

Revisión Documental Sistemática de Enfoques y Técnicas para la Construcción de Arquitecturas en un Contexto de Líneas de Productos de Software

Juan Herrera¹, Alfredo Matteo², Francisca Losavio²
jchr1982@gmail.com, alfredojose.matteo@gmail.com, francislosavio@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

Resumen: Las líneas de productos de software son un enfoque centrado en el desarrollo de software basado en la reutilización, mediante el proceso de ingeniería de líneas de productos y particularmente en la fase de ingeniería del dominio es donde se diseña la arquitectura de referencia y/o la arquitectura de líneas de productos, para ello dos enfoques son utilizados en la industria, partiendo de un amplio conocimiento del dominio de la aplicación se encuentra el enfoque proactivo y partiendo de los sistemas existentes se encuentra el enfoque reactivo, es de particular interés identificar los artefactos utilizados en las estrategias utilizadas por ambos enfoques diferenciándolos ya sea por el tipo de reutilización seleccionado o por la incorporación de nuevos artefactos en cada uno de los métodos propuestos, en consecuencia una revisión documental sistemática de los estudios que abordan el tema es indispensable para determinar los alcances y limitaciones de las propuestas encontradas.

Palabras Clave: Líneas de Productos de Software; Arquitectura de Referencia; Arquitectura de Líneas de Productos; Proactivo; Reactivo; Revisión Documental Sistemática.

Abstract: The software product lines are a software development approach based on reuse, through the process of product lines engineering; particularly in the domain engineering phase is where the reference architecture and/or product line architecture is designed. For this purpose, two approaches are commonly used in industry, one based on extensive knowledge of the application domain or proactive approach, and the other one, or reactive approach, based on building on existing systems; in particular, it is interesting to identify the artifacts used in the strategies employed by both approaches, differentiating either for the selected type of reuse or incorporation of new artifacts in each of the proposed methods. In consequence, a systematic literature review of studies that address this issue is essential to determine the scope and limitations of the proposals found.

Keywords: Software Product Line; Reference Architecture; Product Line Architecture; Proactive; Reactive; Systematic Literature Review.

I. INTRODUCCIÓN

El enfoque denominado LPS (Líneas de Productos de Software) se centra en el desarrollo de software basado en la reutilización, que se fundamenta en la identificación y captura de las similitudes y variabilidades de productos de software dentro de un dominio determinado [1]; con ello se pretende reducir el tiempo, esfuerzo, costo, complejidad y una mejora de la productividad en el desarrollo y mantenimiento de sistemas de software complejos [2].

Es a través de la ILPS (Ingeniería de Líneas de Productos de Software) donde la reutilización es planificada y aplicada [1]. La ILPS ha demostrado ser un paradigma para el desarrollo de una diversidad de sistemas intensivos de software utilizando

tecnologías existentes para la construcción a gran escala de productos adaptados a las necesidades individuales de los clientes a un menor costo, en menor tiempo y con mayor calidad [1].

El paradigma de ILPS comprende dos grandes procesos [1]: ID (Ingeniería del dominio) que es responsable de establecer la plataforma reutilizable y de definir las similitudes y variabilidades de la línea de productos. IA (Ingeniería de la aplicación) que se encarga de derivar aplicaciones de líneas de productos a partir de la plataforma establecida en la ingeniería de dominio.

Las actividades incluidas en el proceso de ID son la gestión del producto, la ingeniería de requisitos del dominio, el diseño del

dominio, la realización del dominio y las pruebas del dominio [3]. La gestión del producto se ocupa de los aspectos económicos, en particular de las estrategias de mercado relacionadas con las líneas de productos. La ingeniería de requisitos del dominio es responsable de la identificación de los requisitos comunes y variables. El diseño del dominio define la arquitectura de referencia de la línea de productos proporcionando una estructura común de alto nivel para todas las aplicaciones derivadas. La realización del dominio se ocupa del diseño detallado y la implementación de componentes de software reutilizables. Las pruebas del dominio revelan la evidencia de los defectos en los artefactos del dominio y crea artefactos de prueba reutilizables para la evaluación de las aplicaciones. De allí, para el propósito de esta investigación, solo es de interés la actividad relacionada con el diseño del dominio y particularmente con el artefacto que se deriva de esta actividad que es la *arquitectura del dominio*. El *dominio* es un campo de estudio que define un conjunto de requisitos comunes, la terminología y la funcionalidad de cualquier programa de software construido para resolver un problema.

En este contexto, se pueden encontrar dos tipos de arquitecturas de software para el dominio: AR (Arquitectura de Referencia) y ALP (Arquitectura de Líneas de Productos). Ambas apuntan a mejorar el desarrollo del software mediante la estandarización de las arquitecturas de un conjunto de sistemas de software [4].

En general, una *arquitectura de referencia* acumula conocimiento de un dominio, identifica soluciones abstractas de un problema y promueve la reutilización de las experiencias en diseño alcanzando una sólida y bien reconocida comprensión y conocimiento de un dominio, ofreciendo soluciones estandarizadas para un dominio de aplicación más amplio [4][5]. Es generalmente construida siguiendo un enfoque denominado top-down, descendente o proactivo [6].

Teniendo en cuenta la importancia de las ARs, diferentes dominios de aplicaciones, como el automotriz, la aeronáutica y la robótica, han propuesto y utilizado estas arquitecturas [4].

La *arquitectura de líneas de productos*, en cambio, es más especializada, centrándose en un subconjunto específico de los sistemas de software existentes de un dominio y ofrecen soluciones estandarizadas para una familia pequeña de sistemas [4], lo que permite crear sistemas de forma más confiable y rentable [2].

Una diferencia esencial es que las ALPs tratan con la variabilidad entre los productos, y se sitúan en un nivel de abstracción más bajo en comparación con las ARs [4].

La ALP es también una arquitectura genérica, pero es construida a partir de productos existentes, por lo tanto “hereda” de cierta forma las propiedades arquitectónicas, por ejemplo el estilo arquitectónico, y las funcionalidades relevantes de cada producto considerado. Por eso es construida mediante un enfoque denominado bottom-up, ascendente, extractivo o reactivo [6][7], que considera técnicas de reingeniería en diferentes etapas del desarrollo. Es considerado un enfoque útil en la práctica, cuando en la industria no se tiene una LPS y se quiere construir una, a partir de productos que funcionan y han sido elaborados anteriormente; es a partir de ellos que se quiere generalizar y construir la ALP. Las ARs

están generalmente situadas en un nivel de abstracción más alto en comparación con las ALPs [4].

A partir de aquí, el análisis y alcance del desarrollo de la AR y ALP en esta investigación se caracteriza por identificar los métodos y los artefactos que se utilizan y/o reutilizan para construir la arquitectura (AR o ALP) a partir de dos tipos de enfoques: proactivo y reactivo [7], su justificación se basa en las nuevas tendencias en el desarrollo de las líneas de productos como medio para la personalización masiva del software y de allí los dos enfoques para la construcción de las arquitecturas, estos modelos son: el enfoque proactivo, y el enfoque reactivo [6][7].

En consecuencia el interés de este trabajo consiste en considerar la AR o ALP, centrándose en los enfoques proactivo y reactivo.

El objetivo principal de este trabajo es explorar, organizar, resumir y analizar las principales tendencias que se han propuesto o utilizado en la construcción de las ARs y/o ALPs. Para ello, se ha adoptado y llevado a cabo una revisión sistemática bibliográfica, mediante la guía proporcionada por Kitchenham para estudiar y analizar la literatura en el campo de la ingeniería del software [8][9][10].

Este trabajo está estructurado de la siguiente manera, además de esta introducción y las conclusiones: en la Sección II se presentan los enfoques y modelos 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 Y MODELOS A CONSIDERAR

A. Personalización Masiva de Software y sus Beneficios

La *personalización masiva* del software es un enfoque que se concentra en los medios de producción eficiente y el mantenimiento de múltiples productos de software similares, aprovechando lo que tienen en común y gestionando las variantes entre ellos. El objetivo es que el enfoque cambie del desarrollo y mantenimiento de varios productos distintos, hacia el desarrollo y mantenimiento de una sola LPS [7]. Cabe decir, que la personalización masiva es la producción a gran escala de productos adaptados a las necesidades individuales de los clientes [1].

B. Enfoques para la Adopción de la Personalización Masiva de Software

La transición de las organizaciones a la personalización masiva de software puede lograrse con uno de los tres enfoques de amplia adopción: proactivo, reactivo, y extractivo [7].

El *enfoque proactivo* es apropiado cuando los requisitos para el conjunto de los productos necesarios y futuros, están bien definidos y estables. Este enfoque requiere un esfuerzo considerablemente para la organización, pero esto cae bruscamente una vez que la línea de producción se ha completado [1][7]. Las tareas de alto nivel que se llevan a cabo son: a) realizar un análisis de dominio y del alcance para identificar la variación que será soportada en la línea de producción; b) modelar la arquitectura de la línea de productos para soportar todos los productos en la línea de producción y c) diseñar las partes comunes y variables del sistema [7].

En el *enfoque reactivo*, la organización aumenta su línea de productos debido a la demanda de nuevos productos o cuando surgen nuevos requisitos a los productos existentes. Las tareas de alto nivel que se llevan a cabo son: a) caracterizar los requisitos para el nuevo producto en relación con los actualmente soportados en la línea de producción y b) si el nuevo producto está dentro del alcance de la línea de producción actual, se crea la definición del producto, en caso contrario, se amplía el alcance de la línea de producción actual para incluir los nuevos requisitos [7].

En el *enfoque extractivo*, la organización aprovecha los sistemas existentes de software mediante la extracción de lo común y lo diverso del código fuente, para sintetizarlos en una sola línea de producción. Requiere utilizar técnicas de reingeniería para reconstruir la arquitectura del sistema a partir del código fuente, o de la documentación existente. Las tareas de alto nivel que se llevan a cabo son: a) identificar elementos comunes y la variación en los sistemas existentes y b) factorizar en una sola línea de productos [7].

En esta investigación solo se utilizarán los enfoques proactivo y reactivo. El enfoque extractivo es considerado como similar al enfoque reactivo, ya que en ambos se reutilizan los sistemas existentes [1]: sistemas de software ad hoc y/o LPS presentes en la organización.

Las tareas que van a servir de base en el proceso de la toma de decisiones para determinar en cada uno de los estudios si el método de construcción de la arquitectura utiliza el enfoque proactivo (similar al enfoque top-down, según el proceso de desarrollo de software tradicional [11]) o por el contrario, utiliza el enfoque reactivo (similar al enfoque bottom-up [11]) son:

Se considera que el método pertenece al enfoque proactivo si describe las siguientes tareas en este orden: 1) análisis del dominio, 2) definición del alcance determinado por la variedad de productos que la organización considera que van pertenecer a ese dominio y 3) el modelado de la arquitectura.

Se considera que el método pertenece al enfoque reactivo si describe las siguientes tareas en este orden: 1) elige el conjunto de sistemas existentes que formarán parte de la línea de productos, 2) identifica las similitudes y diferencias entre los sistemas existentes y 3) el modelado de la arquitectura.

III. REVISIÓN DOCUMENTAL SISTEMÁTICA

La revisión sistemática se realizó en el contexto de la AR y ALP, con el objetivo de identificar todos los estudios relevantes relacionados con los métodos de construcción de estas arquitecturas. Se llevó a cabo en el primer trimestre del 2013. 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 [8][9][10].

En esta revisión documental sistemática se pretende conocer y comprender los enfoques y/o técnicas de diseño de las ARs y ALPs, particularmente el interés se centra en si aplican el enfoque proactivo o reactivo en la construcción de estas arquitecturas, además, de determinar el nivel de reutilización con relación a métodos y artefactos arquitecturales tradicionales o por el contrario si los métodos describen nuevos métodos y/o artefactos para la construcción de estas. Por

último identificar las similitudes y diferencias entre las ARs y ALPs.

En la Tabla I se describen las fases, las etapas que incluyen cada fase y los artefactos generados en el proceso de revisión sistemática.

Tabla I: 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:* Como parte de un proyecto de investigación mayor que investiga la identificación y caracterización de las ARs y ALPs debido a que ambas arquitecturas tienen definidos sus roles en lo que respecta a objetivos específicos de las arquitecturas de software y en particular para el desarrollo de las LPS. La identificación y comprensión de enfoques y/o técnicas existentes que apoyan la construcción de estas arquitecturas es relevante para comprender sus alcances y limitaciones.

2) *Desarrollo del Protocolo:* El protocolo aborda 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. En otras palabras, el protocolo es un plan o conjunto de pasos a seguir en un estudio.

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 para construir las ARs y ALPs, las siguientes PIs (Preguntas de Investigación) son consideradas:

- P11. ¿Cuáles son los métodos utilizados para el diseño de las ARs y ALPs tomando en consideración el enfoque proactivo (top-down) o reactivo (bottom-up) como estrategias para el diseño de las arquitecturas?
- P12. ¿Los métodos reutilizan o adaptan algún patrón, estilo o modelo arquitectónico tradicional o describen nuevos métodos o artefactos para la construcción de las arquitecturas?

- PI3. ¿Cuáles son las diferencias encontradas entre las ARs y las ALPs?

Adicionalmente, otras preguntas fueron formuladas:

- PI2.1. ¿Cuál es la forma o estrategia de reutilización más utilizada?
- PI2.2. ¿Cuál de los dos modelos (proactivo y reactivo) es el más utilizado?
- PI2.3. ¿Cuáles otros aspectos fueron evidenciados en los métodos analizados?

Con relación a la estrategia de búsqueda, primero se establecieron las fuentes de búsqueda para encontrar los estudios primarios, particularmente se seleccionó el motor de búsqueda ResearchGate¹ como la única fuente de búsqueda por ser una herramienta de investigación científica, que permite a los investigadores buscar contenido en revistas arbitradas, conferencias, workshop y repositorios institucionales con información relevante para la realización de la revisión sistemática.

Para establecer la estrategia de búsqueda, teniendo en cuenta las preguntas de investigación, inicialmente se identificaron las principales palabras clave “Reference Architecture” y “Product Line Architecture”. En la Tabla II se encuentran todas las expresiones de búsqueda utilizadas en la revisión sistemática.

Tabla II: Expresiones de Búsqueda en la Revisión Sistemática

Fuente consultada = “http://www.researchgate.net”		
Expresiones de búsqueda	Hits	Aceptados
(“product line architecture” OR “product line architectures”) AND (“approach”)	91	29
(“product line architecture” OR “product line architectures”) AND (“method” OR “methods”)	48	
(“product line architecture” OR “product line architectures”) AND (“proactive”)	5	
(“product line architecture” OR “product line architectures”) AND “reactive”	2	
(“product line architecture” OR “product line architectures”) AND “top-down”	0	
(“product line architecture” OR “product line architectures”) AND “bottom-up”	0	
Total	146	
(“reference architecture” OR “reference architectures”) AND (“approach”)	210	40
(“reference architecture” OR “reference architectures”) AND (“method” OR “methods”)	100	
(“reference architecture” OR “reference architectures”) AND “proactive”	1	
(“reference architecture” OR “reference architectures”) AND “reactive”	2	
(“reference architecture” OR “reference architectures”) AND “top-down”	3	
(“reference architecture” OR “reference architectures”) AND “bottom-up”	3	
Total	319	
Total general	465	69

Con relación a los criterios de inclusión y exclusión. 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 CIs (Criterios de Inclusión) que se utilizan para incluir los estudios relevantes en esta revisión sistemática fueron:

- CI1: El estudio define AR y/o ALP en el ámbito de las LPS.
- CI2: El estudio propone o emplea un método para construir las ARs y/o ALPs en el ámbito de las LPSs.

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: El estudio no propone o utiliza el modelo proactivo (top-down) y/o reactivo (bottom-up) para construir las ARs y/o ALPs;
- CE2: El estudio donde su texto completo no esté disponible o solo aparece el resumen; y
- CE3: El estudio que esté escrito en un idioma diferente al inglés o español.

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.

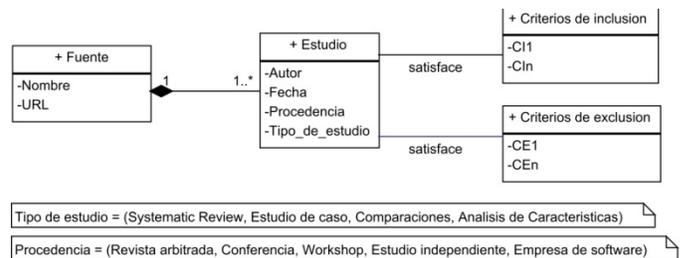


Figura 1: Modelo de Datos Inicial para la Revisión Sistemática

De las 69 publicaciones definitivamente 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 representado en la Figura 2, y que incluye la lectura de sus contenidos. 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.

¹ <http://www.researchgate.net>

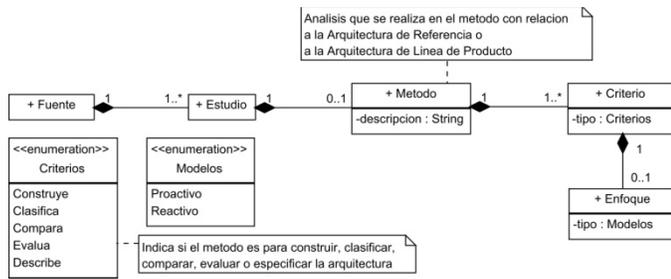


Figura 2: Modelo de Datos para el Vaciado de los Datos

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, tomando en cuenta el modelo de datos de la Figura 2.

Tabla III: Formato para la Recolección de Datos

Criterio	Estudio		Método	TA	Asuntos de interés en relación a los artefactos utilizados por el método		
	Año	Ref			Reutiliza y adapta	Reutiliza tal cual	Agrega artefacto

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

Tabla IV: Propiedades de Extracción

Propiedad	Valor	Descripción
Criterios	Construye	Construyen la arquitectura bajo el enfoque reactivo (bottom-up); o proactivo (top-down).
	Clasifica	Clasifican las arquitecturas
	Compara	Comparan métodos de construcción arquitectural.
	Evalúa	Evalúan la calidad arquitectural.
	Especifica	Describen formalmente la arquitectura.
Estudios	Año	Año de publicación del estudio.
	Ref.	Referencia bibliográfica y/o webgráfica.
Método	Nombre	Nombre del método identificado en el estudio.
TA	Tipo de arquitectura	Indica si el método se basa en la AR o ALP.
Asuntos de interés en relación a los artefactos utilizados por el método	Reutiliza y adapta	Indica si el método, técnica, patrón, estilo, estándar o framework es reutilizado y adaptado para construir la arquitectura.
	Reutiliza tal cual	En este caso los artefactos son reutilizados sin modificarlos para construir la arquitectura.
	Agrega artefacto	En este caso se crea o propone algún nuevo método, técnica, patrón, estilo, estándar o framework para construir la arquitectura.

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 en las base de datos de publicaciones con las expresiones de búsquedas, 465 estudios fueron localizados, de los cuales solo 69 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 y porcentajes de los estudios y que a manera de resumen fueron

agrupados en los dos elementos fundamentales comunes en los criterios de búsqueda expresados en la Tabla II.

Tabla V: Relación de Estudios por Criterios de Búsqueda

Criterios de búsqueda	Artículos			
	Incluidos	Excluidos	Total de aciertos	% Incluidos
Reference Architecture	40	279	319	57.97
Product Line Architecture	29	117	146	42.03
Total	69	396	465	

La Tabla VI presenta la relación de los estudios cuantificados por año y tipo de publicación, de los 69 estudios que se consideraron pertinentes para esta revisión documental sistemática.

Tabla VI: Relación de Estudios por Años y Tipos de Publicación

Años	Tipo de publicación			
	Revistas	Conferencias y talleres	Otros ²	Total
1998	1	1		2
1999	3	1	1	5
2000	1		1	2
2001	1		1	2
2002	1	2	1	4
2003	3		1	4
2004	3	3	1	7
2005	1	6	1	8
2006	2		1	3
2007	2	2		4
2008	3	1	1	5
2009	1	6	1	8
2010	1	1	1	3
2011	2	5	1	8
2012	2	1	1	4
Total	27	29	13	69

Resalta el hecho que desde el año 2004 hasta la fecha, es donde se concentra el mayor número de investigaciones, observándose un gran interés en los investigadores del área de la ingeniería de software en realizar estudios relacionados con las ARs y ALPs para LPS.

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.

1) *Con Relación a la Pregunta PII: ¿Cuáles métodos utilizan como estrategia el enfoque proactivo o el enfoque reactivo para el diseño de la AR y ALP?*

La Tabla VII resume los enfoques o técnicas que abordan el proceso de construcción de las arquitecturas desde la perspectiva del enfoque reactivo o *bottom-up*.

² Otros se refiere a los repositorios que tienen algunas universidades donde almacenan las investigaciones de sus estudiantes

Tabla VII: Construyen la Arquitectura (Enfoque Reactivo)

E ³	Método (autor(es)) (ref.)	TA ⁴
E1	Extra view to the QADA (Tan et al., 2012) [12]	ALP
E2	Framework RA aplicando SOA (Duro et al., 2005) [13]	AR
E3	RA based in Reverse Engineering (Grosskurth & Godfrey, 2005) [14]	AR
E4	RA basado en patrones (Mnñoz et al., 2005) [15]	AR
E5	ARES (Architecture Reference) approach [16]	AR
E6	Empirically-Grounded RA approach (Galster & Avgeriou, 2011) [17]	AR
E7	PuLSE-DSSA approach (Bayer et al., 2004) [18]	AR
E8	Prototype of reengineering approach (Jirapanthong, 2012) [19]	ALP
E9	Layered PLA approach (Bastarrica, 2006) [20]	ALP
E10	KobrA method (Anastasopoulos, 2002) [21]	ALP
E11	FAST (Family-Oriented Abstraction, Specification, and Translation) (Harsu, 2002) [22]	ALP
E12	An MDE-based PLA (Deng, 2009) [23]	ALP
E13	RA based in Prior Systems (Pinzger et al., 2004) [24]	AR
E14	Any-time variability approach (Hoek, 2004) [25]	ALP

La Tabla VIII resume los métodos que abordan el proceso de construcción de las arquitecturas desde la perspectiva del enfoque proactivo o *top-down*.

Tabla VIII: Construyen la Arquitectura (Enfoque Proactivo)

E ³	Método (autor(es)) (ref.)	TA ⁴
E15	Framework ACROSeT (Iborra et al., 2009) [26]	RA
E16	ERA (E-contracting Reference Architecture) approach (Angelov & Grefen, 2008) [27]	RA
E17	Interface-Activities Model (IAM) RA (Pollard & Duke, 2005) [28]	RA
E18	ProSA-RA (Nakagawa et al., 2009) [29]	RA
E19	A component-oriented approach centered on MDE (Panunzio, 2011) [30]	RA
E20	Cross-domain RA approach (Dobrica & Niemelä, 2008) [31]	RA
E21	ASRA (Agent System Reference Architecture) approach (Nguyen et al., 2010) [32]	RA
E22	Proactive Design Method for PLAs (Oizumi et al., 2012) [33]	ALP
E23	Decision-oriented approach (Dhungana, 2007) [34]	ALP
E24	SEPA Methodology (Barber & Graser, 1999) [35]	RA
E25	NEXOF-RA approach (Stricker et al., 2010) [36]	RA
E26	Process for Managing Variability (Kim et al., 2011) [37]	ALP
E27	UML profile approach for variations (Dobrica & Niemelä, 2007) [38]	ALP
E28	Style-specific approach (Lalanda, 1999) [39]	ALP
E29	Feature oriented approach (Zhu et al., 2007) [40]	ALP

En la Tabla IX, se resumen los resultados obtenidos en las Tablas VII y VIII, mostrando que no existe una tendencia o preferencia muy marcada en utilizar el enfoque proactivo o el reactivo, independientemente si es para diseñar la AR o ALP.

2) *Con Relación a la Pregunta PI2:* Identificar si los métodos utilizan algún patrón, estilo o modelo arquitectónico en particular como artefacto para la construcción de las ARs y ALPs, particularmente en lo referente a que artefactos reutiliza y adapta, solamente reutiliza o por el contrario agrega o incorpora algún nuevo artefacto en el proceso de construcción de estas arquitecturas.

En la Tabla IX se muestra la distribución de los estudios según el enfoque de construcción caracterizado por el tipo de arquitectura y asuntos de interés relacionados con las técnicas utilizadas por el método desde el punto de vista de la reutilización de artefactos arquitectónicos existentes.

Tabla IX: Distribución de Estudios por Enfoque de Desarrollo y Asuntos de Interés

Enfoque o modelo	Tipo de Arquitectura			Asuntos de interés en relación a artefactos utilizados en el método		
	AR ⁵	ALP ⁶	Total	Reutiliza y adapta	Reutiliza tal cual	Agrega artefacto
Proactivo	9	6	15	7	6	8
Reactivo	7	7	14	7	11	3
Total	16	13	29	14	17	11

Los artefactos son aquellas técnicas y/o elementos arquitecturales (estilo arquitectónico, patrón arquitectónico, patrón de diseño, etc.) que usan y/o implementan los métodos para la construcción de las arquitecturas.

Al realizar un análisis de las tendencias utilizadas en la construcción de las arquitecturas en los estudios analizados, según los datos suministrados por la Tabla IX, las siguientes preguntas fueron resueltas:

PI2.1. ¿Cuál es la forma de reutilización más utilizada en la construcción de estas arquitecturas?

Es evidente que la tendencia es la de reutilizar los artefactos existentes, en algunos casos se le realizan modificaciones a estos artefactos, en otros casos se incorporan en el método tal cual como originalmente fueron definidos.

PI2.2. ¿Cuál de los dos modelos (proactivo y reactivo) es el más utilizado en la construcción de estas arquitecturas?

La evidencia indica la tendencia en utilizar ambos enfoques, el proactivo y el reactivo por igual.

PI2.3. ¿Cuáles otros aspectos fueron evidenciados en los métodos analizados para la construcción de las arquitecturas?

La Tabla X muestra otras evidencias que reflejan el interés que tienen algunos autores, no solo en la construcción de las arquitecturas, sino también en clasificarlas, compararlas, evaluarlas y especificarlas formalmente.

Tabla X: Distribución de los Estudios por Criterios de Análisis

Criterios	TA		Total
	AR	ALP	
Construyen	16	13	29
Clasifican	3	1	4
Comparan	2	3	5
Evalúan	9	10	19
Especifican	0	3	3

Se destaca que los estudios [12][18][21][27][29][33][41], satisfacen dos de los criterios, particularmente describen un método para construir la arquitectura y luego la evalúan a través de características de calidad. Solo 1 estudio [42] mostró un enfoque para clasificar y evaluar las ARs.

³ Estudios

⁴ Tipo de arquitectura

⁵ Arquitectura de referencia

⁶ Arquitectura de líneas de productos

3) *Con Relación a la Pregunta "PI3": ¿Cuáles son las semejanzas y diferencias entre la AR y la ALP?*

La Tabla XI resume las similitudes y las diferencias entre ambas arquitecturas [4][12][14][16][17][19][30][31][36][43][44][45][46][47][48][49].

Tabla XI: Semejanzas y Diferencias entre AR y ALP

Semejanzas
Garantiza la interoperabilidad de los sistemas a través de la estandarización.
Facilita la creación de instancias de nuevas arquitecturas.
La flexibilidad y la interoperabilidad son dos de los requisitos no funcionales que guían su diseño.
Determinan el ámbito de la línea de productos que limita los productos que se pueden construir.
Se producen durante la actividad del diseño del dominio en la fase de ingeniería del dominio.
Comparten una arquitectura y un conjunto de componentes reutilizables entre una familia de productos.
Describe elementos básicos de la arquitectura de software para los sistemas que se derivan del mismo dominio.
Son arquitecturas de software que no contiene detalles de implementación.
Diferencias
La AR requiere la existencia de dominios bastantes maduros debido a su alto nivel de abstracción.
AR tiene mayor nivel de abstracción con respecto a la ALP.
Las ALP tienen que ver con la variabilidad entre los productos.
Las ALPs representan un grupo de sistemas que comparten elementos similares, pero producidos independientemente por una sola organización.
Las ARs representan el espectro visible y futuro (conjunto, gama, rango, variedad) de los sistemas en un dominio.
Las ALPs son menos abstractas que la RA, pero más abstractas que las arquitecturas concretas, es decir se sitúan en un nivel intermedio entre la AR y el producto concreto.
Las ALP son un tipo de arquitectura de referencia.
La ALP puede ser instanciada en una arquitectura de software mediante el refinado, extendiéndola, y en particular, especificando decisiones de implementación.
El rol de las ARs es capturar el conocimiento de las arquitecturas existentes. Sobre la base de la misión, visión y estrategias, y sobre las necesidades futuras de los clientes, y proporciona orientación a las múltiples organizaciones que evolucionan o crear nuevas arquitecturas.
La AR se crea mediante la captura de los elementos esenciales de las arquitecturas existentes y teniendo en cuenta las necesidades futuras y oportunidades, que van desde las tecnologías específicas, a los patrones de los modelos de negocio y segmentos de mercado.
La ALP especifica la descomposición de un sistema en subsistemas, la interacción entre subsistemas y la distribución de funcionalidad entre ellas.

IV. DISCUSIÓN

Los resultados de la revisión sistemática indican que, a pesar del gran interés en las AR y en las ALPs, la mayoría de los estudios aplican diferentes criterios en la forma o manera de concebir o diseñar la AR y las ALPs por lo cual ampliar la investigación en cuanto a la unificación de criterios en esta área sería de gran aporte en la comunidad de desarrolladores.

La evidencia indica que el término AR no se origina por el surgimiento de la metodología de ingeniería de líneas de productos de software como otro enfoque para la reutilización y la mantenibilidad de familias de productos de software, ya anteriormente se hablaba de las arquitecturas de referencias, esto se ve reflejado por los 25 estudios de los 69 que abordan las AR y no las relacionan con el SPL de forma explícita.

Las LPS es otro enfoque que hace énfasis en la reutilización de los activos de software presentes en la organización, la

preferencia es la de reutilizar los artefactos existentes (estilo arquitectónico, patrón, etc.), modificándolos o no, y en pocos casos incorporando nuevos artefactos en los métodos de desarrollo definidos en la organización.

De las políticas de desarrollo de software en la organización a nivel industrial, la toma de decisión de utilizar el enfoque proactivo o el reactivo dependerá de: si se quiere construir una nueva LPS sin considerar los productos existentes, modificar una LPS existente o reutilizar los productos existentes y de allí construir una nueva LPS.

Aunque hay autores que indican que la AR y la ALP son diferentes en cuanto al objetivo para el cual fueron concebidas, concretamente en el alcance del dominio determinado por la organización, en la mayoría de los estudios analizados, los autores utilizan ambos términos de forma equivalente.

V. CONCLUSIONES

La principal contribución de este trabajo fue presentar una perspectiva detallada sobre los enfoques de desarrollo proactivo y reactivo propuestos para la construcción de las AR y PLA, así como determinar la forma de reutilización de los artefactos en la construcción de estas arquitecturas.

La elección de parte de la organización del enfoque proactivo o reactivo como estrategia para el desarrollo de la arquitectura, dependerá de los tiempos de entrega de los productos y de la capacidad de inversión disponible en la organización; si la organización dispone de los recursos suficientes entonces tendrá la capacidad de abordar el enfoque proactivo que requiere gran cantidad de talento humano y de tiempo para diseñar la arquitectura; si la organización tiene tiempo en el mercado produciendo productos de software, el enfoque reactivo suele ser el elegido por esta, sin embargo se ha de tener en cuenta que los productos existentes no fueron inicialmente pensados o concebidos bajo el esquema de LPS, por lo cual el proceso de reingeniería necesario puede ser considerable, dependiendo de la madurez que exista en la organización en sus procesos de desarrollo de software y del uso de estándares de calidad en los productos presentes.

Existe una tendencia a la reutilización de artefactos existentes, como por ejemplo: estilos y patrones arquitectónicos, patrones de diseños, etc., que la incorporación de nuevos artefactos para la construcción de la AR y/o ALP.

Existe una carencia en el uso de modelos de calidad, por ejemplo el ISO 9126 o su actualización el ISO 25010, por lo cual una revisión documental sistemática que responda a la pregunta de cuáles estrategias y/o técnicas se han aplicado para proveer de calidad a las ARs y/o ALPs, y a partir de allí desarrollar un modelo de calidad para estas arquitecturas.

Esta revisión sistemática provee a la comunidad de desarrolladores de un insumo que apoye la toma de decisiones en la elección de utilizar el enfoque proactivo o reactivo para el abordaje de las LPSs como práctica organizacional para el desarrollo de sus productos de software, así como utilizar la AR o ALP como punto de partida para el diseño, dependiendo de la madurez que tenga la organización en cuanto a la metodología utilizada para el desarrollo de software y de la visión empresarial en relación a los futuros desarrollos.

REFERENCIAS

- [1] K. Pohl, G. Boeckle, and F. van der Linden, *Software Product Line Engineering - Foundations, Principles, and Techniques*, Springer Verlag, Berlin / Heidelberg, 2005.
- [2] E. Nakagawa, P. Oliveira, and M. Becker, *Exploring the Use of Reference Architectures in the Development of Product Line Artifacts*, in proceedings of the 15th International Software Product Line Conference, Vol 2, ACM, 2011.
- [3] B. Penzenstadler, *Evaluation and Testing of Approaches to Support Variants in Model-Based Design for Software Product Lines*, University of Passau, Thesis, 2006.
- [4] E. Nakagawa, P. Oliveira, and M. Becker, *Reference Architecture and Product Line Architecture: A Subtle but Critical Difference*, Software Architecture. Springer Berlin Heidelberg, pp. 207-211, 2011.
- [5] M. Guessi, L. B. R. de Oliveira, and E. Y. Nakagawa., *Representation of Reference Architectures: A Systematic Review*, Department of Computer Systems, University of São Paulo, SP, Brazil, 2011.
- [6] A. Rashid, J.C. Royer, and A. Rummler, *Aspect-Oriented Model-Driven Software Product Lines. The AMPLE Way*, Cambridge University Press, Cambridge, 2011.
- [7] C. W. Krueger, *Easing the Transition to Software Mass Customization*, in Software Product-Family Engineering, Springer Berlin Heidelberg, 2002, pp. 282-293.
- [8] 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.
- [9] B. Kitchenham, O. Brereton, D. Budgen, M. Turner, J. Bailey, and S. Linkman, *Systematic Literature Reviews in Software Engineering, a Systematic Literature Review*, Information and software technology, vol. 51, no. 1, pp. 7-15, 2009.
- [10] B. Kitchenham, *Procedures for Performing Systematic Reviews*, Keele University and National ICT Australia Ltd, Tech. Report TR/SE-0401 and NICTA TR 0400011T.1, 2004.
- [11] B. Basem El-Haik and A. Shaout, *Software Design for Six Sigma: A Roadmap for Excellence*, John Wiley and Sons, 2010.
- [12] L. Tan, Y. Lin, and H. Ye. *Quality-Oriented Software Product Line Architecture Design*, Journal of Software Engineering and Applications, vol. 5, no. 7, pp. 472-476, 2012.
- [13] N. Duro, F. Moreira, J. Rogado, J. Reis, and Nestor Peccia, *Developing a Reference Architecture for the Ground Segment Software*, IEEEAC paper #1134, Version 1.14, IEEE, 2005.
- [14] A. Grosskurth and M. W. Godfrey, *A Reference Architecture for Web Browsers*, in proceedings of the 21st IEEE International Conference on Software Maintenance (ICSM'05), pp. 661-664, 2005.
- [15] J. Muñoz, J. Muñoz, F. J. Alvarez, and M. Romo, *Aplicación de un Método para Generar una Arquitectura de Referencia que Integre Sistemas Heredados y Bases de Datos*, Universidad Autónoma de Aguascalientes, México, 2005.
- [16] W. Eixelsberger, *Recovery of a Reference Architecture, A Case Study*, in proceedings of the third International Workshop on Software Architecture, ACM, pp. 33-36, 1998.
- [17] M. Galster and P. Avgeriou, *Empirically-Grounded Reference Architectures, A Proposal*, in proceedings of the joint ACM SIGSOFT conference--QoSA and ACM SIGSOFT symposium--ISARCS on QoSA and architecting critical systems--ISARCS, ACM, pp. 153-158, 2011.
- [18] J. Bayer, T. Forster, D. Ganesan, J. F. Girard, I. John, J. Knodel, R. Kolb, and D. Muthig, *Definition of Reference Architectures Based on Existing Systems, Lifecycle and Process for Family Integration*, Fraunhofer IESE-Report no. 2004, vol. 34.
- [19] W. Jirapanthong, *Experience on Re-Engineering Applying with Software Product Line*, International Journal of Computer Engineering Science (IJCES), vol. 2, no. 5, 2012.
- [20] M. C. Bastarrica, N. Hirschfeld-Kahler, and P. O. Rossel, *Product Line Architecture for a Family of Meshing Tools*, in Reuse of Off-the-Shelf Components, Springer Berlin Heidelberg, pp. 403-406, 2006.
- [21] M. Anastasopoulos, C. Atkinson, and D. Muthig, *A Concrete Method for Developing and Applying Product Line Architectures*, in Objects, Components, Architectures, Services, and Applications for a Networked World, Springer Berlin Heidelberg, pp. 294-312, 2003.
- [22] M. Harsu, *FAST Product-Line Architecture Process*, Tampere University of Technology, 2002.
- [23] G. Deng, D. C. Schmidt, A. Gokhale, J. Gray, Y. Lin, and G. Lenz, *Evolution in Model-Driven Software Product-Line Architecture*, Designing Software-Intensive Systems: Methods and Principles, 2008.
- [24] M. Pinzger, H. Gall, J. F. Girard, J. Knodel, C. Riva, W. Pasman, C. Broerse, and J. G. Wijnstra, *Architecture Recovery for Product Families*, in Software Product-Family Engineering, Springer Berlin Heidelberg, pp. 332-351, 2004.
- [25] A. van der Hoek, *Design-Time Product Line Architectures for Any-Time Variability*, Science of Computer Programming, vol. 53, no. 3, pp. 285-304, 2004.
- [26] A. Iborra, D. Alonso, F. J. Ortiz, J. Pastor, P. Sánchez, and B. Álvarez, *Design of Service Robots - Experiences using Software Engineering*, Robotics & Automation Magazine, IEEE, vol. 16, no. 1, pp. 24-33, 2009.
- [27] S. Angelov and P. Grefen, *An E-Contracting Reference Architecture*, Journal of Systems and Software, vol 81, no. 11, pp. 1816-1844, 2008.
- [28] J. Pollard and R. Duke, *A Reference Architecture for Instructional Educational Software*, in ACM International Conference Proceeding Series, pp. 43-52, 2005.
- [29] E. Y. Nakagawa, E. F. Barbosa, and J. C. Maldonado, *Exploring Ontologies to Support the Establishment of Reference Architectures, An Example on Software Testing*, in European Conference on Software Architecture WICSA/ECSA 2009, pp. 249-252, 2009.
- [30] M. Panunzio, *Definition, Realization and Evaluation of a Software Reference Architecture*, Technical Report UBLCS-2011-07, Department of Computer Science, University of Bologna, Italy, 2011.
- [31] L. Dobrica and E. Niemelä, *An Approach to Reference Architecture Design for Different Domains of Embedded Systems*, in Software Engineering Research and Practice, pp. 287-293, 2008.
- [32] D. N. Nguyen, K. Usbeck, W. M. Mongan, C. T. Cannon, R. N. Lass, J. Salvage, W. C. Regli, I. Mayk, and T. Urness, *A Methodology for Developing an Agent Systems Reference Architecture*, in Agent-Oriented Software Engineering XI, Springer Berlin, pp. 177-188, 2011.
- [33] W. N. Oizumi, A. C. Contieri, G. G. Correia, T. E. Colanzi, S. Ferrari, I. M. S. Gimenes, E. A. Oliveira, A. F. Garcia, and P. C. Masiero, *On the Proactive Design of Product-Line Architectures with Aspects, An Exploratory Study*, in proceedings of the 36th Computer Software and Applications Conference (COMPSAC 2012), pp. 273-278, 2012.
- [34] D. Dhungana, R. Rabiser, and P. Grünbacher, *Decision-Oriented Modeling of Product Line Architectures*, in the 6th Working IEEE/IFIP Conference of Software Architecture (WICSA'07), Mumbai, India, January 2007.
- [35] K. S. Barber, T. Graser, J. Holt, and C. Silva, *Representing Domain Reference Architectures by Extending the UML Metamodel*, in 20th International Conference on Software Engineering and Knowledge Engineering, San Francisco, USA, July 1999.
- [36] V. Stricker, K. Lauenroth, P. Corte, F. Gittler, S. De Panfilis, and K. Pohl, *Creating a Reference Architecture for Service-Based Systems: A Pattern-Based Approach*, in proceeding of Towards the Future Internet - Emerging Trends from European Research, 2010.
- [37] Y. G. Kim, S. Kee Lee, and S. B. Jang, *Variability Management for Software Product-Line Architecture Development*, International Journal of Software Engineering and Knowledge Engineering, vol. 21, no. 07, pp. 931-956, 2011.
- [38] L. Dobrica and E. Niemelä, *Modeling Variability in the Software Product Line Architecture of Distributed Middleware Services*, VTT Technical Research Centre of Finland, Oulu, Finland, 2007.
- [39] P. Lalanda, *Style-Specific Techniques to Design Product-Line Architectures*, Thomson-CSF Corporate Research Laboratory, Domaine de Corbeville, Orsay, France, 1999.
- [40] C. Zhu, Y. Lee, W. Zhao, and J. Zhang, *A Feature Oriented Approach to Mapping from Domain Requirements to Product Line Architecture*, in Software Engineering Research and Practice, pp. 219-225, 2007.
- [41] J. Bayer, O. Flege, and C. Gacek, *Creating Product Line Architectures*, in Software Architectures for Product Families, Springer Berlin Heidelberg, pp. 210-216, 1999.
- [42] C. Maga and N. Jazdi, *A Survey on Determining Factors for Modeling Reference Architectures*, in proceedings of OOPSLA, Florida, USA, 2009.
- [43] G. Muller and P. van de Laar, *Researching Reference Architectures and their Relationship with Frameworks, Methods, Techniques, and Tools*, 7th Annual Conference on Systems Engineering Research (CSER 2009), S08-46, Loughborough University, UK, 2009.

- [44] P. Avgeriou, *Describing, Instantiating and Evaluating a Reference Architecture: A Case Study*, Enterprise Architect Journal, 2003.
- [45] B. Graaf, H. van Dijk, and A. van Deursen, *Evaluating an Embedded Software Reference Architecture - Industrial Experience Report*, in proceedings of the 9th European Conference of Software Maintenance and Reengineering (CSMR 2005), pp. 354-363, 2005.
- [46] U. Eklund, Ö. Askerdal, J. Granholm, A. Alminger, and J. Axelsson, *Experience of Introducing Reference Architectures in the Development of Automotive Electronic Systems*, 2005.
- [47] S. Giesecke, W. Hasselbring, and C. von Ossietzky, *Taxonomy of Architectural Style Usage*, in proceedings of the 2006 conference on Pattern Languages of Programs, ACM, 2006.
- [48] B. P. Gallagher, *Using the Architecture Tradeoff Analysis Method to Evaluate a Reference Architecture: A Case Study*, CMU/AEI-2000-TN-007, Carnegie-Mellon University, Pittsburgh Software Engineering Institute, 2000.
- [49] J. Bayer, D. Ganesan, J. F. Girard, J. Knodel, R. Kolb, and K. Schmid, *Definition of Reference Architectures based on Existing Systems*, Fraunhofer IESE-Report, vol. 34, 2004.