

# La enseñanza de construcción de soluciones genéricas en un curso de Análisis y Diseño de Sistemas

Marcela Daniele Daniel Romero

Departamento de Computación – Facultad de Ciencias Exactas, Fco-Qcas y Naturales

Universidad Nacional de Río Cuarto

[marcela@dc.exa.unrc.edu.ar](mailto:marcela@dc.exa.unrc.edu.ar) - [danielomarromero@yahoo.com.ar](mailto:danielomarromero@yahoo.com.ar)

## Resumen

El Proceso Unificado es una metodología de desarrollo de software que propone la construcción de sistemas con un proceso iterativo e incremental, centrado en la arquitectura y dirigido por casos de uso. En cada una de las etapas del proceso se consigue definir y refinar los casos de uso o funcionalidades del sistema, hasta lograr un producto ingenieril de calidad. Desde el año 1999, esta metodología forma parte de los contenidos de la asignatura Análisis y Diseño de Sistemas del tercer año de las carreras de Computación en la Universidad Nacional de Río Cuarto, y es la metodología más usada en el proyecto final de los alumnos que optan al título de Analistas en Computación.

Este trabajo propone plantillas genéricas para la descripción, análisis y diseño de casos de uso en los problemas de inserción, eliminación, modificación y búsqueda de elemento, cuyo principal propósito es que los alumnos aprendan a construir modelos que si bien representan una solución para el problema planteado, dichos modelos resulten en soluciones genéricas que puedan ser instanciadas por el mismo problema real planteado en otro contexto. Las plantillas propuestas utilizan UML como herramienta de modelado estándar.

**Palabras Claves:** proceso unificado, casos de uso, etapas, modelos, plantillas genéricas

## Introducción

Las metodologías de desarrollo de software y las técnicas de modelado fueron creadas para simplificar la complejidad del desarrollo de sistemas, y por lo tanto hay que utilizarlas en cualquier proceso de producción de software. Tom DeMarco [14] 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 en el proyecto, cliente, usuarios y desarrolladores. La mayoría de los métodos de desarrollo de software usados en la actualidad adoptaron la filosofía propuesta por este autor, con las particularidades propuestas por cada uno. Un modelo es una abstracción del sistema, especificándolo desde una vista determinada y un cierto nivel de abstracción. Los modelos que forman un sistema están siempre relacionados entre ellos.

El Proceso Unificado [4] es un proceso de desarrollo de software que define *quién* está haciendo *qué*, *cuándo*, y *cómo* para construir o mejorar un producto de software. Esta definido por tres características principales: dirigido por casos de uso, centrado en la arquitectura, iterativo e incremental.

En la primera etapa denominada Captura de Requerimientos, propone como principal artefacto crear una lista de casos de uso y actores del sistema. Un caso de uso es una funcionalidad del sistema que otorga valor agregado a algún actor o usuario del mismo. Luego cada caso de uso requiere de una detallada descripción donde se indica: una precondición, una poscondición, un flujo de eventos principal y el flujo de eventos alternativo.

En las siguientes etapas denominadas Análisis y Diseño, el Proceso Unificado tiene el propósito de obtener modelos del sistema más precisos y lograr una arquitectura estable, que permita realizar una correcta implementación del sistema traduciendo estos modelos a un lenguaje de programación.

La metodología propone que cada modelo construido en una etapa lógicamente debe basarse en los modelos construidos en la etapa anterior, y de esta manera asegura que el sistema obtenido responde a una construcción evolutiva e incremental.

En la etapa de Análisis se detectan las clases de análisis a partir de las descripciones de cada caso de uso y se construye un Diagrama de Clases UML [3][16] de análisis para cada uno. Además, se definen los escenarios más evidentes de cada caso de uso y se modelan con Diagramas de Interacción UML permitiendo identificar responsabilidades de las clases. En la etapa de Diseño, cada clase obtenida en el análisis es transformada a una o más clases de diseño, y para cada clase de diseño se definen de manera completa y precisa sus atributos y métodos. Esta es la base o arquitectura estable requerida para avanzar hacia la implementación del sistema.

Desde el año 1999, el Proceso Unificado forma parte de los contenidos de la asignatura Análisis y Diseño de Sistemas incluida en el plan de estudios del tercer año de las carreras de Analista en Computación, Profesorado y Licenciatura en Ciencias de la Computación en la Universidad Nacional de Río Cuarto. Además, es la metodología más usada en el proyecto final de los alumnos que desarrollan un sistema. En el año 2004, los docentes de la asignatura desarrollaron un proyecto innovador denominado “Definición y uso de Plantillas Genéricas para la descripción de Casos de Uso” [19], cuyo principal objetivo se centró en encontrar una manera de enseñar la metodología que permita al alumno plantear soluciones genéricas a problemas similares. Este proyecto cubrió la etapa de Captura de Requerimientos, donde se propusieron las *plantillas genéricas para la descripción de casos de uso* en los problemas de inserción, eliminación, modificación y búsqueda de elemento. Por los asombrosos resultados obtenidos y la unánime aceptación tanto de docentes como de alumnos, de la asignatura en cuestión y de asignaturas posteriores que se vieron afectadas, tales como Ingeniería de Software y Proyecto, en el año 2005 se continuó trabajando con la definición de plantillas genéricas para cubrir las siguientes etapas del proceso de desarrollo denominadas Análisis y Diseño.

### **Fundamentación y Objetivos**

La definición y el uso de plantillas genéricas para la descripción de casos de uso es una importante mejora en el accionar de los alumnos cuando requieren plantear soluciones a problemas que requieren de igual tratamiento que otros ya estudiados y resueltos. Además, a través de estas plantillas los alumnos despiertan la necesidad de definir sus propias plantillas para dar solución a otros problemas que se le presenten de manera reiterada. El propósito es que los alumnos aprendan a construir modelos que si bien representan una solución para el problema planteado, dichos modelos resulten en soluciones genéricas que puedan ser instanciadas por el mismo problema real planteado en otro contexto.

Dado que la descripción de casos de uso se torna repetitiva y tediosa, y que además, esa repetición no aporta sino que muy por el contrario, resulta en una extensa documentación que dificulta la lectura y el entendimiento de los casos de uso, y para proyectos de buen tamaño esta característica es más importante aún.

También es importante de que las plantillas para la descripción de casos de uso permitan a los alumnos darse cuenta de que en problemas tales como: agregar, borrar, modificar o buscar un elemento no interesa cual fuere el elemento específico sino que basta con tener una plantilla que de la solución a cualquier problema de dicha naturaleza y simplemente reemplazar los atributos formales por los valores reales que se definen y plantean para cada caso de uso.

Los objetivos de la propuesta son:

- Reducir la cantidad de documentación repetida que deben desarrollar los alumnos cuando describen los casos de uso en el marco de la primera etapa de la metodología de desarrollo Proceso Unificado.

- Disminuir la dificultad que se genera para comprender los diversos modelos de solución planteados para resolver el mismo tipo de problemas.
- Utilizar horas de clase mal invertidas en la descripción de todos los casos de uso para introducir un nuevo tema que es Patrones de Diseño.
- Ayudar al alumno a plantear modelos genéricos para solucionar problemas con idénticas o similares características.
- Enseñar al alumno a instanciar problemas reales en un modelo general propuesto.

## Plantillas Genéricas para la descripción de Casos de Uso

En el año 2004, en la asignatura Análisis y Diseño de Sistemas se trabajó en un proyecto[19] para el mejoramiento de la enseñanza de grado, con el objeto de encontrar una manera de aplicar el Proceso Unificado como una metodología que permita plantear soluciones genéricas a problemas similares, y que puede ser usada tanto por profesionales para el desarrollo de sus productos de software, como por alumnos que desarrollan productos de software como trabajo final.

El mencionado proyecto cubre la primera etapa, Captura de Requerimientos, donde se proponen los esquemas o *plantillas genéricas para la descripción de casos de uso* y que pueden ser aplicadas a problemas de: *inserción, eliminación, modificación y búsqueda de elemento*. Cada plantilla define al caso de uso con un nombre, una precondición, una poscondición, los actores relacionados al caso de uso, el flujo de eventos principal y el flujo de eventos alternativos. Entre los símbolos <<...>> se especifican los parámetros formales. Cuando la plantilla es instanciada con un caso de uso específico los parámetros formales son reemplazados por los valores reales.

A modo de ejemplo se muestra la plantilla para la *inserción de elemento* en la figura 1, y una forma de instanciación de dicha plantilla con un caso real se muestra en la figura 2 para el caso de uso *Inserción de Cliente*.

| <b>Plantilla 1: Inserción de &lt;&lt;elemento&gt;&gt;</b>                                                                                                                                                                                                                                                                                                                               |                                                                                 |
|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------|
| <b>Parámetros:</b><br><b>Elemento:</b> ítem a ser insertado. Un ítem está compuesto por atributos clave y atributos.<br><b>Atributos clave:</b> las propiedades que identifican al elemento unívocamente.<br><b>Atributos:</b> propiedades que componen el elemento.<br><b>Reglas de negocio</b> ( $r_1, \dots, r_n$ ): indicar las reglas que deben ser verificadas en el caso de uso. |                                                                                 |
| <b>Nombre:</b> Inserción de <<elemento>>.                                                                                                                                                                                                                                                                                                                                               |                                                                                 |
| <b>Pre-condición:</b> existe un <<elemento>> a ser ingresado.                                                                                                                                                                                                                                                                                                                           |                                                                                 |
| <b>Post-condición:</b> <<elemento>> queda registrado en el sistema, o <<elemento>> ya estaba registrado.                                                                                                                                                                                                                                                                                |                                                                                 |
| <b>Descripción:</b> realiza la inserción de un <<elemento>>, controlando la existencia del elemento en el sistema y el cumplimiento de las reglas del negocio ( $r_1, \dots, r_n$ ) asociadas al <<elemento>>.                                                                                                                                                                          |                                                                                 |
| <b>Actor:</b> nombre de cada uno de los actores que interactúan con el caso de uso.                                                                                                                                                                                                                                                                                                     |                                                                                 |
| <b>FLUJO DE EVENTOS PRINCIPAL</b>                                                                                                                                                                                                                                                                                                                                                       |                                                                                 |
| ACTOR                                                                                                                                                                                                                                                                                                                                                                                   | SISTEMA                                                                         |
| 1. Ingresar <<atributos clave>> del <<elemento>>.                                                                                                                                                                                                                                                                                                                                       | 2. Verifica existencia por <<atributos clave>>.                                 |
| 3. Ingresar el resto de los <<atributos>> del <<elemento>>.                                                                                                                                                                                                                                                                                                                             | 4. Verifica corrección de <<atributos>> ingresados.                             |
|                                                                                                                                                                                                                                                                                                                                                                                         | 5. Verifica reglas de negocio << $r_1, \dots, r_n$ >> asociadas al caso de uso. |
|                                                                                                                                                                                                                                                                                                                                                                                         | 6. Realiza el alta del <<elemento>>                                             |
| <b>FLUJO DE EVENTOS ALTERNATIVO</b>                                                                                                                                                                                                                                                                                                                                                     |                                                                                 |
| 2.1. El sistema informa de la existencia del <<elemento>> identificado con <<atributos clave>>.                                                                                                                                                                                                                                                                                         |                                                                                 |
| 4.1. El sistema informa que al menos uno de los <<atributos>> ingresado no es correcto <sup>1</sup> .                                                                                                                                                                                                                                                                                   |                                                                                 |
| 5.1. El sistema informa las reglas $r_i$ que no se verifican (con $1 \leq i \leq n$ ).                                                                                                                                                                                                                                                                                                  |                                                                                 |

Figura 1: Plantilla Genérica para la Inserción de un elemento

| Caso de Uso: <b>Inserción de Cliente</b>                  |       |                                                                                                                |                           |
|-----------------------------------------------------------|-------|----------------------------------------------------------------------------------------------------------------|---------------------------|
| Instancia: <b>Plantilla 1</b> (Inserción de <<elemento>>) |       |                                                                                                                |                           |
| Elemento: Cliente.                                        |       |                                                                                                                |                           |
| ATRIBUTOS                                                 | CLAVE | VERIFICACION                                                                                                   | ACCION                    |
| CUIT                                                      | SI    | Debe ser NO NULO. El formato del cuit debe ser: 2 caracteres, un guión, 8 caracteres, un guión, 1 caracter.    |                           |
| NOMBRE                                                    |       | Debe ser NO NULO.                                                                                              |                           |
| CONDICION DE IVA                                          |       | El cliente puede ser: Responsable Inscripto, Responsable No Inscripto, Monotributo, Exento o Consumidor Final. |                           |
| MUTUAL                                                    |       | La mutual debe seleccionarse de las previamente cargadas en el sistema.                                        | Include(Buscar Mutual)    |
| CUENTA                                                    |       | Debe ser NO NULO.<br>Se debe crear la cuenta del cliente.                                                      | Include(Inserción Cuenta) |

Regla r1: Cada cliente debe tener una cuenta.

Figura 2: Inserción de Cliente usando la Plantilla

## Plantillas Genéricas para Análisis y Diseño

En las etapas de Análisis y Diseño, el Proceso Unificado tiene el propósito de obtener modelos del sistema más precisos y lograr una arquitectura estable que permita realizar una correcta implementación del sistema traduciendo estos modelos a un lenguaje de programación.

En el año 2005, se continuó trabajando en la evolución de los esquemas desarrollados para la primera etapa y se definieron las plantillas correspondientes para las etapas de Análisis y Diseño [20,21]. El principal propósito de la etapa de Análisis es realizar cada caso de uso descrito en la etapa anterior, definir las principales clases, sus atributos y responsabilidades, y se modelan con un diagrama de clases UML y diagramas de colaboración UML por cada escenario planteado.

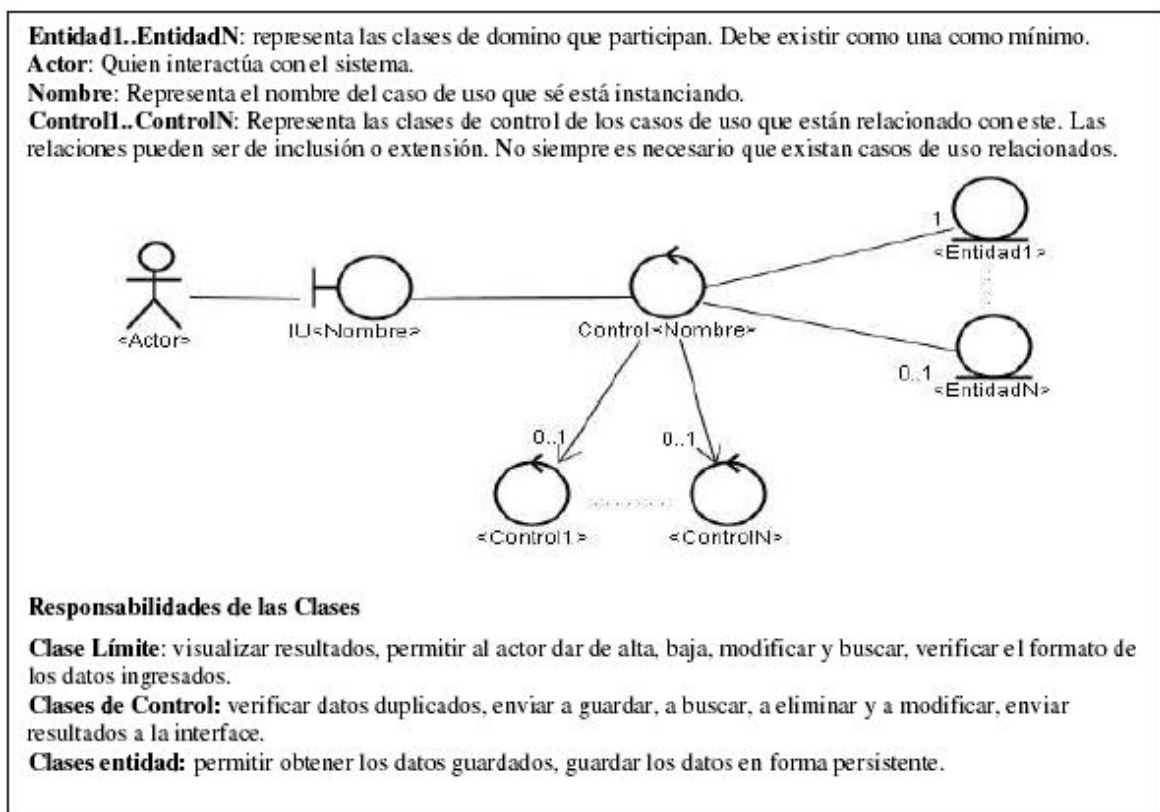


Figura 3: Plantilla Genérica para el Diagrama de Clases de Análisis de un Caso de Uso



En esta propuesta se define una única plantilla para modelar el análisis de los casos de uso de inserción, modificación, eliminación y búsqueda de elemento, y se muestra en el diagrama de clases de la Figura 3. El diagrama indica que según cada caso particular, la clase de control del caso de uso modelado puede poseer relación con 1 o más entidades diferentes y con ninguna, 1 o más clases de control de otros casos de uso. Además, en la figura 4 se muestra un diagrama de colaboración UML que refleja la plantilla genérica definida para el caso de uso *Inserción de elemento* y para un escenario específico (el resto de las plantillas están definidas en [20]). El diagrama contiene los objetos participantes y los mensajes que se envían entre los mismos para llevar a cabo la inserción de un elemento en el escenario descrito.

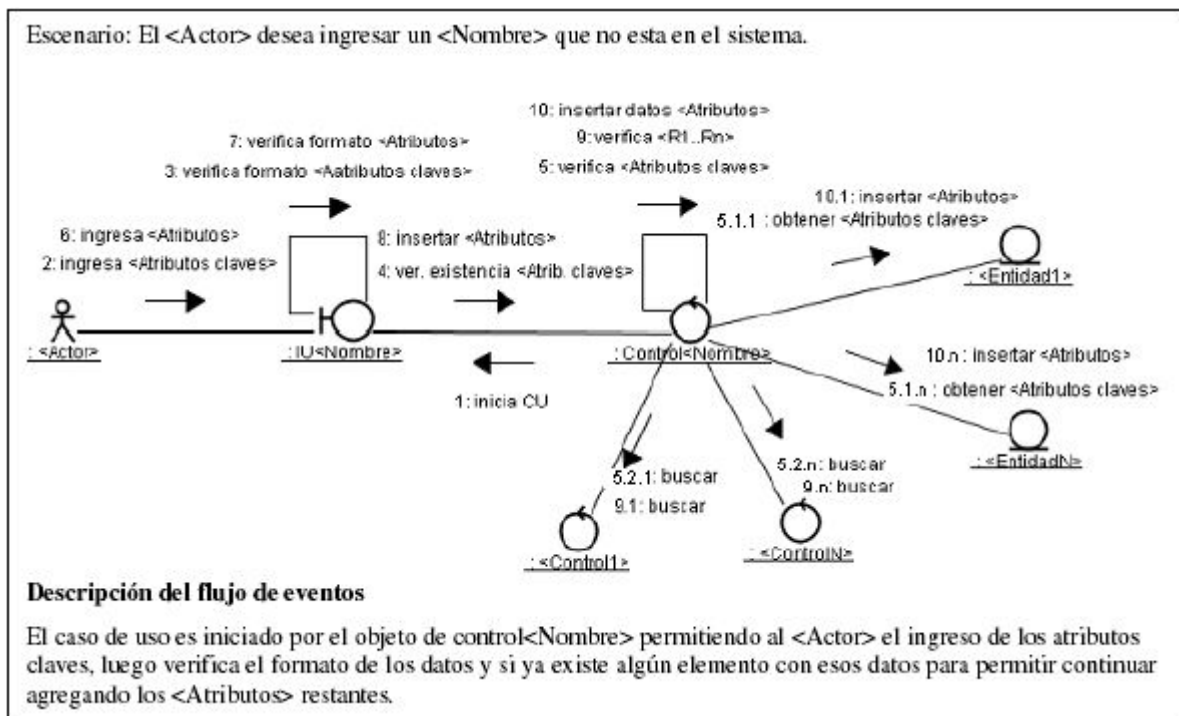


Figura 4: Plantilla Genérica para el Diagrama de Colaboración de la Inserción de elemento

Para el desarrollo del diseño de cada caso de uso, cada clase obtenida en la etapa anterior es transformada a una o más clases de diseño definidas de manera completa y precisa con atributos y métodos, formando la arquitectura estable del sistema, la base sólida necesaria para la implementación del sistema y puesta en marcha.

En este trabajo se propone una plantilla genérica (Figura 5) para modelar las clases del diseño para cada caso de uso del tipo de problemas de inserción, modificación, eliminación y búsqueda de elemento. Se utilizan el patrón Mediator [17] para la implementación del control de interfaces gráficas, y el patrón Registry [18] para el manejo de clases persistentes. El modelado de los mensajes enviados entre los objetos en un escenario particular son también representados de manera genérica en la Figura 6, la que muestra en un Diagrama de Secuencia UML la plantilla que modela en forma genérica un escenario planteado para la inserción de elemento (el resto de las plantillas están definidas en [20]).

## Implementación de la propuesta

Para la implementación de este proyecto pedagógico en el aula, se detalla un resumen de las actividades realizadas en el orden cronológico dado en el aula:

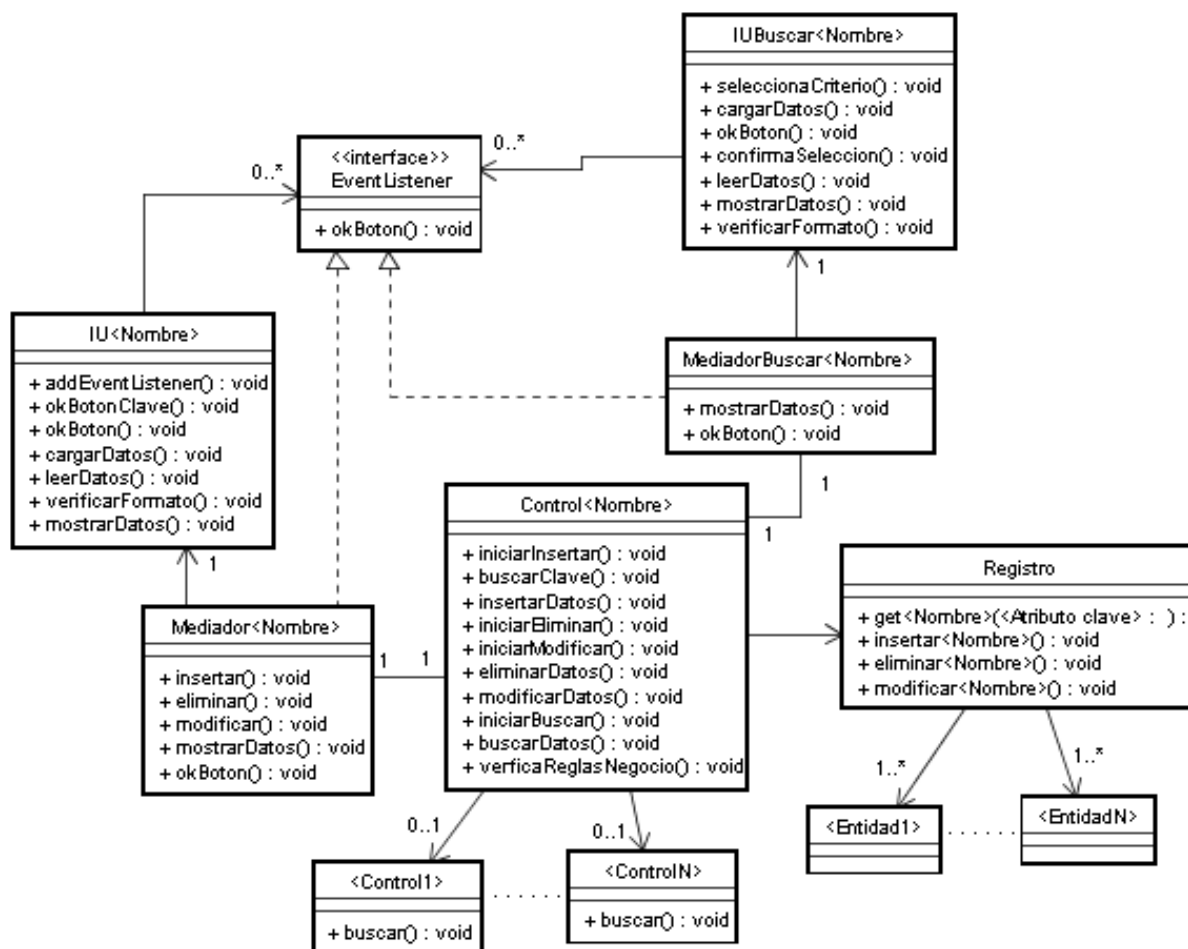


Figura 5: Plantilla Genérica para el Diagrama de Clases de diseño de un Caso de Uso

- Introducir la primera etapa del Proceso Unificado: Captura de Requerimientos, Descripción de Casos de uso. Definición y Uso de Plantillas genéricas. Y resolver una guía práctica que consiste en definir casos de uso y actores a partir del modelo de negocio obtenido, realizar un diagrama de casos de uso y describir cada uno. Detectar los casos de uso que tienen idéntica descripción por pertenecer al mismo tipo de problemas e instanciar las plantillas genéricas para la descripción de casos de uso.
  - Introducir la segunda etapa del Proceso Unificado: Análisis. Definir clases de análisis a partir de la etapa previa. Modelar con Diagrama de Clases y Diagramas de Colaboración UML. Definir y usar las plantillas el análisis. Resolver la guía de ejercicios donde el alumno debe obtener las clases de análisis a partir de la descripción de casos de uso de la etapa previa. Diseñar las clases de análisis con un Diagrama de Clases UML. Plantear escenarios y modelar con Diagramas de Colaboración de UML e instanciar las plantillas de análisis para los casos de uso genéricos.
  - Desarrollar la tercera etapa del Proceso Unificado: Diseño con Patrones de Diseño. Definir clases de diseño a partir de la etapa previa. Modelar con Diagrama de Clases y Diagramas de Secuencia UML. Presentar y usar las plantillas genéricas para el diseño en los casos de uso que sea posible. Resolver la guía de ejercicios donde el alumno debe obtener las clases de diseño a partir de las clases de análisis. Diseñar las clases de diseño con un Diagrama de Clases UML. Tomar los escenarios planteados en la etapa previa y modelar con Diagramas de Secuencia UML. Utilizar las plantillas de diseño para los casos de uso genéricos.
- Además, el equipo de docentes participantes realiza las siguientes actividades:
- Analizar los resultados obtenidos con la aplicación de las plantillas genéricas en años anteriores.

- Revisar y refinar las plantillas genéricas propuestas.
- Revisar y modificar las guías de ejercicios prácticos correspondientes.
- Estudiar el impacto en los agentes, en asignaturas posteriores y en el futuro accionar profesional de los de los alumnos.
- Revisar y reajustar la forma de implementación de la propuesta en años anteriores.

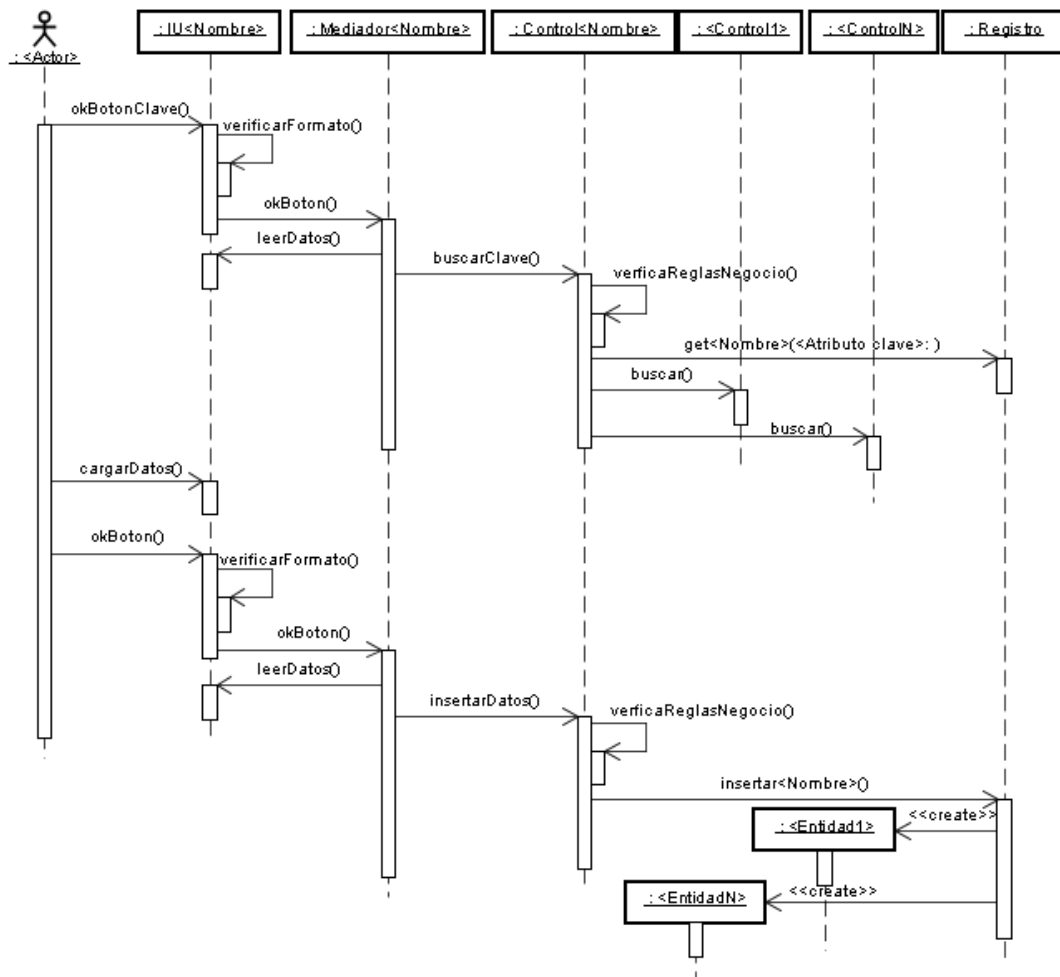


Figura 6: Plantilla Genérica para el Diagrama de Secuencia de la Inserción de elemento

## Evaluación de la propuesta

Para la evaluación de la propuesta se tiene en cuenta:

- Valorar la repercusión del uso de plantillas como una nueva forma de describir los casos de uso.
- Constatar que los objetivos de la asignatura se cumplen aún con un importante cambio en el proceso de aprendizaje.
- Apreciar las ventajas que propone esta innovación tanto en asignaturas posteriores como en el futuro accionar profesional de los alumnos.
- Recabar las experiencias y dificultades planteadas por el equipo de docentes participantes durante la implementación de la propuesta.
- Analizar el rendimiento de los alumnos en el examen parcial, puntualmente en el ejercicio que hace referencia a la descripción de casos de uso instanciando alguna de las plantillas definidas, haciendo un análisis de los errores que comenten y contrastándolos con los cometidos en años anteriores, donde no se usaron plantillas.
- Recabar la opinión de los alumnos respecto a la definición y uso de plantillas.

- Analizar la habilidad adquirida por los alumnos para aplicar las plantillas correctamente.
- Analizar la habilidad adquirida por los alumnos para definir sus propias plantillas en otros problemas que se presenten con similares características.

Además sirven como indicadores para realizar la evaluación:

- Grado de satisfacción de docentes y directivos con respecto a la propuesta realizada en torno a la descripción de los casos de uso.
- Cantidad y tipo de dificultades detectadas a medida que los docentes aplican la propuesta.
- Analizar el uso de las plantillas en las asignaturas correlativas (Ingeniería de Software y Proyecto) y resumir los resultados obtenidos.
- Recabar la opinión de nuestros graduados respecto de la aplicación de las plantillas para especificar los casos de uso en el desarrollo de sistemas.

## Resultados Obtenidos y Conclusiones

Las plantillas propuestas permiten que los alumnos razonen la homogeneidad en el comportamiento de problemas de inserción, eliminación, modificación o búsqueda de elementos. Y que sin interesar cual fuere el elemento específico, las plantillas genéricas dan solución a cualquier problema de dicha naturaleza, simplemente reemplazando los atributos formales por los valores reales definidos para cada caso de uso. Además, el uso de estas plantillas ayuda a pensar qué otros problemas presentan un comportamiento similar y podrían ser tratados con éstas u otras soluciones planteadas de manera genérica.

La aplicación de las plantillas genéricas para la descripción, análisis y diseño de casos de uso, presenta importantes beneficios para las asignaturas involucradas, los alumnos afectados, los docentes participantes de dichas asignaturas y en el futuro accionar profesional de alumnos graduados. Por los resultados obtenidos, presentados en un informe detallado que puede consultarse en [22], pudo concluirse que los alumnos respondieron con absoluta convicción en la adopción de dichas plantillas, tanto en la asignatura planteada como en asignaturas posteriores donde se les presentaron posibles situaciones de aplicación y ellos mismos pudieron reconocer la posibilidad de aplicar las plantillas aprendidas. Además, reconocieron que las plantillas les sirven como un mecanismo para construir sus propias plantillas genéricas de solución, en otros tipos de problemas que también demuestran un comportamiento reiterado.

Una de las principales ventajas de las plantillas, y es un motivo importante de su amplia aceptación, que además de resolver el problema de manera correcta y homogénea para los de su familia, permite reducir considerablemente el volumen de documentación generada al final del desarrollo de un software, evitando la repetición de modelos que resuelven el mismo tipo de problema. Además, permiten un considerable ahorro de tiempo permitiendo incorporar otros temas de interés en la asignatura.

Sabemos que la ingeniería busca la calidad, y por lo tanto la ingeniería de software es la producción de software de calidad. En este sentido, una de las características a tener en cuenta es la *facilidad de uso* del software. En torno a esto, las plantillas producen un beneficio importante en el usuario final de los sistemas, ya que en general la apariencia de las funcionalidades con comportamientos similares, es muy homogénea y esto hace que el sistema sea más fácil de usar y de rápida adaptación.

Otra de las características a tener en cuenta en el desarrollo de software es la *facilidad de mantenimiento*. El mantenimiento del software se asocia a modificaciones requeridas por cambios en los requerimientos del usuario, en el formato de los datos, cambios en la documentación, cambios en el hardware o por factores del mundo externo que inciden en el software, la corrección de errores y/o defectos. Las plantillas genéricas facilitan la tarea del desarrollador en cuanto al mantenimiento del software, ya que debe analizar el problema dependiendo del tipo al que corresponde y no cada problema de manera individual.



Para los docentes que participaron del dictado y uso de las plantillas se facilitó su trabajo debido a que los alumnos comprendieron mejor como detectar problemas de un mismo tipo y modelar de manera generalizada la solución, ello se traslada a los siguientes temas donde los alumnos intentan definir y usar sus propias plantillas para lograr soluciones genéricas a problemas particulares.

Este equipo trabaja permanentemente en el refinamiento de las plantillas definidas contrastándolas en diferentes contextos. El próximo anhelo es dar continuidad a esta tarea a través del desarrollo de una herramienta que implemente las plantillas genéricas, permitiendo automatizar la construcción de software, a través de la interpretación de un modelo real, con un conjunto de restricciones y de preferencias dadas.

## Referencias y Bibliografía consultada

- [1] G. Booch. Análisis y Diseño orientado a Objetos. Con aplicaciones. Addison Wesley.
- [2] K. Arnold, J. Gosling. "El Lenguaje de Programación Java". Addison Wesley.
- [3] G. Booch, J. Rumbaugh, I. Jacobson. The Unified Modeling Language. Addison Wesley. 1999.
- [4] I. Jacobson, G. Booch, J. Rumbaugh. The Unified Software Development Process. Addison Wesley. 1999.
- [5] R. Pressman. Software Engineering A Practitioner's Approach. Mc Graw Hill. Fifth Edition. 2001.
- [6] F. Buschmann, R. Meunier, H. Rohnert, P. Sommerlad, M. Stal. Pattern-Oriented Software Architecture, Volume 1: A System of Patterns. March 1997
- [7] M. Fowler, K. Beck, J. Brant, W. Opdyke, D. Roberts. Refactoring: Improving the Design of Existing Code.
- [8] I. Sommerville. Software Engineering. Addison Wesley.
- [9] D. Sanloz holonic. Evaluación de Proyectos, 1998, 2000 <http://www.geocities.com/Eureka/Office/4595/evalproy.html>.
- [10] IEEE: Standard Collection: Software Engineering, IEEE Standard 610.12-1990, IEEE 1993.
- [11] Lilia Verónica Toranzost. Organización de estados iberoamericanos para la educación, la ciencia y la cultura – o.e.i. Evaluación de proyectos- Marco metodológico. Junio 2001. [www.campus-oei.org/calidad/marco.PDF](http://www.campus-oei.org/calidad/marco.PDF).
- [12] Naur, P. y B. Randall(eds.) Software Engineering: A Report on a conference Sponsored by the NATO Science Committee, NATO, 1969.
- [13] Sánchez, M. La investigación sobre el desarrollo y la enseñanza de las habilidades de pensamiento. *Revista Electrónica de Investigación Educativa* 4, (1). 2002. <http://redie.ens.uabc.mx/vol4no1/contents-amestoy.html>
- [14] Tom DeMarco. Libro: Structured Analysis and System Specification, 1979.
- [15] Zelkowitz, M.V., Shaw, A.C. y Gannon, J.D.: Principles of Software Engineering and Design. Prentice Hall, 1979.
- [16] Object Management Group, 2000, OMG Unified Modeling Language Specification, Version 1.3, ad/00-03-01.
- [17] Erich Gamma, Richard Helm, Ralph Johnson, John Vlissides. Design Patterns: Elements of Reusable Object-Oriented Software. Addison-Wesley. 1995.
- [18] "Patterns in Java. Volume 1. A Catalog of Reusable Design Patterns Illustrated with UML". Mark Grand. John Wiley & Sons Inc. 1998.
- [19] Marcela Daniele, Daniel Romero, Nicolás Florio (2004). Definición y uso de Plantillas Genéricas para la descripción de Casos de Uso. Evaluado y Aprobado como Proyecto de Innovación e Investigación para el Mejoramiento de la Enseñanza (PIIMEG 2004) por la

Secretaría de Ciencia y Técnica y la Secretaría Académica de la Universidad Nacional de Río Cuarto. Resolución Rectoral N° 302/04. Año 2004.

- [20] Marcela Daniele, Daniel Romero (2005). Evolución de Plantillas Genéricas para la descripción de Casos de Uso a Plantillas para el Análisis y Diseño. Evaluado y Aprobado como Proyecto de Innovación e Investigación para el Mejoramiento de la Enseñanza (PIIMEG 2005) por la Secretaría de Ciencia y Técnica y la Secretaría Académica de la Universidad Nacional de Río Cuarto. RR N° 109-110/05.
- [21] Marcela Daniele, Daniel Romero (2005). Evolución de Plantillas Genéricas para la descripción de Casos de Uso a Plantillas para el Análisis y Diseño. Publicado en los anales de Workshop de Investigadores en Ciencias de la Computación (WICC 2005), mayo 2005. Universidad Nacional de Río Cuarto. ISBN: 950-665-337-2.
- [22] Marcela Daniele, Daniel Romero (2005). Proyectos e informes finales del Proyecto de Innovación e Investigación para el Mejoramiento de la Enseñanza de Grado, “Definición y Uso de Plantillas Genéricas para la descripción de Casos de Uso” y “Evolución de Plantillas Genéricas para la descripción de Casos de Uso a Plantillas Genéricas para Análisis y Diseño”. <https://dc.exa.unrc.edu.ar/materias/sistemas/archivos/PIIMEG/>. Responsable: Ing. Marcela Daniele.