

UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS

Facultad 5

Centro de Consultoría y Desarrollo de Arquitecturas Empresariales

Laboratorio de Investigaciones en Gestión de Proyectos



**Modelo para el desarrollo de un ecosistema de software orientado a soluciones para la
gestión de proyectos**

Trabajo final presentado en opción al título de
Máster en Gestión de Proyectos Informáticos

Autor: Ing. Jessie Castell González

Tutor: Dr. C Pedro Yobanis Piñero Pérez

La Habana, Noviembre 2012

AGRADECIMIENTOS

A mis padres por estar siempre, todos mis logros de ustedes son.

A mi hermano por la fuerza y el orgullo que me inspira.

A mis abuelos, tía y primo, por el apoyo y los ánimos.

Agradezco a quien soporta todo cada día, a Angel por el amor.

A mi suegra por sus comentarios, para la tesis y para la vida.

A los amigos, casi hermanos que acompañan mi vida todo el tiempo y en cualquier lugar: especialmente a Lianepolda, a Greisy, Mayre, Yusel, Omar, Maribel, Orestes, el PKO y todos los que no alcance tiempo y espacio para nombrar.

A Pedro, mi tutor y guía, el mayor impulsor de este trabajo y quien merece toda mi gratitud.

A mis compañeros de trabajo que ayudaron en esta tarea constante, gracias por el apoyo.

A todos los que contribuyeron de cualquier manera, gracias infinitas.

DECLARACIÓN JURADA DE AUTORÍA

Declaro por este medio que yo Jessie Castell González, con carnet de identidad 85120910495 soy la autora principal del trabajo final de maestría “Modelo para el desarrollo de un ecosistema de software orientado a soluciones para la gestión de proyectos” desarrollada como parte de la Maestría en Gestión de Proyectos Informáticos y que autorizo a la Universidad de las Ciencias Informáticas a hacer uso de la misma en su beneficio, así como los derechos patrimoniales con carácter exclusivo.

Y para que así conste, firmo la presente declaración jurada de autoría en La Habana a los __ días del mes de _____ del año _____.

Jessie Castell González

RESUMEN

El incremento de las potencialidades de los sistemas informáticos actuales exige un desarrollo de productos más ágil, seguro y personalizado, que posibilite satisfacer las cambiantes necesidades del negocio y del mercado. La productividad que se logre alcanzar durante el desarrollo de software se relaciona directamente con la madurez y efectividad de estrategias de trabajo orientadas a la reutilización e integración de componentes ya implementados que eviten la duplicación de esfuerzos.

Algunas de las variantes más empleadas para disminuir los esfuerzos y el tiempo de producción durante el desarrollo de software se basan en la reutilización de funcionalidades ya desarrolladas. Tal reutilización puede lograrse a través de la integración de soluciones de apoyo, la concepción de arquitecturas empresariales de alto nivel, y el desarrollo de ecosistemas de software. Esta última variante potencia la noción de productos mejor adaptados y que obtienen de su entorno el mayor provecho posible.

Los sistemas de gestión de proyectos precisan del intercambio de información con diversos sistemas que existan en una organización, para alcanzar mayor efectividad en la toma de decisiones en tiempo real.

La presente investigación aborda la propuesta de un modelo para el desarrollo de un ecosistema de software adaptado a las necesidades y características de los sistemas de gestión de proyectos. El objetivo es definir un modelo para el desarrollo de un ecosistema que posibilite incrementar los niveles de productividad de estos sistemas, a partir de la integración de diversos componentes y la implicación de participantes dentro del mismo.

Palabras clave: *ecosistema, gestión de proyectos, integración, productividad, modelo.*

Abstract

Increasing the potential of current systems requires a more agile product development, customizing and ensuring, to make possible rapid change attending the business needs and market. Productivity is achieved during that is related with the maturity and effectiveness of own work strategies aimed at reuse and integration of components already implemented to decrease the efforts to be made.

Some of the variants used to decrease production time and effort is by integrating already built systems that provide support functions, the development of software ecosystems that enhance the design of products and better adapted to their environment to obtain the greatest possible benefit. In other hand, the systems that manage information for appropriate management projects require the exchange of information with various systems that exist in the organization, as well as increased functionality for system interoperability.

This research deals with the proposal of a model for the development of an ecosystem of software tailored to the needs and characteristics of the project management systems. The objective is to define a model for the development of an ecosystem that enables increased productivity levels of this kind of systems, from the integration of various components and participants involved.

Key words: *ecosystem, integration, project management, productivity, model.*

ÍNDICE GENERAL

INTRODUCCIÓN	2
CAPÍTULO 1: ANÁLISIS DE TENDENCIAS PARA EL DESARROLLO DE ECOSISTEMAS DE SOFTWARE E INTEGRACIÓN DE SISTEMAS	10
INTRODUCCIÓN	10
ANÁLISIS BIBLIOMÉTRICO.....	10
1.1. ANÁLISIS DEL DESARROLLO DE ECOSISTEMAS DE SOFTWARE.	10
1.2. ANÁLISIS DE LAS PRINCIPALES TENDENCIAS PARA LA INTEGRACIÓN DE SISTEMAS.	18
1.3. ANÁLISIS DE LOS PROCESOS EN UNA ORGANIZACIÓN.	24
1.4. ANÁLISIS DE LOS MECANISMOS DE INTEGRACIÓN DE SISTEMAS PARA LA CONSTRUCCIÓN DE ECOSISTEMAS DE SOFTWARE ORIENTADOS A LA GESTIÓN DE PROYECTOS.	25
CONCLUSIONES PARCIALES DEL CAPÍTULO.....	28
CAPÍTULO 2: MODELO DE DESARROLLO DEL ECOSISTEMA DE SOFTWARE PARA LA GESTIÓN DE PROYECTOS.	30
INTRODUCCIÓN	30
2.1. DESCRIPCIÓN DEL MODELO.	30
2.2. PRINCIPIOS GENERALES DEL MODELO.	30
2.3. BASES DEL MODELO.....	31
2.3. DEFINICIÓN DE LOS PROCESOS DEL MODELO.	31
CONCLUSIONES PARCIALES DEL CAPÍTULO.....	49
CAPÍTULO 3: VALIDACIÓN DEL MODELO PROPUESTO.	50
INTRODUCCIÓN	50
3.1. DESCRIPCIÓN DEL ESTUDIO DE CASO: DESARROLLO DE UN ECOSISTEMA DE SOFTWARE PARA EL SISTEMA GESPRO EN EL ENTORNO UCI.....	50
3.2. ANÁLISIS DE LA VARIABLE PRODUCTIVIDAD EN EL DESARROLLO DEL SISTEMA GESPRO.	56
3.3. ANÁLISIS DE LAS CARACTERÍSTICAS DEL ECOSISTEMA.	60
3.4. ANÁLISIS DE LA COMPETITIVIDAD EN EL MERCADO DEL ECOSISTEMA.....	64
CONCLUSIONES PARCIALES DEL CAPÍTULO.....	67
CONCLUSIONES GENERALES.....	68
RECOMENDACIONES	68
BIBLIOGRAFÍA	69
ANEXOS.....	74

ÍNDICE DE FIGURAS Y TABLAS

FIGURA 1. NIVELES DE UN ECOSISTEMA DE SOFTWARE. FUENTE: [JANSEN, ET AL. 2009B].	11
FIGURA 2. MODELO CONCEPTUAL DE UN ECOSISTEMA DE SOFTWARE. FUENTE: [KJETIL HANSSSEN 2010].	13
FIGURA 3. RELACIONES DE INTERCAMBIO DE PRODUCTOS Y SERVICIOS. FUENTE: [POPP 2010].	14
FIGURA 4. MODELO CONCEPTUAL PARA EL DESARROLLO DE ECOSISTEMAS DE SOFTWARE. FUENTE: [RECENA SOTO, ET AL. 2012].	18
FIGURA 5. ESTILO DE INTEGRACIÓN DE TRANSFERENCIA DE ARCHIVOS. FUENTE: [HOHPE, ET AL. 2003].	20
FIGURA 6. ESTILO DE INTEGRACIÓN DE BASE DATOS COMPARTIDA. FUENTE: [HOHPE, ET AL. 2003].	21
FIGURA 7. ESTILO DE INTEGRACIÓN DE MENSAJERÍA. FUENTE: [HOHPE, ET AL. 2003].	22
FIGURA 8. COMPONENTES DEL MODELO. (FUENTE: ELABORACIÓN PROPIA)	30
FIGURA 9. PROCESOS DEL MODELO (FUENTE: ELABORACIÓN PROPIA).	32
FIGURA 10. TIPOS DE ACTIVOS DENTRO DEL ECOSISTEMA DE SOFTWARE (FUENTE: ELABORACIÓN PROPIA).	34
FIGURA 11. VISTA DEL ECOSISTEMA DE SOFTWARE PARA LA GESTIÓN DE PROYECTOS (FUENTE: ELABORACIÓN PROPIA).	41
TABLA 1. REQUISITOS DE INFORMACIÓN SOLICITADOS POR GESPRO.	51
FIGURA 12. VISTA DE LAS CAPAS DEL ECOSISTEMA DE SOFTWARE GESPRO (FUENTE: ELABORACIÓN PROPIA).	54
FIGURA 13. MODELO DE DESPLIEGUE - VISTA DE DESPLIEGUE E INFRAESTRUCTURA (FUENTE: ELABORACIÓN PROPIA).	55
TABLA 2. CANTIDAD DE FUNCIONALIDADES DE INTEGRACIÓN POR ÁREA DE CONOCIMIENTO.	56
FIGURA 14. FUNCIONALIDADES DE INTEGRACIÓN RELACIONADAS CON LAS ÁREAS DE CONOCIMIENTO.	57
TABLA 3. ESFUERZO DEL EQUIPO DE GESPRO ANTERIOR Y POSTERIOR AL DESARROLLO DEL ECOSISTEMA.	58
FIGURA 15. COMPORTAMIENTO DE LA EFICIENCIA DE LOS RRHH EN GESPRO.	58
FIGURA 16. ESFUERZO HORAS/HOMBRES PARA EL DESARROLLO DE LOS ACTIVOS DEL ECOSISTEMA, ANALIZADOS INDEPENDIENTEMENTE.	59
FIGURA 17. COSTO DE PRODUCCIÓN TOTAL DE DESARROLLO DE LOS ACTIVOS DEL ECOSISTEMA, ANALIZADOS INDEPENDIENTEMENTE.	59
FIGURA 18. AHORRO DE ESFUERZO EN EL DESARROLLO DE LOS SISTEMAS INDEPENDIENTEMENTE AL ECOSISTEMA.	60
FIGURA 19. CRECIMIENTO DE LA CANTIDAD DE FUNCIONALIDADES POR MÓDULOS DE GESPRO.	61
FIGURA 20. CRECIMIENTO DEL ECOSISTEMA DE SOFTWARE ATENDIENDO A LOS ACTIVOS.	62
TABLA 4. CANTIDAD DE EGRESADOS POR AÑO DE LAS ASIGNATURAS QUE EMPLEAN EL ECOSISTEMA GESPRO.	63
FIGURA 21. VALORES ACUMULATIVOS DE EGRESADOS DE LAS ASIGNATURAS QUE EMPLEAN EL SISTEMA GESPRO.	63
TABLA 5. COMPARACIÓN DEL ECOSISTEMA GESPRO CON SISTEMAS DE GESTIÓN DE PROYECTOS: CUBRIMIENTO DE LAS ÁREAS DE CONOCIMIENTO.	64
TABLA 6. COMPARACIÓN DEL ECOSISTEMA GESPRO CON SISTEMAS DE GESTIÓN DE PROYECTOS: FUNCIONALIDADES Y CARACTERÍSTICAS.	65
FIGURA 22. COMPARACIÓN CON ECOSISTEMAS: CANTIDAD DE ACTIVOS.	65
FIGURA 23. COMPARACIÓN CON ECOSISTEMAS: COSTO TOTAL ESTIMADO DE PRODUCCIÓN POR ECOSISTEMA.	66
FIGURA 24. COMPARACIÓN CON ECOSISTEMAS: COSTO ESTIMADO DE PRODUCCIÓN POR ACTIVO.	66

INTRODUCCIÓN

Antecedentes y situación problemática

Los problemas más elementales de la gestión de proyectos se encuentran dirigidos a las variables o áreas de conocimiento más críticas: alcance, tiempo y costos, lo cual se conoce como el diamante de la gestión de proyectos, incluyéndose además la calidad del software [Piñero Pérez 2010]. Sobre estas áreas giran las principales necesidades de información de la gestión de proyectos, dado su nivel de impacto y de criticidad dentro de esta para el éxito y factibilidad final de los proyectos [Pacelli 2004].

Existen múltiples factores que pueden afectar la gestión de proyectos, entre ellos la falta de información real y actualizada que se tenga del estado de los mismos, y de los principales elementos o áreas de conocimiento que la componen, de manera tal que imposibilite que se re-direccione el desarrollo y re-estimen los recursos adecuadamente [Pacelli 2004; Suárez 2007].

Para corregir los principales problemas de la gestión de proyectos se aconseja como método general realizar una buena gestión de cada una de las áreas de conocimiento que la abordan, en correspondencia con lo que las más importantes escuelas proponen [IPMA 2006; PMI 2008]. Una buena gestión se apoya también en tener la información precisa y actualizada en el momento adecuado para, entre otras cosas, poder tomar decisiones sobre los proyectos en tiempo real y actualizar las áreas de conocimiento que hayan sido afectadas.

Es importante destacar el papel que juegan los proyectos para el logro de los objetivos estratégicos de las organizaciones, por lo que gestionarlos efectivamente no debe quedar solo en una intención.

La gestión de proyectos es un área interdisciplinaria donde convergen la gestión de los costos, la gestión de los recursos humanos, la gestión de las adquisiciones, entre otras. Todos estos componentes son gestionados de una u otra forma a través de sistemas específicos. La gestión de los proyectos requiere de la gestión unificada de cada una de estas áreas internas y transversales de la organización, a partir de la comunicación con estos sistemas. De esta manera se logra que todos los sistemas de información de la organización tributen hacia los objetivos de negocio de la organización.

La alineación de los procesos de negocio con las Tecnologías de la Información (TI) constituye en la actualidad un problema a solventar en aras de satisfacer los objetivos de negocio de la organización. La sinergia que exista entre procesos y TI incrementará las capacidades del negocio, así como un mayor análisis estratégico sustentado en la integración de ambos [Armstrong et al. 2010].

Con el actual crecimiento y versatilidad de las necesidades de negocio, la industria del software que soporta los procesos de negocio debe orientar sus soluciones y servicios hacia la interoperabilidad e

intercambio de información con otros sistemas, en conjunto con otras estrategias que garanticen mejor competitividad en el mercado.

Actualmente se tiende hacia la creación de alianzas para el desarrollo de sistemas, de manera tal que se compartan mercados, se personalicen los productos, aumente la usabilidad y sostenibilidad de los sistemas. En este sentido las soluciones informáticas, tanto de propósito general como específico, deben propiciar una fácil adaptación en diversos entornos. La búsqueda de soluciones a la medida, a la par del avance tecnológico, los constantes cambios del mercado y los clientes, constituyen un reto para el desarrollo de software.

Múltiples acercamientos han surgido a través de los años, grandes iniciativas desde las líneas de productos de software, estilos de arquitecturas empresariales de alto nivel como son las arquitecturas orientadas a servicios, variadas soluciones integrales, hasta la concepción de ecosistemas de software. La selección de una u otra estrategia dependerá de las necesidades particulares de cada sistema y las condiciones del entorno en que se implanten.

Un ecosistema de software se concibe como la unión coherente y organizada de soluciones y componentes de software que soportan cierta necesidad de negocio, que en su conjunto satisfagan una necesidad de mercado [Bosch 2009].

La noción de la necesidad de un ecosistema de software surge a partir del interés que muestren varios sistemas, componentes y demás participantes en integrarse y coexistir como un nuevo producto integrado, obteniendo un aumento en los valores de la productividad, personalización y productividad en el mercado [Lungu 2009].

Resulta útil la concepción y construcción de ecosistemas de software, de diversa escala y propósito, que faciliten la integración de los componentes y sistemas de un entorno específico, que cubran las necesidades de información de la organización de manera sostenible, y garanticen la seguridad de los datos.

La simple mención de ecosistemas de software reconocidos como el de Microsoft, Apple, Eclipse y SAP [Angeren et al. 2011; Jansen et al. 2009b] muestra la acertada decisión que representaría emprender el desarrollo de un ecosistema de software de productos o familias de productos, que aumente su competitividad en el mercado. El costo de producción de dichos ecosistemas puede variar en dependencia del modo de producción que se defina para el mismo, así como de la cantidad de activos que integren.

El desarrollo de ecosistemas consiste fundamentalmente en la integración y el establecimiento de las relaciones y contratos posibles entre las partes que lo conformen. Por ende la construcción de ecosistemas de software en particular conllevaría a una integración de componentes y sistemas de software para alcanzar un propósito mayor que independientemente.

El interés de integrar los sistemas de las organizaciones es cada vez mayor, aunque es un tema que debe ser no solo bien intencionado sino eficazmente gestionado. Para alcanzar determinados niveles de interoperabilidad, la integración de sistemas debe implicar y coordinar elementos de un determinado ámbito o dominio, tales como: procedimientos, datos, infraestructuras y aplicaciones, entre otros [Chen et al. 2008; Kasunic et al. 2004].

Actualmente para la gestión de proyectos existen sistemas como Redmine, DotProject, Trac y Microsoft Project, que a pesar de tener ciertas funcionalidades implementadas aún estas se consideran insuficientes para lograr la integración. Tampoco los sistemas para la gestión de la información empresarial, como es el caso de los ERP (del inglés *Enterprise Resource Planning*), han logrado establecer claramente una integración con sistemas externos a ellos. Aún cuando han dedicado esfuerzos a construir módulos propios para gestionar proyectos no llegan a abarcar todas las funcionalidades de este tipo de sistemas. Dicha situación convierte el tema de la integración como una problemática actual [Sehgal 2007].

Esta realidad no es ajena al sistema de gestión de proyectos GESPRO, sistema para la gestión integrada de proyectos, que desde su surgimiento en el 2008 cuenta con un grupo permanente de desarrolladores [Piñero Pérez et al. 2011a].

GESPRO como sistema que unifica la información de las áreas de procesos de la gestión de proyectos dentro de una organización, requiere de la información primaria que gestionen sistemas como los ERP [Vera 2006] y otros sistemas externos. Los mencionados sistemas brindan información relevante que puede usarse para la gestión de sus principales áreas. Sin embargo todavía GESPRO no cuenta con los mecanismos para una adecuada integración con otros componentes externos a él.

La Universidad de las Ciencias Informáticas (UCI) ha asumido la Suite GESPRO como sistema base para la gestión de proyectos en su red de centros de desarrollo de soluciones y servicios informáticos, sin embargo el sistema no ofrece una solución integral dentro de los entornos en los que se ha desplegado, dentro y fuera de la universidad. Aún es incipiente su interconexión con otros sistemas tanto de gestión empresarial de la organización, como con otros componentes de software y hardware.

Adicionalmente, como parte de la comercialización de sus productos al mercado nacional e internacional necesita aumentar la competitividad ofreciendo soluciones integrales para las empresas. Igualmente GESPRO debe permitir la gestión adecuada de los proyectos que maneja internamente, que potencie la producción de productos y servicios más competitivos.

En los entornos donde se encuentra desplegado y en explotación el sistema GESPRO se han detectado las siguientes deficiencias [Piñero Pérez 2012]:

- Falta de definiciones y estándares para garantizar la integración con sistemas ERP.

- Limitada cantidad de recursos humanos y materiales para satisfacer el creciente número de solicitudes de la introducción de herramientas de gestión de proyectos para la resolución de problemas de la práctica.
- Duplicidad en la información que manejan los sistemas en un mismo entorno causada por la falta de integración entre las soluciones para la gestión empresarial.
- Inconsistencia en las informaciones al mantener los datos duplicados y persistir un modelo manual de actualización de los datos contenidos en las aplicaciones.

Problema científico de la investigación

La insuficiente definición de mecanismos para el desarrollo de ecosistemas de software que se adapten a los sistemas para la gestión de proyectos afecta la productividad en el desarrollo de estos sistemas.

Objeto de investigación

Ecosistemas de software.

Objetivo general

Elaborar un modelo para el desarrollo de ecosistemas de software orientados al desarrollo de soluciones para la gestión de proyectos.

Objetivos específicos

- Elaborar el marco teórico referencial acerca de modelos para el desarrollo de ecosistemas de software, así como de los principales enfoques y mecanismos para la integración de sistemas.
- Definir los elementos del modelo para el desarrollo de ecosistemas de software orientados a soluciones para la gestión de proyectos.
- Aplicar y validar el modelo propuesto a partir del análisis de un estudio de caso del sistema GESPRO.

Campo de acción

Ecosistemas de software para la gestión de proyectos.

Tipo de investigación

Descriptiva: se utiliza para describir y especificar los elementos característicos de los ecosistemas, así como las tendencias actuales para el desarrollo de los mismos.

Hipótesis

Si se logra elaborar y aplicar un modelo para el desarrollo de un ecosistema de software orientado al desarrollo de soluciones para la gestión de proyectos, entonces se mejorará el nivel de productividad en el desarrollo de estos sistemas.

Operacionalización de la variable independiente y dependiente

Variable independiente: Modelo para el desarrollo del ecosistema de software orientado a la gestión de proyectos.

Dimensión	Indicadores	Unidad de medida
Características del ecosistema	Robustez	Crecimiento del producto y el ecosistema
	Creación de nichos de mercado	Valor agregado del ecosistema Variedad de soluciones y servicios ofertados
Competitividad en el mercado	Comparación con otros sistemas de gestión de proyectos	Cubrimiento de las Áreas de conocimiento de la gestión de proyectos Funcionalidades que cubre Tipo de licencia (Abierta/Cerrada)
	Comparación con otros ecosistemas	Cantidad de activos Costo de producción

Variable dependiente: Productividad en el desarrollo del sistema de gestión de proyectos.

Dimensión	Indicadores	Unidad de medida
Nivel de interoperabilidad	Funcionalidades de integración desarrolladas	Cantidad de funcionalidades de integración (U)
	Áreas de conocimiento de gestión de proyectos cubiertas	Cantidad de áreas cubiertas por las funcionalidades de integración (U)
Eficiencia en el desarrollo	Esfuerzo de producción	Cantidad de horas-hombres por concepto de ahorro del esfuerzo de producción de los activos integrados
	Eficiencia económica del ecosistema	Costo de producción del ecosistema por concepto de ahorro del costo de producción de los activos integrados
	Eficiencia de los Recursos Humanos	Cantidad de funcionalidades de integración desarrolladas /Cantidad de recursos humanos utilizados

Muestreo

Población: Escenarios de aplicación del sistema de gestión de proyectos GESPRO.

Muestra: Se realizara una investigación que se valida a partir de su aplicación en el caso de estudio de GESPRO versión 13.05 actualmente desplegado en la red de 23 centros de desarrollo de la UCI, lo cual representa un 89 % de la población total de 26 escenarios donde está actualmente desplegado.

Métodos de investigación

Métodos teóricos: se emplearon el Histórico-Lógico, Hipotético- Deductivo, Sistémico.

Durante el análisis bibliográfico se revisaron las principales fuentes de información y autores que abordan la problemática de la investigación, examinándolas desde un enfoque valorativo. Además la investigación sigue el método hipotético-deductivo al diseñar a partir del problema de la investigación los objetivos e hipótesis. Por último se empleó el método sistémico debido a que el problema de la investigación se concibió como un todo, a partir de los datos obtenidos y principales deficiencias detectadas, la revisión realizada acerca de los ecosistemas de software y las tendencias para su desarrollo, se funden en un único sistema sostenible e integral.

Métodos empíricos: se emplearon el método Experimental y la Medición.

La investigación se basó en los datos reales anteriores y posteriores a la aplicación del modelo en el sistema de gestión de proyectos GESPRO, en diferentes versiones del mismo.

Diseño de la investigación

El diseño de la investigación fue pre-experimental siguiendo la estrategia de pre-prueba/post-prueba con un solo grupo [Sampieri et al. 2006].

Se analizó el comportamiento de la variable dependiente de la investigación a partir de la aplicación del modelo de desarrollo de un ecosistema de software, se establecieron comparaciones de acuerdo al escenario anterior y posterior a la aplicación. El grupo de control lo constituyó la versión 13.05 de GESPRO anterior el desarrollo del ecosistema, el grupo experimental lo representó la misma muestra tras desarrollar el ecosistema.

En la investigación se empleó la estrategia de Estudio de Casos al no contarse con los datos suficientes para pruebas estadísticas de rigor.

Métodos y herramientas de medición

Cuestionarios y encuestas: se aplicaron a un grupo de expertos con la finalidad de comparar los escenarios de integración tipos a seleccionar como parte del desarrollo del ecosistema, atendiendo a distintos indicadores a medir.

Entrevistas: aplicadas a jefes de proyecto, arquitectos y desarrolladores, tanto para el levantamiento de información relevante para la formulación de la propuesta como para la posterior captura para la validación de la misma.

Revisión documental: se revisaron documentos y artefactos del desarrollo del sistema de gestión de proyectos GESPRO.

A través de métricas se realiza la comparación de las variables dependientes en la aplicación del Estudio de Caso.

Aporte práctico de la investigación

El aporte de la investigación se resume en la conceptualización y aplicación de un modelo para el desarrollo de un ecosistema de software orientado a soluciones para la gestión de proyectos, constituido por componentes: principios, conceptos bases, procesos, actividades, artefactos, entradas, salidas y técnicas y herramientas.

Listado de publicaciones, eventos y avales de la investigación.

- Aymé Perdomo Alonso, Jessie Castell González, L.E. Hernández Vega, C. Pupo Ortiz, A. Suárez Hernández, L. López Nicot, M. Romero González, Y. Guelmes León, R. Llerena Ferrer, Y. Pérez Olmos y L.M. Teijón Acosta, “Desarrollo de un Proyecto Piloto basado en SOA y BPM para las Operaciones Industriales Biotecnológicas”, en V Congreso Latinoamericano de Ingeniería Biomédica, 2011: Palacio de las Convenciones, La Habana, Cuba. p. 4.
- Maypher Román Durán, Jessie Castell González, Yisel Jinoria Fernández, Dianelys Espinosa Zaldívar, “El Diagnóstico Informacional para las organizaciones: propuesta de procedimiento para su ejecución”, en Congreso Internacional de Información INFO 2010, La Habana, Cuba. 2010. p. 10
- Jessie Castell González, Yisel Jinoria Fernández, Dianelys Espinosa Zaldívar, Maypher Román Durán, “Metodología para el desarrollo de ontologías”, en UCIENCIA 2010. 2010: Universidad de las Ciencias Informáticas, La Habana, Cuba. p. 14.
- Maypher Román Durán, Jessie Castell González, Dianelys Espinosa Zaldívar, Yisel Jinoria Fernández. “Experiencias del uso efectivo de las ontologías en el análisis semántico de la información organizacional”. en VII Congreso Nacional de Reconocimiento de Patrones. 2009. Santiago de Cuba, Cuba.
- Jessie Castell González, Yisel Jinoria Fernández, Dianelys Espinosa Zaldívar, Maypher Román Durán, “Metodología para el desarrollo de ontologías como herramienta de representación de la información empresarial”, en Segundo Taller cubano EUREKA 2010 "Hacia una tecnología semántica y trans-disciplinaria para la inteligencia organizacional y el desarrollo sostenible". 2009: Hotel Nacional, La Habana, Cuba.
- Jessie Castell González, Aymé Perdomo Alonso, Yoisy Pérez Olmos. “Experiencias de una iniciativa orientada a procesos para el Centro de Inmunología Molecular”. en 10ma Semana Tecnológica FORDES 2010. 2010. La Habana, Cuba.

- Yusel Sablón Fernández, Jessie Castell González And Muñoz., M.S. “Modelo Basado en Casos para la planificación de proyectos de software”. en VI Conferencia Científica UCIENCIA 2012. La Habana, Cuba.: UCI, 2012, p. 15.
- Jessie Castell González y Dianelys Espinosa Zaldívar. “La integración semántica dentro del proceso de desarrollo de iniciativas SOA/BPM” en II Taller Internacional “Las TIC en la Gestión de las Organizaciones” XIV Convención Informática Habana 2011, 2011, p.8.

Estructura de la tesis

La tesis se estructura en 3 capítulos. En el primero se realiza un análisis exhaustivo de las principales tendencias y conceptos para el desarrollo de ecosistemas de software y para la integración de sistemas. Conjuntamente se analizan los sistemas de información en una organización, en particular los sistemas más conocidos y empleados para gestión de proyectos y la gestión empresarial, refiriéndose específicamente a las capacidades que brindan para la integración.

En el segundo capítulo se define un modelo para la construcción de ecosistemas de software para el sistema GESPRO, estableciendo para ello principios del modelo y una serie de pasos con sus entradas, salidas, herramientas y técnicas.

En el capítulo 3 se hace un análisis sobre los resultados del pre-experimento realizado, a través de la comparación de las dos muestras relacionadas. Se evalúan los resultados de los indicadores establecidos para las variables dependientes. Por último se formulan las conclusiones generales y las recomendaciones de la investigación.

CAPÍTULO 1: ANÁLISIS DE TENDENCIAS PARA EL DESARROLLO DE ECOSISTEMAS DE SOFTWARE E INTEGRACIÓN DE SISTEMAS

Introducción

En el capítulo se analizan los principales conceptos asociados a la investigación. Se estudia el término de ecosistemas de software, así como los principales aspectos y las tendencias a tener en cuenta para su desarrollo. Se describen las principales tendencias para la integración de sistemas de software como principal base para el desarrollo del ecosistema. Se realiza además una revisión de los principales sistemas de información relacionados con la gestión de los procesos empresariales en las organizaciones, en particular con el sistema para la gestión de proyectos GESPRO, también se realiza un análisis como base para la propuesta. Se realiza el análisis bibliométrico de la investigación.

Análisis bibliométrico

	Últimos 5 años		Años Anteriores	
Libros y monografías	6	35%	11	65%
Tesis de doctorado	3	60%	2	40%
Tesis de maestría	4	100%	0	0%
Tesis de grado	1	100%	0	0%
Artículos en Revistas referenciadas en Web of Science, SCOPUS	8	100%	0	20%
Memorias de eventos	17	85%	3	15%
Artículos publicados en la web	4	57%	3	43%
Páginas web	5	100%	0	0%
Reportes técnicos y conferencias	6	86%	1	14%
Entrevistas personales	1	100%	0	0%
Totales	55	73%	20	27%

1.1. Análisis del desarrollo de ecosistemas de software.

El término ecosistema en sí representa la unión simbiótica de varios elementos, es un concepto heredado de la biología. En el caso particular de la presente investigación se abordará el desarrollo de ecosistemas de software.

Según la bibliografía revisada, se ha podido constatar que los ecosistemas son interpretados, desarrollados e implantados desde diversos puntos de vistas o enfoques. Como objetivo principal para la concepción de sus componentes está ganar posicionamiento en el mercado, integrar las soluciones en un entorno, producir valor agregado a cierto producto aumentando sus funcionalidades. A continuación se analizarán y valorarán los principales enfoques para su desarrollo.

Entre los ecosistemas de software más reconocidos se encuentran el de Microsoft, MySQL/PHP, Iphone [Jansen, et al. 2009b], SAP, Eclipse [Angeren, et al. 2011], entre otros.

Enfoque de mercado

El enfoque de mercado profundiza en las relaciones que se establecen hacia participantes externos al ecosistema, relaciones contractuales, etc. El autor [Frantz 2008] entiende por un *Ecosistema de software* (SECO, por las siglas en inglés *Software ECOSystem*) como el conjunto de entidades de negocios que actúan como una unidad e interactúan para compartir un mercado común de aplicaciones informáticas y servicios bajo un sistema ordenado de interrelaciones entre ellas.

El ecosistema de software puede ser visto desde varias perspectivas, ver Figura 1, o niveles donde intervienen elementos con similares características u objetivos. Estos niveles se describen a continuación [Boucharas et al. 2009; Jansen, et al. 2009b]:

- 1) Nivel de vendedores de software: considerado como el nivel donde se desarrollan los productos y servicios. Es en este nivel donde se toman las decisiones acerca de cómo integrar los servicios y productos dentro de la organización, el diseño de una arquitectura extensible y portable, así como la posible planeación de líneas de productos de software. Es igualmente importante la gestión del conocimiento dentro de este nivel, pues determinará la eficiencia del modo de producción a alcanzar en el tiempo.
- 2) Nivel de proveedores de una red de software (Software Supply Network - SSN): representa el conjunto de hardware, software y servicios de la organización concentrados en satisfacer una demanda del negocio. El reto en este nivel consiste en decidir cuáles proveedores y compradores de software elegir.
- 3) Nivel de ecosistema de software: los principales retos de este nivel lo representan la caracterización y la modelación del ecosistema, en dependencia de las características particulares de la organización, los roles que intervendrán dentro de esta y la magnitud del mismo. Otra de las problemáticas es determinar cuál estrategia de mercado emplear para obtener más beneficios.

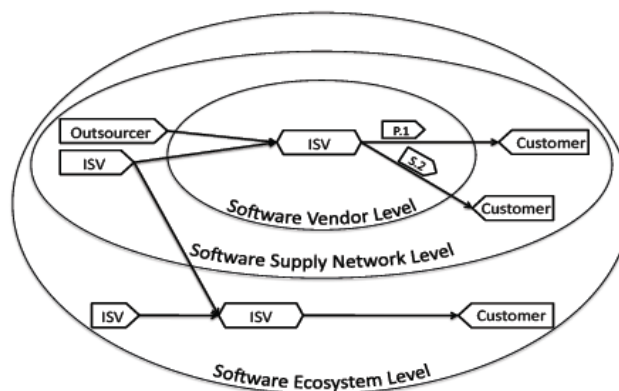


Figura 1. Niveles de un ecosistema de software. Fuente: [Jansen, et al. 2009b].

A valoración de la autora se considera más oportuna la construcción de un ecosistema a partir del nivel de vendedores y desarrolladores de software, aunque además se tomen aspectos de los restantes niveles. Este debe ser el nivel núcleo y eje del desarrollo.

Los referidos niveles o perspectivas adicionalmente están ligados a algunos conceptos claves que definen un ecosistema de software [Jansen et al. 2009a]:

- Mercado: Básicamente el público objetivo del ecosistema, y uno de los elementos más importantes a definir en su desarrollo.
- Tecnologías y plataformas: Ciertamente uno de los elementos más fuertes dentro de un ecosistema de software será decidir cuáles tecnologías, plataformas y marcos de trabajo seleccionar para su desarrollo.

Existen otros tipos o especies de ecosistemas relacionados al ecosistema de software, como son: el ecosistema de negocio, el ecosistema digital y el ecosistema de tecnología [Adomavicius et al. 2005]. También se han creado ecosistemas específicos a ciertos dominios, todos con el propósito de concebir y dar mayor valor a un producto.

Desarrollar ecosistemas de software añade valor a aquellos sistemas que por sí solos no ofrecen suficiente competencia en el mercado, ya sea porque no ofrecen fácil integración con otros sistemas, o bien porque un proyecto no puede asumir el desarrollo de todos los componentes y funcionalidades que lo harían competitivo. Los ecosistemas de software tienden hacia la reutilización y personalización de las soluciones, según el interés específico para lo cual se construya.

Las restricciones de un ecosistema están estrechamente relacionadas con las características del ecosistema de software en particular que se pretenda construir. Otras restricciones a tener en cuenta podrían ser los proveedores de licencias, las agencias de comercialización, las restricciones geográficas, políticas y económicas.

Sin embargo, dichas limitantes ayudan a comprobar y chequear sistemáticamente el estado del ecosistema y cómo este se perciba externamente, en cuanto a si todos sus componentes y participantes juegan el rol establecido, si los clientes están conformes, si el ecosistema por sí mismo es sustentable, productivo y robusto [Jansen, et al. 2009a].

En cuanto a las medidas del estado de un ecosistema se tienen en cuenta elementos extraídos de los grandes ecosistemas biológicos y ecológicos: la robustez, la productividad y la creación de nichos de mercado [Huang 2010; Iansiti et al. 2002].

La robustez se refiere a la capacidad de resistencia y persistencia en cierto entorno y de la estructura del ecosistema, en el caso de los ecosistemas de software pudiera medirse como el

posicionamiento y la estabilidad en el mercado, el prestigio del producto, la cantidad de elementos que combina y permiten identificar que el ecosistema funciona como un todo.

La productividad por su parte mide la efectividad del ecosistema, la ganancia que obtiene del entorno y la variación en el tiempo de la efectividad en el desarrollo de los sistemas. Además se refiere al proceso de innovación y generación de nuevo conocimiento a partir del desarrollo del ecosistema.

Relacionado con este último concepto se hace imprescindible la adaptación e innovación mediante la creación de nuevos nichos de mercado, refiriéndose a la capacidad para incrementar el valor de los productos, la variedad de soluciones y servicios, incorporación de nuevas tecnologías, incorporación a nuevos escenarios, etc. Se resume en el crecimiento y desarrollo permanente de nuevas y diversas capacidades en los ecosistemas [Iansiti, et al. 2002].

El autor Geir Kjetil [Kjetil Hanssen 2010] aborda la importancia de explotar precisamente las capacidades de innovación del ecosistema, con la mayor apertura y diversidad posibles. En la Figura 2 se han identificado los componentes principales del ecosistema, y la manera en que estos interactúan.

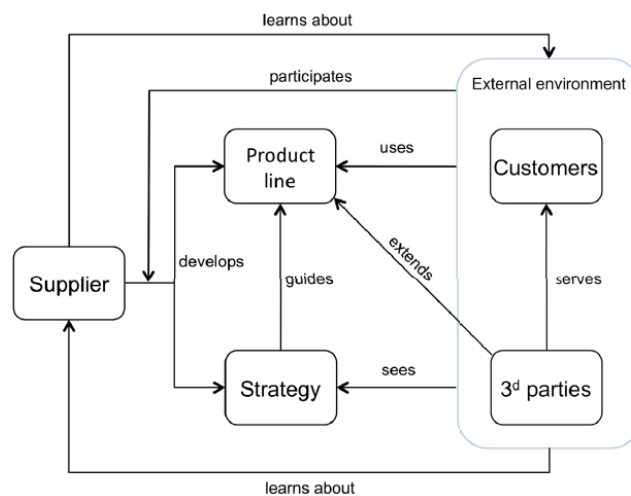


Figura 2. Modelo conceptual de un ecosistema de software. Fuente: [Kjetil Hanssen 2010]

Se considera que la implicación y relaciones que se establezcan con la organización son un factor determinante en el éxito y la salud de los ecosistemas. Se han definido características internas de los ecosistemas, relacionadas con su composición, las restricciones que se establecen y la estabilidad, entre otras.

Dentro de los ecosistemas pueden coexistir los siguientes roles o participantes [Wong 2011]: clientes finales del ecosistema o productos dentro de este, consultantes de la industria y el negocio afín, proveedores de infraestructura de software, integradores de sistemas, etc. Los participantes representan quizás uno de los elementos dentro de los ecosistemas más importantes, es necesario

determinar la función y el costo que conlleva la inclusión de cada participante, los beneficios y proveen al ecosistema. La cuestión está entonces en cómo obtener el máximo provecho de cada uno.

Los proveedores lo representan tanto los desarrolladores de los productos que formen parte del ecosistema como las entidades certificadoras y proveedoras de licencias. Si los productos y herramientas son suministrados por comunidades de desarrollo de código abierto, se considera que la relación será mucho más simple y sustentable. En la Figura 3 se representan las principales relaciones que se establecen entre los roles del ecosistema.

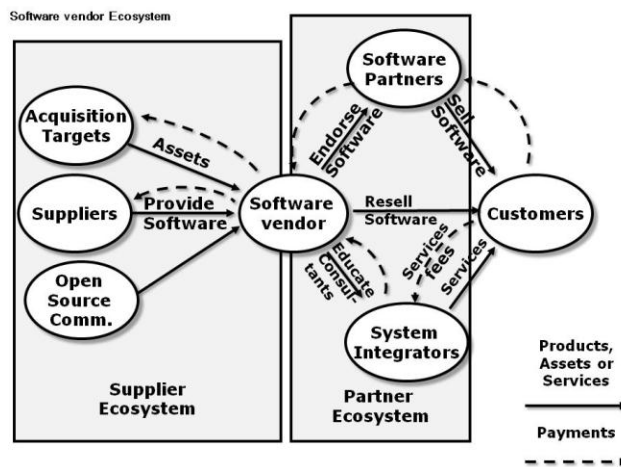


Figura 3. Relaciones de intercambio de productos y servicios. Fuente: [Popp 2010].

Los autores que orientan sus investigaciones hacia este enfoque analizan principalmente la dimensión comercial de los ecosistemas, a partir de las relaciones económicas entre ellos, pero no los identifican como una alternativa para el desarrollo rápido de soluciones.

Enfoques basados en la integración de soluciones

Al constituir el ecosistema de software la conexión de varios sistemas, recursos, herramientas y servicios en un entorno y con un objetivo específico. Se considera que la base para el desarrollo de los mismos está en el establecimiento de sólidas arquitecturas empresariales que se apoyen en la integración de sus componentes. Evidentemente un ecosistema es algo más que la mera integración de sus partes, además deberá proveer de un entorno que facilite la productividad de cada uno de los sistemas, como del ecosistema en general [Recena Soto 2012].

Bosh por su parte define un ecosistema de software como el conjunto de soluciones de software que habilitan, automatizan y soportan cierta necesidad de negocio [Bosch 2009].

Existen ciertos desafíos durante la construcción de ecosistemas: deben crearse las arquitecturas lo suficientemente estables, seguras, confiables y flexibles para soportar el ecosistema; la heterogeneidad de los sistemas debe tratarse correctamente así como admitir que los sistemas

puedan ser abiertos o propietarios y de igual forma se integren y compartan información relevante para el ecosistema [Barbosa et al. 2011].

Otro de los principales retos según Lungu [Lungu et al. 2010] está relacionado con la concepción de sistemas y proyectos como una isla, solución no factible, por lo cual se hace necesario establecer conexiones de integración entre ellos.

Enfoques híbridos

El autor Piñero [Piñero Pérez et al. 2011c] define un ecosistema de software como el sistema de relaciones entre las entidades soportado por la definición de arquitecturas y plataformas comunes que facilitan la integración de soluciones y componentes, el intercambio de información, recursos, artefactos y activos en general.

El mismo autor considera además que las relaciones entre todas las entidades del sistema deben estar soportadas por relaciones económico - financieras y convenios de trabajo bien determinados. Como objetivo tienen compartir segmentos del mercado aprovechando las fortalezas de cada uno de los integrantes del ecosistema, y aumentar la productividad y competitividad de las soluciones que se desarrollen. Sus componentes principales son: dominio de aplicación, familia de productos que conforman la cartera de soluciones, activos del ecosistema generalmente representados por componentes COTS¹, arquitectura que potencie la integración de las soluciones, conjunto de los modos de producción de las entidades del ecosistema y el conocimiento relacional asociado a las relaciones entre las entidades.

La autora considera que este último concepto está completo, pues engloba los términos, variables y principios fundamentales para concebir el desarrollo de un ecosistema de software.

A pesar de que en algunos trabajos se evidencia la similitud del desarrollo de ecosistemas de software con el desarrollo de software abierto, no se comprende como un concepto mucho mayor, como podría ser la apertura del ecosistema. Característica empleada para denotar si las relaciones entre los participantes de este son abiertas o cerradas. Relación que iría más allá de solo desarrollar el mismo bajo licencia abierta o privativa, sino que delimitaría las relaciones, principios y conceptos dentro del ecosistema.

Entre las principales razones para la construcción de ecosistemas de software está satisfacer las necesidades de los clientes en un menor período de tiempo y con un mayor retorno de la inversión. La más importante quizás es la creciente necesidad de personalización de las aplicaciones informáticas, según las tendencias mundiales, de manera tal que garantice una mejor competitividad en el mercado [Bosch 2009].

¹Commercial Off-the-shelf Software, se denominan así los componentes cuando son diseñados para ser fácilmente instalados en conjunto o no con otros componentes bases. Ayuda a tener sistemas configurables y personalizados.

Otras de las razones por las que una organización podría emprender una iniciativa de construcción de ecosistemas de software es que aumenta el atractivo ante los usuarios de sus sistemas básicos [Bosch 2012]. También disminuye los costos por concepto de desarrollo y despliegue, y propicia un mejor entendimiento de la arquitectura [Barbosa, et al. 2011].

Relación de los ecosistemas de software con las Líneas de Productos de Software

Las líneas de productos de software (LPS) se presentan como el preámbulo de los ecosistemas de software y se pueden analizar como parte de los modos de producción que estos representan. El concepto de LPS ha sido ampliamente definido [Bosch 2010; Pestano Pino 2011] como el conjunto de productos de software con funcionalidades comunes de un mismo dominio de negocio.

La producción de LPS propicia una disminución de costos asociados a la producción y un aumento en la productividad. En la última década se ha presentado como una de las alternativas para el desarrollo de software con mayor aceptación y adopción [Bosch et al. 2009].

Las LPS proveen la plataforma adecuada para una producción continua y eficiente que sostiene la constitución de ecosistemas de software. Su diferencia fundamental radica en que las LPS tienen un enfoque intra-organizacional, mientras que los ecosistemas están enfocados a relaciones con factores externos a la organización [Bosch, et al. 2009].

El autor Geir Kjetil [Kjetil Hanssen 2010] analiza la evolución de las LPS hacia ecosistemas de software de acuerdo a fenómenos desencadenantes: interés creciente de los involucrados, incrementar la visibilidad y la extensibilidad.

Al analizar la correspondencia entre los conceptos de LPS y SECO se puede concluir que para el desarrollo de ecosistemas, de diverso tamaño y dominio, se aconseja como modo de producción del desarrollo del ecosistema las LPS. El modo de producción se refiere fundamentalmente a la forma de organización del proceso productivo de la LPS, aunque también se debe incluir la de los componentes que se desarrollen de forma independiente.

La diferencia consiste en que el ecosistema en sí incluye otros elementos externos, como es el caso de los participantes antes analizados.

El concepto de *Software Suppliers* o proveedores software podrá estar representado por las LPS y los productos que desarrollen. Si la LPS se encuentra en el mismo entorno donde se despliega o construye el ecosistema, esta relación se convierte en un beneficio aún mayor.

El nivel que se alcance en las variables para evaluar a los ecosistemas, identificados anteriormente, también depende de la efectividad del modo de producción de las LPS que lo constituyan.

Pautas para el desarrollo de ecosistemas de software

En las revisiones realizadas se constató que el desarrollo de ecosistemas no se ha organizado, aunque se distinguen las nociones de algunas fases y buenas prácticas para ello.

Se ha descrito una fase de identificación y análisis de las relaciones dentro del ecosistema, donde se recomienda el uso de SEM (del inglés *Software Ecosystem Modeling*) para la enumeración y modelado de las partes que componen el ecosistema [Yu et al. 2011].

En el trabajo “*A Survey of Associate Models used within Large Software Ecosystems*” [Angeren, et al. 2011] se realiza una comparación entre varios ecosistemas establecidos, con el objetivo de extraer características tipos de los ecosistemas. Entre ellas se identificó la necesidad de definir tanto los objetivos como estrategias para su desarrollo.

El desarrollo de ecosistemas de software ha sido estudiado desde distintos campos y comunidades separadamente, se han relacionado con los modelos de desarrollo para líneas de productos de software y con los modelos de desarrollo de código abierto. Se asocian además con conceptos arquitectónicos como seguridad, estabilidad y disponibilidad [Barbosa, et al. 2011].

Sin embargo estos autores consideran que aún no existe consenso en lo que se considera como ecosistema de software, causado fundamentalmente por ser un campo de estudio relativamente nuevo.

Según [Recena Soto 2012] un ecosistema tiene relacionadas las siguientes buenas prácticas:

- **Construcción:** Se refiere a los procesos básicos del desarrollo de software: Compilación, Empaquetado y Distribución.
- **Generación automática de documentación:** Aborda la idea de concebir la documentación como un artefacto más dentro del proyecto.
- **Pruebas:** Se refiere a los niveles de pruebas unitarias, funcionales, de integración, de sistema, entre otras, que se le realizan a los sistemas.
- **Inspección continua:** Se refiere al control y seguimiento sobre el proyecto y su estado de desarrollo, de manera tal que se detecten fallos a tiempo.
- **Despliegue continuo:** Trata acerca de la constante verificación del despliegue de la herramienta en disímiles entornos, la comprobación ante fallos después del despliegue.

Tener en cuenta estas buenas prácticas propicia la transparencia, agilidad, productividad y trazabilidad, así como un mayor control durante el desarrollo del ecosistema de software.

Se ha encontrado la propuesta de la plataforma Clinker [Recena Soto et al. 2012] que combina aplicaciones y herramientas probadas para el desarrollo de ecosistemas de software. Recopila las mejores prácticas para el desarrollo de ecosistemas, ver Figura 4. Su propósito general es propiciar

esta base tecnológica, incluso brinda un servicio en la nube para aquellos clientes que no posean la infraestructura necesaria.

Sin embargo, pese a ser una aproximación certera, como un enfoque puramente tecnológico esta no ofrece una solución metodológica de cómo desarrollar el ecosistema, qué pasos seguir ni cuáles elementos tener en cuenta para ello. Por lo cual para el desarrollo de ecosistemas de software continúa faltando una guía clara que organice estos elementos coherentemente.

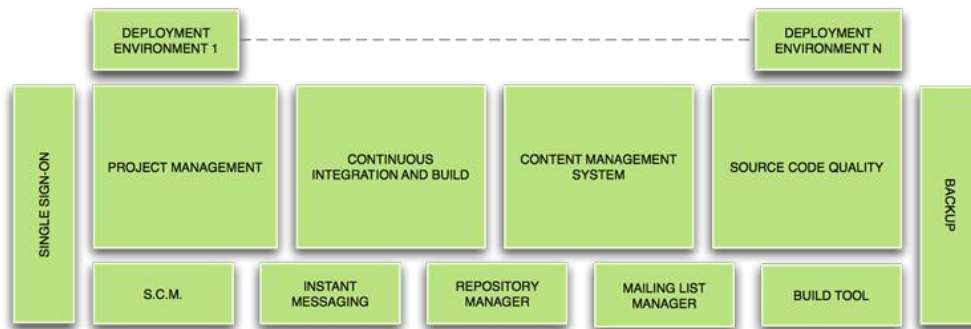


Figura 4. Modelo Conceptual para el desarrollo de ecosistemas de software. Fuente: [Recena Soto, et al. 2012]

Partiendo de las características de los grandes ecosistemas de software revisados, se puede afirmar que por lo general se concibe un sistema núcleo o integrador del ecosistema, que cohesiona coherentemente disímiles componentes relacionados ya sea por tecnología, por negocio o por la información que manejan.

De esta sección se puede concluir que aunque se definen pautas generales para la construcción de ecosistemas, no se evidencian metodologías, modelos o métodos que organicen las actividades para su obtención. Pese a que los enfoques sistémicos o híbridos unifican los mejores elementos de los analizados anteriormente, aún los conceptos analizados carecen de metodologías y métodos de aplicación, se presentan en su mayoría como conceptos abstractos que pueden ser desarrollados en diferentes ámbitos. Se identifica la necesidad de desarrollar métodos para la implementación práctica de los mismos, que organicen los principales elementos a tener en cuenta durante el desarrollo del ecosistema en una guía consistente.

Queda establecido que para desarrollar ecosistemas de menor o mayor tamaño se hace necesario lograr esta interconexión o integración entre todos y cada uno de los sistemas que formen parte de él. No se recomienda concebir ecosistemas con elementos desconectados.

1.2. Análisis de las principales tendencias para la integración de sistemas.

A partir del análisis anterior y la necesidad imperiosa de la integración como elemento/estrategia vital para el desarrollo de los ecosistemas de software, a continuación se realizan análisis de las tendencias para la integración de los sistemas y servicios, o activos, dentro de una organización.

Existen varios enfoques para la integración de sistemas informáticos en una organización, desde los más tradicionales como la integración a nivel de ficheros, a nivel de base de datos y la integración punto a punto. Se considera que estos enfoques ofrecen soluciones básicas que no están en correspondencia con la evolución de las TIC, los usuarios y las expectativas del mercado en general. Es por ello que surgen enfoques más novedosos en cuanto a integración de sistemas se refiere.

Algunos de los criterios y principios en los que se basa el éxito de cualquier iniciativa de integración están directamente influenciados por [Hohpe et al. 2003]:

- Nivel de acoplamiento entre los sistemas: deben minimizarse las dependencias entre los sistemas integrados, de manera tal que los cambios que son inevitables, causen el mínimo de daños posibles.
- Simplicidad en la integración: La integración debe lograrse lo más simple posible, la implementación debe estar bien documentada permitiendo así la integración continua de nuevos sistemas de la empresa.
- Tecnologías: se necesitará para garantizar la integración un conjunto de herramientas que pueden ser de código abierto o estar bajo licencia privativa. La tecnología es un criterio altamente significativo, porque de ello depende un buen rendimiento en la integración.
- Formato de los datos: Para lograr que los sistemas se entiendan, habrá que establecer formatos estándares para los datos que se transmitan o traductores intermedios que cumplan esta función.
- Asincronía: es un principio básico cuando se trata de sistemas que pueden estar interconectados remotamente.

Existen además varios niveles de interoperabilidad en los sistemas de información, según el instituto SEI (del inglés *Software Engineering Institute*) de la universidad Carnegie Mellon, en total 5 [Carney et al. 2004]: Aislado, Conectado, Funcional, de Dominio y de Empresa.

Adicionalmente propone algunos atributos en su modelo LISI (del inglés *Integration and Interoperability Models for Systems of Systems*) relacionados con el aseguramiento del nivel de interoperabilidad [Kasunic, et al. 2004]:

- Procedimientos y políticas: estándares de la organización, políticas de seguridad, datos de administración, operaciones, regulaciones, normas, etc.
- Datos: el dato específico que se transmite o es necesario transmitir, abarca cuestiones de formatos de datos y las estructuras para lograr la interoperabilidad.
- Aplicaciones: las aplicaciones o sistemas de información que intercambiarán y procesarán los datos.

- **Infraestructura:** abarca la mayoría de los recursos de una organización que se relacionan con la integración, tales como hardware, redes y conexiones, equipamiento de seguridad, servicios, etc.

Dentro de los enfoques para lograr la integración de los sistemas que conformen los ecosistemas de software están los denominados patrones de integración empresarial, como soluciones a problemas dentro de soluciones comunes de integración.

Estos han sido organizadas en grandes grupos y caracterizadas para propósitos más específicos. Dichos patrones ofrecen variantes de integración en diferentes escenarios. A continuación se realiza una breve descripción de los estilos y algunos grupos de patrones más significativos.

Patrones de integración empresariales.

Todo tipo de iniciativas han sido desarrolladas según las diferentes opciones de integración que se tengan: transferencias de archivos y ficheros, bases de datos compartidas, invocaciones a procedimientos remotos y mensajería a través de la publicación de mensajes asíncronos. Cómo y cuándo se usen uno u otro estilo depende del caso particular que aplique, pudiéndose combinar si fuese necesario.

- **Transferencia de archivos:** denominado con el término en inglés *File Transfer* [Hohpe, et al. 2003]. Es un estilo de integración básico, pero considerado como el último recurso cuando el resto no es soportado o no sea posible establecer una comunicación coherente. Es un medio natural de comunicación entre sistemas que difieren de plataforma, lenguaje de implementación, etc. Un formato comúnmente usado es XML (del inglés *Extensible Markup Language*).

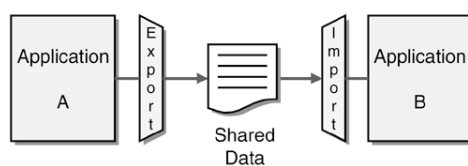


Figura 5. Estilo de integración de transferencia de archivos. Fuente: [Hohpe, et al. 2003].

Se puede inferir que la responsabilidad que se aplica en este estilo es más humana, conlleva realizar este proceso con una frecuencia oportuna, determinar cuáles transformaciones son necesarias para garantizar la correcta lectura y captura de los datos. Por lo cual la autora considera que es un estilo básico que contrae la apariencia de disímiles errores y distorsiones de la información, aunque representa uno de los estilos más fáciles de aplicar en soluciones emergentes. Una de las desventajas del estilo de integración de *File Transfer* es la demora en la actualización. La frecuencia en la sincronización de los datos es vital para la organización y para el resto de los sistemas que están integrados.

- Base de Datos compartida: Si varios sistemas utilizan similares sistemas bases de datos y esquemas de datos es posible una integración mediante este estilo, compartiendo una misma base de datos.

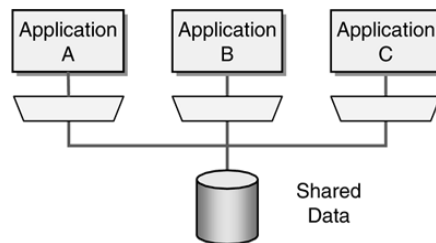


Figura 6. Estilo de integración de base datos compartida. Fuente: [Hohpe, et al. 2003].

La autora considera que el gran reto de este estilo es diseñar una base de datos y esquema común entre varios sistemas, desafío altamente complejo si se tiene en cuenta que cada aplicación tiene su propia semántica. Aunque compartir una misma base de datos tiene un rendimiento relativamente alto estarían altamente acopladas las aplicaciones que se integren mediante este estilo, lo cual acarrea una alta dependencia entre los sistemas y sus funcionalidades. Representa además un riesgo para la seguridad de los datos y el encapsulamiento, elemento que debe gestionarse bien si se aplica este estilo.

- Invocación de procedimiento remoto: los estilos antes descritos permiten que los sistemas puedan compartir o exportar su información, en muchos escenarios de aplicación resultaría insuficiente emplear solo estos dos estilos, teniendo en cuenta los cambios frecuentes sobre las aplicaciones y los datos que manejan. Como estilo el *Remote Procedure Invocation* permite la llamada a funcionalidades de otros sistemas de manera rápida y eficaz, aplicando el principio del encapsulamiento donde cada aplicación es la única propietaria de los datos que maneja y cualquier petición de cambio sobre estos debe pasar por ella.

Existen varias iniciativas para la llamada a estos procedimientos remotos, lo que se conoce como RPC (del inglés *Remote Procedure Call*) y entre los que se encuentran: COM (del inglés *Component Object Model*), CORBA (del inglés *Component Object Request Broker Architecture*), Java RMI (del inglés *Remote Method Invocation*). Sin embargo son los servicios web la tecnología más aceptada y usada en los últimos tiempos, por emplear protocolos como SOAP (del inglés *Simple Object Access Protocol*), HTTP y SMTP y formatos de datos tan comunes como el XML.

La autora considera que lo fundamental en este estilo es crear las interfaces necesarias para que varios clientes puedan acceder a la misma información incluso desde distintos protocolos. El reto estará entonces en alcanzar un buen rendimiento de las llamadas remotas, se tiende a comparar con las llamadas locales, pero pueden tardar más y volver la integración lenta. El acoplamiento sigue siendo alto entre las aplicaciones, las dependencias continúan siendo fuertes y la manera en que se diseñe la integración posiblemente esté altamente sujeta a cambios.

- Mensajería: el principio esencial del estilo de integración *Messaging* es la capacidad que tienen los mensajes de ser asíncronos, lo cual es una ventaja en sistemas distribuidos, permitiendo realizar múltiples peticiones sin esperar la recepción de la respuesta. Es una condición importante si se parte de las asunciones antes descritas, que sugieren que el cambio es inevitable así como las interferencias en las comunicaciones, por lo que si esperar no es una limitante, el rendimiento en este estilo será mucho mayor.

Desde estas premisas se puede diseñar la integración conectando sistemas de manera que se minimicen las dependencias (patrón de asignación de responsabilidades Bajo acoplamiento), pero que a su vez estas tengan alta cohesión con el sistema y la arquitectura de integración en general.

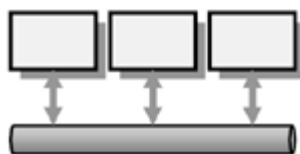


Figura 7. Estilo de integración de mensajería. Fuente: [Hohpe, et al. 2003].

Si bien la mensajería se considera el estilo más recomendado para integrar sistemas, se añaden elementos y definiciones a tener en cuenta. Se le asocian a este varios conceptos como el propio mensaje que se envía, el canal que se utilizará para ello, el consumidor y el emisor del mensaje, el enrutador, el adaptador y el traductor del mensaje, entre otros [Shon 2008].

La mensajería se ve aplicada en los servicios web, tecnología ampliamente empleada, que fomenta la reutilización, el rendimiento, la agilidad, el encapsulamiento, etc. entre sistemas [Erl 2008; Hohpe 2006]. Mediante el acceso a un Bus de Servicios Empresariales (ESB, por las siglas en inglés de *Enterprise Service Bus*) los sistemas podrán publicar y consumir los servicios web ahí alojados, con una infraestructura de seguridad adecuada. El uso de un ESB resuelve las dificultades que introducen las conexiones punto a punto, ya que todas las conexiones y peticiones son intervenidas, controladas y respondidas por el ESB.

A través de la mensajería la respuesta es más inmediata que mediante la transferencia de archivos, mejor encapsulada que las bases de datos compartidas y ofrece más seguridad que las invocaciones a procedimientos remotos.

A consideración de la autora el estilo de mensajería representa el estilo más adecuado para enfrentar los retos actuales para la integración de sistemas y la búsqueda de la interoperabilidad. Asocia conceptos y tecnologías novedosas que están siendo probadas hoy en día con buenos resultados.

Finalmente la autora considera que estos estilos representan la base para realizar cualquier tipo de integración de sistemas y activos de software, partiendo desde los básicos como la transferencia de

archivos hasta la mensajería con mayor grado de dificultad, la selección de uno u otro patrón estará en dependencia de la necesidad particular del escenario donde deba aplicarse.

El propio autor Hohpe [Hohpe 2004] ha propuesto otros criterios y clasificaciones para decidir cuál estilo de integración emplear en las arquitecturas empresariales que se diseñen:

- Según la intención general de la integración.
- Según la vía o medio en que se conecten los sistemas.
- Según los mecanismos de interacción/comunicación entre las aplicaciones.

Arquitectura Orientada a Servicios.

Otra de las tendencias y enfoques para resolver la problemática de la integración es la Arquitectura Orientada a Servicio (SOA, del inglés *Service-Oriented Architecture*). Esta es considerada un paradigma o enfoque de arquitectura empresarial que maneja pequeñas capacidades, sistemas o servicios web distribuidos, con total compromiso con las necesidades de la organización [OASIS 2008]. Representa la evolución de los tradicionales estilos arquitectónicos, incrementando los beneficios de estos [TOGAF 2007].

Las relaciones que se establecen entre SOA y los ecosistemas de software es clara, ambos se enfocan en integrar componentes y productos dentro de la organización en un entorno determinado, de manera tal que aumente o se cree un valor añadido en las organizaciones, con mayor agilidad [Ferronato 2007].

Tiene múltiples beneficios en la organización, entre los que se encuentra su flexibilidad, bajo costo para el desarrollo e integración, agilidad y alineación al negocio [Dillon et al. 2008]. La alineación que promete SOA con las TI y el negocio de la organización permite además que las unidades lógicas distribuidas por su cuenta funcionen de forma autónoma. SOA maneja estas unidades expuestas en forma de servicios, lo cual evidencia la denominada orientación a servicios, independientemente de la tecnología con la que se desarrolla cada una de estas unidades [Erl 2005; Karnouskos et al. 2007].

SOA como una arquitectura abstracta a alto nivel, conlleva la construcción de sistemas y aplicaciones de software con bajo acoplamiento, permitiendo el descubrimiento, composición y consumo de servicios web desde otros sistemas. Hace posible la escalabilidad progresiva, la flexibilidad, la reutilización, y la interoperabilidad, etc. [MCGOVERN et al. 2006; Soto 2007].

Para entender SOA, debe comprenderse bien lo que significa la orientación a servicios, para ello es obligatorio definir los servicios web, aunque SOA no se reduce al desarrollo de servicios web. Los servicios web añaden nuevas funcionalidades para la descripción, publicación, descubrimiento, composición y coordinación a través de la web, soportando interacción y manejo de información mediante los mensajes que transmite.

Entre los principales escenarios donde se emplean los servicios web, y por lo cual están siendo tan bien acogidos, se encuentran la integración de aplicaciones empresariales, muchas de ellas heterogéneas, con diversas fuentes y desarrolladas desde diferentes plataformas. Los servicios web fueron creados para mejorar la interacción hombre-aplicación y aplicación-aplicación, entre otras razones fueron diseñados para otorgar interoperabilidad a los sistemas [Ramesh Jain 2009].

Entre los múltiples conceptos y tecnologías relacionadas con la orientación a servicios, se encuentra el Bus de Servicios Empresarial. Es necesario destacar que la existencia de un ESB por sí mismo no certifica la concepción de una SOA, aunque sí sienta las bases para adoptarla. Es considerada la estructura sobre la cual se soporta la comunicación entre los servicios, o bien su columna vertebral o *backplane*. Permite tanto la integración entre sistemas internos como externos a la organización. Entre las principales funciones que tiene están: mediación entre los mensajes que se envíen, direccionamiento de los mensajes de forma segura y correcta, coreografía y orquestación de servicios, etc.

En la actualidad existen disímiles ESB en explotación entre ellos Microsoft BizTalk Server ESB, JBoss ESB, WSO2 ESB, IBM WebSphere ESB, etc. [Indrasiri 2011]. Para su selección se deben tener en cuenta criterios tales como el comportamiento de los tipos de adaptadores para protocolos, las transformaciones y tipos de acciones para el procesamiento de los datos, la coreografía y orquestación de servicios, el tratamiento de errores y la extensibilidad de los servicios [DAVIS 2009], entre otros criterios más relacionados a los requerimientos técnicos, como pudieran ser el tipo de licencia de la herramienta, si hay soporte y documentación, etc.

1.3. Análisis de los procesos en una organización.

Existen fundamentalmente tres clasificaciones para los procesos dentro de una organización: procesos estratégicos, de apoyo y sustantivos o claves. Los procesos estratégicos son aquellos que delimitan y elaboran los planes y estrategias de la organización según su misión y objeto social. Los procesos claves o procesos de la cadena de valor producen el servicio o producto según el objeto social, satisfacen determinada demanda y aportan valor para la empresa, mientras que los procesos de apoyo son los destinados a dar soporte al resto de los procesos, desde el aseguramiento logístico y de recursos humanos, para llevar a cabo el resto de los procesos [Sanz et al. 2002].

Desde el punto de vista de los procesos, una organización puede ser de producción continua u orientada a proyectos. Las empresas de producción continua son aquellas que su cadena de valor no dependen de la ejecución de proyectos, mientras que las organizaciones orientadas a proyectos, como puede ser el caso del sector de la construcción o la propia industria de software, responden a objetivos establecidos, como parte de un esfuerzo temporal y finito.

Dentro de la construcción de los ecosistemas se revisan todo tipo de sistemas, pero es necesario señalar la necesidad de nutrir y actualizar la información para la gestión de proyectos con la gestión

de la información básica de la organización. De esta manera la gestión de cada una de las principales áreas de gestión de proyectos será más efectiva y por consiguiente la toma de decisiones.

En la sección posterior se analizan un conjunto de sistemas de gestión empresarial y gestión de proyectos, con el objetivo de examinar las capacidades actuales con las que cuentan para la integración dentro del ecosistema de software que se desea construir.

1.4. Análisis de los mecanismos de integración de sistemas para la construcción de ecosistemas de software orientados a la gestión de proyectos.

Sistemas para la gestión empresarial

Los sistemas ERP se consideran en cierta medida un acercamiento a la integración de los procesos en una organización. Cubren parcialmente algunos de sus procesos de negocio, a través de las funciones de las áreas departamentales, y permiten consultar la información general de lo que ocurre en la empresa para el control y la toma de decisiones [Hendricks et al. 2007; Lazo 2010].

Se analizarán brevemente cuáles son las funcionalidades y mecanismos de algunos de los principales sistemas que gestionan este tipo de información.

VERSAT SARASOLA

VERSAT SARASOLA es un sistema ERP desarrollado por el antiguo Ministerio de la Industria Azucarera (MINAZ) cubana, comercializado por la empresa DESOFT. Es el sistema más desplegado en el país. Cuenta con un total de 12 módulos entre los que se encuentran configuración y seguridad, contabilidad general y de gastos, costos y procesos, análisis económico empresarial y control de activos fijos [EICMA 2010].

Emplea como base datos SQL Server 2000, la interfaz de la aplicación está implementada en Delphi, por lo cual es una herramienta construida bajo tecnologías propietarias. El sistema más fuerte que posee es de Contabilidad.

Este sistema no posee mecanismos fuertes de integración con otras aplicaciones, aunque permite algunas funcionalidades básicas para la interoperabilidad a nivel de ficheros, pues permite exportar los comprobantes contables de la aplicación.

CEDRUX

CEDRUX es la iniciativa de ERP cubano que se desarrolla actualmente por la UCI, específicamente por el centro CEIGE (Centro de Informatización de la Gestión de Entidades) de la Facultad 3, cuyo principal objetivo es sustituir los sistemas ERP que actualmente se comercializan en el país por millonarios costos anuales por pago a licencias. Es un producto que se ajusta a las necesidades a las empresas cubanas, a las legislaciones y normas vigentes. Es una aplicación web, construida sobre

tecnologías libres, por lo que se considera promueve la soberanía tecnológica en Cuba [Lazo Ochoa 2011].

CEDRUX cuenta con mecanismos y protocolos establecidos para la integración de sus subsistemas y demás componentes internos, sin embargo en las definiciones arquitectónicas que se han realizado en algunas investigaciones no se aborda ni profundiza cómo integrar componentes externos a él [Codorníu 2011; Martínez 2010]. Aunque vale señalar que es interés del sistema CEDRUX lograr la interoperabilidad tanto entre otros sistemas ERP vigentes en la organización como con sistemas externos a él [Cobas et al. 2010], lo cual favorece en gran medida las iniciativas que en este sentido se pueden desarrollar.

SAP

Entre unos de los principales ERP del mundo se encuentra SAP AG (del inglés *Systems, Applications and Productos in Data Processing*), una empresa alemana que ofrece una diversidad de productos y servicios. Las aplicaciones de la SAP Business Suite están basadas en una arquitectura orientada a servicios, abierta y sobre la plataforma SAP NetWeaver.

Soporta la integración de los sistemas heterogéneos existentes en las empresas, así como el desarrollo de nuevos sistemas. Ofrece varios niveles de integración: *fronted* (frontal, a través de portales), *information* (intercambiando información a través de sistemas de inteligencia de negocio *Business Intelligence Systems*), *back-end* (de fondo, a través de un servidor de aplicaciones) y nivel de aplicación que permite la integración de los procesos de la organización [Heilig et al. 2008].

SAP NetWeaver está dirigida a disminuir los costos mediante la creación de una arquitectura empresarial que garantice mayor flexibilidad antes los cambios inevitables de los procesos de la organización, así como una mejor integración entre las aplicaciones y sistemas basándose en estándares abiertos que amplíen las posibilidades de interoperabilidad futura [Heilig, et al. 2008; SAP 2012]. SAP ERP se ha logrado integrar satisfactoriamente con varios sistemas de renombre internacional, como es el caso de BizTalk Server [Kabat et al. 2010].

Sistemas para la gestión de proyectos

Los sistemas de gestión de proyectos se pueden considerar sistemas de información empresarial, dado que los proyectos no son unidades externas a las organizaciones, sino que se ejecutan con los propios recursos y esfuerzos de las mismas, y para estas es el principal beneficio. Existen disímiles sistemas para la gestión de los proyectos, a continuación se analizan solo algunos.

GESPRO

Como iniciativa GESPRO ha estandarizado los procesos principales de la gestión de proyectos, en el caso específico de la UCI, de los proyectos de desarrollo de software, permitiendo crear un marco

tecnológico e informativo para la toma de decisiones a diferentes niveles: nivel personal, nivel de proyecto, nivel de centro de producción y nivel de la alta dirección [Piñero Pérez et al. 2011b].

GEPSRO se construyó utilizando los principios de líneas de productos de software LPS y tomando como base el modelo propuesto por el Project Management Institute [PMI 2004]. A partir de un nuevo modelo para el desarrollo basado en líneas de productos de software inspirado en otros modelos existentes la LPS de GESPRO define cinco elementos fundamentales [Pestano Pino 2011; Piñero Pérez, et al. 2011b]:

- Dominio de aplicación: el dominio lo constituye la Gestión de Proyectos.
- Definición de la familia de productos: la LPS que desarrolla el sistema GESPRO y sus personalizaciones para diferentes escenarios, propicia la integración y concepción futura como un ecosistema de software.
- Definición del modo de producción: la definición del modo de producción tanto como de los activos reutilizables se encuentra en documentos técnicos y trabajos previos de tesis [Pestano Pino 2011; Piñero Pérez, et al. 2011b].
- Definición de los activos reutilizables de la línea.
- Definición de la arquitectura que soporta GESPRO.

Sin embargo una parte de la información básica o primaria la manejan otros sistemas externos al cual aún GESPRO no se encuentra interconectado. En algunos casos incluso la comunicación existe pero solo parcialmente, sin el diseño de una arquitectura y protocolos eficientes para ello según el caso.

GESPRO tiene definidos en la Vista de Integración de su arquitectura los protocolos de comunicación entre los diversos subsistemas que conforman la suite. La integración externa por su parte es garantizada en gran medida solo a través de algunas funcionalidades de exportación e importación de ficheros de tipo CSV² de algunos módulos.

El estado actual de integración con otros sistemas es considerado aún incompleto, no se responde a un conjunto de funcionalidades que aportarían información desde el punto de vista gerencial a la organización y proyectos en general. Esto permite definir nuevos niveles y elementos (sistemas informáticos) a medida que se establezcan los protocolos y necesidades de información para la integración.

Microsoft Project

Herramienta profesional del paquete de Microsoft Office ampliamente reconocida y empleada en la mayoría de los proyectos por su facilidad de uso, es una herramienta bajo licencia propietaria.

Administra colecciones de proyectos, facilitando el uso de los diagramas de Gantt para la administración de cronogramas, creación de líneas bases, recursos, estimaciones de tareas, costos, etc.

Ofrece interoperabilidad e integración con productos de la familia Microsoft, exportando para MS Access y MS Excel, importa tareas desde el MS Outlook [Muhlbauer et al. 2007], permite importar y exportar cronogramas en archivos CSV y XML. Por estas y otras razones se considera una herramienta ligera para gestionar proyectos de gran envergadura, incluso dentro de proyectos pequeños no permite encargarse de todas las áreas de conocimiento que la gestión de proyectos abarca. Es un sistema minimalista base sobre el cual se pueden montar sistemas integrales.

Trac

Trac es una herramienta web de código abierto bajo licencia BSD³ escrita en Python, para la gestión y seguimiento de errores de proyectos de software. Provee interfaces para sistemas de control de versiones como Subversion [Project. 2012]. La administración se realiza mediante consolas y ficheros de configuración. Permite manejar hojas de rutas e hitos de proyectos [Muhlbauer, et al. 2007].

Sincroniza desde otras bases de datos externas, a través del comando sqlite de su base de datos PostgreSQL, importando específicamente lo que se requiera. Permite además la importación de ficheros CSV, XLS (ficheros de Microsoft Excel). Ha implementado plugins que permiten la importación desde otros sistemas de trazo de proyectos e incidencias como el Jira, Bugzilla, Mantis, etc.

Los datos son manejados principalmente desde repositorios SVN, base de datos y reportes desde plugins. Sin embargo la generalidad de los datos que se importan es proveniente de herramientas de trazo y no contempla entradas desde sistemas de gestión empresarial u otros tipos de sistemas que aporten información valiosa a los proyectos.

Conclusiones parciales del capítulo

- Los sistemas tanto de gestión de proyectos como de gestión empresarial no conciben fuertes mecanismos para la integración, solo algunas variantes altamente acopladas y poco estandarizadas.
- No existen soluciones integrales entre sistemas de gestión de proyectos y de gestión empresarial que posibiliten adecuada gestión de la información y personalización de productos en la organización y en el mercado.

² del inglés *Comma Separated Values*, es un tipo de documento en formato abierto donde las entradas de datos se separan por coma.

³ Licencia de software libre - Berkeley Software Distribution, similar a la licencia MIT, más permisiva que la GPL, permitiendo contener software no libre en las aplicaciones.

- Del análisis bibliográfico acerca de los ecosistemas se puede concluir que los enfoques planteados no se ajustan a resolver las problemáticas fundamentales de la gestión de proyectos y las características del entorno organizacional donde estos residan, solo establecen pautas generales y elementos a considerar dentro de los ecosistemas.
- A partir del análisis de los principales elementos de los ecosistemas de software se concluye que el desarrollo de estos constituye la variante más clara e idónea para propiciar la integración de los sistemas de gestión de proyectos con otros componentes y sistemas de gestión de información.
- Se considera haber dado cumplimiento al objetivo específico de realizar la revisión bibliográfica, analizando las principales tendencias actuales para el desarrollo de ecosistemas y en particular para satisfacer las demandas de la gestión de proyectos.

CAPÍTULO 2: MODELO DE DESARROLLO DEL ECOSISTEMA DE SOFTWARE PARA LA GESTIÓN DE PROYECTOS.

Introducción

En el presente capítulo se formula la propuesta de un modelo para la construcción de ecosistemas de software para la gestión de proyectos, se incluyen definiciones de procesos para desarrollar el ecosistema a partir de una integración entre los sistemas que lo conformen. A su vez cada proceso está compuesto por actividades con sus respectivas entradas, salidas, herramientas y técnicas. Se definen principios del modelo y conceptos bases, así como roles para desempeñar estas tareas.

2.1. Descripción del modelo.

El modelo que se propone para el desarrollo del ecosistema de software para la gestión de proyectos se compone de un conjunto de principios que guían la aplicación del mismo, la definición de sus conceptos bases, así como una definición de procesos que a su vez se componen por entradas, salidas, técnicas y herramientas, ver Figura 8. Estos elementos se definen y describen a continuación.

El modelo se ajusta a las características de los sistemas de gestión de proyectos, además se orienta a dar solución a las principales problemáticas de la gestión de proyectos planteadas en el Capítulo 1 del presente trabajo.

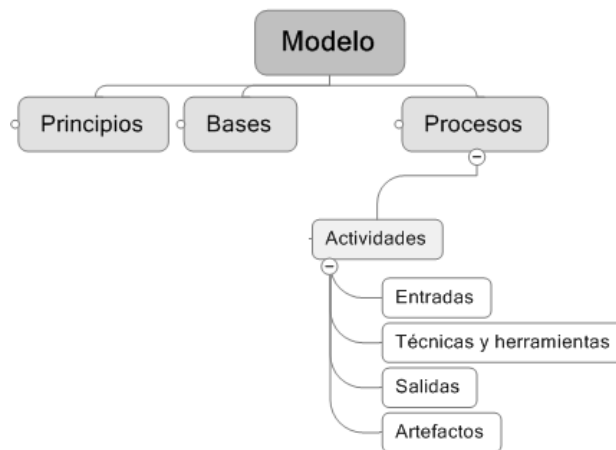


Figura 8. Componentes del modelo. (Fuente: Elaboración propia)

La propuesta que se presenta está basada en tres componentes fundamentales: principios, conceptualización del modelo y procesos que describen su desarrollo.

2.2. Principios generales del modelo.

La autora de la investigación considera que deben ser definidas determinadas premisas antes de comenzar una iniciativa de este tipo, debido a que establecen pautas para el desarrollo exitoso y delimitan responsabilidades de cada parte implicada para ello.

- P1.** Compromiso institucional: se requiere un compromiso de la dirección de la organización y de los departamentos y principales proveedores de los activos que formarán parte del ecosistema.
- P2.** Sustentabilidad: el ecosistema se construirá bajo la premisa de ser sostenible en el tiempo, por lo que en cada uno de sus procesos se tienen en cuenta aspectos que lo garanticen.
- P3.** El modelo se centra en la mejora continua: elemento que se garantiza por parte de los desarrolladores y el equipo de dirección del sistema centro del ecosistema, siguiendo la buena práctica de inspección continua, de manera tal que el ecosistema de software evolucione en el tiempo a partir de la retroalimentación de las deficiencias detectadas. Se relaciona con la gestión del conocimiento y constituye un ciclo dentro del desarrollo del ecosistema guiado por la investigación, la innovación y la producción continua.
- P4.** Los activos dentro del modelo se sustentan en la producción basada en líneas de productos de software.

2.3. Bases del modelo.

Las bases del modelo lo constituyen los principales ejes que conceptualizan el modelo y guían sus procesos.

- Dominio de aplicación: la definición del dominio de aplicación del modelo representa uno de los elementos más importantes. En el caso particular del modelo propuesto el dominio es la gestión de proyectos.
- Familia de productos: representado por la cartera de soluciones que conforman el ecosistema. Contiene tanto activos del ecosistema, como la gama de productos y subsistemas propios del sistema de gestión de proyectos sobre el cual se ajustará al mismo.
- Activos del ecosistema: representados por los elementos COTS del ecosistema, sistemas que se integrarán y el resto de las clasificaciones de activos que se manejarán dentro del modelo.
- Arquitectura: el modelo estará centrado en la arquitectura, los principales elementos que definen los procesos se documentarán en las diferentes vistas arquitectónicas, potenciando principalmente la integración de las soluciones del ecosistema.
- Modos de producción: elemento fundamental que define la forma de desarrollo de los activos y las soluciones del ecosistema, así como la concepción del ecosistema como un todo.
- Conocimiento relacional: una de las bases del modelo está asociada a la definición de las relaciones entre los activos y participantes del ecosistema [Díaz-Balart 2004].

2.3. Definición de los procesos del modelo.

Para el desarrollo del ecosistema de software el modelo se propone seguir una serie de procesos que se describen a continuación. Dentro de cada uno de estos se definen entradas, salidas, técnicas y herramientas, ver Figura 9. En los Anexos del 4 al 7 se modelan cada uno de los procesos.

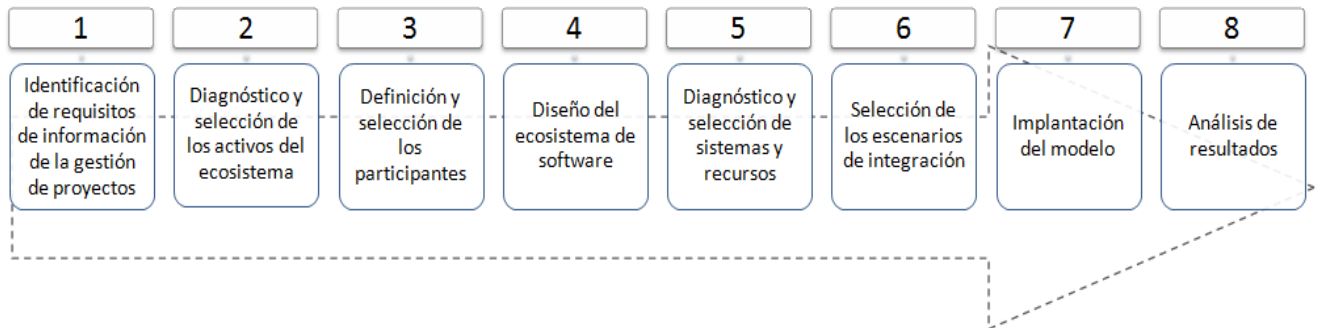


Figura 9. Procesos del modelo (Fuente: Elaboración propia).

2.4.1. Proceso 1: Identificación de requisitos de información de la gestión de proyectos.

Su objetivo principal es identificar las necesidades de información que se requieren para la gestión de proyectos y que justifican la construcción del ecosistema. El equipo de analistas en conjunto con el jefe del proyecto de desarrollo del sistema de gestión de proyectos determinan, especifican e informan cuáles son los requisitos de información del sistema, y de manera general cómo estos deben ser provistos.

Con esto se refiere a los requerimientos que en cuanto a funcionalidades de integración deben satisfacerse, con ciertos criterios o niveles de satisfacción. Principalmente el analista del sistema de gestión de proyectos es quien se encarga de realizar el levantamiento. Se pueden incluir otros roles como arquitectos, algunos gerentes de la organización que emplean el sistema y los clientes e interesados.

En este proceso se ejecuta la actividad:

Actividad 1.1: Identificar requisitos de información.

Entradas:

→ Documentación de proyectos previos de integración: puede que hayan existido iniciativas anteriores de integración del sistema de gestión de proyectos con otros sistemas, lo cual sentará las bases para algunas de las necesidades que puedan aún persistir.

Técnicas y herramientas:

☞ Técnicas de recopilación de información: Para el levantamiento de las necesidades de información, tormenta de ideas, entrevistas personales o grupales con los desarrolladores y gerentes del sistema de gestión de proyectos.

☺ Revisión documental y funcional: resultaría muy efectivo revisar la documentación del sistema de gestión de proyectos y de las funcionalidades actuales que están siendo explotadas en la organización, así como las principales no conformidades que el sistema ha ido capturando, informes de la gerencia de la organización ante fallos o peticiones de información.

Salidas:

← Planilla de requisitos de información: se recogen los requisitos con descripciones de los datos requeridos, formato, la calidad esperada, etc. Ver Anexo 1.

Es importante describir suficientemente los requisitos de información para lograr una claridad del sistema dentro del ecosistema que puede proveerlos, igualmente los datos específicos que se solicitan y el formato o medio de transmisión en el que se solicitan. De esta manera se podrán seleccionar adecuadamente los activos que conformarán el ecosistema.

Las necesidades de información se podrán manifestar en dos sentidos, al ser el centro del ecosistema el sistema de gestión de proyectos, los requisitos de información podrán identificarse como necesidades de información del sistema de gestión de proyectos de los sistemas externos y viceversa.

De tal manera se concibe el ecosistema del sistema de gestión de proyectos, como una unión de sus elementos justificada y cohesiva. Este paso contempla las etapas tradicionales de la ingeniería de requisitos (IR) durante el proceso de desarrollo de software, por lo que tras la identificación se continúa con el resto de los pasos que la IR plantea para la administración, control y seguimiento de los mismos, etc.

2.4.2. Proceso 2: Diagnóstico y selección de los activos dentro del ecosistema.

En la investigación se entiende por activos a los productos, servicios, sistemas, herramientas, la infraestructura de red, de hardware y otros elementos que se puedan agrupar dentro del ecosistema de gestión de proyectos, incluso los activos de los procesos de la organización donde se implante el ecosistema [PMI 2008].

En diferentes publicaciones se definen términos como activos de software [Alvarez et al. 2012], activos de los procesos de la organización [PMI 2008]. Sin embargo en la propuesta se abarcan además otros elementos colaterales dentro de un ecosistema, como pueden ser el hardware y los recursos humanos, aunque los principales activos dentro de la propuesta lo constituirán los sistemas de gestión empresarial dentro de la organización (ERP), así como otros sistemas de información relevantes desde el punto de vista de la gestión de proyectos.

En la Figura 10 se observan los tipos fundamentales de activos que serán tomados en cuenta de una u otra forma dentro del ecosistema, aunque pueden existir otras clasificaciones y elementos a tomar en cuenta.

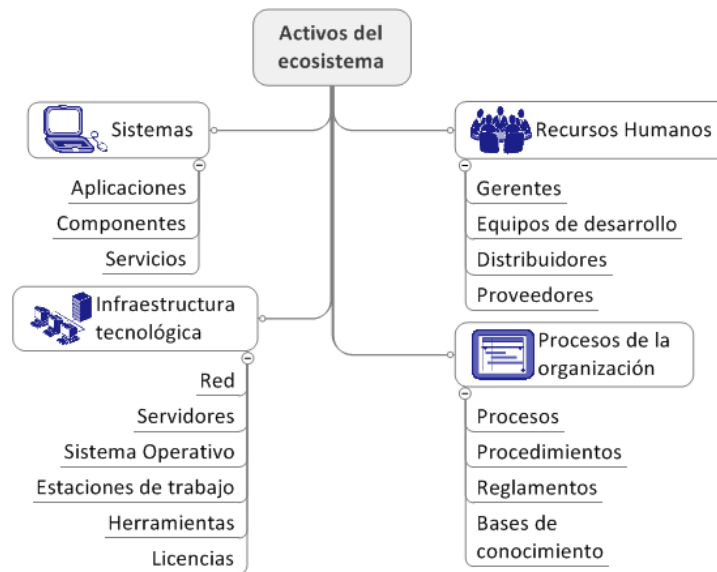


Figura 10. Tipos de activos dentro del ecosistema de software (Fuente: Elaboración propia).

En este proceso se diagnostican esencialmente dos clasificaciones principales de los activos: los recursos y sistemas de la organización, según los grupos de sistemas que fueron conceptualizados y diseñados en el paso anterior. Se revisan las herramientas, los mecanismos, protocolos o interfaces que proveen para integrarse con sistemas externos, así como sus políticas y demás regulaciones de seguridad, que pudieran restringir o limitar los escenarios que se seleccionen.

Este proceso tiene las siguientes actividades: Diagnosticar los sistemas, Diagnosticar los recursos y Seleccionar los activos. En el Anexo 4 se especifica este proceso.

Actividad 2.1: Diagnosticar los sistemas.

Se refiere al diagnóstico de los sistemas que comprenderán el ecosistema, donde el sistema base lo constituye el sistema de gestión de proyectos. Con este diagnóstico se pretende conocer las características y restricciones que tiene cada sistema a considerar, en cuanto a seguridad, lenguaje de programación y tecnología en general se refiere.

Normalmente se deben diagnosticar solo aquellos sistemas de interés colaborativo, que brinden algún tipo de información relativa a la gestión de proyectos, o como un sistema de apoyo a este. Se recomienda no emplear recursos ni tiempo en calificar aquellos sistemas con los cuales con antelación se conoce no será posible una integración.

Entradas:

→ Documentación técnica de la arquitectura: información acerca de las funcionalidades implementadas, etc., así como reportes del estado actual de las funcionalidades del sistema que se puedan recoger en manuales de usuario.

Técnicas y herramientas:

- ☺ Análisis de ecosistemas de software implantados: analizar si en la organización existen ecosistemas de software con antelación, aunque el propósito no sea el de la gestión de proyectos, compuestos por algunos de los sistemas que se deseen diagnosticar en el actual y por tanto de estos se tome información útil.
- ☺ Técnicas de recopilación de información: Revisión documental y funcional del sistema las funcionalidades en tiempo real de los sistemas. Se sugiere la realización opcional además de entrevistas, pues independientemente de las revisiones que el analista y el arquitecto de integración realicen a la documentación de cada sistema, se podrán efectuar entrevistas a arquitectos de cada uno, donde se pueda determinar el estado actual de cada aplicación.
- ☺ Análisis de factibilidad técnica: se sugiere su uso para determinar la factibilidad técnica de los sistemas diagnosticados. Como técnica contiene un conjunto de pasos de los cuales para el diagnóstico solo se llevarán a cabo los dos primeros: análisis de participantes y análisis de problemas. En el Anexo 3 se describe con mayor detalle esta técnica.

Salidas:

← Informe de diagnóstico de los activos del ecosistema: documento dividido en sesiones donde se resuman los resultados del diagnóstico según cada tipo de activo. En este caso particular se especificará por cada sistema las funcionalidades que posee actualmente, las restricciones de seguridad, etc., así como su relación directa con cada requisito de información levantado y si lo satisface o no, como se refleja en el Anexo 2.

Actividad 2.2: Diagnosticar los recursos.

Deben diagnosticarse los recursos con los que cuenta la organización que puedan ser relevantes para asumir la integración de los activos de tipo sistema. Principalmente las características y restricciones que determinarán el diseño de los escenarios que luego serán seleccionados. Los activos de tipo recurso pueden incluir tanto los de los procesos de la organización como los de tipo infraestructura tecnológica, sobre todo los relacionados con esta última clasificación.

Entradas:

- Especificación de requisitos de software de los sistemas revisados: para revisar los requisitos no funcionales de seguridad, hardware, software, etc., entre otros que se puedan consultar y ser útiles para el levantamiento de los recursos de la organización de manera general.
- Manuales o informes técnicos acerca de la infraestructura tecnológica de la organización: usualmente estos documentos lo elaboran las direcciones técnicas, de soporte y de seguridad informática, donde se puede obtener diseños de la red, restricciones y planes de seguridad, etc.

Técnicas y herramientas:

- ☞ Técnicas de recopilación de información: Revisión documental, revisión funcional y entrevistas. Este último por si es posible contar con la asesoría del departamento de seguridad y redes de la organización, para revisar la actual infraestructura de las redes en la organización, así como los protocolos permitidos para la comunicación, entre otros datos que resulten relevantes.
- ☞ Análisis de factibilidad técnica: se sugiere su uso para determinar la factibilidad técnica de los activos. Solo se ejecutarán sus dos primeros pasos: análisis de participantes y análisis de problemas. En el Anexo 3 se describe con mayor detalle esta técnica.

Salidas:

- ← Informe de diagnóstico de los activos del ecosistema - actualizado: en una sección dentro del informe se resume el estado actual de cada uno de los tipos de recursos: red, hardware, entre otros.

Actividad 2.3: Seleccionar los activos.

El modelo propone tener en cuenta la evaluación de ciertos indicadores desde el comienzo del diagnóstico, aunque son especialmente útiles para la selección final de aquellos activos que finalmente se escogerán para conformar el ecosistema.

Sobre la base de las vistas arquitectónicas propuestas en [Lazo Ochoa 2011], en particular la Vista de Despliegue e Infraestructura, donde se delimita la necesidad del control de versiones de los activos, el plan de mantenimiento, etc., se sugieren los siguientes indicadores:

- Los sistemas han de ser extensibles y comprensibles: se prevé que los sistemas puedan ser modificados si fuese necesario, para lo cual se impone que sean de código abierto. Esto garantiza el principio del modelo de la sostenibilidad.
- Los activos de la organización deben contar con un mantenimiento regular, o sea, tener soporte en el tiempo, gestionarse su documentación, respuesta a solicitudes de cambio. De manera tal que de acuerdo a las necesidades de información de la gestión de proyecto, se soliciten actualizaciones de algunas de sus funcionalidades, etc.

- Los activos deben ser de fácil comprensión por los desarrolladores, o sea el grado de asimilación deberá ser alto, la curva de aprendizaje debe ser la mínima posible. Se revisan aquellos sistemas propios del equipo de proyecto, la organización, el país y posteriormente los que brinde el mercado mundial si fuese el caso. Esto le otorga prioridad a los productos endógenos, garantizando una disminución en los costos, lo cual representa una característica que luego se traslada al ecosistema.
- Debe existir un equilibrio entre costo-beneficio de los activos que se seleccionen.
- Los activos deben proveer mecanismos y facilidades para la integración, o al menos proveer de un entorno para su concepción y desarrollo rápido.
- Los activos deben responder a los requisitos de información demandados por el sistema de gestión de proyectos, o al menos ser de apoyo en el ecosistema.
- Los sistemas en particular deberán ostentar un considerable posicionamiento en el mercado o buenos resultados tras su utilización. Como buena práctica se recomienda no incluir aquellos activos que no hayan sido probados adecuadamente.

Entradas:

- Informe de diagnóstico de los activos del ecosistema – actualizado.
- Informes de competitividad en el mercado de los sistemas diagnosticados: lo podrán realizar los agentes de marketing y publicidad de la organización, al menos garantizar que el producto esté establecido, visible y reconocido en cierto ámbito del mercado.

Herramientas y técnicas:

- ☺ Análisis de ecosistemas de software implantados: revisar ecosistemas implantados y bajo cuáles premisas se seleccionaron sus activos, de manera tal que se establezcan comparaciones y se extraigan elementos útiles.
- ☺ Análisis de factibilidad técnica: se sugiere el uso de esta técnica para la selección de los activos. De los pasos que contiene la técnica en las actividades anteriores se ejecutaron los dos primeros, en esta actividad se ejecutan los restantes: análisis de objetivos y alternativas, documentación de alternativas y aplicación de modelos de evaluación, y finalmente la selección de las alternativas. Se tienen en cuenta en cada uno de estos pasos la ponderación de los indicadores antes propuestos. En el Anexo 3 se describe con mayor detalle esta técnica.

Salidas:

- ← Informe de selección de activos del ecosistema de software: se emitirá un informe donde a grandes rasgos se mencionen los sistemas diagnosticados haciendo referencia al Informe de

diagnóstico de los activos del ecosistema, y cuáles de ellos fueron seleccionados finalmente para integrar. Se evalúan los criterios para la selección de los activos.

← Acta de compromiso de responsable de cada activo seleccionado: Relacionado con el principio del compromiso institucional es un acta formal, donde cada responsable de activo se comprometa a proveer determinada información o el acceso según se establezca y pacte.

2.4.3. Proceso 3: Definición y selección de los participantes dentro del ecosistema.

Se definirá qué rol asumirá cada activo dentro del ecosistema, atendiendo a las clasificaciones propuestas para los activos dentro del mismo en la Figura 10.

Sistemas e infraestructura tecnológica

La consideración de los sistemas e infraestructura tecnológica como participantes dentro del ecosistema, se aborda a detalle dentro del escenario de integración que se seleccione más adelante, aquí solo basta con contemplarlos como participantes imprescindibles.

Recursos humanos involucrados en el desarrollo

Existen otros participantes que no forman parte directamente en el desarrollo del ecosistema, pero son parte indispensable del mismo. Tal es el caso de los proveedores de licencias, suministros y servicios de la organización concernientes a los activos relacionados con la gestión de proyectos.

Para llevar a cabo la integración de los activos durante el desarrollo del ecosistema se requiere el desempeño de roles y la expresión de las competencias necesarias para asumir y ejecutar satisfactoriamente la integración. Dentro del propio desarrollo del ecosistema los roles principales lo constituyen el jefe de proyecto del sistema de gestión de proyectos, como máximo representante de las decisiones que sobre el ecosistema se tomen; analistas y desarrolladores, tanto del sistema de gestión de proyectos como de los sistemas y demás activos que se determinen incluir, aunque tengan menor responsabilidad durante el desarrollo.

Luego de explorar sobre los roles para el desarrollo de ecosistemas y sistemas en general, en diversos escenarios de aplicación, se tomó como fuente base la investigación del autor René Lazo Ochoa [Lazo Ochoa 2011], así como otras revisiones que al respecto de la definición y propósitos del rol de arquitecto se tuvieron en cuenta [Unde 2008].

De dicha fuente se seleccionaron los siguientes roles: Arquitecto de Datos y Arquitecto de Integración, donde se reutilizan solo algunas de las responsabilidades y competencias definidas según las características de la integración de los activos que componen el ecosistema. Además se podrán emplear desarrolladores y analistas.

En este proceso se ejecuta principalmente una actividad:

Actividad 3.1: Seleccionar participantes del ecosistema.

Entradas:

- Cronograma de desarrollo del ecosistema: se revisa el cronograma elaborado con las actividades y necesidades de recursos humanos.
- Relación de Recursos Humanos: se revisa la disponibilidad, competencias y habilidades del personal para asumir el desarrollo, primero, dentro del equipo de proyecto del sistema de gestión de proyectos y luego en los equipos de proyecto de los restantes activos.
- Informe de diagnóstico de los activos del ecosistema: se revisa el estado actual y la disponibilidad de los participantes, fundamentalmente los de tipo Recursos Humanos que asumirán el desarrollo del ecosistema e interactuarán con el mismo, y los propios sistemas y recursos a integrar.

Técnicas y herramientas:

- ☺ Selección por analogía: se revisan proyectos de desarrollo de ecosistemas, o de integración que resulten similares al desarrollo actual, de manera tal que se establezcan parámetros comunes en la elección de los roles y relaciones dentro del ecosistema.
- ☺ Establecimiento de relaciones contractuales: si fuese necesario, en aquellos casos de participantes de tipo distribuidores, proveedores, consultores externos, comercializadores, entre otros, será necesario establecer, delimitar y formalizar la relación costo-beneficio que tendrán dentro del ecosistema.
- ☺ Organigrama de la organización y/o proyecto: se sitúan dentro del organigrama los roles y la relación de jerarquía para el desarrollo del ecosistema.
- ☺ Asignación de responsabilidades: se descompone el trabajo hasta los niveles más bajos posibles, a través de estructuras de desglose de trabajo (EDT). Luego se asignan los elementos a cada participante.

Salidas:

- ← Plan de Gestión de los Recursos Humanos: se recoge en un plan las evidencias de aquellos RRHH seleccionados por roles y responsabilidades, cómo se gestionarán, etc.
- ← Cronograma y estructura de desglose de trabajo (EDT) actualizado: como parte de la gestión de proyectos se actualizan los recursos dentro del cronograma de desarrollo del ecosistema y los elementos de la EDT de los cuales es responsable.

← Vistas de la Arquitectura del sistema de gestión de proyectos: se recogen los participantes y roles principales dentro del ecosistema con una descripción que se podrá actualizar en los pasos siguientes acerca de las relaciones y tipo de interacción con el ecosistema.

2.4.4. Proceso 4: Diseño del ecosistema de software.

A partir de las necesidades identificadas, los activos y los participantes seleccionados, se diseña el ecosistema de software. Se establecen, describen y estructuran las relaciones dentro del ecosistema de manera general, previo a la selección de los escenarios de integración, debido a que es necesario concebir un diseño candidato en el cual se sustenten los posteriores estilos de integración, ver Figura 11.

En este proceso se ejecuta la actividad:

Actividad 4.1: Diseñar el ecosistema de software.

Entradas:

- Documentación de ecosistemas de software implantados: revisar ecosistemas implantados con anterioridad, donde se puedan revisar diseños anteriores de estos.
- Vistas de la Arquitectura del sistema para la gestión de proyectos: para determinar la integración de los activos nuevos, revisar la descripción de cada una de las vistas, en particular la Vista de Integración y la Vista de Despliegue e Infraestructura.

Técnicas y herramientas:

- ☞ Modelación: El modelado puede ser formal, incluir detalles técnicos y de estructura, o solo un esbozo de los principales componentes. Se pueden emplear tanto modelos abstractos de relaciones entre los componentes del ecosistema, como lenguajes específicos para ellos, tal es el caso del modelado de ecosistemas de software (SEM) analizado en el capítulo 1.
- ☞ Herramientas de modelado: se recomienda el uso de cualquier herramienta básica que permita diagramar a grandes rasgos los niveles del ecosistema, activos y participantes.

Se proponen 5 capas o niveles dentro del ecosistema en correspondencia con las clasificaciones de activos que se proponen en la investigación, aunque en cada capa se pueden incluir activos de otras categorías, ver Figura 11:

- Capa de activos de tipo Sistemas, específicos para la gestión de proyectos: capa donde se encuentra el sistema para la gestión de proyectos como centro del ecosistema, junto a otros subsistemas o módulos que lo compongan.

- Capa de activos de tipo Sistemas, generales para la gestión empresarial: se consultarían los sistemas propios de la organización para la gestión de los recursos humanos, el control de los costos, independientemente de que el sistema de gestión de proyectos cuente con módulos para la gestión interna de estas áreas.
- Capa de activos de tipo Recursos Humanos: abarca todos los recursos humanos involucrados tanto en el desarrollo, despliegue y comercialización de los productos del ecosistema.
- Capa de activos de tipo Infraestructura y Recursos: contiene las plataformas que aseguran el funcionamiento y soporte del ecosistema como un todo, incluyendo plataformas para el intercambio de información, mensajería y la integración de sus componentes.
- Capa de activos de los procesos de la organización: resulta difícil solo contemplar este tipo de activos dentro de una capa, pues pueden estar contenidas en varias, como es el caso de la capa de sistemas de gestión empresarial.

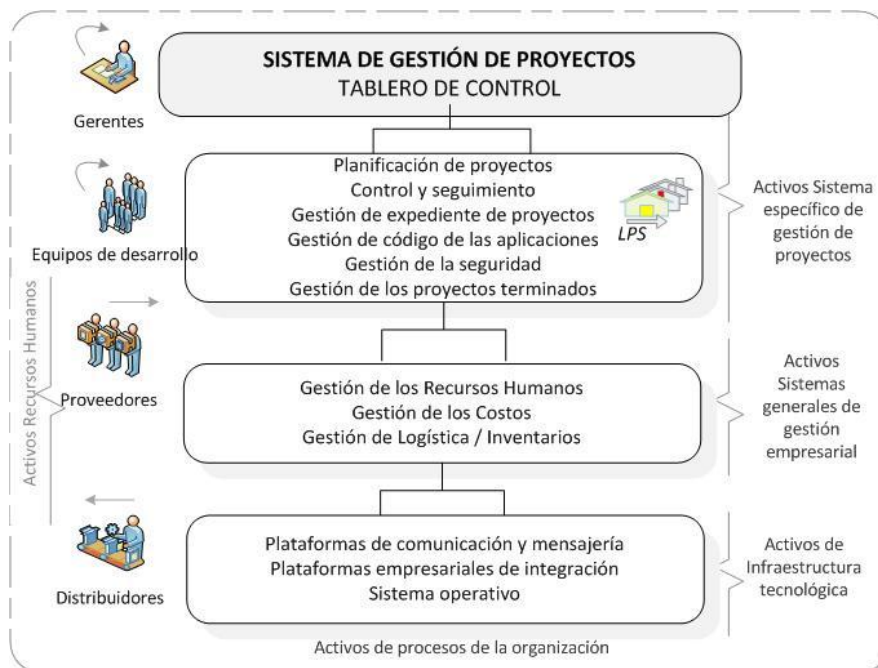


Figura 11. Vista del ecosistema de software para la gestión de proyectos (Fuente: Elaboración propia).

Salidas:

- ← Vistas de la Arquitectura del sistema para la gestión de proyectos-actualizada: se debe recoger dentro de su vista de arquitectura cómo se establecen las relaciones dentro ecosistema, referente a los componentes y demás activos externos. En particular en la Vista de Despliegue e Infraestructura se recogen las especificaciones y sistemas para el control de versiones de cada activo, plan de mantenimiento de los activos, las especificaciones de los requisitos de hardware, software, etc.

2.4.5. Proceso 5: Selección de los escenarios de integración.

Se revisan de acuerdo al diseño del ecosistema planteado anteriormente cuáles escenarios de integración son más propicios para lograr la integración de los activos dentro del ecosistema.

Los escenarios de aplicación de la propuesta están dirigidos a satisfacer las necesidades de integración, de acuerdo a las características que los sistemas seleccionados finalmente. Según las facilidades que los sistemas tengan para la integración se podrá seleccionar uno u otro escenario de integración. El modelo propuesto considera ciertos escenarios que pueden seleccionarse de acuerdo a ello, atendiendo a los distintos enfoques, mecanismos y estilos de integración que se abordaron en el capítulo 1.

Los escenarios se ordenarán el nivel de integración que se alcance, de manera tal que el escenario ideal de integración contemple capacidades de los escenarios anteriores, pudiéndose además concebir híbridos según sea necesario. Los escenarios que se pueden presentar son:

- 1) Integración a nivel de ficheros: En este escenario se requieren mecanismos simples siguiendo el estilo de integración de transferencia de archivos y ficheros.
- 2) Integración a nivel de base de datos: escenario común entre sistemas de la misma organización o que han sido desarrolladas por el mismo equipo de proyecto, donde existe la comunicación y nivel de acceso y seguridad suficientes para permitir el acceso a las bases de datos.
- 3) Integración a nivel de servicios: Dentro de este escenario en particular se verán varias modalidades o sub-escenarios, pues la integración a nivel de servicios, como se explicó en el capítulo 1, está relacionada con diferentes conceptos del estilo de mensajería: canales, mensajes, enrutamiento, etc.
 - a. Integración a nivel de servicios A - Conexiones punto a punto: Este escenario puede aplicarse cuando uno y otro sistema intercambian información directamente, ambos funcionan a su vez como servidor y receptor.
 - b. Integración a nivel de servicios B - Bus de Servicios Empresariales sin adopción de SOA: los sistemas se conectarán siguiendo el estilo de mensajería, pero las conexiones serán mediante un ESB. Los sistemas accederán a este para la publicación y consumo de los servicios mediante una infraestructura de seguridad adecuada.
 - c. Integración a nivel de servicios C - Bus de Servicios Empresariales con adopción de SOA: el uso de un ESB no incluye obligatoriamente la adopción de una SOA ni viceversa, sin embargo se considera que el uso de ambos amplía mucho más sus beneficios.

Adicionalmente a los escenarios que se han presentado puedan darse para la integración, se pueden concebir escenarios híbridos, que mezclen características de varios de los estilos estudiados, de

manera tal que se pueda garantizar la interoperabilidad acudiendo a múltiples mecanismos para ellos según el caso.

En este proceso se ejecuta la actividad:

Actividad 5.1: Seleccionar escenarios de integración.

Se hace necesario establecer ciertos indicadores que permitan seleccionar adecuadamente el escenario más propicio para la construcción del ecosistema de software, atendiendo a los principios descritos, los roles y su capacidad para asumir uno u otro escenario, así como el estado de los activos seleccionados.

Entradas:

- Informe de diagnóstico de los activos del ecosistema: pieza clave para establecer las características técnicas, protocolos y mecanismos que provee cada activo, fundamentalmente los de tipo sistema.
- Documentación de proyectos previos de integración: Documentos, artefactos e informes técnicos de anteriores soluciones de integración por las que haya pasado la organización, o algún sector de esta, que sirvan como base para establecer los tipos de escenarios a asumir. Tales especificaciones pueden estar dirigidas a requisitos de integración por cubrir o cubiertas, requisitos no funcionales, regulaciones de la empresa, etc.
- Lecciones aprendidas: pudieran resultar útiles las experiencias previas que se tengan en el desarrollo de proyectos de integración, en escenarios similares a los que se proponen.

Herramientas y técnicas:

- ☞ Técnicas de recopilación de información: Entrevistas y encuestas que resultan útiles para poder determinar cuál escenario escoger de acuerdo a las experiencias de los miembros del proyecto.

Primeramente es necesario determinar cuáles serán los criterios que para ellos resulten relevantes durante el proceso de elección, se recomienda evaluar el rendimiento, la disponibilidad, la seguridad, entre otros indicadores. Para cada organización y equipo de proyecto estos criterios podrían variar, según las características y la calidad que se pretenda o se pueda alcanzar. Luego se seleccionará el grupo al cual se le pondrá a consideración la evaluación de los criterios por cada escenario. Por último se establecerían los valores posibles para cada criterio y el peso que el criterio tendrá para la final selección.

- ☞ Sistemas basados en el conocimiento: se propone el empleo de sistemas de conocimiento, como pueden ser los Sistemas Basados en Casos (SBC) que infieran a partir de los casos

anteriores de proyectos de integración y de construcción de ecosistemas de software, cuáles escenarios son más propicios emplear según sea el caso.

Salidas:

- ← Vista de Integración del sistema para la gestión de proyectos-actualizada: luego de recoger a detalle los elementos del ecosistema se actualizan las relaciones entre estos.
- ← Descripción de los escenarios de integración seleccionados: Como salida opcional se puede elaborar un documento con la justificación de la elección del escenario o los escenarios a aplicar, de manera que sirva como base para futuros procesos de selección, pudiéndose establecer patrones de los criterios con mayor peso, con éxitos y fracasos de las iniciativas de integración que se establezcan posteriormente.
- ← Base de casos: si se decide emplear un SBC se registra esta nueva selección como un caso dentro del sistema. Posterior al paso de análisis de resultados se puede incluir un nuevo rasgo donde se especifique el éxito o no del proyecto con ese escenario, o el nivel de respuesta a los requisitos de información, entre otros.

2.4.6. Proceso 6: Implantación del ecosistema.

Se realiza la implementación de la integración de los activos de acuerdo al escenario seleccionado, se realizan pruebas, se documenta cada fase de la implementación del ecosistema y finalmente se despliega en diferentes entornos, siguiendo la buena práctica de despliegue continuo.

La implantación del modelo consiste básicamente en implementar la solución de integración de acuerdo al escenario que se haya seleccionado, a partir de aquí se deberán recoger todas las incidencias para el posterior análisis en el proceso 7.

En este proceso se ejecutan las actividades: Implementar, Probar y Desplegar las soluciones de integración del ecosistema.

Actividad 6.1: Implementar la solución de integración del ecosistema.

El objetivo de la actividad consiste, en dependencia de los escenarios que se eligieron, integrar los componentes dentro de cada uno. Para ello se incorporan nuevos roles como desarrolladores y diseñadores de software.

Entradas:

- Descripción de los escenarios de integración seleccionados: en cada caso las salidas y descripciones del diseño, las especificaciones y las recomendaciones de los escenarios

seleccionados previamente, son entregados y revisados, de modo que el desarrollador comprenda lo que debe ser desarrollado.

- Planilla de requisitos de información: en el Anexo 1 se reflejan las necesidades de información levantadas y que son revisadas para entender los datos que deben ser retornados, contiene las especificaciones en cuanto a formato, calidad de servicio esperado y tipos de protocolos, entre otros elementos.
- Informes de diagnóstico de activos del ecosistema: son útiles para comprender las especificaciones de cada sistema y recurso que integren dentro del ecosistema.
- Informe de diagnóstico de activos del ecosistema: se revisan la especificaciones de los activos seleccionados en cuanto a las vistas arquitectónicas que tengan definidas para comprender la estructura interna, protocolos establecidos, lenguajes de programación empleados, entre otros detalles técnicos que faciliten una correcta implementación de la solución de integración.

Técnicas y herramientas:

- Metodología de desarrollo de software: se deberá seleccionar una metodología para ejecutar el proceso de desarrollo de software que establezca los artefactos, pasos y roles dentro de las tres actividades del proceso.
- Herramientas y/o plataformas de integración y desarrollo.
- Repositorios de código y componentes: recurso para compartir componentes COTS y demás activos a integrar en el ecosistema.
- Herramientas de control de cambios: para gestionar los cambios en el código fuente y la documentación asociada al proceso de desarrollo.

Salidas:

- ← Artefactos de implementación en dependencia de la metodología de desarrollo de software que se emplee.
- ← Vistas de la Arquitectura del sistema para la gestión de proyectos - actualizada: Descripción de la arquitectura final del sistema centro del ecosistema de software, atendiendo a elementos de infraestructura, los protocolos permitidos, variantes en caso de fallas, etc.

Actividad 6.2: Probar la solución de integración del ecosistema.

Durante esta actividad se revisan los elementos de la calidad de la solución o si fuera el caso la calidad del servicio QoS (del inglés *Quality of Service*) en términos de disponibilidad, rendimiento, seguridad, entre otros [Khoshafian 2007].

Entradas:

- Descripción de los escenarios seleccionados: se tienen como punto de partida todas las especificaciones dentro de la documentación asociada a los escenarios que se implantaron, para establecer los elementos que se deben revisar y probar.
- Artefactos de la implementación: se revisan otros artefactos de la implementación en dependencia de la metodología de desarrollo de software que se emplee y el escenario de integración seleccionado.

Técnicas y herramientas:

- ↻ Diseño de Casos de prueba: mediante el diseño de casos de prueba se verifica la calidad de los servicios o de las funcionalidades implementadas en el proceso anterior.
- ↻ Herramientas de pruebas: se recomienda emplear herramientas automáticas para revisar los elementos de la implementación del ecosistema.

Salidas:

- ← Documentación de los casos de pruebas realizados: se documentan especificando los resultados en cada caso.
- ← Informe de realización de pruebas: en dependencia de la metodología de desarrollo de software que se emplee se emitirá un documento de No conformidades y de informe de errores según sea el caso.

Actividad 6.3: Desplegar solución de integración del ecosistema.

El ecosistema debe ser desplegado, total o parcialmente, después de probar los elementos y los escenarios de integración que se implementaron internamente. El despliegue se refiere tanto a entornos virtuales durante las pruebas, como entornos reales donde se desplegará finalmente el ecosistema. Siguiendo la buena práctica de despliegue continuo, esta actividad está dirigida a comprobar que el ecosistema es funcional y cumple con los requisitos no funcionales y funcionales a partir de los requisitos de información identificados por el sistema para la gestión de proyectos.

Entradas:

- Vistas de la Arquitectura del sistema para la gestión de proyectos.
- Informe de diagnóstico de los activos del ecosistema: se revisan las especificaciones de los activos, ya sea en el Informe de diagnóstico de los activos o en otros documentos técnicos donde se especifiquen detalles acerca de los requisitos no funcionales de cada sistema,

descripción de componentes y los servicios. Se podrán identificar elementos claves para la infraestructura de despliegue.

- Descripción de los escenarios de integración seleccionados: se revisan las especificaciones de los escenarios de integración que fueron seleccionados. Se identifican los elementos principales de cada escenario que se incluirán en el modelo de despliegue, tanto los que se elaboren para una sección del ecosistema, como el que se conciba de manera general.

Técnicas y herramientas:

- ↻ Modelación: técnica para elaborar los modelos de despliegue, se modelan los nodos físicos y lógicos, y los conectores dentro de la infraestructura del ecosistema para su despliegue exitoso. Se podrán emplear herramientas de modelado si fuese necesario.

Salidas:

- ← Vista de Integración del sistema de gestión de proyectos-actualizada: se especifican las relaciones de los nodos y sus conexiones. Nótese que se tienen en cuenta principalmente los activos de tipo infraestructura tecnológica y de sistemas.
- ← Vista de Despliegue e Infraestructura del sistema para la gestión de proyectos: descripción y diagramación del modelo de despliegue parcial por secciones integradas del ecosistema o ya sea general.

2.4.7. Proceso 7: Análisis de resultados.

Se analizan los resultados tras la implantación del ecosistema en entornos virtuales y reales, se identifican no conformidades en la integración de los activos, así como el cumplimiento de los requisitos de información previstos y que originaron la construcción del ecosistema.

La actividad principal del proceso es:

Actividad 7.1: Cerrar el proyecto de desarrollo del ecosistema.

El análisis de resultados consiste fundamentalmente en analizar si los requisitos de integración fueron satisfechos a partir de las pruebas de aceptación realizadas en el paso anterior, determinar las causas e incidencias por lo cual no se logró la integración de los activos seleccionados, y definir posibles futuros pasos para resolver las demandas de integración. Si se consideró emplear SBC puede registrarse el grado de satisfacción con los requisitos de información del sistema de gestión de proyectos, para futuras tomas de decisiones y análisis.

Entradas:

- Documentación de los casos de pruebas realizados: se revisan los resultados de los casos de prueba.
- Informes de realización de pruebas: se revisan los resultados de las pruebas realizadas a los componentes de los escenarios de integración implementados.
- Vistas de la Arquitectura del sistema de gestión de proyectos: se revisa la documentación de los escenarios implantados de manera que se compruebe si se cumplieron las normativas, principios y demás especificaciones asociadas al escenario implementado. Es una entrada importante pues en dependencia de cuáles escenarios se hayan aplicado serán los tipos y la profundidad de las revisiones que se debieron realizar.
- Planilla de requisitos de información: para determinar el grado de satisfacción de los requisitos identificados en el proceso 1.

Técnicas y herramientas:

- ☞ Técnicas de recopilación de información: Tormenta de ideas, entrevistas y encuestas para la identificación de las lecciones aprendidas por las partes involucradas en el proyecto de construcción del ecosistema, en términos de cómo se ejecutó el proyecto, cuáles fueron las principales deficiencias, etc. Es un factor importante en la detección de problemas y búsqueda de soluciones como vía para potenciar la evolución del ecosistema.
- ☞ Revisión del expediente de proyecto y documentación asociada al mismo: se comprueba que exista documentación para futuros proyectos y transferencias necesarias.

Salidas:

- ← Lecciones aprendidas: las lecciones aprendidas formarán parte de la base de conocimiento de la organización, como un activo para futuros proyectos que se realicen.
- ← Informe de cierre del proyecto de desarrollo del ecosistema: contiene las lecciones aprendidas y los datos generales del cierre del proyecto, tales como las demandas o requisitos de información identificados al inicio de proyecto y cuántos de ellos fueron cubiertos al concluirlo. Se especifica cómo se realizó la prueba de aceptación final del ecosistema y en sentido general cuáles fueron los criterios de calidad que se pudieron certificar. Puede contener referencias a las Vistas de la Arquitectura del sistema de gestión de proyectos para archivar el estado de cada evolución del escenario.

Conclusiones parciales del capítulo

- Se formalizó un modelo de integración compuesto por principios, conceptos bases y 7 procesos fundamentales que se organizan en actividades para guiar el desarrollo del ecosistema de software a través de la identificación, selección, integración y despliegue de los elementos que lo componen.
- Se concibió como parte del modelo una definición de los activos de un ecosistema, así como un conjunto de indicadores para su selección dentro del ecosistema en correspondencia con los principios del modelo.
- Se describieron los procesos y actividades del modelo, proponiéndose además artefactos para algunos de ellos.
- El modelo propuesto permite el desarrollo de un ecosistema de software orientado a sistemas para la gestión de proyectos de fácil adaptación.

CAPÍTULO 3: VALIDACIÓN DEL MODELO PROPUESTO.

Introducción

En el capítulo se describe la evaluación y validación de la propuesta de modelo para el desarrollo del ecosistema de software ajustándolo al sistema para la gestión de proyectos GESPRO. Los componentes lo conforman el propio sistema GESPRO y sus subsistemas como centro del ecosistema, un número de sistemas de gestión empresarial en actual explotación por la UCI, así como otros sistemas y componentes de los centros de desarrollo. Partiendo de la aplicación del estudio de caso mediante el pre - experimento diseñado, se realiza una comparación atendiendo a las variables de la investigación, según los valores que mostraban antes y después de desarrollado el ecosistema. Finalmente se llega a las conclusiones parciales del capítulo.

3.1. Descripción del estudio de caso: desarrollo de un ecosistema de software para el sistema GESPRO en el entorno UCI.

Para la presente investigación se seleccionó para la validación de la misma el pre-experimento como vía para comparar dos muestras, la primera de ellas será el ambiente sin la construcción y la segunda luego de la implantación del ecosistema de software. Se compararán las variables independientes en ambos escenarios. Se seleccionó como sistema de gestión de proyecto base del ecosistema el sistema GESPRO, en particular el desarrollo del ecosistema en el entorno UCI. A continuación se describe el estudio de caso.

En la UCI coexisten diferentes sistemas heterogéneos que manejan información relevante para la universidad, así como para sistemas de gestión de proyectos como es el caso de la Suite GESPRO. La universidad independientemente de los procesos sustantivos relacionados con la formación del profesional tiene centros de desarrollo de aplicaciones de software que desarrollan proyectos con diversos fines. Desde hace pocos años en la UCI se ha estandarizado el uso de GESPRO para gestionar los proyectos, sin embargo no se logra la comunicación e integración deseada con el resto de los sistemas de la universidad.

El entorno de la UCI se complejiza aún más por la dimensión y descentralización de los centros de desarrollo, lo cual trae como consecuencia que los componentes y productos que se desarrollen no se reutilicen ni se compartan sus datos mediante una integración estándar, abierta y flexible. Todo ello conduce a que el desarrollo del ecosistema de software de GESPRO esté condicionado por cierta falta de compromiso institucional y departamental, así como de estrategias claras para establecer la integración y acceso a dichos componentes, servicios y activos en general. Pese a ello GESPRO ha garantizado ciertos niveles de conectividad e interoperabilidad.

A continuación se realizará una valoración de la aplicación del modelo en el entorno UCI, revisando la ejecución de cada uno de los procesos propuestos.

Cumplimiento de los principios del modelo

De manera general los principios del modelo fueron cubiertos casi íntegramente:

- **Compromiso organizacional:** En particular este principio constituye un factor clave que pudo haberse intencionado mejor (ya sea de manera general en la organización, así como en secciones, centros y las principales áreas), a través de estrategias y acciones claras para la participación e integración de sus activos dentro del ecosistema.
- **Sustentabilidad:** El principio se cumplió al garantizar la inclusión de sistemas y componentes de libre acceso y de código abierto, bajo las premisas de la soberanía tecnológica del ecosistema.
- **Mejora continua:** el proceso de desarrollo de software se basó en una inspección continua de la calidad de los artefactos y soluciones desarrolladas.
- **Modo de producción de LPS:** el modo de producción del sistema base del ecosistema GESPRO se realiza a través de la concepción de su LPS [Piñero Pérez, et al. 2011b].

Ejecución de los procesos del modelo

Proceso 1: Identificación de las necesidades de información de GESPRO

Los requisitos de información del sistema GESPRO surgieron dado el interés por ajustarse a las necesidades de la gestión de proyectos de la organización, en especial dada la urgencia de resolver las problemáticas de las principales áreas de conocimiento. Se realizaron entrevistas con especialistas y el líder del proyecto [Piñero Pérez 2012]. De esta manera se obtuvieron y describieron como requisitos del ecosistema de software GESPRO los expuestos en la Tabla 1.

Tabla 1. Requisitos de información solicitados por GESPRO.

ID	Requisitos de integración	Datos que se requieren	Área del conocimiento
RI - 1	Importar los recursos materiales del proyecto	Catálogo de recursos (código, nombre, precio unitario, cantidad de unidades)	Gestión de las adquisiciones
RI - 2	Consultar listado de recursos materiales disponibles		
RI - 3	Importar costos	Código elemento, nombre elemento, centro costo, saldo actual, fecha, subelemento, código de subelemento.	Gestión de Costos
RI - 4	Actualizar los datos de los recursos financieros de los proyectos	Listado de ingresos y gastos asociados al proyecto (en ambas monedas).	
RI - 5	Consultar informe mensual de contabilidad	Datos asociados a centro de costos (facturas por centro de costos)	
RI - 6	Registrar / Actualizar los datos de los RRHH del proyecto	Datos de estudiantes (nombre y apellidos, solapin, año académico, índice académico, nombre madre, nombre padre, provincia, municipio)	Gestión de los Recursos Humanos (RRHH)

		Datos de trabajadores (nombre y apellidos, edad, cargo, años experiencia, categoría científica, categoría docente, tarifa horaria)	
RI - 7	Importar tareas	Datos de las tareas del proyecto (tema, estado, proyecto, prioridad, asignado a, tiempo estimado, etc.)	Gestión del tiempo

Proceso 2: Diagnóstico y selección de los activos dentro del ecosistema.

Los activos a considerar dentro del ecosistema estaban sujetos en gran medida a aquellos desarrollados en la universidad, pues serían de más fácil acceso y revisión durante el diagnóstico. Se analizaron principalmente los activos de tipo sistemas entre las categorías que propone el modelo, y a partir de la evaluación de los indicadores propuestos se realizó el análisis de factibilidad técnica.

Se obtuvo un diagnóstico de cuáles eran los sistemas empleados por cada área o departamento y de qué manera estos satisfacían los requisitos de integración. Se realizó una valoración de los mecanismos que para el intercambio de datos y la integración con otros sistemas proveían. Independientemente a ello se analizó en cada caso de la sustentabilidad de cada uno para su posible inclusión en el ecosistema, así como el resto de los indicadores para la factibilidad técnica de los activos.

Durante el diagnóstico de los activos de tipo infraestructura tecnológica se revisaron los componentes tecnológicos de la organización, prestando especial atención a los componentes y herramientas mediadoras para las conexiones entre los sistemas, de manera que se garantizara la interoperabilidad a los niveles más altos.

En la universidad solo existe públicamente una UDDI (en la dirección <http://uddi.uci.cu>), donde se han publicado servicios de algunos sistemas, de manera tal que a partir de las operaciones de los servicios se acceda a la información. Sin embargo dicha UDDI no cumple con algunos estándares para la interoperabilidad, por ejemplo los WSDL de los servicios publicados no cumplen con las especificaciones para los servicios web WS-*. Adicionalmente, los datos que exponen estos servicios no satisfacen todos los solicitados por los requisitos.

Como recursos de los procesos de la organización se revisaron los contratos, relaciones y convenios existentes con proveedores de software, hardware. Además se revisaron las características de la infraestructura tecnológica de la universidad, determinándose aquellos activos que formarían parte del ecosistema.

Otros activos que se emplean son las bases de datos de proyectos terminados, los servidores de componentes reutilizables, etc. En el ecosistema estarán presentes los principales activos de tipo sistema:

- Paquete de Ayuda a la Toma de Decisiones y Soluciones Integrales (PATDSI) que incluye el Chart Server para los gráficos dentro de los reportes.

- SIG: Sistema de información geográfica.
- Excriba (con base en Alfresco): para la gestión documental.
- CEDRUX: se requiere del mismo una integración con el componente Acaxia para la gestión de la seguridad.
- Servidor - cliente de correo Zimbra.
- Subversion: servidor SVN para el control de cambios del código fuente del sistema.
- Como sistema núcleo del ecosistema el sistema de gestión de proyectos GESPRO compuesto por el núcleo Redmine, el subsistema GESPRO gerencial y el GESPRO operacional.

El sistema GESPRO especifica ciertas definiciones arquitectónicas que sirven como premisas para el desarrollo de un ecosistema de software:

- Se promueve la integración mediante servicios web.
- El estilo arquitectónico empleado para la construcción GESPRO es Arquitectura en N capas, en este caso tiene 7 capas: infraestructura, sistema, presentación, datos, seguridad, tecnología e integración. De manera que se potencia y facilita la adición de nuevos componentes en cada capa.
- La Suite GESPRO se ha construido de manera que pueda integrarse a otros módulos, herramientas y sistemas externos. Para ello desarrolla extensiones de la plataforma base Redmine v 1.0 como componentes COTS. Adicionalmente es capaz de asimilar disímiles estilos arquitectónicos de los componentes que adiciona o a los cuales se conecta.
- La vista de arquitectura de GESPRO combina elementos de varios estilos arquitectónicos: Arquitecturas basadas en componentes y Arquitectura Orientada a Servicios, en gran medida son estilos arquitectónicos heredados del núcleo Redmine. La interoperabilidad con otros sistemas lo facilita a través del uso de los estándares XML y REST [Ordax 2012].

Como recursos además se emplean para el desarrollo y despliegue de GESPRO las distribuciones del sistema operativo Ubuntu, lo cual garantiza el principio de sustentabilidad del ecosistema.

Proceso 3: Definición y selección de los participantes dentro del ecosistema.

Se definieron como principales participantes los miembros del Laboratorio de Gestión de proyectos del Centro de Consultoría y Desarrollo de Arquitecturas Empresariales (CDAE), como principales impulsores, desarrolladores y gestores del sistema GESPRO. De manera general se definieron las interrelaciones entre el resto de los activos.

Además se contó con la colaboración de los centros de desarrollo DATEC, CEIGE, GEYSED para el desarrollo de algunos activos que se integran en el ecosistema.

Proceso 4: Diseño del ecosistema de software

El diseño del ecosistema a través de sus participantes y relaciones y sus principales activos, por cada una de las capas, se muestra en la Figura 12.

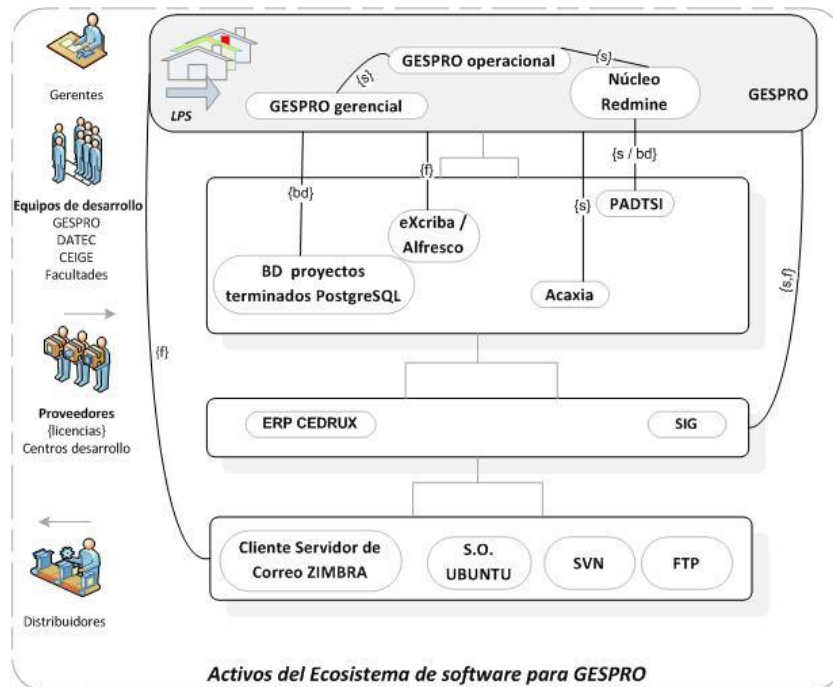


Figura 12. Vista de las capas del ecosistema de software GESPRO (Fuente: Elaboración propia).

Los gerentes de GESPRO y del resto de los activos de tipo sistemas coordinan la integración de cada uno de los activos. Los proveedores de estos, de cualquiera de las categorías propuestas por el modelo de activos, juegan un papel fundamental en el aseguramiento, soporte y mantenimiento de los mismos.

Los distribuidores están representados por miembros del propio equipo de desarrollo de la Suite GESPRO y propician el despliegue del ecosistema hacia otras áreas de la organización u otras externas.

Proceso 5: Selección de los escenarios de integración

A partir del diagnóstico que se realizó, se elaboró un diseño general del escenario, luego se seleccionaron los escenarios de integración y se conformó el diseño final del ecosistema. Para el estudio de caso en particular se eligieron los siguientes estilos de integración, considerando además escenarios híbridos:

- A nivel de ficheros: para el acceso al FTP para consumir archivos CSV y XML a importar dentro del GESPRO. Igualmente mediante ficheros el GESPRO se conecta al Excriba, al Zimbra y el ERP CEDRUX.

- A nivel de base datos: entre el GESPRO gerencial y la base de datos de proyectos terminados.
- A nivel de servicios: con la conexión punto a punto entre el GESPRO gerencial y el operacional, entre el GESPRO, Acaxia y PATDSI.

Proceso 6: Implantación del ecosistema

Para la implementación de los componentes, plugins y demás funcionalidades se empleó el Framework Ruby on Rails v2.3.5, soportado por otras herramientas y librerías para el entorno de implementación. En la Figura 13 se muestra el modelo de despliegue de la Vista de Despliegue e Infraestructura del GESPRO.

Se concibieron nodos periféricos dentro del diagrama tales como la impresora, pues el módulo de Reportes constituye una de las funcionalidades o módulos más empleados dentro de GESPRO, estos reportes son empleados para la toma de decisiones en los proyectos y en la gerencia de la organización. Asimismo se ha incluido un Punto de Acceso (AP) para aquellas conexiones inalámbricas que accedan desde las PC clientes a GESPRO, situación que podría darse en algunos escenarios (centros de desarrollo) donde se despliegue finalmente.

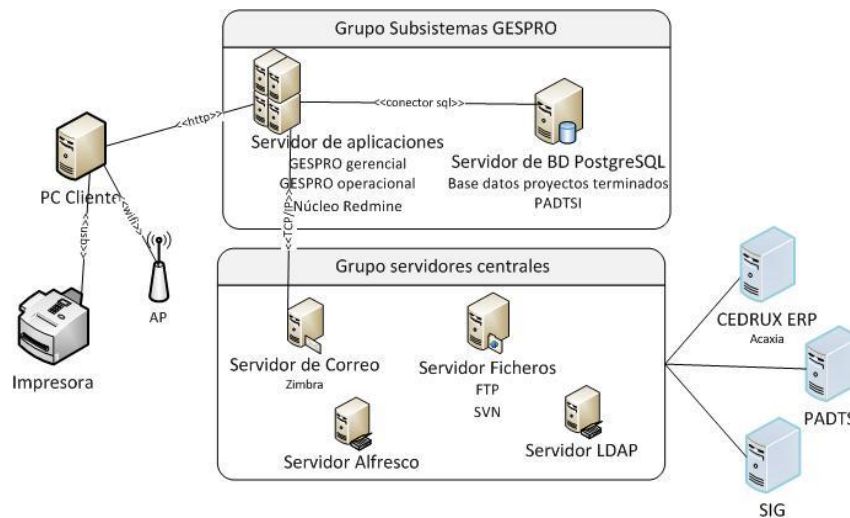


Figura 13. Modelo de despliegue - Vista de Despliegue e Infraestructura (Fuente: Elaboración propia).

Proceso 7: Análisis de resultados

Finalizado el proyecto para desarrollar el ecosistema de software se analizaron las principales problemáticas. Los requisitos de información fueron satisfechos en su totalidad, aunque no se retornaron con todo el formato requerido.

En sentido general el desarrollo del ecosistema fue exitoso, cumpliendo con las expectativas y requisitos de información. Permitió probar el modelo propuesto en un entorno real y complejo, lo

cual sienta las bases para posteriores personalizaciones del ecosistema de software, tanto con el sistema el propio sistema GESPRO como base como con otros sistemas de gestión de proyectos.

Un análisis más exhaustivo se realiza a continuación en correspondencia con las variables de la investigación.

3.2. Análisis de la variable productividad en el desarrollo del sistema GESPRO.

Los análisis que se realizan en las secciones siguientes se harán atendiendo a dos periodos o etapas principales:

- Anterior al desarrollo del ecosistema, del 2009 al 2010: fecha de comienzo del proyecto de personalización de GESPRO con base en Redmine.
- Posterior al desarrollo del ecosistema, del 2010 al 2012: fecha de comienzo en la integración de componentes COTS y activos del ecosistema.

Nivel de Interoperabilidad

La interoperabilidad se midió en cuanto a las funcionalidades de integración desarrolladas en el sistema GESPRO con sistemas externos a él, estableciendo la comparación entre cuántas existían anteriormente a la adopción del ecosistema y luego de implantarlo.

En la Tabla 2 se han relacionado las funcionalidades de integración divididas por las principales áreas de conocimiento de la gestión de proyectos, que forman algunos módulos del GESPRO.

Tabla 2. Cantidad de funcionalidades de integración por área de conocimiento.

	Cantidad de funcionalidades integración - Áreas del conocimiento					Total
	G.Riesgos	G.Alcance	G. Tiempo	G. RRHH	G.Costos	
Anterior al desarrollo del ecosistema	0	0	1	0	0	1
Posterior al desarrollo del ecosistema	1	1	3	2	1	8

La primera versión 13.05 del sistema GESPRO tenía implementadas escasas áreas de gestión de proyectos. Las funcionalidades de integración relacionadas con las áreas son básicamente: a nivel de ficheros y nivel de base de datos. En un inicio solo se incluían funcionalidades de integración en una de las áreas, luego de la implantación del ecosistema, un total de 5 de las principales áreas cuentan con funcionalidades que proveen integración con otros sistemas, ver Figura 14.

Este elemento contribuye a una mejor gestión de estas áreas y de la información que estas manejan para la gestión de proyectos, resolviendo parcialmente uno de los principales problemas de la gestión de proyectos que lo constituye la falta de actualización de la información.

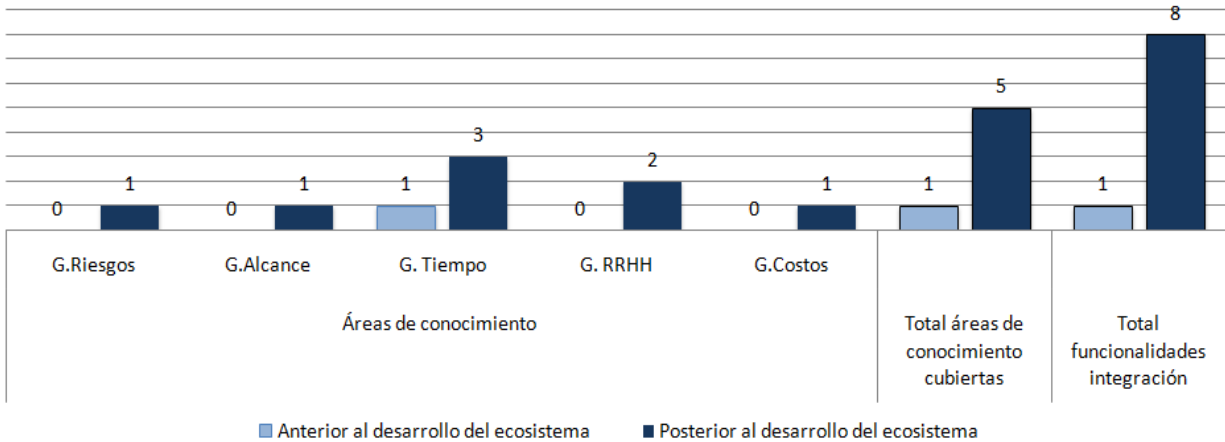


Figura 14. Funcionalidades de integración relacionadas con las áreas de conocimiento.

Adicionalmente a las funcionalidades que responden necesidades claves de las áreas de conocimiento de la gestión de proyectos, se implementaron otras funcionalidades de integración con herramientas para la gestión documental (Alfresco), generación automática de reportes (PATDSI) y el sistema para la información geográfica (SIG).

Eficiencia en el desarrollo del sistema GESPRO

La eficiencia en el desarrollo del sistema GESPRO se midió atendiendo a la eficiencia de los Recursos Humanos que componen el equipo de proyecto del sistema, el esfuerzo de producción y la eficiencia económica de la producción.

En el caso de los proyectos de software libre o que se basan en el desarrollo abierto es difícil estimar los costos, pues provienen de la colaboración de comunidades de desarrolladores, y se calcularía fundamentalmente por el pago de salario de estos desarrolladores.

Eficiencia del equipo de desarrollo

La eficiencia del equipo de desarrollo se midió atendiendo a la cantidad de funcionalidades de integración y la cantidad de RRHH empleados en su desarrollo, antes y después de la implantación del ecosistema, como medida de cuán eficiente ha sido el equipo de desarrollo en la actualización del sistema.

- Cantidad de funcionalidades de integración desarrolladas = CFI
- Cantidad de recursos humanos utilizados = CRHU
- Eficiencia del equipo de proyecto GESPRO = E (ep)

El esfuerzo del equipo de proyecto se calculó como la cantidad de funcionalidades de integración implementadas y la cantidad de Recursos Humanos utilizados en ello.

$$E(ep) = \frac{CFI}{CRHU} \quad (1)$$

En la Tabla 3 se muestran los valores de acuerdo a los resultados del pre-experimento. Como se describiera anteriormente existían solo 2 CFI antes del desarrollo del ecosistema y 11 en total tras su implantación.

Tabla 3. Esfuerzo del equipo de GESPRO anterior y posterior al desarrollo del ecosistema.

Indicadores	Cantidad de funcionalidades de integración (CFI)	Cantidad de Recursos Humanos Utilizados (CRHU)	Eficiencia del equipo de proyecto de GESPRO E(ep)
Anterior al desarrollo del ecosistema	2	10	0.2
Posterior al desarrollo del ecosistema	11	14	0.78

A partir de estos resultados se muestra un aumento en productividad del equipo de desarrollo de GESPRO, luego de implantar el ecosistema de software de GESPRO, ver Figura 15.

La medida del aumento de la eficiencia del equipo de desarrollo está dada fundamentalmente con el aumento de facilidades de integración con los componentes del ecosistema que facilita un desarrollo ágil y abierto.

El esfuerzo de producción está orientado a medir los esfuerzos en el desarrollo interno de GESPRO, evaluados en el indicador anterior, y adicionalmente el esfuerzo aproximado de producir cada uno de los sistemas o componentes con los que se integra GESPRO.

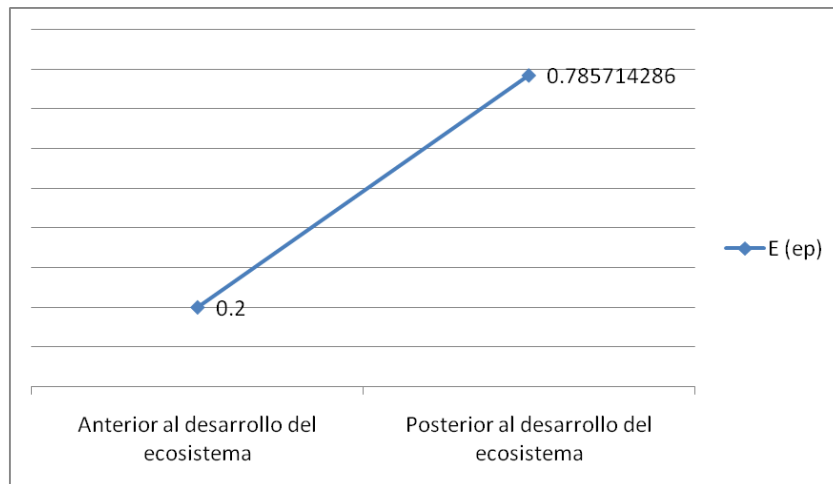


Figura 15. Comportamiento de la eficiencia de los RRHH en GESPRO.

Esfuerzo de producción del ecosistema

A continuación se realiza un análisis del esfuerzo de producción de los activos más relevantes que se integran en el ecosistema GESPRO. El objetivo de esta medición es establecer el esfuerzo que representa desarrollar cada uno de los componentes y sistemas que el ecosistema de GESPRO

integra, en términos de tiempo para el desarrollo y cantidad de Recursos Humanos destinados a ello, ver Figura 16.

- Total de tiempo de desarrollo = Td
- Cantidad de desarrolladores = Cd
- Esfuerzo en el desarrollo = Ed

$$Ed = Cd \times Td \quad (2)$$

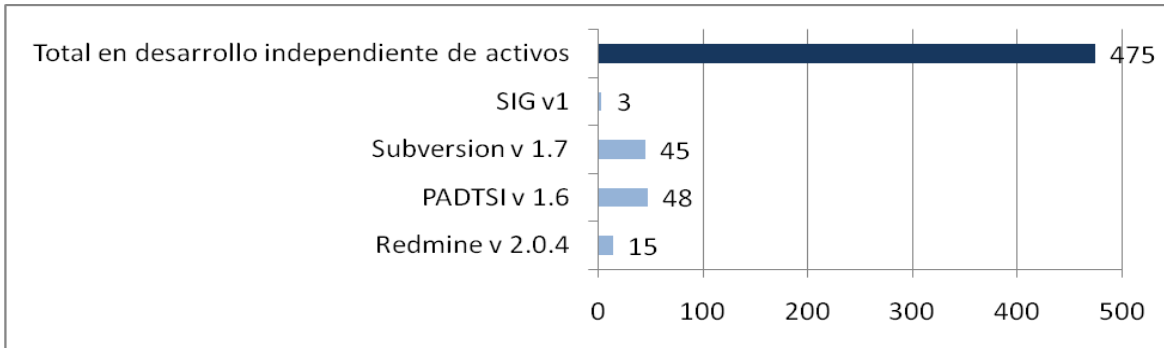


Figura 16. Esfuerzo horas/hombres para el desarrollo de los activos del ecosistema, analizados independientemente.

Eficiencia económica del ecosistema

A partir de entrevistas personales realizadas a miembros y jefes de proyectos se tomó un estimado de desarrolladores por cada sistema, como tiempo de desarrollo el período de lanzamiento de la versión que actualmente se encuentra integrado en el ecosistema.

- Tarifa horaria = Th

$$Ed = Cd \times Th \times Td \quad (3)$$

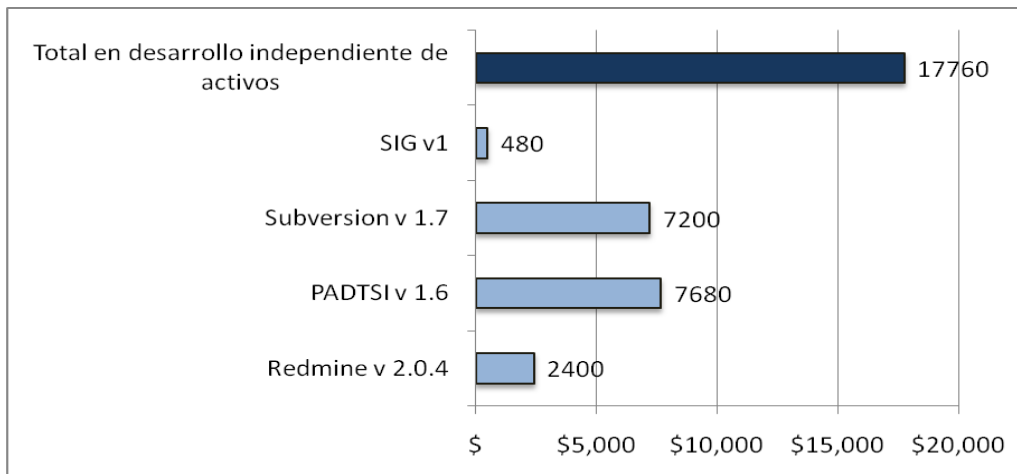


Figura 17. Costo de producción total de desarrollo de los activos del ecosistema, analizados independientemente.

En la Figura 17 se estima el esfuerzo de desarrollo de los principales sistemas y componentes, calculándose el total de esfuerzo que hubiera costado asumir la implementación de cada sistema integrado dentro del ecosistema. Para el cálculo del esfuerzo de cada sistema se tomó como tarifa horaria \$12.00 por desarrollador y un total de 160 horas de trabajo al mes.

Tomando como base para el total del esfuerzo se realizó la suma lineal de los valores de los esfuerzos calculados por cada uno de los sistemas integrados. En la Figura 18 se observa una diferencia notable entre el esfuerzo de producción de los sistemas de forma independiente y del ecosistema de software GESPRO, que reutiliza estos sistemas desarrollados previamente y solo crea funcionalidades para la integración con estos.

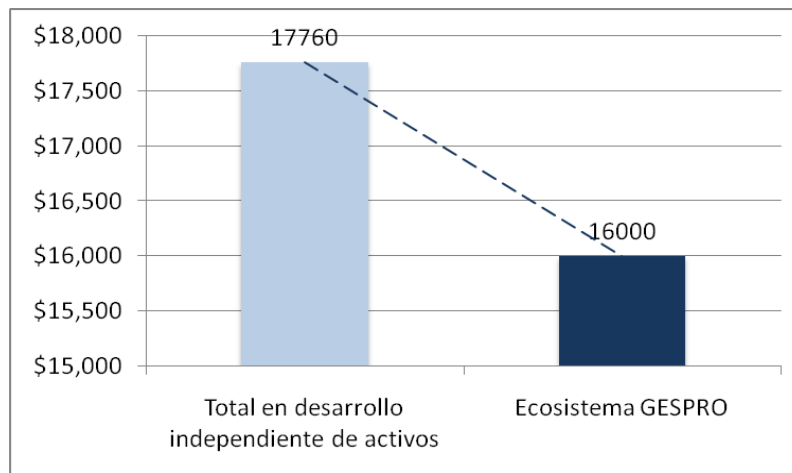


Figura 18. Ahorro de esfuerzo en el desarrollo de los sistemas independientemente al ecosistema.

La diferencia del esfuerzo se calcula en \$1760 pesos, lo cual se traduce en un ahorro de 25 meses y 19 desarrolladores.

El anterior análisis demuestra el impacto económico de la propuesta, expresado en el ahorro que representa el desarrollo del ecosistema de software para la gestión de proyectos, atendiendo a los esfuerzos destinados a ellos, justificándose en una disminución en los costos, tiempos y recursos humanos que hubiesen sido necesarios si se emprendiera el desarrollo separado de los componentes del ecosistema.

3.3. Análisis de las características del ecosistema.

El análisis de las características del ecosistema va centrado en evaluar la robustez del ecosistema y la creación de nichos de mercado como unas de las mediciones más eficaces que se proponen para los ecosistemas de software, en conjunto con la productividad. Igualmente la competitividad como la medida en que el sistema GESPRO aumenta las funcionalidades con respecto a otros sistemas del mercado.

Robustez del ecosistema

Se refiere fundamentalmente al crecimiento de producto, es una característica que identifica las tasas de crecimiento del ecosistema en un período dado, como indicador del valor que este genera. Se miden los siguientes indicadores:

- Cantidad de versiones del sistema núcleo del ecosistema orientado a GESPRO.
- Cantidad de áreas del conocimiento de la gestión de proyectos cubiertas, ya sea por funcionalidades de integración o funcionalidades en general.
- Cantidad de funcionalidades desarrolladas.
- Cantidad de funcionalidades de integración = CFI.
- Cantidad de plugins desarrollados propiamente por el equipo de proyecto.
- Cantidad de activos integrados dentro del ecosistema.

La cantidad de funcionalidades desarrolladas se evaluaron como los requisitos implementados por cada uno de los módulos del GESPRO. En la Figura 19 se constata un crecimiento significativo de las funcionalidades, sumando un total de 734 funcionalidades entre ambos períodos.

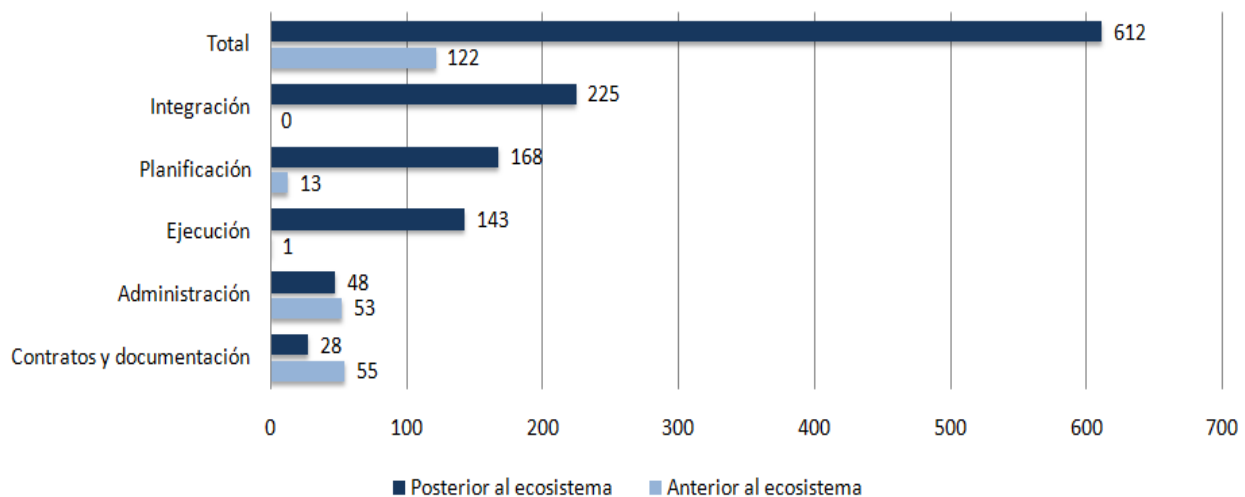


Figura 19. Crecimiento de la cantidad de funcionalidades por módulos de GESPRO.

Las funcionalidades de integración son un subconjunto de las funcionalidades desarrolladas en general por el sistema, como parte de los requisitos levantados, aunque se separan en la gráfica para obtener una vista más detallada del crecimiento, a partir de los valores obtenidos en los análisis anteriores.

En la Figura 20 se evidencia un crecimiento de los principales elementos que integra el ecosistema, se contabilizaron algunos de los indicadores revisados en las secciones anteriores, así como el total de activos del ecosistema.

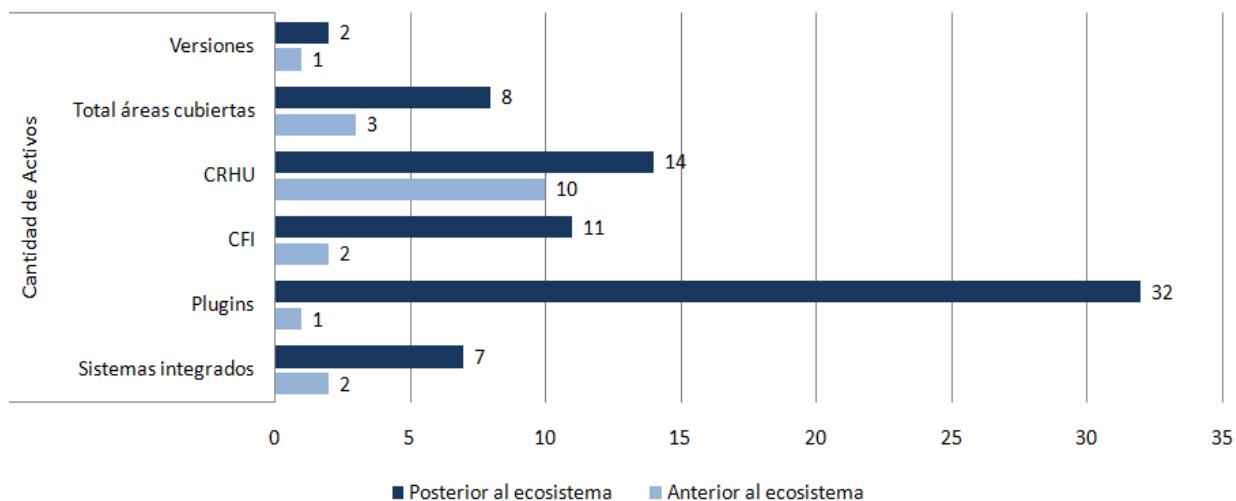


Figura 20. Crecimiento del ecosistema de software atendiendo a los activos.

Creación de nichos de mercado

El indicador de la diversidad del ecosistema, reconocido en la bibliografía como la creación de nichos de mercado, se refiere a la variedad de las soluciones y los servicios que brinde el ecosistema de software al mercado, o ya sea el valor agregado de estas soluciones en el entorno. La creación de nichos de mercado es una medida de la salud del ecosistema, como se revisara en el capítulo 1 de la investigación.

Al revitalizar el ecosistema de nuevos componentes y ofertas que creen la diferencia en un entorno altamente cambiante y competitivo, la creación de nichos sienta la diversidad de un producto en el mercado y la búsqueda incesante de nuevos públicos objetivos.

Entre estos valores y servicios que el ecosistema GESPRO oferta se encuentran:

- Generador de reportes: funciona independientemente como componente COTS.
- PATDSI – Chart Server: componente desarrollado como parte del ecosistema para la inclusión de gráficos en la generación de reportes.
- Servicio de personalización del ecosistema GESPRO, bajo licencia GNU GPL v2.0, para el control y seguimiento de proyectos y la toma de decisiones en los siguientes tres niveles: nivel de proyecto, el nivel de entidad ejecutora y nivel gerencial de la corporación.
- Servicio de despliegue de la plataforma por anillos permitiendo la asimilación de la plataforma y migración de los datos de los proyectos existentes para la nueva plataforma.
- Servicio para la configuración y montaje de la plataforma en entorno de alta disponibilidad y rendimiento en Centros Datos de PDVSA, facilidades para el monitoreo y la salva de seguridad.

- Servicio de suscripciones a actualizaciones automáticas de extensiones a la Suite GESPRO y servicios de soporte técnico.
- Servicio de consultoría y montaje de proyectos de desarrollo para PDVSA identificados en diagnósticos anteriores.
- Servicio de montaje y personalización de programa para la formación continua y autónoma en gestión de proyectos de PDVSA.
- Uso del ecosistema en las asignaturas del Programa de Maestría en Gestión de Proyectos Informáticos: Curso básico de gestión de proyectos, Dirección integrada de proyectos, Gestión de riesgos, Gestión de Alcance y tiempo y Herramientas para la gestión de proyecto. Las cifras de cantidad de egresados de cada curso por año se muestran en la Tabla 4.

Tabla 4. Cantidad de egresados por año de las asignaturas que emplean el ecosistema GESPRO.

Asignaturas	Año 2010	Año 2011	Año 2012	Total
Curso básico de gestión de proyectos	3	82	1	86
Dirección integrada de proyectos	2	1	39	75
Gestión de riesgos	1	3	1	17
Gestión de Alcance y tiempo	2	35	33	70
Herramientas para la gestión de proyecto	0	10	12	22

El uso del ecosistema GESPRO dentro del programa de cursos de la maestría representa tanto un beneficio social y académico del sistema, como la garantía de retroalimentación y renovación científica a partir de las investigaciones que se derivan de la maestría. En la Figura 21 se observan los valores acumulados por año de los egresados de los cursos de la maestría que emplean al sistema GESPRO como parte de su programa.

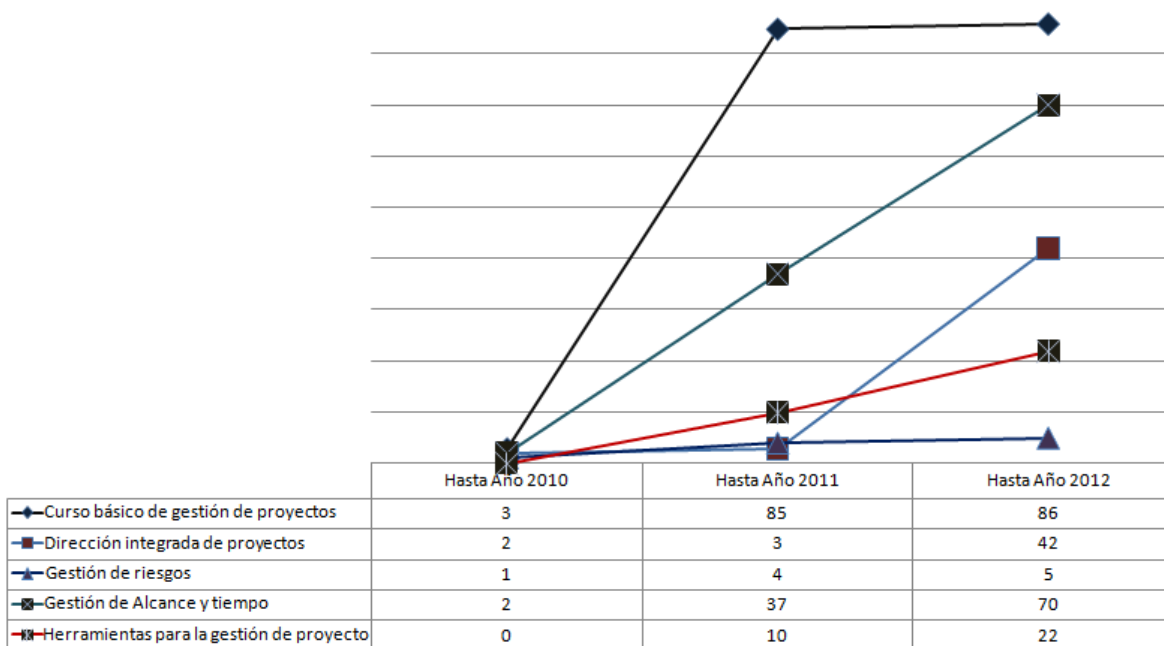


Figura 21. Valores acumulativos de egresados de las asignaturas que emplean el sistema GESPRO.

Se evidencia con el amplio número de servicios que ofrece el ecosistema, como parte de la solución integral, las diversas vías de impacto en el mercado con las que cuenta el ecosistema GESPRO, lo cual lo establece como un sistema competitivo frente a soluciones de este tipo.

3.4. Análisis de la competitividad en el mercado del ecosistema.

Comparación con sistemas de gestión de proyectos

En la Tabla 5 se realiza una comparación entre GESPRO y varios de los sistemas para la gestión de proyectos más empleados, atendiendo al cubrimiento de las principales áreas de conocimiento de la gestión de proyectos, así como otras funcionalidades asociadas a la gestión documental, la gestión de portafolios de proyecto. Además de otras características que evidencian la competitividad de cada producto.

Algunos de estos sistemas fueron analizados brevemente en el capítulo 1 de la investigación, donde se revisaron fundamentalmente los mecanismos que para la integración con otros componentes y sistemas proveían, donde en cada caso los niveles y mecanismos empleados eran altamente acoplados, principalmente a nivel de ficheros.

Se han empleado para identificar el nivel de cubrimiento del área de conocimiento de cada sistema de gestión de proyectos los valores de: Si (++), Parcial (+) y No ().

Tabla 5. Comparación del ecosistema GESPRO con sistemas de gestión de proyectos: Cubrimiento de las áreas de conocimiento.

Sistemas de gestión de proyectos	Cubrimiento de las Áreas de conocimiento								
	Gestión tiempo	Gestión Riesgos	Gestión RRHH	Gestión Costos	Gestión Calidad	Gestión Adquisiciones	Gestión Integración	Gestión Comunicaciones	Gestión Involucrados
MS Project	++		+	++	+	++	++	++	
Open Project	++		+	++		+	++	++	
Redmine	++		+					+	
Ecosistema GESPRO	++	++	++	+	++	+	++	++	+

Como se observa en la comparación de la Tabla 5 el ecosistema GESPRO cubre totalmente todas las áreas de conocimiento en las que se organiza la gestión de proyectos, aunque no de manera total la Gestión de los Costos, la Gestión de las Adquisiciones y la Gestión de los Interesados, módulos en los que aún carece de funcionalidades para el cubrimiento de todos los procesos para su gestión.

En la Tabla 6 se analizan otras funcionalidades y las características de cada sistema, de acuerdo a indicadores relevantes en sistemas de este tipo.

En cuanto a otras características vistas en la comparación con productos similares, se identifican como principales debilidades del ecosistema GESPRO la Gestión Documental, así como los servicios

y facilidades de soporte que brinda, elementos que deben ser potenciados para aumentar su competitividad en el mercado.

Tabla 6. Comparación del ecosistema GESPRO con sistemas de gestión de proyectos: Funcionalidades y Características.

Sistemas de gestión de proyectos	Funcionalidades			Características		
	Soporte	Gestión Documental	Gestión de portafolios de proyectos	Basado en la web	De código abierto	Soberanía tecnológica
MS Project	++	+	+			
Open Project	+				++	+
Redmine	+	++	+	++	++	++
Ecosistema GESPRO	+	+	++	++	++	++

Comparación con ecosistemas de software

Se revisaron varios ecosistemas de software para establecer la comparación, entre los más importantes y reconocidos a nivel mundial en la industria del software como tal: Eclipse [The Eclipse Foundation 2012] y Microsoft [Microsoft 2012]. Se compararon siguiendo tres indicadores fundamentales: la cantidad de activos o productos que integra el ecosistema, el costo de producción total estimado y el costo promedio de cada uno de sus activos.

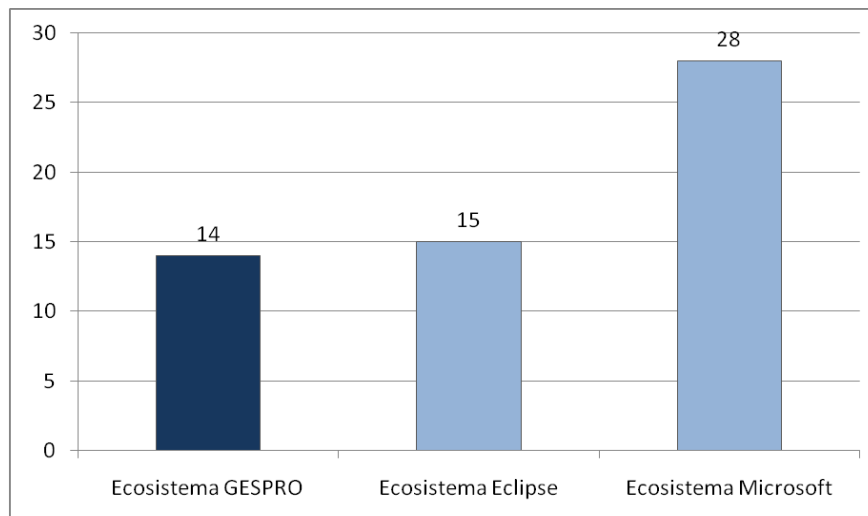


Figura 22. Comparación con ecosistemas: Cantidad de activos.

Para el cálculo del costo de producción estimado por cada activo se tomó el costo de producción total estimado y se dividió entre la cantidad de activos del ecosistema, para establecer un promedio de costo de producción por activo.

En el caso de Eclipse se tomaron los valores estimados a partir de un estudio empleando el método COCOMO [Barahona et al. 2007]. En el caso particular del ecosistema Microsoft, que agrupa aproximadamente 28 productos dentro de su familia, al ser un ecosistema cerrado, no puede estimarse el costo de producción pues ofrece solo la información relativa a los precios de cada

producto en el mercado, lo cual es muy variable. El costo de producción de GESPRO ya fue calculado anteriormente, principalmente por concepto de salarios de sus recursos humanos.

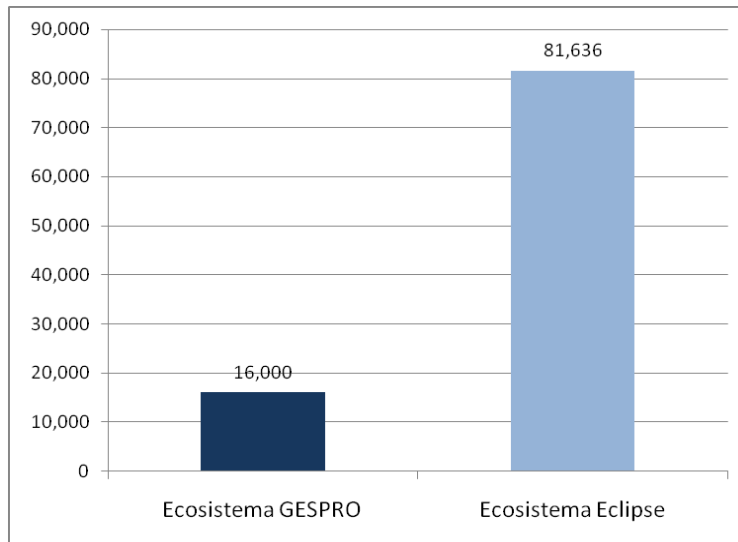


Figura 23. Comparación con ecosistemas: Costo total estimado de producción por ecosistema.

El análisis de la factibilidad económica del desarrollo del ecosistema de GESPRO se realiza principalmente sobre la base del bajo costo de producción necesario para su desarrollo, debido a la reutilización de los activos que integra el ecosistema y el cumplimiento del principio de sostenibilidad del modelo.

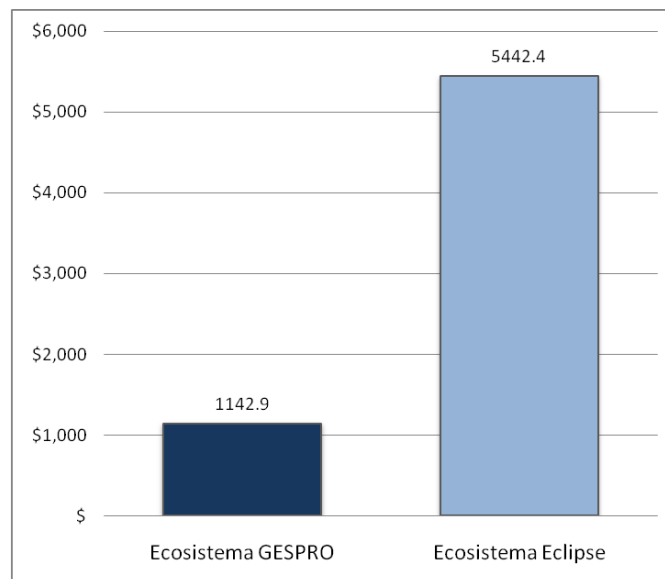


Figura 24. Comparación con ecosistemas: Costo estimado de producción por activo.

El costo de producción individual por cada activo, ver Figura 24, se calculó mediante la división del costo total estimado de producción entre la cantidad de activos que integran el ecosistema.

El desarrollo del ecosistema propició el ahorro de esfuerzos horas/hombres y costo de producción a partir de la integración de activos ya desarrollados, así como la eficiencia de los recursos humanos internos. Elementos que combinados permiten valorar el aporte económico de la propuesta como alto.

Conclusiones parciales del capítulo

- Se aplicó el modelo de desarrollo de un ecosistema orientado a la solución para la gestión de proyectos GESPRO en el entorno de la UCI, instanciándose satisfactoriamente los componentes del modelo.
- Se aumentó la productividad en el desarrollo del ecosistema de software GESPRO con un ahorro de esfuerzo equivalente a 475 horas/hombres y \$1760 en costo de producción.
- Se evaluaron del ecosistema desarrollado como positivas los atributos de productividad, robustez y creación de nichos de mercado.
- El ecosistema GESPRO abarca todas las áreas de conocimiento aunque carece de funcionalidades en para la Gestión de Adquisiciones, Costos e Interesados.
- El ecosistema GESPRO aún tiene debilidad en sus servicios de soporte y en las funcionalidades para la Gestión Documental.

CONCLUSIONES GENERALES

- Los principales autores que trabajan la temática de investigación de ecosistemas de software no definen procedimientos ni modelos para su desarrollo, solo identifican componentes a considerar y buenas prácticas durante el desarrollo.
- No existen acercamientos para el desarrollo de ecosistemas de software que se adapten a las particularidades de soluciones para la gestión de proyectos.
- El modelo propuesto define y organiza el desarrollo de un ecosistema de software orientado a soluciones para la gestión de proyectos compuesto por principios y bases generales, 7 procesos con sus respectivas entradas, salidas, técnicas y herramientas.
- El modelo propuesto fue aplicado exitosamente orientado a la solución de proyectos GESPRO en el entorno de la UCI, concluyéndose que se elevaron los niveles de productividad en el desarrollo de este sistema.
- El aporte económico de la propuesta se evidenció en el ahorro del costo de producción de \$1760 de los activos que se integraron ya desarrollados, lo cual se traduce en el ahorro de 475 horas/hombres.
- Se concluyó que el sistema GESPRO ofrece una solución competitiva ante otros productos de gestión de proyectos en el mercado.

RECOMENDACIONES

Se recomienda para la continuidad de la investigación y en particular para la mejora del modelo propuesto:

- Extender la aplicación de modelo a otros entornos productivos, a fin de demostrar una mayor aplicabilidad y reusabilidad del modelo.
- Añadir funcionalidades para la Gestión de los Costos, Gestión de las Adquisiciones y Gestión de los Interesados en el ecosistema GESPRO para aumentar su competitividad en el mercado frente a productos sustitutos, así como los servicios que ofrece como parte del Soporte y la gestión documental.
- Propiciar dentro del ecosistema de GESPRO mecanismos para la integración con los ERP en explotación en el entorno.

BIBLIOGRAFÍA

- ADOMAVICIUS, G., BOCKSTEDT, J.C., GUPTA, A. AND KAUFFMAN, R.J. TECHNOLOGY ROLES IN AN ECOSYSTEM MODEL OF TECHNOLOGY EVOLUTION. *MISRC Working Papers Series* [Type of Work]. 2005, 2012], pp. 35. Available from Internet:<http://www.misrc.umn.edu/workingpapers/fullpapers/2005/0504_030305.pdf>.
- ALVAREZ, L.G., RUBIERA, I.C. AND JIMÉNEZ, M.S.V. PROPUESTA DE SISTEMA DE ADMINISTRACIÓN DE PRODUCTOS DE SOFTWARE. In *UCIENCIA*. La Habana, Cuba: UCI, 2012, p. 14.
- ANGEREN, J.V., KABBEDIJK, J., JANSEN, S. AND POPP, K.M. A Survey of Associate Models used within Large Software Ecosystems. In S. JANSEN, J. BOSCH, P. CAMPBELL AND F. AHMED. *Proceedings of the Workshop on Software Ecosystems 2011*. Brussels, Belgium, 2011, p. 27-39.
- ARMSTRONG, B., BAKER, B. AND COLLINS, J. IBM z/Transaction Processing Facility: Overview and Enterprise Integration using SOA. *RedBooks. Redguides for Business Leaders* [Type of Work]. 2010, pp. 32. Available from Internet:<<http://www.redbooks.ibm.com/redpapers/pdfs/redp4611.pdf>>.
- BARAHONA, J.M.G., PASCUAL, J.S. AND ROBLES, G. Introducción al software libre [online]. 2.0.1. [Madrid]: 2007 [cited sept 29 2012]. Available from World Wide Web:<<http://curso-sobre.berlios.de/introsobre/2.0.1/sobre.pdf>>.
- BARBOSA, O. AND ALVES, C. A Systematic Mapping Study on Software Ecosystems. In S. JANSEN, J. BOSCH, P. CAMPBELL AND F. AHMED. *Proceedings of the Workshop on Software Ecosystems 2011*. Brussels, Belgium, 2011, vol. 746, p. 15-26.
- BOSCH, J. From Software Product Lines to Software Ecosystems. In *13th International Software Product Line Conference (SPLC 2009)*. San Francisco, CA, USA, 2009, p. 10.
- BOSCH, J. Toward Compositional Software Product Lines. *IEEE Software*, Washington, D.C, USA, May/June 2010 2010, vol. 27, no. 3, p. 29-34.
- BOSCH, J. Software ecosystems: Taking software development beyond the boundaries of the organization. *The Journal of Systems and Software*. Elsevier Inc., 20 March 2012 2012, vol. 85, no. 1453–1454, p. 2.
- BOSCH, J. AND SIJTSEMA, P.B. From Integration to Composition: On the Impact of Software Product Lines, Global Development and Ecosystems. *Journal of Systems and Software*, June 2009. 2009, p. 22.
- BOUCHARAS, V., JANSEN, S. AND BRINKKEMPER, S. Formalizing Software Ecosystem Modeling. In *IWOCE'09*. Amsterdam, The Netherlands: ACM, 2009, p. 10.
- CARNEY, D. AND OBERNDORF, P. Integration and Interoperability Models for Systems of Systems. In *Systems and Software Technology Conference*. Pittsburgh, PA, USA: Carnegie Mellon. Software Engineering Institute, 2004, p. 35.
- CHEN, D., VALLESPER, B. AND DACLIN, N. An Approach for Enterprise Interoperability Measurement. In *Model Driven Information Systems Engineering: Enterprise, User and System Models - MoDISE-EUS'2008*. Montpellier - France, 2008, vol. 341, p. 12.
- COBAS, P.M.N. AND REY, M.G. Estrategia de interoperabilidad para la transferencia de datos entre sistemas ERP en Cuba. In *GESTEC*. La Habana, Cuba: DISAIC, 2010, p. 12.

CODORNÍU, C.L. Solución arquitectónica de la Configuración General del sistema para la parametrización de negocio de Cedrux. In *CENTRO DE INFORMATIZACIÓN DE LA GESTIÓN DE ENTIDADES*. La Habana, Cuba: UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS, 2011, p. 80.

DAVIS, J. *Open Source SOA*. edited by L.W. CYNTHIA KANE. Edtion ed. Greenwich, Connecticut, EEUU: Manning Publications Co., 2009. 449 p. ISBN 978-1-933988-54-2.

DÍAZ-BALART, F.C. *Ciencia, tecnología y sociedad. Hacia un desarrollo sostenible en la Era de la globalización*. Edtion ed. La Habana, Cuba: Editorial científico-técnica, 2004. 143 p. ISBN 959-05-0342-X.

DILLON, T., CHANG, E., MEERSMAN, R. AND SYCARA, K. *Advances in Web Semantics I. Ontologies, Web Services and Applied Semantic Web*. edited by E.J.S. CHANG, KATIA. Edtion ed. Berlin, Heidelberg, Germany: Springer, 2008. 404 p. ISBN 978-3-540-89783-5.

EICMA, D. Versat Sarasola, Un efectivo sistema cubano de contabilidad. 2010. Available from Internet:<<http://www.eimagr.cu/index.php/component/content/article/60-portada/101-versat-sarasola-un-sistema-para-trabajar>>.

ERL, T. *Service-Oriented Architecture: Concepts, Technology, and Design*. Edtion ed. Upper Saddle River, New Jersey, USA: Prentice Hall PTR, 2005. 792 p. ISBN 0-13-185858-0.

ERL, T. *SOA: Principles of Service Design*. edited by M.L. TAUB. Edtion ed. Boston, MA, USA: PRENTICE HALL, 2008. 608 p. ISBN 0-13-234482-3.

FERRONATO, P. Ecosystem Oriented Architecture (EOA) vs SOA. In P.D. F. NACHIRA, A.NICOLAI, M.LE LOUARN, L.RIVERA LÈON. *Digital Business Ecosystems*. Luxembourg: European Commission, in association and with the support of the FP6 projects DBE and OPAALS, 2007, p. 111-118.

FRANTZ, R.Z. Integración de Aplicaciones: Un Lenguaje Específico de Dominio para el Diseño de Soluciones de Integración. In. Seville, Spain: University os Seville, 2008.

HEILIG, L. AND KARCH, S. SAP NetWeaver®: The Official Guide. 2008, pp. 85. Available from Internet:<http://www.thali.ch/pdf/p14927_1.pdf>.

HENDRICKS, K.B., SINGHAL, V.R. AND STRATMAN, J.K. The impact of enterprise systems on corporate performance: A study of ERP, SCM, and CRM system implementations. *Journal of Operations Management*, 2007, vol. 25, p. 65–82.

HOHPE, G. Integration Styles. 2004, vol. 2012, [cited abril 2012]. Available from Internet:<http://www.enterpriseintegrationpatterns.com/ramblings/08_integrationstyles.html>.

HOHPE, G. Conversations Between Loosely Coupled Systems. In *SD West*. Santa Clara, CA, USA: Google, 2006, p. 15.

HOHPE, G. AND WOOLF, B. *Enterprise Integration Patterns: Designing, Building, and Deploying Messaging Solutions*. edited by I. PEARSON EDUCATION. Edtion ed. Boston, MA, USA: Addison Wesley, 2003. 736 p. ISBN 0-321-20068-3.

HUANG, P. ESSAYS ON INNOVATION ECOSYSTEMS IN THE ENTERPRISE SOFTWARE INDUSTRY. In *Management*. Georgia, USA: Georgia Institute of Technology, 2010, vol. Doctor of Philosophy in the College of Management, p. 174.

IANSITI, M. AND LEVIEN, R. *The New Operational Dynamics of Business Ecosystems: Implications for Policy, Operations and Technology Strategy*. Edtion ed. Cambridge, Massachusetts, Estados Unidos., 2002. 115 p.

INDRASIRI, K. An answer to Enterprise Service Integration. In *WSO2Con*. Waters Edge, Sri Lanka, 2011, p. 27.

IPMA, I.P.M.A.-. *ICB- IPMA Competence Baseline*. Edition ed. Nijkerk, The Netherlands, 2006. 212 p. ISBN 0-9553213-0-1.

JANSEN, S., BRINKKEMPER, S. AND FINKELSTEIN., A. Business Network Management as a Survival Strategy: A Tale of Two Software Ecosystems. In S. JANSEN, S. BRINKKEMPER, A. FINKELSTEIN AND J. BOSCH. *Proceedings of the first International Workshop on Software Ecosystems 2009*. Virginia, USA, 2009a, p. 15.

JANSEN, S., FINKELSTEIN, A. AND BRINKKEMPER, S. A Sense of Community: A Research Agenda for Software Ecosystems. In *31st International Conference on Software Engineering - ICSE 09*. Vancouver, Canada, 2009b, p. 4.

KABAT, C., KOKA, N. AND PARTNERS, P.A.M. Unleash SAP using the Microsoft Platform: Using BizTalk Server to Bring Two Worlds Together. 2010, [cited febrero 2012], pp. 30. Available from Internet:<<http://www.microsoft.com/download/en/details.aspx?displaylang=en&id=16323>>.

KARNOUSKOS, S., BAECKER, O., SOUZA, L.M.S.A.D. AND SPIEB, P. Integration of SOA-ready Networked Embedded Devices in Enterprise Systems via a Cross-Layered Web Service Infrastructure. In *Patras, Grecia*. Emerging Technologies and Factory Automation, 2007. ETFA. IEEE: IEEE, 2007, p. 8.

KASUNIC, M. AND ANDERSON, W. Measuring Systems Interoperability: Challenges and Opportunities. In *Software Engineering Measurement and Analysis Initiative*. CA, USA: Carnegie Mellon University, 2004, p. 63.

KHOSHAFIAN, S. *SERVICE ORIENTED ENTERPRISES*. Edition ed. Boca Raton, FL, USA: Auerbach Publications, Taylor & Francis Group, 2007. 465 p. ISBN 0-8493-5360-2.

KJETIL HANSSEN, G. From Agile Software Product Line Engineering Towards Software Ecosystems. In *Faculty of Information Technology, Mathematics and Electrical Engineering*

Department of Computer and Information Science. Trondheim, Norway: NTNU- Norwegian University of Science and Technology, 2010, vol. Philosophiae Doctor. PhD in Information and Communications Technology, p. 190.

LAZO OCHOA, R. Modelo de referencia para el desarrollo arquitectónico de sistemas de software en dominios de gestión. In. La Habana: Universidad de la Ciencias Informáticas, 2011, vol. Máster en Ciencias Técnicas, p. 97.

LAZO, S. Impacto del Enterprise Resource Planning (ERP) en las Empresas. *Revista 360*. Universidad Interamericana de Puerto Rico, 2010, vol. 5, p. 7.

LUNGU, M. Reverse Engineering Software Ecosystems. In *Faculty of Informatics*. Lugano, Switzerland: University of Lugano, 2009, vol. degree of Doctor of Philosophy, p. 208.

LUNGU, M., LANZA, M., GÎRBA, T. AND ROBBES, R. The Small Project Observatory: Visualizing software ecosystems. *Science of Computer Programming*. Elsevier B.V. Ámsterdam, Países Bajos, 3 September 2009 2010, vol. 75, no. 264–275, p. 264–275.

MARTÍNEZ, N.S. Guía metodológica para gestionar la integración de componentes en los proyectos del sistema CEDRUX. In. La Habana: Universidad de las Ciencias Informáticas, 2010, vol. Máster en Ciencias Técnicas, p. 102.

MCGOVERN, J., SIMS, O., JAIN, A. AND LITTLE, M. *Enterprise Service Oriented Architectures. Concepts, Challenges, Recommendations.* edited by J. MCGOVERN, R. DASS, A. FINKELSTEIN AND J. GØTZE. Edtion ed. Netherlands: Springer, 2006. 434 p. ISBN 978-1-4020-3705-4.

MICROSOFT. Microsoft Home Page. Devices and services. In. Redmond, Washington, EUA, 2012, vol. 2010.

MUHLBAUER, R., KLAMBAUER, T., RUPRECHTSBERGER, W. AND SCHAITL, J. Comparison of Project Management Software. In. Berlin, Germany: Institut für Informatik 2007, p. 43.

OASIS. Reference Architecture for Service Oriented Architecture Version 1.0. In JEFF A. ESTEFAN, KEN LASKEY, F.G. MCCABE AND DANNY THORNTON. OASIS, 2008, p. 104.

ORDAX, P.E.N. Propuesta de solución a la Vista de Arquitectura de Integración del Sistema GESPRO 12.05. In. La Habana, Cuba: Universidad de las Ciencias Informáticas, 2012, vol. Ingeniero en Ciencias Informáticas, p. 88.

PACELLI, L. *The Project Management Advisor: 18 Major Project Screw-Ups, and How to Cut Them Off at the Pass.* Edtion ed. Upper Saddle River, New Jersey, USA: Financial Times Prentice Hall, 2004. 167 p. ISBN 0131490478.

PESTANO PINO, H. Propuesta de modelo de desarrollo para líneas de productos de software en centros de producción. In *Laboratorio de Gestión de proyectos. Dirección técnica IP.* La Habana: UCI, 2011, vol. master, p. 80.

PIÑERO PÉREZ, P.Y. Entrevista "Estado actual del sistema GESPRO". In. La Habana, Cuba, 2012, p. 1.

PIÑERO PÉREZ, P.Y., PESTANO PINO, H., ACOSTA, M.V., LÓPEZ, S.T., MATIAS, M.I., JORRÍN, M.G., ROLDÁN, I.M., PÉREZ, J.A.L., RIZO, J.M., FRANQUEIRO, E.A.M., SANTANA, F.N.A., VICTORE, R.D., HERNÁNDEZ, M.T. AND CALAÑA, E.S. GESPRO 11.05 Sistema para la Dirección Integrada de Proyectos para la Gestión de la Producción. In D.T.D.L.P. LABORATORIO DE GESTIÓN DE PROYECTOS. *COMPUMAT.* Cuba, 2011a, p. 48.

PIÑERO PÉREZ, P.Y., PESTANO PINO, H., ACOSTA, M.V., LÓPEZ, S.T., MATIAS, M.I., JORRÍN, M.G., ROLDÁN, I.M., PÉREZ, J.A.L., RIZO, J.M., FRANQUEIRO, E.A.M., SANTANA, F.N.A., VICTORE, R.D., HERNÁNDEZ, M.T. AND CALAÑA, E.S. Paquete de servicios y productos para la Dirección Integrada de Proyectos. In *XVI Forum de Ciencia y Técnica.* La Habana, Cuba, 2011b, p. 56.

PIÑERO PÉREZ, P.Y., PINO, H.P., ACOSTA, M.V., ABELARDO, F.N., LUGO, J.A., ROLDÁN, I.M., AHMED, E., RIZO, J.M., PIÑERO, P.R., IZQUIERDO, M., TORRES, S., PÉREZ, A.D. AND JORRÍN, M.G. Experiencias en el uso de PostgreSQL en el sistema GESPRO, un enfoque práctico. *Revista Cubana de Ciencias Informáticas (RCCI), UCI, La Habana, Cuba, 22/3/2011* 2011c, p. 10.

PIÑERO PÉREZ, Y. 2010. Gestión del Alcance: procesos de planeación, definición y desglose. In *Proceedings of the Gestión del Alcance: procesos de planeación, definición y desglose.*, UCI, La Habana, Cuba, November 19, 2010 2010 UCI.

PMI *Guía de los Fundamentos de la Dirección de Proyectos.* Edtion ed. Newtown Square Pennsylvania: Project Management Institute, 2004. 409 p. ISBN 1-930699-73-5.

PMI *GUÍA DE LOS FUNDAMENTOS PARA LA DIRECCIÓN DE PROYECTOS (GUÍA DEL PMBOK®).* Edtion ed. Pennsylvania, EEUU: Project Management Institute, Inc., 2008. 502 p. ISBN 978-1-933890-72-2.

POPP, K.M. Definition of supplier relationships in software ecosystems as a basis for future research. In P. TYRVÄINEN, S. JANSEN AND M.A. CUSUMANO. *International Conference on Software Business - ICSOB 2010.* Jyväskylä, Finland: Springer 2010, 2010, p. 6.

PROJECT., T.T. The Trac Project. In.: Edgewall Software <http://trac.edgewall.org/>, 2012, vol. 2012.

RAMESH JAIN, A.S. *Semantic Web Services Challenge. Results from the First Year*. edited by T.M. CHARLES PETRIE, HOLGER LAUSEN. Edtion ed. New York / Heidelberg, USA: Springer Science+Business Media, LLC., 2009. 290 p. ISBN 978-0-387-72495-9.

RECENA SOTO, M. Ecosistemas de desarrollo software. Líneas de automatización. In *Ecosistemas de desarrollo software. Líneas de automatización*. Escuela Técnica Superior de Ingeniería Informática. Sevilla, España, 2012, p. 20.

RECENA SOTO, M., MUÑIZ, A., GOMEZ, A., BEAS, J.M., SUTILO, A. AND BENITEZ, F. Clinker Software Development Ecosystem. In. Sevilla , España: klicap <http://clinkerhq.com/>, 2012, vol. 2012.

SAMPIERI, R.H., COLLADO, C.R. AND LUCIO, P.B. *Metodología de la investigación*. Edtion ed. Mexico, D.F., Mexico: McGraw-Hill Interamericana, 2006. 882 p. ISBN 970-10-5753-8.

SANZ, J.B., CALVO, M.A.C., PÉREZ, R.C., ZAPATA, M.A.R. AND PANCHON, F.T. *Guía para una gestión basada en procesos*. Edtion ed. Andalucía, España: Imprenta Berenkintza, 2002. 140 p. ISBN 84-923464-7-7.

SAP. SAP HELP PORTAL <http://help.sap.com/netweaver>. In.: SAP AG, 2012, vol. 2012.

SEHGAL, R. User Empowerment: An Enabler of Enterprise Systems success. In *Faculty of Information Technology*. Queensland University of Technology Brisbane, Australia, 2007, vol. Doctor of Philosophy, p. 341.

SHON, E. Enterprise Integration Patterns. In *Engineering Seminar. CSE 294 (C00): Research Meeting in CSE*. University of California, San Diego, USA: Software & Systems Architecture and Integration Team, 2008, p. 49.

SOTO, M.J.R. SOA, una perspectiva. In *ETSII*. Sevilla, España: Departamento de Lenguajes y Sistemas Informáticos. Universidad de Sevilla, 2007, p. 30.

SUÁREZ, R.C. Metodología de Gestión de proyectos en las administraciones públicas según ISO 10.006. In *Departamento de Explotación y Prospección de Minas*. Principado de Asturias, España: Universidad de Oviedo, 2007, vol. doctoral, p. 312.

THE ECLIPSE FOUNDATION. The Eclipse Foundation - open source community website. In. Ottawa, Ontario, Canada: The Eclipse Foundation., 2012, vol. 2012.

TOGAF. Service-Oriented Architecture (SOA). In. San Francisco, CA, USA: The Open Group, 2007, p. 33.

UNDE, A. The Role of an Architect. Becoming an Architect in a System Integrator. *The Architectural Journal*. Microsoft Corporation, USA, April 2008 2008, no. 15, p. 7-10.

VERA, A.B. Implementación de sistemas ERP, su impacto en la gestión de la empresa e integración con otras TIC. *Conferencia Académica Permanente de Investigación Contable, CAPIC. REVIEW* [Type of Work]. 2006, vol. 4, [cited diciembre 2011], pp. 16. ISSN 0718-4662.

WONG, R.A. Assessment of Open Source Business Strategies in the Domain of Embedded Systems. In *Department of Information Processing Science*. Oulu, Finlandia: University of Oulu, 2011, vol. master, p. 116.

YU, E. AND DENG, S. Understanding Software Ecosystems: A Strategic Modeling Approach. In S. JANSEN, J. BOSCH, P. CAMPBELL AND F. AHMED. *Third International Workshop on Software Ecosystems*. Brussels, Belgium, 2011, p. 65-76.

ANEXOS

Anexo 1. Ficha de necesidades de información.

Captura de necesidades de información	
Id del requisito	<<Id del requisito, ejemplo de formato: RI 1>>
Sistema solicitante	<<Nombre del sistema que solicita la información>>
Información que necesita	<<Enunciado de la necesidad de información>>
Datos solicitados	<<Identificación de los datos solicitados (Dato 1, Dato 2, Dato 3)>>
Frecuencia de actualización	<<Frecuencia de actualización para el requisito, puede ser semanal, quincenal, mensual, trimestral, anual, etc.>>
Requerimiento de su negocio que demanda esta información:	<<Enunciado de la necesidad a partir del negocio (Justificación) que conlleva la necesidad>>
Proceso de negocio al que tributa	<<Área de conocimiento, nombre del proceso o módulo que requiere la información>>
Estructura en la que actualmente recibe la información (si la recibe)	<<Estructura, formato o mecanismo en que actualmente se recibe la información>>
Aplicación o BD donde reside esa información	<<Aplicación (es) que se espera satisfaga (n) la necesidad>>
Mecanismo que usa para extraer la información (si la recibe)	<<Si existe actualmente cual es el mecanismo empleado para acceder a la información>>
Problemas que impiden o limitan el acceso a la información	<<Limitantes y clasificación que imposibilitan tener la información >>
QoS con la que desea recibir la información	<<Calidad del servicio que se espera al recibir la información>>

Anexo 2. Informe de diagnóstico de activos.

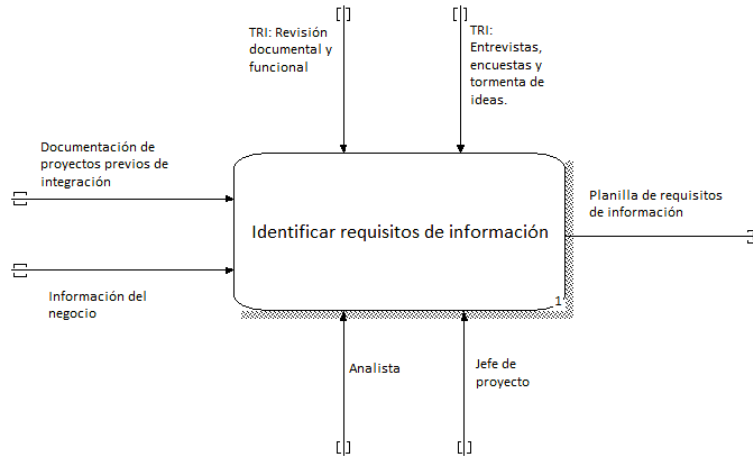
Nombre del sistema o aplicación	<<Especificar nombre>>	
Id y nombre del requisito de información relacionado	<<Especificar el id y el nombre del requisito que se asocia al levantamiento del estado de la aplicación >>	
Funcionalidades que posee actualmente		
Nombre Funcionalidad 1	¿La provee?	¿Cómo la provee?
<<Nombre de la funcionalidad de la aplicación>>	<<Sí/No>>	<<ficheros, servicio web, etc.>>
Diagnóstico de recursos		
Clasificación del recurso	<<Especificar una clasificación o proponer otra según el caso: Red / Hardware / Software / Recursos Humanos / ... >>	
Nombre del recurso	<<Especificar el nombre específico del recurso >>	
Área o entorno del recurso	<<Área o entorno donde se encuentra el recurso particular>>	
Estado actual del recurso		
Protocolos que soporta actualmente		
Restricciones de seguridad		
Estado esperado		
Protocolos a garantizar para la integración		

Anexo 3. Descripción de la técnica de Análisis de factibilidad técnica.

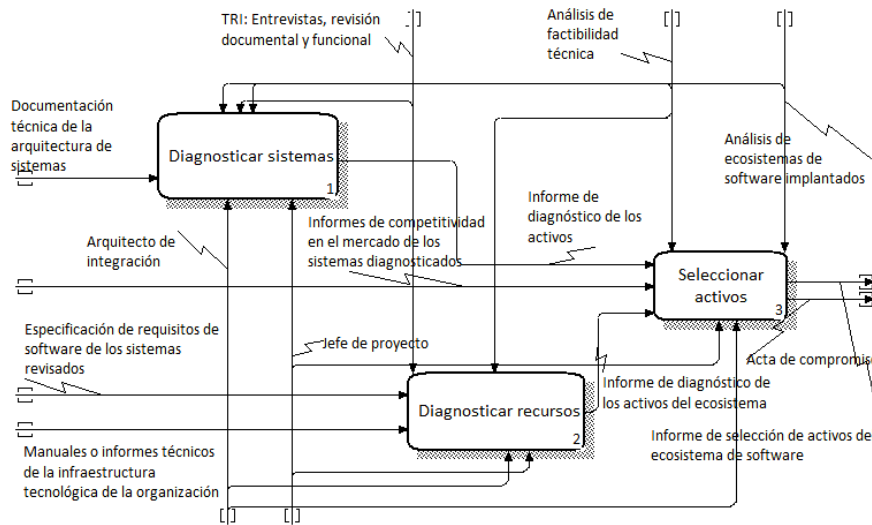
- Paso 1: Análisis de participantes: Análisis cualitativo de los participantes mediante su interés e impacto.
- Paso 2: Análisis de problemas: Identificar el problema núcleo a través de un árbol de problemas.

- Paso 3: Análisis de los objetivos: Identificar los objetivos y un único objetivo núcleo a través de un árbol de objetivos reformulado a partir de un árbol de problemas.
- Paso 4: Documentación de alternativas.
- Paso 5: Selección de alternativas: Mediante la toma de decisiones, criterios de expertos, entre otras técnicas.

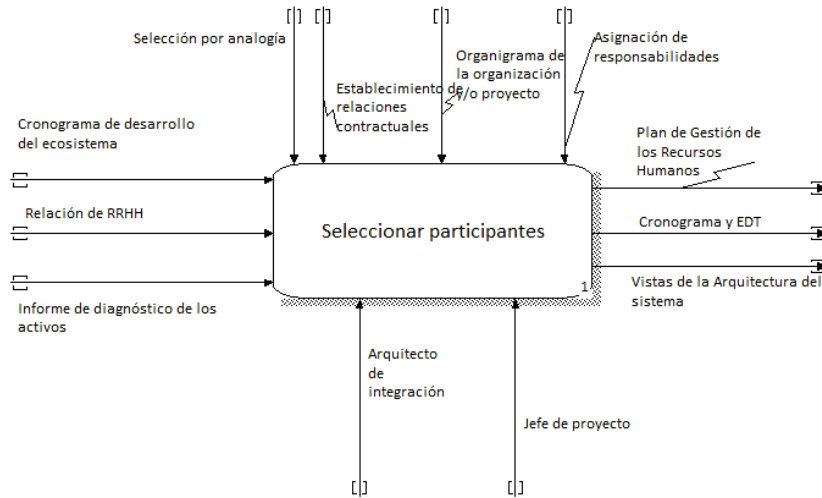
Anexo 4. Descripción del Proceso 1 en Notación IDEF0.



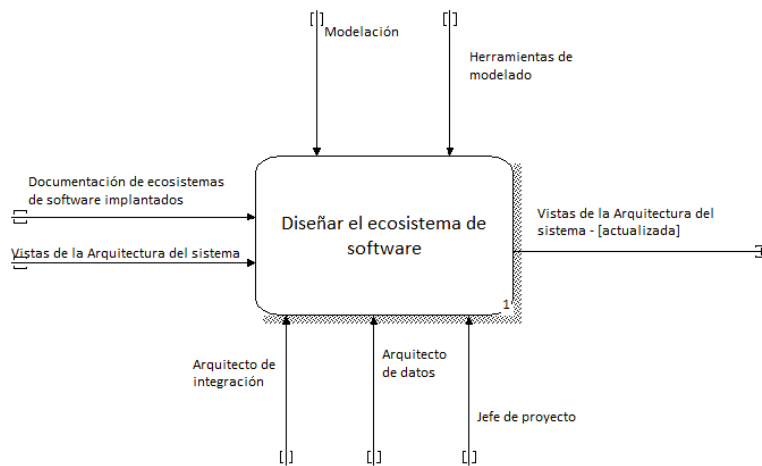
Anexo 5. Descripción del Proceso 2 en Notación IDEF0.



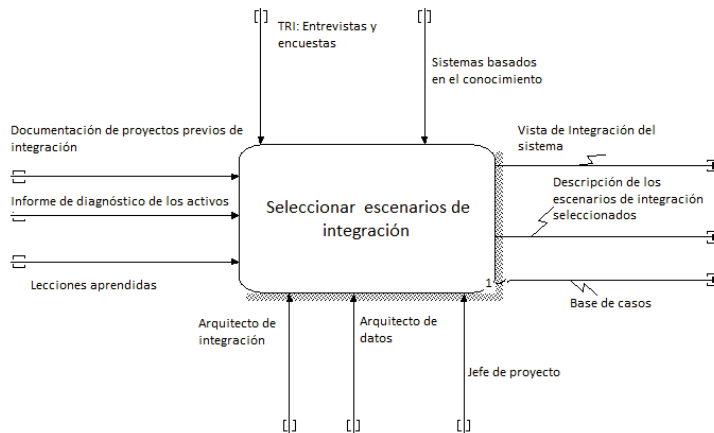
Anexo 6. Descripción del Proceso 3 en Notación IDEF0.



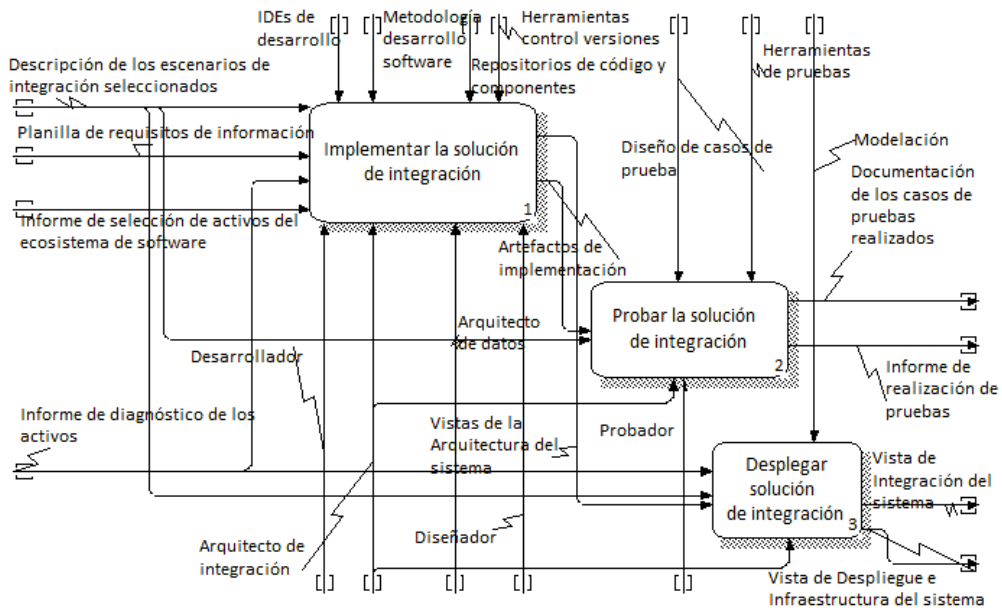
Anexo 7. Descripción del Proceso 4 en Notación IDEF0.



Anexo 8. Descripción del Proceso 5 en Notación IDEF0.



Anexo 9. Descripción del Proceso 6 en Notación IDEF0.



Anexo 10. Descripción del Proceso 7 en Notación IDEF0.

