

Revisión Documental Sistemática en el Ámbito de la Ingeniería del Dominio para Líneas de Productos de Software Orientados a Servicios

Juan Herrera¹, Francisca Losavio², Alfredo Matteo², Oscar Ordaz³

jchr1982@gmail.com, francislosavio@gmail.com, alfredojose.matteo@gmail.com, oscarordaz55@gmail.com

¹ PFG Informática para la Gestión Social, Universidad Bolivariana de Venezuela, Caracas, Venezuela

² Escuela de Computación, Universidad Central de Venezuela, Caracas, Venezuela

³ Escuela de Matemática, Universidad Central de Venezuela, Caracas, Venezuela

Resumen: Una Revisión Documental Sistemática (RDS) permite extraer conocimiento sobre un tema de investigación; esto se hace mediante preguntas adecuadas, a partir del gran volumen de información disponible en internet. RDS se enfoca en una pregunta de investigación que trata de identificar, evaluar, seleccionar y sintetizar evidencias de investigaciones de alta calidad relevantes para esa pregunta. El objetivo principal del trabajo es realizar una RDS en el tema de la Ingeniería del Dominio (ID) de Líneas de Productos de Software Orientados a Servicios (LPSOS), quien combina los enfoques de Líneas de Productos de Software (LPS) y de Arquitecturas Orientadas a Servicios o del inglés Service-Oriented Architecture (SOA). Esta investigación pretende determinar trabajos recientes existentes en la literatura en cuanto a actividades, artefactos y técnicas para un proceso de desarrollo del dominio, según Björner y de análisis del dominio según Pohl et al.. Como resultado de esta revisión se detectó, combinando los enfoques mencionados, que las fases principales de un proceso general de ID para LPSOS son: análisis, diseño e implementación, inspiradas en las fases definidas por Pohl et al. para la ID en un contexto de LPS. Dentro de estas fases se identificaron las actividades, artefactos y técnicas más usadas en la práctica. Estos resultados serán utilizados para definir un proceso sistemático de Análisis de Dominio para LPSOS, que además contemplará el tratamiento de requisitos de calidad, aspecto relevante para un contexto de producción industrial de software y no tratado sistemáticamente en los enfoques revisados.

Palabras Clave: Ingeniería del Dominio; Línea de Productos de Software; Arquitectura Orientada a Servicios; Línea de Productos de Software Orientadas a Servicios; Revisión Documental Sistemática; Calidad del Software.

Abstract: A Systematic Literature Review (SLR) allows to extract knowledge on a research topic; this is done through appropriate questions, from the large volume of information available on the internet. SLR is focused on a research question that tries to identify, appraise, select and synthesize high quality research evidence relevant to that question. The main goal of this work is to perform an SLR on the subject of Domain Engineering (DE) of Service-Oriented Software Products Line (SOSPL), who combines the approaches of Software Products Line (SPL) and Service-Oriented Architecture (SOA). This research aims to determine existent recent works in the literature regarding activities, artifacts, and techniques for development of the domain, according to Björner and domain analysis according to Pohl et al.. As a result of this review, main phases of a general DE process of for SOSPL were identified: analysis, design and implementation, inspired in the phases defined by Pohl et al. for DE in an SPL context. Within these phases activities, artifacts and techniques most used in practice were identified. These results will be used to define a systematic process of Domain Analysis for SOSPL, which also includes the treatment of quality requirements, relevant aspect for industrial software production the context, not systematically treated in the revised approaches.

Keywords: Domain Engineering; Software Products Line; Service-Oriented Architecture; Service-Oriented Software Products Line; Systematic Literature Review; Software Quality.

I. INTRODUCCIÓN

El desarrollo de sistemas de software de gran complejidad y de gran escala representa un desafío para los investigadores y desarrolladores en el área de la IS (Ingeniería de Software); surge entonces la tendencia al desarrollo basado en

componentes reutilizables, para de ahí configurar nuevos sistemas con rapidez y de más bajo costo [1][2][3].

Desde finales de los '60, y aun más en la década de los '90, el paradigma de las LPS (Líneas de Productos de Software) ha

cochado impulso en la industria del software. En lugar de desarrollar sistemas a partir de cero, estos deben ser contruidos a partir de partes reutilizables. En vez de componer un sistema siempre de la misma manera, éste debería ser adaptado a las necesidades de los clientes (del inglés, “stakeholders”), donde éstos pueden elegir entre un conjunto de opciones de configuración. El enfoque de LPS permite la construcción de soluciones individuales basadas en un repositorio de componentes de software reutilizables. La necesidad de proporcionar soluciones individuales responde a diversas necesidades en el software con respecto a la funcionalidad, las plataformas destino y las propiedades no funcionales, como por ejemplo rendimiento y espacio de memoria [4].

Las LPS prometen distintos beneficios [4][5], de los cuales los más importantes son: a) adaptabilidad a las necesidades del cliente, b) reducción de costos, c) mejora de la calidad global y d) disminución del tiempo de comercialización. Este enfoque representa un gran desafío para los investigadores y desarrolladores en el área de la IS, principalmente por la tarea de desarrollar artefactos (activos o del inglés “assets”) suficientemente genéricos para pensar en su reutilización y en la configuración de nuevos sistemas contextualizados acorde con las necesidades de los clientes [2][3].

Para el abordaje adecuado de este enfoque, la ILPS (Ingeniería de Líneas de Productos de Software) ofrece métodos y técnicas eficaces para la reutilización sistemática en el desarrollo de software con el fin de: a) apoyar arquitecturas de software configurables y b) permitir la personalización masiva de los sistemas de software [6][7]. Ha sido reconocida como un enfoque exitoso para la gestión de la variabilidad y la ingeniería de la reutilización. La ILPS consta de dos ciclos de vida principales: ID (Ingeniería del Dominio) e IA (Ingeniería de la Aplicación) [8]. La ID abarca el análisis y la identificación del ámbito de aplicación de la línea de productos, que incluye la captura de todo el conocimiento sobre el dominio de interés a través del modelado de partes comunes (del inglés “commonality”) y de puntos de variación (del inglés “variability points”) que están presentes en la arquitectura de la línea de productos.

Al considerar las fases del proceso de ID desde la perspectiva de las LPS [8], el *análisis del dominio* permite la captura y análisis del conocimiento sobre el dominio. El resultado principal del análisis del dominio es un artefacto denominado *modelo del dominio*. Este modelo representa las propiedades comunes y variables de la familia de sistemas en el dominio y las relaciones entre ellas. Conceptos del dominio y sus relaciones mutuas son analizados y modelados. Los conceptos del dominio representan el vocabulario del dominio y pueden ser representados por ontologías. Cada concepto luego es extendido por sus características comunes y variables y las dependencias entre ellos. Las características variables determinan el espacio de configuración para la familia de sistemas que se resuelve luego en la IA. El *diseño del dominio* produce la arquitectura abstracta genérica o de referencia (del inglés: Product Line Architecture, PLA) o RA (Reference Architecture) para la familia de sistemas, de acuerdo con los patrones arquitectónicos comúnmente aceptados (modelo-vista-controlador, en capas, etc.). Un ciclo completo de ID

construye el conocimiento de las necesidades específicas de los diferentes grupos de interés para los que se lleva a cabo la adaptación y configuración de la LPS.

Por otra parte, la *Arquitectura Orientada a Servicios* (del inglés: Service Oriented Architecture, SOA) [9][10] es una arquitectura centrada en la comunicación en redes a través de servidores Web que actúan como intermediarios para satisfacer demandas de servicios por clientes distribuidos en ubicaciones geográficamente distantes. SOA ha cobrado enorme auge en la práctica reciente de desarrollo de software rápido, a bajo costo y es ampliamente tratada actualmente a nivel de investigación en la ingeniería de software; constituye un paradigma para el diseño, desarrollo e implementación de aplicaciones de computación distribuida e independiente de la tecnología basada en estándares para el descubrimiento, el enlazado y ensamblaje de servicios de software débilmente acoplados [11]. Un *servicio* se define como una unidad discreta de la funcionalidad de negocio que está disponible a través de un contrato de servicios [12]; puede ser visto como un componente de software reutilizable. SOA es una arquitectura para la construcción de soluciones empresariales basadas en servicios [13]. Particularmente se ocupa de la construcción independiente de servicios alineados al negocio que pueden ser combinados en procesos de negocio de alto nivel y soluciones computacionales en el contexto empresarial. El valor real de SOA no es solo la provisión adecuada de los servicios, sino más bien es cuando los servicios reutilizables se combinan para crear procesos de negocios ágiles y flexibles. SOA ha demostrado su rentabilidad en el desarrollo de sistemas de software interoperables, reutilizables, adaptables y de rápido desarrollo, por lo cual existen importantes ventajas mutuas en la convergencia de SOA y LPS [1][14][15][16][17][18][19], dando origen al enfoque de LPSOS (Líneas de Productos de Software Orientadas a Servicios).

En todos los puntos antes tratados, el aseguramiento de la calidad del sistema como producto de software, es una actividad crucial para el éxito, ya que se ubican en un contexto de producción industrial; por lo tanto, cuando se trata del desarrollo de LPS, la reutilización masiva de los activos de software hace que los atributos de calidad (propiedades medibles de un artefacto de software) impacten en la calidad de todos los productos de la LPS [20]. La evaluación de la calidad es una tarea difícil en los sistemas de software monolíticos, y lo es aún más cuando se trata de LPSOS, ya que requiere analizar las características de calidad de una LPS bajo SOA, considerando propiedades funcionales y no funcionales comunes y variantes [21][22]. Por otra parte, la evaluación de la calidad es una parte esencial en la optimización y configuración de la LPS, ya que debe proporcionar métricas cuantitativas globales para la calidad de la familia de sistemas, generalmente basadas en la especificación de la arquitectura, como por ejemplo: costo, desempeño, disponibilidad, confiabilidad, escalabilidad y seguridad [23], pero también específicas para un sistema miembro de la familia [24].

Ahora bien, en un contexto de LPSOS, particularmente durante el ciclo de ID, la determinación de la calidad del servicio (del inglés: Quality of Service, QoS) es fundamental, debido a que varía acorde a características individuales, por lo cual los ingenieros del dominio deben asegurar que los servicios que

van a formar parte del sistema satisfacen los rangos de valores establecidos por los QoS solicitados por las partes interesadas [23].

En los estudios anteriormente referenciados se han detectado las siguientes evidencias de problemas aún no completamente resueltos, para algunos de los cuales este trabajo pretende proporcionar una solución:

- No se describen en detalle las actividades a realizar para la adquisición del conocimiento del dominio.
- No se describen diferentes perspectivas en el dominio que puede ser analizadas, por ejemplo, las facetas del dominio, utilizadas en el enfoque de Bjorner [25].
- Los requisitos funcionales son los abordados en general, pero los requisitos no funcionales y/o requisitos de calidad son dejados de lado, desde las siguientes perspectivas [23]: características de calidad específicas de la línea de productos (mantenibilidad, complejidad, etc.) y características de calidad específica del dominio o calidad global (disponibilidad, seguridad, etc.).
- La determinación de las características de calidad no es realizada en las etapas tempranas del proceso de ID (análisis del dominio); se ha detectado que son de vital importancia para derivar de forma adecuada el conjunto de productos de la LPSOS.

Es de particular interés en el contexto de esta investigación, que se centra básicamente en la fase de Análisis del Dominio de la ID, por ser la más importante y previa a la derivación de los productos concretos de la LPS en la IA, identificar las actividades, artefactos y técnicas utilizadas en un proceso completo de análisis del dominio para el desarrollo de las LPSOS, diferenciándolas: a) por los artefactos de entrada y de salida, b) por el enfoque de diseño utilizado (proactivo o reactivo) para la arquitectura, con el fin de describir como se analiza y realiza el proceso de análisis del dominio en la ID y c) también es de especial interés identificar las técnicas (UML (Unified Modeling Language), FODA (Feature-Oriented Domain Analysis), etc.) utilizadas como notaciones para sus principales artefactos.

Una RDS (Revisión Documental Sistemática) [26] permite extraer el conocimiento sobre el estado del arte en un tema específico de investigación; este conocimiento es extraído, mediante preguntas adecuadas, a partir de un gran volumen de información disponible. En consecuencia una RDS de la literatura que aborda el tema de la ID de LPSOS es indispensable para determinar los alcances y limitaciones de las propuestas encontradas. El objetivo de este trabajo es utilizar la RDS para identificar las actividades, artefactos y técnicas comúnmente utilizadas en la ID encontrados en la literatura reciente, para luego integrarlos para definir un proceso único de análisis del dominio para LPSOS incorporando también aquellas actividades y/o artefactos faltantes.

Este trabajo está estructurado de la siguiente manera, además de esta introducción y las conclusiones: en la Sección II se presentan los enfoques a considerar; en la Sección III se presenta el análisis de la revisión documental sistemática

realizada. En la Sección IV se discuten los resultados obtenidos. Finalmente, en la Sección V se da un resumen de los aportes de la investigación.

II. ENFOQUES A CONSIDERAR

A. ID (Ingeniería del Dominio)

Un *dominio* es un área de conocimiento que utiliza conceptos comunes para la descripción de los fenómenos, requisitos, problemas, capacidades y soluciones. Se asocia generalmente con una terminología bien definida o parcialmente definida. Esta terminología se refiere a los conceptos básicos del dominio, sus definiciones (es decir, sus significados semánticos), y sus relaciones. También puede referirse a las conductas que se desean, prohibidas, o percibidas dentro del dominio [27]. Un *dominio de aplicación* comprende cualquier aspecto en el cual la computación pueda aplicarse. Según Bjorner [25], básicamente existen tres clases de dominios de aplicación:

- La clase de aplicaciones que pueden caracterizarse por el apoyo a la enseñanza o el estudio de un campo: el software educativo o de formación, software experimental de demostración de teoremas, o similares.
- La clase de aplicaciones que pueden caracterizarse por el apoyo al desarrollo de sistemas informáticos propios: compiladores, sistemas operativos, sistemas de gestión de bases de datos, sistemas de comunicación de datos, etc.
- La clase de aplicaciones que pueden caracterizarse por el apoyo al desarrollo de software comercial o industrial, esto incluye los sistemas de información en general y las aplicaciones empresariales, entre otras.

Particularmente, la última clase de estos dominios de aplicación son los de interés en el trabajo de Bjorner [25]. Por ello, la selección del trabajo de este autor es argumentada por la razón que las aplicaciones implementadas mediante SOA fueron diseñadas con la intención de apoyar a través de esta arquitectura basada en un intermediario para la comunicación, los dominios de aplicación relacionados con los sistemas de información gerencial y/o empresarial que utilizan un estilo clásico de capas, como por ejemplo, capa de presentación, lógica y datos.

La ID es el conjunto de actividades cuyo objetivo es desarrollar, mantener y administrar la creación y evolución de los dominios [25][27]. Ha sido de especial interés para los sistemas de información y de las comunidades de ingeniería de software, por varias razones: a) la necesidad de mantener y reutilizar los conocimientos existentes, b) la necesidad de gestionar los crecientes requisitos de la variabilidad de la información y de los sistemas de software, y c) la necesidad de obtener, formalizar, y compartir conocimientos especializados en diferentes ámbitos, en evolución [27].

La ID como disciplina tiene una importancia práctica, ya que debe proporcionar los métodos y técnicas que pueden ayudar a reducir el tiempo de entrega al mercado, los costos de desarrollo y los riesgos en los proyectos, además que deben ayudar a mejorar la calidad del producto y el rendimiento sobre

una base consistente [27]. La ID es utilizada, investigada y estudiada en diversas áreas, principalmente:

- ILPS (Ingeniería de línea de productos de software)
- ILED (Ingeniería de lenguaje específico de dominio)
- Modelado conceptual.

Según Bjorner [25], antes de diseñar el software, se deben comprender sus requerimientos, y antes que se puedan desarrollar los requerimientos, se debe comprender el dominio de la aplicación en que se va a construir. Para comprender el dominio de la aplicación, se debe realizar una etapa de análisis del dominio desde las perspectivas de las facetas del dominio bajo la óptica de los grupos de interés. Es decir, cada grupo de interés tendrá características particulares en cada una de las facetas que conforma el dominio de la aplicación. La ID reduce el esfuerzo de trabajo relacionado con algunos aspectos de la IS, especialmente los vinculados con la ingeniería de requisitos.

La Tabla I muestra la relación entre las fases de la ID expresada por Bjorner [25] y las fases expresadas por Pohl et al. [8]. Esto se hace con la intención de mostrar como los conceptos expresados por Bjorner en ID, complementan los de Pohl bajo el enfoque LPS. El enfoque propuesto por Bjorner fue seleccionado debido a que las fases del proceso para el desarrollo del dominio son claramente definidas y argumentadas, y debido al gran nivel de detalle en la descripción de las actividades realizadas, expresadas mediante las facetas (o vistas) del dominio.

El análisis del dominio basado en las facetas del dominio desde las perspectiva de los stakeholders, permitirá conocer el dominio de la aplicación al considerar de forma más adecuada los aspectos funcionales y no funcionales (requisitos de calidad) que deben ser satisfechas desde diferentes vistas.

Tabla I: Relación entre las Fases de la ID según Dines Bjorner [25] y Pohl et al. [8]

Perspectiva de Bjorner [25]	Perspectiva de Pohl et al. [8]
Proceso de desarrollo del dominio: Identificación de los stakeholders; Adquisición del dominio; Análisis del dominio y formación de conceptos	<i>Análisis del dominio:</i> modelo de procesos de negocios; modelo de características
Modelado del dominio (facetas del dominio): Procesos del negocio; Intrínsecos; Tecnologías soportadas; Organización y gerencia; Reglas y regulaciones; Scripts (guiones); Comportamiento humano	<i>Diseño del dominio:</i> arquitectura genérica abstracta; arquitectura de referencia (top-down) y/o arquitectura de línea de productos (botton-up)

B. Proceso para el Desarrollo del Dominio, dentro de la ID Propuesto por Bjorner

Contempla las siguientes fases o subprocesos:

- identificación de los stakeholders
- adquisición del conocimiento del dominio

- análisis del conocimiento del dominio y formación de conceptos
- modelado del dominio (diseño del dominio)
- verificación y validación del dominio y
- teoría del dominio.

1) *Los Stakeholders del Dominio y sus Perspectivas:* Los stakeholders del dominio se refieren a una persona, o grupo de personas unidas de alguna manera, o por una institución, empresa o un grupo de este tipo, caracterizado por su interés común en el dominio o dependencia del dominio. Por la perspectiva de los stakeholders del dominio lo comprendemos a él, o una comprensión del dominio compartido por un grupo de stakeholders específicamente identificado; una vista que puede diferir de un grupo de interés a otro grupo de interés del mismo dominio. La identificación de las perspectivas de los stakeholders (por ejemplo, vistas) incorpora el desarrollo de principios, técnicas y herramientas. Sin la clara identificación y enlace con todas las partes relevantes interesadas del dominio no se puede aspirar a construir un modelo de dominio creíble.

2) *Adquisición del Dominio:* La adquisición del dominio comprende el proceso de obtención de datos sobre el dominio, esto es, obtener (capturar) datos (hechos) de las partes interesadas del dominio, de redactar las descripciones acerca de ellos, y de su estructura (es decir, su organización y/o clasificación aproximada de estas descripciones). Esto incluye la recolección de datos, de la literatura y de nuestras observaciones acerca del conocimiento del dominio. Este conocimiento incluye entidades fenomenológicas, funciones, eventos y comportamientos.

3) *Análisis y del Dominio y Formación de Conceptos:* El análisis del dominio comprende el estudio de la adquisición del conocimiento del dominio, con los siguientes objetivos: a) descubrir inconsistencias, conflictos y la falta de completitud dentro de ellas, b) formar conceptos de estas declaraciones de adquisición del conocimiento del dominio. La formación de conceptos del dominio comprende la abstracción de los fenómenos del dominio, en conceptos.

4) *Modelado del Dominio (Modelo de Facetas del Dominio):* Las facetas del dominio comprende un conjunto finito de formas genéricas de analizar un dominio, cada forma genérica representa una vista particular del dominio o faceta del dominio, de tal modo que las diferentes facetas abarcan conceptualmente diferentes vistas, y juntas conforman el dominio. A continuación se enuncian las principales facetas del dominio según [25]:

- Procesos del negocio (conjuntos de acciones que se realizan en el dominio)
- Características intrínsecas (lo que es común a todas las facetas)
- Tecnologías de apoyo (implementación de las facetas)
- Organización y gestión

- Reglas y regulaciones (normativas)
- Comportamiento del recurso humano (stakeholders)

Un modelo de dominio se refiere al conjunto de uno o más modelos acordes con las facetas del dominio, estas pueden ser reescritas (y posiblemente formalizadas) en un modelo consolidado (un único modelo en donde confluyen todas las facetas).

Para modelar una faceta del dominio, primero hay que adquirirla; entonces se debe analizar lo que se ha adquirido, y formar conceptos de lo que se ha analizado, para luego describirla: a) aproximadamente, b) en términos de las entidades del dominio (terminología), c) de forma narrativa y d) posiblemente formalizando la faceta [25].

5) *Validación del Dominio*: La validación del dominio se refiere a la garantía, con los stakeholders, en particular los clientes, que las descripciones del dominio que se producen como resultado de la adquisición del dominio, análisis del dominio, la formación de conceptos y el modelado del dominio es acorde con la forma en que los grupos de interés miran el dominio.

C. Arquitectura Orientada a Servicios (SOA)

SOA se ha convertido en un concepto ampliamente estudiado y utilizado en la investigación y la práctica de la ingeniería de software. SOA es una arquitectura para el diseño, desarrollo e implementación de aplicaciones de computación distribuida e independiente de la tecnología basada en estándares para el descubrimiento, el enlazado y ensamblaje de servicios de software débilmente acoplados que son transmitidos mediante una red [11].

SOA proporciona el marco conceptual para la realización de sistemas orientados al servicio al permitir que los sistemas de software puedan ser compuestos y reconfigurados dinámicamente usando servicios detectables en la red. Desde el punto de vista de la ILPS, este modelo de despliegue promete beneficios significativos sobre el modelo tradicional de implementación del software como un producto: en primer lugar, la naturaleza dinámica de SOA significa que puede soportar las necesidades y expectativas del usuario en un entorno de cambios continuos; en segundo lugar, los servicios pueden ser combinados con diferentes configuraciones y contextos simplificando el despliegue de variantes de los productos adaptados a las necesidades de los distintos clientes, pero sobre todo basados en los mismos servicios básicos [28].

Particularmente SOA se ocupa de la construcción independiente de servicios alineados al negocio que pueden ser combinados en significativos procesos de negocio de alto nivel y soluciones en el contexto empresarial [29]. El valor real de SOA no es solo la construcción de los servicios, sino más bien cuando los servicios reutilizables se combinan para crear procesos de negocios ágiles y flexibles. La agilidad y flexibilidad se producen cuando los nuevos procesos pueden rápidamente y eficientemente ser creados a partir del conjunto de servicios existente [12].

SOA permite el ensamblado, orquestación y el mantenimiento de las soluciones empresariales para reaccionar rápidamente a los cambiantes requisitos empresariales [30].

D. Calidad del Software

La calidad ha sido típicamente definida como un nivel de excelencia, de conformidad con las especificaciones, de satisfacción de requisitos, de atributos distintivos, de estar libre de defectos, deficiencias y de las significativas variaciones para cumplir con las expectativas del cliente [31][32].

Establecer un modelo de calidad para el desarrollo del dominio de la aplicación que involucra los procesos de identificación de los stakeholders, adquisición del dominio, análisis del dominio y modelado del dominio, planteados en Bjorner [25], garantizaría la validez y confiabilidad del dominio desarrollado. El aseguramiento de la calidad del producto es una actividad crucial para el éxito de la industria del software, pero es, si cabe, más importante cuando se trata del desarrollo de líneas de producto software, dado que la reutilización masiva de activos de software hace que los atributos de calidad (propiedades medibles de un artefacto de software) de estos activos impacten en la calidad de todos los productos de una línea de producto [17].

La calidad se ha convertido en un atributo crítico de los productos de software ya que su ausencia produce pérdidas financieras, de salud, y a veces de vida. Al mismo tiempo la definición, o alcance, del dominio de la calidad del software ha evolucionado continuamente desde una perspectiva técnica a una perspectiva que abarca aspectos humanos tales como la facilidad de uso y la satisfacción [21][24].

III. REVISIÓN DOCUMENTAL SISTEMÁTICA

La RDS se llevó a cabo en el primer y segundo trimestre del 2014. Kitchenham indica que la gestión del proceso de revisión sistemática se fundamenta en tres fases principales: Planificación de la revisión, Realización de la revisión e Informe de la revisión [26]. En la Tabla II se describen las fases, las etapas que incluyen cada fase y los artefactos generados en el proceso de revisión sistemática.

Tabla II: Proceso de Revisión Documental Sistemática

Fases	Etapas	Artefactos
Planificación de la revisión	Justificación de la necesidad de una revisión.	Protocolo
	Especificación de la(s) interrogante(s) de investigación.	
	Desarrollo del protocolo de revisión.	
Realización de la revisión	Identificación de los estudios relacionados.	Estudios aceptados
	Selección de los estudios primarios.	
	Evaluación de la calidad de los estudios.	
	Extracción de los datos.	Modelo de datos
Síntesis de los datos.		
Informe de la revisión	Especificación de los mecanismos de difusión.	Tablas resumen
	Presentación de los resultados	

A. Planificación de la Revisión

Esta fase define los objetivos de la investigación y el protocolo en que la revisión se ejecutará.

1) *Identificación de la Necesidad para una Revisión Sistemática:* Estrategias y actividades utilizadas durante el proceso de ID para el desarrollo de LPSOS, con el fin de determinar los alcances y limitaciones de las propuestas encontradas.

2) *Objetivos:* Revisar los procesos de desarrollo actuales en el área de la ID para el enfoque de LPS para el desarrollo de Familias de Productos Orientadas a Servicios; Identificar las estrategias y actividades utilizadas en las fases de análisis, diseño e implementación durante el proceso de ID en el desarrollo de LPSOS; Recolectar evidencia acerca de las investigaciones actuales que indiquen sus implicaciones en la práctica; Identificar problemas pendientes por resolver y posibles áreas para la mejora de estos procesos.

3) *Desarrollo del Protocolo:* El protocolo es un plan o conjunto de pasos a seguir en un estudio, constituido por las preguntas de investigación, las estrategias de búsqueda, criterios y procesos de selección, los criterios de evaluación de la calidad, el modelo y proceso de extracción de datos, y el plan de análisis de los datos.

Las preguntas de investigación tienen el objetivo de encontrar todos los estudios para comprender y resumir las evidencias acerca de los enfoques propuestos o utilizados durante el desarrollo del proceso de ID, las siguientes PIs (Preguntas de Investigación) fueron consideradas:

- PI1: ¿Cuáles son las actividades comúnmente realizadas en las etapas de análisis, diseño e implementación en el proceso de ID bajo el enfoque de LPSOS?
- PI2: Desde el punto de vista de las LPS implementadas con SOA. ¿Cuáles son los artefactos comúnmente utilizados y técnicas utilizadas para su construcción en la fase de ID para LPSOS?
- PI3: Desde el punto de vista de la fase de ID. ¿Cuál es el enfoque utilizado para el diseño de la arquitectura de referencia y/o arquitectura de líneas de productos (desde la perspectiva de la investigación realizada en [33]: enfoques proactivo y reactivo)?
- PI4: ¿Existen vistas de calidad que tomen en consideración algún estándar o modelo de calidad del producto (ISO9126-1, ISO25010 [32]) que tomen en consideración las características de calidad particulares para el diseño de LPSOS?
- PI5: ¿Existen herramientas de análisis o diseño que apoyen el proceso de ID para el desarrollo de LPSOS?

Con relación a la estrategia de búsqueda, primero se establecieron las fuentes de búsqueda para encontrar los estudios primarios, particularmente se seleccionaron los siguientes gestores de búsqueda: Google Academic, Scirus, ResearchGate y ScienceDirect.

Para establecer la estrategia de búsqueda, teniendo en cuenta las preguntas de investigación, inicialmente se identificaron las principales palabras de acuerdo a la terminología utilizada por Bjorner [25], que a continuación se enuncian: Software Product Line, Service-Oriented Architecture, Domain Engineering, Domain Stakeholders, Domain Acquisition, Domain Analysis, Domain Design, Domain Modeling, Domain Verification, Domain Validation, Analysis Tool.

Con relación a los criterios de inclusión y exclusión, aquí se establecen los criterios por los cuales los estudios serán evaluados para decidir si deben ser seleccionados o no en el contexto de la revisión sistemática.

Por lo tanto, los CI (Criterios de Inclusión) que se utilizan para incluir los estudios relevantes en esta RDS:

- CII: Que el artículo aborde el área de la Ingeniería de Dominio para el desarrollo de Líneas de Productos de Software para Familias de Productos Orientados a Servicios utilizando SOA.

Los CE (Criterios de Exclusión) que se utilizan para excluir los estudios que no contribuyen a responder a las interrogantes de la investigación son:

- CE1: Que el artículo no aborde la integración de LPS y SOA como alternativa para el desarrollo de productos de software
- CE2: Que el artículo este escrito en un lenguaje diferente al inglés, portugués y español
- CE3: Que el artículo no esté disponible en el formato PDF
- CE4: Que el artículo tenga como contenido una presentación en un congreso o evento realizado por los autores.
- CE5: Que el artículo este repetido (el mismo artículo, pero con dos nombres diferentes), en este caso solo se acepta un solo ejemplar.

Con respecto a la extracción de datos y método de síntesis, se prevé la construcción de tablas para la obtención de datos relacionados con las preguntas de investigación, en caso de que sea aplicable.

En la selección inicial de las publicaciones de entre los resultados de búsqueda, se dio lectura a los títulos y resúmenes de las publicaciones encontradas. En la Figura 1 se muestra el modelo de datos inicial utilizado en la revisión sistemática.

Las publicaciones fueron almacenadas en carpetas independientes de acuerdo a la expresión de búsqueda utilizada, de esta forma facilitar la ubicación de los estudios para realizar el análisis con mayor fluidez y comodidad.

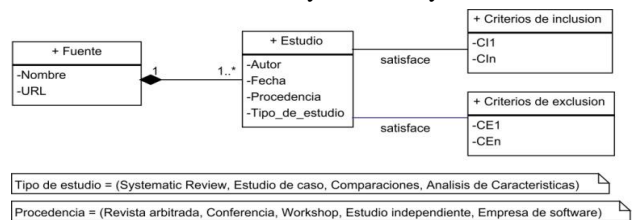


Figura 1: Modelo de Datos Inicial para la Revisión Sistemática

Del conjunto de publicaciones seleccionadas se procede a la extracción de la información relevante utilizando el modelo de datos que será utilizado en las fases de realización e informe de la revisión. En la recolección de datos, las citas textuales de los artículos analizados se conservarán en el idioma original (inglés) para evitar el posible sesgo al traducir y posteriormente parafrasear según la interpretación del investigador sobre la cita leída. Esto permite una vez realizados los aportes a la investigación, poder referirse a las citas por posibles malas interpretaciones de las mismas, para validar y para futuras referencias. No obstante, al final de la revisión sistemática los resultados finales reflejados en la fase de informe de la revisión se colocarán en español que es el idioma nativo del investigador.

Para estandarizar la forma en que la información estará representada, se definió un formato para recopilar datos de los estudios seleccionados. La Tabla III muestra el formato que será utilizado para registrar los resultados de la extracción de los datos.

Tabla III: Formato para la Recolección de Datos

Estudio				
Definición ID				
Etapas	Actividades	Propósito	Artefactos de entrada	Artefactos de salida
Análisis				
Diseño				
Implementación				
Enfoque				
Comentarios				

La Tabla IV describe las propiedades de extracción que serán utilizados a lo largo de la revisión sistemática.

Tabla IV: Propiedades de Extracción

Propiedades	Descripción
Estudio	Identificación del artículo objeto de estudio.
Definición	Concepto utilizado en el estudio para definir la ID.
Etapas	Indica las etapas utilizadas en los trabajos analizados para descubrir el proceso de desarrollo del dominio.
Actividades	Indica las actividades realizadas en cada una de las etapas anteriores.
Propósito	Objetivo o meta propuesta por la actividad a realizar.
Artefactos de entrada	Artefactos utilizados como insumos para poder realizar una actividad.
Artefactos de salida	Artefactos producidos como consecuencia de haber realizado una actividad.
Enfoque	Enfoque utilizado para el desarrollo del dominio; puede ser Proactivo o Reactivo.
Comentarios	Aspectos de interés reflejados en la investigación.

B. Realización de la Revisión

En esta fase, los estudios son identificados, seleccionados y evaluados de acuerdo con el protocolo previamente establecido. Como resultado de la búsqueda 231 estudios fueron localizados, de los cuales solo 67 estudios fueron aceptados para su análisis.

Para la selección final de los estudios aceptados se realizó un proceso filtrado que consistió en la aplicación de los criterios de inclusión y de exclusión, en la Tabla V se muestra un

resumen de los resultados indicando la cantidad en números de los estudios aceptados y rechazados.

Tabla V: Relación de Estudios Localizados y Aceptados según la Fuente de Datos

Fuentes de datos	Artículos	
	Aceptados	Localizados
Google Academic	20	36
Scirus	30	167
ResearchGate	8	10
ScienceDirect	9	18
Total	67	231

En los estudios aceptados se descartaron los artículos repetidos donde los identificadores de los archivos (nombres) son iguales; también se descartaron aquellos artículos, que al examinar sus contenidos se encontró que eran idénticos, aun cuando sus identificadores eran diferentes.

A continuación en las Tablas VI y VII se muestra la distribución de los estudios aceptados agrupados por categorías o ámbitos de análisis, basados en la terminología utilizada en [25] en las etapas de desarrollo del dominio de aplicación.

Tabla VI: Etapas (Categorías) para el Desarrollo del Dominio [25]

Desarrollo del Dominio		
Categorías de análisis		Total
1	Stakeholders Identification	2
2	Domain Acquisition	2
3	Domain Analysis and Concept Formation	32
4	Domain Modeling: (domain design)	18
5	Domain Validation and Verification	0
6	Domain Theory	0
7	Domain Documentation	1
Total		55

Tabla VII: Otras Categorías de Interés para la Investigación

Otras categorías de análisis	Total
Domain-Driven Design	2
Quality Model	1
Analysis Tool	0
Design Tool	0
Domain Architecture	9
Total	12

A continuación, un análisis más detallado se llevó a cabo en cada estudio incluido y el informe de la revisión.

C. Informe de la Revisión

En esta última fase, se presentan los resultados analíticos de la revisión sistemática. La extracción de los datos obtenidos y la síntesis del conocimiento considerando cada pregunta de investigación se discute a continuación.

Las respuestas a las preguntas de investigación versan sobre los aportes de la investigación realizada por Dines Bjorner [25] para el desarrollo del dominio de propósito general, del trabajo realizado por Pohl et al. [8] sobre el procesos de ID específicamente para las LPS, y de los trabajos realizados en [3][14][16][17][34][35][36][37][38][39] vinculados con las LPS para el desarrollo de FPOS (Familias de Producto Orientados a Servicios), que particularmente utilizan como arquitectura de implementación la SOA (Arquitectura Orientada a Servicios).

1) *Con Relación a la Pregunta P11 y P12: ¿Cuáles son las actividades comúnmente realizadas en las etapas de análisis, diseño e implementación en el proceso de ID bajo el enfoque de LPSOS?*

Para responder a estas dos preguntas, se utilizaron las siguientes dos estrategias; primero, de los estudios aceptados inicialmente, se seleccionaron aquellos estudios que abordan o describen un proceso de desarrollo para LPSOS; segundo, para representar los resultados del análisis de estos estudios, y mostrarlos de una forma más amigable, se utilizó un modelo de datos para la sistematización de los datos. De los 67 estudios aceptados, solo 10 estudios fueron seleccionados, debido a que abordan y/o describen un proceso de desarrollo para la construcción de LPSOS, que es el objetivo principal de interés del estudio.

Las Tablas VIII, IX y X describen el conjunto de actividades, artefactos generados y técnicas utilizadas respectivamente en la construcción de LPSOS a partir de los estudios analizados: [3][14][16][17][34][35][36][37][38][39] vinculados con las LPS para el desarrollo de Familias de Producto Orientados a Servicios. Los 10 estudios analizados abordan el proceso de ID bajo el enfoque de LPSOS basándose en las siguientes 3 etapas: análisis, diseño e implementación.

Tabla VIII: Actividades Realizadas en las Fases de Análisis, Diseño e Implementación de la LPSOS

Etapas	Actividades
Análisis	Análisis de factibilidad de la LPS Análisis del dominio Análisis de características: similitudes y variabilidades de la LPS Análisis de los activos existentes Análisis de los requerimientos de la familia
Diseño	Identificación de los elementos arquitecturales Especificación de la arquitectura Definición de la técnica de implementación de la variabilidad
Implementación	Construcción de los componentes/servicios Plan de producción (<i>Roadmap</i>) de los productos de la LPS

Tabla IX: Artefactos Desarrollados en las Fases de Análisis, Diseño e Implementación de la LPSOS

Etapas	Artefactos
Análisis	Documento de factibilidad Modelo conceptual del dominio Modelo de características Listado de componentes/servicios disponibles Tabla de identificación de requerimientos
Diseño	Servicios/Componentes Arquitectura de referencia Modelo de configuración Modelo de componentes/servicios Modelo de estado Modelo de colaboración, interacción, comunicación. Mecanismos de implementación de la variabilidad.
Implementación	Componentes SPL desarrollados, reutilizados, o comprados a terceros. Composición de los componentes/servicios

Tabla X: Técnicas Utilizadas en las Fases de Análisis, Diseño e Implementación de la LPSOS

Etapas	Técnicas
Análisis	Diagrama de clases (UML) Notación de Modelado de Procesos de Negocios (BPMN) Diagrama de actividades (UML) Diagrama de Casos de Uso (UML) FODA, variantes de FODA Tablas Requisitos Funcionales – No Funcionales
Diseño	Service-Oriented Architecture (SOA) Service-Component Architecture (SCA) Vistas arquitecturales: Diagrama de clases (UML), Diagrama de componentes (UML), Diagrama de estado (UML), Diagrama de colaboración (UML). Lenguaje de Variabilidad Común (LVC) Mecanismos de configuración, generación
Implementación	SCA, SOA Orquestado/configuración/composición

2) *Con Relación a la Pregunta “PI3”: ¿Desde el punto de vista de la fase de ID, cual es el enfoque utilizado para el diseño de la arquitectura de referencia y/o arquitectura de líneas de productos?*

Desde el punto de vista de la fase de ID, el enfoque utilizado para el diseño de la arquitectura de referencia y/o arquitectura de líneas de productos (desde la perspectiva de la investigación realizada en [33] por la mayoría de los estudios analizados fue el enfoque proactivo.

3) *Con Relación a la Pregunta “PI4”: ¿Existen vistas de calidad que tomen en consideración algún estándar o modelo de calidad del producto?*

No existe evidencia de algún estudio que proponga o utilice alguna vista de calidad, modelo de calidad o implemente algún estándar de calidad del producto, utilizando ambos enfoques: LPS y SOA. Solo un trabajo [23] implementa estrategias que incorporan características de calidad en un enfoque LPSOS, es decir que consideran la calidad desde la perspectiva de QoS para la concepción de la LPS y toman en consideración la arquitectura de implementación SOA, sin embargo, aunque esta investigación hace un análisis superficial del modelo de calidad ISO9126-1 y del ISO25010 [32], argumentando que debe adecuarse a las particulares características de las LPSOS, no se propone una vista de calidad basada en alguno de los estándares antes mencionados.

Por otro lado, se evidenciaron dos trabajos independientes que abordan o proponen un modelo de calidad, uno de ellos describe un procedimiento para incorporar un modelo de calidad para una LPS [40], el segundo si aborda un modelo de calidad basado en el ISO25010, pero solo para SOA bajo el paradigma de la Computación Orientada a Servicios (del inglés: Service-Oriented Computing, SOC) [41].

4) *Con Relación a la Pregunta “PI5”: ¿Existen herramientas de análisis o diseño que apoyen el proceso de ID para el desarrollo de LPSOS?*

No se describen herramientas y/o prototipos que apoyen o aborden el proceso de desarrollo de la LPSOS, ni que utilicen alguna de vista de calidad del dominio aplicando un modelo de calidad o estándar (ISO9126, ISO25010, entre otros); sin embargo existen algunos estudios que abordan el proceso, por ejemplo desde la perspectiva de la LPS [5][20][42], particularmente se encuentra una herramienta en línea a través del enlace: www.splot-research.org.

Por otra parte, en el estudio [43] que aborda el análisis de las características de las herramientas del tipo open source que apoyan procesos de LPS, muestra que la mayoría de las propuestas se basan o parten del modelado de las características de la LPS [44][45][46].

IV. DISCUSIÓN

Es conveniente y necesario describir aquellos aspectos que resaltan de cada uno de los 10 procesos analizados, aparte de las actividades realizadas en las etapas de análisis, diseño e implementación del dominio; entre ellas se destacan las siguientes evidencias:

Existen muy pocos estudios que abordan adecuadamente la identificación de los stakeholders (partes interesadas del dominio).

Particularmente no se identifican claramente los stakeholder (actores) involucrados para efectuar la adquisición del dominio, aspecto fundamental según Bjorner [25], para obtener una visión global del dominio de la aplicación.

Solo algunos estudios [3][36] identifican los roles de los stakeholders y en que etapas del proceso de desarrollo del dominio actúan o toman decisiones; pero solo en las fases de análisis, diseño e implementación del dominio.

También existen pocos estudios que aborden algún método, técnica o procedimiento relacionado con la adquisición del dominio.

En todos los estudios seleccionados para la investigación se observó que ninguno valida y/o verifica el dominio desarrollado por los ingenieros de software (arquitectura de referencia, componentes/servicios) contra el dominio especificado por los expertos del dominio del problema (conocedores de los procesos de negocios en la organización).

La mayoría de los estudios analizados, solo abordan las fases de análisis y modelado del dominio.

Solo algunos describen como implementar el dominio (arquitectura de referencia [47]).

La arquitectura de referencia descrita por los autores, no toman en cuenta alguna vista de calidad del dominio, como el descrito en [48], o toman en consideración algún estándar de calidad, por ejemplo: ISO9126 o la versión actualizada ISO25010 [32].

Se evidencia en los estudios, que los conocedores del dominio por lo general no son del área de la informática o afines, si la organización requiere una LPS implementada a través de SOA, no se describe o enuncia como se realiza la transformación de las decisiones del negocio al BPM (Modelo de Procesos de Negocios), FM (Modelo de Características) en el caso de la

LPS y los servicios que ha de soportar la arquitectura, particularmente para SOA; existe mucha ambigüedad y poca claridad en los estudios analizados.

Solo los stakeholders conocedores del dominio del problema con experiencia o estudios en el campo de la informática tendrían las habilidades o capacidades necesarias para realizar la tarea anterior con éxito o por lo menos con más precisión desde el punto de vista del desarrollo de LPSOS que se pretende realizar en la organización.

Tampoco se definen vistas de calidad y/o estándares de calidad para validar de manera formal el proceso de adquisición, análisis y diseño del dominio para la LPSOS [20][42].

Los 10 estudios abordan parcialmente la documentación total del dominio, según lo planteado por Bjorner [25], referente a las facetas del dominio.

Ninguno de los 10 estudios analizados aborda las 6 etapas (identificación de los stakeholders, adquisición del dominio, análisis del dominio, diseño del dominio, validación/verificación del dominio y desarrollo de la teoría del dominio) para el desarrollo del dominio propuesta por Bjorner [25].

Se destaca que los 10 estudios abordan el proceso de desarrollo de la ID de forma diferente, es decir, aun cuando los procesos utilizan algunas actividades y artefactos comunes, cada uno de ellos realiza el proceso visto de forma global, con actividades, artefactos y técnicas diferentes.

V. CONCLUSIONES

La principal contribución de este trabajo fue realizar una RDS para presentar una perspectiva detallada sobre las actividades, artefactos y técnicas que comúnmente se utilizan en un proceso de desarrollo para la ID de LPSOS. Existe una carencia en los estudios analizados en lo referente al uso de estándares de calidad en los procesos de desarrollo identificados y falta de trazabilidad entre los modelos utilizados, sobre todo en lo que respecta a los RNF (Requisitos No Funcionales) del modelo del negocio al modelo del sistema de software en un dominio dado. Esta revisión sistemática provee a la comunidad de desarrolladores el estado del arte en lo concerniente con el desarrollo completo de la disciplina de ID para LPSOS.

Actualmente se está definiendo un proceso de análisis de dominio para LPSOS, tomando en cuenta los análisis realizados a partir de esta RDS. Se ha seleccionado como caso de estudio para aplicar este proceso al dominio de los SIS (Sistemas Integrados de Salud). Este dominio es de particular importancia en esta investigación por tres aspectos claves: por una parte será utilizado para aplicar las principales actividades, artefactos y técnicas evidenciadas en la RDS para la ingeniería de requisitos de SIS, los cuales son sistemas de información, generalmente en 4-capas (presentación, proceso, datos y comunicación), orientados a servicios, ya que utilizan en su mayoría una arquitectura basada en SOA en su capa de comunicación. Por otra parte, en este dominio no se encuentra definida claramente una LPS y las arquitecturas tipo SOA que permiten la interoperabilidad de servicios, no abarcan funcionalidades esenciales de SIS, como por ejemplo la interoperabilidad de las historias clínicas del paciente, a menos que se tenga la adopción sistemática de estándares por parte de

la comunidad de salud; este requisito es uno de los más importantes del dominio. Por lo tanto, las tareas relacionadas con la salud del ciudadano, como son, por ejemplo, la de remitir un paciente a otros centros de salud especializados, se dificultan por la falta de interoperabilidad de las historias clínicas con otros centros de salud; se recalca que solo la adopción de estándares permitiría esta facilidad. A nivel internacional, para las historias clínicas electrónicas se encuentran ya definidos algunos estándares como el HL7, pero aún no comúnmente adoptado a nivel mundial. Finalmente, a nivel nacional tampoco hay estándares adoptados que permitan realizar la integración de estos sistemas. Es de hacer notar sin embargo, que ya fue aprobada como Ley Orgánica de la Nación la Ley de Telesalud, el 12 de Agosto del 2014¹. El estudio de los requisitos planteados en esta Ley constituirá parte de las reglas del negocio a ser incluidas en el modelo del negocio del dominio de los SIS, del cual la LPSOS será derivada.

Con este caso de estudio se pretende demostrar las ventajas y/o fortalezas, y/o debilidades del proceso definido, el cual permitirá una primera validación del proceso.

AGRADECIMIENTOS

Este trabajo ha sido parcialmente financiado por los proyectos grupales DISofT No. 2011001343 del PEII-Fonacit; DesCCaP No. PG-03-8014-2011 y DARGRAF No. 03-8730-2013-1 del CDCH; además, por el Postgrado en Ciencias de la Computación, Facultad de Ciencias, Universidad Central de Venezuela y el Banco Central de Venezuela.

REFERENCIAS

[1] P. Istoan, G. Nain, G. Perrouin, and J. Jezequel, *Dynamic Software Product Lines for Service-Based Systems*, in proceedings of the 2009 Ninth IEEE International Conference on Computer and Information Technology (CIT'09), Xiamen, China, October 2009.

[2] P. Istoan, *Software Product Lines for Creating Service Oriented Applications*, Masters internship at IRISA Rennes Research Institute, TRISKELL Research Team, Rennes, France, June 2009.

[3] F. M. Medeiros, E. S. de Almeida, and S. R. de Lemos Meira, *Designing a Set of Service-Oriented Systems as a Software Product Line*, in proceedings of the Fourth Brazilian Symposium on Software Components, Architectures and Reuse (SBCARS), pp. 70-79, Salvador, Bahia, Brazil, September 2010.

[4] S. Apel, D. Batory, C. Kästner, and G. Saake, *Feature-Oriented Software Product Lines*, Springer, 2013.

[5] Q. Munir and M. Shahid, *Software Product Line: Survey of Tools*, Master's thesis, Linköping University, Department of Computer and Information Science, 2010.

[6] B. Mohabbati, M. Asadi, D. Gašević, M. Hatala, and H. Müller, *Combining Service-Oriented and Software Product Line Engineering: A Systematic Mapping Study*, Information and Software Technology, vol. 55, no. 11, pp. 1845-1859, 2013.

[7] R. Dos Santos and M. Fantinato, *The Use of Software Product Lines for Business Process Management: A Systematic Literature Review*, Information and Software Technology, vol. 55, pp. 1355-1373, 2013.

[8] K. Pohl, G. Bockle, and F. van der Linden, *Software Product Line Engineering: Foundations, Principles, and Techniques*, Springer, 2005.

[9] Wide Web Consortium, *Web Services Architecture Requirements*, W3C Working Group Note, 11 February, 2004.

[10] J. Nicolai, *SOA in Practice: The Art of Distributed System Design*, O'Reilly, Sebastopol, California, USA, 2007.

[11] M. Galster, P. Avgeriou, and D. Tofan, *Constraints for the Design of Variability-Intensive Service-Oriented Reference Architectures—An Industrial Case Study*, Information and Software Technology, vol. 55, no. 2, pp. 428-441, 2013.

[12] M. Rosen, B. Lublinsky, K. Smith, and M. Balcer, *Applied SOA: Service-Oriented Architecture and Design Strategies*, Wiley & Sons, 2008.

[13] J. Estefan, K. Laskey, F. McCabe, and P. Thornton, *Reference Architecture Foundation for Service Oriented Architecture Version 1.0*, OASIS Committee Draft, vol. 2, no. 14, 2009.

[14] M. Abu-Matar and H. Gomaa, *An Automated Framework for Variability Management of Service-Oriented Software Product Lines*, in proceedings of the IEEE 7th International Symposium on Service Oriented System Engineering (SOSE), pp. 260-267. San Francisco, California, USA, March 2013.

[15] A. Ezenwoke, S. Misra, and M. Adigun, *An Approach for e-Commerce On-Demand Service-Oriented Product Line Development*, Acta Polytechnica Hungarica, vol. 10, no. 2, 2013.

[16] P. G. G. Queiroz and R. T. V. Braga, *Uma Abordagem de Desenvolvimento de Linha de Produtos com uma Arquitetura Orientada a Serviços*, Master's thesis, ICMC-Universidade de Sao Paulo, Sao Carlos, Sao Paulo, Brazil, November 2009.

[17] S. Günther and T. Berger, *Service-Oriented Product Lines: Towards a Development Process and Feature Management Model for Web Services*, in proceedings of the 12th International Software Product Line Conference (SPLC'08), pp. 131-136, Limerick, Ireland, September 2008.

[18] J. Lee and G. Kotonya, *Combining Service-Oriented with Product Line Engineering*, Software, IEEE, vol. 27, no. 3, pp. 35-41, 2010.

[19] C. Parra, X. Blanc, and L. Duchien, *Context Awareness for Dynamic Service-Oriented Product Lines*, in proceedings of the 13th International Software Product Line Conference (SPLC'09), pp. 131-140, San Francisco, California, USA, August 2009.

[20] H. J. González, *Integration of Quality Attributes in Software Product Line Development*, Master Tesis en Ingeniería del Software, Métodos Formales y Sistemas de Información (ISMFSI), 2012.

[21] W. Suryn, *Software Quality Engineering: A Practitioner's Approach*, IEEE, Published by John Wiley & Sons, Inc., Hoboken, New Jersey, 2014.

[22] S. Wagner, *Software Product Quality Control*, Springer, 2013.

[23] B. Mohabbati, D. Gasevic, M. Hatala, and H. Muller, *A Quality Model and Evaluation Method for Service-Oriented Software Product Line and Configurable Business Processes*, ACM Transactions on Software Engineering Methodology, September 2013.

[24] L. O'Brien, L. Bass, and P. Merson, *Quality Attributes and Service-Oriented Architectures*, Software Engineering Institute, Carnegie Mellon University, 449, September 2005.

[25] D. Björner, *Software Engineering 3: Domains, Requirements, and Software Design*, Texts in Theoretical Computer Science, Springer-Verlag Berlin Heidelberg, 2006.

[26] B. Kitchenham, *Guidelines for Performing Systematic Literature Reviews in Software Engineering*, Version 2.3, EBSE Technical Report, Software Engineering Group, School of Computer Science and Mathematics, Keele University, UK and Department of Computer Science, University of Durham, UK, 2007.

[27] I. Reinhartz-Berger, A. Sturm, T. Clark, S. Cohen, and J. Bettin, *Domain Engineering. Product Lines, Languages and Conceptual Models*. Springer, 2013.

[28] G. Kotonya, J. Lee, and D. Robinson, *A Consumer-Centred Approach for Service-Oriented Product Line Development*, in proceedings of Working IEEE/IFIP Conference on Software Architecture & European Conference on Software Architecture (WICSA/ECSA 2009), pp. 211-220, Cambridge, UK, September 2009.

[29] Q. Gu and P. Lago, *On Service-Oriented Architectural Concerns and Viewpoints*, in proceedings of Working IEEE/IFIP Conference on Software Architecture & European Conference on Software Architecture (WICSA/ECSA 2009), pp. 289-292, Cambridge, UK, September 2009.

¹ <http://conocimientolibre.cenditel.gob.ve/files/2014/05/LEY-TELESALUD.pdf>

- [30] M. Asadi, B. Mohabbati, N. Kaviani, D. Gašević, M. Bošković, and M. Hatala, *Model-Driven Development of Families of Service-Oriented Architectures*, in proceeding of the First International Workshop on Feature-Oriented Software Development (FOSD'09), pp. 95-102, Denver, Colorado, USA, October 2009.
- [31] M. Allauddin, F. Azam, and M.J. Zia, *A Survey of Quality Assurance Frameworks for Service Oriented Systems*, International Journal of Advancements in Technology, vol. 2, no. 2, pp. 188-198, 2011.
- [32] ISO/IEC 25010. Software Engineering - *Software Product Quality Requirements and Evaluation (SQuaRE) - Quality Model*, International Organization for Standardization (ISO), 2010.
- [33] J. Herrera, F. Losavio, and A. Matteo, *Revisión Documental Sistemática de Enfoques y Técnicas para la Construcción de Arquitecturas en un Contexto de Líneas de Productos de Software*, Primera Conferencia Nacional de Computación, Informática y Sistemas (CoNCISa 2013). Naiguatá, Venezuela, Octubre 2013.
- [34] P. Dolog, *Engineering Adaptive Web Applications*, Doctoral Dissertation, University of Hanover, Germany, 2006.
- [35] A. Lapouchnian, *Exploiting Requirements Variability for Software Customization and Adaptation*, Doctoral Dissertation, University of Toronto, Canada, 2011.
- [36] C. Parra, *Towards Dynamic Software Product Lines: Unifying Design and Runtime Adaptations*, Doctoral Dissertation, Université des Sciences et Technologie de Lille, Lille I, France, 2011.
- [37] D. Smith and G. Lewis, *Service-Oriented Architecture and Software Product Lines: Pre-Implementation Decisions*, in proceedings of the 3rd International Workshop on Service Oriented Architectures and Product Lines (SOAPL'09), pp. 166-171, San Francisco, California, USA, September 2009.
- [38] M. Bošković, D. Gašević, B. Mohabbati, M. Asadi, M. Hatala, N. Kaviani, and E. Bagheri, *Developing Families of Software Services: A Semantic Web Approach*, Journal of Research & Practice in Information Technology, vol. 43, no. 3, 2011.
- [39] J. Lee, D. Muthig, and M. Naab, *A Feature-Oriented Approach for Developing Reusable Product Line Assets of Service-Based Systems*, Journal of Systems and Software, vol. 83, no. 7, pp. 1123-1136, 2010.
- [40] A. Trendowicz and T. Punter, *Quality Modeling for Software Product Lines*, in proceedings of the 7th ECOOP Workshop on Quantitative Approaches in Object-Oriented Software Engineering (QAOOSE'03), 2003.
- [41] A. Goeb and K. Lochmann, *A Software Quality Model for SOA*, in proceedings of the 8th International Workshop on Software Quality (WoSQ'11), pp. 18-25, Szeged, Hungary, September 2011.
- [42] S. Montagud, *Un Método para la Evaluación de la Calidad de Líneas de Productos Software basado en SQuaRE*, Master's Thesis, (In Spanish), Master en Ingeniería del Software Metodos Formales y Sistemas de Información. Universidad Politécnica de Valencia, Spain, 2009.
- [43] S. Segura, D. Benavides, A. Ruiz-Cortés, and P. Trinidad, *Open Source Tools for Software Product Line Development*, Open Source and Product Lines, 2007.
- [44] M. Acher, P. Collet, P. Lahire, and R. B. France, *Familiar: A Domain-Specific Language for Large Scale Management of Feature Models*, Science of Computer Programming, vol. 78, no. 6, pp. 657-681, 2013.
- [45] L. Baresi, S. Guinea, and L. Pasquale, *Service-Oriented Dynamic Software Product Lines*, Computer, vol. 45, no. 10, pp. 42-48, 2012.
- [46] J. Lee, D. Muthig, and M. Naab, *An Approach for Developing Service Oriented Product Lines*, in proceedings of the 12th International Software Product Line Conference (SPLC'08), Limerick, Ireland, September 2008.
- [47] M. Njima, M. ter Beek, and S. Gnesi, *Product Line Architectures for SOA*, in proceedings of the 2011 International Conference on Software Engineering Research and Practice (SERP'11), Las Vegas, Nevada, USA, July 2011.
- [48] F. Losavio, and A. Matteo, *Reference Architecture Design Using Domain Quality View*, Journal of Software Engineering & Methodology, vol. 3, no. 1, 2013.