# Herramienta para administración y validación de requerimientos de sistemas

Oscar Carlos Medina, Marcelo Martín Marciszack, Mario Alberto Groppo

GIDTSI, Grupo de Investigación, Desarrollo y Transferencia de Sistemas de Información
Universidad Tecnológica Nacional, Facultad Regional Córdoba
Maestro López esq. Av. Cruz Roja Argentina, Ciudad Universitaria – (5016) Córdoba
oscarcmedina@gmail.com, marciszack@gmail.com, sistemas@groppo.com.ar

**Abstract.** El presente trabajo describe una aplicación web denominada SIAR (Sistema Integral de Administración de Requerimientos) que gestiona los requerimientos de un sistema de información mediante Casos de Uso, según los lineamientos de UML (Lenguaje Unificado de Modelado). Los Casos de Uso son útiles para la generación y análisis de requisitos de sistemas. La finalidad principal de SIAR es la administración de Casos de Uso con una herramienta informática que agilice su registración, normalice su contenido y posibilite implementar validaciones funcionales, como por ejemplo un método automatizado de análisis de consistencia de Casos de Uso, para lo cual el sistema genera un grafo con la transición de estados de cada Caso de Uso, expresado en el protocolo XPDL (Lenguaje de Definición de Flujo de Trabajo), que es analizado en un simulador de autómata finito determinista para verificar la cohesión de los escenarios en él definidos.

**Palabras Clave:** Caso de Uso, Requerimientos, UML, XPDL, Autómata finito determinista.

## 1 Introducción

SIAR (Sistema Integral de Administración de Requerimientos) es la denominación que se le dio a una aplicación web que gestiona los requerimientos funcionales de un sistema de información según los lineamientos de UML (Lenguaje Unificado de Modelado).

Esta aplicación se originó dentro del proyecto de investigación "Validación de Requerimientos a través de Modelos Conceptuales" del GIDTSI (Grupo de Investigación, Desarrollo y Transferencia de Sistemas de Información), dependiente del Departamento Ingeniería en Sistemas de Información de la Universidad Tecnológica Nacional, Facultad Regional Córdoba que "busca dar solución a uno de los principales problemas de la Ingeniería de Requisitos relacionado a la elicitación y especificación de requerimientos, que vincula las distintas etapas del proceso de desarrollo de software manteniendo la trazabilidad de los mismos hasta su validación e implementación".

Como el modelo de requerimientos tiene por objetivo la registración y estandarización de requerimientos funcionales, la construcción de SIAR tiene por fin administrar en forma integral los requerimientos funcionales de un proyecto de software según la metodología UML. SIAR es una aplicación web que permitió registrar en forma normalizada los casos de uso y cuya primera versión comprende el siguiente alcance:

- Administración de los atributos de un proyecto de software y su versionado.
- Diseño y validación del Modelo Conceptual.
- Gestión de los alcances de cada versión del proyecto y los casos de uso asignados.
- Administración de los atributos de un caso de uso, incluyendo actores, pre-condiciones, post-condiciones, cursos de acción, normal y alternativos, y su versionado.

- Clasificación, priorización y trazabilidad de los casos de uso.
- Visualización de consultas y generación de reportes en distintos formatos, inclusive XPDL, para comunicarse con otras aplicaciones.
- Gestión de atributos de procesos de negocio, de actividades de negocio que los componen y los casos de uso asociados a estas actividades.
- Registración y consulta de un glosario por proyecto, con entradas y sinónimos, siguiendo las recomendaciones de LEL (Léxico Extendido del Lenguaje), que es una estrategia de modelado de requisitos basada en lenguaje natural.

## 2 Construcción de una herramienta para la administración integral de Casos de Uso

La herramienta ha sido diseñada con un criterio de lógica que permite describir la siguiente relación de cascada:

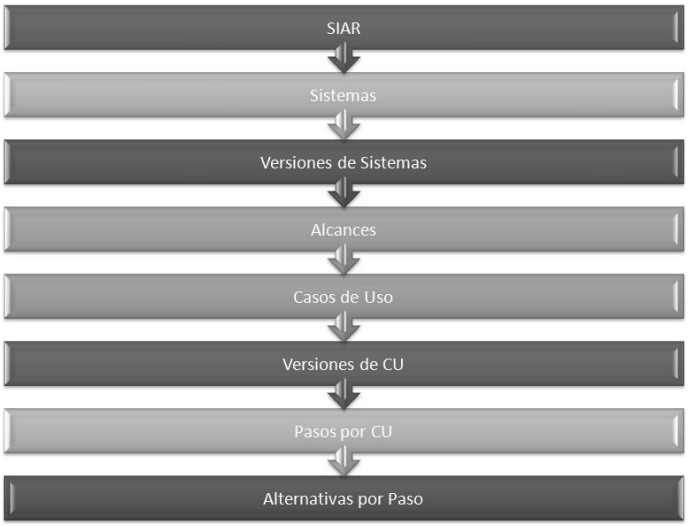


**Fig. 1.** Relaciones de SIAR..

SIAR permite trabajar con diferentes Sistemas, cada uno con sus propias Versiones, cada Versión cubre un número limitado de Alcances, cada Alcance gestiona un grupo de Casos de Uso, cada Caso de Uso está compuesto por una secuencia ordenada de Pasos, finalmente un paso puede o no tener Alternativas.

Los Pasos de un Caso de Uso permiten efectuar tareas que son descriptos con el soporte del sistema que se ha desarrollado.

Desde el punto de vista conceptual un Caso de Uso es la descripción de una secuencia de interacciones entre el sistema y uno o más Actores. Las personas o entidades que participarán en un caso de uso se denominan Actores.

Cada caso de uso tiene pre-condiciones que se deben cumplir para que el flujo de eventos se puedan llevar a cabo. Dichas pre-condiciones son las reglas o condiciones que se deben cumplir antes de que sea iniciado el caso de uso.

También posee post-condiciones que reflejan el estado en que se queda el Sistema una vez ejecutado el caso de uso.

Puede clasificarse con diferentes niveles de Complejidad según el criterio de cada Analista funcional en Muy Alta, Media y Baja.

A su vez, un Caso de Uso puede ser del tipo Concreto o Abstracto. Un caso de uso es Abstracto si no puede ser realizado por sí mismo, o si no es Concreto ya que puede ser iniciado por un actor y realizado por sí mismo.

Los pasos son las actividades que deberán realizarse para llevar a cabo algún proceso (Rumbaugh, Jacobson, Booch, 1999).

Cada paso posee alternativas que a su vez puede tener pasos y estos volver a tener alternativas.

Cada caso de uso está codificado con 5 dígitos que conforman el Identificador del Caso de Uso:

1. Id del Sistema.
2. Id de la Versión del Sistema.
3. Id del Alcance de la Versión.
4. Id del Caso de Uso.
5. Id de la Versión del Caso Uso.

# 3 Alcance de SIAR versión 1.0

El aplicativo busca cubrir las siguientes necesidades:

## 3.1 Administración de requerimientos

Se planteó la necesidad de generar un sistema que no solo administre los casos de usos por separado, sino también gestione en forma integral todo el proyecto que lo contiene. Es decir, sistema con sus versiones, por cada versión del sistema sus alcances y por cada alcance sus casos de usos con sus versiones.

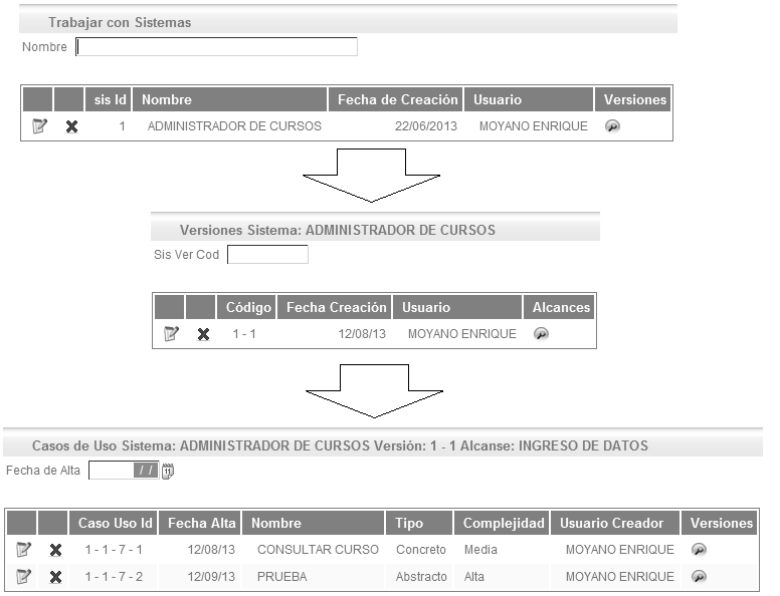


**Fig. 2.** Consulta de Casos de Uso de un Sistema.

Para cumplir con esta funcionalidad se incluyó también el concepto de sistema, versiones y alcances de versiones. A su vez cada caso de uso incluye pasos y alternativas. También se propone hacer una trazabilidad de los cambios.

### 3.2 Configuración del entorno de desarrollo

Para su implementación se utilizó una plataforma de desarrollo "case", generando en lenguaje de programación C# y almacenando la información en una base de datos MYSQL.

### 3.3 Interfaz de usuario

El desarrollo del sistema es web porque solo se precisa Internet para acceder a la aplicación desde una computadora personal. No se escogió un desarrollo de escritorio porque sería necesario contar con un entorno de configuración local en el equipo que se desee acceder al sistema.

### 3.4 Administración de proyectos de sistemas

Como se mencionó anteriormente, el trabajo tiene por objetivo principal administrar las descripciones y mantenimiento de casos de uso en forma integral.

### 3.5 Administración de Casos de Uso

Durante la descripción de un Caso de Uso, pueden administrarse las pre-condiciones, post-condiciones, pasos y alternativas como así también los datos básicos de un Caso de Uso. Del mismo modo, en la descripción de un nivel o paso, una vez definido el mismo puede modificarse o darse de baja, esto se ve reflejado en las versiones del CU. Lo mismo con cada alternativa de un nivel o paso.

### 3.6 Versionado de Casos de Uso

La aplicación permite versionar sistemas y versionar Casos de Uso. Se ofrecen distintas herramientas para llevar a cargo esta tarea, por ejemplo el listado de comparación de versiones de un caso de uso, detalla los cambios entre dos versiones seleccionadas. Puede ocurrir que el usuario que consulta el versionado del Caso de Uso no sea el mismo que lo generó, de esta manera puede ver reflejada en el listado la evolución del CU desde el principio hasta la última modificación.

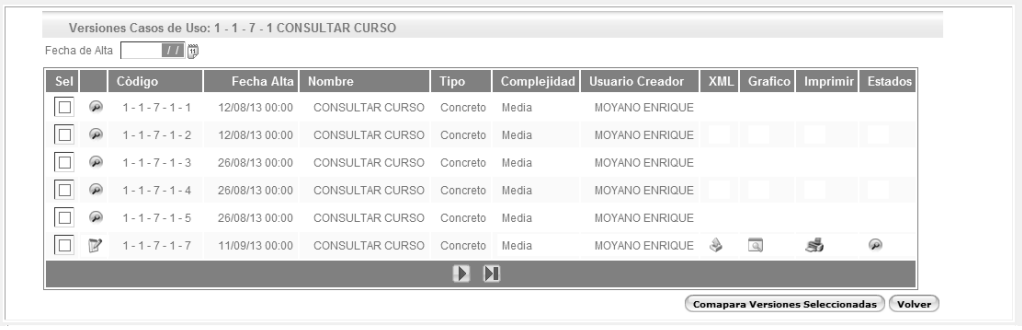


**Fig. 3.** Versionado de un Caso de Uso.

### 3.7 Exportación a archivos XML

Esta es una funcionalidad que proporciona el sistema para poder intercambiar datos con otras aplicaciones por ejemplo un autómata finito. El archivo XML representa en este protocolo de intercambio de datos el grafo de estados del Caso de Uso que surge de la equivalencia con sus pasos y alternativas.

### 3.8 Consultas

Es posible obtener las siguientes salidas del sistema:

a) Sistemas
b) Versionado de Sistemas
c) Alcances por Versión de Sistema
d) Casos de Uso por Alcance
e) Versionado de Casos de Uso
f) Comparación entre dos Versiones de un Caso de Uso
g) Reportes de Casos de Uso

Para la versión vigente de un Caso de Uso se pueden imprimir los siguientes reportes:

- Conversión a archivo XML para ingresar a Autómata Finito Determinista.
- Gráfico de estructura del Caso de Uso en forma de árbol.
- Reporte PDF del Caso de Uso.
- Tabla de Estados del Caso de Uso.

h) Información de Usuarios

## 4 Validación de consistencia de Casos de Uso con simuladores de autómatas finitos

En la actualidad UML es reconocido como el estándar para el modelado de proyectos de software orientado a objetos.

Los componentes principales de esta metodología son los Casos de Uso ya que especifican el comportamiento deseado del sistema. Por definición el caso de uso “especifica una secuencia de acciones, incluyendo variantes, que el sistema puede ejecutar y que produce un resultado observable de valor para un particular actor”.

Una vez generado un caso de uso, es necesario comprobar y asegurar su validez. La verificación de consistencia de la secuencia de acciones descripta en el caso de uso es una de las tareas que permiten su validación.

Es deseable que esta verificación pueda realizarse de manera automatizada para lo cual se podría trabajar con autómatas finitos deterministas, ya que un autómata finito es un conjunto de estados y un control que se mueve de un estado a otro en respuesta a entradas externas (Marciszack, Pérez, Castro, 2013). Se llama determinista al autómata que puede estar únicamente en un estado en un momento determinado. El desafío es transformar el caso de uso en un autómata finito determinista para validar su cohesión secuencial.

Partiendo de estas premisas, el aporte del presente análisis a la validación de requerimientos que se propone lograr el proyecto de investigación, consiste en dar respuesta a la siguiente pregunta:

¿Es factible determinar un método, basado en simuladores de autómatas finitos deterministas, que verifique la consistencia de la secuencia de acciones, incluyendo variantes, definidas en un caso de uso?.

La funcionalidad de SIAR que se detalla a continuación propone una alternativa viable para esta necesidad, que se realiza en 3 pasos:

4.1 Registración normalizada del requerimiento.

4.2 Transformación del Caso de Uso en una máquina de estados.

4.3 Validación de la consistencia secuencial de los cursos de acción del Caso de Uso.

### 4.1 Registración normalizada del requerimiento

Como el modelo de requerimientos tiene por objetivo la captura de requerimientos funcionales (Jacobson y otros, 1992), la primera tarea que se emprendió, fue la construcción de SIAR, una herramienta cuyo fin es administrar en forma integral los requerimientos funcionales de un proyecto de software según el estándar UML y permite registrar en forma normalizada los casos de uso.

### 4.2 Transformación del Caso de Uso en una máquina de estados

En segunda instancia se definió un conjunto de reglas de conversión del caso de uso en un grafo de estados para que sea la entrada a un simulador de autómata finito determinista.

Una vez completa la versión de un caso de uso, se genera un grafo de estados a partir de las siguientes definiciones:

- Todo caso de uso tiene al menos un curso de acción normal.
- Un caso de uso puede no tener ningún, o tener uno o tener varios cursos de acción alternativos.
- Cada paso de un curso de acción de un caso de uso responde a un estado y es un nodo de la máquina de estados.
- Todo caso de uso tiene un paso inicial, y es el primer paso de todos los cursos de acción. Este paso es el estado/nodo inicial.
- Todo caso de uso tiene un paso final por aceptación, y es el último paso del curso de acción normal. También puede ser el último paso de algún curso alternativo. Este paso es el estado/nodo final por aceptación.
- Un caso de uso puede no tener ningún, tener uno o tener varios finales por error, y son el último paso de un curso alternativo. Estos pasos son estados/nodos finales por error.
- Todo caso de uso pasa de un estado/nodo a otro por medio de una función de transición.
- Partiendo de un estado/nodo origen se puede continuar en un único estado/nodo destino, o en dos nodos/estados destino alternativos dependiendo de una condición de bifurcación.

• El grafo de estados asociado al caso de uso tiene un alfabeto de tres símbolos para indicar qué evento lo cambia de un estado/nodo a otro:

- o A = Por medio de una Acción determinada.
- o S = Cuando Si se cumple una condición que bifurca los cursos de acción.
- o N = Cuando No se cumple una condición que bifurca los cursos de acción.


**Fig. 4.** Bifurcación en la especificación de trazo fino de un Caso de Uso.

Partiendo de un estado/nodo origen, en la función de transición puede estar asociado solamente uno de los símbolos: A, N o S. Con esto se cumple la condición necesaria de un autómata finito determinista. De esta manera, si la transición entre dos estados/nodos se da dentro de un mismo camino, se asocia el símbolo A. Si en cambio interviene una bifurcación, la función de transición hacia el estado/nodo destino por cumplimiento de la condición de bifurcación, se asocia el símbolo S. Por el otro camino de la bifurcación, se asocia el símbolo N.

Tabla De Estados Casos de Uso: 1 - 1 - 1 - 1 CONSULTAR CURSOS Versión: 6

| Estado / Paso Origen | Estado / Paso Destino | Transición | Estado Final por Error | Tipo |
|---|---|---|---|---|
| 1 | 2 | A | | S |
| 2 | 3 | A | | S |
| 3 | 4 | A | | S |
| 4 | 4 - A | N | | C |
| 4 - A | 4 - A - 1 | A | | S |
| 4 - A - 1 | 4 - A - 2 | A | SI | S |
| 4 | 5 | S | | C |
| 5 | 6 | A | | S |
| 6 | 6 - A | S | | C |
| 6 - A | 6 - A - 1 | A | | S |
| 6 | 7 | N | | C |

**Fig. 5.** Tabla de estados de un Caso de Uso. Transformación automática por SIAR.

Una vez generado el grafo de estados se expresa en protocolo XPDL, por ser el más adecuado para intercambiar modelos de procesos entre distintas herramientas. Este lenguaje "da soporte a la definición y a la importación/exportación de procesos, con el objetivo de que, aunque se modele un proceso en una aplicación, este modelo pueda ser usado por otras aplicaciones de modelado y/o por otras aplicaciones que trabajen en el entorno de ejecución" (Pérez, 2007).

```
<transicion
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:type="java:dominio.automatas.TransicionFinita">
    <simbolo>
        <simbolo>S</simbolo>
    </simbolo>
    <estado-inicio>
        <denominacion>4</denominacion>
    </estado-inicio>
    <estado-fin>
        <denominacion>5</denominacion>
    </estado-fin>
    <denominacion>f( 4, S) = 5</denominacion>
</transicion>
```

**Fig. 6.** Fragmento de archivo XPDL generado por SIAR.

### 4.3 Validación de la consistencia secuencial de los cursos de acción del Caso de Uso

Finalmente se ingresa el archivo XPDL, que representa al caso de uso como grafo de estados, al programa "Autómata Finito" que forma parte del conjunto de "Herramientas Prácticas para el Aprendizaje de Informática Teórica". Esta aplicación java "permite definir autómatas y gramáticas para la enseñanza y ejercitación de los alumnos de la asignatura Sintaxis y Semántica del Lenguaje que se dicta en la carrera de Ingeniería en Sistemas de Información de la Universidad Tecnológica Nacional Facultad Regional Córdoba" (U.T.N. F.R.C., 2009).

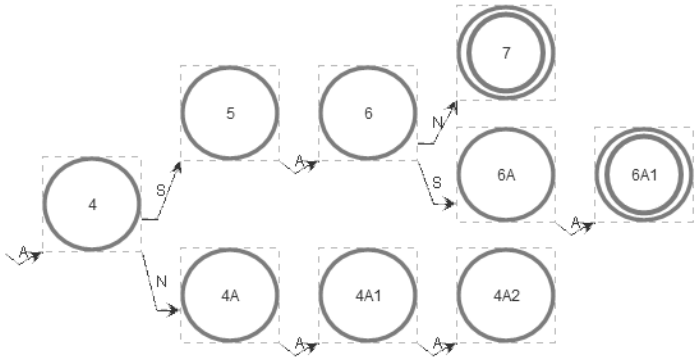


**Fig. 7.** Fragmento del grafo de estados en el simulador de autómatas finitos.

El simulador posibilita probar el grafo de estados y comprobar si es aceptado o rechazado. En el segundo caso se puede visualizar el detalle para detectar la inconsistencia, la cual puede ser corregida en una nueva versión del caso de uso, generando un nuevo grafo de estados automáticamente. De esta manera se logra control y trazabilidad de los cambios en los cursos de acción de los requerimientos funcionales. Vale recordar a F. P. Brooks cuando dijo: "la tarea más importante que el ingeniero de software hace para el cliente es la extracción iterativa y el refinamiento de los requerimientos del producto".

### 4.4 Limitaciones

El software limita su alcance con las siguientes restricciones:

- En cada paso abarca alternativas hasta un nivel, sin embargo cada alternativa puede incluir más de un paso.
- Los Casos de Usos no registran relaciones de inclusión y extensión..

### 4.5 Resultados

Se pone a consideración este método automatizado, implementado en el aplicativo SIAR que registra, normaliza y transforma un caso de uso a formato XPDL y un simulador de autómatas finitos, que permite verificar la consistencia secuencial de los distintos caminos del caso de uso.
El aporte que realiza SIAR al proyecto en el que fue concebido, "Validación de Requerimientos a través de Modelos Conceptuales", es el de constituirse como plataforma de software integradora de las aplicaciones que se utilizan en cada una de las líneas de investigación.
El presente desarrollo está fundamentado en una "Propuesta Metodológica para validación de Requerimientos Funcionales a través de Modelos Conceptuales" registrada en la República Argentina con Derecho de autor de producciones tecnológicas (rubro "Modelo de organización y/o gestión, Ciencia y cultura-Ciencia y tecnología") según Expediente No.5229942. Esta metodología expone cómo se llega a identificar la necesidad de construir SIAR y la especificación de requerimientos funcionales de sus alcances, módulos y funcionalidades.
También "S.I.A.R. – Sistema Integral de Administración de Requerimientos" está registrado en la República Argentina con Derecho de autor de producciones tecnológicas (rubro "Máquina, equipo, instrumento y/o herramienta o su/s componente/s. Informática-software. Ciencia y cultura-Ciencia y tecnología") según Expediente No.5229955.
En lo que respecta a la funcionalidad de análisis de consistencia, SIAR ofrece un método automatizado para validar la cohesión de un caso de uso desde el punto de vista de la transición de estados definidos intrínsecamente en los pasos de su especificación funcional. Permitiendo también enlazar este proyecto con un trabajo académico de la carrera Ingeniería en Sistemas de Información, el del Grupo de Herramientas Didácticas de Informática Teórica que contribuyó con el simulador de autómatas finitos.
Finalmente se tiene previsto en el año 2015 hacer una transferencia del software a una empresa especializada en desarrollos de seguridad informática de la región en el marco de un programa gubernamental de generación y transferencia de conocimientos.

## 5 Conclusión

En el presente trabajo se han analizado y comprendido el procedimiento y el modelo de datos asistido por una aplicación web denominada SIAR que gestiona los requerimientos funcionales de un sistema de información según los lineamientos de UML. Haciendo posible también enlazar este proyecto con un trabajo académico de la carrera Ingeniería en Sistemas de Información, el del Grupo de Herramientas Didácticas de Informática Teórica que contribuyó con el simulador de autómatas finitos, que permitió implementar un método automatizado para validar la cohesión de un caso de uso desde el punto de vista de la transición de estados definidos intrínsecamente en los pasos de su especificación funcional.

Como lo señalaron Taurant y Marciszack: "Actualmente existen en el mercado una gran variedad de herramientas y metodologías que permiten la gestión de requerimientos de software a lo largo del ciclo de vida de desarrollo. Independientemente de la herramienta o metodología utilizada, la creación y mantención de un

gran número de modelos y artefactos es realizada por el analista en forma manual, generando con gran frecuencia inconsistencias entre los modelos generados, impactando en la trazabilidad de los requerimientos."

Es por esto que se propuso con SIAR, el desarrollo de una herramienta que permita al analista gestionar los requerimientos en forma asistida y parcialmente automatizada, por medio de la generación de modelos conceptuales como representaciones de máquinas abstractas, considerando que se alcanzaron estos objetivos parcialmente.

Una alternativa factible para continuar trabajando con SIAR es ampliar la trazabilidad de los modelos de requerimientos de manera tal que permita medir el impacto de los cambios desde el Modelo Conceptual hasta el Modelo de Sistema de Información agregando técnicas de validación y funcionalidades de gestión de proyectos de software. Hasta el momento la metodología propuesta y el modelo de datos de la herramienta que la soporta, posibilitan la trazabilidad desde el modelo de requerimientos del sistema con su origen en el modelo de datos, pero aún no se alcanza a medir el impacto en el cambio de dichos modelos. Se está indagando la incorporación de "Patrones", según el concepto de la Ingeniería de Software, en la validación de Modelos Conceptuales como el próximo paso a seguir para ampliar el alcance de esta herramienta.

Además se hace necesario validar dentro del Modelo Conceptual también los Requerimientos no funcionales desde el punto de vista del usuario, entendiendo que usabilidad es el atributo de calidad que mide la facilidad de las interfaces de un sistema.

## Referencias


Rumbaugh, J., Jacobson,I., Booch,G. (1999). The Unified Modelling Language Reference. Addisson Wesley.

Chakraborty, Samarjit (2003). Formal Languages and Automata Theory-Regular Expressions and Finite Automata-. Computer Engineering and Networks Laboratory Swiss Federal Institute of Technology (ETH) Zurich.

Marciszack, Marcelo, Pérez,Ramiro, Castro, Claudia Castro (2013). Validación de Requerimientos a través de Modelos Conceptuales – Modelos y Transformaciones. WICC 2013.

Jacobson, Ivar y otros (1992). Object Oriented Software Engineering. A Use Case Driven Approach. Addison Wesley.

Pérez, J. D. (2007). Notaciones y lenguajes de procesos. Una visión global. Tesis de Doctorado Universidad de Sevilla.

U.T.N. F.R.C. (2009). Proyecto Construcción de Herramientas Didácticas para la enseñanza y ejercitación práctica en laboratorio de Informática Teórica en las Carreras con Informática. Manual de Usuario – Grupo de Herramientas Didácticas.


## Bibliografía adicional


Sommerville, I. (2005). Software Enginnering, Computing Departament, Lancaster University, John Willey&Sons Ltd.

Davis, A. (1993). Software requeriments. Object, functions and states. Pretince Hall international Inc.

Brooks, Frederik P. (1987). No Silver Bullet. Essence and Accidents in Software Engineering. IEEE Computer.

Leonardi, C., Leite, J.C.S., Rossi, Gustavo (2001). Una estrategia de Modelado Conceptual de Objetos, basada en Modelos de requisitos en lenguaje natural. Tesis de Maestría Universidad Nacional de la Plata.