

Reingeniería de una Línea de Productos de Software: Un Caso de Estudio en el Subdominio de Ecología Marina

Natalia Huenchuman¹ *, Agustina Buccella^{1,2}, Alejandra Cechich¹, Matias Pol'la^{1,2}, Maria del Socorro Doldan³, Enrique Morsan³, and Maximiliano Arias¹

¹ GIISCO Research Group

Departamento de Ingeniería de Sistemas - Facultad de Informática
Universidad Nacional del Comahue
Neuquen, Argentina

natalia.huenchuman@gmail.com, [agustina.buccella,alejandra.cechich,matias.polla}@fi.uncoma.edu.ar](mailto:{agustina.buccella,alejandra.cechich,matias.polla}@fi.uncoma.edu.ar),
ariasmaxi89@gmail.com

² Consejo Nacional de Investigaciones Científicas y Técnicas - CONICET

³ Instituto de Biología Marina y Pesquera "Almirante Storni"

Universidad Nacional del Comahue - Ministerio de Producción de Río Negro
San Antonio Oeste, Argentina
{msdoldan,qmorsan}@gmail.com

Abstract. La ingeniería de líneas de productos de software y la ingeniería de software basada en componentes persiguen objetivos similares: minimizar los costos y el esfuerzo en el desarrollo de nuevos sistemas utilizando al reuso como la herramienta principal. Aunque poseen bases diferentes en cuanto a la manera de llevar a cabo un desarrollo de software, ambas ingenierías pueden ser combinadas para maximizar el reuso efectivo e implementar sistemas de alta calidad en un menor tiempo. En este trabajo se presenta la reestructuración de una línea de productos creada para el subdominio de Ecología Marina para orientarla estrictamente a componentes reusables (utilizando herramientas de código abierto).

Keywords: Reingeniería de Software, Sistemas de Información Geográficos, Líneas de Productos de Software, Ecología Marina, Herramientas de Código Abierto

1 Introducción

La Ingeniería de Software Basada en Componentes (ISBC) [7, 17] provee metodologías, técnicas y herramientas basadas en el uso y ensamblaje de piezas pre-fabricadas (desarrolladas en momentos diferentes, por distintas personas y posiblemente con distintos objetivos de uso) que puedan formar parte de nuevos sistemas a desarrollar. El objetivo final de esta ingeniería es minimizar los costos y tiempo de desarrollo de los sistemas, incluso mejorando la calidad de los mismos. A pesar de que la tecnología de componentes de software ha comenzado hace ya un largo tiempo atrás, aproximadamente desde la década del 60 [12], todavía hay aspectos que siguen siendo analizados para obtener beneficios tangibles al desarrollar nuevos sistemas. A su vez, han surgido nuevos paradigmas que poseen objetivos similares a los de la ISBC, pero que se enfocan en aspectos diferentes a la hora de construir sistemas. Uno de estos paradigmas, fuertemente analizado en la actualidad, es la Ingeniería de Líneas de Productos de Software (ILPS) [3, 5, 15, 18] la cual provee mecanismos para la definición de activos comunes junto con una variabilidad controlada dentro de un dominio en particular. Las principales diferencias entre ambos enfoques, líneas de productos y componentes, radican en dos aspectos principales. En primer lugar, en las líneas de productos de software se reusan los activos que fueron diseñados explícitamente para su reutilización, es decir, se crean con un objetivo de predictibilidad contra el oportunismo del desarrollo de componentes. En segundo lugar, las líneas de productos son tratadas como un todo, no como productos múltiples que se ven y se mantienen por separado como en el desarrollo de componentes⁴. Sin embargo, a pesar de que en el enfoque de líneas de productos no se obliga explícitamente al desarrollo de los activos utilizando componentes de software, el uso combinado de ambos enfoques contribuye a reducir el

* Este trabajo esta parcialmente soportado por el proyecto UNCOMA F001 "Reuso Orientado a Dominios" como parte del programa "Desarrollo de Software Basado en Reuso".

⁴ A Framework for Software Product Line Practice, Version 5.0. http://www.sei.cmu.edu/productlines/frame_report/pl_is_not.htm

acoplamiento e incrementar la cohesión, mejorando la modularidad y la evolución de los sistemas construidos [1, 2].

En trabajos anteriores [13, 14] se realizó una línea de productos de software (LPS) aplicada al subdominio de Ecología Marina que sirvió como una plataforma de servicios comunes, aplicables a todos los productos de la línea, y servicios variables, aplicados solo a algunos productos. Dicha línea fue creada a partir de la metodología de desarrollo de LPS orientada a subdominios [14] la cual combinaba ventajas de otras metodologías muy referenciadas tanto en el ámbito académico como en la industria [3, 6, 10, 15]. A pesar de que la LPS sirvió en un primer momento para crear un producto dentro del dominio (creado para el Instituto de Biología Marina y Pesquera “Almirante Storni”⁵), surgieron varios inconvenientes a la hora de realizar modificaciones o crear nuevos productos para otras organizaciones; el código se encontraba altamente acoplado y no permitía un reuso sencillo y real de la plataforma y de la instanciación de la variabilidad. Para solucionar este inconveniente se buscó entonces rediseñar la LPS aplicando técnicas que permitan un desarrollo de componentes de software para lograr así un reuso efectivo dentro del dominio geográfico. Esto generó que no sólo fuera necesario un cambio en la tecnología en que estaba implementada la LPS, sino también cambios en el enfoque de desarrollo; lo que llevó a un proceso de reestructuración de la LPS. Dicho proceso incluyó parte de reestructuración de código, debido a que los componentes fueron reescritos en otro lenguaje; e ingeniería directa, debido a que se modificó el diseño de la línea agregando nuevos requerimientos y modificando los previos.

En la literatura existen varios trabajos que proponen nuevas metodologías o presentan casos de estudio que reestructuran líneas de productos de software hacia un enfoque de componentes [5, 9, 11, 16, 19]. Por ejemplo, en [9] se presenta la aplicación de una metodología para reestructurar un sistema de soporte de inventarios de negocios. A su vez, en [16], se presenta una aplicación del método de Análisis de Opciones para la Reingeniería (OAR) centrado en la minería de componentes existentes de una línea de productos o de una nueva arquitectura de software. Por último, en [11] se realiza una comparación de las propuestas más referenciadas en la literatura enfocándose en los nuevos desafíos respecto al área de reestructuración o reingeniería de líneas de productos. En este trabajo utilizamos una metodología que combina algunas de estas propuestas con nuestra metodología enfocada en subdominios de aplicación [13, 14].

Este artículo está organizado de la siguiente manera. A continuación se describe la metodología utilizada para la reestructuración de una LPS en el dominio geográfico junto con las actividades involucradas para el análisis, diseño y desarrollo de componentes reusables. La metodología se presenta mediante la ilustración de un caso de estudio real en donde se pueden ver los pasos realizados para la reconstrucción de la LPS creada para el subdominio de ecología marina, y de los beneficios obtenidos. Finalmente en la última sección se presentan las conclusiones y trabajo futuro.

2 Reestructuración de la Línea de Productos de Software

Para llevar adelante la reestructuración de la LPS, se combinó el método de *Análisis de Opciones para la Reingeniería* (OAR), presentado en [5] junto con metodología propuesta para el dominio de ecología marina, presentada en [4, 14]. El método OAR fue seleccionado debido a las experiencias de su aplicación [5, 16] y la viabilidad de aplicarla a la LPS heredada.

En la Figura 1 se muestran las siete actividades de la metodología correspondientes a la fase de *Ingeniería de Dominio* del desarrollo de la LPS en donde se puede ver la influencia de los estándares para la realización de algunas de ellas. A continuación se describen las tareas involucradas en estas siete actividades junto con los trabajos que se han llevado a cabo para aplicarlas al caso de estudio dentro del subdominio de ecología marina:

- *Análisis de Documentación e Información*: esta actividad incluye principalmente adquirir el conocimiento del dominio analizado, los ámbitos donde tiene alcance y el estudio de la documentación brindada (artefactos de software plasmados como modelos de variabilidad, arquitectura de referencia, modelo conceptual, etc.). También, se deben considerar los beneficios que

⁵ <http://ibmpas.org/>

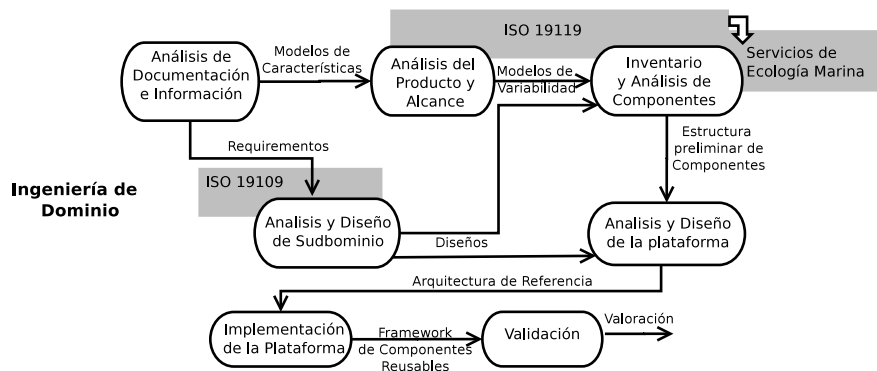


Fig. 1. Metodología para reestructurar la LPS

aporta a la organización el producto implementado y los requerimientos que no fueron relevados anteriormente.

Para realizar estas tareas, se analizó nuevamente el dominio de ecología marina mediante la información provista por referentes expertos en dichos dominios. Con esta información se pudieron analizar los requerimientos iniciales contra los nuevos resultando en la adición de dos nuevas características, la ejecución de determinadas consultas espaciales y la generación de datos estadísticos. Dichos requerimientos no fueron contemplados en el diseño anterior de la LPS, por lo que debieron ser agregados a la nueva versión. La descripción detallada del diseño original de la LPS puede verse en [13, 14].

- *Análisis del Producto y Alcance*: esta actividad se realiza junto al análisis obtenido de la etapa anterior que sirve como guía aportando la funcionalidad y el comportamiento propio del negocio, las decisiones de diseño e implementación y los nuevos requerimientos que se buscan cumplir. Aquí se deben estudiar las herramientas de software que fueron utilizadas, analizar las nuevas características, e investigar y seleccionar las nuevas herramientas de software a utilizar que cumplan con los requisitos solicitados y se adecúen a las reglas definidas en los estándares propuestos por el Consorcio Geo-Espacial (OGC)⁶ y el Comité Técnico ISO/TC 211⁷; en particular de la normas de Arquitectura de Servicios (OpenGIS Service Architecture)⁸ y de la ISO 19119⁹.

En nuestro trabajo, primero se analizó la LPS heredada debido a los problemas surgidos al momento de reusarla; su diseño altamente acoplado generaba que el reuso de la funcionalidad sea muy difícil de realizar tomando mucho tiempo y esfuerzo crear nuevos productos de la línea. En general, estos inconvenientes se generaban básicamente debido a la tecnología empleada para el desarrollo de la plataforma. La arquitectura de la LPS heredada fue desarrollada a nivel de módulos utilizando librerías de código abierto para la gestión de datos geográficos. Las mismas fueron: PostGIS¹⁰, para la creación de la base de datos espacial; Geoserver¹¹, como servidor de mapas para publicar datos a partir de las principales fuentes de datos espaciales que utilizan estándares abiertos; y OpenLayers¹², como cliente ligero web, elegido debido a la facilidad de acceso a su configuración y adaptación del código fuente. A pesar de que muchas de estas herramientas son muy utilizadas en aplicaciones actuales, no permiten desarrollos independientes que incrementen la modificabilidad y evolución. Por ejemplo, podemos afirmar que la herramienta OpenLayers lidera el ranking de uso en aplicaciones geográficas de código abierto en la actualidad. Sin embargo, a pesar de ser fácil de usar, posee muchos inconvenientes derivados de su tecnología similar a JavaScript. Al ejecutarse del lado del cliente genera que

⁶ <http://www.opengeospatial.org/>

⁷ <http://www.isotc211.org/>

⁸ The OpenGIS Abstract Specification: Service Architecture, 2002.

⁹ Geographic information. Services International Standard 19119, ISO/IEC,2005.

¹⁰ <http://postgis.refractive.net/>

¹¹ <http://geoserver.org/display/GEOS/Welcome>

¹² <http://openlayers.org/>

haya que considerar el código dependiendo las particularidades de cada navegador, y además su bajo nivel de tipado también genera varias inconsistencias. Además, el inconveniente más importante es que toda la lógica del negocio debe ser incluida en la interface de cada módulo generando un alto acoplamiento en el código que imposibilita el reuso efectivo de los servicios de la plataforma. Este análisis de las herramientas utilizadas en la LPS heredada y de las nuevas herramientas a aplicar estuvo guiado por la capacidad de las mismas de permitir la implementación de componentes que soporten el desarrollo de la arquitectura definida según los estándares geográficos. Dicha arquitectura, de al menos 3 niveles, promueve la separación de la interface, en una capa de *interface de usuario* que es responsable de la interacción con el usuario; de la lógica del negocio, en una capa de *procesamiento geográfico* responsable de coordinar e implementar la funcionalidad requerida por el usuario; y de la administración y almacenamiento de los datos geográficos, en una capa de *modelado geográfico* responsable del almacenamiento de los datos físicos y su manipulación. La principal ventaja de esta arquitectura es la separación de la funcionalidad en tres capas independientes que interactúan a través de sus interfaces bien definidas. Considerando entonces esta arquitectura, y luego de un estudio exhaustivo de las posibles herramientas de código abierto disponibles en la Web [8], se eligieron principalmente cinco de ellas encargadas de implementar los requerimientos de cada capa de la arquitectura. La Figura 2 muestra dichas herramientas dentro de la capa que deben implementar.

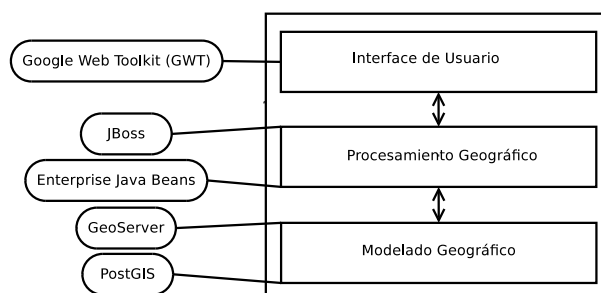


Fig. 2. Herramientas de código abierto seleccionadas para gestionar cada capa

Como vemos, para la capa de *modelado geográfico* se decidió emplear el mismo sistema gestor de base de datos espaciales *PostGIS* que en la LPS heredada [14], ya que posee una serie de ventajas que lo posicionan primero entre las opciones de este tipo de software. Entre ellas se destacan que es software libre, tiene licencia GNU General Public License (GPL), es compatible con los estándares de OGC, soporta tipos espaciales, índices espaciales, etc. Dentro de la misma capa, con respecto al servidor geográfico también se decidió mantener el mismo, *Geoserver*, principalmente, por la mantenibilidad, debido a que es muy simple manejar la configuración a través de una interface web amigable. Una vez seleccionadas las herramientas iniciales, se comenzó la búsqueda de las herramientas para el desarrollo de componentes, los cuales son implementados como parte de la capa de *procesamiento geográfico*. Entre las variadas herramientas existentes, se seleccionó la tecnología Enterprise Java Beans (EJB)¹³ debido a su amplia utilización por simplificar el proceso de creación de componentes permitiendo al programador abstraerse de los problemas generales de una aplicación (conurrencia, transacciones, persistencia, seguridad, etc.) para centrarse en el desarrollo de la lógica de negocio en sí. Luego, para el despliegue de los EJBs se eligió al servidor *JBoss*¹⁴. El mayor desafío fue luego la selección de las herramientas para crear la interface web de usuario, es decir, los clientes web de servicios geográficos (correspondientes a la capa de *interface de usuario*), ya que existen en la Web decenas de estas herramientas cada una con diferentes características. Por ejemplo, algunas de ellas usan únicamente tecnología del lado del cliente mientras que la amplia mayoría depende de funcionalidades del lado del servidor. Así permiten la ejecución de tareas avanzadas como seguridad, administración de usuarios y grupos, análisis espacial y personalización de controles

¹³ Oracle, <http://www.oracle.com/technetwork/java/javaee/ejb/index.html>

¹⁴ <http://www.jboss.org>

y funcionalidades de interfaces gráficas de usuario, entre otras. Afortunadamente, el OGC ha promovido el uso de estándares para servicios web de mapas que han ayudado a establecer un marco común de trabajo para acceder a información geográfica en internet (Web Map Service, Web Feature Service, Web Coverage Service), presentarla por medio de estilos (Style Layer Descriptor), filtrarla (Filter encoding), almacenarla, transportarla (Geography Markup Language y Keyhole Markup Language) y procesarla (Web Processing Service). A su vez, muchos de los clientes web existentes tienen dependencias entre ellos, algunos han desaparecido y otros se toman como base de nuevos desarrollos. En la Figura 3 se muestra dicha dependencia entre los clientes web para GIS extraída de OSGeo¹⁵. En este trabajo, se utilizó ese conjunto de herramientas como conjunto inicial y se las clasificó de acuerdo a tres aspectos principales: cuáles de ellas son proyectos oficialmente abandonados (marcados con un triángulo rojo), cuáles no poseen una versión reciente (identificados con triángulo amarillo), es decir que llevan más de un año sin una nueva versión, y cuáles pueden ser consideradas como vigentes hoy en día (indicadas por un tilde verde).

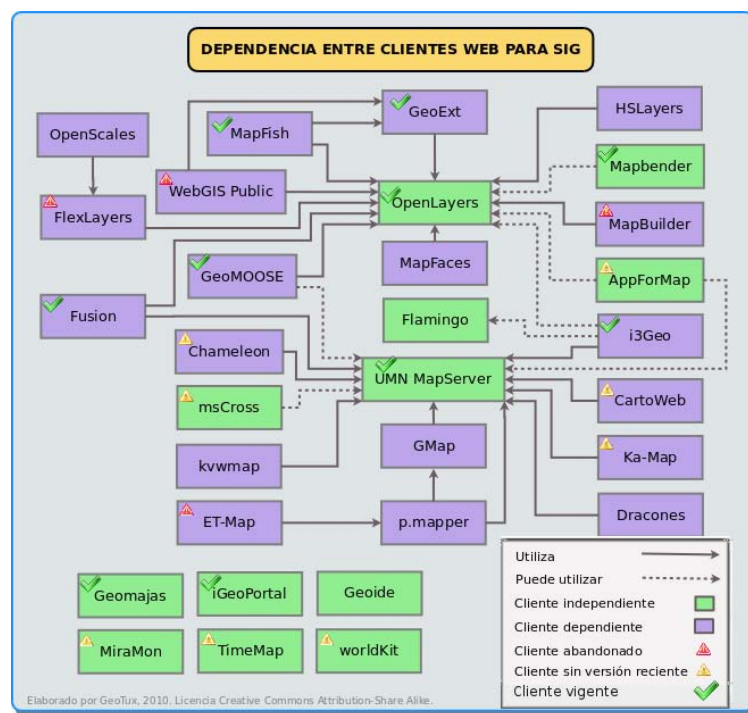


Fig. 3. Dependencia entre clientes web para GIS (Figura obtenida de OSGeo) junto con su vigencia

La mayoría de proyectos mostrados en la Figura 3 giran en torno a dos paradigmas: *UMN MapServer*¹⁶ y *OpenLayers*. Los clientes que utilizan como base *UMN MapServer*, fueron creados aprovechando las características que este cliente dispone como mapa, escala, mapa de referencia, herramientas de navegación básica, identificación de objetos espaciales y su Interface de Programación de Aplicaciones (API) llamada *MapScript*¹⁷ que ha sido implementada en diferentes lenguajes de programación. A su vez, algunos clientes utilizan opcionalmente *UMN MapServer* por medio de *MapScript* como *AppForMap*¹⁸ y *GeoMOOSE*¹⁹. Por otra parte, la nueva generación de clientes utiliza *OpenLayers* debido a su óptimo rendimiento en las tareas de mostrar los mapas a través de una interface web. Diferentes empresas contribuyen a su

¹⁵ http://wiki.osgeo.org/wiki/Comparacion_de_clientes_ligeros_web_para_SIG/

¹⁶ <http://mapserver.org/>

¹⁷ <http://mapserver.org/mapscript/index.html>

¹⁸ <http://appformap.mapuse.net/>

¹⁹ <http://www.geomoose.org/>

desarrollo y proyectos como MapBuilder²⁰ han finalizado para acelerar su progreso. Existen algunos clientes que permiten también elegir una manera adicional para renderizar sus mapas con este paradigma como i3Geo²¹ y Flamingo²². Por último, existen clientes que no se han basado en otros sino que han sido originados de manera independiente, como *Geomajas*²³, *iGeoPortal*²⁴, y *Mapbender*²⁵.

Como la mayoría de estas APIs (Application Programming Interface) para clientes ligeros están escritas en Javascript, en una primera aproximación se buscó encapsular el código, utilizando OpenLayers y GeoExt en servlets. Sin embargo, esta modalidad seguía teniendo los inconvenientes propios del lenguaje (no está totalmente orientado a objetos y no es un lenguaje tipado). Por tal motivo, se descartaron los clientes ligeros cuyo lenguaje era JavaScript, y analizaron diferentes frameworks de desarrollo. Así se llegó al framework desarrollado por Google, Google Web Toolkit²⁶ (GWT), sobre el que está desarrollado el proyecto de GeoMajas. GWT es una herramienta que facilita la creación de componentes web Javascript. Como se describió anteriormente, el desarrollo de componentes Javascript suele resultar en un proceso tedioso, ya que cada navegador tiene sus propias particularidades y el programador de las aplicaciones debe comprobar que funcionan correctamente en cada uno de los ellos. Por ello GWT proporciona una herramienta que permite al desarrollador programar sus aplicaciones web en lenguaje Java, y que posteriormente al compilarlas devuelven el código Javascript y HTML equivalente. Además, es el propio GWT el encargado de hacer que el Javascript generado funcione correctamente en los distintos navegadores. A partir de la lectura de la documentación de este framework, se encontraron numerosos proyectos vinculados a GWT, entre ellos uno relacionado a OpenLayers, denominado *GWT-OpenLayer*²⁷. Este es un wrapper Java para la API JavaScript OpenLayers que permite a los proyectos GWT usarlo como una librería (.jar). Es así que después de numerosas pruebas y análisis, se pudo obtener finalmente la combinación de herramientas que cumplía con los requisitos planteados para implementar la capa de interface de usuario. Por un lado se continúa utilizando OpenLayers, seleccionado por su facilidad, eficiencia y uso, y por otro lado, se pueden desarrollar componentes web utilizando el lenguaje Java. Otra ventaja importante en la elección de esta herramienta fue que GWT soporta invocación de métodos remotos (RPC) permitiendo una perfecta integración con la tecnología EJB.

- *Análisis y Diseño del Suddominio*: La información obtenida en la actividad anterior se utiliza para analizar y organizar las funciones o servicios que el subdominio debe ofrecer junto con las herramientas disponibles para llevarlos a cabo. El modelado del dominio permite encontrar los aspectos comunes y las variabilidades que caracterizan a las aplicaciones dentro del mismo. Como el dominio a analizar se refiere específicamente al dominio geográfico, se deben aplicar las guías definidas en la norma 19109²⁸ mediante el uso de los tipos espaciales definidos en la norma 19107²⁹.

En esta actividad se especificó el modelo conceptual final de acuerdo a los requerimientos nuevos y anteriores obtenidos en las etapas previas. Dicho modelo fue realizado mediante los lineamientos de normas explicadas previamente.

- *Inventario y Análisis de Componentes*: En esta actividad se identifican los componentes directamente de la plataforma heredada que pueden ser extraídos para incluirse como parte de la línea y conformar la arquitectura. Luego como parte del análisis de los componentes candidatos, se seleccionan aquellos tal cual se encuentran implementados y posteriormente se determinan los cambios necesarios para adaptarlos a las nuevas necesidades.

Aquí se realizó un inventario de los módulos creados en los trabajos previos [14] verificando que

²⁰ www.mapbuilder.net

²¹ <http://www.gvsig.org/web/projects/i3Geo>

²² <http://www.flamingo-mc.org>

²³ <http://www.geomajas.org>

²⁴ <http://wiki.deegree.org/deegreeWiki/iGeoPortal>

²⁵ <http://www.mapbender.org/>

²⁶ <http://code.google.com/intl/es-ES/webtoolkit/>

²⁷ <http://www.gwt-openlayers.org/>

²⁸ Geographic information. Rules for Application Schema. Draft International Standard 19109, ISO/IEC, 2005

²⁹ Geographic information. Spatial Schema. International standard 19107, ISO/IEC, 2003

su especificación coincide con la descripción de los servicios que cada uno debía proveer. En dichos trabajos, se utilizó la norma ISO 19119 para derivar los servicios que se debían ofrecer dentro del dominio de ecología marina. En la Tabla 1 se muestra una lista simplificada de los módulos con la información de los servicios provistos por cada uno, según la LPS heredada.

Módulos	Servicios que implementa
Interface Gráfica	Mostrar/ocultar capas - Herramientas de zoom - Herramientas de desplazamiento, Escala
Administrador de Características Geográficas	Mostrar y consultar datos sobre características geográficas - Ver, consultar y editar datos de características geográficas gráficamente
Detección de Cambios	Encontrar diferencias en los datos de un mismo tipo dentro de un área geográfica específica en distintos momentos
Análisis de Proximidad	Obtener todas las características geográficas dentro de un área específica
Estadísticas Gráficas	Generar estadísticas con datos de características gráficas
Acceso a Características Geográficas	Realizar consultas a un repositorio de características geográficas - Administrar los datos de las características geográficas
Acceso a Mapas	Acceder a mapas geográficos

Table 1. Inventario de módulos

A partir de este inventario de módulos y de los nuevos requerimientos solicitados se realizó un análisis para obtener los componentes candidatos. Para esto se utilizó la lista de servicios provista en los trabajos previos (derivada de la ISO 19119 para el dominio de ecología marina) adicionándole los servicios necesarios para cumplir con los nuevos requerimientos. Posteriormente se creó la nueva arquitectura de referencia para la LPS basada en las tres capas (Figura 2). En la Tabla 2 se muestran los componentes candidatos junto a la descripción de los servicios que proveen, agrupados de acuerdo a las capas mencionadas anteriormente.

Componentes	Descripción	Capa de Arquitectura
Visor Geográfico	Muestra el espacio geográfico total - Muestra/oculta capas	Interface de Usuario
Visor de Atributos	Muestra datos sobre características geográficas - Muestra detalles de las capas existentes - Muestra gráficos adicionales	Interface de Usuario
Características de Visualización	Funcionalidad de zoom - Desplazamiento - Escala - Refresco	Interface de Usuario
Herramientas de Manipulación Geográfica	Permite realizar figuras geométricas sobre el mapa, tales como puntos, líneas polígono	Interface de Usuario
Interfaz Gráfica	Permite ingresar datos - Permite visualizar los resultados de las distintas operaciones realizadas	Interface de Usuario
Análisis de Proximidad	Obtiene todas las características geográficas dentro de un área específica	Procesamiento Geográfico
Detección de Cambios	Encuentra diferencias en los datos de un mismo tipo dentro de un área geográfica específica en distintos momentos	Procesamiento Geográfico
Estadísticas Gráficas	Permite generar distintos gráficos estadísticos a partir de datos de diversas características	Procesamiento Geográfico
Consultar datos de la Base de Datos	Realiza distintos tipos de consulta acerca de la información almacenada	Procesamiento Geográfico
Acceso a Características Geográficas	Realiza consultas a un repositorio de características geográficas - Administrar los datos de las características geográficas	Modelado Geográfico
Acceso a Mapas	Accede a mapas geográficos	Modelado Geográfico

Table 2. Detalle de componentes de la nueva LPS

En la capa *Interface de Usuario* el módulo *Administrador de características geográficas* poseía tanto los servicios de visualizador de datos geográficos como de editor de características geográficas. Para mejorar la modificabilidad y evolución, se decidió crear tres nuevos componentes separando dichos servicios con las siguientes funcionalidades: el componente *Visor geográfico*, cuya funcionalidad es mostrar el espacio geográfico total y mostrar/ocultar capas; el componente *Visor de atributos* que muestra un detalle de las capas existentes, permite seleccionar y de-seleccionar capas y muestra gráficos adicionales; y el componente *Características de visualización* que posee las funcionalidades de zoom, desplazamiento, escala y refresco. Se agregó también el componente *Herramienta de manipulación gráfica* que permite realizar figu-

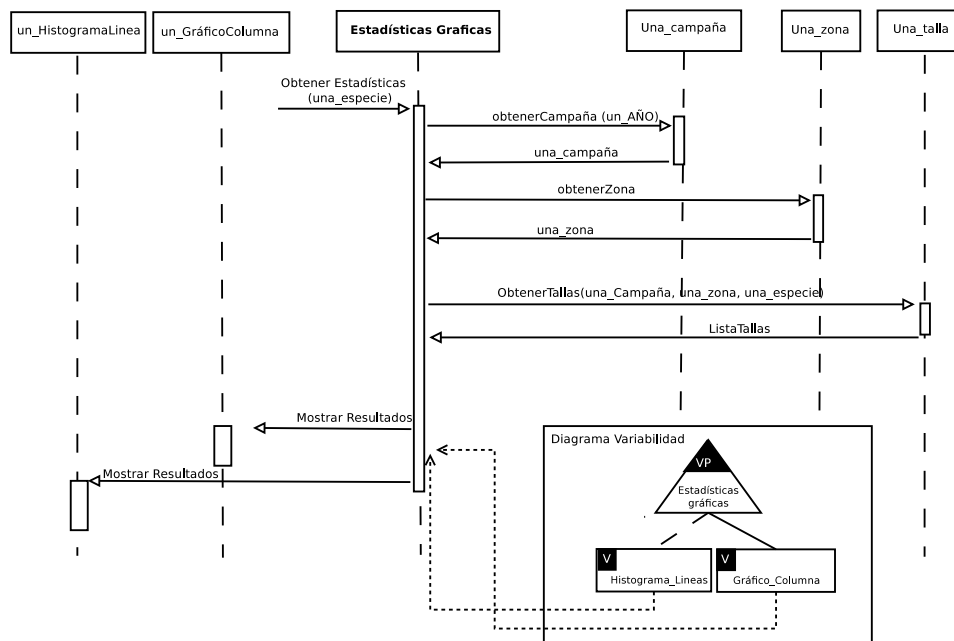


Fig. 4. Diagrama de secuencia correspondiente al componente *Estadísticas Gráficas*

ras geométricas, tales como puntos, líneas y polígonos, y luego enviar la información para su análisis.

En la capa de *Procesamiento geográfico* se realizaron varias modificaciones y se agregaron nuevos componentes de acuerdo a los requerimientos. Por ejemplo, el componente de *Análisis de proximidad* se mantuvo con la misma funcionalidad que poseía el módulo del mismo nombre, al igual que *Detección de cambios*. El componente *Estadísticas gráficas* en cambio, se redefinió completamente, permitiendo obtener y analizar estadísticas a través de los datos obtenidos del componente *Consultar datos de la base de datos*. Los componentes *Exportar mapas* y *Conteo de características geográficas* se desarrollarán en trabajos futuros. Por último en la capa de *Modelado geográfico* se crearon componentes con la misma funcionalidad que los módulos que se encontraban previamente, pero se reimplementaron con las nuevas herramientas.

- *Análisis y Diseño de la Plataforma*: En esta actividad se crea la arquitectura de referencia final basada en las características definidas en los procesos anteriores y la estructura de componentes. Se plantea la forma de extracción y desarrollo de componentes y se toman las decisiones sobre la asignación de la funcionalidad y la variabilidad de los mismos.

En nuestro trabajo se creó la arquitectura de referencia final de la LPS basada en los componentes y las capas definidas. A su vez, debido a los nuevos requerimientos surgidos del proceso de reestructuración, se diseñaron y desarrollaron dos nuevos componentes, *Cálculos Mapa* y *Estadísticas Gráficas* que implementan servicios para la ejecución de determinadas consultas espaciales y la generación de datos estadísticos. Este último componente, permite mostrar determinados datos de las especies en forma de histograma, a fin de analizar y comparar los resultados de la recolección de muestras en las diferentes zonas y campañas. La Figura 4 muestra el diagrama de secuencia que modela la interacción entre los objetos.

Como podemos observar, a partir de una campaña, un año, una zona y una especie se obtienen los datos de las tallas de los individuos (medida en mm). Con esta información se muestran los datos en el histograma. En este componente se observa un punto de variabilidad ya que los resultados se pueden mostrar a través de *histogramas en forma de líneas continuas* o bien en *gráficos de barras*.

- *Implementación de la Plataforma*: En esta actividad se implementan todos aquellos componentes de la plataforma de la LPS.

En nuestro trabajo de reestructuración se reimplementaron todos los componentes según la funcionalidad definida en la Tabla 2 y los requerimientos de la arquitectura de referencia. Por

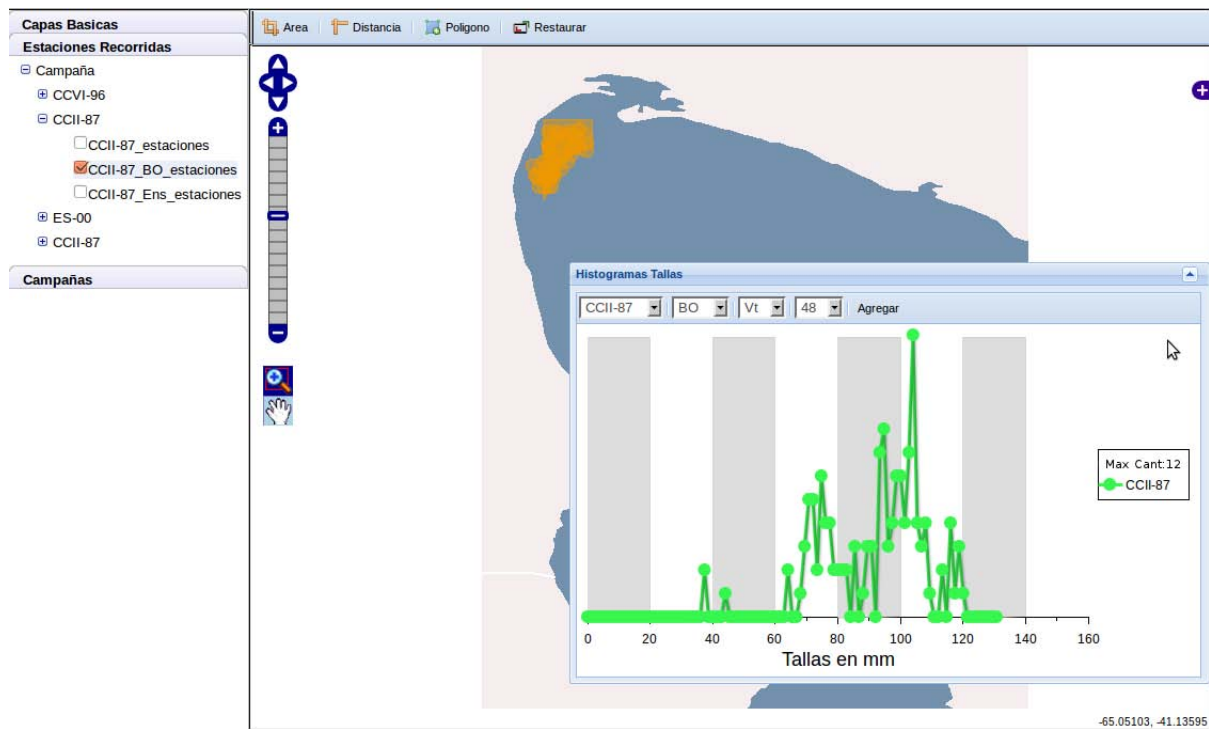


Fig. 5. Pantalla mostrando la interface del servicio *Estadísticas Gráficas*

ejemplo, la Figura 5 muestra la interface de usuario creada para el componente de *Estadísticas gráficas* (ver Figura 4) a través de histogramas en forma de líneas continuas.

El principal objetivo del servicio de estadísticas es lograr una visualización sencilla y rápida del rango de las tallas en las que se encontraron la mayor cantidad de individuos. Por ejemplo, en la figura podemos ver que para la campaña *CCII 87* de la *Viera Tehuelche* la mayor cantidad de individuos encontrados miden entre 60 y 120 mm.

- *Validación*: Se aplican casos de prueba para verificar la plataforma y la especificación de la línea de productos. En esta etapa también se deben realizar las pruebas de los nuevos productos derivados de la línea desarrollada.

Se han realizado pruebas analizando la funcionalidad sobre la nueva implementación de la LPS, en especial sobre los dos productos derivados de la misma. La LPS ya definida como componentes reusables fue utilizada para instanciar dos nuevos productos, uno para el IBMPAS y otro para el Centro Nacional Patagónico³⁰ (CENPAT-CONICET). Dichos productos están disponibles en <http://gissrv.fi.uncoma.edu.ar/SaoProjectUI> y <http://gissrv.fi.uncoma.edu.ar/CenpatProjectUI> respectivamente.

3 Conclusión y Trabajo Futuro

En este artículo se describió la reestructuración de una línea de productos de software a un enfoque orientado a componentes reusables, creados según las particularidades del dominio geográfico, y se lo ilustró de acuerdo a un caso de estudio. La metodología utilizada se compone de una serie de actividades que incluyen la utilización de estándares geográficos y de previas clasificaciones de servicios en el subdominio de ecología marina. El uso de dichos estándares permite delimitar el rango de servicios que el dominio debería ofrecer posibilitando que sea ampliado a otros subdominios geográficos diferentes al de ecología marina. Así la metodología permite reusar también los servicios definidos minimizando el tiempo y esfuerzo en las primeras etapas de la misma. Luego, el caso de estudio descripto muestra varios beneficios derivados de la utilización de dichos estándares, tanto en la especificación de los componentes como en su influencia en la selección de las herramientas

³⁰ <http://www.cenpat.edu.ar/>

de código abierto para la implementación de los mismos. En este trabajo la selección de dichas herramientas fue una de las tareas más complejas y que demandó más tiempo debido a que no sólo había que contemplar que las mismas permitan el desarrollo de componentes según la arquitectura definida en la norma ISO 19119, sino que también posean continuidad en sus desarrollos, buenas documentaciones, foros de consulta activos, flexibilidad para ser extendidos, etc. El resultado final del trabajo se traduce en un conjunto de componentes altamente cohesivos y débilmente acoplados que maximizan la modificabilidad, evolución y especialmente el reuso, ya que permiten ser tratados como unidades independientes para ser ensambladas luego en la creación de la plataforma de una LPS. Así, los tiempos y el esfuerzo dedicados en la creación de nuevos productos se pueden ver disminuídos debido a la simplicidad en el uso y ensamblaje de los componentes de la línea. Como trabajos futuros, se debe validar la nueva metodología aplicándola a otros proyectos de reingeniería de LPS. A su vez, se deben medir los logros alcanzados mediante indicadores de reuso que permitan mostrar a nivel cualitativo y cuantitativo del reuso alcanzado tanto en la creación de una LPS basada en componentes como en la creación de los nuevos productos derivados de ella.

References

1. F. Amin, A. K. Mahmood, and A. Oxley. Reusability Assessment of Open Source Components for Software Product Lines. *International Journal of New Computer Architectures and their Applications (IJNCAA)*, 3(1), 2011.
2. C. Atkinson, J. Bayer, and D. Muthig. Component-based product line development: The kobra approach. 576:289–309, 2000.
3. J. Bosch. *Design and use of software architectures: adopting and evolving a product-line approach*. ACM Press/Addison-Wesley Publishing Co., New York, NY, USA, 2000.
4. Agustina Buccella, Alejandra Cechich, Maximiliano Arias, Matias Pol'la, Maria del Socorro Doldan, and Enrique Morsan. Towards systematic software reuse of gis: Insights from a case study. *Computers & Geosciences*, 54(0):9 – 20, 2013.
5. P. C. Clements and L. Northrop. *Software Product Lines : Practices and Patterns*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 2001.
6. K. Czarnecki, S. Helsen, and U. W. Eisenecker. Formalizing cardinality-based feature models and their specialization. *Software Process: Improvement and Practice*, 10(1):7–29, 2005.
7. G. T. Heineman and W. T. Councill, editors. *Component-based software engineering: putting the pieces together*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 2001.
8. Natalia Huenchuman. Reestructuración de una línea de productos de software para el subdominio de ecología marina. Tesis de Licenciatura en Ciencias de la Computación, 2013.
9. Waraporn Jirapanthong. Experience on re-engineering applying with software product line. *CoRR*, abs/1206.4120, 2012.
10. K. Kang, S. Cohen, J. Hess, W. Nowak, and S. Peterson. Feature-Oriented Domain Analysis (FODA) Feasibility Study. Technical Report CMU/SEI-90-TR-21, Software Engineering Institute, Carnegie Mellon University Pittsburgh, PA., 1990.
11. M. Laguna and Y. Crespo. A systematic mapping study on software product line evolution: From legacy system reengineering to product line refactoring. *Science of Computer Programming*, 78(8):1010–1034, 2013.
12. D. Mcilroy. Mass-produced Software Components. In *Proceedings of Software Engineering Concepts and Techniques*, pages 138–155. NATO Science Committee, January 1969.
13. P. Pernich, A. Buccella, A. Cechich, S. Doldan, and E. Morsan. Reusing geographic e-services: A case study in the marine ecological domain. In Wojciech Cellary and Elsa Estevez, editors, *Software Services for e-World*, volume 341 of *IFIP Advances in Information and Communication Technology*, pages 193–204. Springer Boston, 2010.
14. P. Pernich, A. Buccella, A. Cechich, S. Doldan, E. Morsan, M. Arias, and M. Pol'la. Product-line instantiation guided by subdomain characterization: A case study. *Journal of Computer Science and Technology, Special Issue 12(3)*. ISSN: ., 12(3):116–122, 2012.
15. Klaus Pohl, Günter Böckle, and Frank J. van der Linden. *Software Product Line Engineering: Foundations, Principles and Techniques*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2005.
16. D. Smith, O. Liam, and J. Bergey. Using the options analysis for reengineering (oar) method for mining components for a product line. 2379:316–327, 2002.
17. Clemens A. Szyperski. *Component software - beyond object-oriented programming*. Addison-Wesley-Longman, 1998.
18. Frank van der Linden, Klaus Schmid, and Eelco Rommes. *Software Product Lines in Action: The Best Industrial Practice in Product Line Engineering*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2007.
19. G. Zhang, L. Shen, X. Peng, Z. Xing, and W. Zhao. Incremental and iterative reengineering towards software product line: An industrial case study. In *Proceedings of the 2011 27th IEEE International Conference on Software Maintenance, ICSM '11*, pages 418–427, Washington, DC, USA, 2011. IEEE Computer Society.