

# Diseño de Software guiado por Casos de Uso. Aplicación práctica basada en un proceso de enseñanza-aprendizaje propuesto

María Inés Lund

Instituto de Informática, Universidad Nacional de San Juan, Argentina  
mlund@iinfo.unsj.edu.ar

**Abstract.** Es común utilizar, para el análisis de un sistema, el modelo de casos de uso a través del Lenguaje de Modelado Unificado (UML). Si bien existe numerosa bibliografía al respecto, a la hora de modelar el sistema y la aplicación práctica de los casos de uso, nos encontramos con una gran variedad de criterios sobre cómo usarlos, relacionarlos y especificarlos, que van más allá de la corrección sintáctica que puede ser fácilmente identificada, sino más bien vinculados a la semántica en el uso de los mismos. A lo largo de los años de dictado de la cátedra Diseño de Software, de 4º año de las carreras en informática de la UNSJ, se ha ido desarrollando, primero intuitivamente y luego más formalmente a través de proyectos de investigación, varias reglas, estrategias, convenciones de uso, artefactos de software, etc, que sirven de guía para una buena estrategia de diseño de software. Se expone en el presente trabajo algo de la experiencia adquirida y la estrategia de uso de casos de uso para el diseño de software, a través de aplicaciones prácticas.

**keywords:** casos de uso, diagrama de clases, especificación de casos de uso, plantilla CUPIDO.

## 1 INTRODUCCIÓN

La cátedra Diseño de Software se dicta en 4º año de las carreras Licenciatura en Ciencias de la Computación (LCC) y Licenciatura en Sistemas de Información (LSI), de la Facultad de Ciencias Exactas, Físicas y Naturales de la Universidad Nacional de San Juan. Los alumnos de este nivel tienen conocimientos de programación orientada a objetos y nociones elementales de dicho paradigma, de modelado ágil y conceptos avanzados en ingeniería de requisitos. La cátedra diseño de software se enfoca principalmente al Diseño Orientado a Objetos (DOO), brindando conceptos y conocimientos más desarrollados, en forma teórica y con aplicaciones prácticas, de todos los diagramas de modelado de software que provee UML (Unified Model Language), con el fin de comprender acabadamente el objetivo que se persigue con cada uno de ellos y en qué casos es conveniente o útil aplicarlos.

La práctica consiste en la aplicación de un proceso, guiado por Casos de Uso (CU), a una narrativa de un sistema a diseñar, en forma acotada pero con características que permita aplicar las distintas relaciones y diagramas que provee UML.

Los alumnos se centran en una única realidad a modelar por vez, y a medida que van adquiriendo conocimiento de la misma, van obteniendo, a través de sucesivas iteraciones, diagramas más detallados y completos.

Para la documentación de los CU se usa la plantilla CUPIDo v1.4 [1], generada anteriormente en el proyecto CUProSoft (UNSJ/E-822), que permite plasmar las necesidades, restricciones y condiciones de aplicación de los mismos.

El presente trabajo está ordenado de la siguiente manera, en la sección II se realiza un breve marco teórico, en la sección III se presenta el caso de estudio real. En la siguiente sección se exponen las conclusiones y trabajo a futuro y por último agradecimientos y referencias.

## 2 Marco Teórico

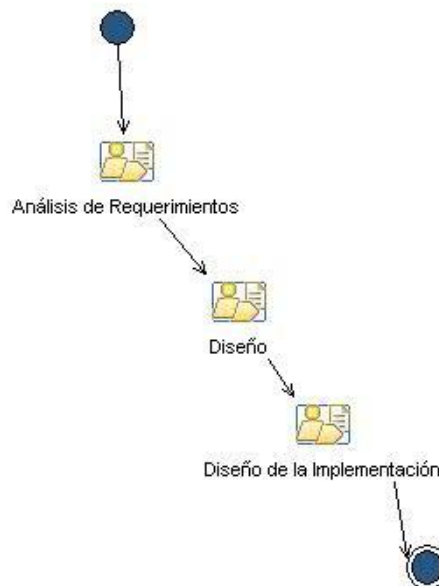
Para construir una aplicación de tamaño considerable, un componente fundamental en el proceso de desarrollo de software es un diseño consistente y bien documentado, se deben generar modelos del sistema que acompañen y guíen el proceso. Dentro de las herramientas de modelado de un sistema encontramos los diagramas de CU [2], que además de ser un recurso para capturar requisitos, ayudan en la delimitación del sistema y son un medio de comunicación entre usuarios, clientes y desarrolladores, ya que permiten describir de forma clara las funcionalidades de un sistema [3]. Sin embargo no siempre brindan la precisión necesaria si no se acompañan con una descripción textual u otro diagrama que lo detalle. Se necesita alguna descripción mínima de los CU [4].

Las plantillas de CU son registros estructurados que facilitan el proceso de documentación de un proyecto de software y la descripción de los requisitos funcionales. Es necesario que un CU sea bien descrito para que sea fácil de leer y entender. En base a esta necesidad y que no existe un estándar para documentar los CU, se identificaron y analizaron diferentes plantillas de autores reconocidos a nivel nacional e internacional [5, 6, 7, 8, 9, 10]. Ninguna se adaptaba por completo 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, 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. Por ello se elaboró una nueva plantilla que integró diferentes aspectos de las plantillas examinadas y se la denominó CUPIDo (Casos de Uso, Plantilla Integradora para Documentarlos). El detalle de la generación de la plantilla fue publicado en [1] y posteriormente validada y mejorada en distintas instancias [11,12], entre otras. Esta plantilla es simple y maneja los caminos alternativos en el mismo flujo de eventos, como subflujos. Existe un instructivo de apoyo para completar la plantilla CUPIDo.

Los Diagramas de Clases muestran cómo las clases son construidas; sus nombres, atributos y operaciones surgen del mismo contexto que se desea modelar y muestran cómo las clases se relacionan con otras por medio de diferentes asociaciones [5]. Describen gráficamente las especificaciones de las clases de software y de las interfaces en una aplicación [13]. El diagrama de clases es el gráfico representativo de un DOO.

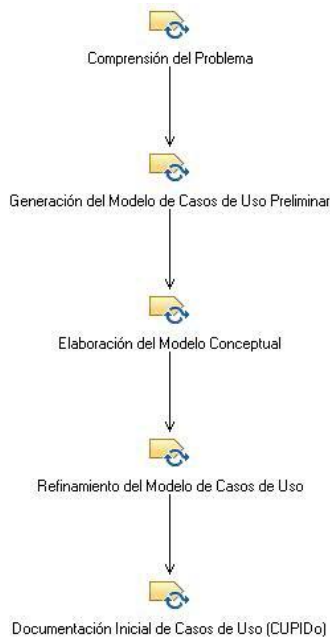
La conducción de la ejecución de las actividades prácticas, de la cátedra Diseño de Software, se basa en un modelo de proceso de enseñanza aprendizaje, publicado en [14]. Este modelo de proceso de enseñanza- aprendizaje se desarrolla como proceso de software, y los artefactos involucrados son utilizados para guiar al alumno en la resolución de un problema de desarrollo de software específico, valiéndose, para su especificación, del lenguaje de metamodelado de procesos SPEM 2.0 [15, 16] y para generar el modelado del proceso de software, la herramienta Eclipse Process Framework Composer (EPFC) [17,18].

Este proceso es iterativo, como se muestra en la Fig. 1.

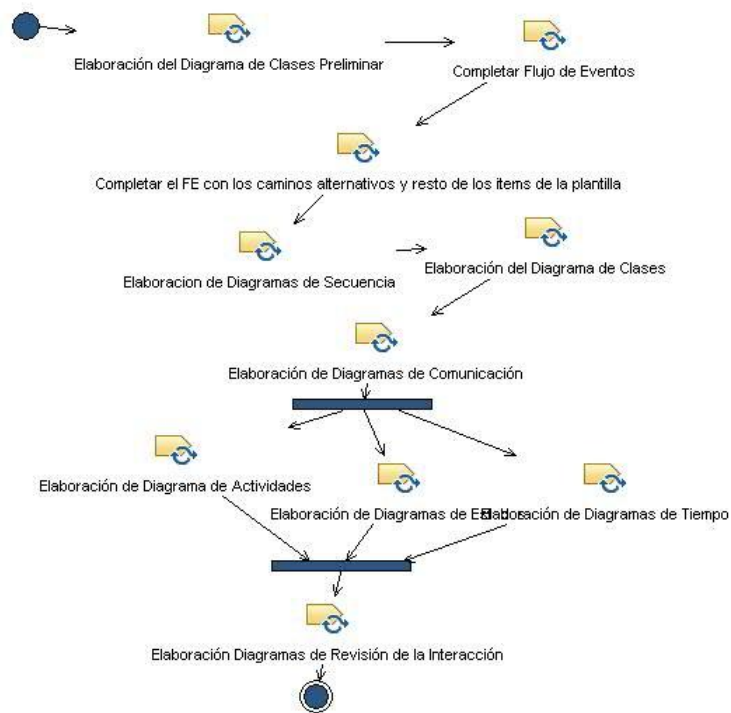


**Fig. 1.** Modelo de Proceso de enseñanza-aprendizaje. Fases [14]

Cada una de estas fases cuenta con etapas, que también se ejecutan en forma iterativa. Estas se pueden observar en las Fig. 2, Fig. 3 y Fig. 4.



**Fig. 2.** Tareas definidas para la fase Análisis de requerimientos [14]



**Fig. 3.** Tareas definidas para la fase Diseño [14].



Fig. 4. Tareas definidas para la fase Diseño de la Implementación [14].

### 3 Aplicación práctica. Caso de estudio

La práctica es elaborada como una sucesión de actividades que siguen los lineamientos planteados en el modelo anterior, estas actividades responden a un proceso iterativo e incremental. Se busca que el alumno, en cada actividad a desarrollar, logre obtener los subproductos (artefactos) de trabajo esperados en cada tarea y fase, adquiera el conocimiento para hacerlo y obtenga un diseño resultante adecuado a la realidad que se desea modelar.

Se plantea a continuación una aplicación práctica, con un ejemplo de estudio, y solo algunos artefactos de trabajo y diagramas resultantes, tanto de la fase de Análisis de requerimientos como de Diseño. No se presenta el ejemplo completo por limitaciones de la publicación.

#### 3.1 Narrativa

##### **Sistema de Gestión Provincial de Emergencias, para la policía de San Juan.**

Se desea desarrollar un sistema que permita llegar a una emergencia, producida a través de un llamado al 911, con los recursos adecuados y necesarios para la ocasión, de la forma más rápida posible. Esto se debe realizar a través de la gestión de recursos disponibles de acuerdo a la zona geográfica donde se produzca la emergencia, enviando el o los recursos adecuados al tipo de emergencia y a la zona donde se produce la misma.

Para lograr el objetivo se deben tener en cuenta las siguientes características:

- Considerar las zonas geográficas: la provincia tiene departamentos, cada departamento, tiene localidades, es decir, que una zona pertenece a una zona superior y

puede tener zonas inferiores. La zona de mayor jerarquía es la provincia de San Juan y es la única que cuenta con todos los recursos.

- Cada zona puede tener algunos recursos propios para atender las emergencias, por ejemplo: ambulancias, coches bomba, patrullas, rescatistas acuáticos, rescatistas de alta montaña, rescatistas vehiculares, profesionales en manejo de materiales peligrosos, etc.
- Para cada tipo de emergencia corresponden ciertos recursos, por ejemplo: si el tipo de emergencia es rescate en montaña, los recursos a enviar son rescatistas de alta montaña y ambulancia. En cambio, si se informa un incendio estructural (casa o edificio), los recursos son coche-bomba, ambulancias, patrulleros.
- Cuando ocurre una emergencia, de acuerdo a la zona del 'evento', y al tipo de emergencia, se envían los recursos adecuados al lugar, primero se buscan los recursos en la propia zona donde se produce la emergencia, si no los tiene busca en la zona inmediatamente superior en la jerarquía, hasta encontrarlo.
- El operador del sistema debe ser capaz de, al recibir la llamada al 911, registrar la emergencia, detalles de la misma, el tipo de emergencia y la zona geográfica donde se ha producido y también datos de quien realiza la llamada. Una vez que esto queda registrado, se buscan y muestran todos los recursos correspondientes y la zona a la que pertenecen. La emergencia queda en estado: 'recibida' y el operador se encarga de avisar, a los jefes de zonas correspondientes, del recurso que tienen que enviar al lugar de la emergencia.
- El operador es avisado por el jefe de la zona de la emergencia, que la misma ha finalizado. El sistema debe registrar que la emergencia queda en estado 'finalizada'.

Se desea, además de registrar la emergencia y la finalización de la misma, que los jefes de zona puedan consultar:

1. Dada una emergencia, mostrar todos los datos, los recursos asignados y la zona de cada uno.
2. Cantidad de emergencias ocurridas por zona, entre fechas, dando detalle del tipo de emergencia.
3. Tipo de emergencia más común, entre fechas, mostrando el detalle de las zonas donde ocurrieron.

Considerar que todos los datos (zonas, tipos de emergencia y tipos de recursos) y sus relaciones, al recibir la emergencia, se encuentran cargados en el sistema.

Narrativas de este estilo se plantean tanto en las clases prácticas de la materia, como en las evaluaciones.

### 3.2 Diagrama de Casos de Uso

En la Fig. 5 se muestra el Diagrama de Casos de Uso de una posible solución a la narrativa propuesta. Esta versión del diagrama surge de la segunda iteración en el modelado, al descubrir comportamiento común en los casos de uso.

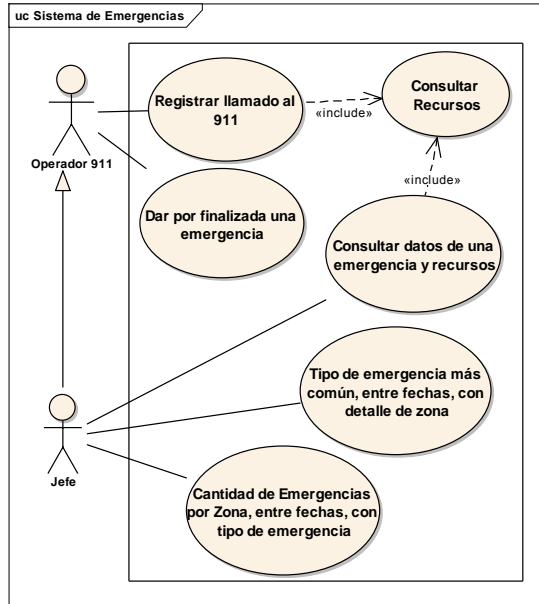


Fig. 5. Diagrama de Casos de Uso

### 3.3 Plantillas CUPIDO para documentar los casos de uso.

En las Fig. 6 y Fig. 7 se muestran las plantillas correspondientes a dos casos de uso.

Proyecto: Gestión Provincial de Emergencias	Paquete: Sistema de Emergencias
Caso de Uso: Consultar Recursos	Nombre del Documento digital: Consultar Recursos.doc
Estado: finalizado	Versión: 1.0   Fecha revisión: 02/07/2013   Hoja: 1/1

Plantilla CUPIDO v1.4

Hoja de Revisión			
Fecha	Versión	Descripción	Autor
02/07/2013	1.0	Creación y descripción de plantilla	MIL

**Plantilla de Caso de Uso "CUPIDO"**

Nº	Campo	Descripción del campo
1	Actor	Operador 911
2	Tipo de Caso de Uso	Incluido
3	Breve Descripción	Muestra los recursos de cada emergencia y la zona de donde provienen
4	Precondiciones	La emergencia ya se encuentra seleccionada
5	Flujo de Eventos	El sistema busca el tipo de emergencia de la emergencia El sistema muestra tipo de la emergencia El sistema busca la zona de la emergencia El sistema muestra la zona de la emergencia [mientras existan recursos para el tipo de emergencia] El sistema busca recurso del tipo de emergencia El sistema muestra el recurso El sistema busca el recurso en la zona de emergencia [si no existe el recurso en la zona] El sistema busca recurso en zona superior hasta encontrarlo El sistema muestra la zona del recurso y datos de la misma
6	Postcondiciones	
7	Extensión (extend)	
8	Inclusión (include)	
9	Requisitos asociados	R02: Mostrar los recursos que corresponden a la emergencia
10	Consideraciones / Observaciones	
11	Frecuencia de Uso	30 por día

Fig. 6. Plantilla de Descripción del CU Consultar Recursos.

Proyecto: Gestión Provincial de Emergencias	Paquete: Sistema de Emergencias
Caso de Uso: Registrar llamado al 911	Nombre del Documento digital: Registrar llamado.doc
Estado: finalizado	Versión: 2.0 Fecha revisión: 02/07/2013 Hoja: 1/1

Plantilla CUPIDO v1.4

## Hoja de Revisión

Fecha	Versión	Descripción	Autor
30/06/2013	1.0	Creación y descripción de plantilla	MIL
02/07/2013	2.0	Corrección del FE y agregado de CU incluido	MIL

## Plantilla de Caso de Uso "CUPIDO"



Nº	Campo	Descripción del campo
1	Actor	Operador 911
2	Tipo de Caso de Uso	Base
3	Breve Descripción	Se recepciona y registra la llamada de emergencia y se muestra todos los recursos que deben participar y la zona de donde provienen
4	Precondiciones	Ocurrencia de llamada telefónica. Operador, Zona, Tipo de emergencia y Tipo de recurso ya existen en el sistema
5	Flujo de Eventos	El operador recepciona la llamada El operador solicita registrar la llamada al sistema El sistema solicita datos de la llamada (teléfono, datos de quien llama, zona de emergencia, tipo de emergencia) El operador ingresa los datos al sistema El operador solicita buscar zona de emergencia en el sistema El sistema busca zona de emergencia El sistema asigna la zona de emergencia al registro de la llamada El operador solicita buscar el tipo de emergencia en el sistema El sistema busca el tipo de emergencia El sistema asigna el tipo de emergencia al registro de llamada El sistema indica que la emergencia está 'recibida' El sistema llama al CU Consultar Recursos
6	Postcondiciones	Llamada registrada en el sistema
7	Extensión (extend)	
8	Inclusión (include)	CU Consultar Recursos
9	Requisitos asociados	R01. Registrar llamada de emergencia con todos sus datos. R02. Mostrar los recursos que corresponden a la emergencia
10	Consideraciones / Observaciones	El Operador debe llamar a los Jefes de zona correspondiente, para que envíen los recursos a la zona de emergencia.
11	Frecuencia de Uso	30 por día

Fig. 7. Plantilla de Descripción del CU Registrar llamado al 911.

En la Fig. 6 se especifica el CU incluido que es invocado desde el CU Registrar llamado al 911, detallado en la Fig. 7. En los flujos de eventos se definen las interacciones entre actor y sistema, y las acciones que realiza el sistema para responder a la solicitud del actor.

### 3.4 Diagrama de Clases

En la Fig. 8 se muestra el diagrama de clases obtenido a partir del modelo de casos de uso y de la ejecución en forma iterativa de todas las actividades y tareas del proceso de enseñanza-aprendizaje planteado, específicamente para las fases de análisis de requerimientos y diseño.

Las operaciones no se muestran por limitaciones de espacio de la publicación.



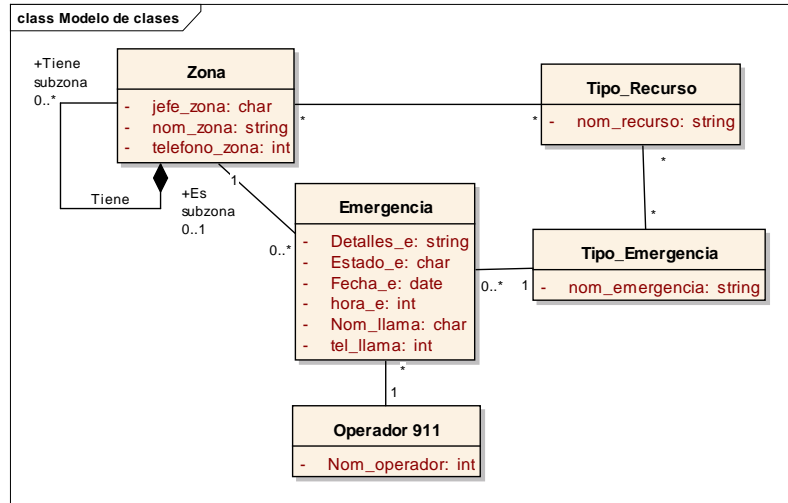


Fig. 8. Diagrama de Clases.

#### 4 Conclusiones y trabajo a futuro

Los alumnos, al finalizar todas las actividades desarrolladas en el marco del proceso de enseñanza-aprendizaje, como un proceso de software, no sólo logran aprender y comprender el uso y aplicación de cada modelo y diagrama de UML útiles para el análisis y diseño de un sistema, sino que llegan a un conocimiento acabado y a una vista completa del sistema a implementar. Esta metodología también les facilita la conceptualización de los contenidos teóricos, reafirmando la relación teoría-práctica.

Si bien se observa cómo los alumnos van adquiriendo conocimiento y experticia al ir elaborando las actividades según el proceso planteado y cuya satisfacción se manifiesta en encuestas al finalizar el curso, aún queda como trabajo a futuro poder medir la calidad del proceso de enseñanza-aprendizaje propuesto, a través de algún instrumento, y en base a la información obtenida introducir mejoras en el mismo.

Con respecto a la plantilla CUPIDo, actualmente se está trabajando en una propuesta de sintaxis para estructurar el flujo de eventos denominada UCEFlow [19].

#### 5 Agradecimientos

A las integrantes del equipo de la cátedra “Diseño de Software”: Lic. María Claudia Gómez, a cargo de la teoría y a la Lic. Laura Nidia Aballay, colaboradora en las actividades prácticas desde hace 3 años. A los alumnos, que año tras año nos presentan nuevos desafíos, lo cual nos motiva a seguir buscando, estudiando y aprendiendo.

Este trabajo se presenta en el marco del proyecto: “Modelo de Casos de Uso. Línea base para el proceso de desarrollo de software”. Aprobado y Financiado por la UNSJ, período 2014-2015.

## Referencias

1. Lund, María Inés, Ferrarini, Cintia, Aballay, Laura Nidia, Romagnano, María Gema, y Meni, Ernesto, «CUPIDo - Plantilla para Documentar Casos de Uso», presentado en V Congreso de Tecnología en Educación y Educación en Tecnología, El Calafate, Santa Cruz, Argentina, 2010.
2. I. Jacobson, Magnus Christerson, P. Jonsson, y Gunnar Overgaard, *Object-Oriented Software Engineering: A Use Case Driven Approach*. Addison-Wesley, 1998.
3. I. Jacobson, G. Booch, y J. Rumbaugh, «Chapter 1: The Unified Process: Use-Case Driven, Architecture-Centric, Iterative, and Incremental», en *The Unified Software Development Process*, Addison-Wesley Professional, 1999, p. 5.
4. M. I. Lund, C. Ferrarini, E. G. Ormeño, y S. Zapata, «Indicadores para la Evaluación de Diseños de Sistemas a través del Modelo de Casos de Uso en la Educación Superior», en *VI International Conference on Engineering and Computer Education – ICECE'2009*, Buenos Aires, Argentina, 2009.
5. D. Kulak y E. Guiney, *Use Cases: Requirements in Context*, 2.<sup>a</sup> ed. Addison-Wesley Professional, 2008.
6. K. E. Wiegers, *More about software requirements: thorny issues and practical advice*. Redmond, WA: Microsoft Press, 2006.
7. Pinciroli, F. y Sabio, G., «Casos de Uso». Universidad de Congreso - Mendoza, 2005.
8. Rubin, David M., «Use of Use Cases. Methodologies and Practices. White Paper». Softstar Research, Inc, 1998.
9. D. Leffingwell y D. Widrig, *Managing software requirements: a use case approach*. Boston: Addison-Wesley, 2003.
10. C. Rolland, C. Ben Achour, C. Cauvet, J. Ralyté, A. Sutcliffe, N. Maiden, M. Jarke, P. Haumer, K. Pohl, E. Dubois, y P. Heymans, «A proposal for a scenario classification framework», *Requir. Eng.*, vol. 3, n.º 1, pp. 23-47, mar. 1998.
11. M. I. Lund, L. Aballay, E. Torres, M. Herrera, y E. Ormeño, «Validación de Usabilidad de una plantilla para documentar Casos de Uso – Estudio Exploratorio», presentado en XVII Congreso Argentino de Ciencia de la Computación, Buenos Aires - Argentina, 2011.
12. L. N. Aballay, M. I. Lund, E. G. Ormeño, S. Cruz Introini, C. Marcuzzi, y G. Jofré, «Plantilla CUPIDo: automatización y avances», presentado en XV Workshop de Investigadores en Ciencias de la Computación, 2013.
13. C. Larman, *UML y patrones: introducción al análisis y diseño orientado a objetos*. Félix Varela, 2004.
14. M. I. Lund, L. N. Aballay, M. C. Gomez, y E. G. Ormeño, «Propuesta de un Proceso de Software de Enseñanza-Aprendizaje para la asignatura Diseño de Software», *Rev. Ibero-am. Technol. En Educ. Educ. En Technol.*, vol. 12, n.º especial n.º 12, pp. 44-52, abr. 2014.
15. OMG, «Software & Systems Process Engineering Metamodel specification (SPEM) Version 2.0». abr-2008.
16. Francisco Ruiz, Javier Verdugo, «Guía de Uso de SPEM 2 con EPF Composer». Universidad de Castilla La Mancha, 01-abr-2008.
17. ECLIPSE, *Eclipse Process Framework Project (EPF)*. ECLIPSE, 2012.
18. Haumer, Peter, «Eclipse Process Framework Composer». IBM, abr-2007.
19. L. Aballay, S. C. Introini, M. I. Lund, y E. Ormeno, «UCEFlow: A syntax proposed to structuring the event flow of use cases», en *Computing Colombian Conference (8CCC), 2013 8th*, 2013, pp. 1-6