrece una perspectiva transversal del catálogo de servicios, con un nivel de granularidad suficiente para identificar violaciones a nivel técnico y orientar acciones de mejora.

Claramente se evidencia que la gestión de niveles de servicio no es una temática nueva, sin embargo ha empezado a cobrar importancia en los últimos años por la evolución de los entornos tecnológicos hacia arquitecturas más dinámicas, orientadas a servicio, en las que no es siempre se tiene control sobre los recursos. En pocos casos se ha adoptado una postura flexible capaz de evolucionar y adaptarse a las nuevas necesidades, existen herramientas especializadas confinadas a lenguajes de programación o a marcos de trabajo centradas en problemas técnicos que atienden el cómo recuperar la información desde la perspectiva del proveedor. El modelo propuesto no pasa por alto la complejidad técnica involucrada en los procesos de recolección de los datos de calidad del servicio, es un planteamiento consiente de la evolución arquitectónica de los sistemas de información, desde simples aplicaciones de escritorio, hasta complejas aplicaciones de arquitectura SOA desplegada en ambientes de nube. Un factor diferenciador frente a otros trabajos, es que se delega la responsabilidad de entregar la información al proveedor del servicio y se

promueve la implementación de componentes de recolección bajo estándares de intercambio de información.

5.1.2 Conclusiones de la herramienta de soporte de auditoría de niveles de servicio de sistemas de información

sQMA busca atender limitaciones identificadas en los trabajos relacionados como que se centran en resolver problemas técnicos del entorno tecnológico; no se basan en estándares para la definición del proceso de auditoría; no se basan en estándares para la definición de la base de conocimiento de calidad de software, emplean un conjunto de métricas aisladas; no abarcan todo el ciclo de vida del producto de software; son rígidos, no permiten extender en tiempo de ejecución la base de conocimiento de calidad de software, requieren complejas configuraciones o implementaciones.

El diseño de la herramienta se enfocó en simplificar y automatizar procesos manuales para minimizar los esfuerzos técnicos invertidos en la configuración de complejas experiencias de evaluación de calidad, permitiendo al Gestor de Nivel de Servicio enfocarse en el diseño de los programas de mejora, en optimizar los costos de operación, y en fortalecer contratos de apoyo.

5.1.3 Conclusiones de la validación del modelo de auditoría de niveles de servicio de sistemas de información

Los casos de prueba funcionales permitieron verificar que los componentes del modelo de auditoría: modelo de dominio y modelo operacional, soportan la configuración de escenarios de calidad diseñados a partir de las métricas de la serie SQuaRE y del ciclo de auditoría según el proceso de gestión de niveles de servicio de ITIL V3. Aun cuando la muestra de métricas seleccionadas no pueda considerarse significativa, se hizo una selección con diferentes grados de complejidad en las funciones de procesamiento y análisis para evaluar aspectos funcionales de la herramienta asociados al modelo de auditoría, entregando elementos de retroalimentación de valor.

Adicionalmente los casos de prueba permitieron validar aspectos técnicos de la herramienta, como el tiempo medio de procesamiento de datos, la estabilidad del sistema bajo condiciones de carga, la facilidad para generar clientes de recolección y notificación datos de calidad.

5.1.4 Conclusiones generales

El modelo de auditoría propuesto, enmarcado en el contexto de la gestión de niveles de servicio y la gestión de calidad de software, se presenta como una integración de ambas perspectivas, para constituir un facilitador del aseguramiento de la calidad de los servicios de información. Al tiempo permite tomar decisiones desde el área de TI basadas en información verificable, depurada, brindando una visión integral de la salud del catálogo de servicios, dirigidas a los procesos internos o contratos de apoyo.

Consideramos que el modelo de auditoría constituye una propuesta de valor, convergente entre la serie SQuaRE e ITIL V3, no identificada en trabajos relacionados, tanto a nivel conceptual como de proceso para integrar buenas prácticas, bajo un modelo simplificado.

Por su parte la herramienta sQMA actúa como habilitador tecnológico. El nivel de automatización aportado puede influir en el grado de aceptación de los posibles usuarios, principalmente en áreas de TI que no tienen la capacidad estructural ni la madurez para

afrontar la implementación de los procesos de gestión de niveles de servicios con el enfoque y alcance de ITIL V3.

5.2 Trabajo futuro

Respecto al modelo de auditoría:

- Extender la comprobación del modelo del lado de ITIL, inicialmente se verificó que los modelos de proceso fueran compatibles a nivel de SLM y la posible relación entre los artefactos. Se debe ahondar en el tema para estudiar las posibles relaciones con las métricas de proceso definidas en ITIL.
- Definir estrategias para asegurar que los proveedores entreguen información no manipulada y verificable. Al tiempo investigar la construcción de monitores que puedan ser instalados en los sistemas evaluados, ya sea intrusiva o no intrusivamente.
- Validar el modelo de dominio y el modelo operacional respecto a otros modelos y estándares tanto de gestión de calidad de software como de gestión de niveles de servicio.
- Considerar el ambiente del escenario una variable del modelo, de esta forma, se esperarían diferentes indicadores para diferentes condiciones del componente del sistema evaluado, logrando una visión más aproximada a la realidad y a las necesidades.

Respecto a la herramienta sQMA:

- Proveer interfaces de usuario para configurar los modelos de calidad.
- Proveer interfaces de usuario para registrar los escenarios de calidad.
- Diseñar un navegador del catálogo de servicios, donde se evidencie la relación entre la evaluación de los escenarios de calidad, y los objetivos de proceso, a forma de tablero de comando.
- Almacenar las muestras de calidad originales en una base de datos orientada a documentos externa.
- Almacenar los scripts en un sistema externo y mejorar el mecanismo de integración para su invocación.

Respecto a la validación:

- Diseñar y ejecutar experiencias con productos de software reales, que se encuentren en producción en una organización para extender la comprobación de la integridad del modelo, su extensibilidad y nuevos requerimientos.
- Validar a nivel no funcional la herramienta para poder concluir al respecto de su extensibilidad, escalabilidad, rendimiento, etc.

Bibliografía

- [1] F. Scalone, "Estudio comparativo de los modelos y estándares de calidad del software," Maestría en Ingeniería de Calidad Master, Facultad Regional Buenos Aires, Universidad Tecnológica Nacional, 2006.
- [2] Axelos, "ITIL V3 - Service Design," in *Service Level Management*, ed: TSO, 2011, p. 123.
- [3] Axelos, "ITIL V3 - Service Design," in *Service Level Management*, ed: TSO, 2011, p. 124.
- [4] Axelos, "ITIL V3 - Service Design," in *Service Level Management*, ed: TSO, 2011, p. 107.
- [5] ISO/IEC, "ISO 25010 - Software engineering - Software product Quality Requirements and Evaluation (SQuaRE) - Software and quality in use models," in *Introduction*, ed: ISO, 2008, p. vi.
- [6] ISO/IEC, "ISO 25010 - Software engineering - Software product Quality Requirements and Evaluation (SQuaRE) - Software and quality in use models," in *Annex C*, ed: ISO, 2008, p. 31.
- [7] ISO/IEC, "ISO 25010 - Software engineering - Software product Quality Requirements and Evaluation (SQuaRE) - Software and quality in use models," in *5 Quality model framework*, ed: ISO, 2008, p. 10.
- [8] ISO/IEC, "ISO 25010 - Software engineering - Software product Quality Requirements and Evaluation (SQuaRE) - Software and quality in use models," in *Annex C.3 Approaches to quality*, ed: ISO, 2008, p. 32.
- [9] ISO/IEC, "ISO 25010 - Software engineering - Software product Quality Requirements and Evaluation (SQuaRE) - Software and quality in use models," in *Annex C.4 Software product quality life cycle model*, ed: ISO, 2008, p. 32.
- [10] ISO/IEC, "ISO 25010 - Software engineering - Software product Quality Requirements and Evaluation (SQuaRE) - Software and quality in use models," in *Annex C.5 Items to be evaluated*, ed: ISO, 2008, p. 24.
- [11] ISO/IEC, "ISO 25010 - Software engineering - Software product Quality Requirements and Evaluation (SQuaRE) - Software and quality in use models," in *Annex C.2 Software quality measurement model*, ed: ISO, 2008, p. 30.
- [12] Osiatis, "Service Level Management - Basic Concepts," vol. 2014, ed, 2014.
- [13] Axelos, "ITIL V3," in *Service Level Management*, ed: TSO, 2011, p. 100.
- [14] IT-processmaps. (2014, 13-05-2014). *ITIL Gestion del Nivel de Servicio - SLM - Roles* *ITIL*. Available: http://wiki.es.it-processmaps.com/index.php/ITIL_Gestion_del_Nivel_de_Servicio_-_SLM
- [15] Axelos, "ITIL V3," in *Service Level Management*, ed: TSO, 2011, p. 109.
- [16] A. Abran, R. E. Al-Qutaish, J.-M. Desharnais, and N. Habra, "An information model for software quality measurement with ISO standards," in *Proceedings of the*

- International Conference on Software Development (SWDC-REK), Reykjavik, Iceland, 2005*, pp. 104-116.
- [17] ISO/IEC. (2014, 15-06-2014). *ISO/IEC 15939:2007*. Available: http://www.iso.org/iso/catalogue_detail.htm?csnumber=44344
- [18] E. Wustenhoff, "Service Level Management in the Data Center," *Sun BluePrints*, p. 5, 2002.
- [19] D. Ameller and X. Franch, "Service Level Agreement Monitor (SALMon)," in *Composition-Based Software Systems, 2008. ICCBSS 2008. Seventh International Conference on*, 2008, pp. 224-227.
- [20] Axelos, "ITIL V3 - Continual Service Improvement," in *Tools to support CSI activities*, ed: TSO, 2011, p. 145.