



## **Universidad de las Ciencias Informáticas**

MEJORA DE CALIDAD EN EL PROCESO DE DESARROLLO DE SOFTWARE  
DIRIGIDO POR CASOS DE USO.

### **Autoras:**

Ing. Yaima Bety Suárez Gijón

**Departamento:** Ciencias Básicas

**Institución:** Universidad de las Ciencias Informáticas-Facultad 10

**Correo electrónico:** [ysuarezg@uci.cu](mailto:ysuarezg@uci.cu)

Ing. Janet Díaz Magdariaga

**Institución:** OSRI (Oficina de Seguridad de Redes Informáticas)

**Correo electrónico:** [janet@osri.gov.cu](mailto:janet@osri.gov.cu)

El proceso de desarrollo de software dirigido por casos de uso es una de las prácticas más utilizadas en el mundo. En la actualidad los casos de uso han alcanzado gran aceptación en el desarrollo de sistemas software por su sencillez, adaptabilidad y facilidad de comprensión.

El objetivo que persigue este trabajo de diploma investigativo es proponer un conjunto de técnicas basadas en el desarrollo dirigido por casos de uso que haga más eficiente el proceso de desarrollo de software y contribuya a la mejora de la calidad de los productos construidos en la UCI.

Para fundamentar la investigación se hizo un estudio en los proyectos productivos que utilizan la metodología RUP (está dirigida por caso de uso) para conocer como se lleva a cabo el proceso de desarrollo de software y hacer una propuesta que contribuya a crear las bases a la mejora de la calidad en la realización de los productos desarrollados en los proyectos de la Universidad.

**Palabras claves**

Proceso de desarrollo de software, casos de uso, mejora de calidad

## **CONTENIDO**

<i>INTRODUCCIÓN</i> .....	4
<i>DESARROLLO</i> .....	6
1. ACEPTACIÓN DE CASOS DE USO EN LA INDUSTRIA DEL SOFTWARE .....	6
2. PROPUESTA DE TÉCNICAS BASADAS EN EL DESARROLLO DE SOFTWARE DIRIGIDO POR CASOS DE USO .....	7
<i>CONCLUSIONES</i> .....	11
<i>REFERENCIAS BIBLIOGRÁFICAS</i> .....	12
<i>BIBLIOGRAFÍA CONSULTADA</i> .....	13

En los últimos años la industria del software ha alcanzado un alto nivel de desarrollo, cada día los sistemas se tornan más complejos, los clientes exigen soluciones más novedosas y con mayor calidad. La obtención de un producto con calidad implica la adopción por parte del equipo de desarrollo de metodologías y estándares para la construcción del software. Permitiendo uniformar los esquemas de trabajo, normalizando cada área de conocimiento de la disciplina encargada del estudio del desarrollo de sistemas informáticos. Esta es llamada desde la década de los 80 como Ingeniería de Software.

En nuestro país el desarrollo de software ha tenido un impacto económico-social en la venta de sus productos. Como aporte a la economía del país surge la Universidad de las Ciencias Informáticas (UCI) con el objetivo de formar ingenieros con un nivel altamente calificado en la rama de la informática y que durante su formación estén vinculados a la producción de software.

Actualmente en los proyectos productivos de la UCI se está encausando esfuerzos para mejorar el proceso de desarrollo, la planificación, especificación de requerimientos y pruebas[1]. Impactando los atributos de calidad fijados para cada uno de los productos a desarrollar. Este trabajo forma parte de un conjunto de investigaciones dirigidas a mejorar las técnicas relacionadas con la producción de software en procesos dirigidos por casos de uso.

Por tanto el problema a resolver queda formulado a modo de interrogante de la siguiente forma: ¿Cómo mejorar los procesos de desarrollo de software dirigidos por casos de uso?

El proceso de desarrollo del software es el objeto de estudio, y el campo de acción que abarca este trabajo es el desarrollo dirigido por casos de uso.

El objetivo general de esta investigación es proponer un conjunto de técnicas basadas en el desarrollo dirigido por casos de uso que haga más eficiente el proceso de desarrollo de software y contribuya a la mejora de la calidad de los productos construidos en la UCI.

Como idea a defender se propone ajustar el proceso de desarrollo de software de los proyectos aplicando técnicas centradas en los casos de uso creando las bases a la mejora de la calidad.

Para desarrollar la investigación nos trazamos las siguientes tareas:

- Estudiar la conceptualización del proceso de desarrollo de software dirigido por casos de uso.
- Estudiar la metodología RUP (que es la que esta dirigida por casos de uso).
- Realizar encuesta a los proyectos de la facultad que utilizan la metodología RUP.
- Analizar los resultados obtenidos de las encuestas aplicadas para conocer la forma que desarrollan sus productos y las principales deficiencias que presentan al hacerlo.
- Proponer el conjunto de técnicas que den solución al objetivo planteado.

## **MATERIALES Y MÉTODOS**

Con el propósito de desarrollar las tareas planteadas, se utilizaron los métodos de investigación siguientes:

El método teórico utilizado en el presente trabajo es el Analítico – Sintético ya que mediante la consulta de libros, revistas, monografías, sitios web, catálogos, publicaciones se ha hecho un análisis profundo del objeto de estudio para sintetizar todas las citas, apuntes, datos, resúmenes tomados al respecto.

También se puso en práctica el método Empírico donde se usó la encuesta para la recogida de información sobre cómo se realiza el proceso de desarrollo de software dirigido por casos de uso en los proyectos productivos y la entrevista para recoger las opiniones de la situación actual sobre el tema investigado.

## **1. ACEPTACIÓN DE CASOS DE USO EN LA INDUSTRIA DEL SOFTWARE**

En la actualidad los casos de uso son un método que, justamente, ayudan al Ingeniero de Software a llevar adelante el desarrollo de un sistema de software. La Industria Cubana del Software (ICSW) está llamada a convertirse en una significativa fuente de ingresos nacional, como resultado del correcto aprovechamiento de las ventajas del considerable capital humano disponible. La promoción de la industria cubana del software en el ámbito internacional ha tenido como línea estratégica aprovechar la enorme credibilidad que tiene Cuba en sectores tales como la salud, la educación y el deporte. Continuar la producción sostenida de software de alta calidad en prestaciones, imagen y soporte, para satisfacer las necesidades nacionales en estos sectores, tiene ya una positiva repercusión en el incremento de la exportación.

[2]

La Universidad de las Ciencias Informáticas y el sistema de empresas cubanas vinculadas a la producción de software jugarán un papel importante en el desarrollo de la Industria Cubana del Software, y en la materialización de los proyectos asociados al programa cubano de informatización.

En la actualidad el proceso de desarrollo de software dirigido por casos de uso es una de las mejores prácticas utilizadas en el mundo, la cual permite capturar y analizar la funcionalidad de un sistema desde el punto de vista del usuario, que desarrollado a través de diferentes modelos permite su implementación y posterior actualización.

Los casos de uso representan las condiciones y capacidades que el sistema de software deberá cumplir y tener respectivamente, desde el punto de vista funcional y son las únicas piezas de información utilizadas por todos los participantes del sistema software.

Cuando los casos de uso son bien aplicados, las probabilidades de éxito de los proyectos de ingeniería de software se incrementa de manera importante. Se utilizan básicamente en el proceso de modelado de sistemas, partiendo de una percepción o perspectiva que nos plantea el paradigma de la orientación a objetos, ya sea desde una petición de un actor o bien desde la invocación desde otro caso de uso. Se

centran en describir cómo es el diálogo entre el usuario y el sistema y se prolongan a lo largo del tiempo mientras dure la interacción del usuario con el sistema.

Todo sistema de software ofrece a su entorno una serie de servicios, donde el caso de uso es una forma de expresar cómo alguien o algo externo a un sistema lo usa. Cuando decimos “alguien o algo” hacemos referencia a que los sistemas son usados no sólo por personas, sino también por otros sistemas de hardware y software.

Es importante destacar que los casos de uso poco tienen que ver con entender a un sistema como un conjunto de objetos que interactúan, que es la premisa básica del análisis orientado a objetos. En este sentido, el éxito de los casos de uso no hace más que dar la razón al análisis estructurado, que propone que la mejor forma de empezar a entender un sistema es a partir de los servicios o funciones que ofrece a su entorno, independientemente de los objetos que interactúan dentro del sistema para proveerlos.

Los casos de uso son independientes del método de diseño que se utilice, y por lo tanto del método de programación. Luego de documentar los requerimientos de un sistema con casos de uso, se puede diseñar un sistema estructurado (manteniendo una separación entre datos y funciones), o un sistema Orientado a Objetos, sin que la técnica sea de mayor o menor utilidad en alguno de los dos casos. Esto da más flexibilidad al método, y probablemente contribuya a su éxito.

La notación de los casos de uso fue incorporada al lenguaje estándar de modelado UML (Unified Modelling Language) propuesto por Ivar Jacobson, James Rumbaugh y Grady Booch, tres de los precursores de las metodologías de Análisis y Diseño Orientado a Objetos, y avalada por las principales empresas que desarrollan software en el mundo. [3]

“Un caso de uso es una técnica para la captura de requisitos de un nuevo sistema o una actualización software. Cada caso de uso proporciona uno o más escenarios que indican cómo debería interactuar el sistema con el usuario o con otro sistema para conseguir un objetivo específico”. [4]

## **2. PROPUESTA DE TÉCNICAS BASADAS EN EL DESARROLLO DE SOFTWARE DIRIGIDO POR CASOS DE USO**

Para perfeccionar el proceso de desarrollo del software y contribuir a la mejora de la calidad de los productos desarrollados en la UCI se decidió proponer las siguientes técnicas basadas en casos de uso:

En la fase de inicio, específicamente en el flujo de trabajo Levantamiento de requisitos se propone aplicar la técnica Storyboard para lograr entender las funcionalidades del sistema y así definir los requisitos. Luego se utiliza la técnica Taller de casos de uso con el objetivo de definir los actores, casos de usos y requisitos adicionales. Estas técnicas se describen a continuación:

**Técnica “Storyboard”:** El propósito principal de esta técnica es entender el flujo global y las interacciones del sistema. Permite realizar la descripción lógica y conceptual de las funcionalidades del sistema para un escenario específico, e incluso la interacción requerida entre los usuarios y el sistema.

Proporciona importantes mecanismos de retroalimentación para descubrir requisitos inciertos o desconocidos. Explora, clarifica y captura la interacción prevista por el usuario como parte de adquisición de requisitos. Es utilizada para diseñar la interfaz de usuario y construir un prototipo de la misma. En las pruebas se usa para probar las características del sistema. Se usa para planear y llevar a cabo el análisis y diseño del trabajo.

**Técnica “Taller de casos de uso”:** El taller de casos de uso se comporta como una especialización de Tormenta de ideas. Es una reunión organizada donde las personas que participan deben tener un amplio conocimiento del proceso de desarrollo de software, es decir, el grupo contendrá personas con diferentes ideas y experiencias. Las principales acciones a realizar en esta técnica son la Definición de los actores, Definición de los casos de uso, Descripción de los casos de uso, Capturar los requisitos adicionales y Tracear los requisitos para Casos de uso.

Es necesario determinar los atributos de los requisitos definidos en el proceso para saber que función cumple cada caso de uso, por lo que se propone que los atributos que no pueden dejar de precisarse son:

**Atributo “estado”:** El estado de los requisitos se establecen después de la negociación y revisión por el equipo de dirección de proyecto y su rastreo progresa durante la definición de la línea del proyecto. Los estados de los requisitos pueden clasificarse en propuestos, aprobados e incorporados.

**Atributo “prioridad”:** La prioridad de los requisitos hace referencia al orden temporal en que debe realizarse ese requisito e indica en las fases que se incluirá cada uno. Este atributo es establecido por el líder del proyecto. Se clasifica en alto, medio y bajo.

**Atributo “Autor” o “Fuente”:** Este atributo permite conocer la procedencia de un requisito, de posibilitar el acceso a una mayor información sobre el requisito, que

puede ser necesaria en caso de modificaciones o resolución de incompatibilidades.

Podría indicar la fuente de donde se han inspirado para elaborar un requisito, de modo que esta información pueda serles útil más adelante si deben especificar el requisito.

**Atributo “Esfuerzo”:** Este atributo es establecido por el equipo de desarrollo. Algunos casos de uso requieren más tiempo y recursos que otros, estimando el número de equipo o persona por semanas, las líneas de código requeridas, por ejemplo, es la mejor manera de evaluar la complejidad y establecer las expectativas de lo que puede o no lograrse en un horario dado.

**Atributo “Riesgo”:** Establecido por el equipo de desarrollo basado en la probabilidad de que el caso de uso experimentará eventos indeseables, como los desbordamientos de esfuerzo, retrasos del horario o la cancelación. La mayoría de líderes de proyectos encuentran la categorización de riesgos como alto, el medio, y bajo. El riesgo puede evaluarse a menudo indirectamente midiendo la incertidumbre (el rango) de los horarios estimados de los proyectos de equipos. El riesgo puede ser alto, medio y bajo.

Para facilitar el proceso de administración de requerimientos se propone aplicar la estrategia de trazabilidad **“El modelo de Casos de uso es una interpretación de la especificación de requerimientos de software”**. En este caso el modelo de casos de uso se usa para modelar y reinterpretar todos los requisitos de software dentro del alcance del proyecto. Cuando se adopta esta estrategia es usual que se haga primero la especificación de los requisitos del software, hay otras técnicas disponibles para dar la información sostenida por el modelo de casos de uso en un formato que parece formal, la SRS tradicional (sobre todo cuando se asume una estrategia donde los requerimientos guían al modelo de casos de uso) sin crear una segunda definición de requisitos de software.

Al adoptar esta estrategia no hay necesidad de que el conjunto de requisitos del software tradicionales sea una declaración completa de la funcionalidad requerida ya que el modelo de casos de uso suministra o asegura la integridad de la especificación funcional. Los requisitos del software tradicionales pueden usarse para capturar los requisitos del software identificados directamente o pueden capturarse por los involucrados (stakeholders).

Utilizando la descripción textual de los casos de uso se aplica el **“Método de cálculo de Puntos de Casos de Uso”** para obtener el tamaño del producto software y a partir del mismo generar las estimaciones del esfuerzo ayudando a organizar y planificar el trabajo individual y el del equipo.

Para lograr una mejor planificación del proyecto se propone usar la estrategia **“Dirigido por Casos de Uso”**.

Utilizando esta estrategia se comienza priorizando los requisitos para llevar a cabo los de prioridad más alta, con el objetivo de minimizar el riesgo de que el proyecto sea cancelado debido a la falta de confianza. Se reduce el riesgo de construir un sistema equivocado ya que se puede obtener la retroalimentación de los usuarios para validar que lo que se está construyendo es correcto.

Brinda diversos beneficios, por ejemplo, desarrolla valiosos y coherentes casos de uso, permite a los diseñadores adquirir un alto entendimiento del negocio mientras están trabajando con los casos de uso, ya que estos describen la funcionalidad fundamental del negocio, facilita la comprensión del plan construido con la información brindada por el usuario, ya que en los casos de uso se recoge de forma detallada y clara.

A partir de las descripciones de los casos de uso definidos en el sistema se propone aplicar la técnica **“Pruebas tempranas”** para generar pruebas ejecutables y mejorar la calidad de los requisitos detectando errores, omisiones, inconsistencias en los mismos. Es diseñada específicamente para ser aplicada cuando el sistema aún no está construido o, incluso, diseñado. La generación de pruebas en etapas tempranas del desarrollo permite corregir los defectos de omisión de escenarios, resultados sin escenarios o escenarios sin resultado, nuevos requisitos funcionales y ambigüedad.

## ***CONCLUSIONES***

---

La realización de este Trabajo de Diploma fue de un gran aporte a los autores por los conocimientos y resultados obtenidos, que fueron los esperados. Se cumplieron los objetivos propuestos de forma satisfactoria.

Se desarrolló un análisis en los proyectos en cuanto al uso de la metodología RUP y las técnicas de casos de uso, donde se aplicó una encuesta a los líderes para conocer como se llevaba en práctica el proceso de desarrollo de software.

A partir del análisis realizado se plantea aplicar las técnicas propuestas anteriormente y que cada responsable de proyecto pueda analizar los resultados de la puesta en práctica y de esta forma retroalimentar y desarrollar conclusiones de la investigación.

Este trabajo servirá de referencia para futuros cursos, para generar nuevos temas de investigación y para paulatinamente aplicar los resultados en todos los proyectos que utilicen la metodología RUP.

## ***REFERENCIAS BIBLIOGRÁFICAS***

---

1. Fortián, J.D. (2006) *Mejora Continua: Reto y Necesidad*.
2. *La informatización en Cuba*. 2004 [cited 2006 13 de noviembre]; Available from:  
[http://www.cubaminrex.cu/Sociedad\\_Informacion/Cuba\\_SI/Informatizacion.htm](http://www.cubaminrex.cu/Sociedad_Informacion/Cuba_SI/Informatizacion.htm).
3. Ceria, S. *Un Método Práctico para Explorar Requerimientos*.
4. *Caso de uso*. 2007 [cited 2006 13 de noviembre]; Available from:  
[http://es.wikipedia.org/wiki/Caso\\_de\\_uso](http://es.wikipedia.org/wiki/Caso_de_uso).

## BIBLIOGRAFÍA CONSULTADA

---

- Koch, M.J.E.y.N. *Ingeniería de Requisitos en Aplicaciones para la Web – Un estudio comparativo*. 2002 [cited 2007 20 de febrero]; Available from: <http://lsiweb.lsi.us.es/docs/informes/LSI-2002-4.pdf>
- J. J. Gutiérrez, M.J.E., M. Mejías, J. Torres. *Mejora de la calidad de los requisitos mediante la generación de pruebas*. [cited 2007 15 de abril]; Available from: [http://www.lsi.us.es/~javierj/investigacion\\_ficheros/Mejorando%20casos%20de%20uso%2003.pdf](http://www.lsi.us.es/~javierj/investigacion_ficheros/Mejorando%20casos%20de%20uso%2003.pdf).
- Javier J. Gutiérrez, M.J.E., Arturo H. Torres, Manuel Mejías y Jesús Torres. *HACIA UNA PROPUESTA DE PRUEBAS TEMPRANAS DEL SISTEMA*. [cited 2007 20 de mayo]; Available from: <http://in2test.lsi.uniovi.es/pris2006/PRIS2006-GutierrezEscalonaMejiasTorres.pdf>.
- Javier J. Gutiérrez, M.J.E., Manuel Mejías y Antonia M. Reina. *MODELOS DE PRUEBAS PARA PRUEBAS DEL SISTEMA*. [cited 2007 20 de mayo]; Available from: <http://www.lsi.us.es/~javierj/publications/MDA14.pdf>.
- Peralta, M. *ESTIMACIÓN DEL ESFUERZO BASADA EN CASOS DE USO*. 2004 [cited 2007 21 de febrero]; Available from: <http://www.itba.edu.ar/capis/rtis/rtis-6-1/estimacion-del-esfuerzo-basada-en-casos-de-usos.pdf>.
- Gracia, J. *Desarrollo de Software Orientado a Objetos*. 2003 [cited 2006 25 de noviembre]; Available from: <http://www.ingenierosoftware.com/analisisydiseno/casosdeuso.php>.
- Javier J. Gutiérrez, M.J.E., Manuel Mejías y Antonia M. Reina. *Definición de casos de uso para procesos de generación automática de pruebas del sistema*. [cited 2007 24 de febrero]; Available from: <http://www.lsi.us.es/~javierj/publications/Hito02.pdf>.
- Mestras, J.P. *Proceso Unificado y la captura de requisitos*. [cited 2007 10 de enero]; Available from: <http://www.fdi.ucm.es/profesor/jpavon/is2/04Requisitos.pdf>.
- Larman, C., *UML y Patrones*. 2da edición.
- Probasco, I.S.a.L. (2000) *Traceability Strategies for Managing Requirements with Use Cases*.  
[http://materjalid.tmk.edu.ee/kaarel\\_allik/ISP/traceabilityStrategies.pdf](http://materjalid.tmk.edu.ee/kaarel_allik/ISP/traceabilityStrategies.pdf)