

Methodological hybrid SOA+CBSD for services oriented software development

Híbrido metodológico SOA + DSBC para el desarrollo de software orientado a servicios

Diego Alejandro Maldonado C.
Grupo de Investigación ARQUISOFT
Universidad Distrital Francisco José de Caldas
Bogotá, Colombia
damaldonadoc@correo.udistrital.edu.co

Alba Consuelo Nieto L.
Grupo de Investigación ARQUISOFT
Universidad Distrital Francisco José de Caldas
Bogotá, Colombia
acnietol@udistrital.edu.co

Sandra Patricia Cala D.
Grupo de Investigación ARQUISOFT
Universidad Distrital Francisco José de Caldas
Bogotá, Colombia
spcalad@correo.udistrital.edu.co

Henry Alberto Diosa
Grupo de Investigación ARQUISOFT
Universidad Distrital Francisco José de Caldas
Bogotá, Colombia
hdiosa@udistrital.edu.co

Abstract—The SOA architectural reference model has been well received since its inception, this is given their qualities of modularity and alignment with business. But even though currently are multiple publications that thoroughly develop the theory of SOA architectural reference model, is hard to find papers to deepen on the implementation of the theoretical concepts of SOA and how to realize these in a software development process that incorporates, from a practical point of view, the theoretical concepts with component-based software development methodologies and techniques to build applications that can be scalables, reusables and suitable for the changes.

This paper exposes the experiences and results about the application of a proposal to manage the development of applications conform to oriented services architectural reference model from models driven engineering, it integrates several methodologies for planning and project management, to guide the principles of software development around the business processes and services composition, it is using Component-Based Software Development (CBSD) to identify, design and to build the components and interfaces of the system that provide and require services. The proposal took a study case based on development of a Academic Management System's basic module with a persistence model dependent a distributed database system.

Keywords—*Service-Oriented Architecture (SOA), Component-based Software Development (CBSD), Unified Process Model (PUM), Petals, Academic Management.*

Resumen—El modelo de referencia arquitectural orientado a servicios (SOA en el inglés y así será citado en adelante) ha tenido gran acogida desde su surgimiento dadas sus cualidades de modularidad y natural alineación con la estrategia de negocio. Aunque en la actualidad se encuentran múltiples publicaciones que desarrollan minuciosamente la teoría del modelo de referencia SOA, es difícil encontrar trabajos que profundicen sobre la puesta en práctica de los conceptos teóricos del modelo de referencia SOA y la realización del mismo en un proceso de desarrollo de software que integre, desde un punto de vista práctico, los

conceptos teóricos con tecnologías y metodologías de desarrollo de software basado en componentes para la construcción de aplicaciones que puedan ser escalables, reutilizables y flexibles al cambio.

Este artículo presenta las experiencias y resultados de la aplicación de una propuesta para abordar el desarrollo de una aplicación conforme al modelo de referencia arquitectónico orientado a servicios desde un enfoque conducido por modelos, integrando varias metodologías para la planeación y gestión del proyecto, para guiar los principios del desarrollo alrededor de procesos de negocio y composición de servicios y usando el desarrollo de software basado en componentes (DSBC, en adelante) para identificar, diseñar e implementar los componentes e interfaces del sistema que proveen y requieren los servicios. La propuesta tomó como caso de estudio el desarrollo de un módulo básico de un Sistema de Gestión Académico de una Institución de Educación Superior (IES, en adelante) con un modelo de persistencia soportado en una base de datos distribuida.

Palabras clave – *Arquitectura Orientada a Servicios(SOA), Desarrollo de Software Basado en Componentes(DSBC), Proceso Unificado para el Desarrollo de Software(PU), Petals, Gestión Académica.*

I. INTRODUCCIÓN

En la actualidad las aplicaciones basadas en software no sólo deben construirse más rápido sino que deben ser escalables, reutilizables, fáciles de mantener y de adaptar al cambio, procurando reducir el esfuerzo, el tiempo y los costos de desarrollo. El modelo de referencia SOA (*Services Oriented Architecture*, en el inglés) se corresponde con un estilo arquitectónico que favorece principios de calidad del software como separación de responsabilidades, modularidad, abstracción e incrementalidad unido a la flexibilidad para el manejo del cambio mediante la composición de servicios [1]

. De otra parte, los componentes de software promueven una alta reutilización y construcción de aplicaciones mediante el ensamblaje de unidades de software altamente cohesivas [2].

La implementación de una solución conforme a SOA a nivel empresarial resulta compleja, terminando en algunos casos en proyectos fallidos dado que el mayor desafío no está asociado con la tecnología sino con aspectos organizacionales como la selección, diseño y control de la arquitectura, gestión de los procesos de negocio a través de la organización, diseño de servicios web reutilizables, estándares y políticas de gobernanza [3] [4]. Metodologías de desarrollo de software como la propuesta por Rosen y Lublinsky [5] aportan elementos de análisis útiles como la cadena de valor, los diagramas de contexto y el modelado de procesos de negocio. Desafortunadamente, estas herramientas metodológicas no satisfacen la necesidad de una especificación detallada de servicios y componentes de una arquitectura y aborda ligeramente aspectos para la gestión y control del proyecto.

Este trabajo ilustra, a través de un caso de estudio, la utilidad práctica de realizar un híbrido metodológico del enfoque de Rosen y sus colegas con el modelo de proceso unificado adaptado al Desarrollo de Software Basado en Componentes (DSBC, en adelante) propuesto por Cheesman y Daniels [6] para resolver el problema de la especificación arquitectónica prescriptiva y descriptiva al detalle[7]. Adicionalmente, durante el desarrollo del proyecto se aplicó el enfoque de desarrollo de software dirigido por modelos[8] empleando como lenguajes de modelado UML [9] y BPMN[10].

El caso de estudio abordado se corresponde con la ejemplificación del modelo de referencia SOA en el desarrollo de un prototipo para la gestión académica de una Institución de Educación Superior (IES, en adelante). Este problema fue postulado porque los procesos de negocio asociados a los sistemas académicos se ven enfrentados frecuentemente a cambios de diversa naturaleza como: La tasa de crecimiento estudiantil, el aumento de programas académicos ofrecidos, la transformación de los espacios físicos y actualizaciones normativas. La volatilidad de dichos sistemas revela la necesidad de proponer alternativas al tratamiento de procesos críticos de la gestión académica con el soporte de arquitecturas de software de fácil escalabilidad, modulares y de complejidad tratable[11].

Como resultado de este ejercicio académico de final de formación pregradual, se obtuvo una guía metodológica práctica para abordar el desarrollo de una solución SOA que combina dos enfoques metodológicos que resultaron ser complementarios desde la perspectiva de identificación, modelado e implementación de procesos de negocio usando servicios reutilizables con una arquitectura de software subyacente basada en componentes. La guía metodológica facilitó la gestión y control del proyecto.

II. METODOLOGÍA

Este trabajo reporta una experiencia exitosa en la combinación metodológica de un enfoque de desarrollo de software orientado a servicios [5] [12] con algunas etapas de un modelo de proceso de desarrollo de software basado en componentes en el marco del proceso unificado [6]; el caso de estudio que fue abordado se ubica en el contexto de los procesos de gestión académica de una IES. En estudio realizado sobre

sistemas de gestión académica y procesos de matrícula e inscripción de espacios académicos en algunas IES se evidenció que una característica inherente es el cambio constante para ajustarse al entorno; factores que inciden son: tasa de crecimiento estudiantil, aumento de programas académicos, transformación de espacios físicos, cambios normativos, entre otros [13][14][15][16][17] [18]. Se encontraron dos limitaciones importantes relacionadas con la actuación y evolución de estos sistemas:

- 1) Las arquitecturas de los sistemas considerados no proveen mecanismos de adaptación al cambio lo suficientemente flexibles para una labor volátil como lo es la gestión académica; esto se presenta porque las condiciones y políticas asociadas a los procesos están fuertemente acopladas con la definición funcional de los mismos, lo cual reduce la posibilidad de efectuar cambios sin afectar la corrección del sistema.
- 2) Se presentan limitaciones en el control de la ejecución de procesos secuenciales y la comunicación entre estos, dando lugar a la generación de inconsistencias en los datos y en la ejecución de los demás procesos, lo que a su vez puede afectar la confiabilidad y robustez.

Este proyecto estableció un marco de desarrollo, que se guió por el modelo de proceso unificado, combinando varias herramientas metodológicas provenientes de dos fuentes: De un enfoque que ejemplifica el modelo de referencia SOA [5] y del DSBC [6]. Estas herramientas metodológicas se incorporaron paulatinamente conforme fueron necesarias, dando como resultado un marco metodológico híbrido pragmáticamente útil para el desarrollo de proyectos SOA.

El Proceso Unificado concibe el desarrollo de artefactos de forma iterativa e incremental, es flexible y adaptable al desarrollo de una solución conforme a SOA. La metodología de desarrollo para SOA propuesta en [5] no ofrece detalles suficientes para apoyar la elaboración y construcción de la aplicación que provee los servicios de negocio, por lo cual fue necesario complementar con herramientas metodológicas propuestas en [6] para la especificación de componentes, interfaces y firmas de operaciones. La Figura 1 presenta el flujo del modelo de proceso con las herramientas metodológicas aplicadas en cada fase. Antes de iniciar se realizó la definición detallada del problema.

A. Detalles relevantes de la combinación metodológica

Como se puede observar en la Figura 1, las actividades propias de cada fase en el marco del modelo de proceso unificado han sido marcadas con dos posibles símbolos que indican la injerencia mayor de un enfoque u otro en el desarrollo y obtención de entregables de cada actividad. Cuando aparecen dos marcas se desea dar a entender que se usa un enfoque genérico de la actividad; es decir, que es aplicable desde la perspectiva de cualquiera de los dos enfoques hibridizados en este trabajo. Para el caso de estudio abordado se llevó a cabo un ciclo de pruebas funcionales alimentadas para su diseño y desarrollo con los diagramas de actividades; así mismo, se aplicaron algunas pruebas no-funcionales relacionadas con aspectos de seguridad, de rendimiento y transaccionalidad.

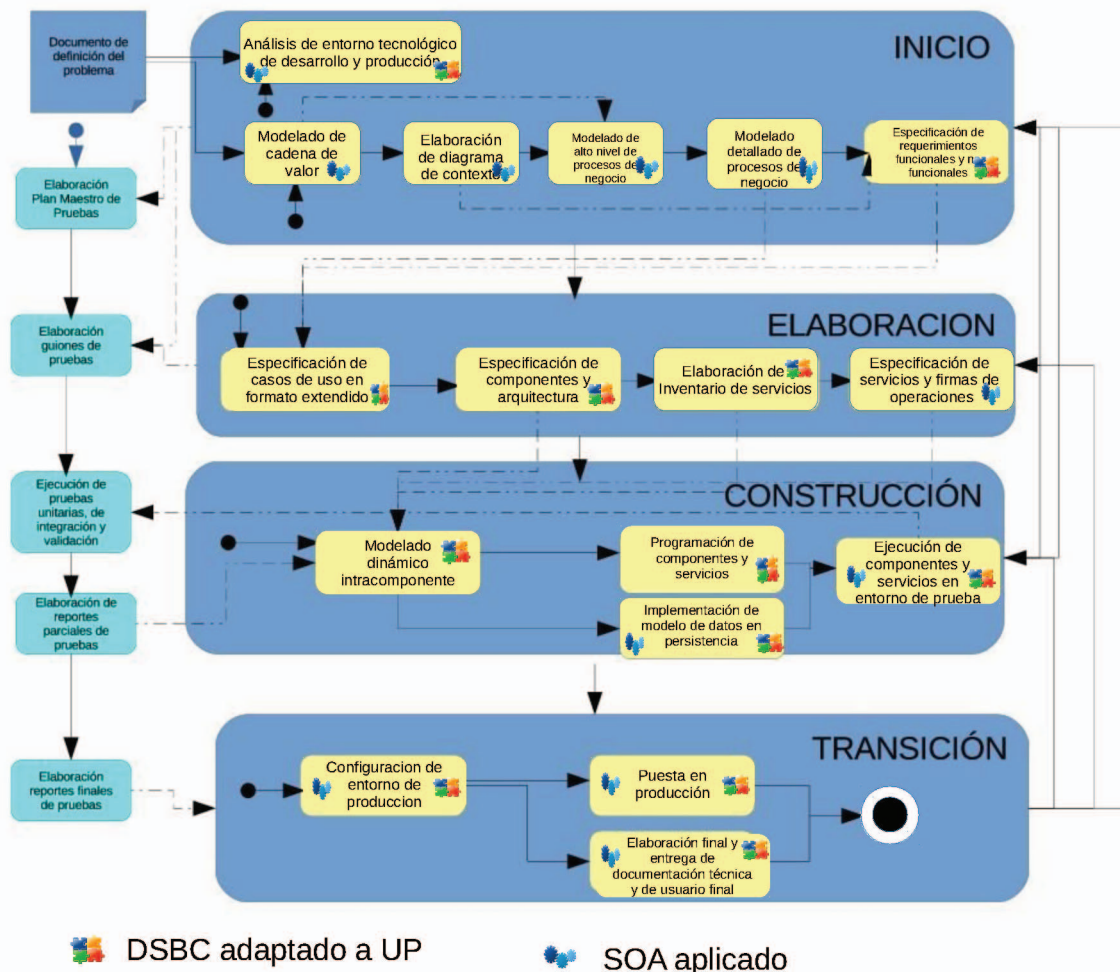


Figura 1: Ilustración de híbrido metodológico

Las siguientes secciones hacen una sucinta descripción de lo abordado por el equipo de trabajo en cada fase:

1) *La fase de inicio:* En este caso, al tratarse del desarrollo de una arquitectura SOA, se hizo énfasis en el análisis de negocio; se generaron los modelos relacionados con la cadena de valor, el diagrama de contexto y los modelos de alto nivel de los procesos de negocio usando BPMN; tal como se propone en [5]. Estas herramientas metodológicas facilitaron el análisis a profundidad de los procesos de negocio para los cuales se desarrolló la solución SOA y permitieron determinar claramente las expectativas y el impacto que ésta tendría sobre el entorno organizacional.

Adicional al reconocimiento del entorno en el que se iba a desarrollar la solución basada en software, fue necesario definir el alcance en términos de requerimientos funcionales tal como se propone en [6]. Como resultado de esta actividad se obtuvo la especificación de requerimientos funcionales asociados a los procesos de negocio de matrícula, inscripción y cancelación de espacios académicos.

2) *La fase de elaboración:* En primera medida, se especificaron los casos de uso en el formato extendido. Este nivel de detalle del modelado funcional basado en casos de

uso aplicó diagramas de actividades conformes a UML 2.0 en el modelado de escenarios. A continuación, el equipo de trabajo evidenció la necesidad de especificar la arquitectura del sistema bajo desarrollo. El enfoque SOA propuesto no satisfizo esta necesidad y se acogió como alternativa el desarrollo de una fase de especificación de componentes similar a lo propuesto por Cheesman y Daniels (Ver Figura 2) con variaciones metodológicas que la enriquecieron. Las variaciones relevantes se concentraron en aplicar el concepto de prescripción arquitectónica [7] para obtener una propuesta de ensamble de componentes prescriptivos del sistema.

Así mismo, en esta fase se asumió el modelado conceptual de las entidades de negocio y el diseño de la persistencia de datos conforme a un modelo relacional.

3) *La fase de construcción. Herramientas y estándares:* En esta fase se definieron los detalles específicos de la implementación y se construyó la primera versión del sistema. También se determinaron elementos complementarios, tales como, la estrategia de composición de servicios, notaciones de código, estándar de implementación, configuración del entorno de desarrollo, entre otros. Se culminó implementando los servicios y el modelo de persistencia conforme a una arquitectura de bases de datos distribuida.

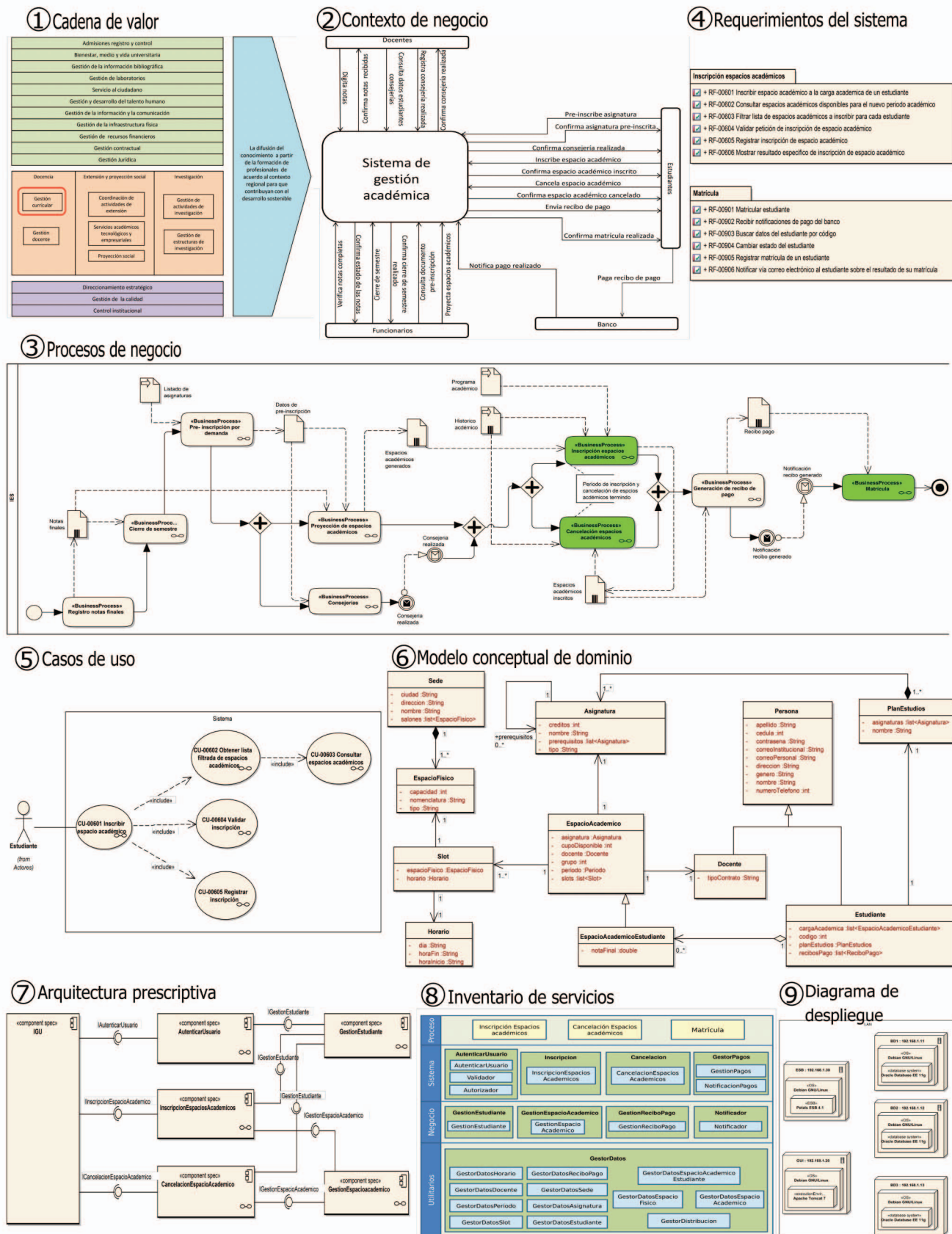


Figura 3: Secuencia resumida de artefactos de diseño entregables por fase

Fase	Artefacto	Descripción	Marco metodológico	Secuencia
Inicio	Cadena de valor	Descripción del Sistema Integrado de Gestión de una IES (procesos de soporte, procesos primarios y procesos estratégicos) .	<i>Applied SOA</i>	1
	Diagrama de Contexto	Identificación de las operaciones de negocio, interacciones e intercambio de información entre los actores y el SGA.	<i>Applied SOA</i>	2
	Modelo de procesos de negocio (a nivel general y nivel detallado)	Flujo de tareas en los procesos intersemestrales de una IES.	<i>Applied SOA</i>	3
	Modelo de requerimientos	Definición de requerimientos por módulo funcional.	Práctica convencional PU.	- 4
Elaboración	Modelo de casos de uso	Especificación de casos de uso en formato extendido.	Práctica convencional PU.	- 5
	Modelo conceptual de negocio	Especificación de las entidades de negocio y sus asociaciones.	<i>Applied SOA</i>	6
	Modelo de persistencia	Diseño del modelo relacional.	Práctica convencional PU.	-
	Modelo de componentes de negocio	Especificación de interfaces del negocio y del sistema. Modelo de ensamblaje de componentes y diseño de arquitectura prescriptiva.	DSBC	7
	Modelo de servicios	Clasificación e inventario de servicios. Firmas de operaciones.	<i>Applied SOA</i> .	8
Construcción	Código fuente y documentación de pruebas.	Implementación de componentes, servicios y composición de procesos de negocio. Implementación del modelo distribuido de persistencia. Ejecución de pruebas.	Práctica convencional PU <i>Applied SOA</i> DSBC	-
Transición	Modelo de despliegue	Configuración del entorno y puesta en producción. Documentación técnica y de usuario.	Práctica convencional PU <i>Applied SOA</i> DSBC	- 9

Tabla I: Artefactos por fase asociados a marcos metodológicos

C. Herramientas tecnológicas e implementación de servicios

La programación de los servicios se realizó sobre la plataforma de software libre *Petals SOA Suite*, versión 4 [19], criterios que guiaron la selección de esta herramienta fueron: Integración con diversas fuentes de datos, administración de servicios, manejo de estándares, lenguajes de programación soportados, facilidad de uso, nivel de documentación, soporte técnico y conceptos favorables de la comunidad desarrolladora. *Petals SOA Suite* cuenta con certificación JBI (*Java Business Integration*) por parte de *Sun Microsystems*, esto último garantizó niveles de conformidad satisfactorios con los estándares actuales (XML, WSDL, UDDI, HTTP, BPEL, JDBC, REST, entre otros). Los componentes de software que proveen los servicios de negocio fueron implementados usando el lenguaje de programación *Java 1.6*.. La interfaz gráfica de usuario fue programada usando *Java Server Faces*. El despliegue del prototipo se hizo sobre el servidor de aplicaciones *Apache Tomcat 7.0.42*.

En el aspecto tecnológico, la experiencia con las herramientas de *Petals* pudo satisfacer las expectativas frente a su facilidad de uso, un nivel de documentación suficiente y accesible para el proyecto aquí descrito. Sin embargo, cabe mencionar que el equipo de trabajo se vió abocado a algunas dificultades en el entorno de desarrollo respecto a deficiencias al crear descriptores XML de objetos complejos; también hubo problemas con el componente de mensajería por correo electrónico para establecer conexiones usando protocolos de seguridad como TLS/SSL.

Otro de los retos que se asumió en este proyecto fue la de gestionar los datos de manera distribuida; se espera socializar los resultados de esta experiencia en una futura publicación.

La persistencia de datos se implementó usando como motor *Oracle 11g Enterprise Edition* [20], que ofrece una tecnología para la comunicación entre nodos de distribución llamada *Database Link* [21].

IV. CONCLUSIONES

La implementación a nivel empresarial de una solución conforme a SOA resulta compleja, terminando en algunos casos en proyectos fallidos dado que el mayor desafío no está asociado con la tecnología sino con aspectos organizacionales como la selección, diseño y control de la arquitectura, gestión de los procesos de negocio a través de la organización, diseño de servicios web reutilizables, implementación de estándares y políticas de gobernanza [3] [4].

La Figura 1 ilustra nuestra propuesta que aplica un híbrido metodológico entre un enfoque de desarrollo de software orientado a servicios [5] y uno de desarrollo de software basado en componentes [6] para lograr una arquitectura de software conforme a SOA que puede dar soporte a procesos académicos intersemestrales en una IES. La adaptación de los enfoques metodológicos, usados en este trabajo, a un modelo de proceso incremental e iterativo como el Proceso Unificado es premisa de ambos.

La fase de inicio usó en su mayor parte actividades desde el modelo metodológico para desarrollo SOA propuesto en [5] y acogió en su parte final la especificación de requerimientos funcionales y no funcionales de la manera convencional. La complementación de esta fase con algunos modelos propios del negocio hace que el modelo funcional represente con mayor fiabilidad las funcionalidades esperadas del producto y que estas a su vez se encuentren alineadas con los objetivos mi-

sionales de la organización para la cual se pretende desarrollar una solución basada en software.

En la fase de elaboración se detectó la debilidad de la metodología de Rosen y sus colegas[5] para gestionar el proyecto centrándose en una especificación arquitectónica que naturalmente guiara al desarrollo de componentes de software. Ante este reto se formuló la posibilidad de aplicar una de las fases del modelo de proceso para desarrollo de software basado en componentes propuesto por Cheesman y Daniels en [6], en específico en la etapa de especificación de componentes (Ver Figura 2). En primera medida, esta combinación metodológica se fortaleció con una especificación de modelos dinámicos usando diagramas de secuencia intercomponente; lo anterior resultó ser útil por la riqueza semántica que ofrecen para identificar con claridad el llamado de los métodos entre interfaces de los componentes. En segundo lugar, la identificación de firmas de operaciones en cada interfaz permitió una primera aproximación a la identificación de los servicios. Y en tercer lugar, el ejercicio de ensamble de interfaces de sistema y de negocio generó la arquitectura prescriptiva que se debía implementar en alguna tecnología; en este caso dentro de un modelo de construcción conforme a SOA.

Esta experiencia documentada puede ser útil para aquellos que enfrenten la gestión de proceso de proyectos de desarrollo de software que tengan como objetivo la componentización de software que exponga servicios conforme SOA. El uso de la arquitectura prescriptiva puede guiar el orden del desarrollo de los componentes, apoyar en la identificación de operaciones candidatas a servicios y valorar los niveles de acoplamiento entre componentes del sistema.

Por último, es de interés para nuestro equipo de trabajo impulsar lo que hemos denominado *Open Models*. Este enfoque de trabajo consiste en hacer visibles y accesibles de manera libre los modelos de software de cada etapa del desarrollo en los proyectos a cargo del grupo de investigación ARQUISOFT; por esta razón, se pueden encontrar los modelos detallados de este proyecto en el portal web que apoya nuestra visibilización y gestión interna (<http://arquisoft.udistrital.edu.co/modelos/modelFenixSGAHTML>)

V. TRABAJO FUTURO

En el desarrollo del prototipo no se consideraron algunos aspectos asociados con el ciclo de vida de una solución SOA, como gobernanza, monitoreo y versionamiento de servicios; dado que el objetivo principal estuvo centrado alrededor de los aspectos metodológicos para la ejemplificación de la arquitectura. Sin embargo, con la propuesta de construcción basada en componentes de software presentada en este trabajo, el prototipo puede ser escalado fácilmente para incluir los aspectos antes mencionados.

El conjunto de servicios desarrollados y sus operaciones correspondieron directamente a las funcionalidades estrictas del prototipo; por lo cual, se propone extender el sistema desarrollando servicios adicionales que proporcionen otras funcionalidades de la gestión académica en las IES.

La composición de servicios para este prototipo se realizó utilizando código duro en Java. Lo anterior debido a que se presentaron algunos inconvenientes en la implementación de

BPEL sobre *Petals* al ejemplificar objetos complejos que contenían otros objetos complejos. Por consiguiente, en una nueva versión se trabajará en el uso de un motor de orquestación para la composición dinámica de servicios.

REFERENCIAS

- [1] C. Ghezzi, M. Jazayeri, and D. Mandrioli, "Software qualities and principles," in *Computer Science Handbook*, A. B. Tucker, Ed. N.W. Corporate Blvd., Boca Raton, Florida 33431: Chapman and Hall/CRC in Cooperation with ACM (Association for Computing Machinery), 2004, ch. 101, pp. 1–26.
- [2] C. Szyperski, D. Gruntz, and S. Murer, *Component Software. Beyond Object-Oriented Programming*. Addison Wesley. ACM press, 2002.
- [3] B. Gold-Bernstein. (2006, Apr.) The Path SOA Enlightenment. [Online]. Available: http://www.ebizq.net/blogs/bethgb/2006/04/the_path_to_soa/
- [4] N. López, R. Casallas, and J. Villalobos, "Challenges in creating environments for soa learning," Presented at the International Workshop on Systems Development in SOA Environments IEEE Computer Society. SDSOA'07, 2007.
- [5] M. Rosen, B. Lublinsky, K. T. Smith, and M. J. Balcer, *Applied SOA: Service-Oriented Architecture and Design Strategies*. Indianapolis, Indiana: Wiley Publishing, Inc., 2008.
- [6] J. Cheesman and J. Daniels, *UML Components. A Simple Process for Specifying Component-Based Software*. New York: Addison-Wesley, 2000.
- [7] R. N. Taylor, N. Medvidović, and E. M. Dashofy, *Software Architecture. Foundations, Theory, and Practice*. 111 River Street, Hoboken, NJ: John Wiley & Sons, Inc, 2010, ch. 3, pp. 58–61.
- [8] N. Markey. (2009, Jan.) MDE Model Driven Engineering reference guide. [Online]. Available: <http://www.theenterprisearchitect.eu/blog/2009/01/15/mde-model-driven-engineering-reference-guide/>
- [9] "Unified Modeling Language (OMG-UML)," Specification, Object Management Group, 2013. [Online]. Available: <http://www.uml.org/#UML2.0>
- [10] "Business Process Model and Notation (BPMN)," Specification, Object Management Group, 2011. [Online]. Available: <http://www.omg.org/spec/BPMN/2.0/>
- [11] M. P. Papazoglou, P. Traverso, S. Dustdar, and F. Leymann, "Service-oriented computing: A research roadmap," 2008.
- [12] J. Zhou and E. Niemelä, "Beyond application-oriented software engineering: Service-oriented software engineering," in *Service-Oriented Software System Engineering. Challenges and Practices*, Z. Stojanovic and A. Dahanayake, Eds. 3 Henrieta Street, Covent Garden. London WC2E 8LU: Idea Group Publishing, 2005, ch. Chapter II, pp. 27–47.
- [13] G. Castro, Jefe de Oficina Asesora de Sistemas, Universidad Distrital Francisco José de Caldas. Entrevistado por Cala S., Maldonado D., Peña D., 2012.
- [14] P. Muñoz, División de registro, Universidad Nacional de Colombia. Entrevistado por Cala S., Maldonado D., Peña D., 2012.
- [15] F. Lozano, Coordinador Programa Contacto, Pontificia Universidad Javeriana. Entrevistado por Cala S., Maldonado D., Peña D., 2012.
- [16] L. Riascos, Universidad del Valle. Correo electrónico enviado a Maldonado D., 2012.
- [17] A. Diaz, Vicedecano Facultad de Finanzas, Fundación Universitaria San Martín. Entrevistado por Cala S., Maldonado D., Peña D., 2012.
- [18] S. Cala M., D. Maldonado, and D. Peña, "Desarrollo de un prototipo de software para gestionar los procesos universitarios de matrícula e inscripción de espacios académicos usando isos (ingeniería de software orientada a servicios)," Universidad Distrital Francisco José de Caldas, Bogotá D.C., Tech. Rep. Anteproyecto, Oct. 2012.
- [19] (2015, may) Petals esb, the distributed open-source enterprise service bus. [Online]. Available: <http://petals.ow2.org/>
- [20] (2015, may) Oracle database 11g release 2. [Online]. Available: <http://www.oracle.com/technetwork/database/enterprise-edition/downloads/index.html>
- [21] (2015, may) Database links. [Online]. Available: http://docs.oracle.com/cd/B28359_01/server.111/b28310/ds_concepts002.htm#ADMIN12083