

# ESTRATEGIA PARA LA REALIZACIÓN DE PROYECTOS DE DESARROLLO DE SOFTWARE EN SISTEMAS DE INFORMACIÓN GLOBAL

Manuel Mejías, M<sup>a</sup> José Escalona<sup>p</sup>, Isabel Ramos, Jesús Torres, Juan A. Ortega

Departamento de Lenguajes y Sistemas Informáticos

Universidad de Sevilla

{risoto, escalona, isabel.ramos, jtorres, ortega}@lsi.us.es

## RESUMEN

La realización de proyectos de ingeniería informática constituye actualmente un sector en auge dentro del ámbito industrial, con grandes repercusiones en el mercado laboral. El desarrollo de las telecomunicaciones, internet y la necesidad de tratar información multimedia, contribuye a aumentar la importancia que la ingeniería de proyectos de desarrollo de software está alcanzando como medio de garantía de la calidad de los productos informáticos construidos.

Los sistemas de información global, entendidos conceptualmente como sistemas software capaces de manejar información multimedia de cualquier tipo y que pueden ser implantados siguiendo cualquier arquitectura actualmente imaginable, ya sea en entorno centralizado o regionalmente distribuido, o bien en internet, constituyen un dominio de interés creciente en la industria dedicada a la producción de software.

Dado este interés, en el presente trabajo se propone una estrategia para realizar los proyectos de desarrollo de software que se enmarcan en el dominio anteriormente indicado. Esta propuesta abarca desde la concepción inicial del proyecto hasta su puesta en explotación, pasando por el diseño y la construcción, y en ella se han considerado aquellos aspectos de la ingeniería de proyectos, en general, y de la ingeniería de proyectos de desarrollo de software, en particular, que contribuyen a lograr la realización de productos de alta calidad.

Como conclusión del presente trabajo y fruto también de la experiencia profesional de los autores se puede afirmar que los principios fundamentales que se contemplan a la hora de aplicar ingeniería para realizar proyectos, son aplicables igualmente cuando el producto a obtener son sistemas software, independientemente de las características específicas que estos tuvieran. Fruto de esta reflexión es la

propuesta para realizar proyectos de desarrollo de software que se presenta en este trabajo.

## **ABSTRACT**

Nowdays, software development projects make up an increasing industrial sector. This sector has repercussions on work market. The telecommunications development, internet and the necessity of managing different ways of information are circumstances to increase the importance of software development project engineering. This software engineering guarantees the quality of software products. Global information systems work with different types of information: text, voice, video, graphic, etc. This systems can use any architecture that we can think up. This architecture can be centralized, distributed or using internet. Software industry is very interested in this kind of systems.

This paper proposes an strategy to make software development projects to produce global information systems. This proposal goes from the initial conception of the project until the final product. The design phase and the construction phase are also considered in this work. The suggestions from the traditional project engineering and the software project engineering are had under consideration in this proposal in order to achieve software products with great quality.

To conclude this paper we can say that fundamental bases of traditional engineering can be applied to any type of product.

## **1- LA INGENIERÍA EN LA PRODUCCIÓN DE SOFTWARE**

Muchas y variadas son las definiciones que podemos encontrarnos del término ingeniería, pero todas ellas orbitan sobre la conjunción de una serie de disciplinas que permiten obtener como resultados unos productos que aporten algún tipo de beneficio a la sociedad en la que se enmarcan.

El *American Institute of Industrial Engineering* define la Ingeniería Industrial como la disciplina que se ocupa de la planificación, el mejoramiento y la instalación de sistemas integrados por hombres, materiales y equipos. Exige conocimientos especializados y una sólida formación en ciencias, matemáticas, físicas y sociales, junto con los principios y los métodos del análisis y diseño propios de la ingeniería, para especificar, predecir y evaluar los resultados que habrán de obtenerse de tales sistemas.

Observando esta definición y haciendo su traslación a otros ámbitos de producción, como por ejemplo al software, se puede concluir sin dificultad que tal definición sigue teniendo vigencia y que las diferencias entre ingenierías son de matices en lo concreto de su quehacer y no respecto a sus conceptos y actitudes. En este sentido, al hablar de ingeniería en la producción de software, el matiz diferenciador más significativo tal vez debamos ubicarlo en la tipología del producto obtenido, que aunque en este caso presenta una ausencia de características físicas, no excluye la necesidad de realizar actividades como la planificación, la especificación, el diseño, la construcción, la instalación y el mantenimiento, todas ellas típicas de cualquier actividad de ingeniería, independientemente del calificativo que le asociemos.

## **2- LOS SISTEMAS DE INFORMACIÓN GLOBAL**

En los últimos tiempos ha surgido el término genérico *sistema de información global* para hacer referencia a aquellos sistemas software que tienen la capacidad de gestionar información heterogénea (textual, vídeo, voz, gráfica, etc.), con grandes necesidades de almacenamiento de información, potentes requisitos funcionales y que pueden ser implantados siguiendo cualquier modelo arquitectónico actualmente existente, ya sea centralizado, regionalmente distribuido o internet.

Dadas las características de estos sistemas, que integran aspectos propios de los sistemas software clásicos, con aquellos otros conocidos como sistemas multimedia o bien con los sistemas webs, la producción de los mismos se ha realizado aprovechando propuestas metodológicas [1] [14] [11] [3] [8] que en estos otros ámbitos han ido apareciendo con el paso del tiempo, adaptando las mismas de forma adecuada.

Sin embargo, tras realizar un estudio comparativo de estas metodologías [6], se puede llegar a la conclusión de que ninguna de ellas cubre todos los aspectos considerados en los sistemas de información global, por lo que resulta necesario plantear una nueva propuesta metodológica que sirva de guía de referencia para la realización de proyectos de desarrollo de sistemas software de información global.

## **3- PLANTEAMIENTO GENERAL DE LA METODOLOGÍA**

La propuesta metodológica que en este trabajo se presenta (GDM, Global Information System Development Methodology) se sustenta en tres pilares básicos: a) es fiel al paradigma de la orientación a objetos para la producción de software; b)

considera fundamental el aspecto de la navegación de los usuarios del sistema a través del dominio de información asociado al mismo y lo desarrolla de forma independiente, y c) considera de forma separada los aspectos relacionados con la interfaz del sistema.

Entre otras características de GDM se puede decir que abarca todo el ciclo de vida del proyecto software, especifica los pasos a seguir para realizar el proyecto, las técnicas a aplicar en cada uno de ellos y los productos intermedios que se van obteniendo en los mismos.

Teniendo en cuenta estas características, el ciclo de vida planteado para GDM se divide en los cinco flujos de trabajo que se detallan en la tabla 1. En ella se muestra además el resultado previsto en cada flujo y el contenido que va aparecer en cada uno de ellos.

Flujo de trabajo	Productos resultantes
Especificación de requisitos	* Catálogo de requisitos del sistema que recoge: Los objetivos del sistema Los requisitos de almacenamiento de información La definición de actores Los requisitos funcionales Los requisitos de interacción Los requisitos no funcionales
Análisis	* Documento de análisis del sistema que recoge: El modelo de clases del sistema El modelo de navegación Los prototipos de interfaz

Diseño	<p>* Documento de diseño del sistema que recoge:</p> <ul style="list-style-type: none"> <li>La división del sistema en subsistemas y la arquitectura</li> <li>El modelo de clases de diseño con los casos de uso</li> <li>Modelo de clases navegacionales refinado con los aspectos de interfaz abstracta y los contextos navegacionales</li> <li>Modelo dinámico de la interfaz abstracta</li> </ul>
Implementación	<ul style="list-style-type: none"> <li>* Programa ejecutable</li> <li>* Manual de usuario</li> </ul>
Pruebas	<ul style="list-style-type: none"> <li>* Plan de pruebas</li> <li>* Resultado del plan de pruebas</li> </ul>

Tabla 1: Visión general de GDM

### 3.1- Especificación de requisitos

El primer flujo de trabajo de GDM es la especificación de requisitos (figura 1) [5]. En ella se realiza la definición de los objetivos del sistema y del entorno de trabajo (actividad 1), se identifican las necesidades de almacenamiento de información (actividad 2) y se describen los actores (roles que los usuarios pueden jugar) que interactúan con el sistema (actividad 3). Los actores detectados se agrupan en dos grandes grupos: a) actores básicos, que integra a aquellos actores que representan un rol simple en el sistema, y b) actores derivados, que son aquellos actores que pueden desempeñar varios de los roles representados por los actores básicos.

A continuación se realiza la definición de los requisitos funcionales (actividad 4), para lo que se utiliza la técnica de los casos de uso [10], así como de los requisitos de interacción (actividad 5) [4], que van a permitir indicar cómo interactúan los actores con el sistema, qué se le muestra en cada momento de su interacción, qué pueden hacer y cómo van a poder moverse por el dominio de información del sistema.

La definición de los requisitos de interacción se hace en base a dos aspectos. Por un lado se analizan y definen los criterios de recuperación de información, para lo cual se utiliza el lenguaje BNL [2], y por otro lado se deben definir los *prototipos de visualización*, que indican la forma en que se muestra al usuario la información

definida en los requisitos de almacenamiento y los actores que pueden visualizar esa información.

Por último, para completar la especificación de los requisitos se definirán todos aquellos otros requisitos del sistema que no han sido contemplado en los requisitos funcionales (actividad 6), como pueden ser los requisitos de seguridad, entre otros.

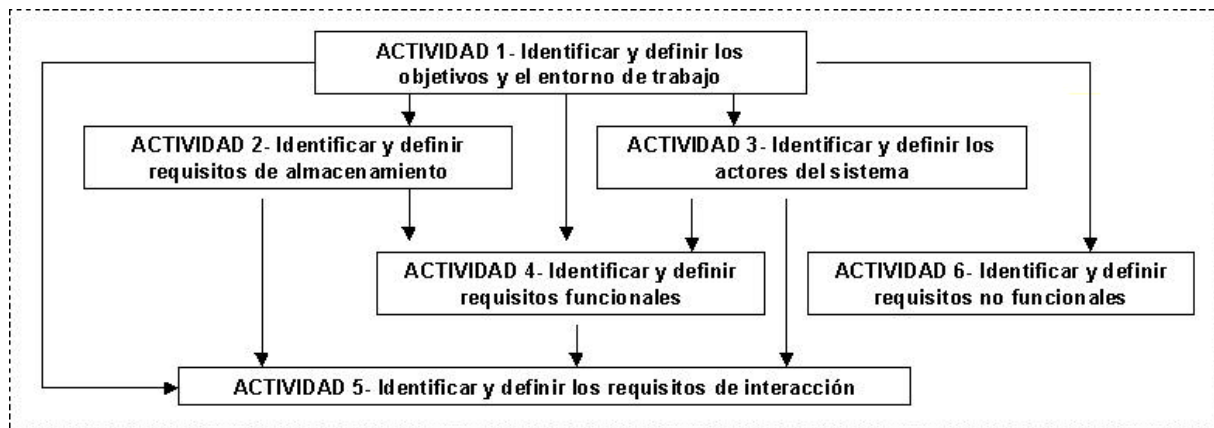


Figura 1: esquema de las actividades de especificación de requisitos

### 3.2- Análisis

Tomando como punto de partida la especificación de requisitos, el análisis (figura 2) realiza la especificación del sistema, tanto en sus aspectos básicos como de navegación e interfaz [7]. A partir de los requisitos de almacenamiento de información se obtiene el modelo conceptual de análisis (actividad 1) y a partir de los requisitos de interacción se definen nuevas estructuras navegacionales que permiten definir un modelo de navegación congruente que garantiza la calidad del sistema que se está desarrollando [12].

El análisis del sistema termina con la definición de la interfaz abstracta (actividad 3), la cual se obtiene a partir del sistema de navegación definido en la actividad anterior y de la definición de los prototipos de visualización realizados en la especificación.

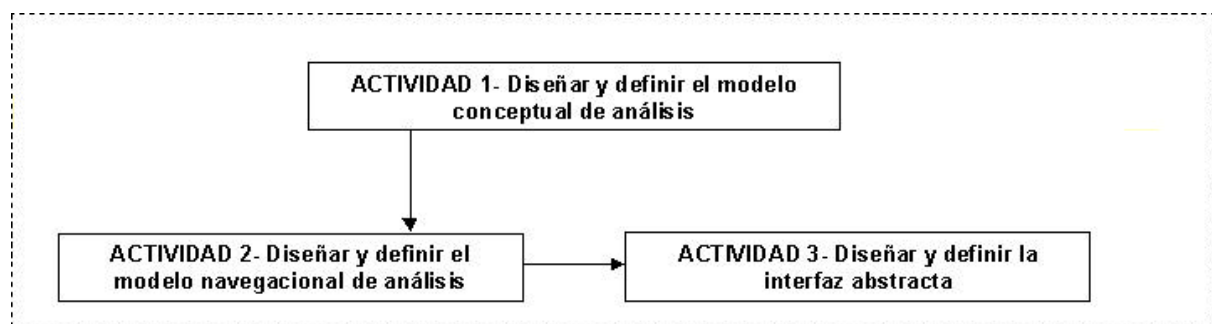


Figura 2: esquema del flujo de análisis

### 3.3- Diseño

El diseño (figura 3) toma como referencia los resultados de los dos flujos anteriores y comienza con la descomposición en subsistemas y la definición de la arquitectura del sistema (actividad 1). Para cada uno de los subsistemas se realizará el diseño de los casos de uso (actividad 2) a partir de la definición de los mismos hecha anteriormente y del modelo de clases de análisis. Tras diseñar los casos de uso se revisa el modelo conceptual de análisis (actividad 3), añadiéndole los métodos necesarios a las clases, nuevas relaciones entre ellas e identificando nuevas clases que se consideren necesarias.

A continuación, partiendo de los modelos obtenidos en las etapas de especificación y análisis se revisa el modelo navegacional para definir y diseñar los contextos navegacionales (actividad 4), que recogen los aspectos de navegación de cada actor en función de la situación de interacción con el sistema en que se encuentre.

Una vez diseñados los contextos navegacionales y considerando la especificación de la interfaz abstracta realizada con anterioridad, se define el modelo dinámico de ésta (actividad 5), que permite indicar cómo cambia la interfaz en función de los eventos que se produzcan, del actor que interactúe y de la situación de navegación en que se encuentre. Una vez hecho esto se redefine el modelo navegacional (actividad 6) para enriquecerlo con los aspectos considerados en las actividades anteriores.

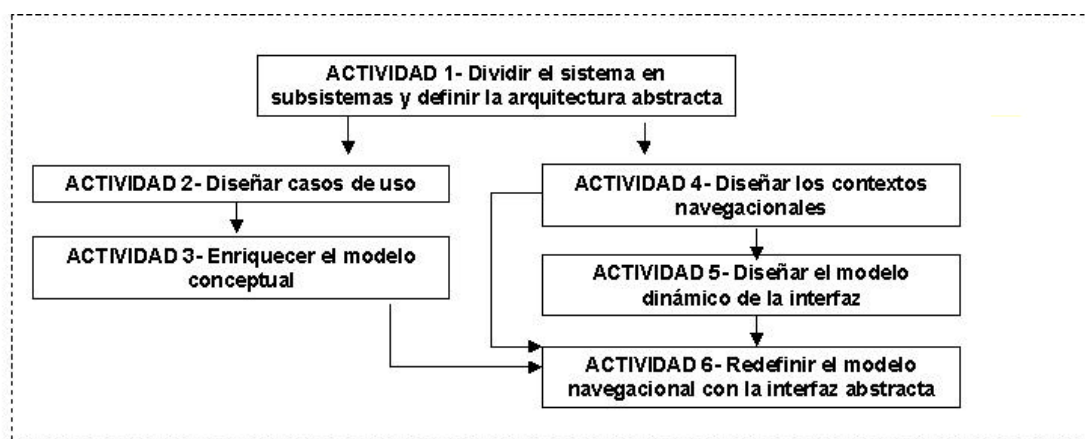


Figura 3: esquema del flujo de diseño

### 3.4- Implementación

El nivel de detalle que se alcanza en el flujo de diseño de GDM hace que la implementación (figura 4) sea relativamente sencilla. Así, en primer lugar se implementa la arquitectura propuesta (actividad 1), a continuación se realiza el

diseño de la base de datos (actividad 2), la implementación del modelo de navegación (actividad 3), la implementación de la interfaz (actividad 4), la implementación de los métodos del modelo básico del sistema (actividad 5) y por último, se debe realizar el manual de usuario del sistema (actividad 6).

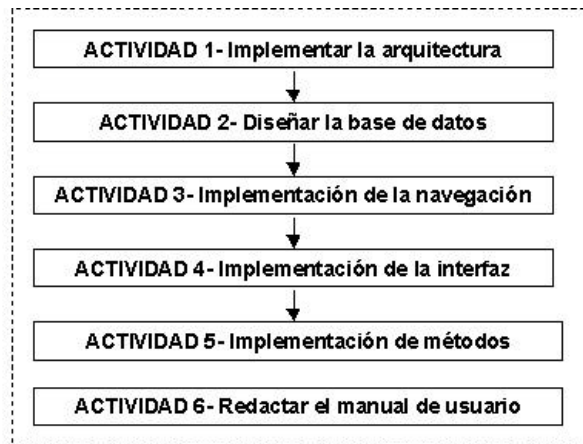


Figura 4: esquema del flujo de implementación

### 3.5- Pruebas

El último flujo de trabajo que engloba GDM es el de pruebas (figura 5). La batería de pruebas a realizar a un sistema depende mucho de la arquitectura y del entorno de desarrollo que se haya elegido. Sin embargo, a nivel abstracto GDM propone que las pruebas comiencen por comprobar la consistencia de la base de datos y la adecuación a los requisitos de almacenamiento (actividad 1). Tras esto, se deben realizar las pruebas sobre la funcionalidad del sistema (actividad 2), sobre el sistema navegacional (actividad 3), sobre el interfaz del mismo (actividad 4), para acabar el flujo de trabajo con la redacción del documento de pruebas (actividad 5).

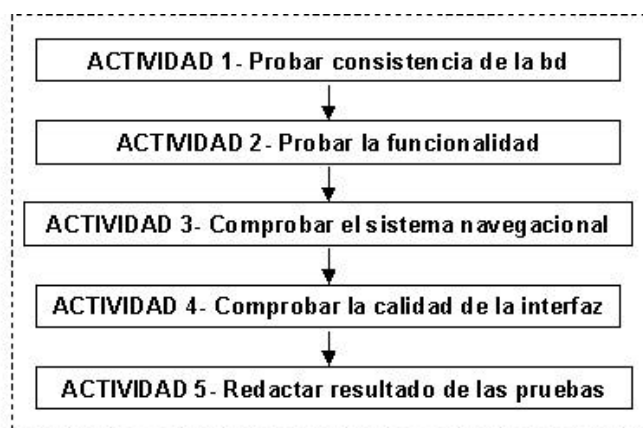


Figura 5: esquema del flujo de pruebas



#### **4- CONCLUSIONES**

En el presente trabajo se han presentado los aspectos globales de una propuesta metodológica que sirva de referencia para realizar proyectos de desarrollo de sistemas software de información global. Dada la estructura de dicha metodología, los conceptos en ella manejados, así como la propia experiencia de los autores, se puede llegar a la conclusión de que los principios básicos aquí utilizados son coincidentes con los que sustentan cualquier actividad de ingeniería, por muy clásica que la misma sea, teniendo siempre presente que el objetivo último de toda ingeniería es obtener un producto de la mayor calidad posible. En este contexto la ingeniería de software no es una excepción y coincide, tanto en sus motivaciones como en su planteamiento general y objetivos, con cualquier otra actividad de ingeniería que pudiéramos considerar, independientemente del adjetivo que le apliquemos.

#### **5- REFERENCIAS**

1. G. Booch, J. Rumbaugh, I. Jacobson. UML User Guide. Ed. Addison-Wesley, 1999.
2. N. Brisaboa, M.J. Durán, C. Lalín, J.R. López, J.R. Paramá, .R. Penabad, A.S. Places. Arquitectura para Bibliotecas Virtuales, 1999.
3. O.M.F. De Troyer, C.J. Leune. WSDM: A User Centered Design Method for Web Sites. Tilburg University, Infolab. 1997
4. M.J. Escalona, M.Mejías, J.Torres, A.Reina. Definición de requisitos de interacción. II Jornadas de trabajos Dolmen. Valencia, Marzo de 2002
5. M.J. Escalona, M.Mejías, J.Torres. Getting requirements in web information systems. XIV I. Conference "Software & Systems Engineering & their Applications". Paris, Diciembre 2001.
6. M.J. Escalona. Metodologías para el desarrollo de sistemas de información global: análisis comparativo y propuesta. Report interno LSI-2002-01. Departamento de Lenguajes y Sistemas Informáticos ([www.lsi.us.es/~informes/informes.htm](http://www.lsi.us.es/~informes/informes.htm)). Universidad de Sevilla, Enero de 2002.
7. M.J. Escalona, M. Mejías, J.Torres, A. Reina. NDT: una técnica para el desarrollo de la navegación. Congreso Ideas 2002. La Habana, Cuba. Abril de 2002

8. F. Garzotto, D. Schwabe, P. Paolini. HDM – A Model Based Approach to Hypermedia Application Design. ACM Transactions on Information Systems, Jan, 1993.
9. T. Isakowitz, E. Stohr, P. Balasubramanian. RMM: A Methodology for Structured Hypermedia Design. Communications of the ACM. Vol. 38, n. 8, August, 1995.
10. I. Jacobson. Modelling with use cases-Formalizing use-case modelling. Journal of Object-Oriented Programming, June 1995
11. N. Kock. Software Engineering for Adaptive Hypermedia Systems. PHD Thesis. Verlag UNI-DRUCK. Alemania 2001
12. L. Olsina. Ingeniería del software en la web. Metodología Cuantitativa para la Evaluación y Comparación de la Calidad en Sitios Web. PHD Thesis, Facultad de Ciencias Exactas de la UNLP. Argentina, Noviembre 1999.
13. A. Reina, J. Torres. Analysing the navigational aspect Second Workshop on Aspect-Oriented Software Development. Febrero 2002. Bonn, Alemania.
14. G. Rossi. An Object Oriented Method for Designing Hypermedia Applications. PHD Thesis, Departamento de Informática, PUC-Rio, Brazil, 1996.
15. Schwabe, D. Rossi, G. A Conference Review System with OOHD. 1st International Workshop on Web-Oriented Software Technology. Valencia, Junio 2001

## **6- CORRESPONDENCIA**

D<sup>a</sup>. María José Escalona Cuaresma.

ETS de Ingeniería Informática. Dpto. Lenguajes y Sistemas Informáticos.

Av. Reina Mercedes S/N (41012 Sevilla)

Tlf. 954553867 (e-mail: [escalona@lsi.us.es](mailto:escalona@lsi.us.es))