

Desarrollo de una arquitectura orientada a servicios para un prototipo de una línea de productos de software

Ramón Gómez-Romero, Karen Cortés Verdin, Juan Carlos Pérez Arriaga, Ángeles Arenas Valdés

Universidad Veracruzana, Facultad de Estadística e Informática,
Xalapa, Ver., México

ultra_rgr@gmail.com, kcortes@uv.mx, juaperez@uv.mx, aarenas@uv.mx

Resumen. Una arquitectura orientada a servicios o SOA (*Service Oriented Architecture*) es una arquitectura de software que proporciona servicios que corresponden a funcionalidad de negocio reutilizable a través de interfaces bien definidas. SOA surge ante la necesidad de poder emplear funcionalidad ya disponible en sistemas heterogéneos distribuidos. Para desarrollar una SOA, existen diversas opciones tecnológicas que permiten implementar servicios web. Un servicio web permite definir componentes o servicios reutilizables que representan tareas de negocio. En este trabajo se presenta el desarrollo de una arquitectura orientada a servicios para una aplicación denominada PAAC. PAAC es el prototipo de una herramienta para registro de actividades y productos académicos que pretende servir de base para el desarrollo de una Línea de Productos de Software (LPS) y sigue un estilo Modelo Vista Controlador (MVC). Dadas las ventajas que ofrece SOA, se decide implementar este estilo arquitectónico para PAAC como una opción en la futura implementación de la mencionada LPS. Adicionalmente, es posible constatar la reutilización de la lógica de negocios de PAAC, gracias a la implementación del MVC.

Palabras Clave: Arquitectura de Software, SOA, Servicios Web, Líneas de Productos de Software, Android.

1 Introducción

Una Línea de Productos de Software (LPS) “es un conjunto de sistemas de software intensivo que comparten un conjunto de características común y administrado las que satisfacen las necesidades específicas de un segmento del mercado o misión y que se desarrollan a partir de un conjunto común de activos esenciales de una manera prescrita” [1]. Con un enfoque de Línea de Productos de Software se busca la reutilización proactiva y planeada, mejorando así la calidad y reduciendo costos y el tiempo de introducción al mercado. La arquitectura de una LPS es quizá el artefacto o activo más importante, ya que permite la

reutilización y cumplir con los requisitos de calidad. La arquitectura, además, es la base común de cada uno de los productos que conforman la LPS. Una Arquitectura de Software se define como "una descripción de más alto nivel de los subsistemas y sus componentes y las relaciones que existen entre ellos" [2]. Una arquitectura orientada a servicios (SOA) tiene como base la implementación de servicios para realizar tareas que lleven a cumplir los requisitos del negocio. En este modelo de arquitectura se tienen tres elementos principales: el consumidor de los servicios, el proveedor de los servicios y el registro o repositorio de los servicios que hace posible que el consumidor haga uso de los servicios del proveedor.

Desde hace algún tiempo, se tiene en desarrollo una herramienta para el registro de productos y actividades académicas (denominada PAAc) para los profesores de la Facultad de Estadística e Informática de la Universidad Veracruzana. Con el objetivo de construir una Línea de productos, se han construido dos prototipos funcionales de la herramienta. Estos dos prototipos, además de ayudar en la especificación de requerimientos para la LPS, han permitido probar tecnología para el desarrollo de la misma. A fin de explorar el uso de una arquitectura orientada a servicios como una opción para el desarrollo de una LPS, se decide, a partir de la versión más reciente de PAAc obtener una SOA. Así, aquí se presenta el desarrollo de la arquitectura orientada a servicios para PAAc. Este documento se encuentra organizado de la siguiente manera: en el segundo apartado se define la arquitectura SOA y sus principales características; en este mismo apartado, además, se describen los servicios web que contribuyen a la implementación de SOA; en el tercer apartado, se presenta el desarrollo de la arquitectura SOA para PAAc, mostrando las diversas vistas que describen esta arquitectura; en el cuarto apartado, se describen las pruebas realizadas a la implementación de SOA y, por último, se dan las conclusiones y los trabajos futuros.

2 Arquitectura orientada a servicios (SOA)

La arquitectura de software de un programa o sistema de cómputo es la estructura o estructuras del sistema que comprende los elementos de software, las propiedades externamente visibles de dichos elementos y relaciones entre ellos" [3]. Los elementos externamente visibles corresponden a los componentes de la arquitectura y se pueden definir como un elemento del sistema en el que se encapsulan servicios y funcionalidades que son accesibles mediante una interfaz. Las relaciones entre los elementos o componentes se encuentran definidos en la interfaz del componente. Como ya se dijo en la Introducción, una arquitectura de software es la base común de los productos de una LPS, promueve la reutilización y es el artefacto en el que se construye la calidad del software.

Una arquitectura orientada a servicios (SOA) trata resolver dos grandes temas que las organizaciones dedicadas al desarrollo de software enfrentan: heterogeneidad y cambio. Heterogeneidad en los sistemas y tecnologías con las que fueron desarrollados dichos

sistemas; cambio en las necesidades del mercados y la tecnología misma que ofrece cada vez un desarrollo más acelerado. Un aspecto característico de una SOA es que sigue un modelo arquitectónico de capas al agregar una capa de servicios de tecnología a la capa a la lógica de negocios deseada. Así, se cuenta con las implementaciones necesarias para las aplicaciones de los clientes [6].

Al desarrollar una SOA, se tiene una manera de diseñar, desarrollar y administrar sistemas de tal forma que:

- Los sistemas proporcionan funcionalidad del negocio a través de interfaces bien definidas.
- Los consumidores de servicios se construyen empleando funcionalidad provista por servicios ya disponibles.
- Hay una clara separación entre la interfaz del servicio y la implementación del mismo.
- La infraestructura permite el descubrimiento, composición e invocación de servicios.
- Los protocolos son predominantemente intercambios basados en mensajes.

La estructura de SOA se muestra en la figura 1. Puede observarse que dicha estructura se divide en dos partes. Al lado izquierdo, los elementos que se encargan de la funcionalidad; mientras que al lado derecho, lo elementos que tratan con la calidad del servicio. A continuación, se describen cada uno de ellos [6]:

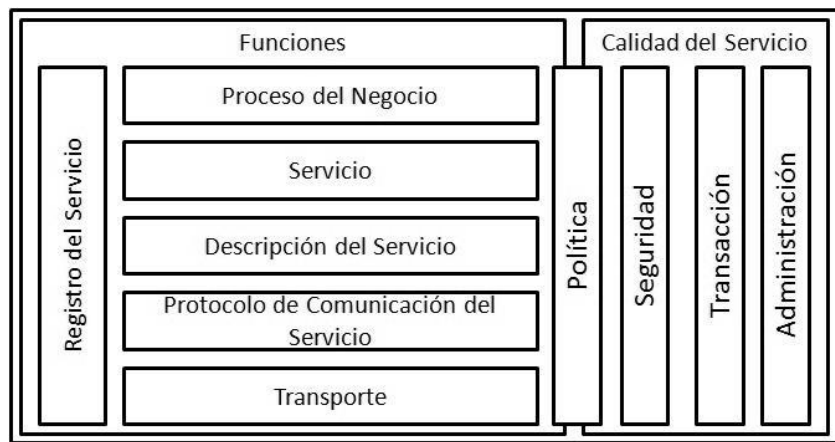


Fig. 1. Elementos de SOA.

- Funcionalidad:
 - Transporte.- Es encargado de trasladar las solicitudes de servicio del consumidor al proveedor de servicios y las respuestas del proveedor al consumidor.

- Protocolo de comunicación del servicio.- Mecanismo de comunicación acordado para que el proveedor de servicios y el consumidor intercambien lo que se solicita y lo que se devuelve.
 - Descripción de servicio.- Esquema que define el servicio, cómo debe ser llamado y los datos que se requieren para llamarlo de manera exitosa.
 - Servicio.- Se describe el servicio que está disponible para su uso.
 - Proceso del negocio.- Servicios que se llaman en un orden en particular, conforme a ciertas reglas a fin de satisfacer un requisito de negocio.
 - Registro del servicio.- Es un repositorio que contiene los servicios y las descripciones de datos a emplear por los proveedores para publicar sus servicios y por los consumidores para encontrar servicios disponibles.
- Calidad de servicio:
 - Política.- Son las condiciones o reglas que el proveedor define para los consumidores de servicios. Hay políticas que se refieren a aspectos funcionales y a aspectos de calidad de servicio.
 - Seguridad.- Conjunto de reglas que pueden ser aplicadas a la identificación, autorización y control de acceso de los consumidores de servicio.
 - Transacción.- Atributos que pueden aplicarse a un conjunto de servicios para que obtengan un resultado consistente.
 - Administración.- Es el conjunto de atributos para administrar los servicios provistos o consumidos.
 - Un Servicio Web o *Web Service* "es un conjunto de aplicaciones o de tecnologías con capacidad para inter-operar en la Web" [4]. Los servicios web, emplean una serie de protocolos y estándares para intercambiar datos entre aplicaciones que pueden estar desarrolladas en distintos lenguajes de programación y ser ejecutadas en cualquier plataforma y que se encuentran distribuidas en redes como Internet. Los Servicios Web utilizan protocolos abiertos para el transporte de la información sobre internet, el más utilizado en HTTP aunque existen otros que también pueden ser utilizados, JMS(*Java Message Service*) y SMTP (*Simple Mail Transfer Protocol*) son algunos ejemplos.

En cuanto a las tecnologías para comunicación de servicios web, se tienen RPC (*Remote Procedure Call*), SOAP (*Simple Object Access Protocol*) y REST (*REpresentational State Transfer*). REST que es la tecnología empleada en este trabajo y constituye un estilo de arquitectura para sistemas hipermedia distribuidos en la web, generalmente llamados RESTful. REST es un conjunto de principios que permiten diseñar arquitecturas en red y está basado en estándares como: HTTP, URI, de representación de recursos (XML, JSON, HTML, entre otros.) y tipos MIME. Para desarrollar aplicaciones RESTful simples, ligeras y rápidas se recomienda seguir estos principios:

- Identificación de Recursos a través de URIs

- Interfaz uniforme
- Mensajes auto-descriptivos
- Interacciones con estado a través de hipervínculos.

3 Prototipo PAAC

PAAC (Productos y Actividades Académicas) es una herramienta que permite el registro de los productos y actividades de los académicos de la Facultad de Estadística e Informática de la Universidad Veracruzana. Se pretende realizar una LPS ya que se ha encontrado variación entre las actividades y productos que un académico o investigador realiza y desea administrar en determinado momento. La primera versión se realizó en 2011[7] y consistió en el desarrollo de un prototipo a emplear para las actividades de captura, análisis y especificación de los requerimientos del dominio. Este primer prototipo consideraba siete casos de uso y empleaba las tecnologías JSF1.2 y páginas JSP bajo un modelo MVC (Modelo Vista Controlador) [2].

En el 2012 se desarrolla la segunda versión del prototipo [5]. En esta segunda versión se hicieron adecuaciones a la base de datos, mejoras a los casos de uso y se incluyeron nuevas tecnologías, bajo el mismo modelo MVC. Además de las mejoras, en este segundo prototipo se buscaba evaluar nueva tecnología para su posterior inclusión en el desarrollo de la LPS. Es así que se usaron JavaServer Faces 2.0, ADF Faces, las páginas JSF y la tecnología AJAX. En la figura 2 de la siguiente página puede apreciarse el diagrama de paquetes correspondiente al MVC de la segunda versión de PAAC, donde:

- Paquete JavaBeans: Contiene todos los JavaBeans que se utilizan el prototipo y constituyen el modelo del MVC.
- Paquete Managed Beans: Contiene todos los controladores de JSF que intervienen entre las vistas y el modelo .
- Paquete Páginas JSF: Contiene todas las páginas (vistas) del prototipo.

La arquitectura SOA que se desarrolló para PAAC está compuesta por siete recursos que a su vez están compuestos por métodos a los cuales se puede acceder a través de una URL. Estos métodos cumplen la funcionalidad de los 21 casos de uso que se implementaron en la segunda versión de PAAC. Para la implementación de esta arquitectura se empleó:

- JEE. - Plataforma para desarrollo de aplicaciones en Java. JEE fue empleada para desarrollar PAAC y se emplea la implementación de Servicios Web.
- JDeveloper 11g. - Es el ambiente de desarrollo empleado..
- HTTP. - Canal para el transporte de mensajes.
- REST.- Para intercambio de mensajes, es ligero y está bien acoplado con HTTP.

- JSON. - Formato ligero de intercambio de datos, que soporta un gran volumen de información, además de ser fácil de leer y escribir.

En la figura 3 de la siguiente página, pueden observarse los recursos de los Servicios Web del paquete Web Service que están disponibles en la red para ser consumidos por un cliente. En esta figura se pueden observar:

- Paquete JavaBeans: Todos los JavaBeans que se utiliza en la herramienta PAAC.
- Paquete Managed Beans: Todos los controladores de JSF que intervienen entre las vistas y los modelos (JavaBeans).
- Paquete Páginas JSF: Todas las vistas (páginas) de la versión Web
- Paquete Web Service: Todos los recursos de los Servicios Web que están disponibles en la red para ser consumidos por un cliente.

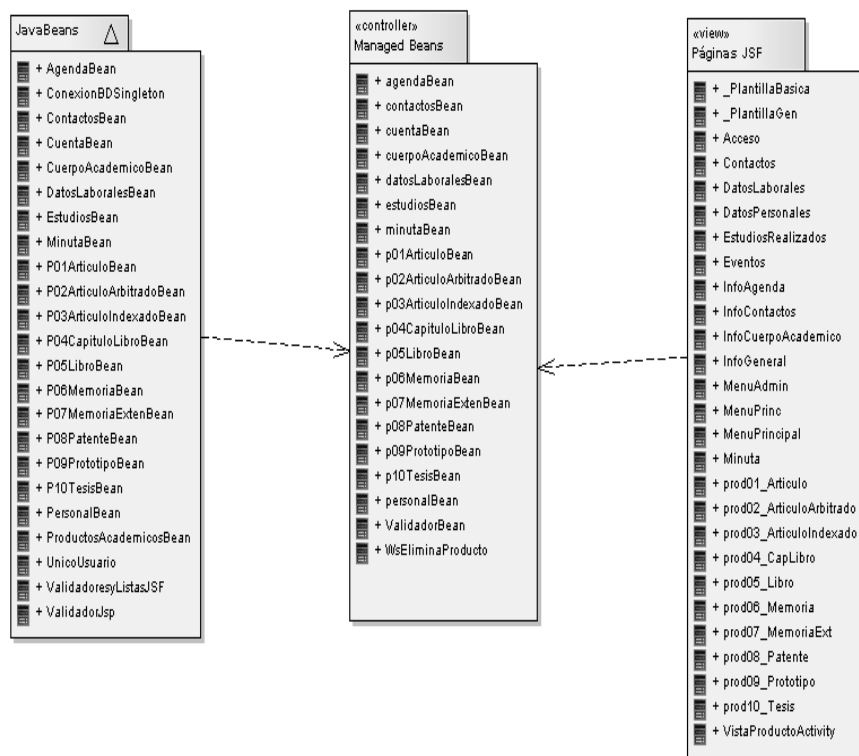


Fig. 2. MVC de PAAC v2.

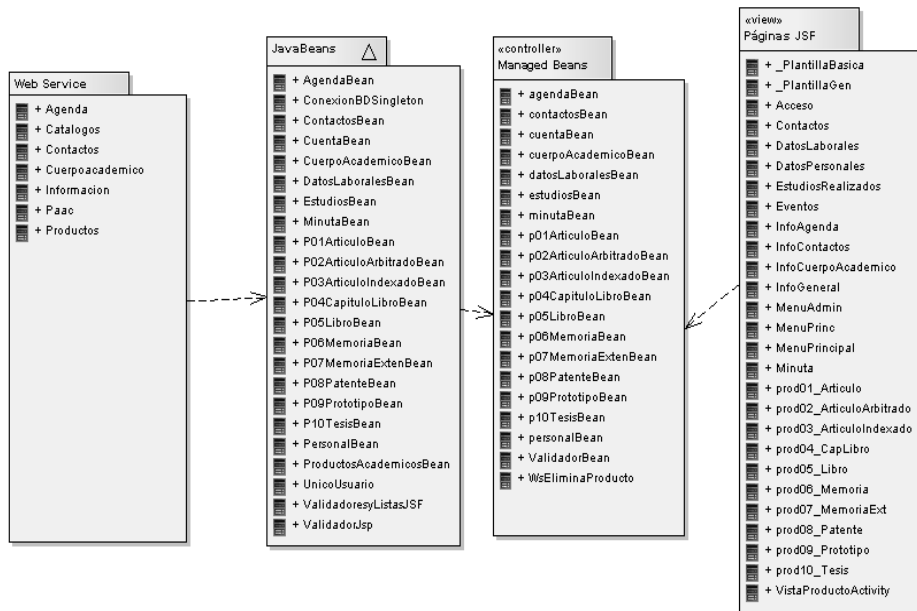


Fig. 3. SOA de PAAc v2.

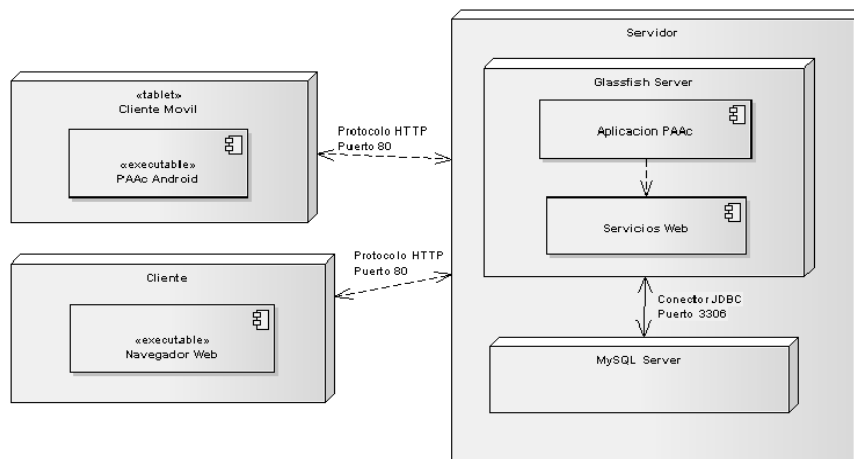


Fig. 4. Modelo de despliegue de la arquitectura.

En la figura 4, muestra el modelo de despliegue en el que se distribuyen los diversos elementos de la arquitectura en el hardware; puede observarse:

- Cliente: PC donde el usuario ejecuta el navegador web, para ingresar al sistema. Se conecta al servidor mediante el protocolo HTTP utilizando generalmente el puerto 80.
- Cliente móvil: Tableta donde el usuario ejecuta la aplicación PAAc basada en Android, para interactuar con la información del sistema. Se conecta al servidor mediante el protocolo HTTP utilizando generalmente el puerto 80.
- Servidor: Se ejecuta Glassfish Server, que es un servidor de aplicaciones en el cual se despliega la aplicación PAAc y los Servicios Web. Éstos a su vez interactúan directamente con PAAc para ser consumidos por un cliente móvil. También se conectan con el servidor MySQL, por medio del conector JDBC (Java DataBase Connector) a través del puerto 3306 (puerto utilizado por defecto en MySQL).

Dentro de la vista dinámica, se desarrollaron los diagramas de secuencia correspondientes pero no se muestran en este trabajo por limitaciones de espacio. Estos diagramas muestran la interacción entre los componentes o elementos del sistema cada vez que un cliente realiza la invocación del servicio web correspondiente.

Se desarrolla una aplicación móvil en sistema operativo Android, denominada Android PAAc para llevar a cabo las pruebas de SOA. Puede observarse la distribución de este cliente móvil en el modelo de la figura 5 arriba.

4 Pruebas

Al desarrollar software orientado a objetos guiado por casos de uso, éstos se convierten en la unidad mínima de funcionalidad, por lo tanto las pruebas realizadas se basan en los casos de uso. En las pruebas de la primera y segunda versión de PAAc se consideró detectar defectos de: configuración, de funciones faltantes, de uso incorrecto o inconsistente de archivos, de violaciones a la integridad de datos y validaciones. Se decidió entonces emplear el mismo conjunto de pruebas para la aplicación Android PAAc y la implementación de la SOA. Así, se corrobora que se cumplan con las mismas funciones ya disponibles en las versiones anteriores [5] y, de esta manera, verificar que no hubiera errores en la implementación de los servicios web. Las pruebas por casos de uso realizadas fueron las siguientes:

- Ingresar al sistema
- Actualizar datos personales
- Registrar estudio realizado
- Actualizar estudio realizado
- Registrar datos laborales
- Actualizar datos laborales
- Registrar producto

Para cada uno de los casos de uso se consideraron los cursos exitoso y fallido. Estos casos de uso y las pruebas correspondientes fueron exactamente los mismos que en las versiones anteriores, lo cual da la ventaja de: 1) aplicar pruebas de regresión sin esfuerzo adicional, 2) reducir el tiempo de pruebas con resultados confiables, 3) verificar la funcionalidad de los servicios web. No se cuenta con datos estadísticos de las pruebas pero éstas pueden resumirse en la tabla 1 para las tres versiones:

Tabla 1. Pruebas por caso de uso.

Versiones de PAAc/ Pruebas por casos de uso	1a. versión	2a. versión	Android (servicios web)
Ingresar al sistema	Sin defectos	Sin defectos	Sin defectos
Actualizar datos personales	Sin defectos	Sin defectos	Sin defectos
Registrar estudio realizado	Sin defectos	Sin defectos	Sin defectos
Actualizar estudio realizado	Sin defectos	Sin defectos	Sin defectos
Registrar datos laborales	Sin defectos	Sin defectos	Sin defectos
Actualizar datos laborales	Sin defectos	Sin defectos	Sin defectos
Registrar producto	Sin defectos	Sin defectos	Sin defectos

Después de realizar las pruebas a la Aplicación PAAc de la plataforma Android, se dieron los siguientes resultados:

- La comunicación entre la aplicación de Android PAAc y los servicios web que se encuentran en el servidor, se llevó a cabo sin ningún problema por medio del protocolo HTTP.
- La funcionalidad de los servicios web probados en cada caso de uso resulto satisfactoria, ya que se llevaron a cabo las mismas funciones que las que se tenían en las versiones anteriores. No se detectaron defectos y los resultados de las operaciones fueron correctos.
- Como era de esperarse, se comprobó que las validaciones que se hacen en la versión anterior, son muy dependientes de la tecnología con que se implementó, específicamente el framework ADF Faces (correspondiente a la vista del MVC). Por lo tanto, hubo que implementar las validaciones correspondientes en la aplicación Android PAAc.

La aplicación Android PAAc cumplió con los resultados esperados en las pruebas ya que se verificó que la implementación de los servicios web fuera correcta y completa con respecto a la lógica de negocio original. Además, se comprobó que hubo reutilización del código de la versión anterior y que los servicios web logran ofrecer una funcionalidad completa a la aplicación cliente en Android.

5 Conclusiones y trabajos futuros

En este trabajo se presentó el desarrollo de una arquitectura orientada a servicios para la herramienta PAAC. PAAC es un prototipo de una LPS para el registro de productos y actividades académicas para la Facultad de Estadística e Informática de la Universidad Veracruzana y sigue una arquitectura MVC. A continuación se presentan las conclusiones y trabajos futuros correspondientes a este trabajo.

5.1 Conclusiones

Mediante la implementación de servicios web se obtiene una arquitectura orientada a servicios para PAAC. Es así que se desarrolla la capa tecnológica que interactúa directamente con el modelo del MVC de PAAC. Para este desarrollo se emplean servicios web en java, utilizando REST y JSON. Se desarrolla, además, una aplicación en Android, denominada Android PAAC, para realizar las pruebas de los servicios web. Se realizaron las pruebas basadas en los casos de uso originales de PAAC, aplicándolas en la aplicación en Android. Se obtienen los siguientes resultados:

- La interoperabilidad con sistemas ya existentes, en este caso PAAC, logrando la reutilización de la lógica de negocios previamente implementada sin mayores problemas.
- La implementación de una Arquitectura SOA a través de servicios web.
- Se desarrolló una aplicación en la plataforma Android que, a través de la SOA, permite utilizar la funcionalidad de la herramienta PAAC.

La aportación de este trabajo está encaminada a abrir paso al desarrollo de diferentes sistemas o aplicaciones para diferentes plataformas. Aunque en esta versión se utilizó la plataforma Android como consumidor del servicio, cualquier tecnología que pueda consumir los servicios web creados en este trabajo podrá implementar toda la funcionalidad de la herramienta PAAC. Es por esto la importancia utilizar una Arquitectura Orientada a Servicios, que permite reutilizar componentes, tener interoperabilidad y ser escalable, ya que otra arquitectura no hubiese podido dar los mismos beneficios.

5.2 Trabajos futuros

Como trabajos futuros, se tienen los siguientes:

1. Incorporar funcionalidades en la aplicación de Android PAAC, correspondiente a los casos de uso faltantes a la versión anterior de esta herramienta. Los casos de uso implementados y probados para las tres versiones (incluyendo la presente versión) se mostraron en la Tabla 1. Sin embargo, en el caso de la segunda versión se tienen disponibles más casos de uso que no fueron incluidos en la presente versión.

2. Realizar pruebas de seguridad y rendimiento a través de Android PAAc para detectar defectos y mejoras a la arquitectura SOA y MVC, según corresponda.
3. Implementar, a partir de las tecnologías que se tienen, una plataforma común para poder llevar a cabo una Línea de Productos de Software. El desarrollo de esta plataforma común requerirá la integración de prácticas de SOA y LPS, lo cual, se está convirtiendo en un nuevo paradigma de desarrollo en el ámbito de las LPS.

Referencias

1. Clements P. C. & Northrop, L. M.: Software Product Lines: Practices and Patterns, Addison-Wesley. Boston, MA, USA (2001)
2. Buschmann, F., Rohnert, H., & Stal, M.: Pattern-oriented software architecture. Volume 1, a system of patterns, J. Wiley and Sons. New York (1996)
3. Bass L., Clements P., Kazman R.: Software architecture in Practice, Addison-Wesley, Boston, MA, USA (2003)
4. Guía Breve de Servicios <http://www.w3c.es/Divulgacion/GuiasBreves/ServiciosWeb>
5. Hernández Pérez, M.: Evaluación y evolución del prototipo de la herramienta PAAc (productos y actividades académicas), Tesis de Licenciatura. Universidad Veracruzana, Xalapa, Ver. México: a (2012)
6. IBM:Patterns: Service-OrientedArchitecture and Web Services, IBM Redbooks. USA (2004)
7. Ruiz Tlapa, R., & Vargas Alonso, M.: Desarrollo de un prototipo funcional de apoyo a cuerpos académicos para una línea de producto de software. Tesis de Licenciatura Universidad Veracruzana, Xalapa, Ver. Mexico (2011)