



Arquitectura basada en vistas para el Ecosistema de Software XEDRO-GESPRO

Architecture based on views for the Software Ecosystem XEDRO-GESPRO

Ing. Esmerida Yanet Pérez Varona ¹

Ms.C Rosel Sosa González ²

Dr.C Pedro Yobanis Piñero Pérez ³

Ing. Arisney Figueredo Ramos⁴

¹ Universidad de las Ciencias Informáticas. La Lisa. La Habana. Cuba

² Universidad de las Ciencias Informáticas. La Lisa. La Habana. Cuba

³ Universidad de las Ciencias Informáticas. La Lisa. La Habana. Cuba

⁴ Universidad de las Ciencias Informáticas. La Lisa. La Habana. Cuba

Resumen

En la investigación se realiza un estudio de las principales corrientes de la arquitectura de software. Se define una arquitectura de software de referencia para el ecosistema XEDRO-GESPRO especializado en soluciones para la gestión de proyectos. La arquitectura propuesta está basada en nueve vistas, recoge los principales elementos a tener en cuenta para el desarrollo de ecosistemas de software, incluye facilidades para el tratamiento de errores y el soporte potenciando la recuperación ante fallas.

Como parte de la aplicación de la propuesta se obtiene una familia de productos para la gestión de proyectos y se demuestra el aumento en la productividad por el alto grado de reutilización que se logra. Se evaluó la calidad de la arquitectura considerando los aspectos establecidos en la ISO 25010, se observa una valoración positiva del modelo por parte de los expertos.



Palabras clave: Arquitectura de software, ecosistemas de software, gestión de proyectos.

Abstract

In this research a study of the main currents of the software architecture is made. The Impact of the technological debts of architecture and their impact on software ecosystems, as wells as the trends in the evolution of architectures. Also, a reference software architecture for the XEDRO-GESPRO ecosystem, specialized in the solutions for the project management, is defined. The proposed architecture is based on nine views, gathers the main elements to take into account for the development of software ecosystems, includes facilities for error handling and the support, enhancing failure recovery.

As part of the application of the proposal, it is get a product family for project management and the increase in productivity is demonstrated by the high degree of reuse achieved. The quality of the architecture was evaluated, considering the established aspects in the 25010 ISO, a positive assessment of the model by the experts it is noted.

Keywords: Software architecture, software ecosystems, project management.

Introducción

En el siglo XXI la complejidad y tamaño de los sistemas de software se ha incrementado de forma vertiginosa, los sistemas han evolucionado desde una concepción monolítica y uniforme a una visión heterogénea y distribuida. La industria del software que soporta los procesos de negocio está orientando sus soluciones y servicios hacia la interoperabilidad e intercambio de información con otros sistemas, la reutilización de aplicaciones existentes y otras estrategias que garantizan mayor competitividad en el mercado. En este contexto es esencial el diseño de la arquitectura de los sistemas de información que soporten la interoperabilidad y el resto de los elementos antes señalados.

En el año 2000 la IEEE definió la arquitectura de software como: “la organización fundamental de un sistema encarnada en sus componentes, las relaciones entre ellos y el ambiente y los principios que orientan su diseño y evolución.” (IEEE Standards Department, 2000).

A partir de esta y otras definiciones que aparecen en la página oficial del SEI, se identifica a la arquitectura como el centro del problema para la integración de soluciones y la introducción de nuevos modelos de desarrollo industrial de software. Uno de estos modelos industriales de desarrollo, que gana fuerza en la actualidad, son los ecosistemas de software, que junto a las líneas de productos de software, constituyen las tendencias principales por sus capacidades para acceder a nuevos mercados y elevar los niveles de productividad (MANIKAS & HANSEN, 2013) (Martínez, 2013).

En el contexto de esta investigación es esencial comprender las relaciones entre las arquitecturas de referencia y los ecosistemas de software. Se presentan a continuación algunas definiciones de estos últimos.

En particular, en el Laboratorio de Investigaciones en Gestión de Proyectos persisten dificultades en la definición de la arquitectura que afecta a las soluciones para la gestión de proyectos entre las que se encuentran:



- Existe una familia de productos, pero se requiere desarrollar una arquitectura que garantice altos niveles de reutilización y bajos costos de producción.
- Se quiere elevar la mantenibilidad del sistema a partir de lograr un sistema de gestión de errores más amigable y que permita guardar las trazas de los errores del sistema.
- El sistema actual no cuenta con facilidades para el cálculo asíncrono de indicadores, dificultad que afecta los procesos de despliegue, actualización y disponibilidad del sistema.
- Potenciar características como la alta cohesión y el bajo acoplamiento entre los activos.
- Fortalecer la introducción de estándares de codificación que potencien el trabajo organizado de los equipos de desarrollo.
- Necesidad de introducir buenas prácticas de integración continua como modelo principal para la calidad de la familia de productos.

Materiales y métodos

Se identifica como un ecosistema de software por un sistema de trabajo entre entidades desarrolladoras de tecnologías de información y comunicaciones con el objetivo de compartir segmentos del mercado. Está soportado por la definición de una arquitectura de referencia y plataformas comunes que facilitan la integración de soluciones y componentes, el intercambio de información, recursos, artefactos y activos en general. Las relaciones entre todas las entidades del sistema deben estar soportadas por relaciones económico-financieras o convenios de trabajo bien determinados.

La arquitectura de referencia para el ecosistema propuesto cumple con los siguientes principios:

- Está basada en un modelo de mejora continua.
- Centrada en resolver necesidades del dominio del ecosistema basada en la dirección integrada de proyectos potenciando la integración de múltiples funcionalidades en un entorno seguro y de manera fiable.
- Está centrada en un modelo de negocios basado en servicios donde se combina una familia de productos para la dirección integrada de proyectos con un sistema de formación especializada en gestión de proyectos.
- El modelo de producción de las entidades principales del ecosistema está basado en líneas de productos de software.

Las vistas de la arquitectura de software propuesta dan la medida de qué se debe hacer y cómo se debe hacer (Michael Jorin, 2016), elementos que son recogidos en un total de nueve vistas que permiten visualizar el ecosistema de software XEDRO-GESPRO desde diferentes perspectivas.

Vista de procesos con impacto en la arquitectura

Se proponen grupos de procesos similares a los propuestos por el PMBOK v5 (PMBOOK, 2013) y la ISO 21500



(ISO 21500, 2012), aunque se adiciona un nuevo grupo de proceso alineado con la conceptualización y el análisis de factibilidad. En el siguiente gráfico se muestra la interrelación de los grupos de procesos y su esfuerzo en función del tiempo y los hitos principales del proyecto, ver Figura 1.

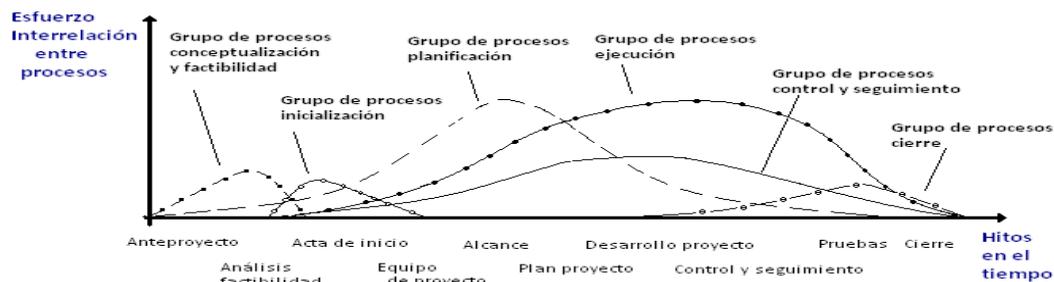


Figura 1: Gráfico de grupos de procesos y su esfuerzo en función del tiempo y los hitos principales del proyecto

Vista de presentación: Esta vista parte de los elementos analíticos provistos por la vista de procesos, sistema y datos. Formaliza la arquitectura de presentación o información de la solución de software, caracteriza la taxonomía de diseño y navegación de la solución. Define las características tecnológicas de la presentación y los patrones de usabilidad de la interfaz del sistema.

Vista de sistema: Esta vista es responsable de tener en cuenta los elementos que Rosenberg y Boehm (Rosenberg, 2017) consideran necesarios para obtener un sistema escalable, evolutivo y fiable aplicando un modelo de desarrollo ágil dirigido por la arquitectura como:

- Personal de desarrollo: Se refiere al nivel del equipo de desarrollo en las tecnologías preseleccionadas.
- Dinamismo: Se refiere al nivel de estabilidad de los requisitos a satisfacer por el producto de software.
- Cultura: Se refiere a la cultura organizacional de la entidad cliente.
- Tamaño del equipo: Se refiere a la cantidad de miembros del equipo de desarrollo.
- Criticidad: se refiere al impacto de los errores que se puedan producir por fallas de la aplicación. Específicamente se puede relacionar con la cantidad de pérdidas materiales y humanas por fallas de la aplicación.

Vista de datos: Es la vista responsable de formalizar la arquitectura de datos, identificar los elementos de persistencia, su distribución y características formales y tecnológicas. La vista caracteriza además los patrones de distribución de los datos. La arquitectura tecnológica de la propuesta provee de:

- Uso de recuperaciones de información a través de tablas resumen. Esta solución implica cierto grado de desnormalización de la base de datos a partir de incluir tablas con datos pre-calculados que se pueden actualizar offline.
- Réplica, intercambio y transformación de información entre diferentes bases de datos, iguales o diferentes en estructura y gestor de bases de datos, en línea y fuera de línea (con o sin conexión), unidireccional o bidireccionalmente.

- Una estrategia de respaldo que comprende la realización de copias de seguridad, con periodicidad diaria, garantizando una copia exacta de la información disponible en el servidor de base de datos.
- Índices necesarios para garantizar elevados niveles de eficiencia en los resultados de las consultas.

Vista de seguridad: Es la vista responsable de formalizar la arquitectura de seguridad de la propuesta. Identifica los elementos de seguridad que están presentes en las computadoras clientes, servidores, servicios, actividades de integración, la red y el ambiente de despliegue, en la interfaz e interacción con clientes. Concentra las principales decisiones de seguridad. En general la seguridad de la propuesta se establece en profundidad y por niveles, o sea, autenticación, roles y permisos; también cuenta con sistema de trazas.

Otros elementos importantes de seguridad que se tienen en cuenta en la solución son:

- Se restringe el acceso a las bases de datos desde los IP en los cuales corren los subsistemas que acceden a ellas.
- Existen estándares de implementación que permiten controlar la inyección de código.
- Existe un cortafuego donde se controla el acceso a la red interna y a los servidores internos de la aplicación.
- Se guardan las trazas en el funcionamiento de la aplicación y de la base de datos en forma de logs, con una estrategia de rotación por 10 días.
- Se realizan copias de seguridad de las bases de datos y del repositorio de activos.

Vista de integración: Es donde se formaliza la arquitectura de integración del sistema, se identifican los estándares de interoperabilidad y patrones de integración. En la propuesta se hace uso del estilo arquitectónico Modelo-Vista-Controlador que enfatiza la facilidad de modificación y la escalabilidad. Es el más generalizado en sistemas en gran escala. Este estilo permite el soporte de vistas múltiples, dado que la vista se halla separada del modelo y no hay dependencia directa del modelo con respecto a la vista, la interfaz de usuario puede mostrar múltiples vistas de los mismos datos simultáneamente; la adaptación al cambio, los requisitos de interfaz de usuario tienden a cambiar con mayor rapidez que las reglas de negocios. Se hace uso de patrones de generales de software para la asignación de responsabilidades (GRASP) como: Experto en información, Creador, Controlador, Alta cohesión, Bajo acoplamiento, Polimorfismo. Además, se establecen estándares de codificación de Ruby on Rails y base de datos.

Vista de tecnología: Es la vista que formaliza la arquitectura tecnológica, define todas las tecnologías, marcos de trabajo y entornos de desarrollo de la propuesta. Tiene un alto impacto en la productividad, así como tener en cuenta, legalidad y licencias de la solución y las tecnologías empleadas. Principales tecnologías:

- Estaciones de trabajo: Linux, Windows.
- Servidores: Ubuntu 16.04 o superior, CentOS 7 o superior.
- Gestor de bases de datos: PostgreSQL 9.5 o superior.
- Mapeo Objeto-Relacional (ORM): Active Record 4.7.2.



Ciencias informáticas: investigación, innovación y desarrollo

- Componente COTS réplica: PostgreSQL con Dblink.
- Lenguaje de programación nivel negocio: JavaScript, Ruby.
- Ambiente integrado de desarrollo (IDE): RubyMine-2017.1.2.
- Marco de trabajo para el desarrollo: Ruby on Rails.
- Marco de trabajo para interfaz usuario: JQuery UI.
- Herramienta para el modelado: Visual Paradigm for UML.
- Herramienta para el control de versiones: GitLab 9.2.2 o superior.
- Componentes COTS para subsistema GESPRO-Reporte: PostgreSQL 9.5, JavaScript y HTML5.
- Servidor web: Nginx 1.10.0.
- Componente COTS Autenticación: LDAP.
- Versionado del producto: Año.Mes.Nivel.

Vista de infraestructura: Se encarga de documentar la arquitectura de infraestructura de la solución, define los elementos arquitectónicos de infraestructura tecnológica de la propuesta. Caracteriza elementos de redes, coexistencia de las otras herramientas informáticas con la solución, ubicación y distribución de los servidores, hardware, se garantiza el respaldo de corriente.

Vista de despliegue: En esta vista se trabajan varios conceptos que influyen en el despliegue y la comercialización de las soluciones. Se discute acerca de los modelos de negocios, la composición del paquete de instalación, los requerimientos de instalación y soporte de las soluciones. Pasos a seguir en el desarrollo de la vista:

- Defina la estrategia de empaquetamiento del producto y política de versionado.
- Declare los requerimientos mínimos de instalación.
- Declare la estrategia para el soporte y mantenimiento de la solución.

Estrategia de empaquetamiento del producto y política de versionado

La escalabilidad del sistema se logra porque cada activo constituye un módulo facilitando la mantenibilidad del sistema con una alta cohesión y un bajo acoplamiento entre los mismos. Se establece el siguiente sistema de versionado “*Año.Mes.Nivel del componente*”. Se deben seguir las siguientes reglas: Los activos en el nivel cero son considerados parte del núcleo y no se recomienda su modificación descontrolada. Además, en caso de ser modificados se debe documentar de forma clara los cambios realizados. Los activos del nivel uno son considerados primarios porque constituyen elementos de configuración de software que brindan servicios a otros activos. Los activos del nivel dos son considerados secundarios porque constituyen elementos de configuración de software que brindan



servicios a otros activos, pero ellos mismos dependen de otros primarios. Los activos del nivel tres son considerados ternarios porque de ellos no dependen otros activos, generalmente representan funcionalidades terminales.

Mantenimiento del sistema

Como principal estrategia de mantenimiento se definen tres niveles de soporte y una taxonomía para la clasificación de las no conformidades de nivel 3.

Soporte de nivel 1: Es el nivel primario de soporte. Las incidencias de este nivel están asociadas usualmente a dudas operacionales de los clientes.

Soporte de nivel 2: Este nivel de soporte recibe todas las incidencias de soporte que han sido escaladas en el nivel 1. Este nivel se encarga de resolver dificultades en la operación del sistema provocadas por problemas de configuración de la plataforma.

Soporte de nivel 3: Este nivel de soporte recibe todas las incidencias de soporte que han sido escaladas en el nivel 2. Este nivel se encarga de resolver defectos en el código del sistema o la gestión de las solicitudes de cambio. Para elevar la facilidad en la detección de defectos de nivel tres se establece un sistema para el tratamiento de errores que permite capturar los errores del sistema, almacenarlos en una base de datos para su posterior consulta y análisis por parte de los especialistas. Por otra parte, las incidencias escaladas a este nivel son clasificadas según el tipo de mantenimiento que la resuelve como, por ejemplo: mantenimiento correctivo, mantenimiento adaptativo, mantenimiento preventivo, mantenimiento evolutivo y mantenimiento perfectivo.

Resultados y discusión

Se realizan un conjunto de pre-experimentos los cuales se describen a continuación.

Descripción de las variables:

G: Grupo de experimentación compuesto por las entidades donde se instala la tecnología.

X: Aplicación de la propuesta al grupo de experimentación.

O1: Observación inicial antes de aplicar la propuesta en el grupo de experimentación.

O2: Observación después de aplicar la propuesta en el grupo de experimentación.

Pre-experimento 1: Análisis de evaluación del cubrimiento de las vistas por parte de la arquitectura de referencia propuesta.

G X O

Pre-experimento 2: Evaluación por parte de expertos de la arquitectura de referencia propuesta como producto, se emplean varios expertos usando triangulación de expertos y evitando el sesgo de la opinión de sólo un experto. El diseño se representa de la siguiente manera:

G X O

Pre-experimento 3: Evaluación por parte de expertos de la arquitectura de referencia propuesta en cuanto a la calidad de uso, se emplean varios expertos usando triangulación de expertos y evitando el



sesgo de la opinión de sólo un experto. El diseño se representa de la siguiente manera:

G X O

Se emplean 10 expertos en gestión de proyectos e informática de diferentes instituciones donde se ha aplicado la propuesta. En general se encuestan expertos de los siguientes países Ecuador y Cuba. Se muestra en la Tabla 1 un resumen que caracteriza a los expertos encuestados y en la Figura 2 se muestra el histograma de frecuencias representado los años de experiencia de los expertos.

Tabla 1. Caracterización de los expertos encuestados para valoración de la propuesta.

Total de expertos	Cantidad doctores	Cantidad de masters	Promedio de años dedicados	Desviación Estándar	Mínimo cantidad de años dedicados	Máxima cantidad de años dedicados	Países representados
10	3	7	8	2,40	6	14	2

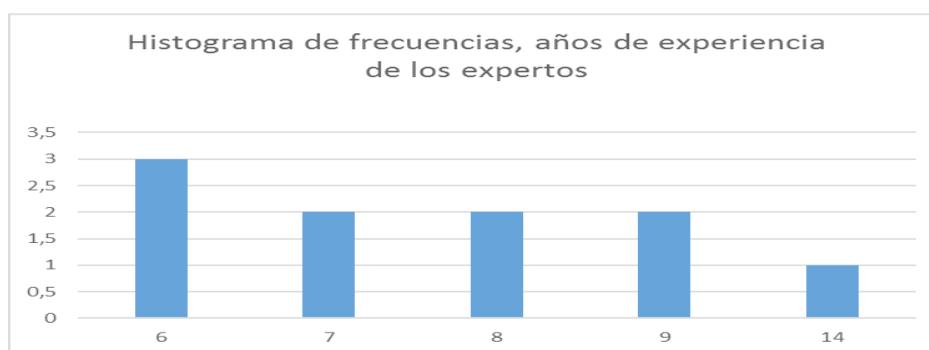


Figura 2: Histograma de frecuencias por años de experiencia de los expertos

Se encuesta a los expertos acerca de los atributos de calidad de producto, establecidos en la norma 25010 (ISO/IEC 25010, 2016). Se incluyen además un conjunto de criterios de calidad de arquitectura centrados en la capacidad de reutilización e integración de diferentes componentes.

Pre-experimento 1: comparación respecto al cubrimiento de las vistas de arquitectura.

Se evalúa la arquitectura propuesta respecto a las dimensiones, cubrimiento de las vistas de arquitectura.

Tabla 2. Cubrimiento de las vistas de la arquitectura de referencia.

Vistas	Cantidad de elementos a cumplir	Nivel de implantación		
		Vista	Real de elementos cubiertos	% Cubrimiento Vista
Proceso	4	Si	4	100,00
Presentación	6	Si	5	83,33

Sistema	7	Si	7	100,00
Datos	4	Si	4	100,00
Seguridad	7	Si	7	100,00
Integración	4	Si	3	75,00
Tecnología	15	Si	15	100,00
Infraestructura	4	Si	4	100,00
Despliegue	8	Si	8	100,00
Índice general de cubrimiento de la aplicación de la arquitectura de referencia				96,60 %

Respecto a la vista de presentación, se identifica que no se completan todos los elementos que debe cubrir la misma. Algunos de estos son:

- No está correctamente diseñado cómo se va a adaptar la aplicación a los diferentes dispositivos, como, móviles, tabletas.
- El color de todo el ecosistema está fuertemente atado al de uno de sus módulos (plugin).
- El menú depende completamente de la forma en que se diseñen por la comunidad que desarrolla el núcleo (Redmine).
- No minimiza al máximo los errores que puede introducir un usuario que haga uso incorrecto de algunas funcionalidades que permite el sistema.

Respecto a la vista de integración, se señala que no cubre elementos muy importantes como:

- No brinda servicios que permitan la integración de la mayoría de sus módulos con aplicaciones externas que sean de importancia para las organizaciones orientadas a proyectos.
- No permite realizar limpieza y actualización de los usuarios del ecosistema que los mantenga alineados con el sistema de autenticación LDAP.
- No está preparado para el proceso de integración continua, causado en gran medida por un bajo uso casi nulo de las pruebas automatizadas que ofrece el marco de trabajo que se utiliza (Ruby on Rails).

Se considera que depende de forma excesiva del sistema gestor de base de datos, lo que dificulta su integración mediante servicios y un posible cambio de gestor de base de datos

Pre-experimento 2: modelo de la calidad del producto

El modelo de la calidad del producto categoriza las propiedades de la calidad del producto de software/ sistema en ocho características: adecuación funcional, eficiencia de desempeño, compatibilidad, usabilidad, fiabilidad, seguridad, mantenibilidad y portabilidad.

Para la evaluación se pidió a los expertos que evaluaran cada criterio empleando el siguiente conjunto de términos lingüísticos $LBTL = \{Muy\ bajo, Bajo, Medio, Alto, Muy\ alto\}$. Para unificar la evaluación de los expertos respecto a cada criterio se empleó la técnica de computación con palabras modelo 2-tuplas. Se obtuvieron los resultados mostrados en la Tabla 3.

Tabla 3. Resultados de la evaluación de expertos de la propuesta respecto a los criterios definidos.

Criterios	Resultado de aplicar 2-tuplas	Varianza en la respuesta de los expertos
C1	(Muy alto; -0.40)	0,63
C2	(Alto; 0.13)	0,83
C3	(Muy alto; -0.50)	0,70
C4	(Muy alto; -0.47)	0,57
C5	(Alto, -0.20)	0,89
C6	(Muy alto; -0.44)	0,58
C7	(Alto; -0.04)	0,97
C8	(Medio; 0.46)	0,63

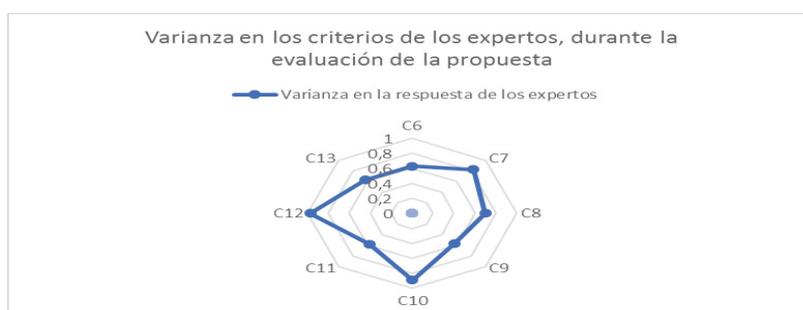


Figura 3. Gráfico que muestra la varianza de en la concordancia de los expertos respecto a cada criterio.

Pre-experimento 3: modelo de la calidad en el uso

La calidad en el uso es el grado en que un producto o sistema puede ser utilizado por usuarios específicos para satisfacer sus necesidades de alcanzar objetivos específicos con eficacia, eficiencia, ausencia de riesgo y la satisfacción

Para la evaluación se pidió a los expertos que evaluaran cada criterio empleando el siguiente conjunto de términos lingüísticos $LBTL = \{Muy\ bajo, Bajo, Medio, Alto, Muy\ alto\}$. Para unificar la evaluación de

los expertos respecto a cada criterio se empleó la técnica de computación con palabras modelo 2-tuplas. Se obtuvieron los resultados mostrados en la Tabla 4.

Tabla 4. Resultados de la evaluación de expertos de la propuesta respecto a los criterios definidos.

Criterios	Resultado de aplicar 2-tuplas	Varianza en la respuesta de los expertos
C1	(Alto; -0.40)	0,54
C2	(Alto; 0.20)	0,44
C3	(Alto; 0.05)	0,68
C4	(Medio; 0.40)	0,73
C5	(Alto; 0.10)	0,63



Figura 4. Gráfico que muestra la varianza en la concordancia de los expertos respecto a cada criterio.

Como se muestra en la Figura 4. Gráfico que muestra la varianza en la concordancia de los expertos respecto a cada criterio. no hay una variación significativa respecto a los criterios de los expertos. No obstante, el criterio con mayor variación respecto a la opinión de los expertos fue en cuanto a la ausencia de riesgos. Se observa mayor concordancia de expertos con respecto a la eficiencia y la cobertura de las funcionalidades del contexto.

Además, se observa, la valoración positiva de la propuesta por parte de los expertos, representada por el valor “Alto” en la calificación conjunta de la mayoría de los criterios. Se identifican que el criterio con una evaluación más baja es el C4. El C4 referido a la ausencia de riesgos en la propuesta en tiempo real, elemento que se tiene en cuenta en los procesos dentro de la dirección integrada de proyectos.

Conclusiones

Los ecosistemas de software constituyen una forma efectiva de construir grandes sistemas de software siempre que se logre establecer en ellos una arquitectura robusta y escalable en el tiempo. Se logra la validación de la arquitectura de referencia propuesta a partir del montaje de la misma en una familia de productos para la gestión de proyectos con un nivel de implantación del 96,6 % y se demuestra el aumento en la productividad por el alto grado de reutilización que se logra. La arquitectura propuesta fue evaluada por un conjunto de expertos considerando los criterios de calidad como producto establecidos en la ISO 25010, se observa, la valoración “Muy alto” de la propuesta por parte de los expertos. Los criterios con la evaluación más baja fueron los asociados a la portabilidad de la propuesta.



Referencias

- IEEE Standards Department. (2000). Architecture working group of the software engineering committee.
- Institute, P. M. (2017). A guide to the project management body of knowledge (PMBOK guide) Sixth Edition / Project Management Institute. Project Management Institute (PMI), Inc. Newtown Square, Pennsylvania 19073-3299 USA.
- ISO. ISO 21500:2012 (2012) Guidance on Project Management. International Organization for Standardization. Disponible en: http://www.iso.org/iso/catalogue_detail?csnumber=50003.
- ISO/IEC 25010. (2016). Ingeniería de Software y Sistemas. Requisitos de la Calidad y Evaluación de Software (SQuaRE).
- Manikas, K., & Hansen, K. M. (2013). Software ecosystems—A systematic literature review.
- Martínez, G. O. (2013). Análisis de grandes ecosistemas de software abierto.
- Michael Jorin, P. P. (2016). Guía base de la arquitectura de software.
- Pacelli, L., 2004. The Project Management Advisor: 18 major project screw-ups, and how to cut them off at the pass. Pearson Education.
- Peña Abreu, M. (2017). Modelo para el análisis de factibilidad de proyectos de software en entornos de incertidumbre. Universidad de las Ciencias Informáticas, La Habana, Cuba.
- PMBOOK. (2013). A guide to the project management body of knowledge. Pennsylvania.
- Rosenberg, D. (2017). Rapid, evolutionary, reliable, scalable system and software development: the resilient agile process., (págs. p. 60-69).
- Stellingwerf, R., & Zandhuis, A. (2013). ISO 21500 Guidance On Project Management: A Pocket Guide (Best Practice). Van Haren.

