

CUPIDO - Plantilla para Documentar Casos de Uso

María Inés Lund, Cintia Ferrarini, Laura Aballay, María G. Romagnano, Ernesto Meni

Instituto de Informática

Facultad de Ciencias Exactas Físicas y Naturales

Universidad Nacional de San Juan

{mlund, ferrarini, laballay, maritaroma, emeni}@iinfo.unsj.edu.ar

Resumen

En este trabajo se propone una plantilla para documentar Casos de Uso, con el fin de guiar al usuario, en particular al alumno, en las etapas de análisis y diseño dentro de un proceso de desarrollo de software.

La propuesta constituye una adaptación de plantillas muy difundidas en importantes empresas de desarrollo de software; a situaciones de desarrollo de software local y/o regional, más específicamente para ámbito de los sistemas administrativos en sus etapas de Análisis y Diseño.

Se expone también el proceso de validación de la plantilla y cómo los alumnos, utilizando la misma como un artefacto de apoyo en el proceso de desarrollo de software, participan en el proceso de validación.

Palabras Clave: Casos de Uso, Ingeniería de software, Modelo de Casos de Uso, Plantilla de Casos de Uso, Proceso de desarrollo de software.

Introducción

Los proyectos de desarrollo de software suelen fallar, por diversos motivos, siendo los más comunes, la entrega tardía y el aumento de los costos de desarrollo, superando lo planificado y presupuestado. La causa de estos problemas es la falta de concordancia o pobre relación entre las necesidades reales de los usuarios finales del sistema y los requisitos elicitados [1].

Actualmente, si se desea construir una aplicación de tamaño considerable, un diseño de software consistente y bien documentado es un componente necesario. No se puede comunicar la estructura de la aplicación o como ésta trabaja, sin alguna representación o modelo que exprese esa aplicación. Carecer de un diseño hará imposible construir un sistema extensible, adaptable y escalable. Por lo tanto, es necesario desarrollar modelos, que acompañen y guíen todo el proceso de desarrollo de software y que permitan su actualización conforme vayan cambiando los requisitos del usuario, manteniéndose consistentes [2].

En DeMarco [3], se propone la Ingeniería de Software basada en modelos, donde compara la construcción de un sistema de software con la construcción de cualquier otro tipo de sistemas ingenieriles, y por lo tanto propone la realización de modelos del sistema antes de la construcción del sistema mismo. De esta forma, el modelo de un sistema provee un medio de comunicación entre todos los participantes del proyecto: clientes, usuarios y desarrolladores. Los métodos de desarrollo de software que se usan en la actualidad adoptaron la filosofía propuesta por este autor, considerando que la construcción de sistemas depende de modelos previamente definidos.

El Modelo de Casos de Uso, constituye uno de los modelos más utilizados por la industria del Software debido principalmente a su simplicidad para definir la funcionalidad del sistema desde el punto de vista de sus usuarios. Además, el modelo de casos de uso ayuda al cliente, a los usuarios y a los

desarrolladores a llegar a un acuerdo sobre cómo debe funcionar el sistema.

El Modelo de Casos de Uso posee un conjunto de elementos llamados ‘casos de uso’, en donde cada caso de uso es una pieza en la funcionalidad del sistema que tiene como fin entregar al usuario un resultado de valor [4].

Al parecer, describir un Caso de Uso sería una tarea sencilla; pero ¿cuánto escribir?, ¿qué detallar?, ¿cómo escribirlo?. Además existen distintas formas de representarlo, como por ejemplo: texto u oraciones estructuradas, diagramas de flujo, diagramas de secuencia, redes de Petri, lenguajes de programación [1]. Sin embargo, más allá de todas las herramientas que ayuden a su definición, el desafío es que éste refleje correctamente la realidad que intenta representar. En general, es de amplia aceptación la descripción textual de los casos de uso, debido a que es más sencillo de interpretar por parte de los diferentes usuarios interesados (stakeholders). Esta descripción textual usualmente se plasma en plantillas [5], constituyéndose en uno de los artefactos [6] claves que el modelador puede observar durante el proceso de desarrollo de software [7].

En este trabajo, se consideran a las plantillas como documentos genéricos que facilitan el proceso de documentación de un proyecto de software, por lo que deben responder a estándares internacionales que normen su formato y contenido.

En la actualidad no existe un estándar para documentar casos de uso. En este trabajo se presenta el artefacto denominado CUPIDO (Casos de Uso, Plantilla Integradora para Documentarlos) la cual es una plantilla para describir Casos de Uso en las etapas de análisis y diseño.

Nuestra propuesta, basada en plantillas o estructuras para documentar casos de uso, de autores reconocidos a nivel nacional e internacional [8][9][11][12][13], se adapta a

las necesidades y ámbitos de desarrollo de software local y/o regional, en el contexto de los sistemas administrativos, donde la formalización y documentación de sistemas es escasa o nula y cuando existe es netamente informal. Salvo algunas organizaciones dedicadas a software factory, en donde están obligadas a cumplir con ciertos estándares internacionales, el resto son anotaciones iniciales en la captura de requisitos y luego nunca actualizadas a los cambios de requerimientos.

Marco Teórico

Ingeniería de Software

Según la definición del IEEE, citada por Lewis [14] “*software es la suma total de los programas de computadora, procedimientos, reglas, la documentación asociada y los datos que pertenecen a un sistema de cómputo*”.

Según el mismo autor, “*un producto de software es un producto diseñado para un usuario*”. En este contexto, la Ingeniería de Software es un enfoque sistemático del desarrollo, operación, mantenimiento y retiro del software, en otras palabras se considera que “*la Ingeniería de Software es la rama de la ingeniería que aplica los principios de la ciencia de la computación y las matemáticas para lograr soluciones costo-efectivas a los problemas de desarrollo de software*” [15].

El proceso de ingeniería de software se define como “*un conjunto de etapas parcialmente ordenadas con la intención de lograr un objetivo, en este caso, la obtención de un producto de software de calidad*” [6].

El proceso de desarrollo de software “*es aquel en que las necesidades del usuario son traducidas en requerimientos de software, estos requerimientos transformados en diseño y el diseño implementado en código, el código es probado, documentado y certificado para su uso operativo*” [16]. Concretamente define quién está haciendo qué, cuándo hacerlo y

cómo alcanzar un cierto objetivo.

La Ingeniería de Software ofrece modelos, métodos y técnicas para desarrollar y mantener software de calidad. Los modelos presentan un marco conceptual donde reflejar las teorías, plasmar propiedades y establecer los principios del diseño de los sistemas. Su importancia radica en que permiten identificar, organizar y realizar razonamientos sobre los componentes y comportamiento del sistema, son la guía para el proceso de diseño del software y puede usarse posteriormente como una referencia para evaluar un diseño particular, razonar sobre la solución realizada y sobre el posible espacio de soluciones [17]. Entre estos puede mencionarse el modelo de casos de uso.

Modelo de Casos de Uso

Se puede decir, de manera informal, que un caso de uso describe la forma en que un sistema es empleado por los actores (usuarios del sistema) para alcanzar sus objetivos. Tiene una notación gráfica y está acompañado de una descripción de lo que hace [18]. Generalmente esta descripción se expresa en plantillas que permiten guiar, portar o construir un esquema predefinido, reflejar los elementos comunes de los Casos de Uso [7,].

Un modelo de casos de uso es un modelo del sistema que contiene actores, casos de uso y sus relaciones y describe lo que hace el sistema para responder a las necesidades de cada actor o usuario del sistema.

En resumidas cuentas, tal como dice Jacobson en [19]: “*El modelo de casos de uso permite que los desarrolladores del software y los clientes lleguen a un acuerdo sobre los requisitos, es decir, sobre las condiciones y posibilidades que debe cumplir el sistema. El modelo de casos de uso sirve como acuerdo entre clientes y desarrolladores y proporciona la entrada fundamental para el análisis, el diseño y las pruebas.*”

Metodología de Trabajo

La investigación realizada adoptó un diseño metodológico teórico – bibliográfico, con posteriores pruebas experimentales (experimentos controlados) siguiendo algunas pautas propuestas por Wohlin [20] en Ingeniería de Software Experimental.

En primer lugar se identificaron autores en modelamiento de casos de uso de amplia trayectoria científica [8][9][11][12][13]. En segundo lugar se recopiló información sobre la utilización y aplicación de casos de uso, además de las diferentes formas de especificar requisitos y casos de uso [11][10]. Posteriormente se identificaron las distintas etapas de la ingeniería de software en donde se utilizan o podrían ser utilizados los modelos de casos de uso, centrando la atención en las características que poseen dichas etapas, en cuanto a las necesidades de información para ser consideradas o no en la plantilla [6][15][16][18][19].

Este trabajo puso de manifiesto que mientras algunas de las plantillas y propuestas para documentar requisitos no contemplaban en forma provechosa las necesidades adaptadas al contexto local y regional, otras imponían requisitos que aumentaban la complejidad y extensión de la documentación [21][8][9][11][12][13]^{1,2,3,4,5}.

Debido a ello y en base a lo relevado, y como dicen Kulak y Guiney en [11] “*Se necesita una plantilla común para los casos de uso en un proyecto. La estandarización en una plantilla es más importante que lo que la propia plantilla parece...O mejor de todo,*

¹ [http://users.dsic.upv.es/asignaturas/facultad/lsi/trabajos/032000\(2\).doc](http://users.dsic.upv.es/asignaturas/facultad/lsi/trabajos/032000(2).doc)

² http://dc.exa.unrc.edu.ar/nuevodic/materias/sistemas/2008/Teoricos/TEORIA_PlantillasGenericas-CU-en4.pdf

³ http://www.vico.org/aRecursosPrivats/UML_TRAD/talleres/mapas/UMLTRAD_101A/

⁴ <http://synergix.wordpress.com/2008/06/04/ejemplo-de-caso-de-uso/>

⁵ <http://www.miguelmatas.es/blog/2007/10/17/plantilla-completa-de-casos-de-uso/comment-page-1/#comment-1167>

puede crear su propia plantilla que aborde las necesidades de su organización”, se consideró la necesidad de diseñar artefactos que permitieran optimizar la descripción de los Casos de Uso a un contexto local y regional, para cada una de las etapas del proceso de desarrollo de software.

Además, considerando que las etapas de Análisis y Diseño son críticas en el proceso de desarrollo de software [18], la plantilla CUPIDO está específicamente diseñada para dichas etapas, reuniendo características de completitud, comprensibilidad, consistencia, simplicidad y facilidad de uso.

El diseño del artefacto plantilla, fue refinado en sucesivas revisiones por cada uno de los integrantes del proyecto CUPROSoft, siguiendo un proceso iterativo – incremental. Cada iteración permitió incorporar/modificar elementos, definiendo un orden lógico de prioridad, que le permitiera a todos los stakeholders comprender mejor los casos de uso.

En la figura 1 se muestra un modelo del proceso iterativo – incremental utilizado por el equipo de investigación. Los círculos representan a los integrantes mientras que las flechas modelan las interacciones entre estos y las sucesivas versiones de la plantilla.

Con cierta periodicidad se realizaron reuniones integrales, a fin de acordar aspectos y unificar criterios de acción.

Como resultado de aplicar este proceso se obtuvo una plantilla, refinada para la etapa de análisis y diseño, denominada CUPIDO.

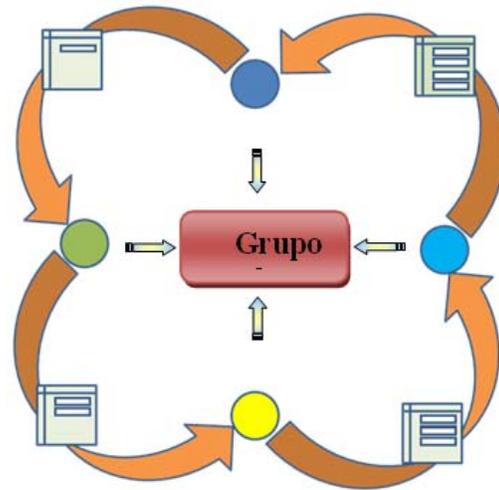


Figura 1: Proceso Iterativo- Incremental aplicado por el equipo de investigación

Organizada de forma que los ítems a llenar fuesen simples, comprensibles, consistentes, fáciles de usar, expresados en lenguaje natural, entre otros aspectos. Con un orden coherente, según los criterios de usabilidad – calidad obtenidos del análisis de los principios de usabilidad y las dimensiones de calidad en el desarrollo de software [22].

La Plantilla CUPIDO

El artefacto plantilla está conformado por tres secciones: Encabezado, Hoja de revisión y Cuerpo, las cuales se detallan a continuación.

Encabezado

En ésta sección de la plantilla se describe el identificador del proyecto, el paquete al que pertenece el caso de uso a describir, el identificador del caso de uso, la versión (se asumen sucesivas mejoras/refinamientos al detalle del caso de uso), hoja (página), estado de construcción en que se encuentre la descripción del caso de uso (puede ser en proceso de elaboración, en verificación o terminada) y por último, nombre del documento digital en el que encuentra la plantilla, como se muestra en la tabla 1.

Proyecto: <Id. Proy.><Nombre del Proyecto>	Paquete: [<Id. Paquete><Nombre del paquete>]		
Caso de Uso: <Id. CU> <Nombre del Caso de Uso>	Versión: x.x	Fecha revisión: dd/mm/aaaa	Hoja: 1/2
Estado: [En elaboración/ En verificación/ Terminada]	Nombre del documento digital: <Nombre del documento>		

Tabla 1: Encabezado de la Plantilla

Los datos del encabezado permiten hacer la trazabilidad del caso de uso a través de los paquetes y proyectos de software y a su vez hacia las posteriores etapas de desarrollo del mismo. Además posibilita la identificación rápida de la última versión del documento y el estado en que se encuentra.

Hoja de Revisión

En la tabla 2 se presenta la “Hoja de Revisión”, en la que cada integrante del proyecto de desarrollo del sistema, que participe en las etapas de análisis y diseño, deberá detallar la fecha, versión y el motivo de esta modificación (un breve detalle).

Fecha	Versión	Descripción	Autor
<dd/mm/aaaa>	<x.x>	<detalles>	<nombre>

Tabla 2: Hoja de Revisión

La hoja de revisión permite monitorear la evolución de la descripción del caso de uso, identificando los momentos de cambio y las razones del mismo. Esto posibilita rastrear las versiones anteriores de la plantilla y analizar en detalle esos cambios si fuera necesario.

Cuerpo de la Plantilla

Como se muestra en la tabla 3, esta sección posee trece campos, dispuestos en tres columnas: la primera con la numeración (orden lógico - secuencial), en la segunda el nombre del campo y en la tercera la descripción del mismo, que es donde el usuario de la plantilla describirá los datos solicitados.

El cuerpo de la plantilla permite realizar la descripción del caso de uso en forma ordenada, con indicaciones separadas de detalles del

Nº	Nombre del Campo	Descripción del Campo
1.	Nombre del CU:	< Nombre representativo de la función que realiza el CU.>
2.	Actor	<Rol de quién inicia el CU>
	Tipo de Asociación	<Asociación del CU con otro/s CU> [Incluido/ extendido/ generalización padre/ generalización hijo/]. En el caso de generalización hijo, se debe indicar el caso de uso padre <Id. CU> y/o <nombre del caso de uso padre>, si forma parte de más de una asociación, se deben indicar todas.
4	Breve Descripción:	<Descripción resumida y clara de la funcionalidad del CU. Puede hacer referencia a documento/s que contenga/n información relacionada al CU.>
5	Precondiciones:	[<Condiciones que se deben cumplir o estado en el cual se debe encontrar el sistema antes de que el CU pueda iniciarse>]
6	Flujo de Eventos:	<Descripción detallada de las acciones de los actores y las respuestas del sistema que se llevarán a cabo durante la ejecución del CU.>
7	Poscondiciones:	[<Estado del sistema a la conclusión de la ejecución del CU>]
8	Extensión (extend):	[<Id. Paquete> y/o <nombre de paquete> si corresponde] [< Id. CU> y/o <nombre del CU llamado desde este CU como excepción (CU al que se extiende)>]
9	Inclusión (include):	[<Id. Paquete> y/o <nombre de paquete> si corresponde] [< Id. CU> y/o <nombre del CU llamado desde este CU como inclusión (CU que se incluye)>]
10	Requisitos asociados	[<Enumeración de Id. de requisitos que satisface el CU> / <nombre del documento que los contiene>]
11	Consideraciones / Observaciones:	[<Comentarios, requisitos adicionales, no funcionales relacionados al CU a tener en cuenta en el diseño o la implementación>]
12	Suposiciones:	[<Son las condiciones o premisas que se tuvieron en cuenta en el análisis, que permitieron incluir a este CU en la descripción del sistema>]
13	Frecuencia de Uso:	[<Estimación del número de veces que el CU se ejecutará por los actores en una unidad de tiempo.>] [<Nº de veces> / <unidad de tiempo>]

Tabla 3: Cuerpo de la plantilla.

mismo sin perjudicar el nivel de abstracción deseado. Se sugiere el uso del lenguaje natural, no se indica la forma de llenarlo, sino qué características específicas deben ser analizadas e indicadas en campos separados, como los casos de uso incluidos o los extendidos.

Si existiera un documento de requisitos de software, se debe especificar cuales son los requisitos involucrados a fin de facilitar la trazabilidad de los requisitos durante el proceso de desarrollo.

Las suposiciones son aquellas condiciones de hecho que se dan en el contexto del dominio y que influyeron en la determinación de éste como un caso de uso con las características descritas, entre otras.

La frecuencia de uso permite analizar aquellos casos de uso más utilizados y apoyar en la toma de decisiones en la etapa que sea necesaria.

Modelo de Validación de la Plantilla

Debido a que consideramos que ninguna plantilla puede arrogarse tener la “última palabra” respecto de su estructura, debiendo en todo momento respetar cada desarrollo de software y por ende cada equipo de desarrolladores, se planteó un modelo de validación de la plantilla que tiene como objetivo obtener e incorporar mejoras en la misma a través de un proceso de retroalimentación (feedback) a fin de alcanzar un artefacto óptimo, consensuado y formalizado.

En el modelo de validación se tuvieron en cuenta los posibles usuarios, sus necesidades, expectativas, perspectivas de uso y las diferentes interacciones e intervenciones que se pueden presentar.

Roles e Interacciones

En la figura 2 se pueden observar los grupos de usuarios representados por círculos (grupos de validación), sus interacciones (flechas), en donde cada grupo de usuarios puede intervenir en un momento diferente o en forma simultánea y cómo estas interacciones permiten validar y retroalimentar la plantilla.

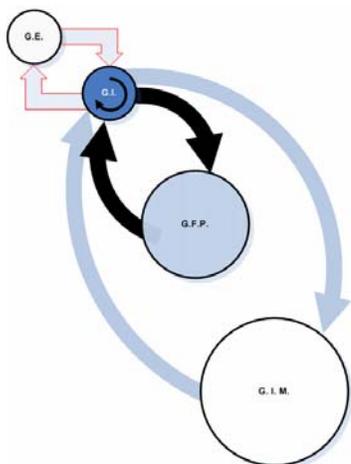


Figura 2: Modelo de Validación

GI: Grupo de Investigación (Integrantes del

proyecto).

GFP: Grupo de Futuros Profesionales (alumnos avanzados)

GE: Grupo de Expertos (Comunidad Científica)

GIM: Grupo de Implementación (Mercado, Profesionales)

El Proceso de Validación

El proceso de validación propuesto consta de cuatro instancias. La primera es realizada por el grupo de investigación (G.I.) propiamente dicho (validación interna o verificación), siguiendo la metodología de trabajo planteada anteriormente, en donde un integrante seleccionado cumple el rol de iniciador del artefacto, quien debe generar una versión preliminar y entregarla a otro integrante; éste asume a partir de ese momento un rol más activo, con la posibilidad de modificar, proponer mejoras o agregar información según su punto de vista. Este proceso se repite hasta que todos los integrantes acuerden una versión final. En una segunda instancia la plantilla será validada a través de pruebas experimentales implementadas bajo la modalidad de talleres de actualización, destinados alumnos avanzados (GFP).

Con los resultados de ésta validación ya incorporados al artefacto plantilla, se propone la tercer validación por parte del Grupo de Expertos o Comunidad Científica (GE), realizando publicaciones o presentaciones de la misma a diferentes eventos, congresos, revistas, etc. y la cuarta instancia de validación corresponde al Grupo de Implementación (GIM.) que lo constituyen los profesionales y las empresas de desarrollo del mercado.

Este proceso de validación propuesto, a excepción de la primera instancia de validación, no necesariamente debe seguir el orden cronológico planteado, incluso algunas instancias de validación pueden darse en forma simultánea, o repetirse hasta llegar a una versión de la plantilla aprobada, consensuada y formalizada.

Cabe notar, que al momento de la presentación de este trabajo se ha realizado la primera y la segunda instancia de validación y parcialmente la tercera.

Conclusiones y perspectivas futuras

Comenzamos con el diseño de CUPIDO que integra detalles de una serie de plantillas vistas, para especificar/describir casos de uso, siguiendo la metodología de trabajo antes descripta.

La primera plantilla generada fue pensada para las etapas de análisis y diseño, e intenta constituirse en una propuesta superadora de las plantillas existentes en el mercado.

Si bien la utilización del modelo de validación de la plantilla tuvo un rol preponderante en el diseño de la propuesta, el detalle del trabajo realizado no forma parte este artículo debido a que su aplicación excede el alcance del mismo. No obstante queremos hacer notar que actualmente se están realizando pruebas experimentales a través de talleres y cursos, para obtener la cuantificación estadística de la usabilidad de la plantilla, a través de cuestionarios [22].

Se pretende que la propuesta planteada evolucione, para su adaptación en forma progresiva y sucesiva, a las restantes etapas del desarrollo de software. Obteniendo de esta manera, una plantilla única, que permita describir los casos de usos en las distintas etapas de desarrollo de software.

Del mismo modo se proyecta realizar el diseño e implementación de alguna herramienta computacional que permita sistematizar y agilizar el proceso de documentación del modelo de casos de uso a través de la plantilla CUPIDO.

La transferencia al medio de CUPIDO se realizará a través de talleres, cursos de actualización y usándola como artefacto de

apoyo en las materias de Diseño de Software de las carreras de informática de la Universidad Nacional de San Juan

Por otra parte, los resultados que se obtengan son de inestimable valor para los investigadores de la educación y los sociólogos, quienes serán los encargados de evaluar el cambio que la tecnología puede ocasionar en la educación y por ende en la sociedad.

Agradecimientos

Este trabajo se enmarca dentro del proyecto “Modelo de Casos de Uso: un eje para el proceso de desarrollo de Software” - CUProSoft. Nº 21/E822-2007-Res. Nº 34/08, 133/08 y 52/09-CS-UNSJ.

Referencias

- [1] Kotonya, G. & Sommerville, I. “*Requirements engineering: processes and techniques*”. John Wiley. © 1998, Ed. 2002.
- [2] Lund, M. I., Aballay, L., Ferrarini, C., Romagnano, M. & Meni, E. “*Modelo de casos de uso ¿una solución para el proceso de desarrollo de software?*”. WICCSI2009. Facultad de Ciencias Exactas, Físicas y Naturales. Universidad Nacional de San Juan. Diciembre, 2009.
- [3] DeMarco, T. “*Structured Analysis and System Specification*”. Prentice Hall PTR Upper Saddle River, NJ, USA, 1979.
- [4] Giandin, C. & Pons, C.F. “*Relaciones entre Casos de Uso en el Unified Modeling Language*”. Revista Colombiana de Computación. Volumen 1, número 1 Págs. 73-90. LIFIA, Universidad Nacional de La Plata.
- [5] Campderrich Falgueras, B. “*Ingeniería de Software*”. Editorial UOC Aragón, 182, 08011 Barcelona. Primera edición. Abril 2003.

- [6] Jacobson, I. "Applying UML in The Unified Process" Presentación. Rational Software. 1998. Presentación disponible en <http://www.rational.com/uml:UMLconf.zip>
- [7] Larman C. "UML y Patrones. Introducción al análisis y diseño orientado a objetos". Prentice-Hall & Pearson. 1999.
- [8] Ambler, S. "Documenting a use case: What to include and why". 2000. <http://www.ibm.com/developerworks/webservices/library/ws-tip-docusecase.html>
- [9] Cockburn, A. "Writing Effective Use Cases". Addison-Wesley. © 2001, Ed. 2005.
- [10] Hurlbut, R. "A Survey of Approaches for Describing and Formalizing Use Cases". Illions Institute of Technology. 1997. www.iit.edu/~rhurlbut/xpt-tr-97-03.pdf
- [11] Kulak, D. and Guiney, E. "Use Cases: Requirement in Context" Ed. Addison-Wesley. 2004. 2ª Edición - 7º impresión 2008.
- [12] Pinciroli, F. Plantilla e Instructivo usado en la carrera de Sistemas de la U. de Congreso – Mendoza. Argentina. 2005
- [13] Rubin, David M. "Use of Use Cases". Softstar Research, Inc. Methodologies and Practices – White Paper. 1998. <http://www.softstar-inc.com/Download/Uses of Use Case.pdf>
- [14] Lewis G. 1994. "What is Software Engineering?" DataPro (4015). Feb 1994. pp. 1-10.
- [15] Cota A. "Ingeniería de Software". Soluciones Avanzadas. Julio de 1994. pp. 5-13
- [16] Jacobson I., Magnus C., Jonsson P, Overgaard, G.. "Object-Oriented Software Engineering: A Use Case Driven Approach". Addison-Wesley. © 1992, Ed.1998.
- [17] Gea M., Gutiérrez F.L., Garrido J.L., Cañas J.J.: "Teorías y Modelos Conceptuales para un diseño basado en grupos", IV Congreso Internacional de Interacción Persona- Ordenador. Vigo, España, 2003.
- [18] Pressman, R. S. "Ingeniería del software: un enfoque práctico". McGraw-Hill. 2002.
- [19] Jacobson, I. "El Proceso Unificado de Desarrollo de software". Addison-Wesley, EE.UU. 2000.
- [20] Wohlin, P. Runeson, M. Höst, M. Ohlsson, B. Regnell y A. Wesslén. "Experimentation in Software Engineering. An Introduction". Kluwer Academic Publisher. © 2000.
- [21] Marcela, D. & Romero, D. "Evolución de Plantillas Genéricas para la descripción de Casos de Uso a Plantillas para el Análisis y Diseño".(PIIMEG 2005). U. Nac. Nacional de Río Cuarto. RR N° 109-110/05. 2005.
- [22] Hayes, B. E. "Como medir la Satisfacción del Cliente: Diseño de Encuestas, Uso y Métodos de Análisis Estadístico". De Oxford México 2000. 2ª Edición.