APLICACIÓN DE LA METODOLOGIA OPENUP EN EL DESARROLLO DEL SISTEMA DE DIFUSIÓN DE GESTIÓN DEL CONOCIMIENTO DE LA ESPE

Santiago Ríos Salgado[1], Ing. Cecilia Hinojosa Raza[2], Ing. Ramiro Delgado Rodríguez[3]

1 Escuela Politécnica del Ejército, Ecuador, santiago.rios9@gmail.com

2 Escuela Politécnica del Ejército, Ecuador, cmhinojosa@espe.edu.ec

3 Escuela Politécnica del Ejército, Ecuador, rndelgado@espe.edu.ec

RESUMEN

En el presente artículo se muestra cómo la metodología OpenUP favoreció el desarrollo del sistema informático, considerado estratégico en el proceso de acreditación nacional e internacional de la Escuela Politécnica del Ejército, con característica de orientación a la web, llamado "Sistema de Difusión de Gestión del Conocimiento de la ESPE". Se consideró OpenUp porque utiliza un punto de vista pragmático, una filosofía ágil enfocada en la naturaleza colaborativa del proceso de desarrollo de software, y adicionalmente ofrece un alto grado de adaptabilidad a las necesidades de un proyecto particular por su carácter iterativo.

Palabras Clave: OpenUP, Metodología de desarrollo de Software, Ingeniería de Software, Difusión de la gestión del conocimiento.

ABSTRACT

This article explains how the methdology OpenUP was applied in the development of the web application "Sistema de Difusión de Gestión del Conocimiento de la ESPE", which is considered strategic in the process of national and international accreditation of the Escuela Politécnica del Ejército. OpenUP was considered because it uses a pragmatic, agile philosophy approach that focuses on the collaborative nature of the software development process, in additional it offers a high degree of adaptability to the needs of a particular project because of its iterative kind.

KeyWords: OpenUP, Software development methodology, Software engineering, Dissemination of knowledge management.

1. INTRODUCCIÓN

OpenUP es una metodología de desarrollo de software, basada en RUP (Rational Unified Process), que contiene el conjunto mínimo de prácticas que ayudan a un equipo de desarrollo de software a realizar un producto de alta calidad, de una forma eficiente. Esta metodología fue propuesta por el grupo de empresas conformado por: IBM Corp, Telelogic AB, Armstrong Process Group Inc., Number Six Software Inc. y Xansa; quienes la donaron a la Fundación Eclipse en el año 2007, que la ha publicado bajo licencia libre.

OpenUP, es un proceso unificado, iterativo e incremental, que se centra en el desarrollo colaborativo de software para generar sistemas de calidad.

La Escuela Politécnica del Ejército (ESPE), organización para la cual se desarrolla la aplicación, es una prestigiosa institución de educación superior del Ecuador, que ha enfocado su quehacer en tres ejes fundamentales, establecidos en su plan estratégico institucional, que son: académico, investigación y vinculación con la colectividad. Dentro de estos tres ámbitos la ESPE ha realizado aportes en beneficio de la sociedad ecuatoria-

na, que hoy en día requieren ser difundidos no solo por alcanzar la acreditación nacional e internacional, sino porque es importante dar a conocer los resultados obtenidos en ese ámbito, razones por las que el Sistema de Difusión de Gestión del Conocimiento de la ESPE es de relevancia.

La Escuela Politécnica del Ejército cuenta con el Departamento de Ciencias de la Computación, el cual aporta a la Institución con numerosas aplicaciones de software que favorecen su quehacer, desarrolladas a través de proyectos de graduación cuyas especificaciones son definidas por: los Docentes del departamento (normalmente los mentalizadores de los proyectos), futuros usuarios de las aplicaciones, los beneficiarios identificados y los desarrolladores.

2. METODOLOGÍA DE DESARROLLO DE SOFTWARE

2.1 OpenUP

OpenUP es un Proceso Unificado que aplica enfoques iterativos e incrementales dentro de un ciclo de vida estructurado, utiliza una filosofía ágil que se enfoca en la naturaleza de colaboración en el desarrollo de software. Es una herramienta agnóstica que puede extenderse para hacer frente a una amplia variedad de proyectos. [1] Está basado en casos de uso, la gestión del riesgo, y una arquitectura centrada a impulsar el desarrollo. [2]

2.1.1 Principios

Los principios básicos en los que se fundamenta OpenUP se muestran en la figura 1:

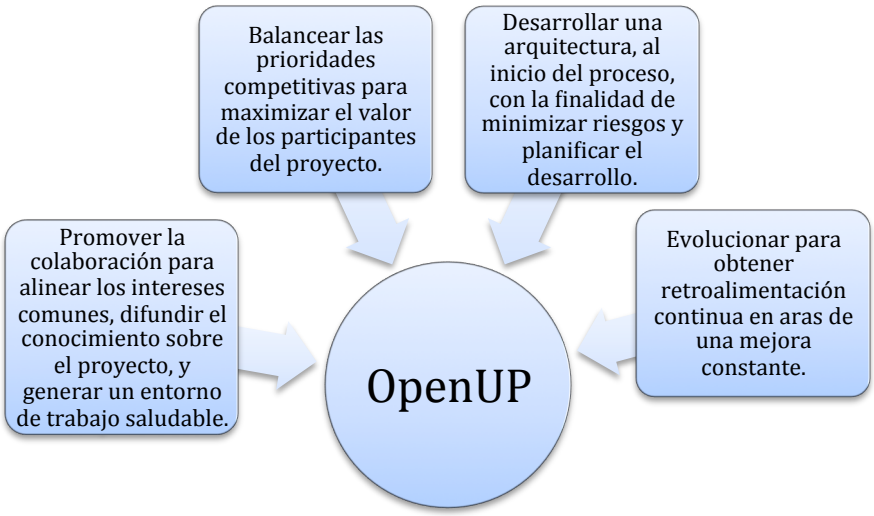

Fig. 1. Principios de OpenUP [3]

2.1.2 Elementos

OpenUP se organiza en dos dimensiones: Contenido metodológico y contenido procedimental. El contenido metodológico es el que define elementos metodológicos tales como disciplinas, tareas, artefactos y procesos, independientemente de como se usen estos o se combinen. El contenido procedimental, por el contrario, es donde se aplican todos estos elementos metodológicos dentro de una dimensión temporal, pudiéndose crear multitud de ciclos de vida diferentes a partir del mismo subconjunto de elementos metodológicos.

Los elementos que forman OpenUP se presentan en la figura 2:

Fig. 2. Elementos de OpenUP [4]

2.1.3 Ciclo de vida

El ciclo de vida de un proyecto, según la metodología OpenUP, permite que los integrantes del equipo de desarrollo aporten con micro-incrementos, que pueden ser el resultado del trabajo de unas pocas horas o unos pocos días. El progreso se puede visualizar diariamente, ya que la aplicación va evolucionando en función de estos micro-incrementos.

El objetivo de OpenUP es ayudar al equipo de desarrollo, a lo largo de todo el ciclo de vida de las iteraciones, para que sea capaz de añadir valor de negocio a los clientes, de una forma predecible, con la entrega de un software operativo y funcional al final de cada iteración. El ciclo de vida del proyecto provee a los clientes de: una visión del proyecto, transparencia y los medios para que controlen la financiación, el riesgo, el ámbito, el valor de retorno esperado, etc.

Todo proyecto en OpenUP consta de cuatro fases: inicio, elaboración, construcción y transición. Cada una de estas fases se divide a su vez en iteraciones. En la figura 3 se muestran estas fases y su relación:

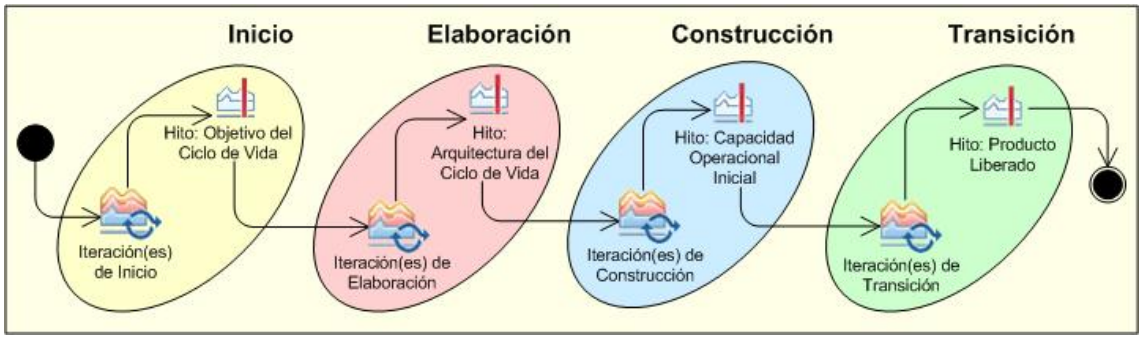

Fig. 3. Ciclo de vida de OpenUP [1]

- **Fase de inicio:** En esta fase, las necesidades de cada participante del proyecto son tomadas en cuenta y plasmadas en objetivos del proyecto. Se definen para el proyecto: el ámbito, los límites, el criterio de aceptación, los casos de uso críticos, una estimación inicial del coste y un boceto de la planificación.
- **Fase de elaboración:** En esta fase se realizan tareas de análisis del dominio y definición de la arquitectura del sistema. Se debe elaborar un plan de proyecto, estableciendo unos requisitos y una arquitectura estables. Por otro lado, el proceso de desarrollo, las herramientas, la infraestructura a utilizar y el entorno de desarrollo también se especifican en detalle en esta fase. Al final de la fase se debe tener una definición clara y precisa de los casos de uso, los actores, la arquitectura del sistema y un prototipo ejecutable de la misma.
- **Fase de construcción:** todos los componentes y funcionalidades del sistema que falten por implementar son realizados, probados e integrados en esta fase. Los resultados obtenidos en forma de incrementos ejecutables deben ser desarrollados de la forma más rápida posible sin dejar de lado la calidad de lo desarrollado.
- **Fase de transición:** Esta fase corresponde a la introducción del producto en la comunidad de usuarios, cuando el producto está lo suficientemente maduro. La fase de la transición consta de las subfases de pruebas de versiones beta, pilotaje y capacitación de los usuarios finales y de los encargados del mantenimiento del sistema. En función de la respuesta obtenida por los usuarios puede ser necesario realizar cambios en las entregas finales o implementar alguna funcionalidad más. [5]

2.1.4 Prácticas

OpenUP es una metodología basada en RUP, y por lo tanto, comparte las mismas prácticas que subyacen por debajo del flujo de trabajo y los roles de OpenUP. Estas prácticas se exponen en la figura 4:

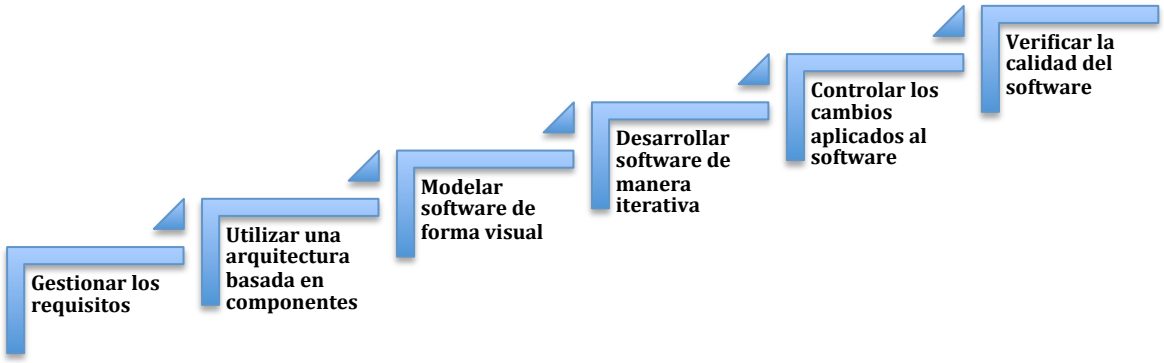

Fig. 4. Prácticas de OpenUP [4]

2.1.5 Entregables

Los entregables que se crearon como resultados de la aplicación de la metodología OpenUP se detallan en la tabla I:

Tabla I: Entregables del proyecto

Fase	Actividad	Documento
Inicio	Iniciación del proyecto	Visión del sistema
	Planeación del proyecto	Plan de desarrollo de software
	Identificación de requerimientos	Especificación de requisitos de software
Elaboración	Desarrollo de la arquitectura	Documento de arquitectura del sistema
Construcción	Definición de pruebas de la solución	Plan de pruebas de software
	Construcción de la solución: • Traducción de pruebas a código • Desarrollo del sistema	 • Creación de archivos con los escenarios de prueba. • Codificación de la solución basándose en el comportamiento definido en las pruebas.
Transición	Despliegue de la solución	Manual de instalación

3. HERRAMIENTAS

Las herramientas utilizadas para el desarrollo del proyecto expuesto se describen en la tabla II:

Tabla II: Herramientas

Fase	Herramientas					
	Descripción	Herramienta utilizada	Definición	Herramienta utilizada	Nombre	Definición
Inicio	Para ayudar a visualizar, mantener y simplificar el manejo de proyectos	Omniplan	Software para planificación y gestión de proyecto, está diseñado para ayudar a visualizar, mantener y simplificar el manejo de proyectos.	Para mantener informados a clientes, equipo y demás involucrados respecto al desarrollo del proyecto.	Teamworkpm .net	Software de administración de proyectos basado en web que permite llevar el control de los proyectos, permite mantener al tanto a clientes, equipo y demás involucrados del proyecto, al tanto de lo que sucede.
Elaboración	Para crear diagramas, diagramas de flujo, cuadros organizacionales e ilustraciones	Omnigraffle	Aplicación para crear diagramas, diagramas de flujo, cuadros organizacionales e ilustraciones.			

Construcción	Para desarrollo de código	Ruby on Rails	Framework de aplicaciones web de código abierto que permite desarrollos iterativos.
	Para descripción de cómo debe comportarse el software.	Cucumber	Plug-in de Rails que permite describir en texto plano cómo debe comportarse el software.
	Para seguir y controlar los cambios realizados en los archivos de un proyecto	Sistema de control de versiones (Git)	Combinación de tecnologías y prácticas para seguir y controlar los cambios realizados en los archivos de un proyecto.
	Para rápida adición y refactorización automática	Rubymine	Editor inteligente de Ruby con fragmentos de rápida adición y refactorización automática, provee diagramas de modelos, vistas generales del proyecto y otras vistas especializadas
Transición	Suite ofimática	Microsoft Office	Suite informática de oficina que abarca e interrelaciona aplicaciones de escritorio

4. DESARROLLO DEL APLICATIVO

En primera instancia se cumplieron reuniones de trabajo entre los mentalizadores del proyecto y el desarrollador, en las que tomando en cuenta la visión de la aplicación, la metodología, y el entorno en el que se desenvolvería el proyecto, se seleccionó la tecnología y técnica a utilizar en el desarrollo, decidiéndose lo siguiente:

- La tecnología debía ser Ruby on Rails por permitir un desarrollo rápido, eficaz e iterativo.
- La técnica Behavior-Driven Development en virtud que permite asegurar la calidad del software, definir el comportamiento deseado y paralelamente elaborar la documentación.

Posteriormente, se cumplieron cada una de las etapas definidas en la figura 3, de la siguiente manera:

Fase: Inicio

1. Se identificaron a todos los involucrados del proyecto:
    - Mentalizadores: director y co-director del proyecto
    - Usuarios: docentes e investigadores de los distintos departamentos.
    - Beneficiarios: comunidad politécnica y sociedad en general.
    - Desarrollador: estudiante egresado de la carrera de Ingeniería en Sistemas e Informática.
2. Se realizaron reuniones de trabajo entre los mentalizadores y el desarrollador mediante las cuales se definió la visión que tendría el Sistema de Difusión de Gestión del Conocimiento enmarcado en la filosofía y funcionamiento de la ESPE, el mismo que debería ser orientado a la web para un fácil acceso por parte de los beneficiarios, y diseñado para que se adapte a la cultura organizacional de la ESPE.
3. Por medio de reuniones, en las que participaron los mentalizadores, algunos de los futuros usuarios y el desarrollador, se establecieron los siguientes requerimientos funcionales:
    a. Se requiere de un administrador, cuyas responsabilidades incluyen:
        i. Administración de departamentos
        ii. Administración de áreas de conocimiento

iii. Administración de líneas de investigación
iv. Administración de sub-líneas de investigación.

b. Se requiere que un usuario tenga la capacidad de:
   i. Administrar programas de investigación.
   ii. Administrar proyectos de investigación
   iii. Administrar resultados de investigación
   iv. Administrar otras actividades relacionadas a la gestión del conocimiento.
   v. Administrar grupos de investigación.

c. El sistema debe ajustarse a las siguientes restricciones:
   i. Se requiere seguir el esquema operativo institucional, de la gestión del conocimiento, el mismo que contempla:
      1. Líneas de investigación, como ejes transversales a los distintos departamentos.
      2. Áreas de conocimiento, que forman parte de los departamentos.
      3. Docentes que pertenecen a los departamentos.
      4. Programas de investigación y vinculación con la sociedad; que están constituidos de proyectos, que son ejecutados por grupos de investigación conformados por docentes y estudiantes de los diferentes departamentos.
   ii. Se requiere que el sistema de difusión acepte solamente material proveniente de personas identificadas como miembros de la ESPE.
   iii. El contenido a ser difundido a través del sistema estará conformado de texto, imágenes, documentos.

Es importante mencionar, que estos requerimientos fueron sujetos de constantes cambios en la etapa inicial, por cuanto las necesidades de los usuarios variaban según el enfoque propio del departamento al que pertenecían.

Fase: Elaboración

4. Tomando en cuenta los requisitos funcionales se generó la arquitectura, sobre la que se soporta el Sistema de Difusión de Gestión del Conocimiento de la ESPE, la misma que se describe por medio de:

   a. Diagramas de casos de uso

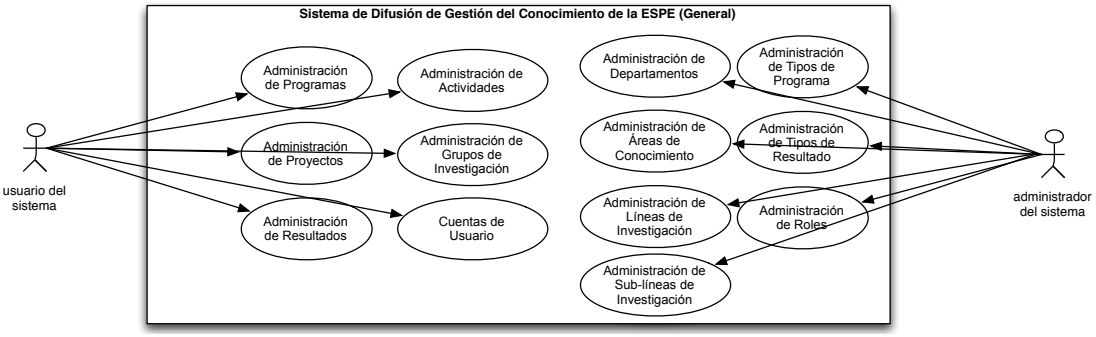

Fig. 5. Diagrama de casos de uso del Sistema de Difusión de Gestión del Conocimiento de la ESPE

   b. Diagramas de sub-sistemas

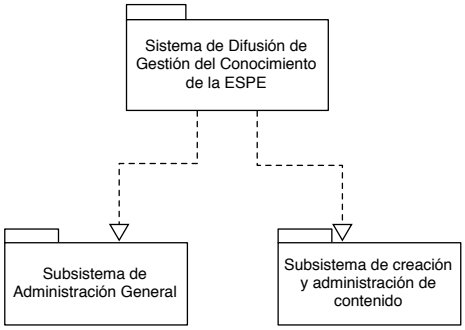

Fig. 6. Diagrama de subsistemas del Sistema de Difusión de Gestión del Conocimiento de la ESPE

c. Diagramas de paquetes

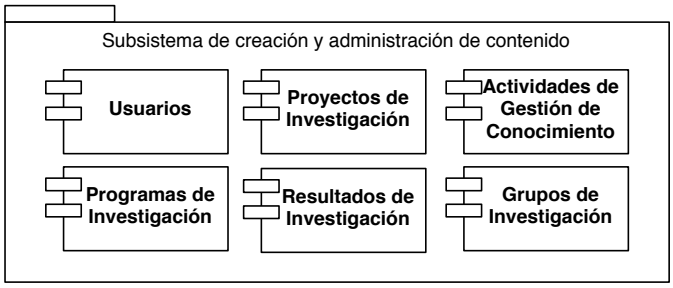

Fig. 7. Diagrama de paquetes del subsistema de creación y administración de contenido del Sistema de Difusión de Gestión del Conocimiento de la ESPE

d. Diagramas de clases

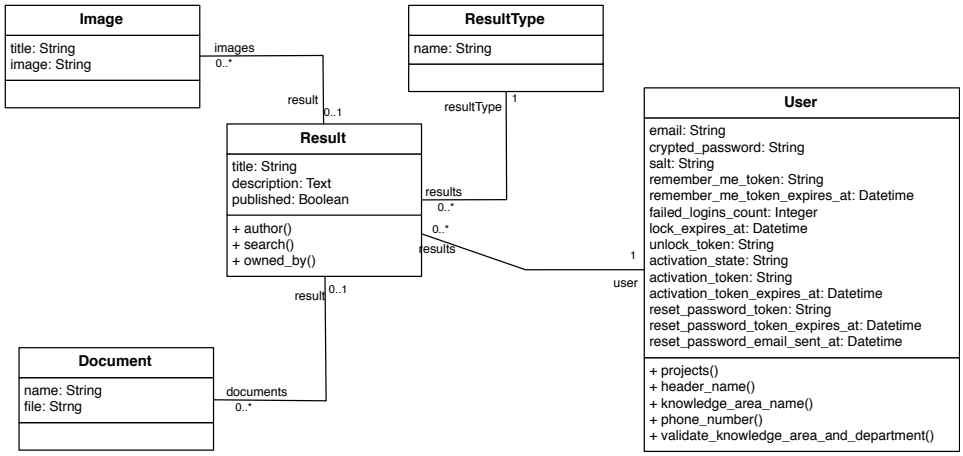

Fig. 8. Diagrama de clases del módulo de resultados

Fase: Construcción

5. Considerando los requisitos funcionales, se establecen los comportamientos deseados del Sistema de Difusión de Gestión del Conocimiento de la ESPE, los cuales fueron traducidos en especificaciones técnicas utilizando un lenguaje ubicuo proporcionado por la herramienta Cucumber que permitió aplicar la técnica Behavior-Driven Development en la construcción de la solución.
6. Utilizando como base la arquitectura definida en la etapa de elaboración, se procedió a la construcción del sistema con las características necesarias para satisfacer los comportamientos definidos en el documento de pruebas de software [6], y a su vez las restricciones de rendimiento definidas en el documento de especificación de requisitos de software [6].

Fase: Transición

7. Finalmente, para realizar el despliegue de la solución, se documentó en el Manual de instalación, los pasos necesarios para la instalación de la misma en un ambiente Linux.

5. RESULTADOS DE LAS PRUEBAS AL SISTEMA

La evaluación de los requerimientos funcionales definidos en el documento de especificación de requisitos de software:

Tabla III: Pruebas funcionales

ID	NOMBRRE	DESCRIPCIÓN	RESULTADO
1	Administrar Programas	Comportamiento relacionado a la creación, actualización, visualización y eliminación de programas de investigación como usuario firmado en el sistema	Cumplido
2	Acciones de Usuarios Anónimos Relacionados a Programas	Definición de rutas de éxito y fracaso para la visualización, creación, actualización y eliminación de programas de investigación como usuario NO firmado en el sistema	Cumplido

3	Administrar Proyectos	Comportamiento relacionado a la creación, actualización, visualización y eliminación de proyectos de investigación como usuario firmado en el sistema	Cumplido
4	Acciones de Usuarios Anónimos Relacionados a Proyectos	Definición de rutas de éxito y fracaso para la visualización, creación, actualización y eliminación de proyectos de investigación como usuario NO firmado en el sistema	Cumplido
5	Administrar Resultados	Comportamiento relacionado a la creación, actualización, visualización y eliminación de resultados de investigación como usuario firmado en el sistema	Cumplido
6	Acciones de Usuarios Anónimos Relacionados a Resultados	Definición de rutas de éxito y fracaso para la visualización, creación, actualización y eliminación de resultados de investigación como usuario NO firmado en el sistema	Cumplido
7	Administrar Actividades	Comportamiento relacionado a la creación, actualización, visualización y eliminación de actividades de investigación como usuario firmado en el sistema	Cumplido
8	Acciones de Usuarios Anónimos Relacionados a Actividades	Definición de rutas de éxito y fracaso para la visualización, creación, actualización y eliminación de actividades de investigación como usuario NO firmado en el sistema	Cumplido

Pruebas de rendimiento al aplicativo:

Tiempo de carga en el browser:

Es el tiempo promedio de carga de página divido en los siguientes segmentos de tiempo:

- **Cola de pedidos:** El tiempo de espera entre el servidor web y el código de aplicación. Números grandes indican un servidor de aplicaciones de ocupado.
- **Aplicación web:** El tiempo pasado en el código de aplicación.
- **Red:** La latencia de la red, o el tiempo que tarda una petición a través del Internet.
- **Procesamiento del DOM:** En el navegador, analizar e interpretar el HTML. Medido por el navegador "DOMContent" del evento.
- **Renderizado de la página:** En el navegador, mostrando el código HTML, Javascript que se ejecuta en línea y carga imágenes. Medido por el navegador "Load" del evento.

Resultados que se describen en la figura 9, los mismos que cumplen las especificaciones de rendimiento, que estipulan que la aplicación debe tener tiempos de carga inferiores a los 5:

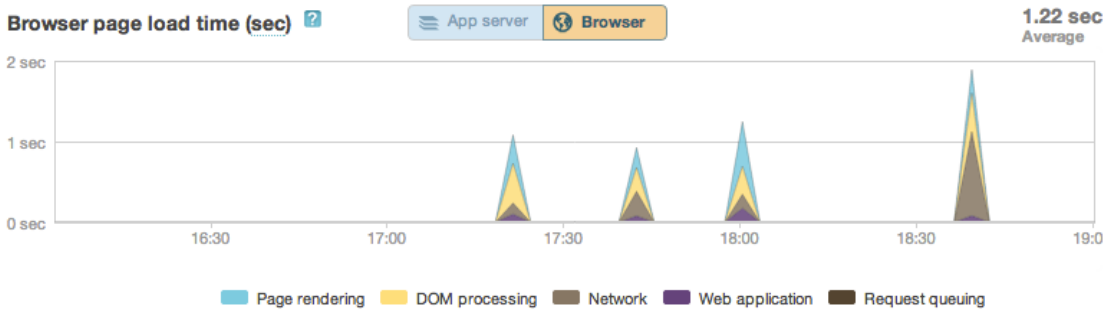

Fig. 9. Tiempo de carga del browser

Tiempo de respuesta del servidor

El tiempo de respuesta del servidor depende de varios factores, como son la base de datos, la programación en sí (en este gráfico se lo refiere como Ruby), y finalmente el encolamiento de las peticiones. Estos datos permiten valorar la calidad del software que se ha desarrollado como parte del Sistema de Difusión de Gestión del Conocimiento de la ESPE. El tiempo de respuesta del servidor debe ser menor a 1 segundo para satisfacer las especificaciones de rendimiento señaladas para la aplicación, como se expone en la figura 10:

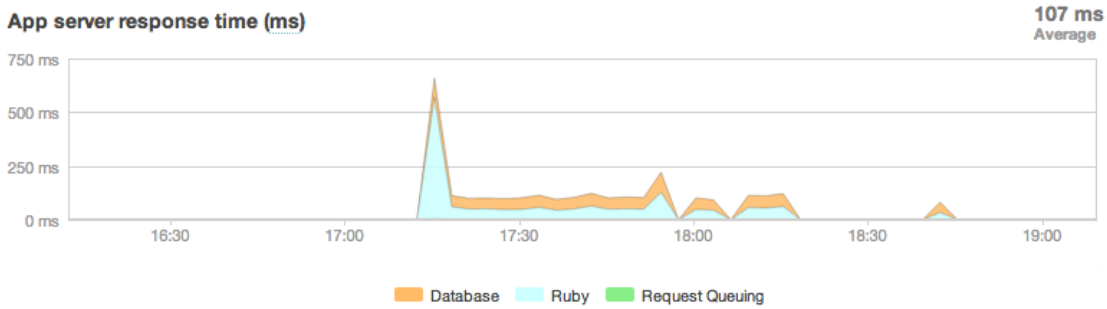

Fig. 10. Tiempo de respuesta del servidor

Mapa de tiempo

Permite definir el tiempo promedio que toma una página en cargarse dentro del navegador de un usuario. Este tiempo está compuesto por el tiempo de respuesta de la base de datos (database), el tiempo de ejecución de la programación del sistema para desplegar una página en especial (tesis), y finalmente el tiempo que se demora el navegador en procesar la respuesta entregada por el servidor (end user). La suma de los tiempos debe ser menor a 5 segundos para cumplir las especificaciones de rendimiento estipuladas, tal como se muestra en la figura 11:

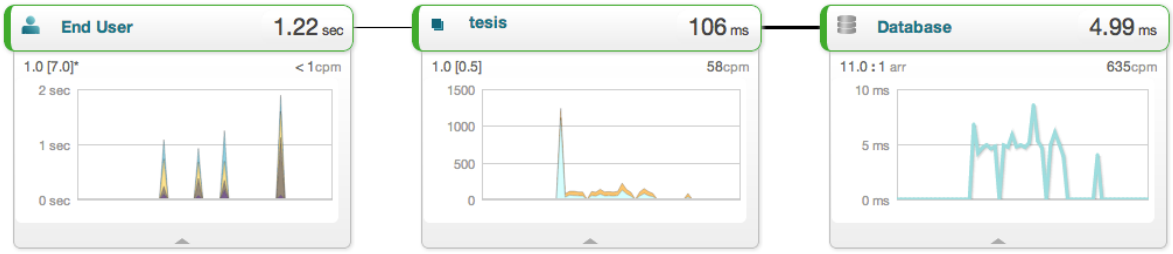

Fig. 11. Mapa de tiempo

6. TRABAJOS RELACIONADOS

Para la Escuela Politécnica del Ejército (ESPE), y en particular para el Departamento de Ciencias de la Computación, el proyecto descrito en este artículo, se constituye en el primero de interés institucional, que se fundamenta en la metodología OpenUP. Cabe mencionar, que existen desarrollos de sistemas en los que se han probado metodologías ágiles, entre los cuales se pueden mencionar los siguientes ejemplos:

- Metodología Agile Unified Process – Sistema web de asignación de aulas de los laboratorios generales de computación de la ESPE [7]
- Proceso Unificado de Desarrollo - Aplicación web para el control de usuarios y recursos informáticos de los laboratorios de computación de la Escuela Politécnica del Ejército [8]
- Metodología UML-Based Web Engineering – Sistema E-Commerce para la gestión de ventas [9]
- Microsoft Solution Framework - Sistema con tecnología web para la gestión de información del departamento de calidad, seguridad, medio ambiente y salud [10]

Por otro lado, en la ESPE, no ha existido otro desarrollo de un sistema que se oriente a la Difusión de Gestión del Conocimiento, y lo que se ha pretendido es, utilizar plataformas de software libre, como por ejemplo micro-sitios web, que las UTICs - ESPE procuran que sean utilizados por la generalidad de la comunidad politécnica.

7. CONCLUSIONES

El uso de una metodología ágil como OpenUP favoreció el desarrollo del Sistema de Difusión de Gestión del Conocimiento de la ESPE por las siguientes razones:

- Permitió que el sistema sea fácilmente adaptable a los requerimientos cambiantes que se dieron du-

rante el desarrollo del proyecto.
- Proporcionó un marco de referencia para dividir el proyecto en etapas.
- Contribuyó en la generación de la documentación del proyecto.
- Orientó el desarrollo del sistema únicamente hacia tareas y procesos que agregaron valor al producto final.
- Permitió realizar una planificación real.
- Facilitó la comunicación dentro del equipo de trabajo.

Los resultados obtenidos de las pruebas demuestran fehacientemente que la metodología OpenUP contribuyó eficientemente al cumplimiento de todas las especificaciones funcionales y de rendimiento establecidas para el Sistema.

Finalmente, por la experiencia adquirida, se puede afirmar que OpenUP, permitió que el Sistema de Difusión de Gestión del Conocimiento de la ESPE tenga características que todo buen software debe poseer:

- Eficiencia: evidenciada por los resultados obtenidos en la medición del desempeño del software, que demuestran un buen uso de los recursos que se manipulan.
- Flexibilidad: por su facilidad de configuración, parametrización, adaptación y escalamiento a los cambios que puedan darse en relación a la gestión del conocimiento dentro de la Escuela Politécnica del Ejército.
- Portabilidad: ya que el aplicativo puede ser fácilmente migrado a diferentes plataformas de hardware o software.
- Integridad: debido a que el sistema es capaz de proteger sus diferentes componentes contra los procesos o elementos que no tengan derecho de acceso a los mismos.

8. REFERENCIAS BIBLIOGRÁFICAS

[1] Néstor Díaz & David Vaquero. (2010, Febrero) Morfeo Formación. [Online]. http://formacion.morfeo-project.org/wiki/index.php/PT3:GPSL:UF6

[2] Per Kroll & Bruce MacIsaac, *Agility and Discipline Made Easy: Practices from OpenUP and RUP*, 1st ed. USA: Addison-Wesley Professional, 2006.

[3] Mario Insunza & Miguel Villanueva. (2011, Septiembre) The Eclipse Foundation. [Online]. http://epf.eclipse.org/wikis/openupsp/openup_basic/guidances/roadmaps/openup_basic_roadmap,_vEruwN-rEdqiM_wFaqLjNg.html

[4] The Eclipse Foundation. (2012, Junio) The Eclipse Foundation. [Online]. http://epf.eclipse.org/wikis/openup/

[5] Loraine Gimson, Gustavo Gil, & Gustavo Rossi. (2012, Junio) Universidad Nacional de la Plata - Argentina. [Online]. http://sedici.unlp.edu.ar/bitstream/handle/10915/24942/Documento_completo__.pdf?sequence=1

[6] Carlos Aucancela, Tatiana Pozo, & Cecilia Hinojosa. (2011, Octubre) Escuela Politécnica del Ejército. [Online]. http://repositorio.espe.edu.ec/bitstream/21000/4764/1/T-ESPE-032870.pdf

[7] Juan Ninahualpa, Rolando Reyes, and Lourdes De La Cruz. (2008, Marzo) Escuela Politécnica del Ejército. [Online]. http://repositorio.espe.edu.ec/bitstream/21000/557/1/T-ESPE-021843.pdf

[8] Jenny Ruíz, Edwin Romero, & Carla Villegas. (2010, Octubre) Escuela Politécnica del Ejército. [Online]. http://repositorio.espe.edu.ec/bitstream/21000/338/1/T-ESPE-029757.pdf

[9] Carlos Chamorro & Mauricio Loachamín. (2009, Abril) Escuela Politécnica del Ejército. [Online]. http://repositorio.espe.edu.ec/bitstream/21000/1064/1/T-ESPE-021954.pdf

[10] Dean Leffingwell, *Agile Software Requirements: Lean Requirements Practices for Teams, Programs, and the Enterprise*. Boston, USA: Addison-Wesley Professional, 2011.

[11] Jeff Langr & Tim Ottinger, *Agile in a Flash: Speed-Learning Agile Software Development*. Canadá: Pragmatic Bookshelf, 2011.

[12] Usman Fazal & Sardar Raham, *Quality Assurance Techniques in OpenUP: Agile processes such as Scrum, XP, OpenUP etc. have techniques that improve software quality*. USA: LAP LAMBERT Academic Publishing, 2011.

[13] Santiago Ríos, Cecilia Hinojosa, & Ramiro Delgado, *Documento del proyecto de graduación: Desarrollo de un Sistema de Difusión de Gestión del Conocimiento de la ESPE, aplicando la Metodología OpenUP y el Framework Ruby on Rails*. Quito, Ecuador: Escuela Politécnica del Ejército, 2013.