

**Traceability
en la elicitación y especificación
de requerimientos**

Leandro Antonelli

**Directores: Alejandro Oliveros
Gustavo Rossi**

Tesis presentada a la Facultad de Informática de la Universidad Nacional de La Plata como parte de los requisitos para la obtención del título de Magister en Ingeniería de Software.

La Plata, Febrero de 2003

*Facultad de Informática
Universidad Nacional de La Plata
Argentina*

Índice general

1.	Introducción	7
2.	Rol de los requerimientos en el ciclo de vida del software	9
2.1.	Los requerimientos como condicionantes de la calidad del software.....	9
2.2.	Importancia de la gestión de requerimientos	9
2.3.	Concepto de traceability	10
2.4.	Importancia de traceability	13
2.5.	Importancia de forward traceability	14
3.	Modelo de LEL y escenarios.....	15
3.1.	Modelo contextual.....	15
3.2.	Organización del conocimiento experto	15
3.3.	Lexico Extendido del Lenguaje	16
3.3.1.	Ejemplo	16
3.3.2.	Proceso de construcción de LEL	18
3.4.	Modelo de escenarios.....	20
3.4.1.	Ejemplo	21
3.4.2.	Proceso de construcción de escenarios	22
3.5.	Client Oriented Requirements Baseline.....	23
3.6.	Tarjetas CRC	24
3.6.1.	Generalidades	24
3.6.2.	Reglas de derivación de tarjetas CRC	24
3.6.3.	Ejemplo	25
3.7.	Relación entre LEL, escenarios y tarjetas CRC.....	25
4.	Herramienta automatizada	27
4.1.	Importancia de una herramienta automatizada en la ingeniería de software.....	27
4.2.	Baseline Mentor Workbench.....	27
4.3.	Evaluación de la usabilidad de BMW	30
4.3.1.	Descripción de la evaluación.....	30
4.3.2.	Participantes	32
4.3.3.	Evaluación de características generales	33
4.3.4.	Evaluación de características detallada	33
4.3.5.	Nuevos rasgos a incorporar	34
4.3.6.	Curva de aprendizaje	35
4.3.7.	Conclusión	35
4.4.	Evaluación del algoritmo de derivación de CRCs implementado en BMW	35
4.4.1.	Comparación del algoritmo implementado en BMW contra la aplicación manual	36
4.4.2.	Usabilidad del algoritmo implementado en BMW	42
4.5.	Futuros trabajos	48
5.	Forward traceability en el modelo de LEL y escenarios.....	49
5.1.	Objetivo	49
5.2.	Medio de lograr el objetivo	51
5.3.	Análisis del algoritmo de derivación de tarjetas CRC.....	52
5.4.	Análisis de los cambios posibles en el Client Oriented Requirements Baseline	54
5.5.	Reglas de traceability forward	57
5.6.	Comparación del modelo de traceability con otros modelos.....	61
6.	Implementación de las reglas de traceability forward.....	63
6.1.	Necesidad de implementar las reglas	63
6.2.	Descripción del modelo ejecutable.....	63
6.3.	Modelo ejecutable.....	64
6.4.	Diseño de objetos	65
6.5.	Nueva arquitectura de BMW.....	67
6.6.	Ejemplos	68

7.	Futuros trabajos.....	72
8.	Referencias.....	73
9.	Anexos.....	76
9.1.	LEL, escenarios y tarjetas CRC del círculo de ahorro.....	76
9.1.1.	LEL	76
9.1.2.	Escenarios	82
9.1.3.	Tarjetas CRC.....	87

Índice de figuras

Figura 1 - Diagrama del ciclo de vida del software propuesto por Davis.	10
Figura 2 - Catarata de errores de Mizuno.	11
Figura 3 – Modelo de traceability de Davis.....	12
Figura 4 – Hipertexto de las entradas de LEL de un organismo gubernamental de la República Argentina.....	18
Figura 5 – Diagrama entidad relación de la estructura de un escenario.	21
Figura 6 – Hipertexto de LEL y escenarios de un organismo gubernamental de la República Argentina.	22
Figura 7 - Hipertexto de LEL, escenarios y tarjetas CRC de un organismo gubernamental de la República Argentina.....	26
Figura 8 – Arquitectura del Baseline Mentor Workbench.	27
Figura 9 – Ventana principal del Baseline Mentor Workbench.	28
Figura 10 – Ventana del browser completo del BMW.	29
Figura 11 – Menues contextuales del BMW.	30
Figura 12 – Ventana de edición de los escenarios del BMW con los menues contextuales.	30
Figura 13 – Ventana de navegación del BMW.	31
Figura 14 – Casos de uso del sistema de administración de planes de ahorro.	44
Figura 15 – Diagrama de clases del sistema de administración de planes de ahorro.	45
Figura 16 – Diagrama de secuencia del armado del plan.....	47
Figura 17 – Diagrama de secuencia de realizar el sorteo.....	47
Figura 18 – Diagrama de secuencia de echar un adherente.	48
Figura 19 – Derivación de una tarjeta CRC a partir de LEL y escenarios.	50
Figura 20 – Modificación de los impactos de la entrada de LEL abogado.	50
Figura 21 – Modificación de término de la entrada de LEL “particulares” por “demandado”.	51
Figura 22 – Mecánica de las reglas de derivación.	53
Figura 23 – Notación utilizada en el diagrama de actividades.	64
Figura 24 – Diagrama de actividades de las reglas de traceability forward.....	65
Figura 25 – Diagrama de clases de la implementación de Traceability en BMW.....	66
Figura 26 – Diagrama de secuencias de eliminar un escenario.	67
Figura 27 – Nueva arquitectura del Baseline Mentor Workbench.....	67
Figura 28 – Entrada de LEL “intimación negativa”.	68
Figura 29 – Escenario “intimación de una razón social” referenciando a la entrada de LEL “intimación negativa” en la última línea de los episodios.	68
Figura 30 – Pedido de confirmación para actualizar las referencias.	69
Figura 31 – Modificación del texto “intimación negativa” por “desconocido”.	69
Figura 32 – Entrada de LEL que posee términos que no son sinónimos.	69
Figura 33 – Referencia a la entrada de LEL “demandado”.	69
Figura 34 – Eliminación de la referencia a “demandado”.	70
Figura 35 – Entrada de LEL “abogado” con sus impactos incompletos.	70
Figura 36 – Tarjeta CRC correspondiente a la entrada de LEL “abogado”.	70
Figura 37 – Agregado de una línea en los impactos.....	70
Figura 38 – Tarjeta CRC con las responsabilidades modificadas.	71
Figura 39 – CRC creada a partir de la modificación en CRC “abogado”.	71
Figura 40 – Segunda CRC creada a partir de la modificación en CRC “abogado”.	71

Índice de tablas

Tabla 1 – Costo relativo de reparar errores a lo largo del ciclo de vida del software.	10
Tabla 2 – Modelo de descripción de una entrada de LEL.	16
Tabla 3 – Entrada de LEL “abogado”.	17
Tabla 4 – Entrada de LEL “cédula de notificación”.	17
Tabla 5 – Entrada de LEL “intimación”.	17
Tabla 6 – Entrada de LEL “intimación negativa”.	17
Tabla 7 – Entrada de LEL “juicio”.	17
Tabla 8 – Entrada de LEL “particulares”.	17
Tabla 9 – Escenario “Confeccionar una cédula de notificación”.	21
Tabla 10 – Escenario “Intimar a una razón social”.	22
Tabla 11 – CRC primaria “abogado”.	25
Tabla 12 – CRC secundaria “cédula de notificación”.	25
Tabla 13 – CRC secundaria “juicio”.	25
Tabla 14 – CRC secundaria “particulares”.	25
Tabla 15 – Cantidad de personas que se graduaron por rango de años.	32
Tabla 16 – Años de experiencia de trabajo en la industria.	32
Tabla 17 – Cantidad de rasgos evaluados con baja utilidad por el grupo de 49 y 12 personas.	33
Tabla 18 – Porcentaje de respuestas obligatorias de los grupos de 49 y 12 personas.	35
Tabla 19 – Tiempo de aprendizaje.	35
Tabla 20 – LEL y escenarios obtenidos en el primer caso de estudio.	37
Tabla 21 – Comparación de las CRCs obtenidas del primer caso de estudio.	37
Tabla 22 – LEL y escenarios obtenidos en el segundo caso de estudio.	38
Tabla 23 – Comparación de CRCs obtenidas del segundo caso de estudio.	39
Tabla 24 – Resumen de los dos casos de estudio.	42
Tabla 25 – Entradas de LEL agregadas al caso de estudio uno.	42
Tabla 26 – Tarjetas CRC derivadas para el caso de estudio de la administradora del plan de ahorro.	43
Tabla 27 – Elementos del Client Oriented Requirements Baseline con sus atributos.	54
Tabla 28 – Naturaleza de los cambios sobre el Client Oriented Requirements Baseline.	55
Tabla 29 – Cambios posibles sobre los atributos de LEL, escenario y tarjetas CRC.	56
Tabla 30 – Cambios que deben ser estudiados.	58
Tabla 31 – Reglas que se aplican en cada caso de análisis.	61

**Traceability
en la elicitación y especificación
de requerimientos**

1. Introducción

La calidad del software depende fuertemente de sus requerimientos. Garvin define calidad a partir de ellos [Garvin 1984]. Davis establece la conveniencia de realizar las pruebas de aceptación del sistema a partir de los requerimientos [Davis 1993]. Boehm, en una de las métricas más reconocidas en la Ingeniería Software [Boehm 2001], establece que reparar errores en las etapas posteriores a la de requerimientos puede llegar a ocasionar un costo de entre 100 a 200 veces mayor que hacerlo en la de definición de requerimientos. Pero las palabras de Ackoff [Ackoff 1974] son las que mejor sintetizan la importancia de los requerimientos: "Fallamos más a menudo porque resolvemos el problema incorrecto, que porque obtenemos una solución inadecuada del problema correcto"

Los requerimientos son dinámicos al igual que el dominio del problema. Su gestión es recomendada por los estándares: ISO/IEC 12207, MIL-STD-498 y DOD-STD-2167A. El SEI coloca a la gestión de requerimientos en el primer escalón del Capability Maturity Model para llegar a un proceso de desarrollo disciplinado.

Si los requerimientos cambian, los productos obtenidos a partir de los requerimientos durante el desarrollo de software y la fase de mantenimiento también lo deben hacer. Este es el problema que contribuye a atacar la rastreabilidad de requerimientos (requirements traceability): seguir la evolución de los requerimientos a lo largo del ciclo de vida. Gotel [Gotel 1994] distingue dos direcciones: hacia sus orígenes (backward) y hacia la implementación (forward).

Forward traceability es muy importante para el desempeño de tres funciones [Wieringa 1995]. Al gerente de proyectos le permite estimar el impacto que los cambios de los requerimientos tienen en el diseño y la implementación de software, y viceversa; y descubrir conflictos en forma temprana. Al diseñador le permite desarrollar el producto de software evitando inconsistencias entre sus partes. Y al cliente le permite conocer donde se implementa cada requerimiento.

Los requerimientos necesitan de un modelo del contexto del dominio que le den un marco. Leite [Leite 1997] propone un modelo de contexto en dos etapas. Primero sugiere comprender el lenguaje del dominio del problema y luego estudiar la dinámica de éste. Él desarrolló el LEL (Léxico Extendido del Lenguaje) y un modelo particular de escenarios para atacar cada uno de los problemas. Estas técnicas resultan apropiadas tanto para el Ingeniero de Requerimientos como para el experto del dominio. En primer lugar porque utilizan lenguaje natural comprensible por ambos. Además porque la organización de LEL y escenarios es similar a la organización que plantea Wood [Wood 1997] del conocimiento en el cerebro del experto.

Leonardi [Leonardi 2001] desarrolló un conjunto de reglas para obtener tarjetas CRC a partir de LEL y escenarios. Así, se puede obtener en forma sistemática un conjunto de objetos para realizar una aplicación bajo esta tecnología.

LEL y escenarios se utilizan para capturar y abstraer conocimiento del dominio en el cual se utilizará un sistema de software. A partir de ambos se obtienen en forma sistemática tarjetas CRC. Cuando la realidad del dominio donde se utilizará un sistema de software cambia, el LEL y los escenarios también deben cambiar para que se mantenga consistente con el dominio que modelan. Luego, las tarjetas CRC también se deben modificar porque sus orígenes lo han hecho.

Los cambios del dominio impactan en el LEL y en los escenarios. Luego se propagan a las tarjetas CRC, para proyectarlos por el resto de los productos del ciclo de vida del software. El propagar estos cambios es un problema de rastreabilidad hacia adelante.

En esta tesis se describe un análisis conceptual de la vinculación entre LEL, escenarios y tarjetas CRC, con el fin de determinar un conjunto de reglas que prescriban, en función de los cambios que se producen en el LEL y los escenarios, qué cambios deben sufrir los demás elementos.

La tarea de determinar los cambios producidos en LEL y escenarios, seleccionar la regla adecuada y aplicarla, es una tarea ardua, propensa a errores y que se puede automatizar. Ramesh [Ramesh 1998] plantea la importancia de utilizar herramientas automatizadas para esta tarea. El Baseline Mentor Workbench administra LEL, escenarios, tarjetas CRC e implementa las reglas de Leonardi. Las reglas de forward traceability se implementarán en Baseline Mentor Workbench.

El objetivo de esta tesis es desarrollar un modelo de rastreabilidad hacia adelante (forward traceability) en el marco de LEL, escenarios y tarjetas CRC. El mismo se implementará en Baseline Mentor Workbench, que se entrega adjunta en un CD-ROM.

Esta tesis está organizada de la siguiente manera. En el capítulo 2 se describe la importancia de los requerimientos y la de hacer forward traceability de ellos por todo el ciclo de vida. En el capítulo 3 se describen el LEL y los escenarios, dos elementos del Client Oriented Requirements Baseline que permiten modelar el contexto de un dominio. También se describen las heurísticas para obtener tarjetas CRC. En el capítulo 4 se describe el Baseline Mentor Workbench junto con sus evaluaciones sobre la usabilidad y el algoritmo de derivación implementado. En el capítulo 5 se describe el análisis de la vinculación entre LEL, escenarios y tarjetas CRC, y se detallan las reglas de traceability forward. Y en el capítulo 6 se describe la implementación de las reglas en el BMW.

2. Rol de los requerimientos en el ciclo de vida del software

Los requerimientos son importantes en el desarrollo del software puesto que determinan la calidad. Por ello debe hacerse una gestión adecuada de los requerimientos y utilizarse un esquema de traceability para seguirles el rastro a lo largo de todo el ciclo de vida de desarrollo del software.

En este capítulo se describe la importancia de los requerimientos en la sección 2.1, la importancia de la gestión de requerimientos en la sección 2.2, el concepto de traceability en la sección 2.3, la importancia de traceability en la sección 2.4 y la importancia de forward traceability en la sección 2.5.

2.1. Los requerimientos como condicionantes de la calidad del software

La importancia de los requerimientos radica en que la calidad del software depende de ellos. Garvin [Garvin 1984] define calidad como: "un concepto complejo y multifacético que puede describirse desde cinco perspectivas":

- ❖ Visión trascendental: Algo que puede ser reconocido pero no definido.
- ❖ Visión del usuario: La calidad se adecua a un propósito.
- ❖ Visión de manufactura: Calidad como conformidad con una especificación.
- ❖ Visión del producto: Calidad atada a características inherentes del producto.
- ❖ Visión basada en el valor: La calidad depende del monto que esté dispuesto a pagar el cliente.

Dos de las cinco perspectivas hacen referencia a los requerimientos. Tanto la visión del usuario como la de la manufactura relacionan calidad con los requerimientos.

Davis coloca a los requerimientos en su diagrama del ciclo de vida del software [Davis 1993] como a la materia prima para realizar la planificación de las pruebas del sistema de software (figura 1).

Boehm [Boehm 2001] establece el costo relativo de reparar errores. Este costo, tiene el valor más bajo en la etapa de requerimientos y crece en forma exponencial en las sucesivas etapas (tabla 1).

Mizuno organizó los errores que se pueden producir a lo largo del desarrollo del software en un modelo de catarata de errores [Mizuno 1983]. El modelo muestra que una especificación incorrecta de requerimientos puede ocasionar errores ocultos dentro del sistema (figura 2).

Las palabras de Ackoff [Ackoff 1974] son las que mejor sintetizan la importancia de los requerimientos: "Fallamos más a menudo porque resolvemos el problema incorrecto, que porque realizamos una mala solución del problema correcto." ("We fail more often because we solve the wrong problem than because we get the wrong solution to the right problem.").

2.2. Importancia de la gestión de requerimientos

Los requerimientos son dinámicos. Como la realidad cambia, el software que se debe adaptar a ella también lo hace, por lo tanto lo hacen los requerimientos de los clientes.

Es necesario mantener concordancia entre los requerimientos cambiantes del cliente y los que utiliza el equipo de desarrollo. Esta sincronización se conoce como gestión de requerimientos [Leffingwell 2000].

La gestión, al igual que los requerimientos, determinan la calidad del software. Por este motivo es un punto que aparece en estándares y guías de procesos de la ingeniería de software.

Paulk determina que uno de los motivos que lleva a que los proyectos de software fracasen es la incorporación ad-hoc de requerimientos volátiles [Paulk 1999]. Esto trae consecuencias indeseables. En primer lugar las estimaciones de tiempo y costo carecen de sentido, ya que nuevos requerimientos alteran los cálculos iniciales. Además, no se puede seguir el rastro a lo largo del proyecto a los requerimientos que no fueron formalmente aceptados.

Distintos estándares reparan en la importancia de la gestión de requerimientos. Tanto ISO/IEC 12207, como MIL-STD-498 y DOD-STD-2167A mencionan la gestión de requerimientos como un proceso clave.

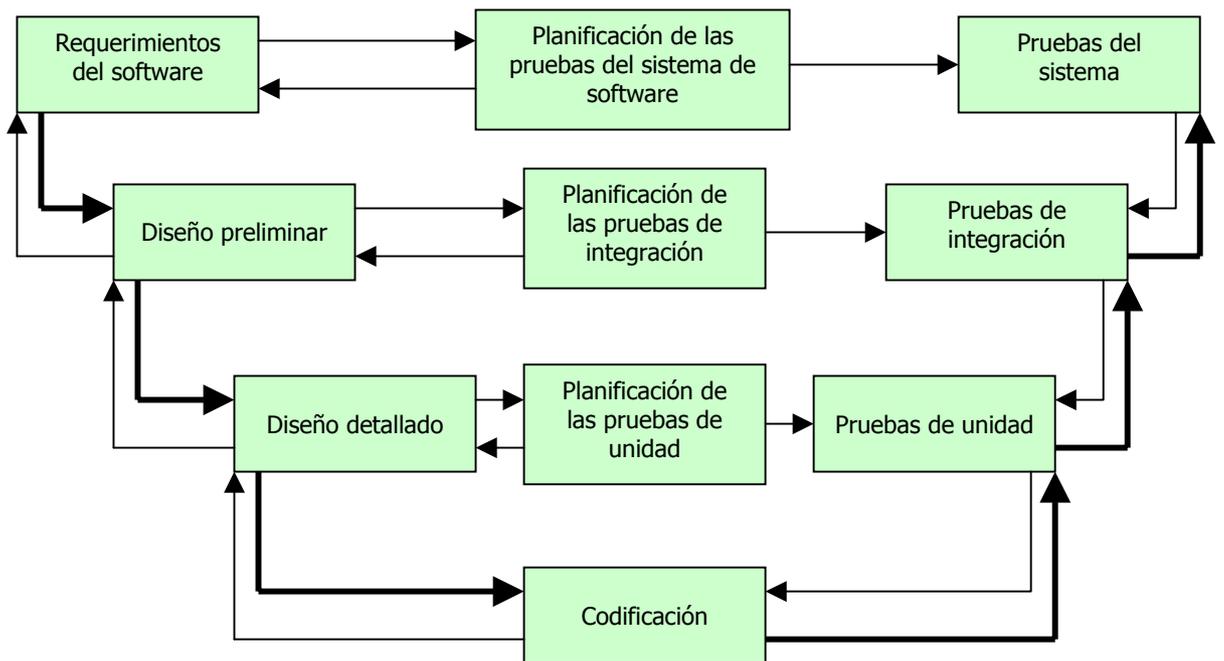


Figura 1 - Diagrama del ciclo de vida del software propuesto por Davis.

Tabla 1 – Costo relativo de reparar errores a lo largo del ciclo de vida del software.

Etapa	Costo relativo de reparación
Requerimientos	0.1-0.2
Diseño	0.5
Codificación	1
Test de unidades	2
Test de aceptación	5
Mantenimiento	20

Pero es sin duda el capability maturity model for software (CMM) quien mayor importancia le da al tema. CMM divide a la madurez del proceso de producción de software en 5 niveles. El nivel 1 es el nivel inicial, el más bajo en que se puede ubicar a una organización por más caos que haya. El nivel 2 presenta las primeras disciplinas para mejorar el proceso. En el nivel 2 es donde se menciona como área clave de proceso a la gestión de requerimientos. Se la ubica como primer paso para llegar a un proceso de desarrollo disciplinado.

2.3. Concepto de traceability

Traceability es un atributo de la **especificación de requerimientos de software (SRS)**. El estándar IEEE 830-1984 lo define como:

"Una especificación de requerimientos es traceable si (i) el origen de cada uno de sus requerimientos está claro y si (ii) facilita la referencia de cada requerimiento en artefactos de desarrollo futuro o documentación" [ANSI/IEEE 1984]

La definición de traceability de este estándar, que se reconoce como la más difundida [Gotel 1994], claramente se concentra en la SRS y requiere propiedades de ella. En primer lugar, que permita remontarse hacia atrás en el origen de los requerimientos y que también posibilite referenciar a cada requerimiento en la documentación de desarrollo.

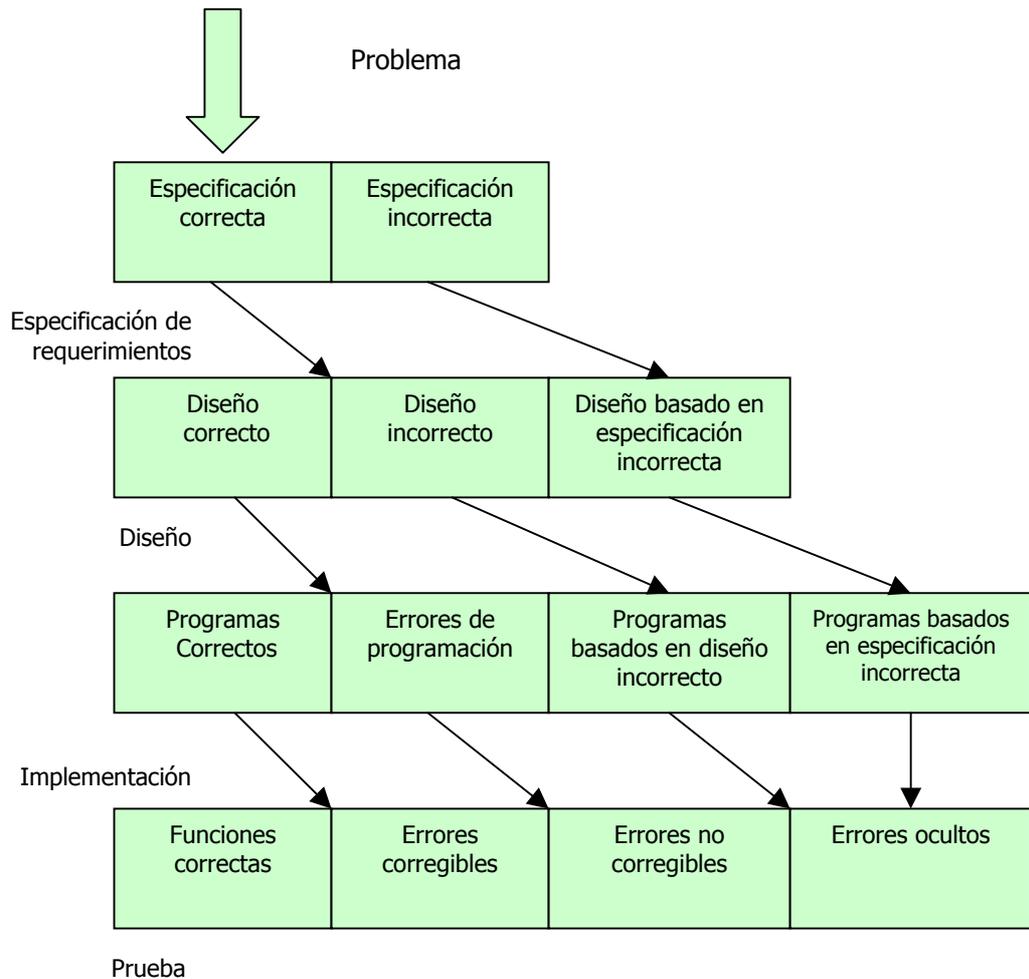


Figura 2 - Catarata de errores de Mizuno.

Por su parte Wieringa [Wieringa 1995] propone una generalización del concepto, extendiéndolo más allá de la especificación de requerimientos de software hasta abarcar todos los documentos que se generan a lo largo del ciclo de vida:

“Los documentos creados durante el desarrollo inicial y actualizados durante todo el ciclo de vida del producto se llaman rastreables si cada parte de cada documento puede ser rastreado a cualquier otra parte de cualquier otro documento relevante, de tal forma que las partes que deben cambiar juntas, están vinculadas entre ellas. Los documentos vinculados pueden describir necesidades del usuario, requerimientos, diseño, implementación, contexto del sistema, etc.”

La generalización de Wieringa es doble. Por un lado traceability es un atributo que poseen otros documentos además de la SRS. Y esos documentos abarcan todo el **ciclo de vida de desarrollo de software** (SDLC).

Para Davis [Davis 1993] traceability son links de cuatro tipos posibles. Cada uno posee centro en los requerimientos que se dirigen desde y hacia las fuentes de requerimientos y especificaciones de diseño (figura 3).

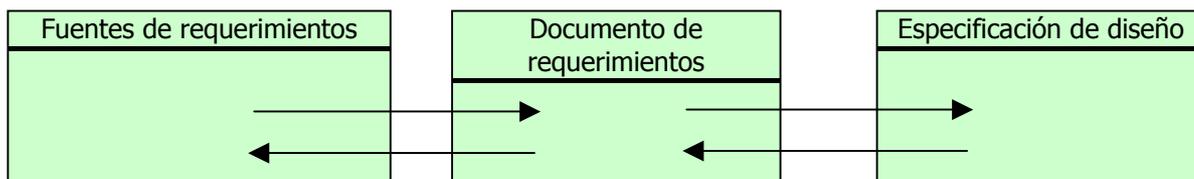


Figura 3 – Modelo de traceability de Davis.

Sommerville [Sommerville 1997] se encuentra en la misma línea que Kotonya [Kotonya 1998] y señala que el aspecto más importante de traceability de la información es el que registra las relaciones entre los mismos requerimientos:

“Información de traceability es información que permite encontrar dependencias entre los requerimientos, y por otro lado, entre requerimientos y componentes de diseño del sistema contra la documentación.”

La relación entre los mismos requerimientos es importante, pero cuando se avanza en el desarrollo, resulta de mayor relevancia para el ingeniero de software la posibilidad de establecer los vínculos de los requerimientos con los distintos elementos del sistema.

La conceptualización más rica, y que en cierta medida es la que inspira los conceptos de Wieringa [Wieringa 1995] es la de Gotel [Gotel 1994]. Traceability es un atributo del proceso que amplía la definición de IEEE para considerar los vínculos de los requerimientos con sus antecedentes y su evolución como tal:

“**Traceability de requerimientos (RT)** se refiere a la habilidad de describir y seguir la vida de los requerimientos en ambas direcciones hacia adelante y hacia atrás (forward and backward traceability). Desde los orígenes, pasando por el desarrollo y la especificación, hacia su posterior entrega y uso, y a través de todos los períodos de refinamiento e iteración de cualquiera de estas etapas.” [Gotel 1994]

Gotel [Gotel 1995] introduce una mayor precisión en las ideas de traceability hacia adelante o hacia atrás con los conceptos de traceability de requerimientos horizontal y vertical. **RT horizontal** es traceability a través de versiones de los requerimientos, en tanto que **RT vertical** es traceability entre las fases previas y posteriores del SDLC. Traceability hacia adelante o hacia atrás se refiere a la dirección en la que se realiza RT. En el caso de la RT vertical, traceability hacia atrás es desde los requerimientos hasta sus orígenes y hacia adelante es desde los requerimientos hasta su implementación. La RT horizontal consiste en avanzar desde una versión hacia versiones posteriores o retroceder hacia versiones anteriores.

Gotel [Gotel 1995] también hace la siguiente distinción en RT teniendo en cuenta la especificación de requerimientos (Requirement Specification, RS):

- ❖ Pre-RS, concierne a aquellos aspectos de la vida de los requerimientos anteriores a la inclusión en la RS.
- ❖ Post-RS, concierne a aquellos aspectos de la evolución de los requerimientos que resultan de su inclusión en la RS.

Pinheiro [Pinheiro 2000] considera traceability a partir de los “rastros” (traces) que son generadas en el proceso de desarrollo de software y que se manifiestan por diversas marcas en los productos de software. En ese contexto, traceability es:

“ la habilidad de definir, capturar y seguir los rastros dejados por los requerimientos en otros elementos del ambiente de desarrollo del software y los rastros dejados por aquellos elementos en los requerimientos”

2.4. **Importancia de traceability**

Kotonya [Kotonya 1998] coloca a los requerimientos en el contexto mayor de la gestión de cambios. Para Kotonya la evaluación del impacto del cambio en el resto del sistema es un aspecto crítico del proceso de **change management**. El cambio puede producirse en tres momentos:

- ❖ Durante el análisis de requerimientos, por lo cual se puede evaluar el efecto del cambio sobre los otros requerimientos.
- ❖ Durante la implementación. En este caso es necesario evaluar el impacto en los requerimientos, diseño e implementación.
- ❖ Durante la operación del sistema. En esta situación se requiere evaluar como son afectados los stakeholders además de los elementos del punto anterior.

La evaluación del impacto del cambio requiere información sobre:

- ❖ Dependencias de requerimientos.
- ❖ Fundamentos de los requerimientos.
- ❖ Implementación de los requerimientos.

Esto se denomina **traceability information**: "la evaluación del impacto del cambio depende de la información de traceability para descubrir que requerimientos son afectados por un cambio propuesto". Esto significa que la importancia reside en disponer de los elementos para ejecutar adecuadamente el **change management** y en especial evaluar el impacto de los cambios en las diferentes etapas del desarrollo de software.

Hay dos razones para traceability, según Davis [Davis 1993]:

- ❖ para diseñar o testear cualquier componente del sistema es necesario saber qué requerimientos satisface (aunque sea parcialmente).
- ❖ para testear el sistema de software es necesario saber qué requerimientos están siendo validados en cada **test**.

Por su parte, Sommerville plantea los beneficios de contar con políticas de traceability en dos guías básicas de la ingeniería de requerimientos (RE): **Define Traceability Policies** y **Maintain a Traceability Manual** [Sommerville 1997b]. Ellos son:

- ❖ Disponer de información en la evaluación de los cambios de requerimientos.
- ❖ Proveer la base para el control de costos y calidad.

Wiegiers [Wiegiers 1999] con un enfoque muy práctico de los requerimientos, encuentra los siguientes beneficios en traceability de los requerimientos:

- ❖ Seguimiento de la funcionalidad que se implementa.
- ❖ Ayudar a demostrar la implementación de todos los requerimientos (**certificación**).
- ❖ Analizar el impacto de los **cambios**.
- ❖ Ayudar al **mantenimiento**.
- ❖ **Reingeniería** de los **legacy systems**.
- ❖ Facilitar el reuso de componentes identificando paquetes de requerimientos, diseños, código, test y otros artefactos.
- ❖ Reducir el **riesgo** que provoca la gente clave que abandona el proyecto (ya que con traceability se conocen los links entre las componentes).
- ❖ **Testeo**

El **Software Engineering Body of Knowledge** menciona entre las actividades de la Ingeniería de Producto de Software al mantenimiento de traceability y consistencia entre los documentos producidos en el desarrollo [Hilburn 1999].

2.5. **Importancia de forward traceability**

Gotel [Gotel 1994] define RT:

"Traceability de requerimientos se refiere a la habilidad de describir y seguir la vida de los requerimientos en ambas direcciones hacia delante y hacia atrás (forward and backward traceability). Desde los orígenes, pasando por el desarrollo y la especificación, hacia su posterior entrega y uso, y a través de todos los períodos de refinamiento e iteración de cualquiera de estas etapas." [Gotel 1994]

En particular nos concentramos en traceability forward, esto es desde los requerimientos hacia la implementación. Wieringa [Wieringa 1995] sintetiza las ventajas que brinda forward traceability para las distintas personas involucradas en la producción de software.

❖ **Project manager**

Para estimar el impacto de un cambio en los requerimientos.

Para descubrir en forma temprana conflictos entre los requerimientos.

Para recoger requerimientos no satisfechos por la implementación y estimar el trabajo necesario para implementar estos requerimientos.

❖ **Cliente**

Para conocer en que parte de la implementación se satisfacen los requerimientos.

Para conocer que tests se deben realizar para validar la implementación de los requerimientos.

❖ **Diseñador**

Para verificar fácilmente que un diseño satisface los requerimientos.

Para estimar como un cambio de los requerimientos impacta en el diseño.

Para entender por qué razón un cierto diseño fue aceptado y otro rechazado aún cuando el diseño fue producido hace un largo tiempo por un diseñador no presente (estas razones pueden relacionar decisiones de diseño a requerimientos no funcionales).

Para reusar componentes de diseño en otros proyectos (ya que se registran los supuestos bajo los cuales la componente funciona).

❖ **Responsable de mantenimiento**

Para estimar el impacto del cambio de un requerimiento en otros requerimientos (descubrimiento de conflictos de dependencias).

Para estimar el impacto del cambio de un requerimiento en la implementación.

3. Modelo de LEL y escenarios

La investigación en el desarrollo de software centra sus esfuerzos en obtener nuevas técnicas y metodologías para realizar la solución, pero se le da poca importancia al problema en cuestión que debe ser inferido de la solución misma. Esta aproximación orientada a la solución puede servir en un área donde los problemas son bien conocidos. Sin embargo, no es el caso del desarrollo de software en donde el análisis del problema es imprescindible ya que la computación avanza a un ritmo muy acelerado y ataca dominios muy variados [Jackson 1999].

Por este motivo es necesario realizar un modelo contextual. El mismo se trata en la sección 3.1. Es imprescindible utilizar una técnica para realizar el modelo contextual que sea compatible con las habilidades y características humanas, para este fin se comenta un estudio sobre el conocimiento experto en el ser humano en la sección 3.2. Dos herramientas adecuadas para esta área son el modelo de LEL y escenarios, los cuales se describen en las secciones 3.3 y 3.4. Ambos elementos forman parte del Client Oriented Requirements Baseline descrito en la sección 3.5. A este modelo se le agregan las tarjetas CRC descritas en la sección 3.6 para contar con un modelo integrado que comienza en el modelo contextual y abarca hasta el diseño preliminar de objetos. Las relaciones entre LEL, escenarios y tarjetas CRC se describen en la sección 3.7.

3.1. Modelo contextual

Un ingeniero civil trabaja muy distinto de un ingeniero de software [Jackson 1995]. Un ingeniero civil lo primero que hace cuando se le pide la construcción de un puente es visitar el sitio. El ingeniero se para en una margen del río y mira todo a su alrededor: el paisaje de ambas márgenes y el río en sí. Aprecia cuan fuerte sopla el viento, cuan rápido se desplaza el caudal, el tráfico en las orillas y sobre el río. Mira al suelo y se imagina como será la composición geológica del terreno. Luego realiza una imagen mental del puente que será construido.

Este proceso constituye el análisis del contexto del problema. Como resultado de este análisis se desarrolla un modelo contextual. Distintos modelos se han propuesto. Jackson [Jackson 1999] propone un modelo basado en "**phenomenas**" (fenómenos). Como los problemas están localizados en el mundo, el análisis de problemas debe estar de acuerdo con el mundo y sus **fenómenos**. Plantea la necesidad de una "**phenomenología**" que no tenga nada que ver con lenguajes de programación o interacción entre objetos, pero que esté íntimamente relacionada con el mundo físico. Una "**phenomenología**" adecuada debe incluir:

- ❖ Entidades: individuos que evolucionan, por ejemplo autos o personas.
- ❖ Eventos: reconocidos por las entidades.
- ❖ Valores: individuos inmutables, por ejemplo enteros o cadenas de caracteres.
- ❖ Estados: relaciones que cambian con el tiempo sobre entidades.
- ❖ Verdades: relaciones invariables sobre entidades.
- ❖ Roles: participaciones de entidades en eventos.

Sin embargo un modelo contextual del problema por sí sólo no es suficiente. Junto con él se deben proveer los requerimientos de la solución que es demandada [Jackson 1995]. En el ejemplo del puente, si se presenta todo el modelo del área en donde se quiere construir un puente, pero se olvida mencionar el detalle de la construcción del puente, todo ese modelo es inútil. Y aunque se mencione la necesidad de la construcción del puente, este requerimiento es impreciso. El puente podría ser peatonal, para automóviles o para camiones. Podría tener un carril o varios. Podría ser fijo o levadizo (para permitir el tráfico fluvial en el río). Estos son algunos de los requerimientos necesarios para poder llegar a la solución deseada.

3.2. Organización del conocimiento experto

Wood [Wood 1997] analiza la organización del conocimiento experto en el cerebro humano. Describe que los conceptos en la memoria humana son asociados unos con otros y la experiencia de recordar consiste en recorrer estas asociaciones.

El conocimiento experto está generalmente organizado en jerarquías. En un macro nivel se presentan organizaciones taxonómicas, mientras que en un micro nivel el conocimiento experto se liga

a patrones de acciones. Wood denomina a los elementos del macro nivel objetos y a los del micro nivel procesos.

El conocimiento de objetos incluye las entidades (concretas y abstractas) de un dominio particular con sus variadas categorías y relaciones. Relaciones taxonómicas como tipos, subtipos y rasgos identificatorios o características de objetos son particularmente importantes.

El conocimiento de procesos es el conocimiento requerido para cumplir con un trabajo utilizando los objetos relevantes.

3.3. **Lexico Extendido del Lenguaje**

El Léxico Extendido del lenguaje (LEL) es un modelo contextual que permite capturar el lenguaje de un dominio. A través de la identificación y definición de los símbolos propios de un contexto se logra un mejor entendimiento de éste. La idea bajo el LEL es muy simple, hay que preocuparse por entender el lenguaje del problema, sin preocuparse por entender el problema [Leite 1997].

El LEL es un especie de glosario, ya que su objetivo principal es registrar signos (palabras o frases) los cuales son peculiares a un dominio. A diferencia del diccionario tradicional en donde cada término tiene un sólo tipo de definición, en el LEL cada signo es descrito de dos formas: a través de la noción y de los impactos.

La noción es la descripción del tipo usual que da el diccionario. Es la descripción del signo por medio de sus propiedades intrínsecas. Mientras que los impactos determinan como el signo descrito se relaciona con los demás. Los impactos describen la incidencia del signo en los demás o la de los demás en el signo.

La construcción del LEL está regida por dos principios. El principio de circularidad y el principio de vocabulario mínimo. El principio de circularidad establece que en la descripción de la noción e impactos se debe maximizar el uso de signos definidos en el LEL. El principio de vocabulario mínimo, complementa el principio de circularidad, y establece que cuando se utilizan signos externos al LEL deben tener una representación matemática clara (por ejemplo: pertenencia, intersección, etc).

Ambos principios determinan que el conjunto de signos sea autocontenido. Es decir que un signo esté expresado en término de otros. De esta forma los signos están interrelacionados. Si cada signo se ve como un nodo de información, junto con la relación entre ellos se puede percibir un grafo. Ahora, como cada nodo está conformado por texto, el grafo no es otra cosa que un hipertexto [Leite 1997].

3.3.1. **Ejemplo**

Se describen 6 signos extraídos del dominio de un organismo gubernamental de la República Argentina que se encarga de la ejecución de deudas contra el fisco.

Este organismo lleva adelante un proceso judicial contra los ciudadanos de la provincia de Buenos Aires que estén en mora con el pago de sus impuestos. El primer paso consiste en notificar al contribuyente sobre la situación de que la provincia ha dejado de recibir sus pagos, con el fin de aclarar cualquier error o para proveerle un mecanismo de facilidades de pago. Si el cese de pagos fue por propia voluntad y hay un rechazo por parte del contribuyente en continuar con los pagos, se inicia un juicio al contribuyente con el fin de obligar a regularizar su situación. Un representante del fisco (un abogado) es quien lleva adelante todo este proceso.

Cada símbolo de LEL se describe de acuerdo al modelo de la tabla 2.

Tabla 2 – Modelo de descripción de una entrada de LEL.

Sinónimos	Identifica la entrada del LEL
Noción	Se describe que es el símbolo
Impacto	Se describe como repercute en el sistema

Los símbolos de LEL no se identifican solamente por una palabra, varias palabras se pueden utilizar para representar la misma idea. Éste es el concepto de sinónimos. Cada signo o símbolo está representado por un conjunto de sinónimos. Cada uno de los sinónimos es separado de los demás por medio de una barra inclinada “/”.

Las entradas de LEL para este ejemplo son "abogado" (tabla 3), "cédula de notificación / cédula" (tabla 4), "intimación" (tabla 5), "intimación negativa" (tabla 6), "juicio" (tabla 7), "particulares / razón social / demandado" (tabla 8).

Tabla 3 – Entrada de LEL "abogado".

Sinónimos	Abogado
Noción	Representante legal del fisco
Impacto	Lleva adelante los <u>juicios</u> del fisco contra los <u>particulares</u> . Arma las <u>cédulas de notificación</u> para enviar a los <u>demandados</u>

Tabla 4 – Entrada de LEL "cédula de notificación".

Sinónimos	Cédula de notificación / cédula
Noción	Medio por el cual se notifica al <u>demandado</u> de su deuda contra el fisco.
Impacto	Si la <u>cédula de notificación</u> es recibida al juicio se le anexa el movimiento de <u>intimación</u> . Si la <u>cedula de notificación</u> no es recibida al juicio se le anexa el movimiento de <u>intimación negativa</u> .

Tabla 5 – Entrada de LEL "intimación".

Sinónimos	Intimación
Noción	Acción de notificar a una <u>razón social</u> de su deuda contra el fisco.
Impacto	La <u>razón social</u> toma conocimiento de su deuda contra el fisco. La <u>razón social</u> recibe una <u>cédula de notificación</u> del <u>abogado</u> que lleva el <u>juicio</u> .

Tabla 6 – Entrada de LEL "intimación negativa".

Sinónimos	Intimación negativa
Noción	Situación en la cual la <u>razón social</u> no recibe la <u>cédula de notificación</u> .
Impacto	El <u>juicio</u> no puede seguir adelante hasta que no se realice la <u>intimación</u> .

Tabla 7 – Entrada de LEL "juicio".

Sinónimos	Juicio
Noción	Acción legal por la cual se obliga a una <u>razón social</u> a pagar una deuda contra el fisco.
Impacto	Se obliga a la <u>razón social</u> a pagar la deuda contra el fisco.

Tabla 8 – Entrada de LEL "particulares".

Sinónimos	Particulares / Razón social / Demandado
Noción	Entidad a la que se le inicia un <u>juicio</u> .
Impacto	Tiene que abonar una deuda contra el fisco.

En las entradas de LEL del ejemplo se muestra el principio de circularidad. Los símbolos de LEL que se utilizan para definir a otros símbolos se los muestra subrayados y en color rojo (ejemplo). La figura 4 muestra las referencias desde cada entrada de LEL hacia las demás.

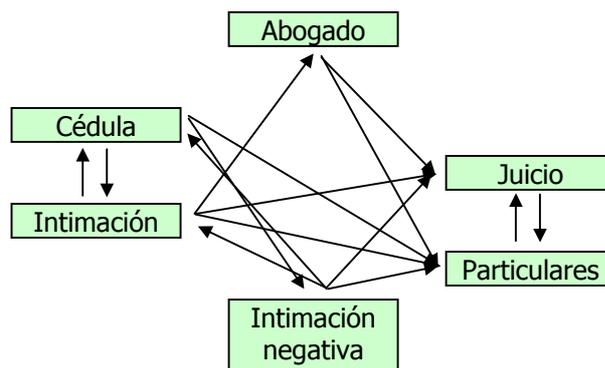


Figura 4 – Hipertexto de las entradas de LEL de un organismo gubernamental de la República Argentina.

3.3.2. Proceso de construcción de LEL

La construcción de LEL comienza por obtener información del dominio. A partir de esta información se elabora una lista de símbolos que se deben conocer para entender al lenguaje del dominio. Estos símbolos se deben clasificar para poder definirlos en forma consistente. Luego de clasificarlos, se los definen y como producto de la definición se pueden descubrir sinónimos, por lo cual se deben reorganizar los símbolos. La información debe ser validada por los expertos del dominio y controlada por el ingeniero de requerimientos. Si algún nuevo símbolo debe ser definido, se repite el proceso.

Los pasos del proceso son los siguientes:

Identificación de las fuentes de información

Las fuentes de información para la construcción de LEL son dos: las personas y los documentos. Cada uno tiene ventajas sobre el otro. Los documentos ofrecen un medio concreto, autocontenido e invariable. El ingeniero de requerimientos puede leer y volver a leer los documentos para una selección minuciosa de los términos. Los textos se pueden analizar en detalle. Por otra parte, las personas pueden aportar a través de entrevistas mucha mayor información. Una característica propia de la expresión oral de los seres humanos es que en forma inconsciente utilizan el principio de circularidad. En cambio en la descripción escrita, para una narrativa más entretenida se introducen muchos sinónimos, lo que puede dificultar el proceso de identificación de símbolos. Es aconsejable utilizar ambas fuentes de información.

Las personas que es aconsejable entrevistar son:

- ❖ las más mencionadas dentro del entorno del sistema.
- ❖ las que toman las decisiones.
- ❖ los responsables del proyecto.
- ❖ los supervisores del proceso.

Por su parte, los documentos que más información brindan son:

- ❖ descripciones textuales del sistema.
- ❖ manuales que definen los procedimientos operacionales.

Generación de la lista de símbolos

Una estrategia aconsejable para generar la lista de símbolos consiste en comenzar con una lectura de la documentación para hacer un análisis preliminar del lenguaje del dominio. Con este análisis se obtiene una lista inicial y se adquiere información para poder llevar a cabo entrevistas. Es aconsejable realizar entrevistas semiestructuradas y estructuradas para acotar el lenguaje. Ya que si se comenzara con entrevistas libres, el volumen de información sería muy difícil de manejar.

Los símbolos que se deben tomar para la lista inicial están dados por palabras o frases que se utilizan con mucha frecuencia, o aquellas que parezcan estar fuera de contexto. El motivo de elegir las palabras o frases que se utilizan con mucha frecuencia es claro. El objetivo del LEL es capturar el lenguaje de un dominio. Los términos más utilizados son significativos en el dominio, por lo tanto

deben estar definidos. En cambio los términos que parezcan estar fuera de contexto se deben a que tienen un significado propio en el dominio, distinto del tradicional. Estos términos, con más razón deben estar definidos.

Cabe destacar que los símbolos no tienen que ser palabras individuales. Pueden ser palabras o frases. La razón es que en un lenguaje se puede encontrar un grupo de palabras, en donde cada una de ellas pueda tener cierto significado, pero si se las combina en una frase tienen otro significado.

Si bien en la etapa de definición es cuando se encuentran los sinónimos, la tarea puede comenzar en esta etapa.

Clasificación de la lista de símbolos

El objetivo de categorizar los símbolos es poder administrar mejor al conjunto. Además, cada categoría determina la forma en que se deba definir cada símbolo. Así se logra una definición consistente y uniforme.

Para clasificar los símbolos se parte de una clasificación general:

- ❖ Sujeto: Elemento activo dentro del dominio que realiza acciones utilizando objetos. El sujeto (persona, maquina, dispositivo) puede llegar a pasar por distintos estados.
- ❖ Verbo: Acción que realiza un sujeto, servicio que brinda un objeto, desencadenante para pasar de un estado a otro.
- ❖ Objeto: Elemento pasivo con los cuales se realizan acciones que puede pasar por distintos estados.
- ❖ Estado: Situación en la que se encuentra un sujeto o un objeto.

Esta es una clasificación inicial que en función del dominio se puede especializar. La especialización puede darse por ejemplo en los sujetos. Un sistema de administración de empleados de una empresa tiene distintos tipos de empleados. Aquellos en actividad, en condición de retiro y retirados. Y estos a su vez pueden contener subcategorías, por lo cual se necesita explotar la categoría inicial.

Esta categorización sirve para agrupar los símbolos relacionados, lo que permite encontrar sinónimos.

Además, la clasificación permite construir un LEL uniforme. Para cada categoría se debe definir su forma de describir la noción e impacto de los símbolos. Así se asegura que los símbolos son descriptos consistentemente y que es posible contrastarlos, para un mejor entendimiento del lenguaje. Por ejemplo algunas pautas para tener en cuenta en la descripción de los símbolos son las siguientes:

- ❖ Para los signos que son sujetos los impactos deben indicar las acciones que realiza.
- ❖ Para los signos que son verbo las nociones deben decir quien ejecuta la acción, cuando sucede y el proceso involucrado en la acción. Para los impactos se deben identificar las restricciones sobre la realización de la acción, que origina esta acción y que es lo que causa esta acción.
- ❖ Para los signos que son objetos la noción debe identificar otros objetos con los cuales se relaciona. Y los impactos serán las acciones que se pueden realizar con este signo.

Estas reglas deben ser extendidas en la medida que la categorización inicial lo sea.

Descripción de símbolos

Los símbolos se describen a partir del conocimiento obtenido de la lectura de la documentación y de las entrevistas. Cada símbolo se describe siguiendo las pautas establecidas en la clasificación de los términos. Esta descripción inicial luego será validada y controlada.

Algunas pautas generales para la descripción de los símbolos, independientemente de la clasificación a la que corresponden, son las siguientes:

- ❖ Un signo puede tener una o más nociones y cero o más impactos.
- ❖ Cada noción e impacto debe ser descripto con oraciones breves y simples.
- ❖ Las oraciones deben responder a los principios de circularidad y de vocabulario mínimo.

- ❖ Las nociones e impacto de un signo pueden representar diferentes puntos de vista o pueden ser complementarios.

Con la descripción de símbolos se facilita la tarea de encontrar sinónimos. Sin embargo, en la validación posterior, se despeja cualquier duda respecto de los sinónimos.

Validación

Consiste en verificar la correctitud del LEL contra el usuario. Debido a la gran cantidad de símbolos que pueden estar definidos en el LEL, es impracticable realizar una validación completa y exhaustiva. Sin embargo, es posible mantener sesiones de entrevistas estructuradas para aclarar dudas.

Control

El proceso de control es el que realiza el ingeniero de requerimientos por sus propios medios. El control no tiene que ver con la correctitud de la información del LEL, por el contrario está relacionado con la estructura:

- ❖ Todos los símbolos deben estar definidos. Aquellos de la lista inicial y también los que aparecen en la descripción de los signos de la lista inicial.
- ❖ Todos los símbolos deben estar dentro de la clasificación correspondiente. Debido a que la clasificación se realiza antes de la descripción, en esta última se puede descubrir que un símbolo se colocó en la categoría incorrecta, por lo cual se lo debe reubicar y verificar su definición.
- ❖ La descripción de los símbolos debe corresponderse con la de la categoría a la que pertenecen.
- ❖ El punto de vista para la descripción de los símbolos debe ser uniforme.
- ❖ No deben dejarse sinónimos definidos como signos independientes.

3.4. Modelo de escenarios

Los aspectos dinámicos del modelo del contexto se pueden capturar a través de los escenarios. Muchos modelos se han realizado: Jacobson con sus Use cases ([Jacobson 1994] y [Jacobson 1994b]) y también Potts, Booch, Zorman, Rubin y Firesmith han hecho sus contribuciones ([Potts 1994], [Potts 1995], [Booch 1994], [Zorman 1995], [Rubin 1992] y [Firesmith 1994]).

Sin embargo, siguiendo el trabajo de Leite [Leite 1997] utilizamos su modelo de escenarios el cual es una combinación de las ideas presentadas por Zorman, Jacobson, Rubin y Potts. Sus características principales son:

- ❖ Un escenario describe situaciones con énfasis en la descripción del comportamiento.
- ❖ Un escenario utiliza la descripción textual como representación básica.
- ❖ Un escenario está naturalmente ligado al LEL al describirse con el vocabulario definido en el este último.

Los atributos que definen a un escenario son: título, objetivo, contexto, recursos, actores y episodios. Título, objetivo, contexto, recursos y actores son oraciones declarativas, mientras que episodios es un conjunto de oraciones de acuerdo con un lenguaje muy simple que hace posible la descripción operacional del comportamiento. Las características de cada atributo son:

- ❖ Título: Es la identificación del escenario. En el caso de un subescenario el título es el mismo que la oración del episodio desde el cual es referenciado.
- ❖ Objetivo: La meta que debe ser alcanzada.
- ❖ Contexto: Describe el estado inicial del escenario.
- ❖ Recursos: Son los elementos con los cuales se llevará a cabo el escenario.
- ❖ Actor: Es una persona o objeto que lleva adelante el escenario.
- ❖ Episodios: Se forman con una serie de oraciones que detallan el comportamiento del escenario. Cada oración indica una tarea en la secuencia del escenario. Hay un par de símbolos especiales. El símbolo “#” se utiliza para indicar acciones no secuenciales.

Cuando no importa la secuencia en un grupo de pasos se los debe encerrar entre "#". Para indicar tareas condicionales se utiliza **if then** con la semántica tradicional.

Los atributos contexto, recursos y episodios poseen modificadores como ser restricciones y excepciones. Restricciones indican condicionantes. Por ejemplo, un recurso no puede usarse en tal situación. Las excepciones plantean casos alternativos. Por ejemplo los episodios se desarrollan siempre en cierta secuencia, sin embargo, excepcionalmente pueden llevarse a cabo otras tareas. La figura 5 muestra un diagrama entidad relación con la estructura de un escenario.

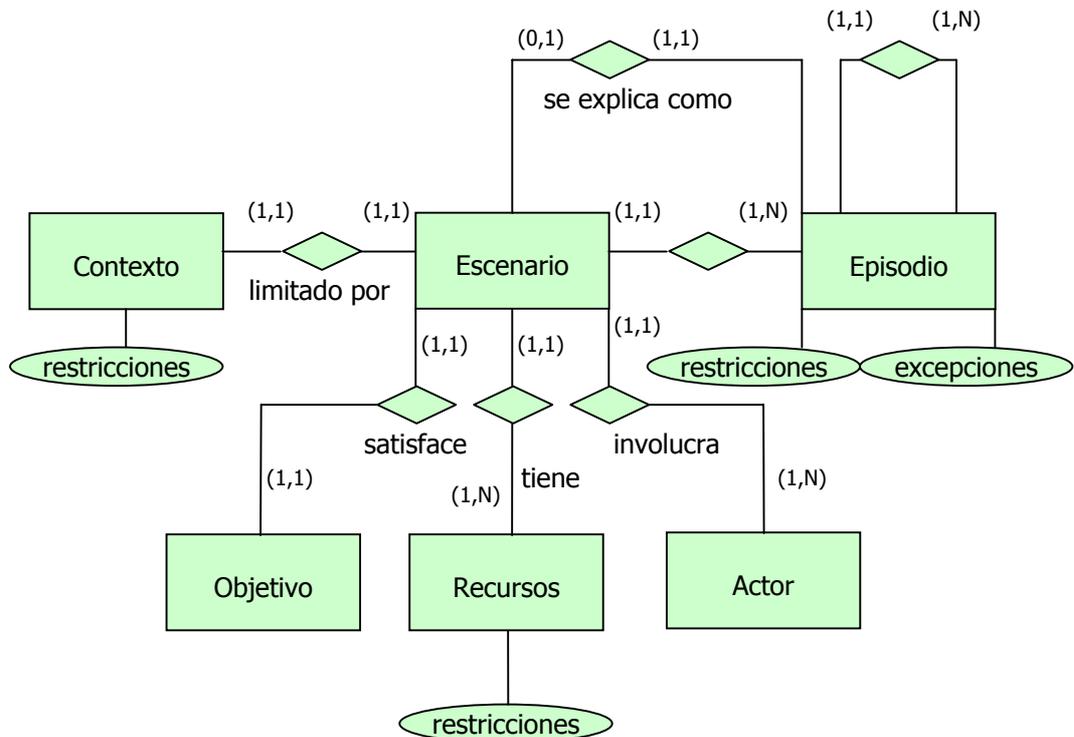


Figura 5 – Diagrama entidad relación de la estructura de un escenario.

De la misma forma que en el LEL los símbolos se relacionan entre ellos dando origen a un hipertexto, los escenarios también se relacionan entre ellos y con el LEL. Existe una relación estructural entre los escenarios. Un escenario puede ser subescenario de otro. Es decir, en un episodio puede aparecer una tarea lo suficientemente compleja, como para necesitar explicarla como otro escenario. Además, los escenarios se describen con texto que está definido en el LEL. Por lo tanto, los escenarios se ligan naturalmente a los símbolos del LEL.

3.4.1. Ejemplo

A partir del mismo dominio planteado en el ejemplo de LEL, se extraen dos escenarios: “**Confeccionar una cédula de notificación**” (tabla 9) e “**intimar una razón social**” (tabla 10).

Las referencias a signos del LEL se indican subrayadas y en color rojo ([ejemplo](#)). Y las referencias a escenarios se muestran subrayadas y en color azul ([ejemplo](#)). La figura 6 muestra el conjunto completo de entradas de LEL y escenarios, y todas las relaciones entre ellos.

Tabla 9 – Escenario “Confeccionar una cédula de notificación”.

Título	Confeccionar una <u>cédula de notificación</u>
Objetivo	Redactar la carta que recibirá el <u>demandado</u>
Contexto	Estudio jurídico
Recursos	Expediente del <u>juicio</u> , <u>cedula de notificación</u>
Actores	<u>Abogado</u>

Episodios	El abogado toma los datos del demandado del expediente del juicio El abogado completa los datos de la cédula
------------------	--

Tabla 10 – Escenario “Intimar a una razón social”.

Título	Intimar una razón social
Objetivo	Notificar a la razón social de la deuda contra el fisco
Contexto	Estudio jurídico
Recursos	Expediente del juicio , cedula de notificación
Actores	Abogado , mensajero
Episodios	El abogado confecciona una cédula de notificación para una razón social El abogado envía la cedula de notificación por correo If la carta es recibida then el abogado anexa al expediente el oficio de intimación If la carta no es recibida then el abogado anexa al expediente el oficio de intimación negativa

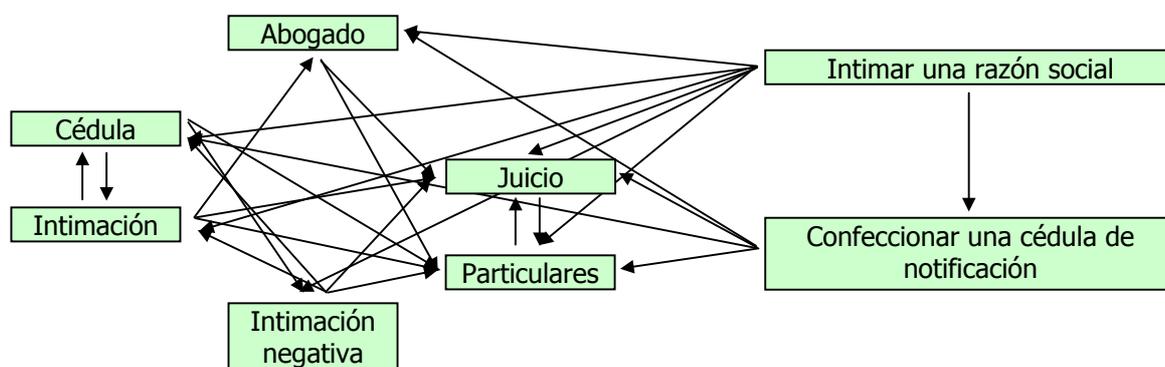


Figura 6 – Hipertexto de LEL y escenarios de un organismo gubernamental de la República Argentina.

3.4.2. Proceso de construcción de escenarios

El proceso de construcción de escenarios toma como punto de partida al LEL. Se toman los símbolos que caen dentro de la categoría de sujetos y se toman sus impactos como potenciales escenarios. Una vez descriptos los escenarios se deben revisar y validar. El proceso es el siguiente:

Identificación de los sujetos

Los símbolos de LEL que caen en la categoría de sujetos corresponden a entidades activas que realizan tareas dentro del dominio. Los impactos del LEL de un símbolo sujeto deben capturar las acciones que éste realiza. Estas acciones son candidatas a convertirse en escenarios.

En la etapa de identificación de sujetos sólo hay que concentrarse en tomar todos los símbolos que caen en la categoría sujeto o en alguna categoría derivada de sujeto si es que se extendió.

Luego hay que dividir la lista en actores principales y secundarios. La calificación de principal y secundario lo determina la relevancia del sujeto dentro del dominio de la aplicación. Esta categorización tiene como fin una mejor organización de los escenarios.

Generación de la lista de escenarios candidatos primarios

A partir de los actores principales se extraen del LEL sus impactos. Cada impacto es un escenario potencial. Se nombra cada escenario con la acción en infinitivo contenida en el impacto. De esta lista inicial hay que eliminar los escenarios repetidos, ya que distintos actores pueden realizar la misma acción.

Descripción de los escenarios candidatos primarios

A partir de la información de la entrada de LEL se pueden cubrir todos los atributos del escenario. El objetivo del escenario está muy relacionado con el título. Es básicamente cumplir con la acción enunciada en el título.

En el contexto se deben indicar las precondiciones y la situación espacio-temporal en que desarrolla el escenario. Esta información se puede extraer de los impactos del sujeto que originó el escenario. Los impactos previos al del escenario en cuestión pueden dar pautas de precondiciones necesarias.

Los recursos son objetos con los que se desarrolla el escenario. La lista de recursos se puede obtener de entradas de LEL de la categoría objeto (o sus derivados) referenciadas por el sujeto que originó el escenario.

Actor es el sujeto del cual se extrajo de sus impactos el escenario candidato en cuestión. Si varios actores coinciden en el escenario, todos ellos deben integrar la lista.

Los episodios se completan con una descripción para que el actor o actores, a partir del contexto inicial y utilizando los recursos logren el objetivo.

Ampliación de la lista de escenario candidatos

A partir de la lista de actores secundarios se realiza el mismo proceso que con los actores primarios.

Descripción de los escenarios candidatos secundarios

Se realiza de la misma forma que se hizo con los escenarios candidatos primarios.

Revisión de los escenarios

Consiste en factorizar el conjunto de escenarios. Se pueden obtener nuevos escenarios como producto de uno o más episodios dentro de un escenario. O inversamente, dos o más escenarios pueden tener episodios u objetivos comunes, por lo cual es conveniente unirlos.

Validación de los escenarios

Consiste en confrontar contra el experto del dominio la información en los escenarios, al igual que se hace con el LEL.

3.5. *Client Oriented Requirements Baseline*

LEL y escenarios son dos herramientas que conforman el Client Oriented Requirements Baseline propuesto por Leite [Leite 1997].

Este modelo tiene como objetivo modelar el contexto. El LEL se ocupa de capturar el lenguaje del dominio tan importante para poder entender el dominio en sí. Y los escenarios se encargan de los aspectos dinámicos del mismo.

La ventaja de este modelo por sobre los demás, es que LEL y escenarios se adecuan a la forma en que se organiza el conocimiento del experto en el ser humano [Wood 1997]. Wood clasifica el conocimiento experto en organizaciones taxonómicas de alto nivel u objetos, y conocimiento procedural de bajo nivel o procesos. Los objetos claramente se plasman en el LEL. La relación existente entre los distintos objetos que se da en el cerebro humano, está explícita entre las entradas de LEL a través del principio de circularidad. De la misma forma que se ligan los conceptos, se ligan las entradas de LEL originando un hipertexto. El conocimiento procedural está plasmado en los escenarios, ya que ellos capturan los aspectos dinámicos.

El hecho de que LEL y escenarios se adecuen a la forma en que se organiza el conocimiento en el ser humano, hace que sea mucho más fácil para el experto presentar la información y para el ingeniero de requerimientos obtenerla y modelarla.

Además, LEL y escenarios son descriptos con texto plano, a diferencia de otras aproximaciones que incorporan gráficos, lógicas de primer orden o notaciones especiales como en [Jackson 1999b], [Jackson 2000] o [Welty 1995]. Utilizar texto facilita la comunicación entre el ingeniero de requerimientos y el experto del dominio, ya que es un medio de representación entendible por ambos. Este punto es sostenido por el análisis presentado en la sección 3 (fuentes de requerimientos). Los productos favoritos, de entre todos los mencionados para la recolección de requerimientos, son los textuales. A pesar de que los diagramas o especificaciones formales son más precisas, la mayoría de los encuestados prefieren el texto narrativo.

También concuerda con las encuestas la forma en que LEL y escenarios son construidos. Se parte fundamentalmente de los expertos del dominio y de la documentación textual. Las encuestas

colocaron a estos procedimientos en primer lugar. Se prefiere realizar entrevistas y análisis de la documentación existente por sobre otras técnicas de elicitación.

Como última ventaja, Client Oriented Requirements Baseline es una forma natural de dividir el problema en piezas mas pequeñas, como se indica su importancia en [Jackson 1995b]. La forma es natural, porque se corresponde con como se organiza la información en el cerebro.

3.6. Tarjetas CRC

A partir de las entradas de LEL y los escenarios se pueden obtener tarjetas CRC. Estas tarjetas identifican posibles clases para diseñar una solución orientada a objetos.

3.6.1. Generalidades

Las tarjetas CRC (colaboraciones y responsabilidades de las clases) fueron introducidas por primera vez por Kent Beck y Ward Cunningham en 1989 en la conferencia de "Sistemas, lenguajes y aplicaciones orientados a objetos". A partir de allí, atrajo la atención de notables desarrolladores de software como Grady Booch, Ed Yourdon, Rebecca Wirfs-Brock y Adele Goldberg.

Cada tarjeta representa un objeto del mundo real. Las CRC deben su nombre a que cada tarjeta identifica una clase. En ella se identifican las acciones que realiza: sus responsabilidades. Las tarjetas CRC también registran las relaciones con otras clases: sus colaboraciones.

Muchos especialistas del paradigma orientado a objeto manifiestan que identificar objetos es una tarea simple e intuitiva, mientras otros sostienen que esto no siempre es verdadero, particularmente con aplicaciones de gran escala.

La técnica de *role playing* ataca este problema. Consiste básicamente en jugar con las tarjetas disponiéndolas sobre una mesa, de la misma forma que en el juego de *scrabel* se cambia el orden de las fichas para buscar nuevas palabras ([Wilkinson 1995] y [Bellin 1997]).

Las tarjetas CRC descriptas anteriormente sirven para elicitación de conocimiento de un dominio y se las conoce como CRC de análisis. Estas tarjetas se pueden transformar en CRC de diseño y utilizarse en un diseño preliminar de objetos.

3.6.2. Reglas de derivación de tarjetas CRC

Leonardi propone una estrategia para obtener tarjetas CRC a partir de LEL y escenarios [Leonardi 2001]. La estrategia consiste básicamente de tres tareas. Se identifican CRC primarias, se identifican CRC secundarias y se determinan las colaboraciones entre todas las CRC.

Encontrar CRC primarias

Las tarjetas CRC primarias representan objetos con comportamiento relevante dentro del dominio. Los actores de los escenarios son candidatos a convertirse en CRC primarias ya que realizan las tareas del escenario para lograr su objetivo.

Los actores son entradas de LEL de la categoría sujeto. Estos sujetos realizan acciones en el dominio, las que son descriptas en los impactos. Entonces, los impactos no son otra cosa más que los servicios que provee, estos servicios son las responsabilidades de la CRC.

De esta forma la tarjeta CRC es obtenida de un actor de un escenario y sus responsabilidades son los impactos de la entrada del LEL.

Encontrar CRC secundarias

CRC secundarias son aquellos colaboradores de las CRC primarias que las ayudan a cumplir con sus responsabilidades.

Las CRC secundarias se encuentran en las responsabilidades de las CRC primarias. Aquellos términos de las responsabilidades que también están definidos en el LEL se convierten en CRC secundarias. Las responsabilidades se obtienen de la misma forma que se obtienen las responsabilidades de las CRC primarias.

Encontrar colaboraciones

Debido a que las CRC secundarias se obtienen a partir de las responsabilidades de las CRC primarias, es trivial que colaboran en alguna medida. Sin embargo, para completar las colaboraciones, es necesario analizar los escenarios. En los episodios de los escenarios se deben buscar las entradas de LEL que originan a las CRC tanto primarias como secundarias. Las tarjetas CRC que participan de un mismo escenario colaboran entre ellas.

3.6.3. Ejemplo

A partir del dominio utilizado para los ejemplos de LEL y escenarios, y utilizando el algoritmo de derivación de tarjetas CRC, se obtienen cuatro tarjetas CRC: “**abogado**” (tabla 11), “**cédula de notificación**” (tabla 12), “**juicio**” (tabla 13) y “**particulares**” (tabla 14).

Tabla 11 – CRC primaria “abogado”.

CRC primaria	<u>Abogado</u>
Responsabilidades	Lleva adelante los <u>juicios</u> del fisco contra los <u>particulares</u> . Arma y envía las <u>cedulas de notificación</u> para enviar a los <u>demandados</u>
Colaboraciones	Juicio, <u>particulares</u> , <u>cedulas de notificación</u>

Tabla 12 – CRC secundaria “cédula de notificación”.

CRC secundaria	<u>Cédula de notificación</u>
Responsabilidades	Si la <u>cedula de notificación</u> es recibida al juicio se le anexa el movimiento de <u>intimación</u> . Si la <u>cedula de notificación</u> no es recibida al juicio se le anexa el movimiento de <u>intimación negativa</u> .
Colaboraciones	<u>Abogado</u> , <u>particulares</u>

Tabla 13 – CRC secundaria “juicio”.

CRC secundaria	<u>Juicio</u>
Responsabilidades	Se obliga a la <u>razón social</u> a pagar la deuda contra el fisco.
Colaboraciones	<u>Abogado</u> , <u>particulares</u>

Tabla 14 – CRC secundaria “particulares”.

CRC secundaria	<u>Particulares</u>
Responsabilidades	Tiene que abonar una deuda contra el fisco.
Colaboraciones	<u>Abogado</u> , <u>Juicio</u> , <u>cedulas de notificación</u>

Al igual que en los ejemplos de LEL y escenarios, las referencias a signos del LEL se indican subrayadas y en color rojo (ejemplo). Y las referencias a tarjetas CRC se muestran subrayadas y en color negro (ejemplo). La figura 7 muestra el conjunto completo de entradas de LEL, escenarios y tarjetas CRC, con todas las relaciones entre ellos.

3.7. Relación entre LEL, escenarios y tarjetas CRC

LEL, escenarios, tarjetas CRC y las heurísticas de derivación de tarjetas CRC conforman un método para partir del modelo de contexto del dominio y lograr los elementos necesarios para realizar un diseño preliminar de objetos.

Todos estos elementos están interrelacionados. El LEL brinda la materia prima para la construcción de los escenarios. Luego, LEL y escenarios, utilizando las reglas de derivación, permiten obtener tarjetas CRC.

Cualquier variación en las entradas del LEL o los escenarios puede determinar que de acuerdo a las heurísticas de derivación se presenten condiciones para que varíen las tarjetas CRC (que aparezcan nuevas, que dejen de existir o sufran modificaciones). La variación del LEL o de algún escenario es común debido a la naturaleza dinámica del dominio.

Por lo tanto, se necesitan de técnicas para propagar en forma automática los cambios que se producen. Esta necesidad es la de proveer forward traceability.

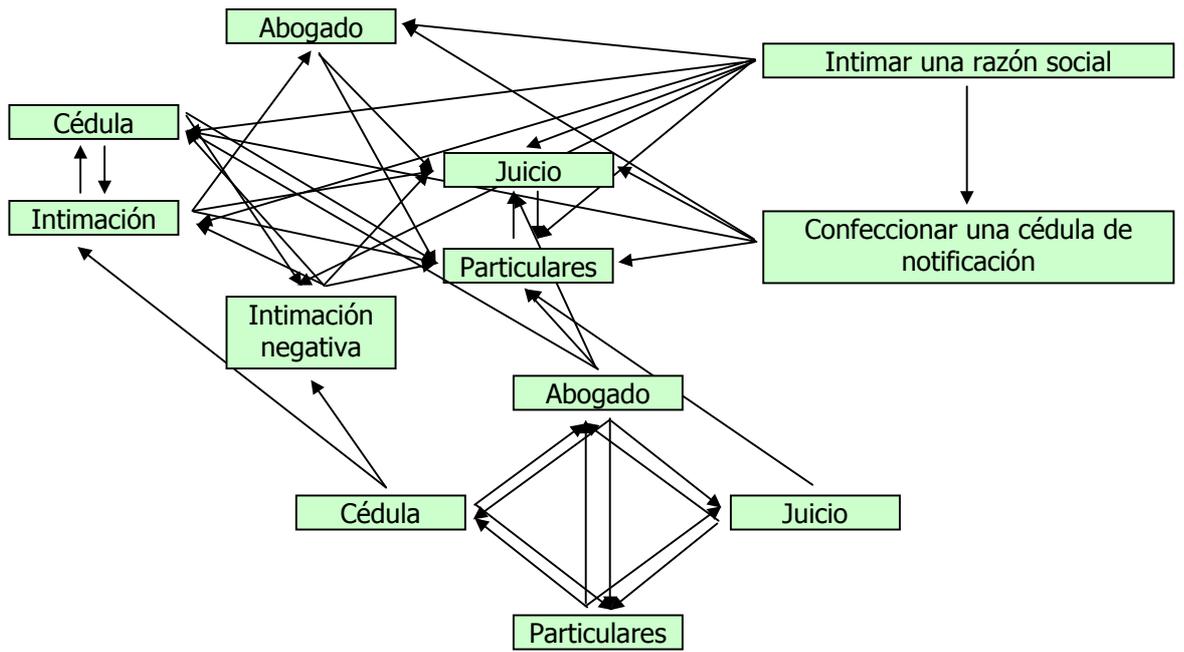


Figura 7 - Hipertexto de LEL, escenarios y tarjetas CRC de un organismo gubernamental de la República Argentina.

4. Herramienta automatizada

Forward traceability es de suma importancia en los proyectos de software y en particular en la gestión de requerimientos, puesto que el volumen de información que se debe manejar es inmenso, sin una herramienta esta tarea es imposible [Alspaugh 1999] [Ramesh 1998].

En la sección 4.1 se indica la importancia de las herramientas. En la sección 4.2 se describe el Baseline Mentor Workbench, una aplicación para trabajar con LEL, escenarios y tarjetas CRC. En las secciones 4.3 y 4.4 se muestran dos evaluaciones para mostrar que es una herramienta adecuada. Se evalúan la usabilidad y el algoritmo de derivación de tarjetas CRC implementado.

4.1. Importancia de una herramienta automatizada en la ingeniería de software

Si sucede algún cambio en cualquiera de los requerimientos se deben reexaminar los links de traceability para evaluar los productos obtenidos a partir de los requerimientos que cambiaron. Luego se debe determinar si todavía son válidos como para que sigan existiendo, o si necesitan crear nuevos [Cockton 2001]. Con grandes volúmenes de información es imposible realizar esta tarea manualmente, porque el tiempo que consumiría sería muy grande y la posibilidad de cometer errores alta [Ramesh 1998].

4.2. Baseline Mentor Workbench

Baseline Mentor Workbench ("BMW") es una herramienta que tiene como función asistir al experto del dominio durante la fase de ingeniería de requerimientos utilizando la metodología del Client Oriented Requirements Baseline [Antonelli 1999]. BMW administra las entradas de LEL, escenarios y las tarjetas CRC.

Dentro del BMW, un Client Oriented Requirements Baseline para un dominio de aplicación dado es llamado proyecto. Dentro de cada proyecto la evolución en el tiempo del mismo es capturada a través de versiones (un proyecto es dividido en varias versiones). Cada versión tiene tres conjuntos: los símbolos de LEL, los escenarios y las tarjetas CRC. Las entradas del LEL y los escenarios se deben ingresar manualmente al BMW, sin embargo las tarjetas CRC son generadas automáticamente. Dentro de cada versión las tarjetas CRC son derivadas de las entradas del LEL y los escenarios pertenecientes a la misma versión. El proceso de derivación es una versión adaptada de la estrategia de derivación presentada en [Leonardi 2001].

La arquitectura de la aplicación es una típica Arquitectura de Repositorio [Sommerville 1995] compuesta por cuatro subsistemas: editor de entradas de LEL, editor de escenarios, generador de tarjetas CRC y browser de navegación (figura 8).

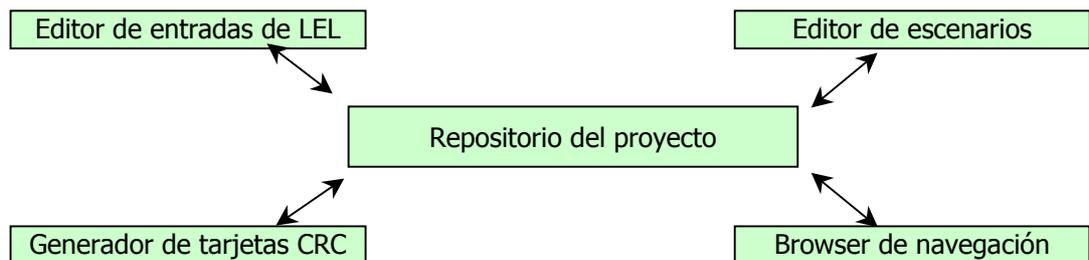


Figura 8 – Arquitectura del Baseline Mentor Workbench.

Los editores de entradas del LEL y escenarios proveen ayudas de edición. En primer lugar, existen expresiones de formato fijo dentro de la descripción de los episodios en los escenarios, para las cuales el BMW escribe un template en forma automática a través de una opción de menú contextual. De esta forma el usuario sólo tiene que completar espacios con la información específica. Otro rasgo consiste en hacer links en forma automática. Durante el proceso de edición, después de que cada palabra es ingresada, la aplicación chequea para determinar si la palabra existe en el conjunto de entradas del LEL. Si es así, un nuevo link es creado y la palabra es enfatizada para indicar el link. Por el contrario, si una palabra nueva es entrada, el usuario puede seleccionar la palabra e indicar que debe ser un link a través de la opción del menú contextual "hot spot to LEL entry". Con esta opción, una nueva entrada es agregada en el conjunto de entradas del LEL.

Con respecto al generador de CRC, BMW deriva tarjetas CRC a partir de las entradas del LEL y los escenarios de acuerdo al algoritmo de derivación expuesto en [Leonardi 2001]. Sin embargo, la implementación en el BMW es una simplificación de este algoritmo que de todas formas cumple con las etapas principales:

- ❖ Hallar clases primarias. Son los actores de los escenarios que también están en el conjunto de entradas del LEL.
- ❖ Hallar responsabilidades de las clases primarias. Se toman de los impactos de las entradas del LEL.
- ❖ Hallar clases secundarias. Son las referencias a entradas del LEL que se encuentran en las responsabilidades de las clases primarias.
- ❖ Hallar responsabilidades de las clases secundarias. Se toman de los impactos de las entradas del LEL.
- ❖ Hallar colaboraciones. Consiste en determinar con que otras tarjetas CRC participa en la resolución de cada escenario.
- ❖ Depurar las tarjetas CRC. Consiste en revisar y eliminar tarjetas CRC y atributos repetidos.

En Client Oriented Requirements Baseline cada entidad (entrada de LEL o escenario) debe ser descripta usando la mayor cantidad de otras entidades (pertenecientes al Baseline) como sea posible. Esto es lo que dicta el principio de circularidad. Luego, dada esta relación entre los elementos del Baseline, se conforma un hipertexto el cual debe ser navegado para comprender la información. Cuando se navega, se muestran links de diferente forma para entradas del LEL, escenarios y tarjetas de CRC. Los links a entradas de LEL son de color rojo y en letra cursiva. Los links a escenarios son azules y en negrita. Y los links a tarjetas CRC son grises. Para identificar los links navegados de los no explorados, se representan los primeros con una luminosidad mayor (son mas claros).

El browser de navegación de BMW soporta el reconocimiento del contexto navegacional. Un contexto navegacional [Schwabe 1995] es un recurso que hace más sencillo el proceso de navegación. Los episodios dentro de un escenario pueden ser descriptos a través de escenarios en sí mismos. Por lo tanto si un escenario tiene muchos episodios y el usuario desea explorar cada uno, tendrá que clicar el link del primer episodio, luego el botón atrás para volver al escenario original, luego el segundo episodio, y así seguir. Sin embargo, la aplicación reconoce contextos navegacionales, y cuando un link con un contexto navegacional es explorado se muestran en la ventana de navegación la lista de links que pertenecen al contexto navegacional junto con los botones anterior y siguiente. Por lo tanto para navegar todos los episodios alcanza con hacer click en el primer episodio y presionar el botón siguiente hasta el último episodio.

La ventana principal (figura 9) permite abrir todas las ventanas de la aplicación y acceder a todas las funciones.

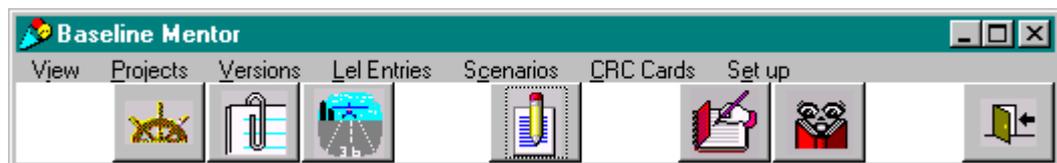


Figura 9 – Ventana principal del Baseline Mentor Workbench.

Para editar LEL y escenarios se debe utilizar la ventana del browser completo. La misma está dividida cinco secciones: proyectos, versiones, LEL, escenarios y tarjetas CRC. La figura 10 muestra la ventana de browser completo, con el ejemplo que se desarrolló para explicar los conceptos de LEL, escenarios y tarjetas CRC. La parte superior contiene la lista de proyectos, sólo uno aparece el proyecto 'Apremios'. Debajo de éste, aparece la lista de versiones, hay sólo una en lista. Debajo de la lista de versiones se encuentran las listas de entradas de LEL, escenarios y tarjetas CRC. Cada una de las listas tiene los elementos que se mostraron en los ejemplos.

Cada una de estas listas tienen menues asociados (figura 11). El primer menú, se corresponde con los proyectos. Las operaciones posibles son las tradicionales: crear, borrar o editar. También se permite grabar o cargar desde un medio de almacenamiento externo. Luego se muestra el menú de

las versiones. Con las versiones es posible crear una nueva a partir de una ya existente. Esto implica la copia de toda la información para su modificación. También es posible editar o eliminar una versión. El menú de versiones incluye dos funciones más: exportar la información a formato HTML o RTF. Luego, le siguen los menús de las entradas de LEL y los escenarios. Ambos permiten las funciones de edición: agregar, modificar y eliminar. También permiten editar elementos referenciados por el elemento seleccionado. Esta función es útil cuando se crea una entrada de LEL y desde la misma se crean otras entradas en blanco. Las funciones **edit LEL entry** y **edit escenario** permiten editar estas entradas creadas en blanco desde el elemento que los originó. Por último, se dispone de funciones para navegar la información como ser **open view** y **open escenarios having LEL entry**. La última función es una consulta muy sencilla que permite recorrer los escenarios que contienen una entrada de LEL en particular. El último menú es el correspondiente a la lista de tarjetas CRC. Permite derivar las mismas a partir de las entradas de LEL y los escenarios utilizando el algoritmo de derivación descripto.

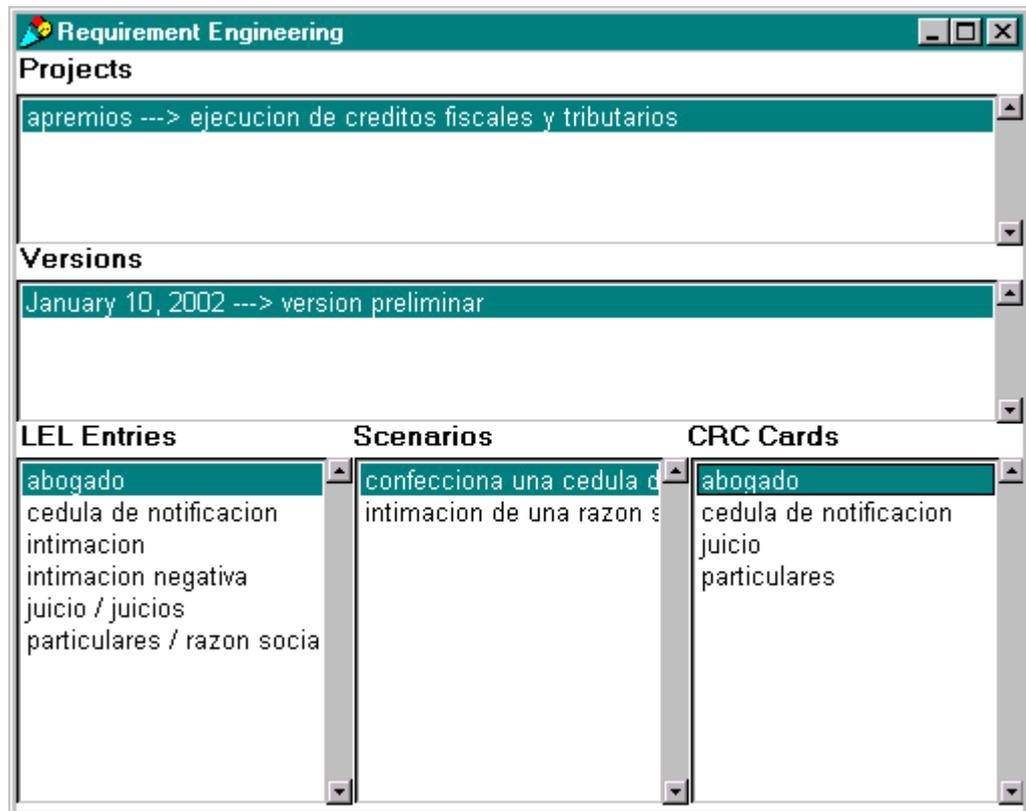


Figura 10 – Ventana del browser completo del BMW.

En la figura 12 se muestra la ventana de edición de los escenarios con los menús contextuales de cada sección. Todos los menús tienen las opciones de edición: cortar, copiar y pegar. Luego se agregan funciones para generar referencias a otras entradas. Las funciones **hot spot to LEL** y **hot spot to escenario** permiten llevar a cabo esta tarea. En principio, los links se generan en forma automática a medida que se escribe, ya que la herramienta reconoce los links existentes y los marca. Pero si durante la edición se escribe una palabra que a posteriori será una entrada de LEL o escenario, se lo puede marcar desde alguna de estas opciones y se crea la entrada vacía en forma automática. Por último, debido a que los episodios se describen de acuerdo a una sintaxis prefijada, el menú contextual para la edición de los episodios incluye opciones para escribir un template de estas líneas.

New	New release	New	New
Del	Del	Del	Del
Edit	Edit	Edit	Edit
Save	Export to Html	Edit LEL Entry	Edit Subscenario
Load	Export to RTF	Open View	Edit Lel Entry
		Open scenarios having LEL entry	Edit Links def usr
			Open View
		Derivate CRC Card	

Figura 11 – Menues contextuales del BMW.

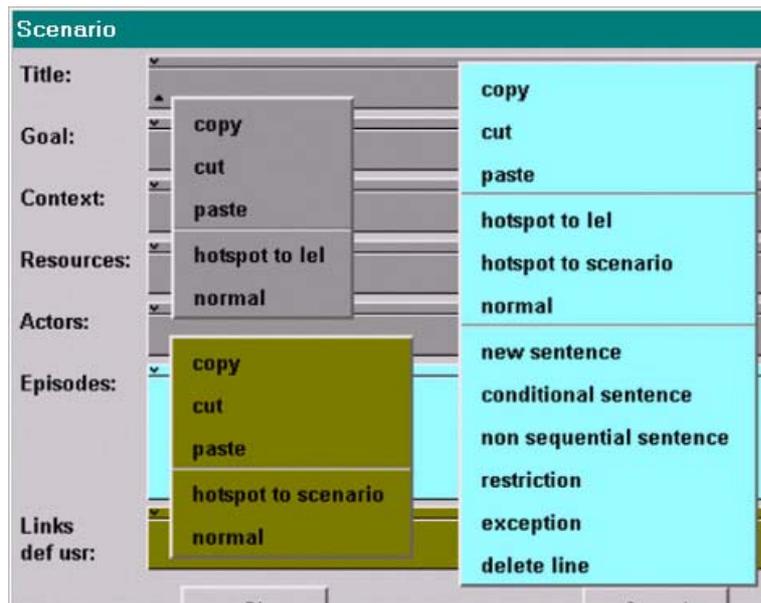


Figura 12 – Ventana de edición de los escenarios del BMW con los menues contextuales.

La ventana de navegación posee tres sectores (figura 13). En la parte superior se muestra la información del escenario navegado. Allí se muestra cada uno de los atributos y su descripción. Debajo se muestra un sector con la historia de navegación. Y debajo de éste, se muestra el contexto navegacional del escenario. Los links se muestran con su color característico y se diferencian los navegados de los no explorados.

4.3. Evaluación de la usabilidad de BMW

Llevamos a cabo una encuesta entre los años 2000 y 2001 con el fin de determinar que tan útil es BMW.

4.3.1. Descripción de la evaluación

En la encuesta participaron 49 personas. Los participantes eran alumnos de un curso de postgrado en la Universidad Nacional de La Plata. El curso se realizó simultáneamente en dos ciudades: La Plata y Córdoba. Estas ciudades están situadas a aproximadamente 700 kilómetros una de otra. La gente de tales ciudades tan distantes experimentan diferentes prácticas de desarrollo de software. Esta variedad hace un buen grupo para la encuesta.

Aunque los participantes eran estudiantes graduados, poseían experiencia en el desarrollo de software. Casi todos ellos (47 de 49) estuvieron involucrados en un proyecto de desarrollo de software real en la industria. La mayoría poseían una experiencia de más de cinco años.

Además de su experiencia, los participantes utilizaron los conceptos adquiridos durante el curso para entender los cuestionarios y responder las preguntas, ya que el curso trataba sobre ingeniería de requerimientos.

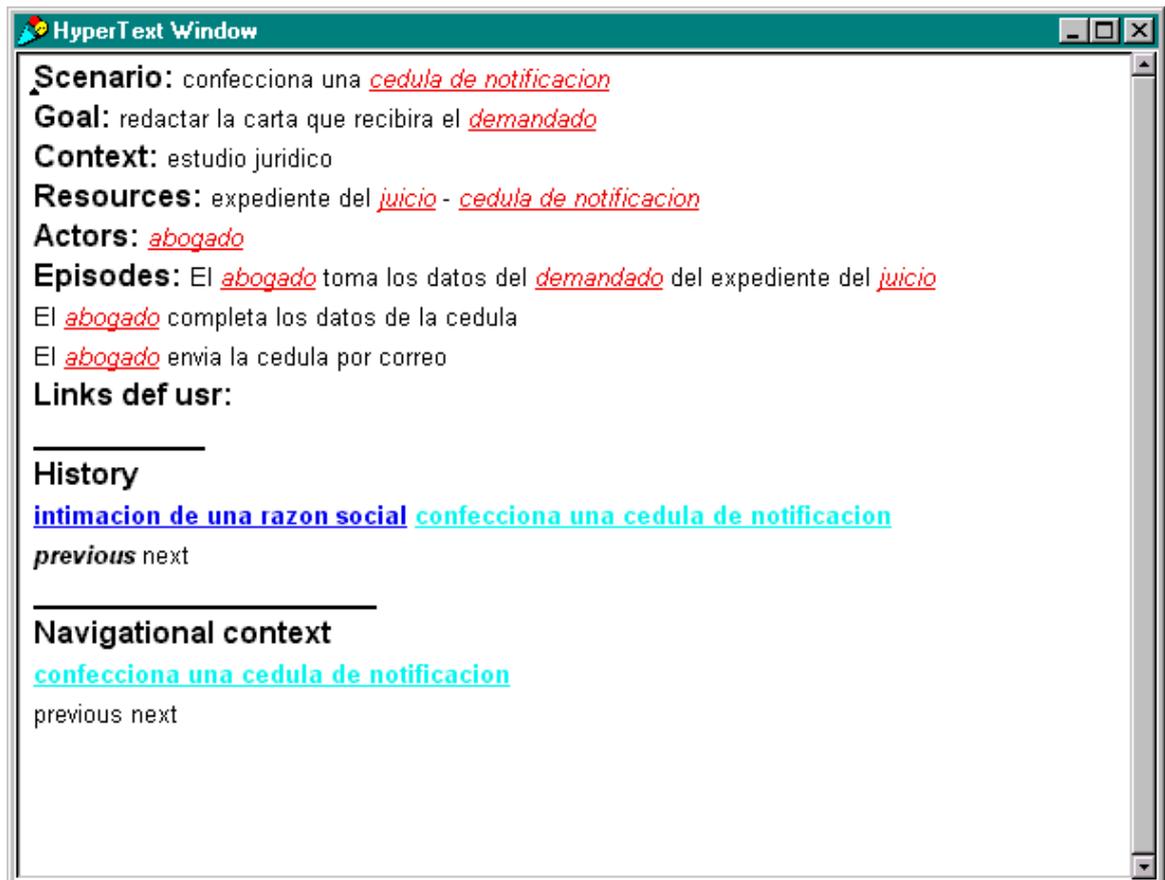


Figura 13 – Ventana de navegación del BMW.

A los estudiantes se les pidió construir el LEL y los escenarios de un problema específico con la ayuda de la herramienta. Al final de la tarea se les pidió completar un cuestionario, el cual estaba dividido en cinco secciones. La primera sección trata sobre información personal, estudios y experiencia. Algunas preguntas son: ¿Qué edad tiene? ¿Cuándo obtuvo su primer título de grado? ¿Cuándo y dónde tomó el curso de maestría? ¿Cuánto tiempo estuvo trabajando en la industria? ¿Cuánto tiempo duró un proyecto de mediana magnitud en el cual trabajó? ¿Cuántas personas participaron del mismo?

La segunda sección pide una opinión general de la herramienta. También compara la calidad del LEL y escenarios obtenidos con la herramienta contra aquellos construidos manualmente o con herramientas de carácter general.

La tercera sección presenta una evaluación de las características que el BMW posee, su importancia y la eficiencia de su implementación. Se seleccionaron los 20 rasgos que mejor describen a la herramienta. Se los agruparon en 4 categorías: rasgos generales, ayudas de edición, consultas y navegación. Rasgos generales básicamente describen características generales y como el Client Oriented Requirements Baseline es capturado por BMW. Las ayudas de edición se refieren a elementos que simplifican el uso de BMW. La categoría de consultas agrupa funciones para buscar y recuperar información. Y navegación agrupa las funciones que permiten recorrer LEL y escenarios. Se pidió a las personas responder dos cosas sobre cada rasgo. Primero, debían evaluar cuan necesario es cada rasgo. Además, debían evaluar cuan adecuadamente estaba soportado el rasgo en la herramienta. Cada pregunta debía responderse con alto, medio y bajo.

La cuarta sección trata los rasgos ausentes en el BMW que podrían ser necesarios. Seleccionamos rasgos que deben estar en cada herramienta CASE y no estaban presentes en BMW [Gabb 1997] [Hermodsson 1998]. La lista incluye 15 rasgos agrupados en cinco categorías: rasgos generales, control de configuración, traceability, interface con otros modelos y herramientas, y reuso. Se les pidió a los participantes responder si la inclusión de los rasgos es obligatoria, recomendada o innecesaria.

La última sección pregunta cuan fácil es aprender a usar BMW. Medimos el tiempo necesario para conocer la funcionalidad básica de la herramienta.

4.3.2. Participantes

Además de determinar cuan útil es BMW para asistir al proceso de elicitación de requerimientos utilizando Client Oriented Requirements Baseline. La evaluación buscó encontrar que extensiones se deben realizar a la herramienta para que sea mas útil.

Los participantes sólo tuvieron que satisfacer una condición para ser adecuados para la evaluación. Ellos debieron haber usado BMW. No era necesario que fueran expertos en ingeniería de requerimientos, ya que el objetivo no fue validar el Client Oriented Requirements Baseline. Tampoco fue necesario que los participantes sean expertos en herramientas CASE, porque no intentamos comparar BMW con otras herramientas CASE. Sin embargo, si fue necesario que los participantes tuvieran estudios académicos con experiencia en el desarrollo de software, y ellos debían conocer a la herramienta; porque ellos debían conocer como elicitar requerimientos con la asistencia de una herramienta de acuerdo a la teoría y a su propia experiencia con el fin de evaluar BMW.

Utilizamos el año de graduación para analizar los estudios académicos. El año determina el conocimiento académico sobre ingeniería de software y sobre ingeniería de requerimientos. La ingeniería de software es una disciplina que comenzó en los 60. Mientras que la ingeniería de requerimientos es más joven, apareció en los 90. Más del 50% de los participantes se graduaron entre 1995 y 2000, mientras 22 personas lo hicieron entre 1976 y 1994 (tabla 15). Estas 22 personas seguramente se han formado en los aspectos de requerimientos más por su práctica que por sus estudios.

Tabla 15 – Cantidad de personas que se graduaron por rango de años.

Año	Cantidad de personas
1976-1990	12
1991-1994	10
1995-2000	27

Los participantes también difieren en la experiencia de trabajo en la industria. Sin embargo, la mayoría de ellos poseen una extensa experiencia. El 65% de los participantes tiene más de cinco años de experiencia (tabla 16).

Tabla 16 – Años de experiencia de trabajo en la industria.

Años	Cantidad de personas
0-1	4
2-3	8
4-5	5
6-7	14
8-9	5
10-	13

Las personas también tienen experiencia con herramientas CASE. De las 49 personas, 33 han trabajado con herramientas CASE. De las 33 personas que han trabajado, 20 tienen más de cinco años de experiencia en diferentes proyectos en la industria. Y 14 de estas 20 han trabajado en proyectos con más de 3 personas. Y otro grupo disjuncto de 14 personas han participado en proyectos que duraron más de 4 meses.

Los participantes no necesitan tener experiencia en ingeniería de requerimientos, ni tampoco necesitan conocer otras herramientas CASE. Sin embargo, casi el 25% (12 de 49) tiene una extensa experiencia. Ellos tienen mas de 5 años de experiencia, han utilizados herramientas CASE, y participaron de al menos un proyecto medio integrado por 3 o más personas que duró 4 o más meses.

En nuestra evaluación analizamos los resultados del total de los 49 participantes. Y comparamos estos resultados con los obtenidos del grupo experimentado de 12 personas.

4.3.3. Evaluación de características generales

La mayoría de las personas (46 de 49) reconocen que aplicar Client Oriented Requirements Baseline con BMW es más fácil que utilizar la técnica manualmente, es decir sin la ayuda de ninguna herramienta. Y las personas que piensan que es más duro utilizar la herramienta, sostienen que el producto obtenido con la herramienta es de mejor calidad que uno obtenido manualmente. Del grupo de 12 expertos, 11 consideran que es más fácil utilizando la herramienta.

BMW también fue comparado con herramientas generales como procesadores de texto y planillas de cálculo. Del grupo de 49, 36 prefieren utilizar BMW. Y 6 personas de las que prefieren usar herramientas generales, sostienen que con BMW obtienen un producto de mayor calidad. Del grupo de 12 expertos, 10 personas coinciden que es más fácil utilizar BMW que herramientas de carácter general.

Además, comparamos BMW contra las expectativas de las personas sobre una herramienta específica para asistir al Client Oriented Requirements Baseline. Del grupo de 49, 37 consideran satisfactoria a la herramienta. Altamente satisfactoria la consideran siete personas. Y sólo cinco personas piensan que la herramienta es insatisfactoria. Del grupo de 12 expertos, 10 la encuentra satisfactoria. Uno la halla altamente satisfactoria y uno insatisfactoria.

4.3.4. Evaluación de características detallada

Elegimos los 20 rasgos que mejor describen a la herramienta con el fin de realizar una evaluación detallada de BMW. Agrupamos los rasgos en cuatro categorías: rasgos generales, ayudas de edición, consultas y navegación. Pedimos a los participantes que evalúen cada rasgo en función de la utilidad y eficiencia con la cual el rasgo es implementado.

Utilidad se refiere a cuan importante son los rasgos para que BMW asista en la construcción de LEL y escenarios. Pero indirectamente determinan simplicidad en la herramienta, ya que rasgos inútiles distraen al usuario de los rasgos que realmente debe utilizar. Eficiencia en la implementación los rasgos condicionan la usabilidad de los mismos.

Utilidad se evaluó como alta, media y baja. Alta y media indican que el rasgo es útil, necesario, mientras que baja denota que el rasgo es innecesario.

El grupo de 49 personas evaluó 10 rasgos como necesarios, ya que le asignaron cero utilidad baja a cada uno. Luego, cuatro rasgos fueron evaluados con baja utilidad una sola vez. Otros dos rasgos fueron evaluados con baja utilidad en dos oportunidades. Y los restantes cuatro rasgos fueron considerados de baja utilidad por cinco personas cada uno.

El grupo de 12 expertos consideró útiles a 10 rasgos (le asignaron cero utilidad baja a cada uno). Sólo 5 rasgos fueron considerados inútiles con una votación cada uno (tabla 17).

Tabla 17 – Cantidad de rasgos evaluados con baja utilidad por el grupo de 49 y 12 personas.

Cantidad de rasgos evaluados conbaja usabilidad
49 participantes	12 expertos	
10	15	0
4	5	1
2		2
4		5

Los cuatro rasgos evaluados con cinco marcas de baja por el grupo de 49 y los cinco rasgos con sólo una marca del grupo de 12 expertos merecen ser eliminados. El resto de los rasgos poseen muy pocas votaciones. Tres rasgos fueron comunes a los dos grupos, por lo tanto la lista de rasgos inútiles se reduce a seis rasgos:

- ❖ La herramienta es monousuario, en lugar de ser multiusuario.
- ❖ Sólo soporta Client Oriented Requirements Baseline, en lugar de abarcar todo el ciclo de desarrollo del software.
- ❖ Exportar información en rich text format.

- ❖ Provee palabras claves inmodificables para describir LEL, escenarios y CRC.
- ❖ Diferentes links para LEL escenarios y CRC en navegación, los cuales debieran ser unificados.
- ❖ La historia de los links navegados se deben eliminar.

Se analizó la eficiencia sobre los mismos 20 rasgos. Cada uno fue evaluado en función de cuan adecuada es la implementación de su interacción con el usuario. Aunque el rasgo sea necesario, se tiene que poder usar de una forma eficiente. Cada rasgo se evaluó como alta, media y baja eficiencia. Alta y media significa que la función puede ser usada, mientras que baja significa que la función debe implementarse nuevamente.

El grupo de 49 participantes determinó que siete rasgos fueron correctamente implementados ya que obtuvieron cero marcas baja. Seis rasgos tuvieron una sola marca baja cada uno. Y dos rasgos fueron evaluados con bajo por dos personas. El bajo nivel de marcas baja determina que las 15 funciones están correctamente implementadas. Sin embargo, los cinco rasgos restantes, se deben volver a implementar.

- ❖ Exportar a rich texto format con 14 marcas
- ❖ Monousuario con 9 marcas
- ❖ Soporta sólo Client Oriented Requirements Baseline con 6 marcas
- ❖ Palabras claves inmodificables con 6 marcas
- ❖ Historia de los links navegados con 5 marcas

El grupo experto de 12 personas determinó que nuevo rasgos (dos más que el grupo de 49 participantes) fueron implementados eficientemente (cero marca como baja eficiencia). Otros ocho rasgos fueron evaluados con sólo una marca de baja eficiencia. Sólo tres rasgos deben volver a implementarse ya que tuvieron mas de cuatro votos cada uno:

- ❖ Monousuario
- ❖ Exportar a rich text format
- ❖ Soporta sólo Client Oriented Requirements Baseline

Ambos grupos concuerdan en estos tres rasgos (aunque en distinto orden) como los más votados. Por otra parte, los porcentajes de votos de baja eficiencia del grupo de los 12 fueron más bajos que los porcentajes del grupo de 49 personas.

Como conclusión, cinco de los seis rasgos evaluados como inútiles también fueron evaluados con baja eficiencia de implementación. Estos cinco rasgos deben ser analizados y reimplementados si es necesario. Y el único rasgo con baja usabilidad pero que no fue evaluado con baja eficiencia ("**Diferentes links para LEL escenarios y CRC en navegación deben ser unificados**") debe ser eliminado.

4.3.5. Nuevos rasgos a incorporar

Seleccionas de [Gabb 1997] y [Hermodsson 1998] rasgos no presentes en BMW que deben estar en toda herramienta CASE. La lista incluye 15 rasgos agrupados en cinco categorías: rasgos generales, configuración de control, traceability, interface con otros modelos y herramientas, y reuso. Se les pidió a las personas indicar si la inclusión de cada rasgo es obligatoria, recomendada o innecesaria. Nos concentramos en analizar las respuestas 'obligatoria', que exigen la inclusión en la herramienta.

Calculamos para cada rasgo el porcentaje de respuestas obligatorias del grupo de 49 participantes y del grupo de 12 expertos. Luego ordenamos los rasgos desde el menos obligatorio al más obligatorio utilizando el porcentaje. El orden determinado por ambos grupos coincidió en la posición de 13 rasgos sobre los 15. Los dos rasgos que fueron colocados en distintas posiciones pertenecen a la categoría rasgos generales. El primero se ubicó a un lugar de diferencia, mientras que el segundo se hizo a cinco posiciones.

Los 11 rasgos menos obligatorios pertenecen a las siguientes categorías: control de configuración, rasgos generales, reuso e interface; teniendo en cuenta que no hay orden entre las

categorías. Pero los rasgos de traceability están todos ellos al final de la lista entre los rasgos más obligatorios (tabla 18).

Tabla 18 – Porcentaje de respuestas obligatorias de los grupos de 49 y 12 personas.

% grupo 49	Categoría	Rasgo	% 12 expertos
6	Configuración	Fecha y responsable de cada modificación	8
12	General	Filtros de elementos	17
14	General	Importación de fuentes externas	8
16	General	Filtros de links	33
16	Reuso	Catalogo de requerimientos	17
18	Configuración	Registro de cambios	17
18	Interface	Exportar a otras herramientas CASE	25
20	Reuso	Uso de módulos reusables	25
20	Configuración	Change log para registrar los cambios	25
29	General	Búsqueda rápida de LEL y escenarios	42
33	General	Help on line	50
47	Traceability	Rastrear links desde/hacia otros elementos	58
57	Traceability	Escribir reportes para validación del usuario	58
61	Traceability	Chequeo de completitud	75
73	Traceability	Validación	92

Los usuarios siempre requieren nueva funcionalidad, más todavía si los rasgos sugeridos son aquellos que deben estar en toda herramienta CASE. Sin embargo, el grupo de 49 evaluó sólo a tres rasgos con más del 50% mientras que el grupo de los 12 lo hizo con cuatro rasgos. La mayoría coincide en que los rasgos de traceability son los que se deben agregar.

4.3.6. Curva de aprendizaje

Se analizó el aprendizaje agrupando a los participantes en cinco grupos de acuerdo al tiempo consumido para aprender a utilizar la herramienta. Ninguno de los participantes necesitó más de 16 horas. Sólo el 8% de los 49 participantes necesitó entre 8 y 16 horas, pero ninguno de ellos pertenecía al grupo de los 12 expertos. Aproximadamente el 80% de las personas necesitó hasta 4 horas, ambos grupos confirman estos datos (tabla 19).

Tabla 19 – Tiempo de aprendizaje.

Tiempo necesario	Grupo de 49	12 expertos
Menos de 2 horas	30%	41.6%
Entre 2 y 4 horas	49%	41.6%
Entre 4 y 8 horas	12%	16.6%
Entre 8 y 16 horas	8%	0%
Más de 16 horas	0%	0%

4.3.7. Conclusión

Una gran cantidad de personas, con diferente perfil académico y de experiencia en la industria, participó en la encuesta. De los 49 participantes el 25% posee una sólida experiencia. Se analizaron las respuestas del grupo total de 49 participantes y se verificaron contra el 25% de expertos.

BMW ha sido generalmente preferido a realizar el trabajo manualmente o usando herramientas de carácter general. Es simple de usar porque casi toda la funcionalidad que provee es necesaria y esencial. Esta simplicidad se refleja en la curva de aprendizaje. Sin embargo, nueva funcionalidad se debe incluir, en particular aspectos de traceability.

4.4. Evaluación del algoritmo de derivación de CRCs implementado en BMW

La evaluación del algoritmo de derivación de tarjetas CRC implementado en BMW consta de dos etapas. En primer lugar se compara el desempeño del algoritmo implementado en la herramienta

contra la aplicación manual del algoritmo. Una vez determinado que la implementación obtiene resultados semejantes a los de la aplicación manual, se analiza si la implementación del algoritmo produce resultados útiles para la construcción de un modelo de objetos.

4.4.1. Comparación del algoritmo implementado en BMW contra la aplicación manual

Para poder evaluar la eficacia del algoritmo implementado en la herramienta se realizó el LEL y los escenarios de un dominio concreto. Luego, se obtuvieron manualmente las tarjetas CRC correspondiente. Y por otro lado, se cargó el LEL y los escenarios en la herramienta, y se derivó en forma automática las tarjetas CRC. En [Antonelli 1999] se muestran algunos resultados preliminares sobre este análisis.

El dominio que se ha utilizado es el de una administradora de planes de ahorro para automóviles. Una alternativa que se brinda en Argentina para poder realizar la compra de un automóvil, es la de inscribirse en un plan de ahorro. Un plan de ahorro es una modalidad de pago en donde el cliente firma un contrato por el cual se compromete a pagar en una cantidad estipulada de cuotas mensuales el valor del automotor (por ejemplo 60 cuotas, que representa un plazo de 5 años). El plan de ahorro se conforma por un grupo suficiente de personas, de forma que mes a mes se vaya recaudando el dinero suficiente para alcanzar el valor de un automóvil. De esta forma, todos los meses se hacen sorteos para entregar un auto a un participante. Otra forma de lograr el automotor en el plan de ahorro es por medio del proceso de licitación. Licitación consiste en hacer una oferta, representada por un número de cuotas que se pagan todas juntas. Quien ofrezca el mayor monto se hace acreedor al automóvil. La administradora de planes de ahorro es la entidad encargada de organizar este sistema.

Una descripción más detallada de una administradora de planes de ahorro puede encontrarse en [Parravicini 1997]. La misma se transcribe a continuación.

“El objetivo de la administradora es la conformación de un grupo de adherentes, los cuales a través de un sistema de ahorro integran un grupo para adjudicar automóviles. Para ser adherente de un grupo el solicitante deberá completar un formulario y presentar la documentación. Si es aceptado se suscribirá a un plan de acuerdo a un contrato de adhesión. Existen diferentes tipos de planes. Los adherentes pagan una cuota mensual, lo que le da derecho a participar en actos mensuales de adjudicación por sorteo o licitación. El sorteo es un beneficio de todos los adherentes con los cuotas al día, mientras que la licitación es para aquellos que lo han solicitado mediante una oferta de dinero (utilizando un formulario de licitación). Un adherente se convierte en adjudicatario al obtener el automóvil por alguno de estos dos medios; en este caso deberá solicitar el bien mediante un formulario especial, abonar cargos especiales en concepto de la entrega del automóvil y continuar con el pago de las cuotas mensuales. El adjudicatario puede optar por un automóvil de mayor o menor valor, para lo cual existen una serie de reglas.

Un adherente puede ceder sus derechos del plan a un tercero, puede renunciar al plan o puede ser separado por la administradora.

El fabricante puede dejar de producir la marca y/o modelo del contrato, lo que afecta de diferente forma a los adherentes o adjudicatarios.

La adhesión al grupo implica el pago de un seguro de vida y se le exige un seguro para el auto una vez que ha sido adjudicado.

Un grupo puede ser disuelto por decisión de los adherentes (en caso de que no se fabrique un reemplazo del bien) o por la administradora siguiendo una determinada política para liquidación de los miembros.”

Sobre este dominio se han hechos dos trabajos [Parravicini 1997] y [Mauco 1997]. Cada uno de estos trabajos consistió en preparar el LEL, los escenarios, encontrar las tarjetas CRC, realizar el modelo lógico conformado por el diagrama de asociación y de interacción, y por último realizar el diagrama de capas. Nuestro objetivo consiste en comparar simplemente las tarjetas CRC que se obtuvieron contra las que BMW derivó a partir de LEL y los escenarios que cada uno de los grupos elaboró.

Primer caso de estudio

Para el primer grupo, las entradas de LEL y los escenarios obtenidos son los que muestra la tabla 20.

Tabla 20 – LEL y escenarios obtenidos en el primer caso de estudio.

<u>LEL</u>	
<u>Adherente / Adherentes</u>	
<u>Adherente deja plan / Dejar plan / Renunciar al plan de ahorro</u>	
<u>Adjudicatario / Adjudicatarios / Adjudicado</u>	
<u>Administradora</u>	
<u>Aseguradora</u>	
<u>Banco</u>	
<u>Bien tipo rechazado / Rechaza el bien tipo / Rechazar bien tipo</u>	
<u>Bienes tipos / Bien tipo / bien</u>	
<u>Cambio de bien tipo</u>	
<u>Carga de administración</u>	
<u>Comunica fehacientemente / Comunicar fehacientemente / Comunicación fehaciente</u>	
<u>Cuota comercial</u>	
<u>Cuota de ingreso</u>	
<u>Cuota mensual / Cuota / Cuotas</u>	
<u>Cuotas puras / Cuota pura</u>	
<u>Cupón de pago</u>	
<u>Derecho de adjudicación</u>	
<u>Derecho de admisión</u>	
<u>Entregar bien tipo</u>	
<u>Evaluar licitaciones / Licitación</u>	
<u>Expulsar adherente / Adherente es expulsado</u>	
<u>Fabricante</u>	
<u>Grupo</u>	
<u>Licitación</u>	
<u>Plan de ahorro / Planes de ahorro / plan</u>	
<u>Plan media cuota</u>	
<u>Planilla de adhesión</u>	
<u>Seguro de bien tipo</u>	
<u>Seguro de vida</u>	
<u>Solicitante</u>	
<u>Solicitud cambio de bien</u>	
<u>Solicitud de adhesión / Solicitudes de adhesión</u>	
<u>Sortear / Sorteo</u>	
<u>Transferir plan</u>	
<u>Valor móvil</u>	

A partir de las entradas de LEL y los escenarios de la tabla 20 se obtuvieron las tarjetas CRC que se muestran en la tabla 21. En ella se muestran tres columnas. En la primera se muestran las tarjetas obtenidas tanto manualmente como por la herramienta. En la segunda aquellas que sólo se obtuvieron manualmente. Y en la tercera columna las que solamente halló la herramienta.

Tabla 21 – Comparación de las CRCs obtenidas del primer caso de estudio.

Manualmente y por BMW	Sólo manualmente	Sólo BMW
Adherente		
Adjudicatario		
Administradora		
Aseguradora		
Banco		

Bien tipo		
	Comunicación fehaciente	
		Cuota de ingreso
		Cuota mensual
		Derecho de adjudicación
		Expulsar adherente
Fabricante		
Grupo		
Licitación – Evaluar licitaciones (Se trata de la misma CRC card, sólo que se eligieron sinónimos distintos para identificarlas)		
		Licitar
Plan de ahorro - Plan (Se trata de la misma tarjeta CRC, sólo que se eligieron sinónimos distintos para identificarlas)		
Planillas de adhesión		
		Rechazar bien tipo
		Renunciar al plan de ahorro
Seguro de bien tipo		
Seguro de vida		
Solicitante		
		Solicitud de adhesión
Sorteo		
		Transferir plan

De las 16 tarjetas CRC obtenidas manualmente, 15 también se obtuvieron en forma automática. Sólo una tarjeta (6.66%) no ha sido hallada. La tarjeta CRC que no fue hallada es "**comunicación fehaciente**". El motivo por el cual no se encontró es que, si bien es una entrada de LEL, no es actor de ningún escenario, por lo tanto no puede ser CRC primaria. Y para que pueda ser secundaria, debiera estar referenciada desde una tarjeta CRC primaria, pero tampoco sucede esto. Solamente es referenciada desde una CRC secundaria "**transferir plan**" (definida por BMW solamente). Por lo tanto ni el algoritmo de derivación, ni el implementado en la herramienta, dan motivos para definir esta tarjeta.

BMW ha encontrado nueve tarjetas CRC que no se han encontrado manualmente. Cuatro de ellas son posibles tarjetas ("**cuota de ingreso**", "**cuota mensual**", "**derecho de adjudicación**" y "**solicitud de adhesión**"), sin embargo cinco de ellas ("**expulsar adherente**", "**licitar**", "**rechazar bien tipo**", "**renunciar plan de ahorro**" y "**transferir plan**") son acciones. Estas cinco acciones también se hubieran encontrado manualmente aplicando minuciosamente el algoritmo. Sin embargo, seguramente se han descartado puesto que no pueden considerarse como tarjetas CRC. A pesar de ello, se convertirán en responsabilidades de otras tarjetas. Y si no es posible hallar las tarjetas CRC responsables de estas acciones, este hecho es una pauta de que está faltando una tarjeta CRC, por lo cual hay que revisar las entradas de LEL y los escenarios.

Por lo tanto, a pesar de que las tarjetas CRC que representan acciones no pueden considerarse como tales, son importantes para hacer un análisis del dominio y brindan un punto de partida para el diseño.

Segundo caso de estudio

En [Mauco 1997] las entradas de LEL y los escenarios encontrados son los que se muestran en la tabla 22.

Tabla 22 – LEL y escenarios obtenidos en el segundo caso de estudio.

<u>LEL</u>	
<u>Actos de adjudicación</u>	<u>Actos de adjudicación de un automóvil</u>

<i>Adherente / Adherentes</i>	Adjudicar automóvil a adherente
<i>Adherente adjudicado</i>	Cancelar el plan
<i>Adherente sorteado</i>	Cancelar parte de la deuda
<i>Adjudicatario</i>	Cobrar cuotas mensuales
<i>Administradora</i>	Construir un grupo de adherentes
<i>Automóvil / Bien tipo</i>	Disolver grupo de adherentes
<i>Banco</i>	Entregar bien tipo a adjudicatario
<i>Cancelación de plan</i>	Expulsar adherente
<i>Cancelación Parcial</i>	Fabricante entrega bien tipo
<i>Cancelación Total</i>	Firmar contrato de adhesión
<i>Carga de administración</i>	Licitación un automóvil
<i>Cobro cuota mensual</i>	Pagar cuota mensual
<i>Compañía de seguros</i>	Pagar póliza de seguro de vida
<i>Comunicación fehaciente</i>	Pagar póliza de seguro del bien tipo
<i>Contrato de adhesión</i>	Presentar oferta de licitación
<i>Cuota de entrega</i>	Reemplazar bien tipo
<i>Cuota de ingreso</i>	Renunciar al plan de ahorro
<i>Cuota mensual / Cuotas mensuales</i>	Solicitud de adhesión al plan de ahorro
<i>Cupón de pago / Cupones de pago</i>	Sortear un automóvil
<i>Disolución del grupo</i>	
<i>Entrega del bien tipo</i>	
<i>Evaluación de licitación</i>	
<i>Expulsión de adherente</i>	
<i>Fabricante</i>	
<i>Formulario de entrega</i>	
<i>Formulario de inscripción</i>	
<i>Formulario de licitación</i>	
<i>Grupo de Adherentes</i>	
<i>Licitación</i>	
<i>Oferta de licitación aceptada</i>	
<i>Oferta de licitación rechazada</i>	
<i>Pago de cuota mensual</i>	
<i>Plan cancelado</i>	
<i>Plan común</i>	
<i>Plan de Ahorro</i>	
<i>Plan media cuota</i>	
<i>Reemplazo del bien tipo</i>	
<i>Renunciar al plan</i>	
<i>Seguro de vida</i>	
<i>Seguro del bien tipo</i>	
<i>Solicitante / Solicitantes</i>	
<i>Solicitante aceptado</i>	
<i>Solicitud de adhesión</i>	
<i>Sorteo</i>	
<i>Valor móvil</i>	

Las tarjetas CRC obtenidas a partir de los elementos de la tabla 22 se muestran en la tabla 23.

Tabla 23 – Comparación de CRCs obtenidas del segundo caso de estudio.

Manualmente y por BMW	Sólo manualmente	Sólo BMW
Acto de adjudicación		
Adherente		
		Adherente adjudicado

Adjudicatario		
Administradora		
Banco		
Bien tipo		
		Cancelación parcial
		Cancelación total
Compañía de seguros		
		Comunicación fehaciente
	Construcción del grupo de adherentes	
Contrato de adhesión		
		Cuota de ingreso
		Cuota mensual
	Cupón de pago	
Disolución del grupo		
	Expulsión de adherente	
Fabricante		
Formulario de entrega		
Formulario de inscripción		
Formulario de licitación		
Grupo de adherentes		
Licitación		
		Plan media cuota
		Renunciar al plan
		Seguro de vida
Solicitante		
	Sorteo	

Se han encontrado 16 tarjetas CRC en forma y por medio de BMW a la vez. A través del proceso automático se encontraron nueve tarjetas que no se las hallaron manualmente. Esta cifra es similar a las CRC extras encontradas por el proceso automática en el caso de estudio número uno. Luego son cuatro las tarjetas que se obtuvieron manualmente y no fue así automáticamente.

Dentro de las 16 tarjetas encontradas en común por ambos métodos, hay dos tarjetas que podrían ser cuestionadas de ser tarjetas CRC por estar más cerca de representar responsabilidades que objetos: **"acto de adjudicación"** y **"disolución del grupo"**. Con dos tarjetas encontradas sólo manualmente sucede lo mismo: **"construcción del grupo de adherentes"** y **"expulsión de adherente"**. Y tres tarjetas encontradas sólo automáticamente también representan responsabilidades: **"cancelación parcial"**, **"cancelación total"** y **"renunciar al plan"**. Como se indicó en el primer caso de estudio, si bien las acciones no pueden considerarse como tarjetas CRC, son importantes para hacer un análisis del dominio.

Es importante analizar las tarjetas CRC encontradas manualmente, que no se lo ha hecho automáticamente, porque esto evidencia una falla en la implementación del algoritmo, el cual tratamos de probar que obtiene los mismos resultados que su aplicación manual.

Las tarjetas no encontradas automáticamente son: **"construcción del grupo de adherentes"**, **"cupón de pago"**, **"expulsión de adherente"** y **"sorteo"**. Con respecto de la primera tarjeta (**"construcción del grupo de adherentes"**), en todo el LEL no hay ninguna referencia a este término. Por lo tanto, el LEL no se ha construido correctamente y este es el motivo de que falte la tarjeta CRC. Debido a que las tarjetas son obtenidas a partir de la información del LEL y los escenarios, si en ninguno de ellos hay ningún tipo de referencia al término en cuestión, no se puede esperar que el mismo aparezca en las tarjetas CRC.

La segunda tarjeta que no ha sido derivada por BMW es **"cupón de pago"**. El algoritmo de derivación dicta que para que una tarjeta CRC primaria se derive debe haber un símbolo de LEL que sea actor de un escenario. Claramente, **"cupón de pago"** no puede ser actor de un escenario. Cuadra para ser un recurso, pero no es una entidad activa como para desempeñarse como actor de

un escenario. De esta forma, la única alternativa es que sea una CRC secundaria. Para ello, debe brindar algún servicio a algún CRC primaria, esto es, debe referirse a ella desde las responsabilidades de la CRC primaria. Sin embargo, desde ninguna CRC primaria se referencia al término **"cupón de pago"**. Y para que esto ocurra alguna entrada de LEL debe contener en sus impactos alguna acción con el **"cupón de pago"**. Pero esto tampoco ocurre. Por lo tanto, al igual que el término **"construcción del grupo de adherentes"**, **"cupón de pago"** no tiene la suficiente justificación en el LEL y escenarios contruidos para que se cree una tarjeta CRC a partir de ellos. Sin embargo, en el dominio si es importante, por lo cual se las define manualmente. El problema radica en que la información obtenida que hizo que se defina la tarjeta, no se volcó adecuadamente en el LEL y los escenarios.

La tercera tarjeta a analizar para determinar por qué motivo no se derivó en forma automática es **"expulsión de adherente"**. Este término está definido en el LEL, sin embargo es una acción. No podría ser nunca categorizado como sujeto, ni en ninguna subclasificación. De esta forma, nunca podría ser actor de un escenario, y así no puede ser una tarjeta CRC primaria. Podría ser una tarjeta secundaria si la entrada de LEL **"adherente sorteado"** (que es el único punto desde donde se la referencia en los impactos) fuera tarjeta primaria. Pero esto no es así. El **"adherente sorteado"**, que es quien puede ser expulsado, no es una tarjeta CRC de ningún tipo, ni primaria ni secundaria. Si no hay ninguna tarjeta que realice la acción o en quien recaiga la acción, el algoritmo nunca la derivaría como tarjeta. De esta forma, la falta de esta tarjeta también se debe a una incompleta elaboración del LEL y los escenarios, no del algoritmo ni de la implementación del mismo.

El último término a analizar para determinar porque motivo no es tarjeta CRC es **"sorteo"**. La situación de sorteo es similar a la de **"expulsión de adherente"**. Es una acción que está definida en el LEL. Nunca podría ser tarjeta primaria, ya que una acción nunca podría ser actor de un escenario. Sin embargo podría ser tarjeta CRC secundaria, si alguna tarjeta CRC primaria realiza el sorteo. Las únicas tarjetas que hacen referencia al sorteo son **"actos de adjudicación"** y **"comunicación fehaciente"**. Ambas tarjetas son secundarias. **"Actos de adjudicación"** es una acción, mientras que **"comunicación fehaciente"** es un objeto, según la clasificación de las entradas de LEL que le dieron origen. Para que pueda ser una tarjeta primaria, la clasificación de la entrada de LEL debe ser sujeto, y ninguna de ellas lo es. El único sujeto que es tarjeta de CRC primaria y tiene la responsabilidad de realizar el sorteo es la administradora. Sin embargo, no es ella quien realiza el sorteo, ni tampoco es responsabilidad de ningún otro sujeto. Por lo tanto, no hay ninguna tarjeta primaria que lo realice y de esta forma la tarjeta CRC secundaria, nunca se puede derivar.

Conclusión

El objetivo inicial de este análisis es determinar si la implementación en BMW del algoritmo de derivación de tarjetas CRC de [Leonardi 2001] es adecuada. Para tal fin se analizaron dos trabajos de construcción del LEL, escenarios y tarjetas CRC sobre un dominio en común. A partir de las entradas de LEL y de los escenarios hallados, se los cargó en BMW para hallar las tarjetas CRC en forma automática. Luego se comparó si se encontraron las mismas tarjetas CRC en forma manual que en forma automática.

Los resultados fueron satisfactorios, el nivel de similitud de las tarjetas encontradas en común fue alto (tabla 24). En el primer caso de estudio se encontraron en forma manual 16 tarjetas y 24 en forma automática. De ellas, 15 coinciden. Para el segundo caso de estudio, 20 tarjetas de hallaron manualmente y 25 automáticamente. De aquí 16 son comunes. Además, al comparar los dos casos de estudios entre sí, se observa que los resultados son similares. Si bien el LEL del segundo caso de estudio es más extenso (47 entradas contra 35), comparten muchas entradas en común. Los números de los escenarios son parejos, 18 contra 21. Y las tarjetas CRC obtenidas fueron manualmente 16 y 20, y automáticamente 24 y 25. Se puede observar que si bien las entradas de LEL presentan una diferencia (en el segundo caso de estudio se definieron un 34% más de entradas que en el primero) y los escenarios también son más (un 16% más) las tarjetas CRC obtenidas son similares. Si bien 16 y 20 tarjetas se obtuvieron manualmente, sólo 15 y 16 están justificadas. Y en forma automática los números son 24 y 25. Esto muestra que los conjuntos de LEL y escenarios llegan a un punto en el cual no agregan más información, se vuelven redundantes, por lo cual el número de tarjetas CRC se estabiliza.

Tabla 24 – Resumen de los dos casos de estudio.

Caso de estudio	Entradas de LEL	Escenarios	Tarjetas CRC obtenidas		
			Manual y automáticamente	Sólo manualmente	Solo automáticamente
1	35	18	15	1	9
2	47	21	16	4	9

Con respecto de las tarjetas CRC encontradas de más por el proceso automático, las mismas se obtendrían manualmente si se siguiera el algoritmo minuciosamente, sin embargo no están presentes en la derivación manual. Algunas de estas tarjetas de más se corresponden con acciones que el algoritmo encuentra como tarjetas CRC, cuando no pueden ser consideradas como tarjetas, aunque sí como responsabilidades de alguna otra. El criterio de quien aplica el algoritmo manualmente le permite encontrar estas tarjetas inadecuadas y descartarlas. Automáticamente no se podría realizar esta selección. Si bien se podría intentar descartar todas las tarjetas CRC que se formen a partir de un símbolo de LEL que sea acción, no es cierto que todas las acciones no deban traducirse en clases, por ello, es preferible dejar estas entradas incorrectas. Además las acciones se van a transformar en responsabilidades de otras tarjetas y si no es posible encontrar la tarjeta adecuada, da una pauta que el análisis de LEL y escenarios debe ser verificado y validado.

Con respecto de las tarjetas halladas manualmente, pero no automáticamente se analizó en detalle el origen de cada una de las cinco tarjetas que no fueron encontradas por el BMW. Para cada una de ellas se llegó a la conclusión de que la información recogida en el LEL y escenarios era insuficiente para que se obtengan las tarjetas CRC aplicando el algoritmo. De esto se deduce que el algoritmo no fue aplicado exactamente, y que durante la construcción de las tarjetas, se ha obtenido información adicional, que completa al LEL y los escenarios, pero que no fue volcada a ellos.

A partir del análisis de las tarjetas CRC obtenidas a través de la aplicación del algoritmo de derivación en forma manual y del implementado en BMW en los dos casos de estudio, y de la similitud entre ambos resultados, concluimos que la implementación en BMW es una implementación adecuada del algoritmo presentado en [Leonardi 2001]. Sin embargo, resta verificar si el algoritmo implementado en BMW obtiene tarjetas CRC útiles en la confección de un modelo preliminar de objetos.

4.4.2. Usabilidad del algoritmo implementado en BMW

Para evaluar la usabilidad del algoritmo de derivación implementado en BMW se optó por derivar tarjetas CRC de un conjunto de LEL y escenarios sobre, y comparar las tarjetas obtenidas contra un modelo de objetos de una aplicación a utilizar en el mismo dominio. El dominio elegido es el de la administradora de planes de ahorro.

Derivación de tarjetas CRC a partir de LEL y escenarios

A partir del caso de estudio número uno, se completaron las entradas de LEL y los escenarios de forma tal que sea autocontenido y que no queden aspectos sin definir.

Para completar el caso de estudio uno, se estudiaron los casos de estudio uno y dos, la especificación de [Parravicini 1997] y se sumó el conocimiento previo del autor sobre el tema. Con todo este conocimiento se trabajó sobre el caso de estudio número uno. Se analizó cada una de las entradas de LEL y de los escenarios. Se buscó que la información sea consistente, es decir que la misma verdad sea referenciada de igual forma, por distintos elementos. También se buscaron aspectos poco claros y precisos para extender la definición. Incluso se agregaron tres nuevas entradas (tabla 25).

Tabla 25 – Entradas de LEL agregadas al caso de estudio uno.

<i>LEL</i>
<i>cambiar el bien tipo adjudicado</i>

A partir del conjunto de entradas de LEL y escenarios de la tabla 20, con el agregado de las entradas de LEL de la tabla 25 y las modificaciones realizadas, a través del algoritmo de derivación implementado en el BMW se obtuvieron 28 tarjetas (tabla 26). El LEL, los escenarios y las tarjetas CRC de este trabajo se detallan en el anexo 9.1.

Tabla 26 – Tarjetas CRC derivadas para el caso de estudio de la administradora del plan de ahorro.

Tarjetas CRC
Adherente
Adjudicatario
Administradora
Aseguradora
Banco
bien tipo
cambiar el bien tipo adjudicado
Cambio de bien tipo
Comunica fehacientemente
Cuota de ingreso
cuota mensual
derecho de adjudicación
disolver el grupo
Evaluar licitaciones
Expulsar adherente
Fabricante
Grupo
Licitación
Plan
planilla de adhesión
Rechazar bien tipo
Renunciar al plan de ahorro
seguro de bien tipo
seguro de vida
Solicitante
solicitud de adhesión
Sorteo
Transferir plan

El nuevo conjunto de tarjetas CRC incluye todas las tarjetas CRC encontradas para el caso de estudio uno (las encontradas en forma manual y automática a la vez, más las sólo encontradas manualmente, más las sólo encontradas automáticamente). La tarjeta **"comunicación fehaciente"** que sólo se había derivado en forma manual se derivó en forma automática con la revisión.

Luego de completar el caso de estudio original se encontraron en forma automática 3 tarjetas CRC más: **"cambiar el bien tipo adjudicado"**, **"cambio de bien tipo"** y **"disolver grupo"**.

Las 28 tarjetas CRC obtenidas (tabla 26) conforman la mitad de la información necesaria para poder verificar la utilidad del algoritmo de derivación. La otra mitad lo conforma un modelo de objetos contra el cual comparar las tarjetas.

Diseño del diagrama de clases a partir de casos de uso

A partir de la información disponible del ejercicio anterior, pero sin utilizar las tarjetas CRC obtenidas se realizó el diseño.

El diagrama estuvo conducido por siete casos de uso [Jacobson 1994] que describen el trabajo de la administradora de planes de ahorro. Los casos de uso son: armar grupo, pagar cuota, realizar sorteo, licitar, evaluar licitaciones, echar adherente y disolver grupo (figura 14).

El diagrama de clases (figura 15) está compuesto por 17 objetos. Dos de ellos tienen comportamiento relevante: Administradora y Adherente.

La Administradora conoce a los inscriptos a conformar el plan de ahorro. Dispone de dos mensajes, en primer lugar solicitarIngreso, con el cual los solicitantes indican su deseo. Y luego con el mensaje armarPlan, se arma el plan de ahorro a partir de los inscriptos. La administradora mantiene contacto con los bancos donde se realizan los pagos. También guarda en su poder las propuestas de licitación que hacen los adherentes. Su protocolo tiene dos mensajes para el pago de cuotas. El mensaje cuotaPaga lo recibe de los bancos para indicarle que algún adherente la ha hecho efectiva. Por otro lado, las cuotas vencidas sólo se pueden pagar en la administradora, así que los adherentes le deben enviar el mensaje pagarCuotaVencida para este propósito. Otra función que realiza la administradora es la de realizar sorteos y licitaciones. Así que responde al mensaje sortear para llevar a cabo el mismo. Para las licitaciones, el mensaje licitar lo envían los adherentes para hacer una propuesta. Luego, el proceso de apertura de sobres y la determinación de la mejor propuesta se realiza enviando el mensaje evaluarLicitacion. Algunas situaciones extraordinarias también son realizadas por la administradora. Por ejemplo, echarAdherente es una de sus responsabilidades. Si el mismo se atrasa con las cuotas o rechaza la adjudicación del bien tipo en tres oportunidades, la administradora tiene la obligación de echarlo. Como producto de echar a un adherente, puede suceder que el plan de ahorro se quede con una cantidad insuficiente de adherentes y el mismo se deba disolver. Por lo tanto, la administradora, también es responsable de disolverPlan.

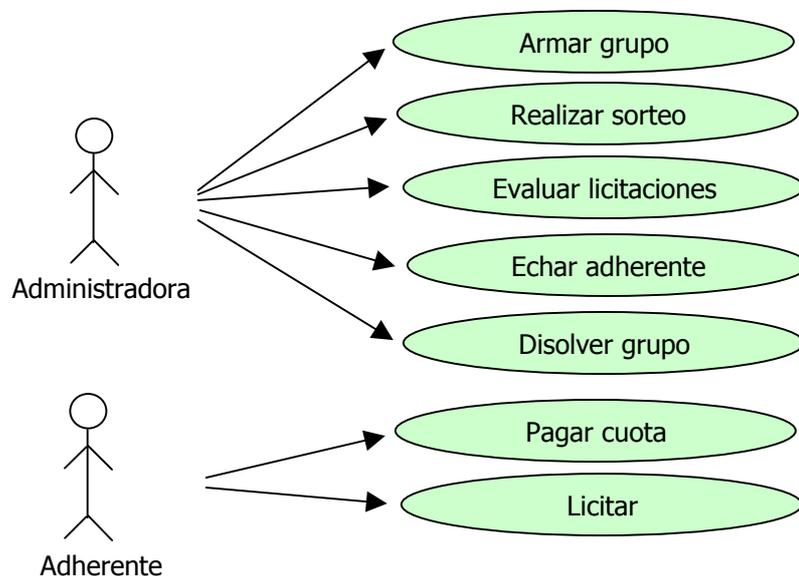


Figura 14 – Casos de uso del sistema de administración de planes de ahorro.

La jerarquía de adherentes es otro elemento importante. Se parte de una clase abstracta Persona, que además de los atributos ordinarios como ser nombre, apellido y numeroDeDocumento, también está asegurado en una compañía. De esta clase, cuelgan la clase concreta Solicitante y la clase abstracta Adherente. El motivo de que las Personas esten aseguradas se debe a que una de las exigencias de los solicitantes es tener seguro de vida, ya que este se debe hacer cargo de saldar el plan en caso de fallecimiento. Luego de cumplidos los requisitos, se le puede indicar al solicitante que está aceptado o rechazado. Si el Solicitante está aceptado, pasa a ser Adherente. Adherente es una clase abstracta, la cual lleva control del numeroDeCuotasPagas, que se indica a través del mensaje pagoCuota. De esta clase, cuelgan dos clases concretas, una es Normal y la otra es Adjudicatario. Un adherente Normal es aquel que no ha recibido el BienTipo aún. Así que el Adherente Normal, mantiene registro del numeroDeRechazos, ya que se lo debe echar cuando llegue a tres. También responde al mensaje estasAdjudicado, frente al cual puede aceptar o no el BienTipo. El Adherente Normal, debe asegurarBienTipo, condición obligatoria luego de indicarle que estaAdjudicado y que opta por aceptarlo. Por último, posee la posibilidad de cambiarBienTipo. Con respecto al Adjudicatario, todo el protocolo referente a la adjudicación no lo posee ya que le es irrelevante, sólo hereda el mensaje pagoCuota.

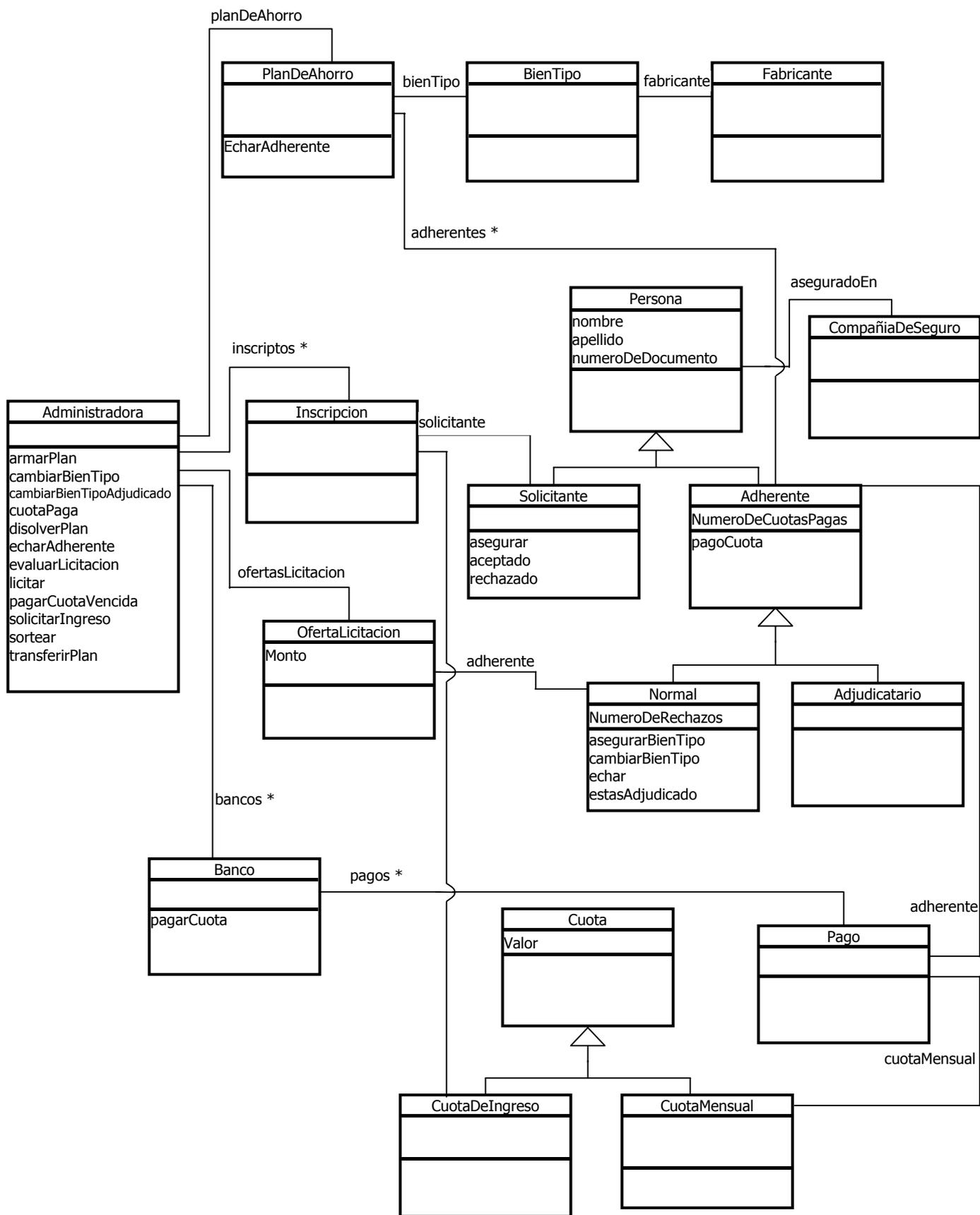


Figura 15 – Diagrama de clases del sistema de administración de planes de ahorro.

PlanDeAhorro es otra clase que merece describirse. Cuenta con dos variables de instancias: bienTipo y adherentes. BienTipo es el que se entrega mes a mes y adherentes son quienes forman el grupo. Y como responsabilidades sólo tiene echarAdherente. A pesar de su simplicidad, es de esperar que en un análisis posterior, gran parte del comportamiento de la Administradora recaiga en PlanDeAhorro. Ya que el PlanDeAhorro es quien tiene las reglas del cual participan los adherentes, y quien sortea mes a mes un BienTipo por el cual también licitan.

El BienTipo es un vehículo con ciertas características, pero lo más importante de él es el Fabricante, quien puede discontinuar un modelo o variar su precio, provocando que se modifique el precio o la cantidad de cuotas del PlanDeAhorro.

Cuota también es una jerarquía. Hay una clase abstracta Cuota, con una variable de instancia valor. Y de la misma cuelgan dos clases concretas. CuotaDeIngreso, el importe que debe pagar cada Solicitante. Y CuotaMensual, el valor que mes a mes paga cada Adherente para cumplir con el PlanDeAhorro.

Para ejemplificar la dinámica del modelo de objetos se muestran tres diagramas de secuencia: el armado de un plan, la realización del sorteo y echado de un adherente.

El armado del plan consiste en registrar las solicitudes de inscripción de los postulantes. Cuando la cantidad de solicitantes es suficiente, se deben evaluar las inscripciones. Se notifican a todos los postulantes si es aceptado o rechazado, para luego constituir el plan de ahorro (figura 16).

Realizar el sorteo consiste en elegir una persona del grupo de adherentes que poseen las cuotas al día (figura 17).

Echar un adherente implica desvincularlo del plan de ahorro. Luego se debe evaluar si la cantidad de adherentes es suficiente para que el círculo continúe en actividad. De no ser así se disuelve el plan (figura 18).

Comparación de las tarjetas CRC y el diagrama de clases

Podemos considerar útil al algoritmo de derivación de tarjetas CRC implementado en BMW si las clases que conforman el modelo de objetos de la administración de plan de ahorro se encuentran entre las tarjetas CRC derivadas.

De los 17 objetos definidos en el diagrama, 14 son clases concretas. De estas 14 clases, 12 están en la lista de tarjetas: PlanDeAhorro (como plan), BienTipo, Fabricante, Administradora, Inscripción (como planilla de adhesión), Solicitante, CompañíaDeSeguro (como aseguradora), Normal (como adherente), Adjudicatario, Banco, CuotaDeIngreso y CuotaMensual.

La jerarquía de Persona tiene todas las clases concretas definidas en tarjetas CRC, sólo que hay un cambio de nombres. La clase concreta Normal, subclase de Adherente, figura con el nombre Adherente en las tarjetas, mientras que en el diagrama Adherente se denomina a la clase abstracta. Se optó por esta decisión ya que un Adjudicatario también es un Adherente. Solicitante y Adjudicatario poseen la misma denominación en las tarjetas y en el diagrama.

No es necesario que las clases abstractas se encuentren en las tarjetas CRC, ya que las clases abstractas surgen de la factorización del comportamiento y del conocimiento de las clases concretas. La clase abstracta Cuota surge de abstraer conocimiento común de CuotaDeIngreso y CuotaMensual por ejemplo.

Si es necesario evaluar que sucede con las dos clases concretas del diagrama de clases, que no están en la lista de tarjetas: OfertaLicitación y Pago. Esto se debe a que estas clases aparecen al refinar el diseño. OfertaLicitación modeliza el monto que ofrece cada adherente en la licitación. Cuando se implementa el mensaje licitar de la Administradora, en ese punto hubiera surgido la necesidad. Lo mismo sucede con la clase Pago. El banco registra el Pago de una CuotaMensual de un Adherente. En la implementación del mensaje pagarCuota de Banco hubiera sido necesario definir esta clase.

El análisis previo vincula cada objeto del diagrama de clases, con su par en la lista de tarjetas CRC. Del proceso inverso, es decir para cada tarjeta CRC buscar su par en el diagrama de clases, surge que hay cuatro tarjetas de más, candidatas a convertirse en objetos. Ellas son: **"derecho de adjudicación", "grupo", "seguro de bien tipo" y "seguro de vida"**.

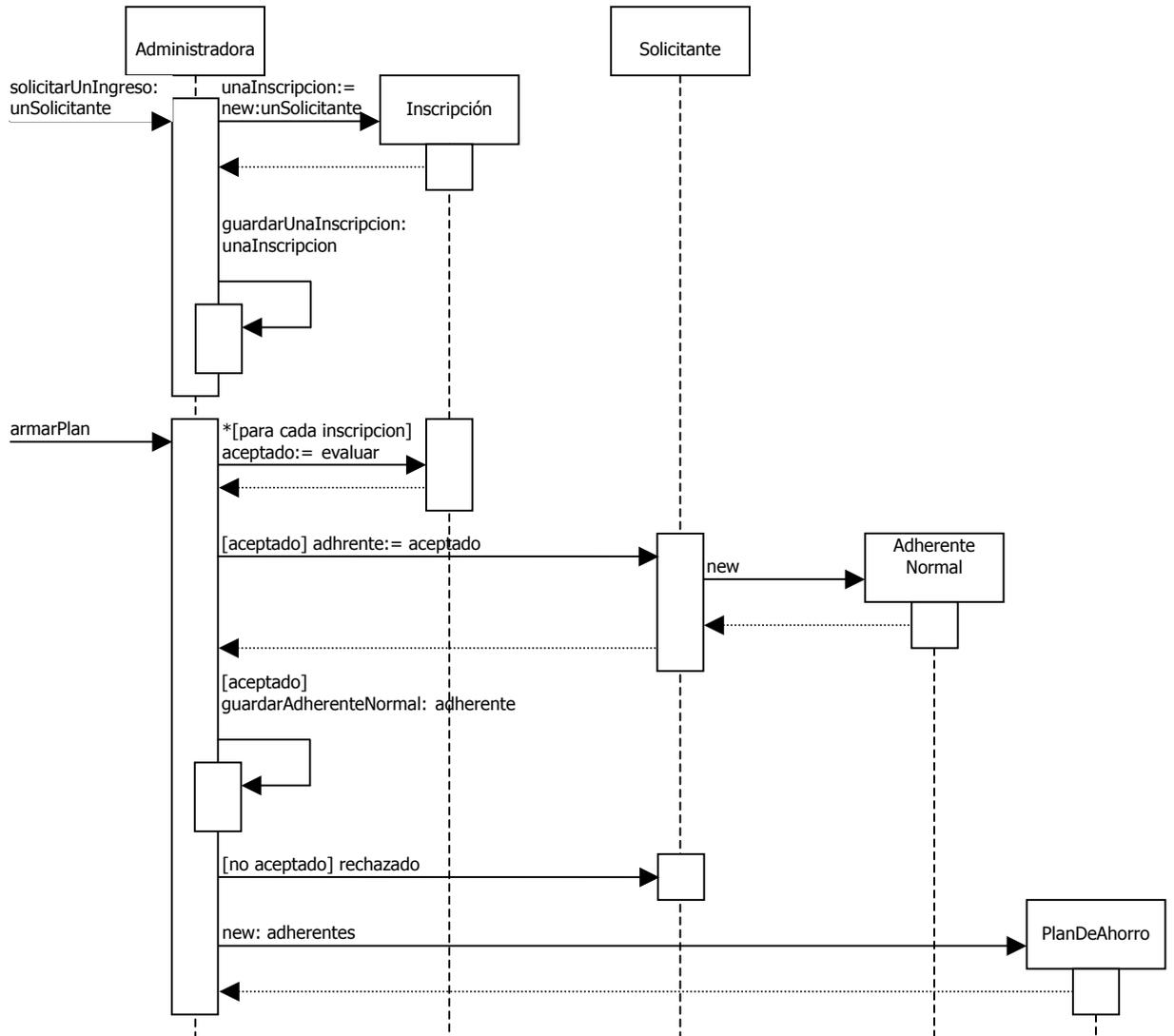


Figura 16 – Diagrama de secuencia del armado del plan.

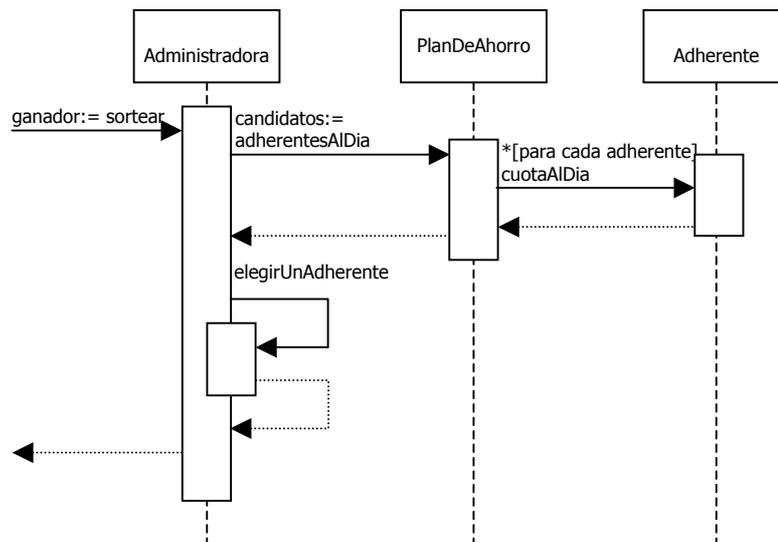


Figura 17 – Diagrama de secuencia de realizar el sorteo.

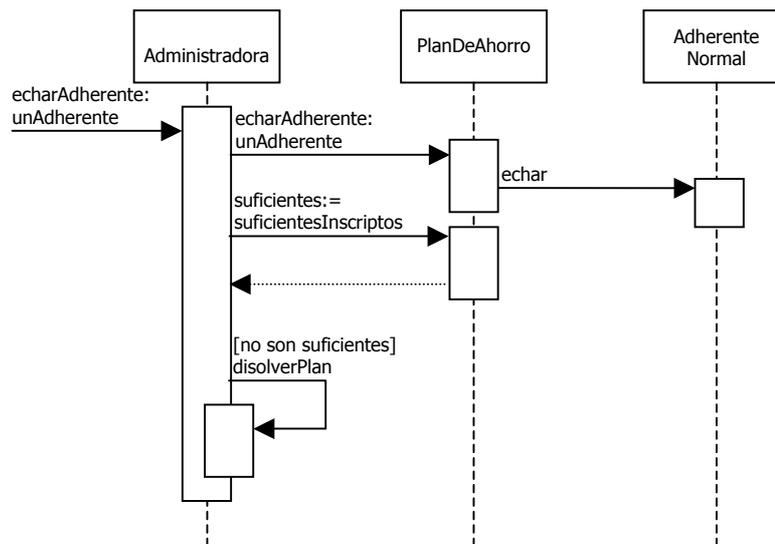


Figura 18 – Diagrama de secuencia de echar un adherente.

La primera tarjeta, **"derecho de adjudicación"**, es un arancel que se debe abonar, para poder recibir el bien tipo. Trantándose de un plan de ahorro, en donde el dinero es importante para la compra del bien, el derecho de adjudicación es relevante, sólo que todos los aspectos de aranceles se han dejado de lado en el diagrama de clases.

La segunda tarjeta, **"grupo"**, está implícita en la variable de instancia **"adherentes"** de la clase PlanDeAhorro. Las responsabilidades del objeto **"grupo"** son las que determinan si se debiera incluir una clase o si es simplemente una colección como la que se modeló en el diagrama.

La tarjeta **"seguro de bien tipo"** se ha omitido en el diagrama. El adjudicatario debiera tener un seguro de su automóvil, pero este hecho no está modelizado.

La tarjeta **"seguro de vida"** esta implícita en el hecho de que la persona está aseguradaEn. Al igual que con sucede con grupo, habría que analizar las responsabilidades para determinar si es necesario modelar el objeto SeguroDeVida o es suficiente con indicar la compañía de seguro.

El resto de las tarjetas CRC que no tienen un objeto asociado se debe a que son responsabilidades a cubrir por algún objeto. De los 12 mensajes que responde la Administradora, 9 se han encontrado por el algoritmo de derivación: solicitarIngreso (como solicitud de Adhesión), sortear (como sorteo), evaluarLicitacion, licitar, echarAdherente (como expulsarAdherente), cambiarBienTipo (como cambio de bien tipo), cambiarBienTipoAdjudicado, transferirPlan y disolverPlan (como disolverElGrupo).

Conclusión de la comparación entre tarjetas CRC y el diagrama de clases

De las 14 clases concretas en el diagrama de clases, 12 se derivaron como tarjetas CRC. Las dos que no fueron derivadas son clases triviales. Además, el algoritmo derivó cuatro tarjetas que no tienen su contraparte en el diagrama de clases y que son relevantes al dominio. Por lo tanto, el algoritmo de derivación de tarjetas CRC implementado en el BMW es útil al brindar la materia prima para la construcción de un modelo de objetos.

4.5. Futuros trabajos

Se está realizando un estudio con estudiantes de postgrado de la Facultad de Informática de la Universidad Nacional de La Plata, con el fin de extender la evaluación del algoritmo. Los estudiantes deben realizar el LEL y los escenarios de un dominio presentado en OOPSLA'1997. Luego, se compararán las tarjetas CRC obtenidas del LEL y los escenarios contra los diseños elaborados en OOPSLA.

El caso de estudio utilizado es el de una planta procesadora de papas fritas. La misma recibe bolsas con papas cortadas en bastones y la planta debe quedarse con las papas aceptables y tratar de reparar las defectuosas. La aceptación depende de la longitud de las papas fritas y de su apariencia.

5. Forward traceability en el modelo de LEL y escenarios

El modelo de LEL y escenarios junto con las reglas de derivación y las tarjetas CRC constituyen un conjunto consistente. El LEL y los escenarios representan un modelo del contexto del dominio mientras que las tarjetas CRC son un elemento de análisis y diseño. Como la realidad cambia, el modelo del contexto (LEL y escenarios) también lo deben hacer, y estos cambios se deben trasladar a las tarjetas CRC.

La sección 5.1 justifica la necesidad de aplicar forward traceability. La sección 5.2 discute alternativas para aplicar traceability forward, llegando a la conclusión de que se necesitan escribir reglas de traceability forward. En la sección 5.3 se analiza el algoritmo de derivación con el fin de entender los vínculos entre LEL, escenarios y tarjetas CRC. En la sección 5.4 se estudian los cambios que pueden sufrir LEL, escenarios y tarjetas CRC. A partir del vínculo entre los elementos y los cambios que pueden sufrir, en la sección 5.5 se describen las reglas utilizando la propuesta de Rolland [Rolland 1994]. En la sección 5.6 se compara el modelo propuesto con otros modelos

5.1. Objetivo

El objetivo primario que se persigue es mantener el conjunto formado por LEL, escenarios y tarjetas CRC en forma consistente de acuerdo a las reglas de derivación de tarjetas CRC. Esto significa que dado un conjunto de tarjetas CRC derivadas a partir de un LEL y un conjunto de escenarios, si se altera algún elemento en el LEL o en los escenarios, las tarjetas CRC deben sufrir los cambios necesarios para representar los mismos elementos que se obtendrían si se derivaran nuevamente las tarjetas CRC.

Tanto LEL como escenarios capturan el modelo del contexto. A partir de ellos se obtienen por medio de reglas de derivación tarjetas CRC. Ellas se utilizan para realizar el diagrama de clases. EL LEL, escenarios, tarjetas CRC y el posterior diagrama de clases están encadenados. Los cambios que se producen en el LEL y los escenarios se deben propagar por todos los elementos. Un modelo por etapas, en el cual se complete el LEL, para pasar a construir los escenarios y luego derivar las tarjetas CRC antes de hacer un diagrama de clases es impensable. La realidad varía día a día, y con ella los modelos de contexto y los sistemas de software. Por lo tanto, se necesitan reglas que en forma sistemática propaguen los cambios entre los distintos elementos.

Para ejemplificar el objetivo de realizar forward traceability se utiliza el ejemplo del Client Oriented Requirements Baseline del organismo gubernamental de la República Argentina (figura 7).

Se utiliza un subconjunto de los elementos y de los atributos a fin de mostrar los aspectos de forward traceability (figura 19). La entrada de LEL "**abogado**" tiene el detalle de los impactos. Luego hay 3 entradas más: "**juicios**", "**particulares**" y "**cédula**", las cuales solamente se mencionan. Del escenario "**intimar una razón social**" interesa solamente el actor que es "**abogado**". Y cuatro tarjetas CRC completan la figura. Una tarjeta primaria: "**abogado**" y tres tarjetas secundarias: "**juicios**", "**particulares**" y "**cédulas**". De la tarjeta primaria se detallan las responsabilidades, de las otras no se muestra ninguna información.

Las flechas indican la justificación de la existencia de cada elemento. La tarjeta primaria "**abogado**" existe ya que hay un escenario ("**Intimar una razón social**") en la cual "**abogado**" es el actor. Y a su vez "**abogado**" es una entrada de LEL. Luego, las responsabilidades de la tarjeta primaria "**abogado**" son los impactos de la entrada de LEL. Las tres tarjetas secundarias existen, ya que en las responsabilidades de la tarjeta primaria "**abogado**" aparecen tres entradas de LEL.

El conjunto presentado de LEL, escenarios y CRC (las cuales se obtuvieron aplicando el algoritmo de derivación) puede sufrir cambios. Por ejemplo, la información del abogado puede cambiar. Tómese como ejemplo que lo único que se conoce del abogado es que lleva adelante los juicios del fisco contra los particulares, pero no es más responsable de armar y enviar las cédulas de notificación.

Con el cambio de los impactos de la entrada de LEL también se modifica la tarjeta CRC "**abogado**". Como las responsabilidades de la tarjeta CRC reflejan los impactos de la entrada de LEL, al cambiar los impactos se deben cambiar las responsabilidades. Previo a la modificación de la tarjeta CRC "**abogado**" se derivaban tres tarjetas: "**juicio**", "**particulares**" y "**cédula**" de ella. Sin embargo, con el cambio introducido, la tarjeta "**cédula**" no está en condiciones de derivarse, puesto que los impactos de la tarjeta primaria CRC "**abogado**" no menciona a la "**cédula de notificación**".

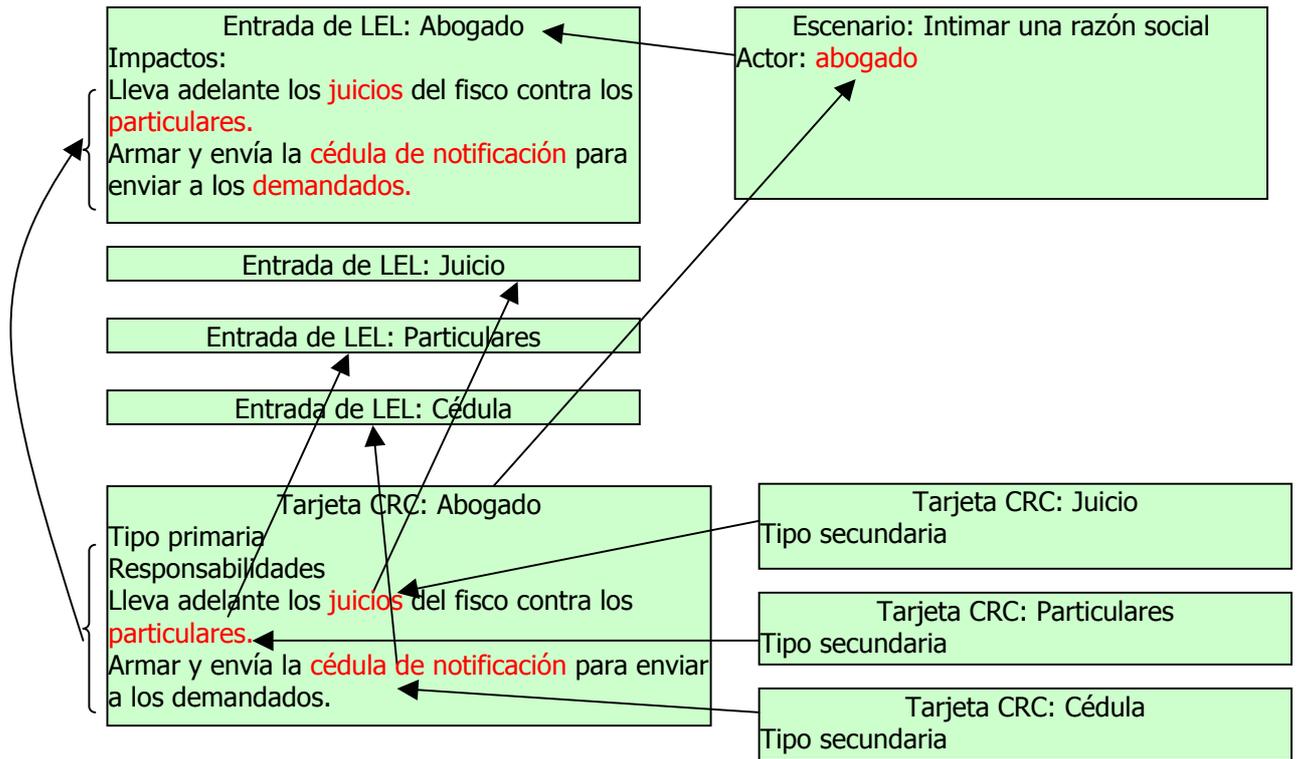


Figura 19 – Derivación de una tarjeta CRC a partir de LEL y escenarios.

La figura 20 muestra los cambios al modificar las responsabilidades de la entrada de LEL "abogado". El texto tachado y las líneas punteadas tanto en las flechas como en las cajas indican que esos elementos no deben existir.

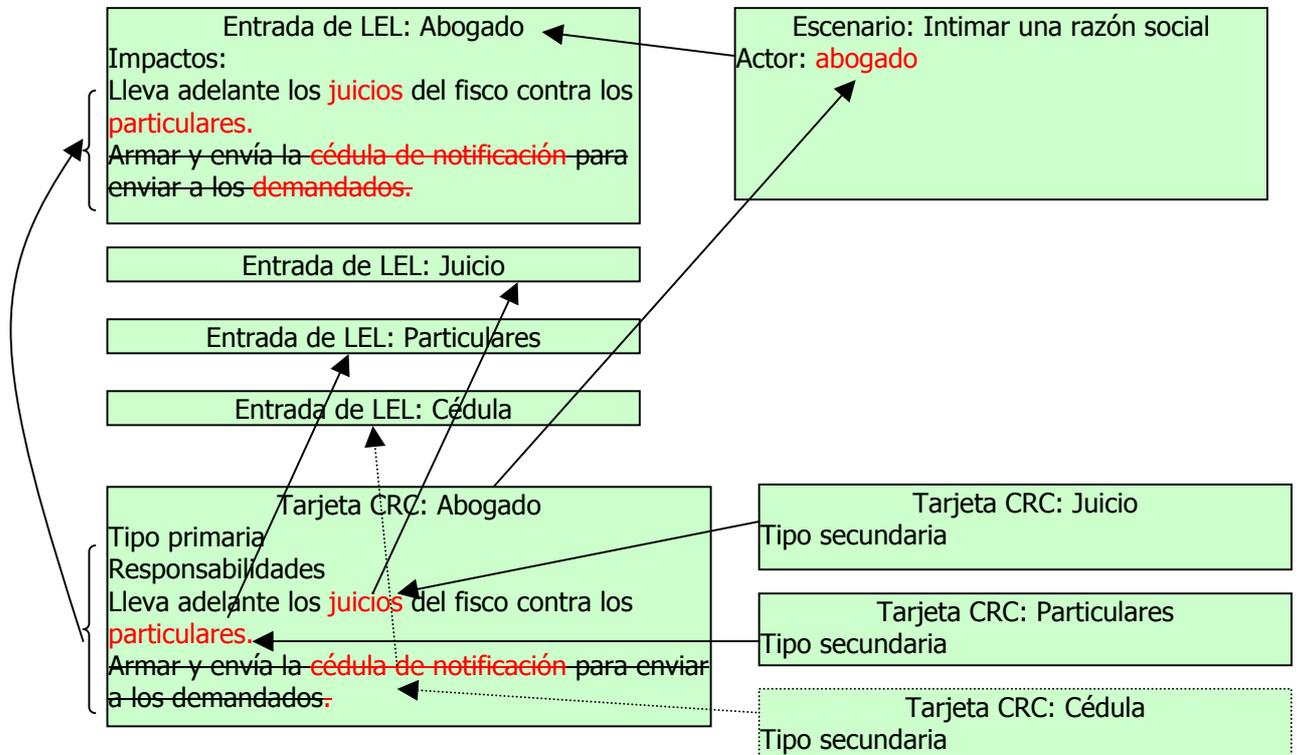


Figura 20 – Modificación de los impactos de la entrada de LEL abogado.

Los cambios no solamente pueden originar la aparición o desaparición de tarjetas CRC. Se pueden presentar cambios más sencillos que también requieran su propagación. Por ejemplo, el término **"particulares"** puede estar mal empleado, y en realidad deben ser **"demandados"**. La entrada de LEL **"particulares"** no se denomina más así, pasándose a llamar **"demandados"**. Sin embargo, de la entrada de LEL **"abogado"** se referencia a **"particulares"**, y como **"particulares"** cambió su denominación también se debe cambiar su referencia. Ahora bien, los impactos de la entrada de LEL **"abogado"** indican que lleva adelante los juicios de los demandados, pero existe una tarjeta CRC **"abogado"**, cuyas responsabilidades son una copia de los impactos de la entrada de LEL con el mismo nombre. De modo que también se debe alterar las responsabilidades de la tarjeta. Y por último, la tarjeta secundaria **"particulares"** debe renombrarse a demandado, como se ha hecho en todas las referencias anteriores (figura 21). Las flechas numeradas desde uno hasta tres, indican la secuencia en que suceden las modificaciones.

Este ejemplo muestra que no solamente modificaciones relacionadas con el algoritmo de derivación son importante capturar para que el Client Oriented Requirements Baseline se mantenga consistente. Una modificación sencilla como ser una palabra por otra (el término que define a una entrada de LEL) también necesita ser tomada en cuenta, y se deben realizar los cambios necesarios para que la consistencia del conjunto se mantenga. Este caso muestra un detalle simple: una persona que se denominaba particular pasa a ser denominado demandado. Este caso hipotético se pudo haber dado por un problema de comunicación durante el relevamiento de LEL y escenarios. Pero una vez detectada la equivocación, debe propagarse el cambio hasta las tarjetas CRC, sino en implementación se estará hablando de una clase **"particular"**, cuando en los requerimientos se habla de **"demandado"**, y con esta confusión nunca se podrá validar la implementación contra los requerimientos.

Por lo tanto, es absolutamente necesario un método sistemático y completo que dicte la forma de propagar los cambios sobre todos los elementos del Client Oriented Requirements Baseline.

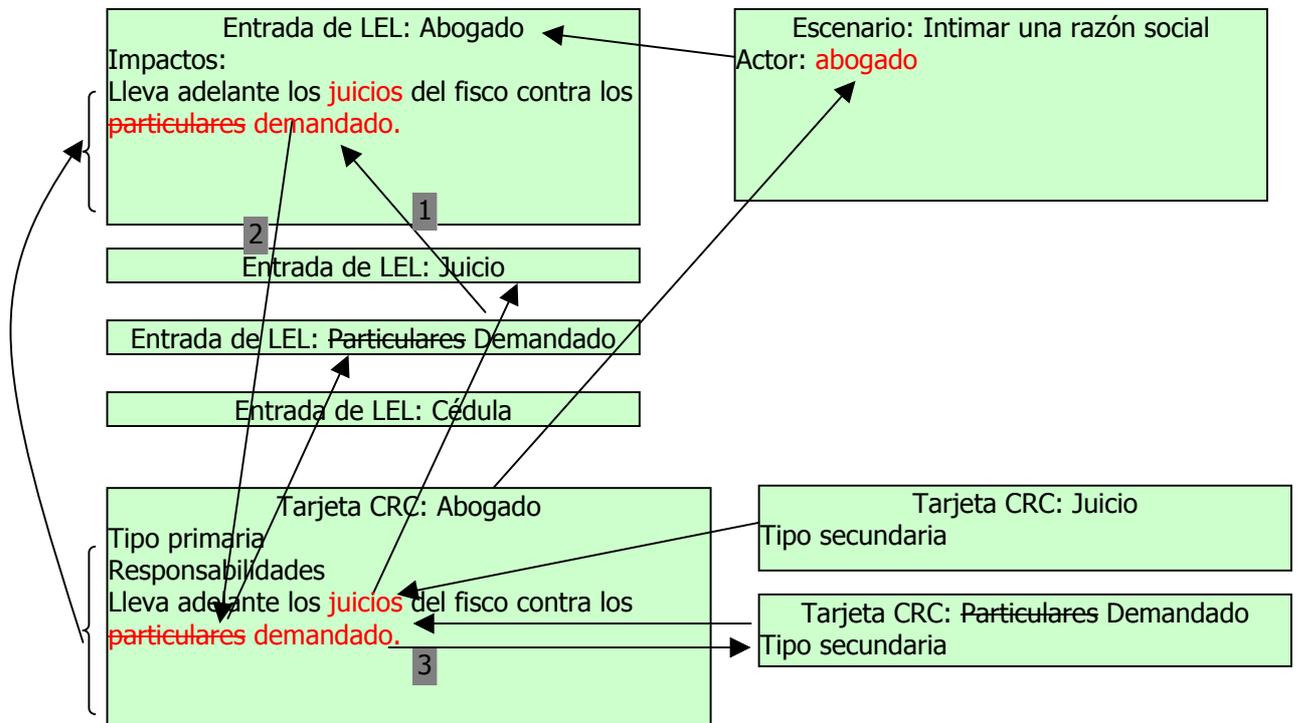


Figura 21 – Modificación de término de la entrada de LEL "particulares" por "demandado".

5.2. Medio de lograr el objetivo

Una alternativa simple para resolver este problema sería frente a cualquier tipo de modificación en el LEL o los escenarios aplicar nuevamente las reglas de derivación para obtener las nuevas tarjetas CRC.

Esta solución tiene el inconveniente de que necesita aplicar las reglas por cada modificación del LEL y los escenarios. Entonces, se podría demorar la aplicación de las reglas hasta no cerrar el LEL y los escenarios. De todas formas, las modificaciones podrían surgir una vez cerrados y se necesitaría aplicar las reglas por cada modificación. Si el algoritmo está implementado en una herramienta automatizada, el aplicado del algoritmo no sería costoso porque estaría automatizado y lo haría una computadora. El volumen de información podría ser una desventaja que podría hacer lento el proceso de derivación, pero de todas formas sería automatizado.

Sin embargo, hay un motivo por el cual aunque el algoritmo esté implementado en una aplicación que corra en una máquina en la cual no demore nada, esta solución no es la más indicada. El punto es que LEL, escenarios y tarjetas CRC es un modelo que está dentro de otro mayor, puesto que a partir de las tarjetas CRC se realiza un diagrama de clases. El mismo se deja librado al criterio del diseñador. No hay reglas que automaticen este paso. Por lo tanto, derivar nuevamente tarjetas a partir de LEL y escenarios, implica verificar las variaciones entre las tarjetas viejas y las nuevas, para determinar cuales fueron los cambios. A partir de los cambios producidos en las tarjetas CRC, se los debe llevar al diagrama de clases, para que estos cambios sigan fluyendo hacia la implementación.

Es importante destacar que tal vez, ni siquiera se quiera modificar el LEL o los escenarios, tal vez se necesite medir los impactos de las posibles modificaciones. Medir los impactos en todo el proceso de desarrollo de software es un elemento importante. El desarrollo de software está regido por una variable importante: tiempo. El cual a su vez determina el costo. Es común medir los impactos de modificaciones en los requerimientos para determinar como varía el costo. Medir los impactos en este modelo sólo es posible si frente a las modificaciones se puede determinar el rastro de las modificaciones que le suceden.

Por lo tanto, es necesario determinar un método sistemático que determine los cambios necesarios para asegurar la consistencia. Realizamos un análisis para desmembrar las reglas de derivación, de forma tal de definir reglas de traceability forward, las cuales sean aplicables en las distintas situaciones de modificaciones de LEL y los escenarios, para asegurar que las tarjetas CRC en todo momento se correspondan con las tarjetas CRC que se obtendrían de aplicar el algoritmo de derivación al LEL y los escenarios.

5.3. Análisis del algoritmo de derivación de tarjetas CRC

Para poder desmembrar las reglas de derivación es necesario hacer un análisis profundo de las mismas y comprender su funcionamiento. Las reglas se tratan en detalle en [Leonardi 2001] y también se ha justificado su esencia en secciones previas de este trabajo. Sin embargo, en este apartado se profundiza acerca de la mecánica de las mismas.

Básicamente las reglas se aplican sobre el LEL y los escenarios para obtener tarjetas CRC. Si bien en este trabajo se mencionó una serie de heurísticas para obtener escenarios a partir de entradas de LEL, las reglas de derivación de tarjetas CRC no toman este supuesto.

La mecánica de las reglas de derivación determina la aparición de tarjetas primarias y secundarias. Mas allá del tipo de tarjeta, hay una relación directa entre una tarjeta CRC y una entrada de LEL. Si existe una tarjeta CRC con cierto nombre, es porque existe una entrada de LEL con el mismo nombre. La relación no se da solamente en el nombre de la misma. La tarjeta CRC está definida por dos atributos más: responsabilidades y colaboraciones. Las responsabilidades son exactamente los mismos impactos que tiene la entrada de LEL que origina a la tarjeta CRC. De esta forma, una tarjeta CRC es una entrada de LEL, en donde los impactos pasan a llamarse responsabilidades.

Sin embargo quedan algunos aspectos no tratados. No todas las entradas de LEL se convierten en tarjetas CRC. Las que sí lo hacen, pueden ser de dos tipos: primarias y secundarias. Luego, las tarjetas CRC además de responsabilidades tienen colaboraciones.

Para que una entrada de LEL se convierta en tarjeta CRC, debe pasar una de entre dos alternativas. O bien tiene que ser un actor de un escenario (tarjetas primarias) o tiene que colaborar con una tarjeta CRC primaria (tarjetas secundarias). Las tarjetas primarias tienen esta denominación puesto que poseen un comportamiento relevante dentro del dominio, llevan a cabo las acciones del escenario. Mientras que las tarjetas secundarias colaboran con las tarjetas primarias para que ellas cumplan con sus responsabilidades. De esta forma, las tarjetas CRC primarias son entradas de LEL que están en la lista de actores de los escenarios. Y las tarjetas CRC secundarias son entradas de LEL que están presentes en las responsabilidades de las tarjetas CRC primarias.

El único punto que resta por tratar son las colaboraciones. Por la forma de definir las tarjetas CRC secundarias son colaboradoras de las primarias. Sin embargo, se pueden encontrar más colaboraciones en los episodios de los escenarios. Los episodios describen las tareas a llevar a cabo para lograr el objetivo del escenario. En estas tareas colaboran tarjetas CRC. Por lo tanto, los episodios también son fuentes de colaboraciones. De esta forma, las colaboraciones dependen de la ocurrencia de distintas tarjetas de CRC en los episodios de los escenarios. Es decir, una tarjeta CRC colabora con otra tarjeta CRC, si las entradas de LEL que originan a cada uno de las tarjetas (independientemente de la naturaleza de cualquiera de las dos: primarias o secundarias) se encuentran en los episodios de un mismo escenario.

Las reglas de derivación pueden ser apreciadas gráficamente en la figura 22. El elemento central del gráfico es la tarjeta CRC "abogado", a partir de ella se pueden apreciar los distintos aspectos de las reglas de derivación. Las relaciones de los elementos se muestran utilizando flechas.

En primer lugar se muestra que la tarjeta CRC "abogado" es una entrada de LEL, como lo indica una flecha que parte de la tarjeta y finaliza en la entrada de LEL. Sin embargo, la flecha pasa por el escenario "Intimar una razón social", específicamente por el atributo actor, el cual no es otro que "abogado". Esto grafica que las tarjetas CRC primarias se corresponden con entradas de LEL que son actores de escenarios.

Luego, las responsabilidades de la tarjeta CRC abogado, son una copia de los impactos de la entrada de LEL con el mismo nombre (hay una flecha que parte de una llave de la tarjeta "abogado" y llega a otra llave en la entrada de LEL "abogado")

En el ejemplo se muestra una tarjeta CRC secundaria: "cédula de notificación". La misma es una entrada de LEL, que se referencia en las responsabilidades de la tarjeta CRC primaria. Una flecha que parte de la tarjeta CRC "cédula de notificación" y llega a la entrada de LEL con el mismo nombre, pasando por el atributo responsabilidades de la tarjeta primaria "abogado", pone de manifiesto su naturaleza.

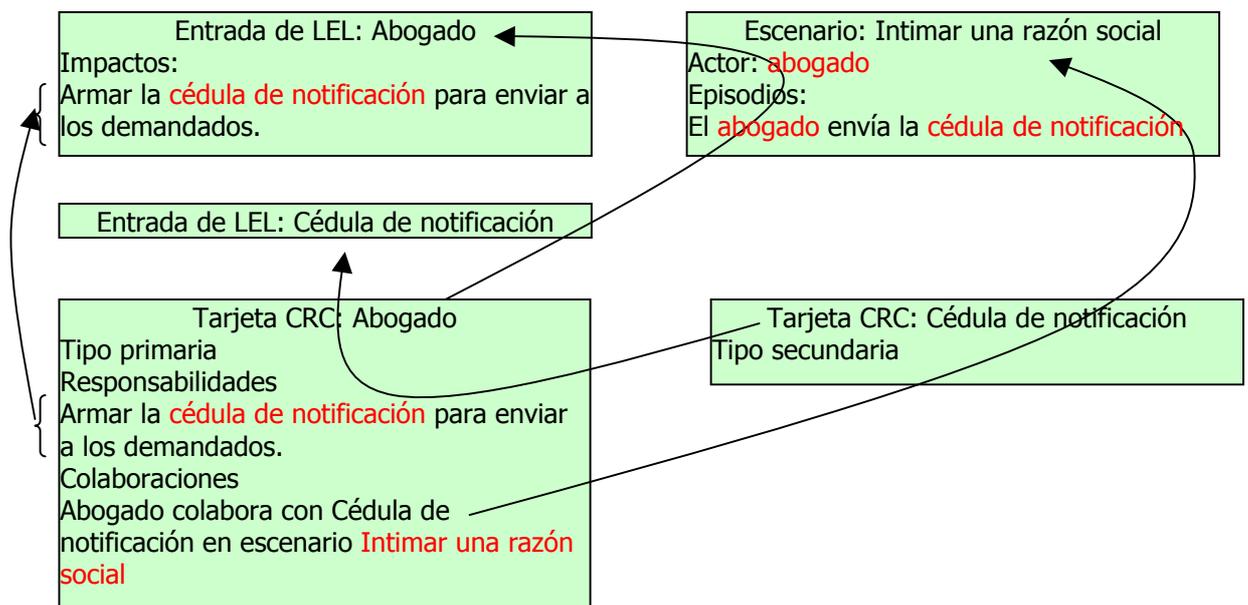


Figura 22 – Mecánica de las reglas de derivación.

Con respecto a las colaboraciones, la tarjeta "abogado" colabora con la tarjeta "cédula de notificación" en el escenario "intimar una razón social". La colaboración se da, porque las entradas de LEL que originan a ambas tarjetas se encuentran en el escenario en cuestión. Este hecho se grafica en la figura con la flecha que parte del párrafo de colaboraciones de la tarjeta "abogado", con fin en el escenario "intimar una razón social", previo paso por la tarjeta "cédula de notificación".

En la figura 22 se sintetizan a través de las flechas, los elementos a analizar en profundidad para definir las reglas que aseguren la consistencia del Client Oriented Requirements Baseline. De

todas formas, con analizar simplemente esos elementos no es suficiente. Hace falta un análisis exhaustivo de los cambios que se pueden dar en el Client Oriented Requirements Baseline, y del impacto que tienen los mismos. Una vez analizadas todas las situaciones con los impactos, es posible escribir las reglas de traceability forward.

5.4. Análisis de los cambios posibles en el Client Oriented Requirements Baseline

Los elementos que conforman el Client Oriented Requirements Baseline son tres: entradas de LEL, escenarios y tarjetas CRC. Cada uno de estos elementos tiene distintos atributos. Las entradas de LEL están formadas por el término que lo identifica (que puede ser uno o varios sinónimos), la noción y los impactos. El escenario tiene un título, un objetivo, un contexto, una lista de actores, una lista de recursos y episodios. Por último, las tarjetas CRC están definidas por un nombre, una serie de responsabilidades y colaboraciones (tabla 27).

Todos los atributos de los elementos son descripciones textuales. En los mismos aparecen referencias a otros elementos del Client Oriented Requirements Baseline. La construcción del LEL está regida por el principio de circularidad, por el cual, para describir una entrada de LEL, se deben emplear otras entradas del mismo LEL. En los escenarios no hay una regla explícita que dicte el uso de los términos definidos en el LEL, pero esto surge naturalmente. Como el LEL captura el lenguaje de un contexto y en los escenarios se describen situaciones del contexto, resulta natural utilizar el lenguaje del LEL. En los episodios de los escenarios surge la alternativa de mencionar otros escenarios, con la consiguiente referencia a estos escenarios. En último lugar, las tarjetas CRC tienen referencias a los tres elementos: entradas de LEL, escenarios y tarjetas CRC. Las referencias a las entradas de LEL aparecen en las responsabilidades de las tarjetas (que se obtuvieron de los impactos de la entrada de LEL que le dio origen a la tarjeta CRC). Y las referencias a los escenarios y las tarjetas CRC se pueden apreciar en las colaboraciones. Las colaboraciones son pares de referencias a escenarios y tarjetas CRC (que mencionan con que otra tarjeta CRC colabora y en el entorno de que escenario).

Tabla 27 – Elementos del Client Oriented Requirements Baseline con sus atributos.

Elementos
Entrada del LEL
Sinónimos
Noción
Impactos
Escenario
Título
Objetivo
Contexto
Actores
Recursos
Episodios
Tarjeta CRC
Nombre
Responsabilidades
Colaboraciones

Resumiendo, cada atributo de entrada de LEL, escenario o tarjeta CRC está compuesto por texto o referencias a otros elementos. Este modelo no es otra cosa más que un hipertexto, ya que se tienen nodos de información (compuesto por cada entrada de LEL, escenario o tarjeta CRC) los cuales son descriptos a través de texto, y los links del hipertexto, surgen de las referencias a los otros elementos.

Para la composición de cada atributo con que se describen al LEL, escenarios y tarjetas CRC es importante hacer la distinción entre texto plano y referencias a otros elementos. Esto es debido a que, por ejemplo, si una entrada de LEL es referenciada desde la lista de actores de un escenario, esto es motivo suficiente para que se defina una tarjeta CRC primaria. Por lo tanto la referencia a una entrada de LEL origina una tarjeta, mientras que si en la lista de actores se coloca uno que no está definido en el LEL, no genera nada. Las tarjetas secundarias se obtienen a través de referencias también. Y las colaboraciones se basan en referencias a entradas de LEL y escenarios.

Con respecto a estos atributos, las variaciones que pueden sufrir son: el agregado de información, su modificación o la eliminación. Estas tres instancias repercuten de distinta forma en el conjunto. Si se agrega información, por ejemplo actores a la lista de un escenario, esto puede originar nuevas tarjetas CRC. Si se modifica el título de un escenario, como el mismo puede estar referenciado desde otros elementos, se deben modificar las referencias. Si lo que se borra es un actor de un escenario, puede suceder que una tarjeta CRC desaparezca, o tal vez que se transforme de primaria a secundaria.

Sin embargo, es importante especificar la naturaleza de la acción. No es lo mismo agregar texto plano a la lista de actores que agregar una referencia a una entrada de LEL. Por lo tanto, para todas las tres acciones enumeradas se debe tener en cuenta que se pueden realizar con texto plano y con referencias. Además, la modificación en sí, puede ser de distintos tipos. Se puede modificar agregando información, eliminando o cambiando datos. Para la lista de términos que definen una entrada de LEL, las tres operaciones tienen implicancias diferentes. Agregar un nuevo sinónimo puede significar que nuevos actores de escenarios se conviertan en referencias a entradas de LEL, por lo cual, nuevas tarjetas CRC deben aparecer. La eliminación, provoca el efecto inverso. Mientras que la modificación de un término por otro provoca que se renombren todas las referencias a éste, quedando la misma cantidad de elementos. Por lo tanto, 10 son los cambios que se pueden presentar en los elementos del Client Oriented Requirements Baseline (tabla 28) y que se deben analizar.

Para determinar las reglas de forward traceability en el marco del Client Oriented Requirements Baseline se debe analizar para cada atributo de cada elemento (tabla 27) cada una de los posibles cambios (tabla 28). Frente a cada posible cambio se debe analizar si la modificación altera la consistencia del sistema LEL-escenarios-tarjetas CRC. Si la modificación produce alguna alteración es necesario escribir una regla que lo corrija.

Para evaluar cada modificación se debe analizar cada una de las 120 combinaciones resultantes de cruzar cada atributo de cada elemento con cada posible cambio. La tabla 29 muestra todas las combinaciones posibles. Sobre el eje vertical se encuentran los elementos, y sobre el horizontal los cambios posibles. Cada intersección está numerada con los valores desde el 1 hasta el 120. El fin de estos números es poder referenciar en forma directa a cada intersección en el análisis para inferir si en cada una se altera o no la consistencia.

Tabla 28 – Naturaleza de los cambios sobre el Client Oriented Requirements Baseline.

Cambios
Agregar información
Agregar texto
Agregar referencia
Modificar información
Agregar texto
Modificar texto
Eliminar texto
Agregar referencia
Modificar referencia
Eliminar referencia
Eliminar información
Eliminar texto
Eliminar referencia

El caso 1 se debe analizar. Ya que agregar un nuevo sinónimo a una entrada de LEL implica que un actor utilizado en un escenario se convierta en una referencia a una entrada de LEL, con lo cual una tarjeta CRC primaria debe ser derivada. También puede ocurrir que en las responsabilidades de una tarjeta primaria aparezcan nuevas referencias a entradas de LEL, esto motiva a la derivación de tarjetas secundarias.

El caso 2 no tiene sentido, ya que no tiene sentido agregar una referencia a una entrada de LEL como sinónimo.

El caso 3 es similar al caso 1, ya que en ambos casos se están agregando sinónimos.

Tabla 29 – Cambios posibles sobre los atributos de LEL, escenario y tarjetas CRC.

		Agregar		Modificar						Eliminar	
		Agregar		Agregar		Modificar		Eliminar		Eliminar	
		Texto	Referencia	Texto	Referencia	Texto	Referencia	Texto	Referencia	Texto	Referencia
Entrada del LEL	Sinónimos	1	2	3	4	5	6	7	8	9	10
	Noción	11	12	13	14	15	15	17	18	19	20
	Impactos	21	22	23	24	25	26	27	28	29	30
Escenario	Título	31	32	33	34	35	36	37	38	39	40
	Objetivo	41	42	43	44	45	46	47	48	49	50
	Contexto	51	52	53	54	55	56	57	58	59	60
	Actores	61	62	63	64	65	66	67	68	69	70
	Recursos	71	72	73	74	75	76	77	78	79	80
	Episodios	81	82	83	84	85	86	87	88	89	90
Tarjeta CRC	Nombre	91	92	93	94	95	96	97	98	99	100
	Responsabilidades	101	102	103	104	105	106	107	108	109	110
	Colaboraciones	111	112	113	114	115	116	117	118	119	120

El caso 4 tampoco tiene sentido ya que es similar al 2.

El caso 5 implica modificar el término que identifica a una entrada de LEL. Esto requiere que se modifiquen todas las referencias a la entrada de LEL modificada.

El caso 6 carece de sentido, al ser imposible agregar una referencia de ninguna forma se puede modificarla.

El caso 7 es inverso al 1 y 3. Al eliminar un sinónimo puede ocurrir que un actor deje de estar referenciando a una entrada de LEL, por lo cual una tarjeta CRC puede dejar de existir.

El caso 8 no tiene sentido, sin agregar referencias no se puede eliminarlas.

El caso 9 es similar al 7, en ambas situaciones se eliminan sinónimos.

El caso 10 es similar al 8, por lo tanto tampoco tiene sentido.

Los casos del 11 al 20 son irrelevantes. En la descripción de la mecánica de la derivación de tarjetas CRC se deja en claro que las nociones de las entradas de LEL no tienen ninguna trascendencia en la derivación (figura 20). Su único fin es la de describir los símbolos para comprender el lenguaje de la aplicación, pero no agregan ninguna semántica a la derivación.

Los casos 21 al 30 sí deben ser estudiados. Los impactos de las entradas de LEL se convierten en las responsabilidades de las tarjetas CRC, por lo tanto, cualquier modificación en los impactos, deben ser reflejados en las responsabilidades. Además, las referencias a entradas de LEL en las responsabilidades de las tarjetas CRC primarias originan tarjetas CRC secundarias. Así que cualquier cambio sobre ellas debe ser sincronizado con las tarjetas CRC existentes.

Los casos 31 y 32 deben ser analizados. Cuando se agrega texto o referencia a un título de un escenario se está definiendo un nuevo escenario (si se agrega, se está definiendo uno nuevo, si se modifica se está cambiando uno existente). Si en algún episodio de otro escenario se indica el mismo título, se lo debe convertir a una referencia del escenario agregado.

Los casos 33 al 38 son todos similares. El título de un escenario, que puede contener a referencias a entradas de LEL, puede estar referenciado desde otro escenario o desde alguna tarjeta CRC. La modificación implica alterar de igual forma las referencias.

Los casos 39 y 40 son simétricos de los casos 31 y 32. La eliminación de un título se debe a la eliminación del escenario en sí, por lo cual las referencias desde otro escenario al escenario eliminado se deben eliminar.

Los casos 41 al 60, pueden dejarse de lado. El objetivo y contexto de los escenarios no tienen incidencia en el resto del modelo.

El caso 61 no tiene ninguna repercusión. El agregar un nuevo actor que no es entrada de LEL no origina ningún cambio.

Sin embargo, el caso 62 se refiere a agregar una referencia a una entrada de LEL en la lista de actores. Esta es una situación similar a la del caso 1. Cuando un actor es una referencia a una entrada de LEL, pueden aparecer nuevas tarjetas CRC.

El caso 63 es similar al 61, y no posee ninguna implicancia.

El caso 64 es similar al 62 y también debe estudiarse

Los casos 65 y 66 no tienen sentido. El concepto de modificación de la lista de actores es diferente al concepto de modificación de los términos de la lista de sinónimos (caso 5). En el LEL, al modificar un término de la lista de sinónimos se está modificando su identificación, y todas las referencias al elemento modificado se deben actualizar. Sin embargo, el modificar una referencia a un actor (ya sea o no una referencia, es decir texto o referencia) implica eliminar un actor para agregar otro nuevo. De esta forma, los casos de modificar están desdoblados en eliminar y agregar.

Los casos 67 al 70 son inversos al 61-64. Por lo tanto, los casos 67 y 69, eliminar texto, no tienen ninguna repercusión. Por el contrario, los casos 68 y 70, eliminar referencias, pueden hacer desaparecer tarjetas primarias.

Los casos del 71 al 80, tampoco son de relevancia. Los recursos no son utilizados en el algoritmo de derivación.

Para los casos referentes a los episodios, sólo interesan los que agregan o eliminan referencias: 82, 84, 88 y 90, ya que las referencias a entradas de LEL y escenarios son las que determinan las colaboraciones en las tarjetas. El agregado de referencias lleva a nuevas colaboraciones y la eliminación de las mismas a que desaparezcan.

Los casos 81, 83, 85, 87 y 89 no son de importancia. El texto no tiene ninguna relevancia.

Con respecto al caso 86, es similar a los casos 65 y 66. No es posible modificar una referencia en un episodio. Lo que realmente ocurre es que se elimina una referencia y se agrega una nueva.

Los casos 91 al 99 son irrelevantes. El nombre de una tarjeta CRC se extrae de la entrada de LEL que lo origina, ningún tipo de cambio sobre el nombre de la tarjeta tiene sentido. Solamente se altera el nombre de la tarjeta CRC al modificarse la entrada de LEL que lo origina y este caso ya se tomó en cuenta (caso 5).

Con respecto a las responsabilidades, los casos 102, 104, 108 y 110, son los que deben evaluarse. Las referencias a entradas de LEL, desde las responsabilidades de las tarjetas CRC primarias, determinan la aparición de tarjetas CRC secundarias. De igual forma, su eliminación está relacionada con la desaparición.

Los casos 101, 103, 105, 107 y 109, se corresponden con la modificación de texto en las tarjetas y no son de importancia.

El caso 106 se refiere a la modificación de referencias. Es similar a los casos 66 y 86. No se modifican las referencias, lo que ocurre en realidad es la eliminación de una y el agregado de otra.

Por último, los casos del 111 al 120 no son de importancia. Las colaboraciones son ilustrativas y no tienen ninguna incidencia en el resto del conjunto.

Por lo tanto, son 37 las acciones que merecen ser estudiadas: 1, 3, 5, 7, 9, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 62, 64, 68, 70, 82, 84, 88, 90, 102, 104, 108 y 110 (tabla 30).

5.5. Reglas de traceability forward

Los 37 casos de modificaciones que ocasionan inconsistencias en el modelo, se analizaron en detalle y se escribieron reglas de traceability forward.

Para la escritura se utiliza el modelo de evolución de artefactos de Rolland [Rolland 1994]. Esta aproximación es orientada a la decisión, esto significa que los artefactos evolucionan como consecuencia de decisiones. Estas decisiones se toman dentro del contexto del dominio, el cual está fuertemente acoplado con la decisión. Situación, decisión y acción son tres conceptos básicos que utilizamos para describir las reglas.

La situación representa el contexto de la decisión, indica el ámbito en el cual se debe hacer evolucionar un artefacto. La decisión refleja la elección de la transformación del producto. Y la acción realiza la transformación del producto. Es una materialización de la decisión. Realizándola, el producto cambia y se pueden generar nuevas situaciones, las cuales podrían estar sujetas a nuevas decisiones.

Tabla 30 – Cambios que deben ser estudiados.

		Agregar		Modificar						Eliminar	
		Agregar		Agregar		Modificar		Eliminar		Eliminar	
		Texto	Referencia	Texto	Referencia	Texto	Referencia	Texto	Referencia	Texto	Referencia
Entrada del LEL	Sinónimos	1		3		5		7		9	
	Noción										
	Impactos	21	22	23	24	25	26	27	28	29	30
Escenario	Título	31	32	33	34	35	36	37	38	39	40
	Objetivo										
	Contexto										
	Actores		62		64				68		70
	Recursos										
Tarjeta CRC	Episodios		82		84				88		90
	Nombre										
	Responsabilidades		102		104				108		110
	Colaboraciones										

Regla I.

Situación: Se ha encontrado una nueva palabra que es sinónimo de una entrada de LEL existente.

Decisión: El agregado del sinónimo a la entrada de LEL.

Acción: Se deben definir las nuevas referencias desde las entradas de LEL, escenarios y tarjetas CRC que utilicen la nueva palabra hacia la entrada de LEL modificada.

Regla II.

Situación: Un sinónimo de una entrada del LEL se escribió en forma incorrecta.

Decisión: La reescritura del sinónimo de la entrada del LEL.

Acción: Se deben cambiar todas las referencias del sinónimo incorrecto escrito en otras entradas del LEL, escenarios o tarjetas CRC.

Regla III.i

Situación: Se han definido en una entrada del LEL dos sinónimos, cuando en realidad son dos entradas independientes.

Decisión: La eliminación de uno de los sinónimos de la entrada de LEL.

Acción: Se deben deshacer las referencias desde otras entradas de LEL, escenarios o tarjetas CRC al sinónimo borrado.

Regla III.ii

Situación: Se han definido en una entrada del LEL dos sinónimos distintos, cuando en realidad se debe usar uno sólo.

Decisión: La eliminación de uno de los sinónimos de la entrada de LEL.

Acción: Se debe reemplazar el sinónimo eliminado por otro sinónimo de la entrada de LEL modificada en todas las entradas del LEL, escenarios y tarjetas CRC que referencian al sinónimo eliminado.

Regla IV.

Situación: Los impactos de una entrada del LEL no son los adecuados.

Decisión: La modificación de los impactos de la entrada del LEL.

Acción: Las responsabilidades de la tarjeta CRC asociada (aquella obtenida a partir de la entrada del LEL) deben sufrir las mismas modificaciones.

Regla V.

Situación: Hace falta definir un nuevo escenario para describir el dominio.

Decisión: La definición de un nuevo escenario, con un título que lo identifique.

Acción: Los episodios que concuerdan con el título del escenario se deben transformar en referencias a este último.

Regla VI.

Situación: El título de un escenario no es el correcto.

Decisión: La modificación del título del escenario.

Acción: Se debe modificar el texto en todos los escenarios y tarjetas CRC que poseen referencias a este escenario.

Regla VII.

Situación: Un escenario deja de ser relevante en la descripción del dominio.

Decisión: La eliminación del escenario.

Acción: Todas las referencias al escenario en cuestión, desde los episodios de otros escenarios se deben eliminar.

Regla VIII.i

Situación: En un escenario se ha omitido un actor. Este actor tiene su correspondiente definición en el LEL. Pero esta entrada de LEL no tiene la tarjeta CRC asociada.

Decisión: El agregado del actor en el escenario.

Acción: Se debe derivar la tarjeta CRC asociada a la entrada del LEL correspondiente al actor.

Regla VIII.ii

Situación: En un escenario se ha omitido un actor. Este actor tiene su correspondiente definición en el LEL. Y esta entrada de LEL tiene la tarjeta CRC asociada.

Decisión: El agregado del actor en el escenario.

Acción: Se deben actualizar las colaboraciones de la tarjeta CRC que se lleven a cabo en el escenario.

Regla IX.i

Situación: Un escenario presenta un actor que no debe serlo. Este actor incorrecto, no lo es de ningún otro escenario, ni tampoco es referenciado en las responsabilidades de una tarjeta CRC primaria.

Decisión: La eliminación del actor del escenario.

Acción: Se debe eliminar la tarjeta CRC asociada al actor.

Regla IX.ii

Situación: Un escenario presenta un actor que no debe serlo. Este actor incorrecto, no lo es de ningún otro escenario, pero es referenciado en las responsabilidades de una tarjeta CRC primaria.

Decisión: La eliminación del actor del escenario.

Acción: Se debe transformar la tarjeta CRC asociada al actor de primaria a secundaria.

Regla X.

Situación: En los episodios de un escenario se omitió referenciar a un término que está definido en el LEL. Este término tiene una tarjeta de CRC asociada.

Decisión: El agregado de la referencia a la entrada del LEL en los episodios del escenario.

Acción: Se deben actualizar las colaboraciones para todas las tarjetas CRC cuyas entradas del LEL asociadas se referencian en el escenario.

Regla XI.

Situación: Los episodios de un escenario presentan una referencia por única vez (no hay ninguna otra referencia en los episodios) a un término que está definido en el LEL y que tiene su correspondiente tarjeta CRC asociada. Sin embargo, la referencia es incorrecta.

Decisión: La eliminación de la referencia a la entrada del LEL de los episodios del escenario.

Acción: Se deben actualizar las colaboraciones para todas las tarjetas CRC cuyas entradas del LEL asociadas se referencian en el escenario.

Regla XII.

Situación: Se debe incorporar por primera vez, una referencia a una entrada del LEL (que no tienen tarjetas CRC asociada) en las responsabilidades de una tarjeta CRC primaria.

Decisión: La incorporación de la referencia a la entrada del LEL.

Acción: Se debe crear la tarjeta CRC secundaria para la entrada del LEL.

Regla XIII.

Situación: Se debe eliminar de las responsabilidades de una tarjeta CRC, la última referencia a una entrada del LEL. Esta última tiene una tarjeta CRC secundaria asociada.

Decisión: La eliminación de la referencia a la entrada del LEL.

Acción: Se debe eliminar la tarjeta CRC secundaria.

Regla XIV.

Situación: Una entrada de LEL que no tiene tarjeta CRC asociada, es suficientemente importante para definir una tarjeta CRC.

Decisión: La creación de la tarjeta CRC

Acción: Se deben actualizar todas las colaboraciones con la nueva tarjeta CRC creada.

Regla XV.

Situación: Una tarjeta CRC definida, no es lo suficientemente importante para que esté definida.

Decisión: La eliminación de una tarjeta CRC

Acción: Se deben eliminar todas las colaboraciones en donde la tarjeta CRC eliminada sea referenciada.

Las 15 reglas enunciadas resuelven los 37 casos encontrados. Las reglas I, III, IV, V, VI, VII, VIII, IX, X, XI, XII y XIII son aplicables a más de un caso. Las reglas III, VIII y IX se necesitan desdoblar en dos reglas puesto que la situación que se puede presentar es distinta para un mismo caso y se necesitan acciones distintas. Luego, se incluyen dos reglas, XIV y XV, que no tienen su contraparte en los 120 casos estudiados, pero que surgen de reglas previas. La tabla 31 muestra la relación entre los casos de análisis y las reglas asociadas.

En general se puede observar que hay una simetría en las reglas que solucionan los casos, siendo la columna modificar-modificar el eje de simetría. El motivo es que agregar es una operación por sí sola, pero dentro de modificar también aparece. Y ya sea que se agrega texto a uno existente o a nada, la regla que se debe aplicar es la misma.

El otro hecho curioso es la regla IV que se aplica a cualquier modificación que sufran los impactos. El punto es que cualquiera sea el carácter de los cambios, al alterar los impactos de una entrada de LEL se deben llevar los mismos a las responsabilidades de la tarjeta CRC asociada.

Para las reglas III, VIII y IX se presentan dos variantes de cada una. Esto se debe a que frente a una modificación se pueden producir dos situaciones distintas, que se deben tratar con reglas distintas.

Las reglas XIV y XV son dos reglas que no se aplican a ninguno de los 120 casos estudiados. Estas reglas se surgen como necesidad de la aplicación de las reglas previas. Al agregar o eliminar

una entrada de LEL o escenario no tiene implicancia por sí mismas, sino por el agregado o eliminación de algún atributo en particular. Mientras que con las tarjetas esto no ocurre. El agregado o eliminación de tarjetas tiene su impacto directo en las colaboraciones, es por ello que es importante el agregado de las reglas XIV y XV.

Tabla 31 – Reglas que se aplican en cada caso de análisis.

		Agregar		Modificar				Eliminar			
		Agregar		Modificar		Eliminar		Eliminar			
		Texto	Referencia	Texto	Referencia	Texto	Referencia	Texto	Referencia		
Entrada del LEL	Sinónimos	I		I		II		III		III	
	Noción										
	Impactos	IV	IV	IV	IV	IV	IV	IV	IV	IV	IV
Escenario	Título	V	V	VI	VI	VI	VI	VI	VI	VII	VII
	Objetivo										
	Contexto										
	Actores		VIII		VIII				IX		IX
	Recursos										
Tarjeta CRC	Episodios		X		X				XI		XI
	Nombre										
	Responsabilidades		XII		XII				XIII		XIII
	Colaboraciones										

5.6. Comparación del modelo de traceability con otros modelos

El esquema de traceability propuesto se puede explicar a través de la propuesta de Rolland [Rolland 1994]. Las entradas del LEL, los escenarios y las tarjetas CRC cuadran dentro de la categoría de **evolutionary objects**. Sobre estos objetos se proponen tres categorías de evolución: **transformación, expansión y mutación**. Estas tres categorías están presentes en nuestro modelo. La modificación de una entrada del LEL o un escenario, puede verse como una evolución de transformación. La evolución del tipo de expansión se presenta con la creación y eliminación de las tarjetas CRC, las cuales forman el **spatial environment** del LEL y los escenarios. Con respecto a la evolución de mutación, las tarjetas de CRC pueden cambiar entre el tipo primario al tipo secundario y por otra parte, las entradas del LEL se pueden mapear en tarjetas de CRC.

Al igual que en [Breitman 2000] nuestro objetivo es hacer traceability desde el punto de vista del desarrollo y no del mantenimiento. No se pretende capturar las distintas configuraciones del LEL, escenarios y tarjetas CRC para dejar en forma explícita los cambios producidos (con su justificación, el responsable, fecha y hora), por el contrario, se hace trace entre las distintas etapas del desarrollo para mantener consistente la información de LEL, escenarios y tarjetas CRC. Aspecto que es necesario ya que estos elementos evolucionan. La evolución que se aprecia, es la del tipo intra escenario según la clasificación de [Breitman 1998], sin embargo de las operaciones que cuadran bajo esta categoría, sólo observamos las de **input, modification y exclusion**.

En [Lavazza 2000] se presentan dos alternativas, modelar traceability de acuerdo al producto y al proceso. En este trabajo están presentes las dos visiones. Traceability de acuerdo al producto está presente en las referencias que existen en la descripción de los atributos que conforman las entradas de LEL, escenarios y tarjetas CRC. Cada atributo se define en forma textual y en la definición aparecen referencias a otros elementos, esto es lo que en [Lavazza 2000] llaman traceability basado en producto. Con respecto a los procesos, el único que aparece en este trabajo es el de derivación de tarjetas CRC a partir del LEL y escenarios.

Si bien el modelo presentado no es un método de diseño, de todas formas se puede explicar en función del trabajo de [Potts 1989]. Los **artifacts** de este modelo son las entradas de LEL, los escenarios y las tarjetas CRC. Los **steps** son las modificaciones que los usuarios realizan sobre las

entradas de LEL y los escenarios. El **issue** es el carácter de la edición realizada, es decir, la modificación de textos o referencias de los distintos atributos que conforman los artifacts. **Position** es la regla que se debe aplicar dada la situación del issue. Y por último, los **arguments**, están dados por las reglas de derivación [Leonardi 2001].

En [Pinheiro 1996] se define un modelo a partir de la herramienta *TOOR*. Se definen artefactos y los links entre ellos. Para los artefactos se definen clases (*templates*) para luego instanciar casos concretos. Cada clase está definida por atributos. En nuestro caso se tendrían tres clases: LEL, escenarios y tarjetas CRC. Con respecto a las relaciones, en nuestro trabajo se presenta la relación de derivación que es el que permite obtener tarjetas CRC a partir de LEL y escenarios. Luego, está la relación *parte de*, que es la que está presente cuando se utiliza un símbolo del LEL para definir otro elemento.

6. Implementación de las reglas de traceability forward

Es necesario implementar las reglas de forward traceability en una herramienta puesto que su aplicación manual es imposible. Se eligió implementarlas en el BMW puesto que la herramienta administra LEL, escenarios y tarjetas CRC, y se demostró la utilidad de la misma. Para implementar las reglas es necesario realizar un modelo ejecutable de las mismas: se utiliza un diagrama de actividades, el cual se traduce a un diagrama de clases para su implementación.

La sección 6.1 describe la necesidad de implementar las reglas. En la sección 6.2 se describe el modelo ejecutable. En la sección 6.3 se presenta el modelo ejecutable. En la sección 6.4 se detalla un diseño orientado a objetos a través de un diagrama de clases y diagramas de secuencia. En la sección 6.5 se describe la nueva arquitectura de BMW. Y en la sección 6.6 se muestran ejemplos de cómo se comporta la aplicación.

6.1. Necesidad de implementar las reglas

Es importante contar con una herramienta automatizada para que frente al cambio de un elemento analice los objetos y los links entre ellos. El volumen de información es muy grande y la tarea manual se torna tediosa y propensa a errores. En el anexo 9.1 se adjunta un modelo de LEL y escenarios relativamente pequeño. Este ejemplo cuenta con 36 entradas de LEL que poseen 348 referencias y 18 escenarios que poseen 155 referencias. Estos números dan una idea de la cantidad de elementos que se deben analizar.

A partir de la importancia de contar con una herramienta automatizada para asistir a los procesos del ciclo de vida del software y sumado a la existencia de una aplicación llamada BMW, la cual le es adecuada para el ingeniero de requerimientos que utiliza el modelo del Client Oriented Requirements Baseline para realizar un modelo contextual del dominio, surge la necesidad de plasmar las reglas obtenidas de forward traceability en la aplicación.

El proceso de implementación de las reglas implica llevar las mismas a un modelo ejecutable. Las reglas están redactadas siguiendo el modelo propuesto en [Rolland 1994]. Para cada regla se plantea una situación, la cual es un medio para indicar el contexto en el cual se debe aplicar la regla. Luego se define una decisión, esto es la modificación que finalmente se realiza en el conjunto formado por LEL, escenarios y tarjetas CRC. Esta decisión, es la que requiere de la aplicación de acciones correctivas para mantener la consistencia del Client Oriented Requirements Baseline. Las acciones correctivas son las que en el modelo de [Rolland 1994] se denominan acciones.

Por lo tanto, el modelo ejecutable debe prescribir las acciones a realizar, en función de la decisión.

6.2. Descripción del modelo ejecutable

Para describir el modelo ejecutable se utiliza un diagrama perteneciente a UML. Se eligió UML puesto que es el estándar más ampliamente utilizado, perteneciente a OMG (object management group) y unifica los tres métodos más difundidos: Booch, Rumbaugh y Jacobson.

Se utiliza el diagrama de actividades para describir el modelo ejecutable de las reglas. En él cada actividad se describe por medio de una burbuja rectangular de bordes redondeados. Las actividades se relacionan por medio de flechas. Una flecha desde una actividad A hacia otra actividad B, significa que la actividad A precede a la actividad B. De una actividad, pueden salir más de una flecha. Cuando esto sucede, junto a las líneas se debe colocar una condición. Las condiciones deben ser disjuntas entre sí, sólo una condición debe verificarse por vez. Las condiciones se indican entre corchetes. Una situación de decisión también se puede indicar a través de un rombo. A él le llega una sola flecha, pero parten más de una. Las barras de sincronización es otro elemento presente en la semántica de los diagramas de actividades. Para indicar que varias actividades se realizan en paralelo, se utilizan barras perpendiculares a las flechas. Para indicar que comienzan actividades en paralelo se coloca una barra en el extremo final de una flecha. A partir de esta barra pueden salir dos o más flechas. La misma barra se utiliza para indicar sincronización de actividades. Esto se denota porque a una barra llegan varias flechas y de ella sale una sola (figura 23).

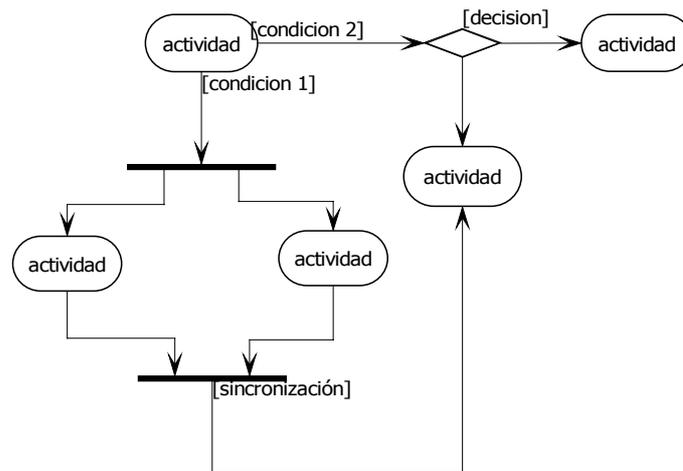


Figura 23 – Notación utilizada en el diagrama de actividades.

6.3. Modelo ejecutable

Las reglas propuestas están formadas por tres elementos: situación, decisión y acción. La situación describe un suceso durante la construcción de LEL y escenarios que lleva a la modificación de ellos. La decisión es la modificación del alguno de los elementos del Client Oriented Requirements Baseline. Y la acción son las tareas correctivas para asegurar la consistencia.

En la descripción del modelo ejecutable sólo interesa la decisión y la acción. La situación es irrelevante, puesto que BMW sólo se limita a identificar la decisión (la modificación de algún elemento) y realizar las acciones de la regla correspondiente. Si la decisión no le permite identificar la regla, se preguntará al usuario para que él (que conoce la situación) determine la acción a seguir. La regla III es un ejemplo en donde aunque decisión es la misma, la situación y la acción son diferentes.

Las acciones pueden corresponderse con decisiones de otras reglas, por lo cual aplicar una regla puede desencadenar la aplicación de otra. O bien, las acciones desencadenan cambios y son estos cambios los que eventualmente se podrían corresponder con reglas. De esta forma las reglas poseen un carácter cíclico en el cual una regla desencadena otras reglas o desencadena funciones. Y una función puede desencadenar reglas.

En el diagrama de actividades (figura 24), en general, la decisión de cada regla se corresponde con una actividad. Las mismas están organizadas en cuatro columnas. Las tres primeras columnas (LEL, escenarios y CRCs) describen las modificaciones que se pueden realizar sobre estos elementos. Las mismas responden al análisis presentado en la tabla 28. Las flechas relacionan las distintas actividades.

En el diagrama aparecen algunos elementos que no estaban presentes en las reglas. En las columnas de LEL y escenarios se muestran actividades de nombre agregar y eliminar, tanto para LEL como para escenarios. Estas actividades disparan otras actividades. Por ejemplo, agregar un escenario implica que se agregue un título, un actor y episodios.

Por otro lado, hay actividades que se eliminan del diagrama por no ser significativas, esto es con el fin de facilitar la lectura. Las actividades que se muestran en la figura 24 son aquellas que desencadenan la aplicación de las reglas, es decir son aquellas que en la tabla 31 tienen una regla asociada. Si la actividad carece de regla asociada, carece de importancia para la implementación, ya que la actividad no tiene relacionada ninguna otra actividad o función correctiva. Por ejemplo editar nociones, en la figura 24 en la columna de LEL, no es relevante, sin embargo se puso a modo de ejemplo. A ella le llegan un par de flechas, pero de ella no sale ninguna. Debido a la falta de importancia, sólo esta actividad se indica en el diagrama, el resto no indican para facilitar la lectura.

En el diagrama cada regla está representada por una actividad. La misma posee el número de la regla correspondiente. En algunos casos, como la regla III que tiene una subclasificación en III.i y III.ii, cada una se describe como actividad independiente. Si bien ambas surgen como una misma regla III, la decisión es la que varía. La situación es común a ambas: el borrado de un sinónimo. Sin embargo, la motivación del borrado es diferente, la actividad en sí es distinta, se borra pero por motivos diferentes. En cambio, la regla VIII, también está subclasificada pero se indica una actividad

sola. En este caso, la actividad es única, condicionada por el estado del Client Oriented Requirements Baseline. Este condicionamiento dicta que justamente se indiquen las alternativas como condiciones.

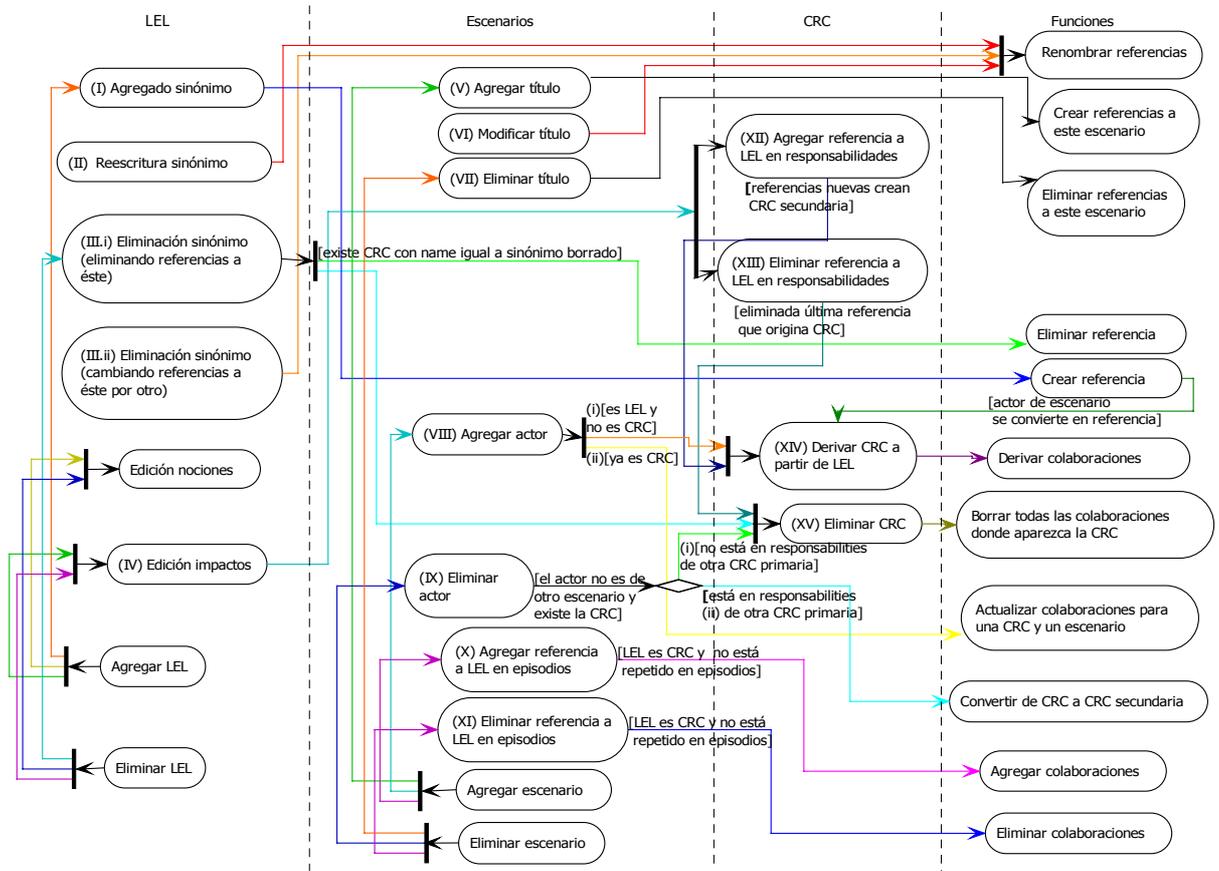


Figura 24 – Diagrama de actividades de las reglas de traceability forward.

6.4. Diseño de objetos

BMW está implementado en Smalltalk. Para incorporar las reglas a la aplicación se realizó un diagrama de clases antes de implementar (figura 25).

Las actividades que representan decisiones de reglas o las funciones están modeladas cada una como una clase de la jerarquía encabezada por la clase abstracta TraceabilityForwardGuard. Cada objeto conoce las acciones que debe realizar para mantener la consistencia y cuáles son las actividades que se deben realizar después. Los dos elementos se combinan en el **mensaje correrEn:para:**. En la implementación del mensaje se utiliza el patrón método template [Gamma 1995]. En el diseño de la jerarquía también se aplicó el patrón strategy [Gamma 1995], puesto que cada clase conoce la estrategia llevar a cabo las reglas de traceability.

El punto de partida para la aplicación de las reglas de traceability lo da el objeto TraceabilityForwardController. Se aplicó el patrón proxy [Gamma 1995] y se coloca a este objeto entre la interface y el CORBManager. CORBManager es el responsable de agregar, modificar y eliminar los elementos. Sin embargo, TraceabilityForwardController intercepta las modificaciones de los elementos del Client Oriented Requirements Baseline y determina si hay que aplicar reglas o no. Esto lo hace con ayuda de las subclases de IncorporadorDeReglas. Estos objetos son un wrapper [Gamma 1995] de los elementos del Baseline. Conocen las reglas a aplicar frente a cambios que puedan sufrir y reenvían los mensajes propios de los elementos a ellos.

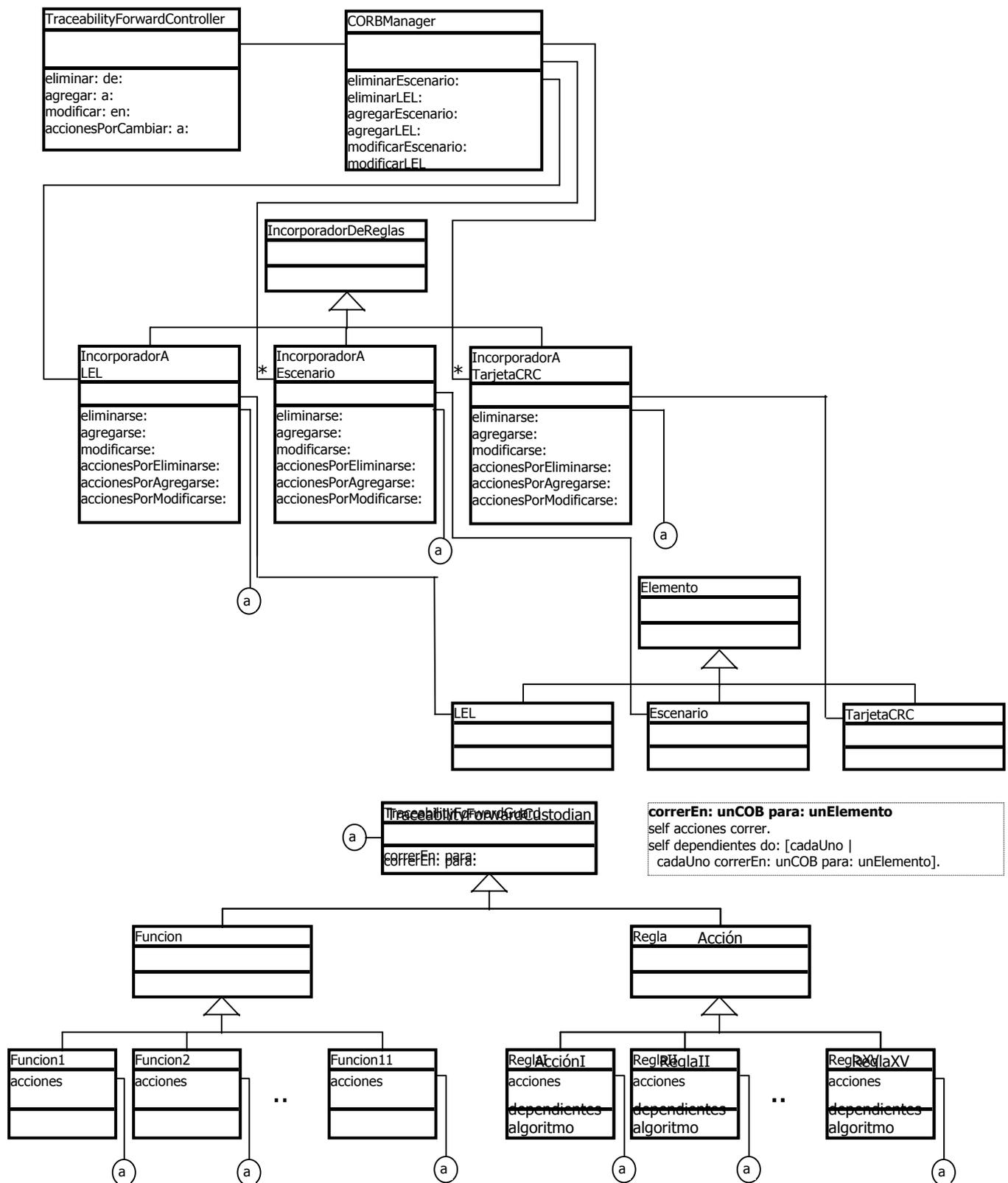


Figura 25 – Diagrama de clases de la implementación de Traceability en BMW.

Para ilustrar la ejecución se muestra en la figura 26 el diagrama de secuencia de la operación eliminar un escenario.

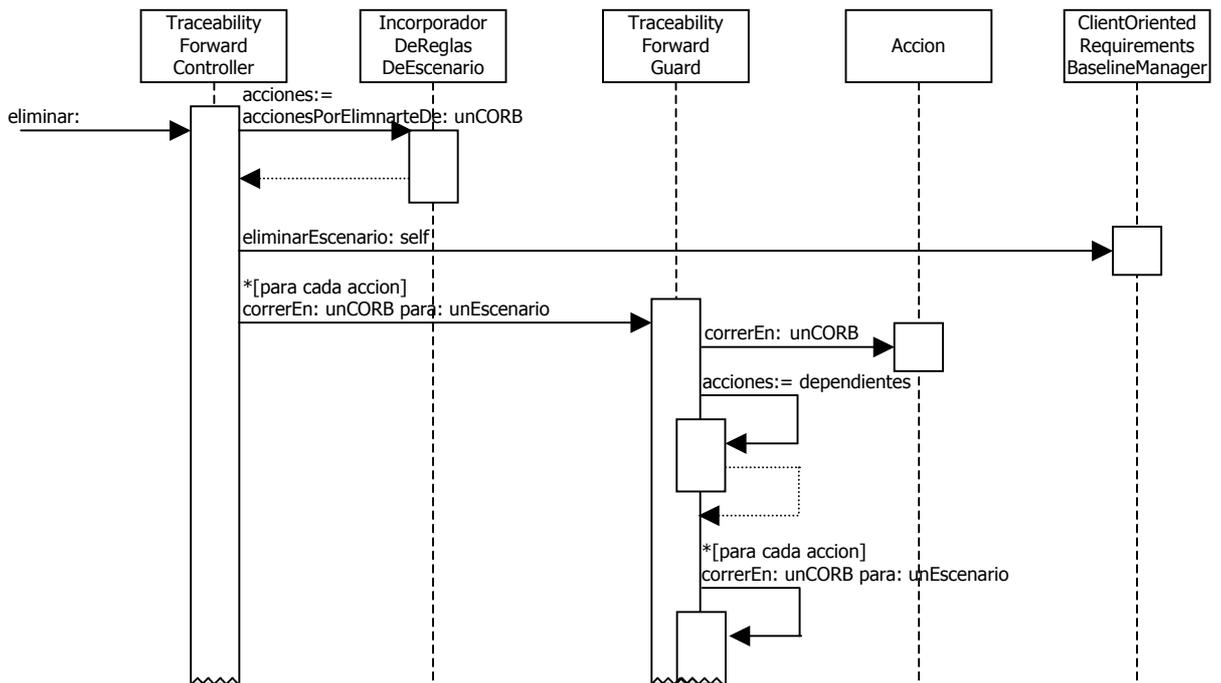


Figura 26 – Diagrama de secuencias de eliminar un escenario.

6.5. Nueva arquitectura de BMW

La arquitectura original del BMW (figura 8) se modificó agregando el módulo de forward traceability [Antonelli 2000] [Antonelli 2001] a los cuatro módulos iniciales: editor de entradas de LEL, editor de escenarios, generador de tarjetas CRC y browser de navegación (figura 27).

El módulo de forward traceability se encarga de implementar las reglas y asegurar que la información del Client Oriented Requirements Baseline se mantenga consistente. Este módulo es el encargado de determinar la naturaleza del cambio frente a cada edición de un elemento. Una vez identificada la naturaleza del cambio es el responsable de tomar la regla adecuada para la situación presentada y llevar a cabo la misma.

Además, este módulo se encarga de otras funciones como ser la detección y marcado de referencias a otros elementos. Durante la edición, mientras se escribe texto, este módulo analiza el mismo y cuando concuerda con algún sinónimo de las entradas de LEL o con algún escenario, la referencia se realiza en ese momento.

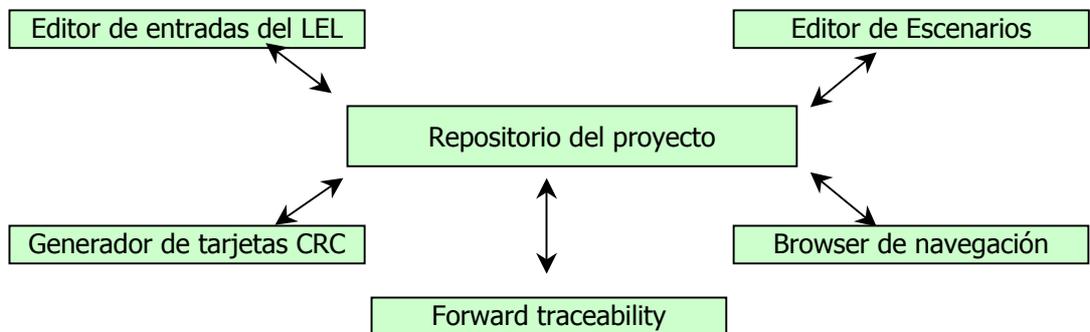


Figura 27 – Nueva arquitectura del Baseline Mentor Workbench.

6.6. Ejemplos

Para ilustrar el funcionamiento del módulo de forward traceability en esta sección se muestran algunos ejemplos. Los mismos provienen del dominio ya tratado y se muestra el comportamiento de la herramienta a través de copias de pantallas.

Tómese como primer ejemplo la regla II. Esta regla determina que frente a la modificación de un sinónimo se deben modificar todas las referencias a éste. Sea una entrada de LEL denominada **"intimación negativa"** (figura 28). Una posible situación es que durante algún proceso de verificación y validación se concluye que el nombre debería ser **"desconocido"**.

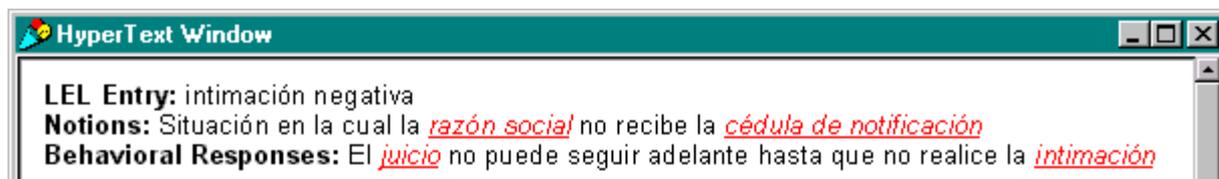


Figura 28 – Entrada de LEL "intimación negativa".

El escenario de la **"intimación de una razón social"** referencia a la entrada de LEL **"intimación negativa"** en la última línea de los episodios (figura 29).



Figura 29 – Escenario "intimación de una razón social" referenciando a la entrada de LEL "intimación negativa" en la última línea de los episodios.

Debido a que **"intimación negativa"**, no es el nombre adecuado para la entrada de LEL, se lo modifica por **"desconocido"** y se graban los cambios. Luego de esta operación, BMW determina que ha habido cambios, sin embargo determina solamente que un sinónimo se ha eliminado (**"intimación negativa"**) y que otro se ha agregado (**"desconocido"**). La herramienta no puede determinar si se modificó solamente el sinónimo que identifica a la entrada de LEL o si se está creando una nueva entrada. Los dos casos que se plantean son: la entrada de LEL sigue siendo la misma, sólo que se modificó su identificador. O al cambiar el nombre es otra entrada de LEL distinta. Para el primer caso, situación de la regla II, se deben mantener las referencias y renombrarlas. Por el contrario, el segundo caso corresponde a una combinación de las reglas I y III.i. La regla I plantea la situación del agregado de un nuevo sinónimo y la III.i de la eliminación en forma definitiva de uno. Debido a que la herramienta no tiene forma de determinar cual es la situación real a juzgar por los cambios, lo debe consultar al usuario a través de una ventana de diálogo (figura 30).

Para la situación planteada se debe responder **"yes"** en la ventana de diálogo. Esto indica que se ha modificado el sinónimo y se deben actualizar las referencias, indicándole a BMW que aplique la regla II. Si se hubiera contestado **"no"** haría una combinación de las reglas I y III.i.

Como resultado de la aplicación de la regla II, en el escenario **"intimación de una razón social"** se reemplaza el texto **"intimación negativa"** por **"desconocido"** (figura 31).

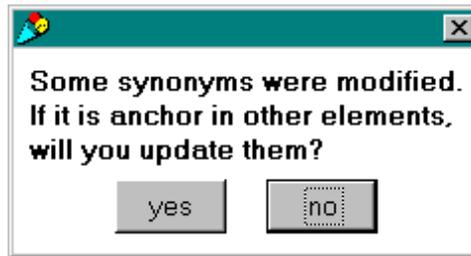


Figura 30 – Pedido de confirmación para actualizar las referencias.



Figura 31 – Modificación del texto "intimación negativa" por "desconocido".

Sea un nuevo ejemplo en donde se define una entrada de LEL con tres sinónimos: "**particulares / razón social / demandado**" (figura 32). Pero se cometió un error puesto que "**demandado**" no es sinónimo de los otros dos. El término "**demandado**" existe y debe estar definido en el LEL, pero no como sinónimo de "**particulares / razón social**". Por otra parte, el término "**demandado**" es referenciado desde la entrada de LEL "**abogado**" (figura 33).

Al editar la entrada de LEL y borrar el sinónimo "**demandado**", BMW advierte la diferencia pero no puede decidir si se la modificación corresponde a las reglas III.i o III.ii. Para la regla III.i la situación es que el sinónimo eliminado origina otra entrada de LEL, por lo cual referencias la entrada del sinónimo borrado se deben eliminar. La regla III.ii supone por el contrario que la referencia se debe mantener. BMW muestra nuevamente una ventana de diálogo (figura 31) para decidir que regla aplicar. Sin embargo, para la situación descrita se contesta "**no**", con la consiguiente ejecución de la regla III.i.

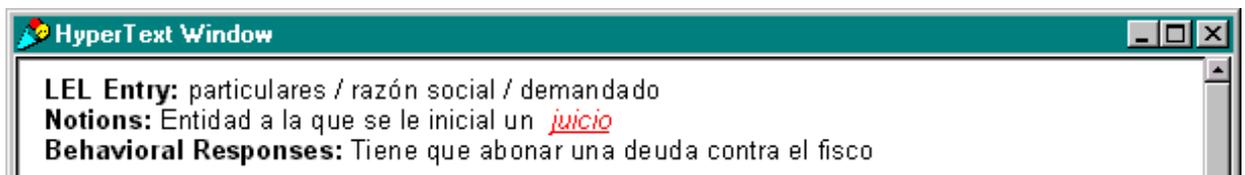


Figura 32 – Entrada de LEL que posee términos que no son sinónimos.

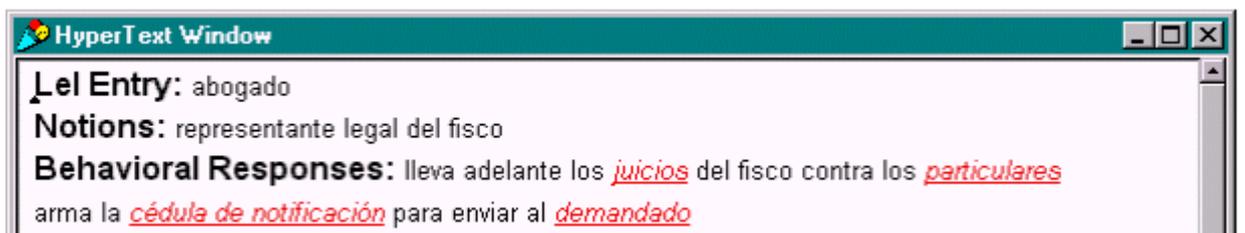


Figura 33 – Referencia a la entrada de LEL "demandado".

De esta forma, la entrada de LEL abogado, sigue manteniendo el texto demandado en los impactos, pero deja de ser una referencia (figura 34).

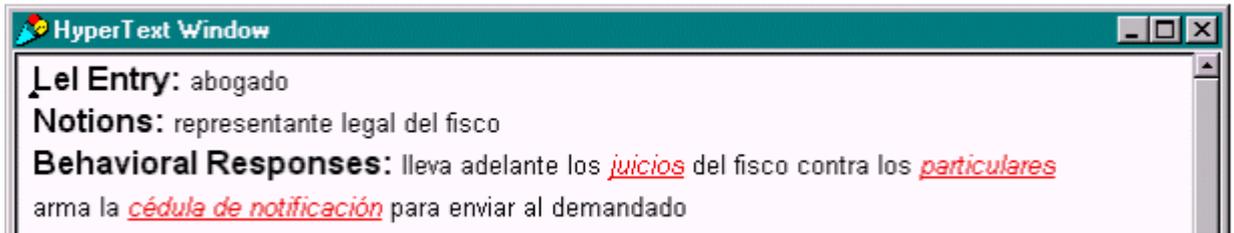


Figura 34 – Eliminación de la referencia a "demandado".

Sea un nuevo ejemplo en la cual se deben modificar los impactos de una entrada de LEL. Para la entrada "abogado" se determina que sus impactos están incompletos (figura 35). La entrada de LEL "abogado" cuenta con una tarjeta CRC derivada (figura 36).

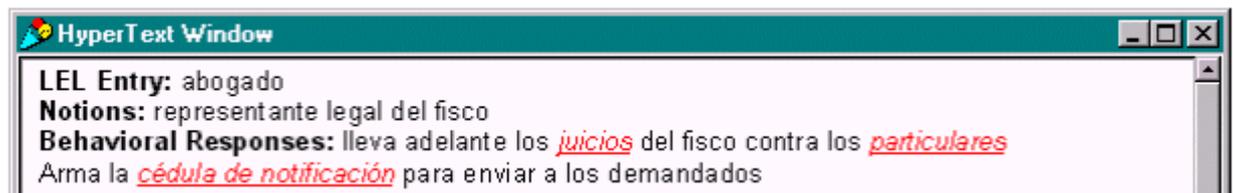


Figura 35 – Entrada de LEL "abogado" con sus impactos incompletos.

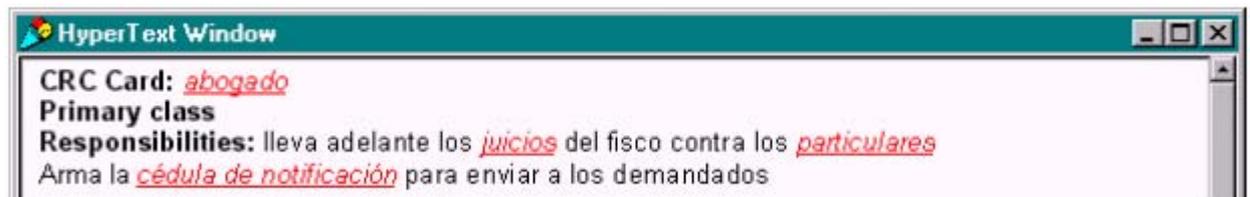


Figura 36 – Tarjeta CRC correspondiente a la entrada de LEL "abogado".

A la entrada de LEL "abogado" se le debe agregar en los impactos, que el abogado libra los oficios de intimación e intimación negativa (figura 37). Al modificarse los impactos de la entrada de LEL, la herramienta detecta que está frente a la regla IV. Sin embargo, la regla IV desencadena la aplicación de otras dos reglas: XII y XIII. La regla XII determina que el agregado de referencias a entradas de LEL crean nuevas tarjetas CRC. Por el contrario, la regla XIII determina que la eliminación de una última referencia a una entrada de LEL elimina la tarjeta CRC correspondiente. Claramente, BMW puede hacer la distinción entre las dos reglas y determinar que está en presencia de la número XII. Por lo tanto, modifica la tarjeta CRC para reflejar los cambios en los impactos de la entrada de LEL (figura 38) y define dos tarjetas CRC secundarias para cumplir con la regla XII (figuras 39 y 40).

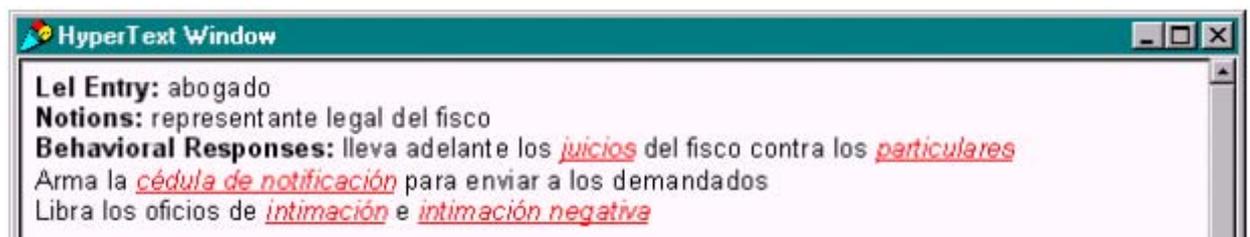


Figura 37 – Agregado de una línea en los impactos.

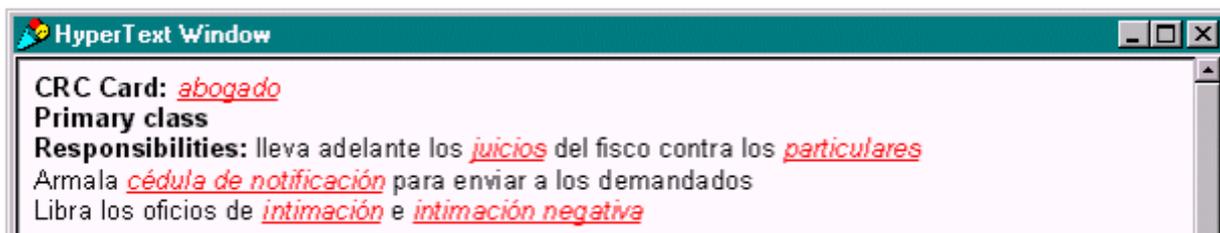


Figura 38 – Tarjeta CRC con las responsabilidades modificadas.

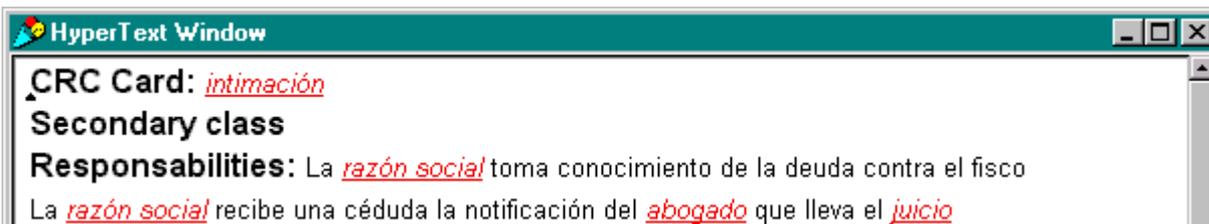


Figura 39 – CRC creada a partir de la modificación en CRC "abogado".

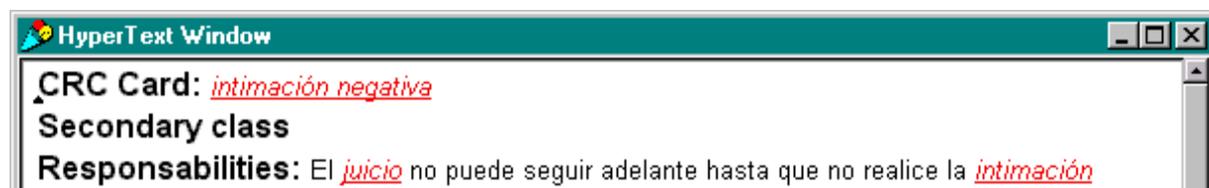


Figura 40 – Segunda CRC creada a partir de la modificación en CRC "abogado".

7. Futuros trabajos

El próximo paso es extender el modelo con la capacidad de traceability backward hacia las fuentes. Se pretende poder identificar a la persona o al manual de donde se extrajo la información para escribir una entrada de LEL o un escenario.

Las fuentes pueden ser muy variadas, como por ejemplo una sesión de brainstorming. De modo que vamos a extender el estudio de las fuentes de requerimientos, para proponer un modelo que las represente.

Una vez determinado el modelo debemos establecer el mecanismo de traceability. Cuando la fuente es una persona, es sencillo identificarla. Sin embargo, si una fuente es un libro habría que identificar además del libro, capítulo, hoja, párrafo y línea, por ejemplo. En el caso de una sesión de brainstorming que ha sido filmada habría que indicar a cuantos minutos desde el comienzo de la información o en que cuadros se encuentran las fuentes.

8. Referencias

- [Ackoff 1974] Ackoff, R., *Redesigning The Future*, Wiley, 1974.
- [Alspaugh 1999] Alspaugh, T.A., Antón, A.I., Barnes, T., Mott, B., "An Integrated Scenario Management Strategy", Fourth International Symposium on Requirements Engineering, RE'99, Limerick, Ireland, Junio, 1999, pp 142-149.
- [ANSI/IEEE 1984] ANSI/IEEE Standard 830-1984: Standard for Software Requirements Specifications, The Institute of Electrical and Electronic Engineers, New York, 1984.
- [Antonelli 1999] Antonelli, L., Oliveros, A., Rossi, G., "Baseline Mentor, An Application that Derives CRC Cards from Lexicon and Scenario", XXVIII JAIIO, II Workshop Iberoamericano en Ingeniería de Requerimientos, WER'99, Buenos Aires, Argentina, Septiembre 9 y 10, 1999.
- [Antonelli 2000] Antonelli, L., Oliveros, A., "Traceability en la Etapa de Elicitación de Requerimientos", Workshop de Investigadores en Ciencias de la Computación, WICC'2000, La Plata, Argentina, Mayo 22 y 23, 2000.
- [Antonelli 2001] Antonelli, L., Oliveros, A., "Traceability en la etapa de elicitación de requerimientos", IV Workshop on Requirements Engineering, WER'2001, Buenos Aires, Argentina, 22 y 23 de noviembre, 2001, pp 1-19.
- [Antonelli 2002] Antonelli, L., Oliveros, A., "Fuentes utilizadas por desarrolladores de software en Argentina para elicitar requerimientos", V Workshop on Requirements Engineering, WER'2002, ISBN 84-96023-01-X, Valencia, España, 11 y 12 de noviembre, 2002, pp 1-19.
- [Bellin 1997] Bellin, D., Suchman Simone, S., *The CRC card book*, Addison-Wesley, 1997.
- [Boehm 2001] Boehm, B., Basili, V. R., "Software defect reduction top 10 list", IEEE Computer, Enero, 2001.
- [Booch 1994] Booch, G., *Scenarios*, Revista Road, Volumen 1, número 3, Septiembre – Octubre, 1994, pp 3-6.
- [Breitman 1998] Breitman, K.K., Leite, J.C., "A Framework for Scenario Evolution", III IEEE International Conference on Requirements Engineering, ICRE'98, Colorado Springs, Colorado, USA, Abril 6-10, 1998, pp 214-221.
- [Breitman 2000] Breitman, K.K., Leite, J.C., "Scenario Evolution: a Closer View on Relationship", International Conference on Requirements Engineering, ICRE'2000, Schaumburg, Illinois, USA, Junio 19-23, 2000.
- [Cockton 2001] Cockton, G., "Let's Get It all Together: Literate Development and the Integration of HCI Research," ACM CHI'98 Basic Research Symposium, Los Angeles, USA, Abril, 19-20 1998, disponible en http://www.cet.sunderland.ac.uk/~cs0gco/brs_full.htm, Version 1.2 de 11/4/01 accedido el 15/1/2003, 2001.
- [Davis 1993] Davis, A., *Software Requirements Objects, functions and states*, PrenticeHall international inc, 1993.
- [Dömges 1998] Dömges, R., Pohl, K., "Adapting Traceability Environments to Project-specific Needs", *Communications of the ACM* Vol 41 Nro 12, Diciembre, 1998, pp 54-62.
- [Firesmith 1994] Firesmith, D. G., *Modeling the dynamic behavior of system, mechanisms, and classes with scenarios*. Revista Road, Volumen 1, número 2, Julio – Agosto, 1994, pp 33-36.
- [Gabb 1997] Gabb, A. P., Maheswaran, N., Allwright, A. M., *Requirements Management Tools Evaluation User Needs and Evaluation Criteria*, Information Technology Division, Electronics and Surveillance Research Laboratory, DSTO-GD-0139, <http://www1.tpgi.com.au/users/agabb/>, Australia, 1997.
- [Gamma 1995] Gamma, E., Helm, R., Johnson, R., Vlissides, J., *Design patterns elements of reusable object-oriented software*, Addison-Wesley Publishing Company, 1995.
- [Garvin 1984] Garvin, D., "What does product quality really mean?", *Sloan Management Review*, Vol 26, No 1, 1984.
- [Gotel 1994] Gotel, O.C.Z., Finkelstein, A.C.W., "An Analysis of the Requirements Traceability Problem", International Conference on Requirements Engineering, ICRE'94, Los Alamitos, California, Abril, 1994, pp 94-101.
- [Gotel 1995] Gotel, O.C.Z., "Contributions Structures for Requirements Traceability", University of London, Department of Computing, Imperial College of Science and Medicine, PhD Thesis, Agosto, 1995.

- [Hermodsson 1998] Hermodsson, K., Regnell, B., Bratthall, L., "Core functionality of Requirements Engineering Tools", master thesis, Quality Laboratories Sweden AB (Q-Labs), IDEON Research Park in collaboration and Department of Communication Systems, Lund University (TTS), Lund, Sweden, Febrero 12, 1998.
- [Hilburn 1999] Hilburn, T.B., Hirmanpour, I., Khajenoori, S., Turner, R., Qasem, A., "A Software Engineering Body of Knowledge", Version 1.0, SEI, Technical Report, CMU/SEI-99-TR-004, ESC-TR-99-004, 1999.
- [Jacobson 1994] Jacobson, I., Basic use-case modeling, Revista Road, Volumen 1, número 2, Julio-Agosto, 1994, pp 15-19.
- [Jacobson 1994b] Jacobson, I., Basic use-case modeling, Revista Road, Volumen 1, número 3, Septiembre-October, 1994, pp 7-9.
- [Jackson 1995] Jackson, M., Software Requirements & specifications. A lexicon of practice, principles and prejudices, Addison-Wesley publishing company, ACM Press. 1995.
- [Jackson 1995b] Jackson, D., Problem Decomposition for Reuse, School of Computer Science, Carnegie Mellon University, Pittsburgh, PA 15213, USA, July 14, 1995.
- [Jackson 1999] Jackson, M., Problem Analysis and Structure, AT&T Research, Florham Park NJ, USA, y consultor independiente, London, England, 1999.
- [Jackson 1999b] Jackson, M., Problem Analysis Using Small Problem Frames, South African Computer Journal 22; Special Issue on WOFACS'98, 1999, pp47-60.
- [Jackson 2000] Jackson, M., "The Real World", AT&T Research, Florham Park NJ, USA, y consultor independiente, London, England, Proceedings of a Symposium to celebrate the work of C A R Hoare, 2000.
- [Kotonya 1998] Kotonya, G., Sommerville, I., Requirements Engineering, Processes and Techniques, Chichester, John Wiley & Sons, 1998.
- [Lavazza 2000] X047 Lavazza, L., Valetto, G., "Enhancing Requirements and Change Management through Process Modelling and Measurement", Fourth International Conference on Requirements Engineering, ICRE'2000, Schaumburg, Illinois, USA, Junio 19-23, 2000, pp 106-115.
- [Leffingwell 2000] Leffingwell, D., Widrig, D., Managing Software Requirements – a unified approach, Object Technology Series (Series editors Booch Jacobson Rumbaugh), Addison-Wesley, ISBN 0201615932, 2000.
- [Leite 1997] Leite, J.C., Rossi, G., et al.: "Enhancing a Requirements Baseline with Scenarios". Proceedings of RE 97', IEEE Third International Requirements Engineering Symposium, IEEE Computer Society Press, 1997, pp 44-53.
- [Leonardi 2001] Leonardi, C., Leite J.C., Rossi G., "Una Estrategia de Modelado Conceptual de Objetos basada en Modelos de Requisitos en Lenguaje Natural", Tesis de maestría, <http://www-di.inf.puc-rio.br/~julio/teses.htm>, Facultad de informática, Universidad Nacional de La Plata, Argentina, Noviembre, 2001.
- [Loucopoulos 1995] Loucopoulos P., Karakostas V., System Requirements Engineering, McGraw-Hill, 1995
- [Mizuno 1983] Mizuno Y., "Software Quality Improvement", IEEE Computer, Vol. 16, No. 3, March 1983, pp 66-72.
- [Mauco 1997] Mauco V., Ridao, M., del Fresno, M., Rivero, L., Doorn, J., Ingeniería de Requisitos, Proyecto: Sistema de Planes de Ahorro, Reporte Final, ISISTAN, UNCPBA, Diciembre, 1997.
- [Nuseibeh 2000] Nuseibeh, B., Easterbrook, S., "Requirements engineering: a roadmap", International Conference on Software Engineering, ICSE 2000, Future of SE Track 2000, 2000, pp 35-46.
- [Parravicini 1997] Parravicini, L., Tolbaños, D., Valentino, F., Leite, J., "Ingeniería de Requerimientos", informe técnico, comunicación privada del autor, 1997.
- [Paulk 1999] Paulk, M.C., "Capability maturity model v1.1", <http://www.sei.cmu.edu>, 1999.
- [Pinheiro 1996] Pinheiro, F.A.C., Goguen, J.A., "An Object Oriented Tool for Tracing Requirements", IEEE Software Vol 13 Nro 2, Marzo, 1996, pp 52-64.
- [Pinheiro 2000] Pinheiro, F.A.C., "Formal and Informal Aspects of Requirements Tracing", III Workshop de Engenharia de Requisitos, WER'2000, Rio de Janeiro, Brasil, Julio, 2000.

- [Potts 1989] Potts, C., "A Generic Model for Representing Design Methods", 11th International Conference on Software Engineering, ICSE'89, Pittsburg, USA, 1989, pp 217-226.
- [Potts 1994] Potts, C., Takahash, K., Anton, A., Inquiry based requirements analysis. IEEE Software, Marzo 1994, pp 21-32.
- [Potts 1995] Potts, C., "Using schematic scenarios to understand user needs", Symposium On Designing Interactive Systems: Processes, Practises, Methods & Techniques, DIS '95, University Of Michigan, USA, 1995, pp 247-256.
- [Ramesh 1998] Ramesh, B., "Factors Influencing Requirements Traceability Practice", Communications of the ACM Vol 41 Nro 12, Diciembre, 1998, pp 37-44.
- [Rolland 1994] Rolland, C., "Modelling the Evolution Artifacts", First International Conference on Requirements Engineering, ICRE'94, IEEE Computer Society Press, Colorado Springs, Colorado, USA, Abril 18-22, 1994, pp 216-219.
- [Rubin 1992] Rubin, K., Goldberg, A.: Object Behavior Analysis. Communication of ACM Vol. 35 No. 9, September 1992, pp48-60.
- [Sommerville 1995] Sommerville, I., Software Engineering, Addison-Wesley, 1995.
- [Sommerville 1997] Sommerville, I., Sawyer, P., Requirements Engineering, Chichester, John Wiley & Sons, 1997.
- [Sommerville 1997b] Sommerville, I., Sawyer, P., Requirements Engineering: A Good Practice Guide, Computing Department, Lancaster University, John Wiley & sons ltd, ISBN 0 471 97444 7, 1997.
- [Schwabe 1995] Schwabe, D., Rossi, G., "An Object Oriented Hypermedia Design Model", Communication of ACM Vol 38 Nro 8, 1995, pp 45-46.
- [Welty 1995] Welty, C. A., "Towards an Epistemology for Software Representations", 10th Knowledge-Based Software Engineering Conference, Boston, Massachusetts, IEEE Computer Society Press, Noviembre, 1995.
- [Wiegers 1999] Wiegers, K.E., Software Requirements, Microsoft Press, Redmond, 1999.
- [Wieringa 1995] Wieringa, R., An Introduction to Requirements Traceability, Reporte Técnico, Faculty of Mathematics and Computer Science, Vrije Universiteit, Esprit Project 2RARE, Noviembre 1, 1995.
- [Wilkinson 1995] Wilkinson, N.M., Using CRC Cards: An Informal Approach to Object-Oriented, AT&T Bell Laboratories, 1995.
- [Wood 1997] L.E. Wood, Semi-structured interviewing for user-centered design, Interactions of the ACM, april-may 1997, pp 48-61.
- [Zorman 1995] Zorman, L.: REBUS: Requirements Envisaging By Utilizing Scenarios. A Dissertation presented to the Faculty of the Graduate School University of Southern California, USA. In partial fulfillment of the Requirements for the Degree Doctor of Philosophy (Computer Science), 1995.

9. Anexos

9.1. LEL, escenarios y tarjetas CRC del círculo de ahorro

9.1.1. LEL

Adherente / Adherentes

- Notion
 - El *adherente* es un *solicitante* que fue aceptado por la *administradora*
 - El *adherente* es el integrante del *grupo*
- Behavioral responses
 - El *adherente* paga una *cuota mensual*
 - El *adherente* puede *transferir plan*
 - El *adherente* puede *licitar*
 - El *adherente* puede *rechazar bien tipo*
 - El *adherente* puede *renunciar al plan de ahorro*
 - El *adherente* se convierte en *adjudicatario* cuando recibe el *bien tipo*

Adherente deja plan / Dejar plan / Renunciar al plan de ahorro

- Notion
 - El *adherente* abandona el *plan de ahorro* por su propia voluntad
- Behavioral responses
 - Puede que el *grupo* tenga que disolverse por falta de miembros

Adjudicatario / Adjudicatarios / Adjudicado

- Notion
 - Es el *adherente* que ganó el *sorteo* y aceptó el *bien tipo* teniendo las *cuotas* al día
 - O, es el *adherente* que ganó la *licitación*
- Behavioral responses
 - El *adjudicatario* debe pagar el *derecho de adjudicación*
 - El *adjudicatario* puede *cambiar el bien tipo adjudicado*, pagando la diferencia si el nuevo *bien* es de mayor valor que el *bien adjudicado*, ó computándose la diferencia para cancelar las últimas *cuotas* del *plan* en caso de ser un *bien* de menor valor. La *administradora* puede negarse a realizar el cambio
 - Elegir *aseguradora* para el *seguro de bien tipo*

Administradora

- Notion
 - Entidad que se encarga de organizar *planes de ahorro* haciendo de intermediario entre el *fabricante* y los *adherentes*
- Behavioral responses
 - La *administradora* evalúa la *solicitud de adhesión* de cada *solicitante*
 - La *administradora* devuelve el valor de la *cuota de ingreso* si el *solicitante* es rechazado
 - La *administradora* arma el *grupo*
 - La *administradora* determina el valor de la *cuota mensual*
 - La *administradora* determina el *seguro de vida*
 - La *administradora* puede *expulsar adherente*
 - La *administradora* cobra la *cuota mensual* en caso de vencimiento
 - La *administradora* realiza el *sorteo*
 - La *administradora* es responsable de *evaluar licitaciones*
 - La *administradora* entrega el *bien tipo*
 - La *administradora* puede *disolver el grupo*
 - La *administradora* puede determinar el *cambio de bien tipo*
 - La *administradora* se *comunica fehacientemente* con el *grupo*

Aseguradora

- Notion
 - Compañía que opera en el país, encargada de asegurar a los adherentes y los bienes tipos
- Behavioral responses
 - La aseguradora asegura al adherente
 - La aseguradora asegura al bien tipo
 - La aseguradora debe pagar el seguro de vida o el de un bien tipo en caso de siniestro

asegurar bien tipo

- Notion
 - Acción por la cual el adjudicatario contrata una compañía aseguradora para cubrir el bien adjudicado
- Behavioral responses
 - El adjudicatario contrata una aseguradora

Banco

- Notion
 - Entidad encargada de cobrar las cuotas mensuales en caso de no estar vencidas
- Behavioral responses
 - El banco cobra las cuotas mensuales
 - El banco envía los comprobantes de pago a la administradora

Bien tipo rechazado / Rechaza el bien tipo / Rechazar bien tipo

- Notion
 - El adherente que ganó el sorteo, no acepta el bien tipo
- Behavioral responses
 - El adherente no se convierte en adjudicatario
 - El adherente es expulsado, si se rechaza el bien tipo por tercera vez

Bienes tipos / Bien tipo / bien

- Notion
 - Vehículo que desean conseguir los adherentes de un determinado plan de ahorro
- Behavioral responses
 - La administradora entrega el bien tipo a un adherente
 - El bien tipo transforma a un adherente en adjudicatario
 - La administradora puede disponer cambio de bien tipo

cambiar el bien tipo adjudicado

- Notion
 - El adjudicatario elige un bien tipo distinto del adjudicado
- Behavioral responses
 - El adjudicatario debe pagar la diferencia si el nuevo bien es de mayor valor que el bien adjudicado
 - La administradora le computa la diferencia para cancelar las últimas cuotas del plan si el nuevo bien es de menor valor que el bien adjudicado
 - La administradora puede negarse a cambiar el bien tipo adjudicado

Cambio de bien tipo

- Notion
 - La administradora elige otro bien tipo para el plan de ahorro, ya que el fabricante decide discontinuar un modelo o cambiar el precio
- Behavioral responses
 - La administradora modifica el valor de las cuotas y el bien tipo siempre y cuando el grupo lo acepte

- La administradora puede disolver el grupo por falta de conformidad de los adherentes

Carga de administración

- Notion
 - Importe abonado por el adherente mensualmente en concepto de gastos administrativos
- Behavioral responses
 - La carga de administración forma parte de la cuota mensual

Comunica fehacientemente / Comunicar fehacientemente / Comunicación fehaciente

- Notion
 - Comunicación realizada entre la administradora y los adherentes mediante telegrama con aviso de retorno, carta documento o nota de recibo
- Behavioral responses
 - Se comunica fehacientemente el cierre del grupo
 - Se comunica fehacientemente al adherente que fue adjudicado
 - Se comunica fehacientemente el cambio de bien tipo

Cuota comercial

- Notion
 - Es el valor de la cuota pura más la carga de administración
- Behavioral responses
 - La cuota comercial forma parte de la cuota mensual

Cuota de ingreso

- Notion
 - Es el valor de la cuota comercial mas el derecho de admisión
- Behavioral responses
 - La cuota de ingreso es abonada por el solicitante al entregar la planilla de adhesión

Cuota mensual / Cuota / Cuotas

- Notion
 - Monto mensual que debe pagar tanto el adherente como el adjudicatario
 - Es la suma de la cuota comercial y del seguro de vida
- Behavioral responses
 - La cuota es pagada por el adherente y el adjudicatario
 - La cuota es calculada por la administradora

Cuotas puras / Cuota pura

- Notion
 - Es el valor móvil dividido la cantidad de cuotas del plan de ahorro
- Behavioral responses
 - La cuota pura forma parte de la cuota mensual

Cupón de pago

- Notion
 - Es el formulario del que la administradora dispone para que los adherentes y adjudicatarios efectuen los pagos correspondientes
- Behavioral responses
 - El cupón de pago es entregado a los adherentes por la administradora
 - El cupón de pago es entregado al banco o a la administradora al pagar la cuota
 - El cupón de pago es sellado y devuelto al adherente al realizar el pago

Derecho de adjudicación

- Notion
 - Importe abonado por el adherente por unica vez cuando es adjudicado con el bien tipo

- Behavioral responses
 - El derecho de adjudicación es abonado por el adherente

Derecho de admisión

- Notion
 - Importe abonado por el solicitante en la cuota de ingreso
- Behavioral responses
 - El derecho de admisión forma parte de la cuota de ingreso

Disolver el grupo

- Notion
 - El grupo es disuelto por la administradora intentar el cambio del bien tipo y no haber concenso de los adherentes
- Behavioral responses
 - La administradora comunica fehacientemente al grupo

Entregar bien tipo

- Notion
 - Acción por la cual la administradora entrega el bien tipo suministrado por el fabricante al adjudicatario
- Behavioral responses
 - Al entregar bien tipo el adjudicatario debe pagar el derecho de adjudicación
 - Al entregar bien tipo la administradora realiza una prenda sobre el bien hasta que el adjudicatario haya terminado de pagar sus cuotas
 - Al entregar bien tipo el adjudicatario debe asegurar bien tipo

Evaluar licitaciones / Licitación

- Notion
 - La administradora analiza ante escribano público cada una de las propuestas entregadas por los adherentes eligiendo la de mayor monto como la favorecida
 - No se evalúa licitaciones si no hay ninguna entregada o si no se alcanza a comprar más de un bien tipo con el dinero recaudado
- Behavioral responses
 - En la licitación el adherente ganador se transforman en adjudicatario
 - Luego de la licitación la administradora le devuelve el cheque de la licitación a los adherentes que no ganaron

Expulsar adherente / Adherente es expulsado

- Notion
 - La administradora saca un adherente del grupo por falta de pago o por rechazar tres veces el bien tipo que ganó por sorteo
- Behavioral responses
 - Al expulsar adherente hay una posibilidad de disolver el grupo por falta de adherentes

Fabricante

- Notion
 - Empresa que produce o comercializa los bienes tipos que compra la administradora
- Behavioral responses
 - El fabricante entrega el bien tipo a la administradora
 - El fabricante notifica los cambios de precio de un bien tipo a la administradora
 - El fabricante notifica la discontinuación de un bien tipo a la administradora

Grupo

- Notion

- Conjunto de *adherentes* que desean obtener un mismo *bien tipo* participando de un mismo *plan de ahorro*
 - El *grupo* inicialmente esta formado por tantos *adherentes* como el doble de la cantidad de *cuotas* establecidas por la *administradora*
 - Behavioral responses
 - Se puede *disolver el grupo* por consenso de los *adherentes*
 - El *grupo* puede aceptar o rechazar el cambio del *bien tipo*
-

Licitar

- Notion
 - Acción por la cual el *adherente* entrega un sobre cerrado con un cheque con el importe de la *licitación* a la *administradora*
 - El monto mínimo de una *licitación* es de una *cuota pura* y el máximo es la cantidad de *cuotas puras* que le faltan pagar al *adherente*
 - Behavioral responses
 - La *administradora* puede aceptar la *licitación*
 - La *administradora* puede rechazar la *licitación*
-

Plan de ahorro / Planes de ahorro / plan

- Notion
 - Es un servicio brindado por la *administradora* a un *grupo* de *adherentes* interesados en obtener un mismo *bien tipo* permitiéndoles pagar en *cuotas*
 - Tiene un número determinado de *cuotas*
 - La forma de adquisición puede ser por *sorteo* o *licitación*
 - Behavioral responses
 - El *bien tipo* o la *cuota* puede ser modificado por la *administradora*
 - El *plan de ahorro* puede ser transferido de un *adherente* a otra persona
-

Plan media cuota

- Notion
 - Tipo especial de *plan de ahorro*
 - El monto de las *cuotas* es la mitad de su valor real y en el momento de la adjudicación el *adherente* debe pagar el 50% del valor del *bien tipo*
 - Behavioral responses
 - El *adherente* debe pagar el 50% del valor de la *cuota*
 - El *adherente* debe pagar el 50% del valor del *bien tipo* cuando es *adjudicado*
-

Planilla de adhesión

- Notion
 - Formulario que entrega el *solicitante* a la *administradora* para formar parte de un *plan de ahorro*
 - El formulario contiene los datos personales, los del *bien tipo* y los del *seguro de vida*
 - Behavioral responses
 - La *planilla de adhesión* es usada por el *solicitante* para pasar a ser un *adherente* La *planilla de adhesión* es llenada por el *solicitante*
 - La *planilla de adhesión* es entregada a la *administradora*
-

Seguro de bien tipo

- Notion
 - Contrato realizado por el *adjudicatario* al *bien tipo* para establecer una garantía para la *administradora*
 - Behavioral responses
 - El *adjudicatario* debe pagar la *cuota* del seguro del *bien tipo* a la *administradora*
 - La *administradora* cobra el seguro del *bien tipo* en caso de siniestro
-

Seguro de vida

- Notion
 - Contrato realizado por la *administradora* al *adherente* cuando ingresa al *plan de ahorro* para terminar su pago en caso de fallecimiento
 - Behavioral responses
 - La *administradora* cobra la *cuota* del *seguro de vida* al *adherente*
 - La *administradora* fija el valor del *seguro de vida*
 - La *administradora* cobra el *seguro de vida* en caso de fallecimiento y entrega el *bien tipo* a los herederos
-

Solicitante

- Notion
 - Persona que desea ingresar al *plan de ahorro*
 - Behavioral responses
 - El *solicitante* pide la *planilla de adhesión* a la *administradora*
 - El *solicitante* completa la *planilla de adhesión*
 - El *solicitante* entrega la *planilla de adhesión* a la *administradora*
-

Solicitud cambio de bien

- Notion
 - Un *adjudicatario* le comunica a la *administradora* que desea cambiar el *bien tipo adjudicado*
 - Behavioral responses
 - El *adjudicatario* le envía la *solicitud cambio de bien* a la *administradora*
 - La *administradora* puede rechazar o aceptar la *solicitud cambio de bien*
-

Solicitud de adhesión / Solicitudes de adhesión

- Notion
 - El *solicitante* entrega la *planilla de adhesión* y la *cuota de ingreso* a la *administradora*
 - Behavioral responses
 - La *solicitud de adhesión* puede ser aceptada por la *administradora* y el *solicitante* pasa a ser *adherente* y se le realiza un *seguro de vida*
 - La *solicitud de adhesión* puede ser rechazada por la *administradora* y se le reintegra el valor de la *cuota de ingreso* al *solicitante*
-

Sortear / Sorteo

- Notion
 - Acción realizada por la *administradora* una vez por mes ante escribano público, en la cual participan todos los *adherentes* del *grupo*
 - Behavioral responses
 - En el *sorteo* se elige un *adherente* ganador
 - Si el *adherente* que ganó el *sorteo* no tiene sus *cuotas* al día o *rechaza el bien tipo* se toma el siguiente de la lista de sorteados
-

Transferir plan

- Notion
 - Acción por la cual un *adherente* que tiene sus *cuotas* al día, vende su *plan* a otra persona
 - Behavioral responses
 - Si el *adherente* no *comunica fehacientemente* la *administradora* puede rechazar dicha transferencia
-

Valor móvil

- Notion
 - Precio de lista del *bien tipo*
- Behavioral responses

- La *administradora* puede modificar el *valor móvil* de acuerdo a descuentos y/o modificaciones sugeridas por el *fabricante*

9.1.2. Escenarios

Adjudicación

- Goal
 - El *adherente* realiza los trámites necesarios para recibir el *bien tipo*
- Context
 - El *adherente* es ganador del *sorteo* con las *cuotas* al día o de la *licitación*
- Resources
 - *Derecho de adjudicación* - Formulario de pedido de vehículo
- Actors
 - *Administradora* - *Adherente*
- Episodes
 - El *adherente* deberá pagar el *derecho de adjudicación*
 - El *adherente* debe llenar un formulario de pedido de vehículo
 - El *adherente* se convierte en *adjudicatario*
- Links defUsr

Armar un grupo

- Goal
 - Formar un nuevo *grupo* de *adherentes* a un *plan de ahorro*
- Context
 - Se tienen tantas *solicitudes de adhesión* como miembros requiere el *grupo*
- Resources
 - *Planilla de adhesión*
- Actors
 - *Administradora*
- Episodes
 - Se arma la lista de los miembros del nuevo *grupo* incluyendo sus datos y los números de orden asignados a cada *planilla de adhesión*
 - Se registra al nuevo *grupo* para participar en la próxima *adjudicación*
 - Se *comunica fehacientemente* la aceptación a los nuevos *adherentes*
 - Se envían los cupones de pago correspondientes al *plan de ahorro* a todos los *adherentes*
- Links defUsr

Asegurar adherente

- Goal
 - Percibir una indemnización en caso de fallecimiento del *adherente*
- Context
 - Ocurre cuando la *administradora* acepta la *solicitud de adhesión* de una persona
- Resources
 - Póliza
- Actors
 - *Administradora* - *aseguradora* - *adherente*
- Episodes
 - La *administradora* elige una *aseguradora* y contrata una póliza para todo el lapso del *plan de ahorro*
 - La *aseguradora* cumple con los trámites internos
- Links defUsr

Asegurar bien tipo

- Goal

- Percibir una indemnización en caso de robo, hurto o siniestro del *bien tipo*
- Context
 - Se produce en el momento en que la *administradora* le entrega el *bien tipo* a un *adjudicatario*
- Resources
 - Póliza - lista de aseguradoras
- Actors
 - *Administradora* - *aseguradora* - *adjudicatario*
- Episodes
 - El *adjudicatario* debe elegir una compañía *aseguradora* de la lista suministrada por la *administradora*
 - Se endosa la póliza a nombre de la *administradora*
 - La *aseguradora* cumple con los trámites internos
- Links defUsr

Cambiar bien tipo

- Goal
 - Fijar un nuevo *bien tipo* y el monto de la *cuota* para el determinado *plan*
- Context
 - Discontinuación del modelo del *bien tipo* por parte del *fabricante* o cambio de precio del *bien tipo* actual
- Resources
 - *Plan de ahorro*
- Actors
 - *Administradora* - *grupo*
- Episodes
 - La *administradora comunica fehacientemente* al *grupo* la propuesta de cambio
 - **if** el *grupo* rechaza el cambio **then** se disuelve el *grupo*
 - **if** el *grupo* acepta el cambio **then** el *plan de ahorro* continua
- Links defUsr

Cambiar bien tipo adjudicado

- Goal
 - Hacerse acreedor de un *bien tipo* diferente al del *plan de ahorro*
- Context
 - El *adherente* debe ser *adjudicatario*
- Resources
 - *Solicitud cambio de bien*
- Actors
 - *Administradora* - *adjudicatario*
- Episodes
 - El *adjudicatario* llena la *solicitud cambio de bien*
 - **exception:** La *administradora* se niega a realizar el *cambio de bien tipo*
 - **if** el nuevo *bien* es de mayor valor **then** el *adjudicatario* debe pagar la diferencia antes de retirar el *bien* elegido
 - **if** el nuevo *bien* es de menor valor **then** la diferencia se le computará para cancelar las últimas *cuotas* del *plan*
- Links defUsr

Dejar plan

- Goal
 - El *adherente* decide renunciar al *plan*
- Context
 - El *adherente* tiene sus *cuotas* al día y decide dejar el *plan de ahorro* por su propia cuenta.
- Resources

- *Plan de ahorro*
 - Actors
 - *Administradora - adherente*
 - Episodes
 - El *adherente comunica fehacientemente* a la *administradora* su decisión de abandonar el *plan de ahorro*
 - **if** faltan miembros para el *grupo* **then** éste se disuelve
 - Links defUsr
-

Disolver un grupo

- Goal
 - Eliminar *grupo* de *adherentes* de un *plan de ahorro*
 - Context
 - El *grupo* se queda con menos *adherentes* de la cantidad mínima o no acepta el cambio del valor de la *cuota* o del *bien tipo*
 - Resources
 - *plan de ahorro - cuotas puras*
 - Actors
 - *Administradora - grupo*
 - Episodes
 - La *administradora comunica fehacientemente* a los *adherentes* la decisión de disolución
 - La *administradora* devolverá el importe de las *cuotas* pagadas a los *adherente* que renunciaron al *plan de ahorro*
 - La *administradora* reintegrará el importe actualizado de las *cuotas puras* o parte de ellas que se hubieran pagado por anticipado
 - Los excedentes son repartidos entre los *adherente* que no hubieran renunciado al *plan de ahorro*
 - Links defUsr
-

Entregar bien tipo

- Goal
 - Poner a disposición del *adjudicatario* el *bien tipo*
 - Context
 - La entrega del *bien tipo* se lleva a cabo en una concesionaria
 - Resources
 - *bien tipo - Seguro de bien tipo*
 - Actors
 - *Administradora - fabricante - adjudicatario*
 - Episodes
 - El *adjudicatario* debe **asegurar bien tipo**
 - El *adjudicatario* retira el *bien tipo* de la concesionaria
 - Links defUsr
-

Evaluar licitaciones

- Goal
 - Adjudicación de un *bien tipo* a un *adherente*
- Context
 - En las dependencias de la *administradora* ante escribano publico
- Resources
 - Sobres con licitaciones
- Actors
 - *administradora* - escribano
- Episodes
 - La *administradora comunica fehacientemente* el día y lugar donde se llevará a cabo la evaluación de la *licitación*

- Se abren los sobres
- Se crea un ranking de *adherentes* ordenados por monto licitado
- **if** existe mas de una *licitación* con el mismo importe **then** entonces se toma como orden el del *sorteo*
- **Adjudicación** del *bien tipo* al *adherente* ganador
- Links defUsr

Expulsar adherente

- Goal
 - Separar a un *adherente* de un *grupo*
- Context
 - Incumplimiento de pago de un *adherente* ó rechazo de un *bien tipo* tres veces
- Resources
 - *Cuotas*
- Actors
 - *Administradora* - *adherente*
- Episodes
 - La *administradora comunica fehacientemente* al *adherente* de su exclusión del *grupo*
 - **if** faltan miembros del *grupo* **then** se procede a **disolver un grupo**
- Links defUsr

Licitar

- Goal
 - Obtener el *bien tipo* sin tener que esperar a ser favorecido por el *sorteo*
- Context
 - El *adherente* debe tener sus *cuotas* al día
- Resources
 - Sobre con la *licitación*
- Actors
 - *Administradora* - *adherente*
- Episodes
 - El *adherente* entrega un sobre cerrado con un cheque con el importe de la *licitación*
 - Si el *adherente* entrega mas del máximo, entonces la *administradora* le devolverá dentro de las 48 hs. la diferencia
- Links defUsr

Pagar cuota

- Goal
 - El *adherente* cumple con la obligación contraída al ingresar a un *plan de ahorro*
- Context
 - Paga en el *banco* una *cuota* que no esta vencida
- Resources
 - *Cupón de pago*
- Actors
 - *Adherente* - *banco* - *administradora*
- Episodes
 - El *adherente* entrega el empleado del *banco* el *cupón de pago* con el importe correspondiente
 - El empleado devuelve el *cupón de pago* sellado
 - El *banco* envía el comprobante de pago a la *administradora*
- Links defUsr

Pagar cuota vencida

- Goal
 - El *adherente* cumple con la obligación contraída al ingresar a un *plan de ahorro*

- Context
 - Paga en las instalaciones de la *administradora* una *cuota* vencida
- Resources
 - *Cupón de pago*
- Actors
 - *Adherente* - *administradora*
- Episodes
 - El *adherente* entrega el empleado de la *administradora* el *cupón de pago* con el importe correspondiente a la *cuota mensual* actualizada, mas recargos por mora
 - El empleado devuelve el *cupón de pago* sellado
- Links defUsr

Rechazar bien tipo

- Goal
 - El *adherente* desea evitar convertirse en *adjudicatario*
- Context
 - El *adherente* debe haber ganado el *sorteo*
- Resources
 - *Bien tipo*
- Actors
 - *Administradora* - *adherente*
- Episodes
 - El *adherente* que salió favorecido en el **Sorteo** no *comunica fehacientemente* que acepta el *bien* dentro de un lapso determinado de tiempo especificado por la *administradora*
 - **if** es la tercera vez que el *adherente rechaza el bien tipo* **then** se procede a **expulsar adherente**
- Links defUsr

Solicitar ingreso

- Goal
 - Una persona desea ingresar a un *plan de ahorro*
- Context
 - El *solicitante* pide información acerca de los *planes de ahorro* existentes
- Resources
 - *Planilla de adhesión* - *Cuota de ingreso*
- Actors
 - *Administradora* - *Solicitante*
- Episodes
 - El *solicitante* pide la *planilla de adhesión*
 - El *solicitante* completa dicha planilla y la entrega, abonando la *cuota de ingreso*
 - **if** la *administradora* rechaza la solicitud evaluada **then** ésta devolverá al *solicitante* el importe correspondiente a la *cuota de ingreso*
 - **if** la *administradora* acepta la solicitud **then** se debe **Asegurar adherente**
- Links defUsr

Sorteo

- Goal
 - Adjudicación de un *bien tipo* a un *adherente*
- Context
 - Ocurre en las dependencias de la *administradora* una vez por mes
- Resources
 - Lista de *adherentes*
- Actors
 - *Administradora* - escribano

- Episodes
 - La *administradora comunica fehacientemente* el día y lugar donde se llevará a cabo el **Sorteo**
 - Se elige al azar un orden para los *adherentes*, con la presencia de un escribano
 - Se toma el primero de la lista de sorteados y sino tiene sus *cuotas* al día o *rechaza el bien tipo* se toma el siguiente de la lista
 - **Adjudicación** del *bien tipo* al *adherente* ganador
- Links defUsr

Transferir plan

- Goal
 - Cambiar el dueño de un *plan*
- Context
 - El *adherente* con sus *cuotas* al día decide transferir el *plan de ahorro*
- Resources
 - *Plan de ahorro*
- Actors
 - *administradora* - *adherente*
- Episodes
 - El *adherente comunica fehacientemente* el cambio a la *administradora*
 - La *administradora* registra las modificaciones ocasionadas por nuevo *adherente*
 - El *adherente* deja de pertenecer al *plan de ahorro*
- Links defUsr

9.1.3. Tarjetas CRC

Adherente

- Responsibilities
 - El *adherente* paga una *cuota mensual*
 - El *adherente* puede *transferir plan*
 - El *adherente* puede *licitar*
 - El *adherente* puede *rechazar bien tipo*
 - El *adherente* puede *renunciar al plan de ahorro*
 - El *adherente* se convierte en *adjudicatario* cuando recibe el *bien tipo*
- Collaborations
 - adjudicatario in scenario **Adjudicación**
 - Administradora in scenario **Dejar plan**
 - Administradora in scenario **Disolver un grupo**
 - Administradora in scenario **Evaluar licitaciones**
 - Administradora in scenario **Expulsar adherente**
 - Administradora in scenario **Licitar**
 - Administradora in scenario **Pagar cuota**
 - Administradora in scenario **Pagar cuota vencida**
 - Administradora in scenario **Rechazar bien tipo**
 - Administradora in scenario **Sorteo**
 - Administradora in scenario **Transferir plan**
 - banco in scenario **Pagar cuota**
 - bien tipo in scenario **Evaluar licitaciones**
 - bien tipo in scenario **Rechazar bien tipo**
 - bien tipo in scenario **Sorteo**
 - Comunica fehacientemente in scenario **Armar un grupo**
 - Comunica fehacientemente in scenario **Dejar plan**
 - Comunica fehacientemente in scenario **Disolver un grupo**

- Comunica fehacientemente in scenario [Evaluar licitaciones](#)
- Comunica fehacientemente in scenario [Expulsar adherente](#)
- Comunica fehacientemente in scenario [Rechazar bien tipo](#)
- Comunica fehacientemente in scenario [Sorteo](#)
- Comunica fehacientemente in scenario [Transferir plan](#)
- cuota mensual in scenario [Disolver un grupo](#)
- cuota mensual in scenario [Pagar cuota vencida](#)
- cuota mensual in scenario [Sorteo](#)
- derecho de adjudicación in scenario [Adjudicación](#)
- Evaluar licitaciones in scenario [Evaluar licitaciones](#)
- Evaluar licitaciones in scenario [Licitar](#)
- grupo in scenario [Armar un grupo](#)
- grupo in scenario [Dejar plan](#)
- grupo in scenario [Expulsar adherente](#)
- plan in scenario [Armar un grupo](#)
- plan in scenario [Dejar plan](#)
- plan in scenario [Disolver un grupo](#)
- plan in scenario [Transferir plan](#)
- planilla de adhesión in scenario [Armar un grupo](#)
- Rechazar bien tipo in scenario [Rechazar bien tipo](#)
- Rechazar bien tipo in scenario [Sorteo](#)
- sorteo in scenario [Evaluar licitaciones](#)

adjudicatario

- Responsibilities
 - El *adjudicatario* debe pagar el *derecho de adjudicación*
 - El *adjudicatario* puede *cambiar el bien tipo adjudicado*, pagando la diferencia si el nuevo *bien* es de mayor valor que el *bien adjudicado*, ó computándose la diferencia para cancelar las últimas *cuotas* del *plan* en caso de ser un *bien* de menor valor. La *administradora* puede negarse a realizar el cambio
 - Elegir *aseguradora* para el *seguro de bien tipo*
- Collaborations
 - Administradora in scenario [Asegurar bien tipo](#)
 - Administradora in scenario [Cambiar bien tipo adjudicado](#)
 - Adherente in scenario [Adjudicación](#)
 - aseguradora in scenario [Asegurar bien tipo](#)
 - bien tipo in scenario [Cambiar bien tipo adjudicado](#)
 - bien tipo in scenario [Entregar bien tipo](#)
 - Cambio de bien tipo in scenario [Cambiar bien tipo adjudicado](#)
 - cuota mensual in scenario [Cambiar bien tipo adjudicado](#)
 - derecho de adjudicación in scenario [Adjudicación](#)
 - plan in scenario [Cambiar bien tipo adjudicado](#)

Administradora

- Responsibilities
 - La *administradora* evalúa la *solicitud de adhesión* de cada *solicitante*
 - La *administradora* devuelve el valor de la *cuota de ingreso* si el *solicitante* es rechazado
 - La *administradora* arma el *grupo*
 - La *administradora* determina el valor de la *cuota mensual*
 - La *administradora* determina el *seguro de vida*
 - La *administradora* puede *expulsar adherente*
 - La *administradora* cobra la *cuota mensual* en caso de vencimiento
 - La *administradora* realiza el *sorteo*
 - La *administradora* es responsable de *evaluar licitaciones*

- La *administradora* entrega el *bien tipo*
- La *administradora* puede *disolver el grupo*
- La *administradora* puede determinar el *cambio de bien tipo*
- La *administradora* se *comunica fehacientemente* con el *grupo*
- Collaborations
 - Adherente in scenario [Dejar plan](#)
 - Adherente in scenario [Disolver un grupo](#)
 - Adherente in scenario [Evaluar licitaciones](#)
 - Adherente in scenario [Expulsar adherente](#)
 - Adherente in scenario [Licitar](#)
 - Adherente in scenario [Pagar cuota](#)
 - Adherente in scenario [Pagar cuota vencida](#)
 - Adherente in scenario [Rechazar bien tipo](#)
 - Adherente in scenario [Sorteo](#)
 - Adherente in scenario [Transferir plan](#)
 - adjudicatario in scenario [Asegurar bien tipo](#)
 - adjudicatario in scenario [Cambiar bien tipo adjudicado](#)
 - aseguradora in scenario [Asegurar adherente](#)
 - aseguradora in scenario [Asegurar bien tipo](#)
 - banco in scenario [Pagar cuota](#)
 - bien tipo in scenario [Cambiar bien tipo adjudicado](#)
 - bien tipo in scenario [Evaluar licitaciones](#)
 - bien tipo in scenario [Rechazar bien tipo](#)
 - bien tipo in scenario [Sorteo](#)
 - Cambio de bien tipo in scenario [Cambiar bien tipo adjudicado](#)
 - Comunica fehacientemente in scenario [Cambiar bien tipo](#)
 - Comunica fehacientemente in scenario [Dejar plan](#)
 - Comunica fehacientemente in scenario [Disolver un grupo](#)
 - Comunica fehacientemente in scenario [Evaluar licitaciones](#)
 - Comunica fehacientemente in scenario [Expulsar adherente](#)
 - Comunica fehacientemente in scenario [Rechazar bien tipo](#)
 - Comunica fehacientemente in scenario [Sorteo](#)
 - Comunica fehacientemente in scenario [Transferir plan](#)
 - cuota mensual in scenario [Cambiar bien tipo adjudicado](#)
 - cuota mensual in scenario [Disolver un grupo](#)
 - cuota mensual in scenario [Pagar cuota vencida](#)
 - cuota mensual in scenario [Sorteo](#)
 - cuota de ingreso in scenario [Solicitar ingreso](#)
 - Evaluar licitaciones in scenario [Evaluar licitaciones](#)
 - Evaluar licitaciones in scenario [Licitar](#)
 - grupo in scenario [Cambiar bien tipo](#)
 - grupo in scenario [Dejar plan](#)
 - grupo in scenario [Expulsar adherente](#)
 - plan in scenario [Asegurar adherente](#)
 - plan in scenario [Cambiar bien tipo](#)
 - plan in scenario [Cambiar bien tipo adjudicado](#)
 - plan in scenario [Dejar plan](#)
 - plan in scenario [Disolver un grupo](#)
 - plan in scenario [Transferir plan](#)
 - planilla de adhesión in scenario [Solicitar ingreso](#)
 - Rechazar bien tipo in scenario [Rechazar bien tipo](#)
 - Rechazar bien tipo in scenario [Sorteo](#)
 - Solicitante in scenario [Solicitar ingreso](#)
 - sorteo in scenario [Evaluar licitaciones](#)

aseguradora

- Responsibilities
 - La *aseguradora* asegurara al *adherente*
 - La *aseguradora* asegurar al *bien tipo*
 - La *aseguradora* debe pagar el *seguro de vida* o el de un *bien tipo* en caso de siniestro
- Collaborations
 - Administradora in scenario **Asegurar adherente**
 - Administradora in scenario **Asegurar bien tipo**
 - adjudicatario in scenario **Asegurar bien tipo**
 - plan in scenario **Asegurar adherente**

banco

- Responsibilities
 - El *banco* cobra las *cuotas* mensuales
 - El *banco* envia los comprobantes de pago a la *administradora*
- Collaborations
 - Administradora in scenario **Pagar cuota**
 - Adherente in scenario **Pagar cuota**

bien tipo

- Responsibilities
 - *Adjudicado* al *adherente*
 - La *administradora* lo entrega a un *adjudicatario*
- Collaborations
 - Adherente in scenario **Evaluar licitaciones**
 - Adherente in scenario **Rechazar bien tipo**
 - Adherente in scenario **Sorteo**
 - Administradora in scenario **Cambiar bien tipo adjudicado**
 - Administradora in scenario **Evaluar licitaciones**
 - Administradora in scenario **Rechazar bien tipo**
 - Administradora in scenario **Sorteo**
 - adjudicatario in scenario **Cambiar bien tipo adjudicado**
 - adjudicatario in scenario **Entregar bien tipo**
 - Cambio de bien tipo in scenario **Cambiar bien tipo adjudicado**
 - Comunica fehacientemente in scenario **Evaluar licitaciones**
 - Comunica fehacientemente in scenario **Rechazar bien tipo**
 - Comunica fehacientemente in scenario **Sorteo**
 - cuota mensual in scenario **Cambiar bien tipo adjudicado**
 - cuota mensual in scenario **Sorteo**
 - Evaluar licitaciones in scenario **Evaluar licitaciones**
 - plan in scenario **Cambiar bien tipo adjudicado**
 - Rechazar bien tipo in scenario **Rechazar bien tipo**
 - Rechazar bien tipo in scenario **Sorteo**
 - sorteo in scenario **Evaluar licitaciones**

cambiar el bien tipo adjudicado

- Responsibilities
- Collaborations

Cambio de bien tipo

- Responsibilities

- La *administradora* modifica el valor de las *cuotas* y el *bien tipo* siempre y cuando el *grupo* lo acepte
- Disolver *grupo* por falta de conformidad de los miembros del *grupo*
- La *administradora* cambia el valor de la *cuota mensual*
- Collaborations
 - adjudicatario in scenario [Cambiar bien tipo adjudicado](#)
 - Administradora in scenario [Cambiar bien tipo adjudicado](#)
 - bien tipo in scenario [Cambiar bien tipo adjudicado](#)
 - cuota mensual in scenario [Cambiar bien tipo adjudicado](#)
 - plan in scenario [Cambiar bien tipo adjudicado](#)

Comunica fehacientemente

- Responsibilities
 - Se *comunica fehacientemente* el cierre del *grupo*
 - Se *comunica fehacientemente* al *adherente* que fue *adjudicado*
 - Se *comunica fehacientemente* el *cambio de bien tipo*
- Collaborations
 - Adherente in scenario [Armar un grupo](#)
 - Adherente in scenario [Dejar plan](#)
 - Adherente in scenario [Disolver un grupo](#)
 - Adherente in scenario [Evaluar licitaciones](#)
 - Adherente in scenario [Expulsar adherente](#)
 - Adherente in scenario [Rechazar bien tipo](#)
 - Adherente in scenario [Sorteo](#)
 - Adherente in scenario [Transferir plan](#)
 - Administradora in scenario [Cambiar bien tipo](#)
 - Administradora in scenario [Dejar plan](#)
 - Administradora in scenario [Disolver un grupo](#)
 - Administradora in scenario [Evaluar licitaciones](#)
 - Administradora in scenario [Expulsar adherente](#)
 - Administradora in scenario [Rechazar bien tipo](#)
 - Administradora in scenario [Sorteo](#)
 - Administradora in scenario [Transferir plan](#)
 - bien tipo in scenario [Evaluar licitaciones](#)
 - bien tipo in scenario [Rechazar bien tipo](#)
 - bien tipo in scenario [Sorteo](#)
 - cuota mensual in scenario [Disolver un grupo](#)
 - cuota mensual in scenario [Sorteo](#)
 - Evaluar licitaciones in scenario [Evaluar licitaciones](#)
 - grupo in scenario [Armar un grupo](#)
 - grupo in scenario [Cambiar bien tipo](#)
 - grupo in scenario [Dejar plan](#)
 - grupo in scenario [Expulsar adherente](#)
 - plan in scenario [Armar un grupo](#)
 - plan in scenario [Cambiar bien tipo](#)
 - plan in scenario [Dejar plan](#)
 - plan in scenario [Disolver un grupo](#)
 - plan in scenario [Transferir plan](#)
 - planilla de adhesión in scenario [Armar un grupo](#)
 - Rechazar bien tipo in scenario [Rechazar bien tipo](#)
 - Rechazar bien tipo in scenario [Sorteo](#)
 - sorteo in scenario [Evaluar licitaciones](#)

cuota de ingreso

- Responsibilities
 - Abonada por el *solicitante* cuando entrega la *planilla de adhesión*
- Collaborations
 - Administradora in scenario [Solicitar ingreso](#)
 - Solicitante in scenario [Solicitar ingreso](#)
 - planilla de adhesión in scenario [Solicitar ingreso](#)

cuota mensual

- Responsibilities
 - Pagar *cuota* por el *adherente* o *adjudicatario*
 - La *administradora* calcula la *cuota*
- Collaborations
 - Administradora in scenario [Cambiar bien tipo adjudicado](#)
 - Administradora in scenario [Disolver un grupo](#)
 - Administradora in scenario [Pagar cuota vencida](#)
 - Administradora in scenario [Sorteo](#)
 - Adherente in scenario [Disolver un grupo](#)
 - Adherente in scenario [Pagar cuota vencida](#)
 - Adherente in scenario [Sorteo](#)
 - adjudicatario in scenario [Cambiar bien tipo adjudicado](#)
 - bien tipo in scenario [Cambiar bien tipo adjudicado](#)
 - bien tipo in scenario [Sorteo](#)
 - Cambio de bien tipo in scenario [Cambiar bien tipo adjudicado](#)
 - Comunica fehacientemente in scenario [Disolver un grupo](#)
 - Comunica fehacientemente in scenario [Sorteo](#)
 - plan in scenario [Cambiar bien tipo adjudicado](#)
 - plan in scenario [Disolver un grupo](#)
 - Rechazar bien tipo in scenario [Sorteo](#)

derecho de adjudicación

- Responsibilities
- Collaborations
 - Adherente in scenario [Adjudicación](#)
 - adjudicatario in scenario [Adjudicación](#)

disolver el grupo

- Responsibilities
- Collaborations

Evaluar licitaciones

- Responsibilities
 - No se realiza la *licitación* por:
 - No haber licitaciones entregadas
 - No se alcanza a comprar más de un *bien tipo* con el dinero recaudado
 - Los *adherentes* ganadores se transforman en *adjudicatarios*
 - La *administradora* le devuelve el cheque de la *licitación* a los *adherentes* que no pagaron
- Collaborations
 - Administradora in scenario [Evaluar licitaciones](#)
 - Administradora in scenario [Licitar](#)
 - Adherente in scenario [Evaluar licitaciones](#)
 - Adherente in scenario [Licitar](#)
 - bien tipo in scenario [Evaluar licitaciones](#)

- Comunica fehacientemente in scenario [Evaluar licitaciones](#)
- sorteo in scenario [Evaluar licitaciones](#)

Expulsar adherente

- Responsibilities
 - Posibilidad de disolución del *grupo* por falta de *adherentes*
- Collaborations

fabricante

- Responsibilities
 - El *fabricante* entrega el *bien tipo* a la *administradora*
 - El *fabricante* notifica los cambios de precio de un *bien tipo* a la *administradora*
 - El *fabricante* notifica la discontinuación de un *bien tipo* a la *administradora*
- Collaborations

grupo

- Responsibilities
 - Se puede *disolver el grupo* por concenso de los *adherentes*
 - El *grupo* puede aceptar o rechazar el cambio del *bien tipo*
- Collaborations
 - Administradora in scenario [Cambiar bien tipo](#)
 - Administradora in scenario [Dejar plan](#)
 - Administradora in scenario [Expulsar adherente](#)
 - Adherente in scenario [Armar un grupo](#)
 - Adherente in scenario [Dejar plan](#)
 - Adherente in scenario [Expulsar adherente](#)
 - Comunica fehacientemente in scenario [Armar un grupo](#)
 - Comunica fehacientemente in scenario [Cambiar bien tipo](#)
 - Comunica fehacientemente in scenario [Dejar plan](#)
 - Comunica fehacientemente in scenario [Expulsar adherente](#)
 - plan in scenario [Armar un grupo](#)
 - plan in scenario [Cambiar bien tipo](#)
 - plan in scenario [Dejar plan](#)
 - planilla de adhesión in scenario [Armar un grupo](#)

Licitar

- Responsibilities
 - La *administradora* acepta la *licitación*
 - La *administradora* rechaza la *licitación*
- Collaborations

plan

- Responsibilities
 - La *administradora* puede cambiar el *bien tipo*
 - Un *adherente* puede transferir el *plan de ahorro*
- Collaborations
 - Administradora in scenario [Asegurar adherente](#)
 - Administradora in scenario [Cambiar bien tipo](#)
 - Administradora in scenario [Cambiar bien tipo adjudicado](#)
 - Administradora in scenario [Dejar plan](#)
 - Administradora in scenario [Disolver un grupo](#)
 - Administradora in scenario [Transferir plan](#)

- Adherente in scenario [Armar un grupo](#)
- Adherente in scenario [Dejar plan](#)
- Adherente in scenario [Disolver un grupo](#)
- Adherente in scenario [Transferir plan](#)
- aseguradora in scenario [Asegurar adherente](#)
- adjudicatario in scenario [Cambiar bien tipo adjudicado](#)
- bien tipo in scenario [Cambiar bien tipo adjudicado](#)
- Cambio de bien tipo in scenario [Cambiar bien tipo adjudicado](#)
- Comunica fehacientemente in scenario [Armar un grupo](#)
- Comunica fehacientemente in scenario [Cambiar bien tipo](#)
- Comunica fehacientemente in scenario [Dejar plan](#)
- Comunica fehacientemente in scenario [Disolver un grupo](#)
- Comunica fehacientemente in scenario [Transferir plan](#)
- cuota mensual in scenario [Cambiar bien tipo adjudicado](#)
- cuota mensual in scenario [Disolver un grupo](#)
- grupo in scenario [Armar un grupo](#)
- grupo in scenario [Cambiar bien tipo](#)
- grupo in scenario [Dejar plan](#)
- planilla de adhesión in scenario [Armar un grupo](#)

planilla de adhesión

- Responsibilities
 - Usada para solicitar el ingreso a un *plan de ahorro*
- Collaborations
 - Adherente in scenario [Armar un grupo](#)
 - Administradora in scenario [Solicitar ingreso](#)
 - Comunica fehacientemente in scenario [Armar un grupo](#)
 - cuota de ingreso in scenario [Solicitar ingreso](#)
 - grupo in scenario [Armar un grupo](#)
 - plan in scenario [Armar un grupo](#)
 - Solicitante in scenario [Solicitar ingreso](#)

Rechazar bien tipo

- Responsibilities
 - Si *rechaza el bien tipo* 3 veces, el *adherente* es expulsado
- Collaborations
 - Adherente in scenario [Rechazar bien tipo](#)
 - Adherente in scenario [Sorteo](#)
 - Administradora in scenario [Rechazar bien tipo](#)
 - Administradora in scenario [Sorteo](#)
 - Comunica fehacientemente in scenario [Rechazar bien tipo](#)
 - Comunica fehacientemente in scenario [Sorteo](#)
 - cuota mensual in scenario [Sorteo](#)
 - bien tipo in scenario [Rechazar bien tipo](#)
 - bien tipo in scenario [Sorteo](#)

Renunciar al plan de ahorro

- Responsibilities
 - Puede que el *grupo* tenga que disolverse por falta de miembros
- Collaborations

seguro de bien tipo

- Responsibilities

- El *adjudicatario* paga la *cuota* del seguro del *bien tipo* a la *administradora*
- La *administradora* cobra el seguro del *bien tipo* en caso de siniestro
- Collaborations

seguro de vida

- Responsibilities
 - La *administradora* cobra la *cuota* del *seguro de vida* al *adherente*
 - La *administradora* fija el valor del *seguro de vida*
 - La *administradora* cobra el *seguro de vida* en caso de fallecimiento y entrega el *bien tipo* a los herederos
- Collaborations

Solicitante

- Responsibilities
 - El *solicitante* pide la *planilla de adhesión* a la *administradora*
 - El *solicitante* completa la *planilla de adhesión*
 - El *solicitante* entrega la *planilla de adhesión* a la *administradora*
- Collaborations
 - *Administradora* in scenario **Solicitar ingreso**
 - *cuota de ingreso* in scenario **Solicitar ingreso**
 - *planilla de adhesión* in scenario **Solicitar ingreso**

solicitud de adhesión

- Responsibilities
 - La *administradora* acepta la *solicitud de adhesión* y el *solicitante* pasa a ser *adherente*
 - Se realiza un *seguro de vida* al *adherente*
 - La *administradora* rechaza la *solicitud de adhesión* y se le reintegra el valor de la *cuota de ingreso* al *solicitante*
- Collaborations

sorteo

- Responsibilities
 - Se define el *adherente* ganador del *sorteo*
 - El *adherente* que ganó el *sorteo* puede rechazar el *bien tipo*
 - En caso de que el *adherente* que ganó el *sorteo* no se convirtió en *adjudicatario* se toma el siguiente *adherente* de la lista de sorteados
 - Si el *adherente* que ganó el *sorteo* no tiene sus *cuotas* al día o *rechaza el bien tipo* se toma el siguiente de la lista de sorteados
- Collaborations
 - *Adherente* in scenario **Evaluar licitaciones**
 - *Administradora* in scenario **Evaluar licitaciones**
 - *bien tipo* in scenario **Evaluar licitaciones**
 - *Comunica fehacientemente* in scenario **Evaluar licitaciones**
 - *Evaluar licitaciones* in scenario **Evaluar licitaciones**

Transferir plan

- Responsibilities
 - Si el *adherente* no *comunica fehacientemente* la *administradora* puede rechazar dicha transferencia
- Collaborations