

“Una heurística de derivación de LEL a Escenarios”

Alumnos

Virginia Rocco 2811/0
Juan Cesar Villalba 5953/9

Director

Gustavo Rossi

Codirector

Leandro Antonelli

TRABAJO FINAL PARA OBTENER EL GRADO DE
LICENCIATURA EN SISTEMAS
DE LA FACULTAD DE INFORMATICA,
UNIVERSIDAD NACIONAL DE LA PLATA

LA PLATA, ARGENTINA

Junio 2010

RESUMEN

“Una heurística de derivación de LEL a Escenarios”

El objetivo de este trabajo de tesis está centrado en derivar escenarios, contribuyendo y enriqueciendo las estrategias existentes. Utilizaremos la información del LEL que aún no ha sido tomada en cuenta, con el propósito de generar escenarios más completos que describan mejor los requerimientos funcionales.

Se construye además una aplicación Web que permitirá editar, registrar y gestionar el LEL y la derivación de escenarios sobre la base de las heurísticas tratadas en la presente tesis.

Agradecimientos

*Especialmente a Leandro por guiarnos en el desarrollo de esta tesis.
Virginia y Juan*

*A mi familia, a mis amigos de siempre y de ahora.
Virginia*

*Especialmente a mi familia y amigos. Y a todos los compañeros con los que he
compartido momentos en el transcurso de la carrera.
Juan*

<i>Índice de Figuras</i>	5
<i>Índice de Tablas</i>	6
<i>Capítulo 1 – Introducción</i>	9
1.1 Prefacio.....	9
1.2 Ingeniería de software	10
1.3 Ingeniería de requerimientos	13
1.4 Model Driven Development.....	15
1.5 Motivación.....	16
1.6 Objetivo	17
1.7 Desarrollos.....	17
1.8 Organización del documento	17
<i>Capítulo 2 – Léxico extendido del Lenguaje</i>	18
2.1 Introducción.....	18
2.2 Objetivo	19
2.3 Características.....	20
2.4 Proceso de construcción	21
<i>Capítulo 3 – Escenarios</i>	26
3.1 Introducción.....	26
3.2 Objetivos.....	27
3.3 Características.....	27
3.4 Proceso de construcción	29
<i>Capítulo 4 – Estrategia de derivación de escenarios enriquecida con estados</i>	34
4.1 Introducción.....	34
4.2 Consideraciones para aplicar la metodología propuesta.....	34
4.3 Etapas generales	35
4.4 Particularidades	42
<i>Capítulo 5 – Caso de estudio</i>	45
5.1 Resumen del caso de estudio	45
5.2 LEL.....	46
5.3 Derivación de escenarios	57
5.4 Aplicación de la estrategia propuesta	64
<i>Capítulo 6 – Herramienta para implementar la estrategia de derivación</i>	69
6.1 Motivación.....	69
6.2 Tecnologías utilizadas	69
6.3 Implementación	72
6.4 Manual del usuario	72
<i>Capítulo 7 – Conclusiones</i>	80
7.1 Trabajos relacionados	80
7.2 Contribución del trabajo	82
7.3 Trabajos futuros.....	84
<i>Capítulo 8 – Referencias</i>	85

Índice de Figuras

Figura 1 - Etapas para la construcción del LEL	22
Figura 2 - Diagrama Entidad-Relación para el modelo de Escenarios.....	29
Figura 3 - Etapas para la construcción de escenarios	30
Figura 4 - Workflow de Inicial a Ingresado	39
Figura 5 - Workflow de Ingresar a Anular y de Autorizar a Anular	43
Figura 6 - Posible secuencia de estados que puede seguir un requerimiento	45
Figura 7 - Flujo MVC utilizado por Cake	71
Figura 8 - Diagrama Entidad Relación de la aplicación desarrollada	72
Figura 9 - Herramienta de derivación - Pantalla inicial.....	73
Figura 10 - Herramienta de derivación - Formulario de creación de LEL.....	73
Figura 11 - Herramienta de derivación - Visualización del LEL	73
Figura 12 - Herramienta de derivación - Formulario de creación de símbolos.....	74
Figura 13 - Herramienta de derivación - Símbolos disponibles	74
Figura 14 - Herramienta de derivación - Creación de un link.....	75
Figura 15 - Herramienta de derivación - Visualización de los símbolos.....	75
Figura 16 - Herramienta de derivación - Edición de un símbolo	76
Figura 17 - Herramienta de derivación - Visualización de un símbolo.....	77
Figura 18 - Herramienta de derivación - Vista de símbolos de un LEL.....	78
Figura 19 - Herramienta de derivación - Derivación de escenarios	78
Figura 20 - Herramienta de derivación - Escenario derivado.....	79
Figura 21 - Herramienta de derivación - Vista detallada de un escenario.....	79

Índice de Tablas

Tabla 1 - Objetivo - Consecuencia del LEL	20
Tabla 2 - Meta-Modelo del LEL.....	21
Tabla 3 - Etapas – Objetivos del LEL	25
Tabla 4 – Esquema para la descripción de Escenarios	28
Tabla 5 - LEL Solicitante	35
Tabla 6 - LEL Administradora	36
Tabla 7 - LEL Planilla de Adhesión	36
Tabla 8 - LEL Cuota de ingreso	36
Tabla 9 - LEL Solicitar Ingresar	36
Tabla 10 - LEL Ingresado.....	36
Tabla 11 - LEL Solicitar Ingresar - Estrategia de derivación.....	37
Tabla 12 - Escenario Solicitar Ingresar	38
Tabla 13 - LEL Solicitar Ingresar - Estrategia de derivación.....	38
Tabla 14 - LEL Ingresar - Estrategia de derivación	39
Tabla 15 - LEL Iniciar – Estrategia de derivación	39
Tabla 16 - Escenario Solicitar ingresar- Paso 1.....	40
Tabla 17 - LEL Ingresar – Estrategia de derivación.....	40
Tabla 18 - Escenario Solicitar ingresar - Paso 2.....	40
Tabla 19 - Escenario Solicitar ingresar - Derivado sin utilizar símbolos de tipo estado	41
Tabla 20 - Escenario Solicitar ingresar - Derivado utilizando símbolos de tipo estado.	41
Tabla 21 - LEL Solicitar iniciar.....	42
Tabla 22 - LEL Iniciar	42
Tabla 23 - Escenario Solicitar iniciar	43
Tabla 24 - LEL Solicitar Anular.....	43
Tabla 25 - LEL Ingresar	44
Tabla 26 - LEL Autorizar	44
Tabla 27 - LEL Solicitar Anular.....	44
Tabla 28 - LEL Administrador	47
Tabla 29 - LEL Project Leader	48
Tabla 30 - LEL Coordinador	48
Tabla 31 - LEL Miembro del Proyecto	48
Tabla 32 - LEL Miembro de Equipo	49
Tabla 33 - LEL Responsable	49
Tabla 34 - LEL Secuencia de Estados	49
Tabla 35 - LEL Rol.....	50
Tabla 36 - LEL Sistema.....	50
Tabla 37 - LEL Usuario del Sistema	50
Tabla 38 - LEL Requerimiento.....	50
Tabla 39 - LEL Proyecto	51
Tabla 40 - LEL Tipo de Requerimiento	51
Tabla 41 - LEL Prioridad.....	51
Tabla 42 - LEL Propiedad	51
Tabla 43 - LEL Equipo.....	52
Tabla 44 - LEL Estado.....	52
Tabla 45 - LEL Estado.....	52
Tabla 46 - LEL Desarrollo.....	52
Tabla 47 - LEL Validación.....	53

Tabla 48 - LEL Terminado	53
Tabla 49 - LEL Crear Proyecto	53
Tabla 50 - LEL Cambiar un Project Leader de un proyecto	53
Tabla 51 - LEL Agregar Equipo.....	54
Tabla 52 - LEL Agregar Miembros.....	54
Tabla 53 - LEL Crear Requerimiento.....	54
Tabla 54 - LEL Cambiar el estado de un requerimiento que se encuentra Creado	54
Tabla 55 - LEL Cambiar el estado de un requerimiento en Desarrollo.....	55
Tabla 56 - LEL Cambiar el estado de un requerimiento en Validación.....	55
Tabla 57 - LEL Cambiar el estado de un requerimiento en Terminado	55
Tabla 58 - LEL Agregar Usuario al sistema.....	55
Tabla 59 - LEL Asignar a un usuario como miembro del proyecto	56
Tabla 60 - LEL Crear un Tipo de Requerimiento	56
Tabla 61 - LEL Configurar la secuencia de estados.....	56
Tabla 62 - LEL Reasignar Requerimiento.....	56
Tabla 63 - LEL Agregar Rol.....	56
Tabla 64 - LEL Asignar un Rol a un miembro del proyecto.....	57
Tabla 65 - LEL Configurar Sistema	57
Tabla 66 - Escenario Crear Proyecto.....	58
Tabla 67 - Escenario Crear equipo para un proyecto	58
Tabla 68 - Escenario Eliminar equipo de un proyecto	59
Tabla 69 - Escenario Crear nuevos requerimientos.....	59
Tabla 70 - Escenario Configurar secuencia de estados para cada tipo de requerimiento.....	59
Tabla 71 - Escenario Definir propiedades para un tipo de requerimiento.....	59
Tabla 72 - Escenario Reasignar un requerimiento.....	60
Tabla 73 - Escenario Cambiar el estado de un requerimiento Creado	60
Tabla 74 - Escenario Cambiar el estado de un requerimiento en Desarrollo	60
Tabla 75 - Escenario Cambiar el estado de un requerimiento en Validación.....	60
Tabla 76 - Escenario Asignar miembros de equipo.....	61
Tabla 77 - Escenario Asignar a miembro de proyecto como coordinador de equipo	61
Tabla 78 - Escenario Asignar miembro de equipo como responsable de requerimiento	61
Tabla 79 - Escenario Listar los requerimientos mediante filtros.....	61
Tabla 80 - Escenario Listar los requerimientos asignados	61
Tabla 81 - Escenario Configurar el Sistema.....	62
Tabla 82 - Escenario Agregar un Project Leader a un proyecto.....	62
Tabla 83 - Escenario Cambiar un Project Leader de un proyecto	62
Tabla 84 - Escenario Crear los tipos de requerimientos.....	62
Tabla 85 - Escenario Configurar tipos de requerimientos para cada proyecto.....	63
Tabla 86 - Escenario Crear los estados.....	63
Tabla 87 - Escenario Configurar los estados soportados para cada proyecto	63
Tabla 88 - Escenario Agregar usuarios.....	63
Tabla 89 - Escenario Asignar a un usuario como miembro de un proyecto.....	63
Tabla 90 - Escenario Agregar roles	64
Tabla 91 - Escenario Asignar roles a miembros del proyecto.....	64
Tabla 92 - LEL Cambiar el estado de un requerimiento Creado.....	64
Tabla 93 - Escenario Cambiar el estado Creado.....	64
Tabla 94 - LEL Creado	65
Tabla 95 - Escenario Cambiar el estado de un requerimiento Creado	65
Tabla 96 - LEL Cambiar el estado de un requerimiento Desarrollo	66
Tabla 97 - LEL Desarrollo.....	66

Tabla 98 - Escenario Cambiar el estado de un requerimiento Desarrollo	66
Tabla 99 - LEL Cambiar el estado de un requerimiento Validación.....	67
Tabla 100 - LEL Validación.....	67
Tabla 101 - Escenario Cambiar el estado de un requerimiento Validación	67
Tabla 102 - LEL Cambiar el estado de un requerimiento Terminado.....	68
Tabla 103 - LEL Terminado.....	68
Tabla 104 - Escenario Cambiar el estado de un requerimiento Terminado	68

Capítulo 1 – Introducción

En este primer capítulo se plantean la motivación y los objetivos de esta tesis, para terminar con una explicación general de la distribución de los temas tratados en cada capítulo.

1.1 Prefacio

La Ingeniería de Requerimientos es el primer paso que se da para construir el software que el cliente desea. El objetivo de la Ingeniería de Requerimientos es producir un modelo de los requerimientos que se convierte en un compromiso entre los desarrolladores y los clientes.

Los modelos son útiles para resaltar los aspectos importantes de la realidad representada, dejando de lado los elementos que puedan distraer la atención de aquello en lo que debemos enfocarnos, mostrando de forma clara los principios fundamentales que rigen al dominio particular por sobre los detalles tecnológicos que pueda requerir el producto final tales como lenguajes, herramientas de comunicación, etc.

Tener modelos como producto del proceso de desarrollo ayuda a reducir el esfuerzo invertido en el retrabajo ocasionado por las modificaciones al sistema [Lientz 81]. Los ahorros en tiempo invertidos en conocer y luego modificar el código pueden ser sustanciales [TMC 04].

En la actualidad hay una gran variedad de metodologías y herramientas con el objetivo de capturar requerimientos para la obtención de esquemas conceptuales y establecer una definición sin ambigüedades de lo que se quiere producir. Una de estas nuevas metodologías es la arquitectura Model-Driven Architecture (MDA) la cual es una iniciativa de Object Management Group (OMG) que propone una nueva manera de considerar el desarrollo y mantenimiento de sistemas de información, usando modelos como los artefactos esenciales del proceso de desarrollo de software. Su propuesta es elevar el nivel de abstracción en el que se desarrollan sistemas complejos separando la especificación de la funcionalidad de su implementación en una plataforma tecnológica específica [Bézivin 08].

MDA promueve el uso de modelos y transformaciones de modelos para el desarrollo de sistemas de software. OMG propone y promueve el uso de diversos lenguajes relacionados con la creación y gestión de modelos (UML, MOF, CWM y QVT) como mecanismos básicos para soportar esta estrategia.

Model Driven Development (MDD) es una aproximación al desarrollo de software basado en el modelado del sistema software y su generación a partir de los modelos. Al ser únicamente una aproximación, sólo proporciona una estrategia general a seguir en el desarrollo de software, pero no define ni técnicas a utilizar, ni fases del proceso, ni ningún tipo de guía metodológica. MDA promueve el MDD y agrupa varios lenguajes que pueden usarse para seguir este enfoque. MDA posee el valor añadido de proporcionar lenguajes con los que definir métodos que sigan MDD. Por lo tanto, MDA

tampoco define técnicas, etapas, artefactos, etc. MDA sólo proporciona la infraestructura tecnológica y conceptual con la que construir estos métodos MDD.

La necesidad de asegurar una buena comprensión entre los ingenieros de requerimientos y los clientes impulsó el desarrollo de métodos que permiten la colaboración entre todos los participantes del proceso de definición de requerimientos. Los ingenieros de requerimientos deben comprender, modelar y analizar el macrosistema donde el software correrá y los usuarios deben confirmar que la visión de los ingenieros es correcta. LEL y escenarios [Leite 97] son dos herramientas adecuadas para esta tarea dado que utilizan el lenguaje natural y se adecuan a las características de los seres humanos [Woods 97]. Es por esto que para la definición de métodos que sigan MDD en esta tesis se eligió usar LEL y Escenarios.

El LEL es un modelo contextual que permite capturar el lenguaje de un dominio tan importante para poder entender el dominio en sí. Los aspectos dinámicos del modelo del contexto se pueden capturar a través de los escenarios. Un escenario es una descripción parcial del comportamiento de la aplicación en un momento específico. Los escenarios además pueden utilizarse para estimar el tamaño funcional de una aplicación [Bertolami 07], y por otro lado, tienen la posibilidad de identificar objetivos o metas generales del proyecto [Thomas 05].

1.2 Ingeniería de software

El software ha evolucionado durante las últimas cinco décadas aunque no al mismo ritmo que el hardware. Durante las tres primeras décadas de la informática (1950-70) el objetivo era el desarrollo de hardware: reducir el costo de procesamiento y almacenamiento de datos. Este objetivo se alcanzó plenamente en la década de los 80. A partir de la década de los 70 se empieza a dedicar esfuerzo a la tecnología para el desarrollo de software. Hoy, el principal objetivo es mejorar tanto la calidad de los productos de software como la calidad de los procesos de desarrollo y mantenimiento.

La crisis del software se refiere a la dificultad de escribir programas libres de defectos, fácilmente comprensibles, y que sean verificables. Las causas son, entre otras, la complejidad que supone la tarea de programar, y los cambios a los que se tiene que ver sometido un programa para ser continuamente adaptado a las necesidades de los usuarios.

Entre otras dificultades el software desarrollado no es capaz de utilizar y explotar el potencial de un hardware tan sofisticado, la capacidad de mantenimiento está afectada por un mal diseño y un uso de recursos inadecuado. La demanda del software es superior a la capacidad de desarrollo.

Además, no existen todavía herramientas que permitan estimar de una manera exacta, antes de comenzar el proyecto, cuál es el esfuerzo que se necesitará para desarrollar un sistema. Este hecho provoca que la mayoría de las veces no sea posible estimar cuánto tiempo llevará un proyecto, ni cuánto personal será necesario. Cuando se fijan plazos normalmente no se cumplen por este hecho.

El Concepto de Ingeniería de Software se acuñó en 1968, en una Conferencia de la OTAN en Alemania, con la intención de que mediante el uso y establecimiento de filosofías y paradigmas de disciplinas de la ingeniería se resolviera la crisis del software.

Existen varias definiciones sobre Ingeniería de software:

- Es la aplicación práctica del conocimiento científico en el diseño y construcción de programas para ordenadores y la documentación asociada requerida para desarrollarlos, operarlos y mantenerlos [Boehm 76].
- El enfoque sistemático para el desarrollo, operación, mantenimiento y eliminación de software [IEEE 83].
- El establecimiento y uso de principios de ingeniería robustos, orientados a obtener software económico que sea fiable y funcione de manera eficiente sobre máquinas reales [Naur 69].
- Disciplina tecnológica y administrativa dedicada a la producción sistemática y mantenimiento de productos de programación que son desarrollados y modificados dentro de unos plazos y costes previamente establecidos. [Fairley 87].

En términos generales, la Ingeniería de Software es la rama de la ingeniería en sistemas vinculada con el desarrollo de grandes y complejos programas de software. Se centra en los servicios proporcionados por tales programas y sus limitaciones, objetivos del mundo real, y la precisa especificación de la estructura y comportamiento del sistema y la implementación de estas especificaciones. Contempla las actividades requeridas para adquirir la garantía de que las especificaciones del mundo real y los objetivos sean cumplidos y la evolución de estos sistemas en el tiempo. También se refiere a los procesos, métodos y herramientas para el desarrollo de los sistemas de software de una forma económica y en tiempo, maximizando la calidad, productividad y minimizando los riesgos.

La ingeniería de software abarca un conjunto de tres elementos claves [Pressman 98]: métodos, herramientas y procedimientos. Los métodos indican “cómo” construir técnicamente el software y abarcan un amplio espectro de tareas que incluyen: planificación y estimación de proyectos, análisis de los requerimientos del sistema y del software, diseño de estructuras de datos, arquitectura de programas y procedimientos algorítmicos, codificación, prueba y mantenimiento. Las herramientas suministran un soporte automático o semiautomático para los métodos mencionados anteriormente. Los procedimientos de la ingeniería de software son los que permiten relacionar los métodos y las herramientas para facilitar un desarrollo racional y oportuno del software.

El proceso de desarrollo de software "es aquel en que las necesidades del usuario son traducidas en requerimientos de software, estos requerimientos transformados en diseño y el diseño implementado en código, el código es probado, documentado y certificado para su uso operativo". Concretamente "define quién está haciendo qué, cuándo hacerlo y cómo alcanzar un cierto objetivo" [Jacobson 98].

Un modelo de ciclo de vida del software representa todas las actividades y productos de trabajo necesarios para desarrollar un sistema de software. A menudo, se dice que los

distintos componentes de software deben pasar por distintas fases o etapas durante su ciclo de vida.

Algunas de las características de tales modelos son:

- Describir las fases principales de desarrollo de software.
- Ayudar a administrar el progreso del desarrollo.
- Proveer un espacio de trabajo para la definición de un detallado proceso de desarrollo de software.

Cada una de las fases definidas puede ser abordada y resuelta de múltiples maneras, con distintas herramientas y utilizando distintas técnicas. Es necesario saber cuándo podemos dar por concluida una fase, quién debe realizarla, qué tareas preceden o anteceden a una dada, qué documentación se utilizará para llevar a cabo esa fase.

El ciclo de vida básico de un software consta de los siguientes procedimientos:

- **Definición de objetivos:** definir el resultado del proyecto y su papel en la estrategia global.
- **Análisis de los requerimientos y su viabilidad:** recopilar, examinar y formular los requisitos del cliente y examinar cualquier restricción que se pueda aplicar.
- **Diseño general:** requisitos generales de la arquitectura de la aplicación.
- **Diseño en detalle:** definición precisa de cada subconjunto de la aplicación.
- **Programación:** es la implementación de un lenguaje de programación para crear las funciones definidas durante la etapa de diseño.
- **Prueba de unidad:** prueba individual de cada subconjunto de la aplicación para garantizar que se implementaron de acuerdo con las especificaciones.
- **Integración:** para garantizar que los diferentes módulos se integren con la aplicación. Éste es el propósito de la *prueba de integración* que está cuidadosamente documentada.
- **Prueba beta** (o *validación*), para garantizar que el software cumple con las especificaciones originales.
- **Documentación:** sirve para documentar información necesaria para los usuarios del software y para desarrollos futuros.
- **Implementación**
- **Mantenimiento:** para todos los procedimientos correctivos (mantenimiento correctivo) y las actualizaciones secundarias del software (mantenimiento continuo).

El orden y la presencia de cada una de estas fases en el ciclo de vida de una aplicación dependen del tipo de modelo de ciclo de vida acordado entre el cliente y el equipo de desarrolladores.

En términos generales la ingeniería del software es una disciplina que ha evolucionado durante las tres últimas décadas con el objetivo de dar una solución a la crisis que ha venido padeciendo el software desde sus comienzos en la década de los 60. La ingeniería del software adopta el “enfoque sistemático” para la producción. La ingeniería del software aplica métodos, herramientas y procedimientos para el desarrollo, mantenimiento y producción del software.

1.3 Ingeniería de requerimientos

El proceso de recopilar, analizar y verificar las necesidades del cliente para un sistema de software es llamado Ingeniería de Requerimientos. La meta de la ingeniería de requerimientos es entregar una especificación de requerimientos de software correcta y completa. La ingeniería de requerimientos apunta a mejorar la forma en que comprendemos y definimos sistemas de software complejos.

Los Requerimientos fueron definidos por la IEEE como [IEEE90]:

1. Condición o capacidad requerida por el usuario para resolver un problema o alcanzar un objetivo.
2. Condición o capacidad que debe satisfacer o poseer un sistema o una componente de un sistema para satisfacer un contrato, un Standard, una especificación u otro documento formalmente impuesto.
3. Representación documentada de una condición o capacidad como en 1 o 2.

Un requerimiento es un atributo necesario dentro de un sistema, que puede representar una capacidad, una característica o un factor de calidad del sistema de tal manera que le sea útil a los clientes o a los usuarios finales. A nivel general los requerimientos pueden clasificarse como requerimientos indicados o reales. Los indicados son los solicitados por el usuario al comienzo del proyecto, en cambio los reales son los que reflejan la satisfacción de las necesidades del usuario en un sistema en particular.

Los requerimientos son el punto de acuerdo entre el cliente y el proyecto de desarrollo de software, este entendimiento es necesario para poder construir un software que satisfaga las necesidades del cliente. Entre los métodos conocidos para el proceso del establecimiento de requerimientos de un sistema de software se puede citar a Pressman [Pressman 98], en donde el proceso de análisis de requerimientos del software lo identifica en cinco etapas fundamentales:

1. *Reconocimiento del problema.* Se deben estudiar las especificaciones del sistema y el plan del proyecto del software. Se necesita comprender el software dentro del contexto del sistema. El analista debe establecer un canal adecuado de comunicación con el equipo de trabajo involucrado en el proyecto. En esta etapa la función primordial del analista en todo momento es reconocer los elementos del problema tal y como los percibe el usuario.
2. *Evaluación y síntesis.* En esta etapa el analista debe centrarse en el flujo y estructura de la información, definir las funciones del software, determinar los factores que afectan el desarrollo de nuestro sistema, establecer las características de la interfaz del sistema y descubrir las restricciones del diseño.
3. *Modelización.* Durante la evaluación y síntesis de la solución, se crean modelos del sistema que servirán al analista para comprender mejor el proceso funcional, operativo y de contenido de la información. El modelo servirá de pilar para el diseño del software y como base para la creación de una especificación del software.

4. *Especificación.* Las tareas asociadas con la especificación intentan proporcionar una representación del software. Esto más adelante permitirá llegar a determinar si se ha llegado a comprender el software, en los casos que se lleguen a modelar se pueden dejar plasmados manuales.
5. *Revisión.* Una vez que se han descrito la información básica, se especifican los criterios de validación que han de servir para demostrar que se ha llegado a un buen entendimiento de la forma de implementar con éxito el software. La documentación del análisis de requerimientos y manuales, permitirán una revisión por parte del cliente, la cual posiblemente traerá consigo modificaciones en las funciones del sistema por lo que deberán revisarse el plan de desarrollo y las estimaciones previstas inicialmente.

Otro método para la obtención de los requerimientos es el propuesto por Loucopoulos [Loucopoulos 95], en el que se plantea que en esta fase hay tres aspectos fundamentales:

- Comprender el problema
- Describir formalmente el problema
- Obtener un acuerdo sobre la naturaleza del problema

Entonces el proceso para obtener los requerimientos del problema se reduce a tres etapas:

- *Elicitación de requerimientos.* La elicitación de requerimientos se define como el proceso de la adquisición de todo el conocimiento relevante para producir un modelo de requerimientos del dominio de un problema.
- *Especificación.* Una especificación puede ser vista como un contrato entre usuarios y desarrolladores de software, que define el comportamiento funcional deseado del artefacto de software (y otras propiedades de éste, tales como performance, confiabilidad, etc.), sin mostrar como será alcanzada tal funcionalidad.
- *La validación* es el proceso que certifica que el modelo de los requerimientos es consistente con las intenciones de los clientes y los usuarios. Esta es una visión más general que el concepto común de validación, pues se produce en paralelo con la elicitación y la especificación, tratando de asegurar que tanto las ideas como los conceptos presentados en una descripción se identifican y explican con claridad. La validación no sólo se aplica al modelo final de los requerimientos, sino también a los modelos intermedios.

Debido a que uno de los objetivos de esta etapa del proceso de desarrollo de software es aumentar el conocimiento del dominio del problema, la comunidad de Ingeniería de Software ha desarrollado diversas estrategias para elicitación y especificación de los fenómenos propios de cada Universo de Discurso. Algunos autores proponen la utilización de aproximaciones basadas en el lenguaje natural; otros se inclinan por los lenguajes artificiales y las representaciones. Unos pocos recomiendan la construcción de un vocabulario que capture la jerga usada por los expertos del dominio. Otros adhieren al uso de escenarios o casos de uso para describir el comportamiento del macrosistema.

En términos generales podemos decir que el proceso de obtención de requerimientos debe lidiar con diferentes puntos de vista, y usar una combinación de métodos, herramientas y personal. El producto de este proceso es un modelo, que produce un documento. Este proceso sucede en un contexto previamente definido que llamamos el Universo de Discurso (UdeD), en el cual el software deberá ser desarrollado y operado. El UdeD incluye todas las fuentes de información y todas las personas relacionadas al software, que son también conocidas como los actores de ese universo [Leite97].

1.4 Model Driven Development

Cuando hablamos de software es muy útil abstraer los detalles tecnológicos de implementación y manejarnos con conceptos de dominio de la forma mas directa posible, para así tratar mejor con la complejidad ocultando lo irrelevante o de pequeño efecto, un sistema o situación compleja puede ser reducida en algo mas comprensible.

MDA fue lanzado por el Object Management Group (OMG) en el 2000 para proponer una nueva manera de considerar el desarrollo y mantenimiento de sistemas de información, usando modelos como los artefactos esenciales del proceso de desarrollo de software.

En la aproximación MDA, los modelos dejan de ser utilizados únicamente para la documentación o como guía para la implementación, convirtiéndose en el artefacto principal del proceso de desarrollo. MDA para incrementar el nivel de abstracción define lenguajes específicos de dominio cuyos conceptos reflejan los conceptos del problema de dominio mientras minimiza aspectos relacionados a la tecnología de implementación subyacente de esta manera resultan mas comprensibles por los usuarios y stakeholders.

Para facilitar la comunicación y la comprensión, tales lenguajes usan formas sintácticas específicas de dominio. A menudo usando representaciones no textuales tal como gráficos y tablas, las cuales son más legibles para dar a conocer los conceptos de dominio en comparación con el texto.

En particular MDA diferencia entre modelos independientes de plataforma y modelos específicos de plataforma. Así, la funcionalidad básica del sistema puede ser separada de su implementación final; la lógica de negocios puede ser separada de la plataforma tecnológica subyacente, etc.

Las transformaciones entre modelos permiten la implementación automatizada de un sistema desde los diferentes modelos definidos o ser reconstruidos desde código legacy para el propósito de modernización de software o migración. Además MDA permite definir modelos del sistema focalizando cada uno en un específico concern, en el nivel de abstracción correcto.

La automatización es el método más efectivo para aumentar la productividad y calidad. El software, por supuesto es un excelente medio para explotar automatización ya que la computadora es en muchas maneras la máquina ideal para construir máquinas complejas. En el caso de Model Driven Development (MDD) la idea es utilizar computadoras para automatizar.

En términos generales MDD da el potencial para proveer mejoras significativas en el desarrollo de software. Es basado en los principios de alto nivel de abstracción y alto nivel de automatización y estandarización. Permitiendo a los expertos de dominio formalmente especificar su conocimiento, y a las personas de tecnología definir como es implementado (usando transformaciones).

1.5 Motivación

Es de real importancia que la construcción de productos de software esté basada en modelos, y que estos estén enfocados en el dominio y la lógica del mismo, separando a estos de las especificaciones tecnológicas del sistema. Es esta última característica la que asegura la producción de productos de software más flexibles, o sea son más fáciles de mantener.

La abstracción realizada en el modelo se ha utilizado como base para una mejor comunicación entre los participantes del problema, así como para una mejor comprensión del mismo. Sin embargo, más allá del modelo en sí, lo que realmente interesa a todas las ramas de la ingeniería es la construcción de métodos, herramientas y técnicas que sirvan de base para la construcción y transformación de modelos. Ya que las transformaciones proporcionan un medio para definir la implementación automática de modelos, con la ventaja de generar una versión preliminar que luego puede ser corroborada y modificada por el ingeniero de software.

Los errores cometidos en la fase de especificación de requerimientos crecen en forma exponencial con el transcurso de las siguientes etapas de desarrollo, por lo que las debilidades de la mayoría de los métodos para la obtención de esquemas conceptuales, se reflejan en las primeras etapas del desarrollo. El principal problema derivado de estas debilidades metodológicas radica en la dificultad de determinar si el modelo conceptual refleja fiel y completamente la esencia del dominio [Insfrán 02].

Existen trabajos que han desarrollado estudios sobre LEL y Escenarios [Hadad 97], lo que permitió desarrollar heurísticas para la construcción de escenarios. Los escenarios derivados se construyen basándose exclusivamente en la información contenida en el LEL, al que se le aplican una serie de heurísticas. Sin embargo, aun existe información del LEL que no ha sido utilizada para la definición de los escenarios, por lo cual es necesario utilizar esta información y enriquecer los modelos siguientes.

El beneficio de LEL y escenarios es que se adecuan a la forma en que se organiza el conocimiento en el ser humano [Wood 97], hace que sea mucho más fácil para el experto presentar la información y para el ingeniero de requerimientos obtenerla y modelarla. Otro beneficio del LEL es que facilita la validación de los requerimientos con el usuario además de permitir mantener el mismo vocabulario durante todo el proceso de desarrollo.

1.6 Objetivo

El objetivo principal de este Trabajo de Tesis esta centrado en derivar escenarios, contribuyendo y enriqueciendo las estrategias existentes. Utilizaremos la información del LEL que aún no ha sido tomada en cuenta, con el propósito de generar escenarios mas completos que describan mejor los requerimientos funcionales.

1.7 Desarrollos

Además realizaremos la construcción de una herramienta, la cual será una aplicación Web que permitirá editar, registrar y gestionar el LEL, la derivación de escenarios sobre la base de las heurísticas desarrolladas por [Leite 95], [Hadad96] y [Hadad97] y la incorporación de la derivación propuesta en este trabajo.

1.8 Organización del documento

Este trabajo esta organizado de la siguiente manera:

En los capítulos 2 y 3 se describe cada uno de los pilares del modelo propuesto: Léxico Extendido del Lenguaje y Escenarios, respectivamente. En el capítulo 4 se describe la estrategia de derivación de escenarios propuesta. El capítulo 5 presenta el caso de estudio. En capítulo 6 se describe la funcionalidad y arquitectura de la herramienta desarrollada, que automatiza el modelo presentado; además se muestra la utilización de la herramienta. Finalmente en el capítulo 7 se realiza una comparación de la estrategia de derivación propuesta con otras estrategias, se presentan las conclusiones a esta propuesta y la experiencia adquirida, además se dejan proyectadas algunas posibles líneas de investigación futuras.

Capítulo 2 – Léxico extendido del Lenguaje

En el desarrollo de software el análisis del problema es imprescindible, para poder entender fielmente el problema y obtener en forma completa y correcta los requerimientos de los usuarios. Por esta razón es necesario realizar un modelo contextual. Para realizarlo es muy importante utilizar una técnica que sea compatible con las habilidades y características humanas. LEL y Escenarios son herramientas adecuadas para esta área.

LEL posee enfoque orientado al cliente, fue propuesto por Leite [Leite 97] y se basa en un metamodelo que contiene descripciones sobre el universo de discurso y el sistema de software que será construido dentro de ese universo de discurso. Estas descripciones son escritas en lenguaje natural siguiendo patrones determinados y relacionadas entre sí. El uso de lenguaje natural posibilita validar en todo momento con los stakeholders las especificaciones obtenidas en el análisis de requerimientos.

En este capítulo se explicarán características, conformación y creación del LEL, el cual basa su propuesta en conocer el vocabulario de la aplicación y su semántica, dejando para un siguiente paso la comprensión del problema.

2.1 Introducción

En el desarrollo de sistemas de software, es habitual que los desarrolladores sufran presiones para comprender y obtener las necesidades de los clientes. Estas presiones se incrementan cuanto más complejo es el contexto del problema forzando a veces que los desarrolladores se conviertan casi en expertos del dominio analizado.

Cuando no se logra una comprensión del contexto del problema es muy probable que se genere un conjunto de requerimientos erróneos o incompletos y por lo tanto un producto de software con alto grado de desaprobación por los clientes/usuarios y un altísimo costo de reingeniería y mantenimiento. Todo aquello que no se detecte, o resulte mal entendido en la etapa inicial provocará un gran impacto negativo en los requerimientos, propagando esta corriente negativa a lo largo de todo el proceso de desarrollo e incrementando su perjuicio cuanto más tardía sea su detección.

Es por eso que algo fundamental en el desarrollo de software es la comunicación. Es importante que los usuarios y desarrolladores utilicen un mismo lenguaje para facilitar la comunicación, comprensión, rastreo, revisión y validación de los productos resultantes en las diferentes etapas del desarrollo. Naturalmente el medio más simple de resolver ese problema surge a partir de la utilización del lenguaje natural y un enfoque orientado en esa dirección es el modelo propuesto por el LEL.

El LEL es un modelo que permite representar y documentar, con tecnología hipertextual un conjunto de símbolos que representan el lenguaje de la aplicación, sin necesidad de entender el problema. Consiste en una descripción de los términos significativos del macrosistema, acotando el lenguaje externo con el uso de símbolos definidos en el mismo LEL y a su vez, minimiza el uso de símbolos externos al lenguaje de la aplicación [Leite90].

El lenguaje de la aplicación es el lenguaje usado por los actores en todas las etapas de desarrollo y es una extensión del lenguaje natural usado normalmente. Una forma de obtener el lenguaje de aplicación es capturar los términos propios del mismo. Este lenguaje puede contener palabras y construcciones específicas del ambiente de la aplicación, o bien, expresiones del lenguaje natural con un significado distinto del corriente. Lo que se busca con el LEL es identificar estos símbolos, para su posterior definición con ayuda de los usuarios.

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].

El LEL puede verse como una especie de glosario, en cual el objetivo principal es registrar signos (palabras o frases) los cuales son característicos de 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 decir el significado del símbolo expresado en lenguaje natural. Es la descripción del signo por medio de sus propiedades intrínsecas o esenciales. Los impactos determinan las relaciones entre los signos. [Antonelli 03]

2.2 Objetivo

El LEL tiene por objetivo entender el vocabulario utilizado por los stakeholders, para lograr una mejor comunicación con éstos entendiendo los términos que utilizan para expresarse sin preocuparse por entender el problema que se debe solucionar. Esto se puede resumir en la tabla 1.

Para la validación de los diferentes productos en el proceso de ingeniería de requerimientos se requiere una gran interacción con el usuario, la que se vería facilitada con un vocabulario común entre usuario-desarrollador.

Otro objetivo de la utilización del LEL, es la posibilidad de utilizarlo como herramienta para la capacitación de los miembros del equipo de desarrollo que se incorporan en etapas avanzadas del desarrollo, basado esto en una documentación consistente de los símbolos que representan el lenguaje de la aplicación.

Objetivo	Consecuencia
Conocer el vocabulario del usuario	<ul style="list-style-type: none"> • Asegurar la comunicación • Facilitar la validación de los requerimientos • Mantener el mismo vocabulario durante todo el proceso de desarrollo
Contar con un instrumento simple de traceability	<ul style="list-style-type: none"> • Documentar consistentemente • Capacitar a nuevos miembros del equipo en la terminología empleada • Generar versiones del LEL a medida que evoluciona el proceso de desarrollo

Tabla 1 - Objetivo - Consecuencia del LEL

2.3 Características

El LEL está formado por un conjunto de símbolos que identifican el lenguaje de la aplicación. Los símbolos son, en general, las palabras o frases utilizadas por el usuario y que repite con más frecuencia. También se incluyen aquellas palabras o frases que son relevantes para el dominio del problema, más allá de su frecuencia de repetición. Los símbolos se obtienen, por ejemplo, de entrevistas, observaciones, lectura de documentos [Leite90]. La semántica de cada símbolo se representa con una o más nociones y cero o más impactos. El impacto puede no existir. La noción indica qué es el símbolo y el impacto, cómo repercute en el sistema [Leite93]. En la Tabla 2 podemos encontrar un resumen sobre como esta formado el LEL.

Al describir las nociones e impactos debemos cumplir dos condiciones o principios: uno de los principios dice que se debe maximizar el uso de símbolos en el significado de otros símbolos, lo cual acota el lenguaje externo al dominio de la aplicación, utilizando para la descripción de la noción y el impacto símbolos ya definidos en el LEL. Esto se denomina “principio de circularidad”. El otro requiere que se minimice el uso de símbolos externos al lenguaje de la aplicación. Este principio se denomina “principio del vocabulario mínimo”.

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 97]

<p>LEL = representación de los símbolos en el lenguaje del dominio de aplicación sintaxis: {Símbolo}₁^N</p> <p>Símbolo = entrada del LEL con un significado especial en el dominio de aplicación sintaxis: {Nombre}₁^N + {Noción}₁^N + {impacto}₁^N</p> <p>Nombre = identificador del símbolo. Más de uno representa sinónimos sintaxis: Palabra / Frase</p> <p>Noción = denotación del símbolo. Debe estar expresado usando referencias a otros Símbolos y usando Vocabulario Mínimo sintaxis: Sentencia</p> <p>Impacto = connotación del Símbolo. Debe estar expresado usando referencias a otros Símbolos y usando Vocabulario Mínimo sintaxis: Sentencia</p> <p>donde Sentencia es compuesta por Signos y No-Signos, estos últimos pertenecientes al Vocabulario Mínimo</p>

Tabla 2 - Meta-Modelo del LEL

2.4 Proceso de construcción

Para comenzar a construir el LEL en primer lugar debemos recolectar información del dominio. Luego en base a esta, empezamos a formar una lista de símbolos que son necesarios para entender el lenguaje del dominio. Estos símbolos se deben clasificar para poder definirlos en forma consistente. Luego de clasificarlos, se 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.

El proceso de construcción del LEL consta de 6 etapas interdependientes, que en algunos casos se desarrollan simultáneamente:

1. Entrevistas.
2. Generación de la lista de símbolos.
3. Clasificación de los símbolos.
4. Descripción de los símbolos.
5. Validación con los clientes.
6. Control del LEL.

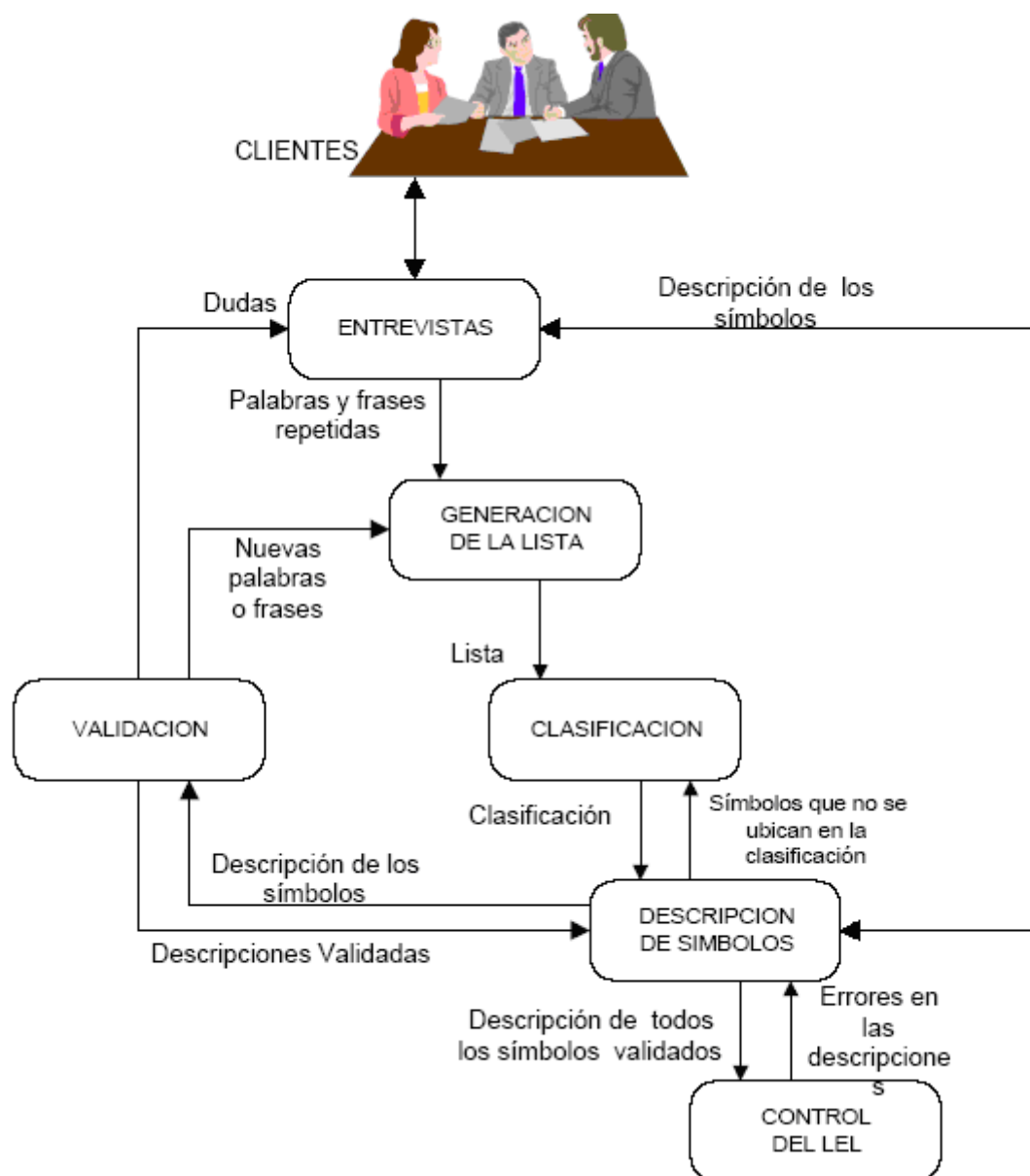


Figura 1 - Etapas para la construcción del LEL

Entrevistas.

Las entrevistas nos permitirán conocer el vocabulario que el cliente emplea en su medio ambiente. La cantidad de entrevistas necesarias depende de: [Hadad97]

- La complejidad de la aplicación.
- La experiencia del ingeniero de requerimientos en construir un LEL.
- Los conocimientos que el cliente entrevistado tiene del tema en estudio.

En primera instancia las entrevistas suelen ser reuniones con los usuarios. La cantidad de información que se puede extraer de una entrevista es grande. Por esta razón 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.

Una táctica recomendada en estas primeras reuniones, es que el entrevistador adopte una actitud pasiva, permitiendo que el entrevistado se exprese libremente con lo que se logrará que el mismo use su propio vocabulario y en forma inconsciente utiliza el principio de circularidad.

Con respecto a la elección de las personas a entrevistar, es recomendable buscar a las que toman decisiones, los jefes naturales, los responsables del proyecto y los supervisores. Hay que tener en cuenta que la idea principal es entender el lenguaje, de ningún modo se pretende determinar las funciones principales del sistema.

Generación de la lista de símbolos.

La **lista de símbolos candidatos** es el conjunto de símbolos obtenidos durante las entrevistas. Los símbolos que se deben tomar para la lista inicial están dados por palabras o frases que se utilizan con mucha frecuencia, o pueden ser símbolos que parezcan estar fuera de contexto, esto es porque el significado con el que se utilizan es distinto al tradicional, así que es necesario definirlos dentro del ambiente de la aplicación. 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. 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 [Antonelli 03].

La lista de símbolos se modifica con frecuencia después de las primeras validaciones, que es cuando se incorpora mayor cantidad de símbolos y sinónimos. Se eliminan los símbolos erróneos y se precisan los conceptos confusos. Esta lista rectificada pasa a ser la **lista definitiva** ya que las modificaciones futuras suelen ser mínimas.

En cuanto a la recolección de signos, un elemento a tener en cuenta es que en la lista de signos obtenidos, pueden existir sinónimos. Estos sinónimos se pueden ir marcando en la medida de lo posible en esta fase. Sin embargo los sinónimos quedan determinados de manera precisa en la parte de definición, donde se detectan descripciones similares para signos distintos.

Clasificación 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. La clasificación también se utiliza para asegurar la integridad y homogeneidad de las descripciones [Hadad96].

Para clasificar los símbolos utilizamos una clasificación general [Leite90]:

- **Sujeto:** Elemento activo dentro del dominio que realiza acciones utilizando objetos. El sujeto 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. Con esta categorización agrupamos los símbolos relacionados, lo que permite encontrar sinónimos.

Además, para cada categoría se debe definir la 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. Pautas para tener en cuenta en la descripción de los símbolos:

- Para un símbolo del tipo *Sujeto*, la noción define quién es el sujeto y en el impacto deben registrarse las acciones que ejecuta o recibe.
- Para un símbolo del tipo *Verbo*, la noción debe decir quién ejecuta la acción, cuándo ocurre y las actividades involucradas con dicha acción. El impacto debe identificar las situaciones que impiden la ocurrencia de la acción, qué otras acciones son disparadas en el ambiente y qué situaciones son causadas por la acción.
- Para un símbolo del tipo *Objeto*, la noción debe definir al objeto e identificar otros símbolos del mismo tipo con los cuales se relaciona. El impacto describe las acciones que pueden aplicarse a este objeto.
- Para un símbolo del tipo *Estado*, la noción define su significado y las acciones que llevan a ese estado. El impacto debe identificar otros estados y acciones que pueden ocurrir a partir de la situación específica.

Descripción de símbolos.

A partir de la lectura de la documentación y entrevistas obtenemos un conocimiento que nos permite realizar la descripción de los símbolos. Describir los símbolos es determinar su noción y su impacto.

En el proceso de generación de la lista candidata podemos hacer una descripción inicial del significado de los símbolos, debido a que las palabras o frases significativas del dominio están dentro de un contexto que el ingeniero de software está escuchando. Se debe realizar una validación con los clientes, y seguir describiendo los restantes símbolos.

Algunas reglas que nos pueden ayudar en este proceso son:

- Un símbolo puede tener una o más nociones y cero o más impactos.
- Cada noción e impacto debe ser descripto con oraciones breves y simples.
- Nociones e impacto para un símbolo pueden representar diferentes puntos de vista o pueden ser complementarios.
- Las oraciones breves y simples de las nociones e impactos deben responder a los principios de circularidad y de vocabulario mínimo.

Validación con los clientes.

Esta fase es para verificar la correctitud del LEL contra el usuario. Lo que nos permitirá arreglar o aprobar el conocimiento del vocabulario del dominio obtenido durante las entrevistas.

Puesto que a veces puede existir una gran cantidad de símbolos definidos en el LEL, quizás sea impracticable realizar una validación completa y exhaustiva. Sin embargo, es posible mantener sesiones de entrevistas estructuradas para aclarar dudas.

Las primeras validaciones permiten chequear la lista candidata y conocer el significado de los símbolos. Las siguientes validaciones son, generalmente, para chequear que el conocimiento obtenido sea el correcto y que no falte información, de no ser así se agrega o modifica el LEL.

Control del LEL.

El proceso de control lo realiza el ingeniero de requisitos por sus propios medios. Este control no tiene que ver con la correctitud de la información del LEL, sino con la estructura. Se controla que todos los símbolos cumplan con la formulación de la clasificación a la que pertenecen, que no existan sinónimos como símbolos diferentes y que la sintaxis empleada sea la misma. Se controla que el LEL terminado para esta etapa, sea consistente y homogéneo. Este control comienza en las primeras descripciones de los símbolos y es fundamental cuando participan varios entrevistadores.

Podemos ver un resumen de los objetivos de cada etapa en la tabla 3.

Etapa	Objetivo
1. Entrevistas	Conocer el vocabulario de la aplicación
2. Generación de la lista de símbolos	Obtener diferentes versiones de la lista de símbolos sin atender a las descripciones de los mismos
3. Clasificación de los símbolos	Asegurar la completitud y homogeneidad de las descripciones de los símbolos
4. Descripción de los símbolos	Establecer los contenidos de la noción e impacto de cada símbolo
5. Validación con los clientes	Rectificar o ratificar el contenido volcado en las descripciones de los símbolos
6. Control del LEL	Asegurar la consistencia y homogeneidad del LEL.

Tabla 3 - Etapas – Objetivos del LEL

Capítulo 3 – Escenarios

En este capítulo se describirán a los escenarios, los cuales permiten modelar el comportamiento de un sistema o sus aspectos dinámicos, y mostraremos su creación a partir del LEL.

3.1 Introducción

Un escenario es una descripción parcial del comportamiento de la aplicación en un momento específico. La utilización de escenarios implica identificar distintas situaciones y describir la acción a llevar a cabo. Los escenarios pueden ser vistos como historias que explican como se usa el sistema. Son principalmente útiles para agregar detalle a una descripción de requisitos. Una vez que se tiene una idea básica de las funcionalidades que un sistema debería proveer, se pueden desarrollar escenarios alrededor de esas funcionalidades.

Los escenarios permiten: conocer el problema, unificar criterios, ganar compromiso con los clientes, organizar los detalles involucrados y entrenar a nuevos participantes. Por otra parte, los usuarios finales y otros “stakeholders” del sistema encuentran más fácil relacionar las funciones provistas por un sistema con ejemplos de la vida real que con descripciones abstractas. Por ello, resulta útil desarrollar un conjunto de escenarios con el objeto de utilizarlos para modelar el comportamiento del sistema. En este caso, los usuarios finales simulan su interacción con el sistema usando el escenario. Así, pueden explicar al equipo de ingenieros de requerimientos lo que están haciendo y la información que necesitan del sistema para llevar a cabo la tarea descrita en el escenario. Además, el proceso de desarrollar un escenario, aún sin considerar la interacción del usuario, puede ayudar a entender los requisitos.

Descubrir escenarios posibles expone el rango de posibles interacciones del sistema y revela facilidades del sistema que podrían ser requeridas [Kotonya 98]. Por otra parte, los escenarios pueden ser utilizados como un medio para obtener métricas. En [Sutcliffe 01], por ejemplo, se propone un método que utiliza escenarios para medir la carga de trabajo de los agentes en términos de las tareas que tienen que ejecutar, y lograr así estimaciones de tiempo y probabilidades de falla.

La ventaja de los escenarios sobre cualquier otro método de elicitación de requerimientos, es que los escenarios guardan una gran similitud a la forma en que los seres humanos entienden y describen los problemas.

Si bien cada escenario describe una situación particular, ninguno es totalmente independiente del resto. Cada uno de ellos tiene una relación semántica con otros escenarios [Booch 91]. Es importante considerar que el nivel de detalle con el que se describen los escenarios depende de dos factores:

- el grado de importancia que el cliente le otorgue a los hechos específicos del problema;
- la fase en la que se encuentra el proceso de desarrollo.

Los escenarios pueden ser escritos de diferentes maneras, pero deberían incluir al menos, la siguiente información:

- Una descripción del estado del sistema antes de entrar al escenario
- El flujo normal de eventos en el escenario
- Excepciones al flujo normal de eventos
- Información acerca de otras actividades que podrían estar sucediendo al mismo tiempo
- Una descripción del estado del sistema después de completar el escenario

3.2 *Objetivos*

Los escenarios cumplen distintos objetivos, dependiendo de la fase en que se describen durante el ciclo de desarrollo del software. En la fase de análisis de requerimientos, los objetivos de los escenarios son:

- capturar los requerimientos;
- proveer un medio de comunicación entre los clientes y los ingenieros de software;
- contar con un instrumento de traceability.

3.3 *Características*

La metodología de Leite [Leite 95] es a la que se adhiere esta tesis, en donde se propone trabajar con documentos en lenguaje natural altamente vinculados entre sí. Entre los documentos que incorpora esta metodología, se encuentran los escenarios, cuyo principal objetivo durante la etapa de elicitación de requisitos, es comprender el problema en su totalidad. En las siguientes etapas acompaña el proceso de desarrollo del software describiendo aspectos de diseño, codificación y la representación de modificaciones a requisitos existentes y la representación de requisitos nuevos.

El modelo de escenario de la metodología de Leite es una estructura compuesta por las siguientes entidades: Título, Objetivo, Contexto, Recursos, Actores, Episodios, Excepciones y el atributo restricción. Actores y Recursos son enumeraciones. Título, Objetivo, Contexto y Excepciones son sentencias declarativas, mientras Episodios son 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 u 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. Para indicar tareas condicionales se utiliza **if then** con la semántica tradicional.

El atributo restricción es usado para caracterizar requisitos no funcionales aplicados a contexto, recursos y episodios. Un escenario puede ser interrumpido por excepciones. Cada excepción se describe como una sentencia simple que especifica la causa de la interrupción. Las excepciones plantean casos alternativos.

El modelo de escenarios provee la descripción de comportamientos con diferentes órdenes temporales. Una secuencia de episodios implica un *orden de precedencia*, pero un *orden no secuencial* requiere agrupar dos o más episodios según la sintaxis mostrada en la Tabla 4. Así, es posible expresar un orden paralelo o indistinto.

<p>Nombre: título del escenario. En el caso de un sub-escenario, el título es el mismo que la sentencia episodio (ver abajo la definición Episodio), sin las restricciones y/o excepciones.</p> <p>Sintaxis: Frase ([Actor Recurso] + Verbo + Predicado)</p> <p>Objetivo: finalidad a ser alcanzada en el contexto del problema. El escenario describe el logro del objetivo.</p> <p>Sintaxis: [Sujeto] + Verbo + Predicado</p> <p>Contexto: ubicación geográfica y temporal del escenario, y/o estado inicial del mismo.</p> <p>Sintaxis: Ubicación + Estado donde Ubicación es: Nombre donde Estado es: [Actor Recurso] + Verbo + Predicado + {Restricciones}</p> <p>Recursos: medios de soporte, dispositivos u otros elementos pasivos necesarios para estar disponibles en el escenario.</p> <p>Sintaxis: Nombre + {Restricciones}</p> <p>Actores: personas o estructuras organizacionales que tienen un rol en el escenario.</p> <p>Sintaxis: Nombre</p> <p>Episodios: conjunto de acciones que detallan el escenario y proveen su comportamiento.</p> <p>Sintaxis: <episodios> ::= <series> <series> ::= <sentencia> <series> <sentencia> ::= <sentencia secuencial> <sentencia no secuencial> <sentencia condicional> <sentencia optativa> <sentencia secuencial> ::= <sentencia episodio> <sentencia condicional> ::= Si <condición> entonces <sentencia episodio> <sentencia no secuencial> ::= # <series> # <sentencia optativa> ::= [<series>] donde <sentencia episodio> se describe: [Actor Recurso] + Verbo + Predicado + {Restricciones} + {Excepciones}</p>

Tabla 4 – Esquema para la descripción de Escenarios

La figura 2 presenta la estructura de un escenario usando el diagrama entidad-relación, donde sólo se muestran las relaciones relevantes. Se observa en dicho diagrama que un episodio puede concebirse como un escenario en sí mismo, esto posibilita la descomposición de un escenario en subescenarios.

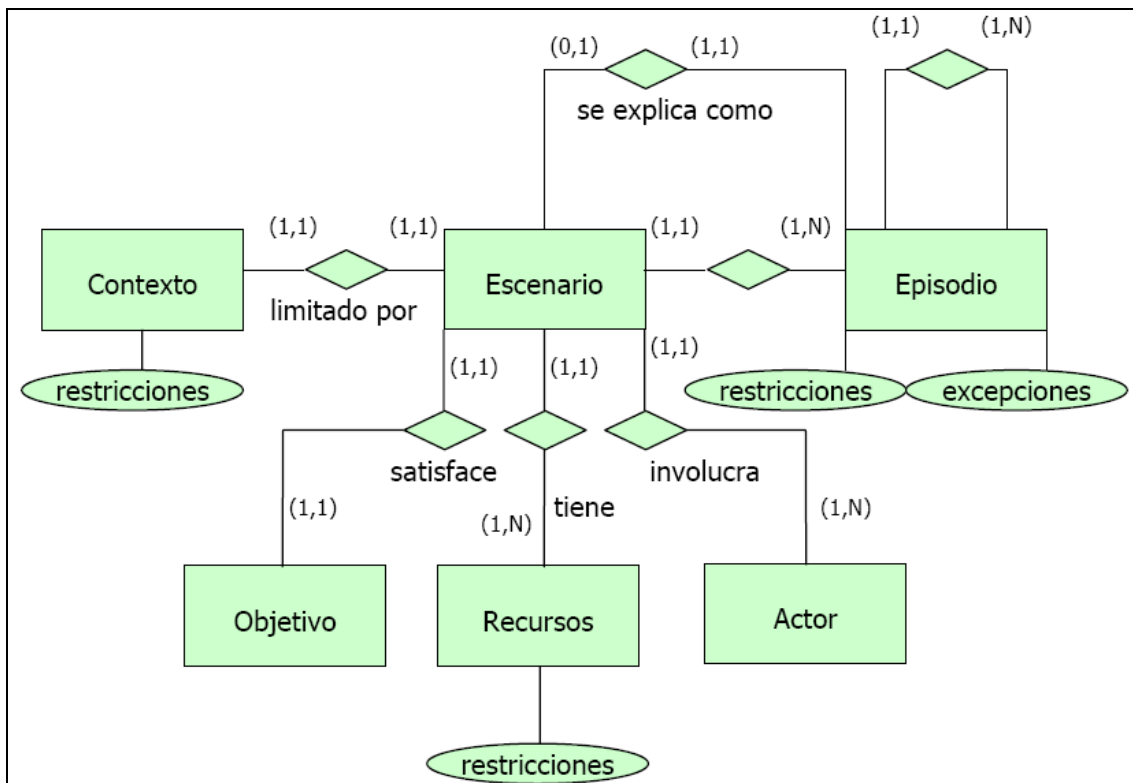


Figura 2 - Diagrama Entidad-Relación para el modelo de Escenarios

3.4 Proceso de construcción

La construcción de los escenarios se basa exclusivamente en la información contenida en el LEL, al que se le aplican una serie de heurísticas [Hadad96] y [Hadad97], para obtener una primera lista de los escenarios candidatos. Posteriormente se realizan nuevas entrevistas con los clientes, con el fin de ampliar información sobre aquellos escenarios cuya descripción resultará incompleta y/o confusa a partir de la información obtenida del LEL.

El proceso de construcción de los escenarios consta de las siguientes etapas:

1. Identificación de los actores de la aplicación;
2. Generación de la lista de escenarios candidatos, a partir de los actores principales;
3. Descripción de los escenarios candidatos, provenientes de los actores principales;
4. Ampliación de la lista de escenarios candidatos, a partir de los actores secundarios;
5. Descripción de los escenarios candidatos, provenientes de actores secundarios;

6. Revisión de los escenarios;
7. Validación de escenarios.

En la figura 3 se presentan gráficamente las etapas antes mencionadas.

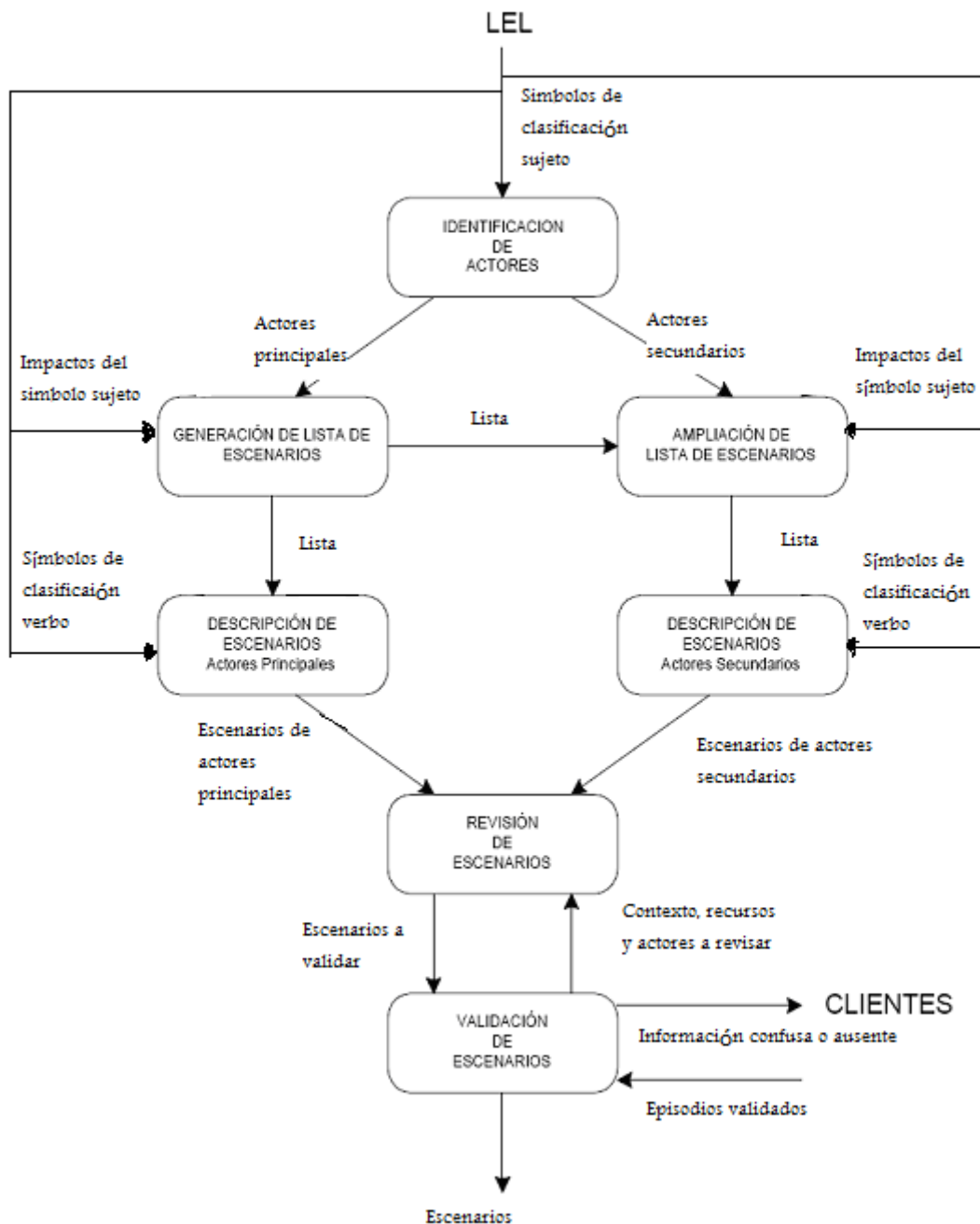


Figura 3 - Etapas para la construcción de escenarios

1. *Identificación de los actores de la aplicación.*

A partir del LEL, se identifican los símbolos que representan a los **actores** del Universo de Discurso. Estos símbolos pertenecen a la **clasificación Sujeto**, que agrupa a los individuos que interactúan en el Universo de Discurso. Se identifica cuáles son los **actores principales** y cuáles los **actores secundarios**. Para ello,

se considera que los actores principales son aquellos que realizan acciones directas sobre la aplicación, mientras que los actores secundarios son los sujetos que reciben y/o brindan información, pero no ejecutan acciones directas sobre la aplicación.

2. *Generación de la lista de Escenarios candidatos, a partir de los actores principales.*

Del símbolo del LEL correspondiente a cada actor principal se obtienen sus **impactos**. Cada impacto representa un posible escenario, que se incorpora a la lista de escenarios candidatos. El **nombre del escenario** se conforma con la acción (verbo) contenida en el impacto en tiempo infinitivo. De esta lista, se eliminan los escenarios repetidos, en el caso de que distintos actores ejecuten la misma acción.

3. *Descripción de los Escenarios candidatos, provenientes de los actores principales.*

A partir de la información de la entrada de LEL se pueden cubrir todos los atributos del escenario. Esta descripción se realiza aplicando las reglas que se detallan a continuación:

Para cada impacto que representa un posible escenario, se verifica si contiene un símbolo del LEL que pertenezca a la **clasificación Verbo**.

En caso afirmativo:

Se busca dicho símbolo en el LEL. Se define el **objetivo** del escenario basándose en el nombre del escenario, el punto de vista de la aplicación y, como ayuda adicional, la **noción** de este símbolo que representa una actividad. Se identifican los **actores** y los **recursos** que surgen de la información de este símbolo, y que pertenecen, a su vez, a la **clasificación Sujeto** y a la **clasificación Objeto** respectivamente. Se definen los **episodios** del escenario a partir de cada **impacto** del símbolo. Para la definición del **contexto**, se observa si en el símbolo del cual se obtuvo este escenario, el impacto que lo originó tenía alguna secuencia con respecto a los restantes impactos. En caso afirmativo, se describe en el contexto como precondición la ejecución previa de dichos impactos que corresponden a escenarios candidatos. También se define en el contexto la ubicación física o temporal en la realización del escenario. Se completa el ítem **dudas** con toda aquella información que parezca confusa o ausente.

Si el impacto **no** contiene un símbolo del LEL que pertenezca a la **clasificación Verbo**:

Se identifican símbolos dentro del impacto. Se deben aplicar las mismas reglas que para el caso que el posible escenario contenga un símbolo del LEL que pertenezca a la clasificación Verbo.

A medida que se describen los episodios pueden surgir las **excepciones**.

4. *Ampliación de la lista de Escenarios candidatos, a partir de los actores secundarios.*

Se aplica a los actores secundarios el procedimiento descrito en el punto 2, para obtener otros escenarios candidatos e incorporarlos a la lista.

5. *Descripción de los Escenarios candidatos, provenientes de los actores secundarios.*

Se aplica el procedimiento descrito en el punto 3 a los nuevos escenarios incorporados a la lista en el punto anterior.

6. *Revisión de los Escenarios.*

Descritos todos los escenarios candidatos, se revisan de la siguiente manera:

- Detección de escenarios candidatos como episodios simples dentro de otros escenarios. Estos escenarios se originan en impactos de los símbolos de actores que no contienen otro símbolo perteneciente a la clasificación Verbo. Por lo tanto, son escenarios cuyos episodios no se han definido en el punto 3, donde se describe a los escenarios candidatos. Se entrevista al cliente para determinar si este escenario involucra más de un episodio. En caso afirmativo, se mantiene el escenario en la lista y, en los escenarios que figura como una acción simple, se reemplaza el episodio por el nombre del escenario. Es decir, ocurre una detección de subescenarios. En caso contrario, se elimina de la lista el escenario candidato.

- Detección de escenarios candidatos como un conjunto de episodios dentro de otros escenarios. En escenarios provenientes de actores principales puede ocurrir que en la descripción de los episodios, un conjunto de éstos correspondan a un escenario proveniente de un actor secundario. Se reemplaza el conjunto de episodios por el escenario proveniente del actor secundario. Nuevamente ocurre la detección de subescenarios.

- Unificación de escenarios. Si dos o más escenarios presentan episodios comunes o poseen el mismo objetivo y el mismo contexto, estos escenarios se agrupan en un sólo escenario. De requerirse, se utiliza la forma condicional para describir episodios diferentes.

- Definición de restricciones. Durante la revisión de los escenarios, se definen las restricciones a los episodios. Estas provienen del contexto del problema y también, en función del contexto del escenario.

- Detección de escenarios que corresponden a excepciones. Cuando un escenario queda “aislado” en cuanto a que su contexto no identifica claramente qué escenarios se realizan previamente, o el escenario no mantiene ninguna relación con otro escenario, se verifica si este escenario puede o no corresponder a una excepción de un escenario o de un episodio dentro de un escenario.

Una vez finalizado este punto se obtiene la lista de escenarios a validar con los clientes.

7. *Validación de Escenarios.*

Se valida con los clientes cada uno de los escenarios incluidos en la lista definitiva, prestando especial atención al ítem dudas si existiese en el escenario. La validación de los escenarios permite detectar errores u omisiones o ampliar información en los episodios. En el caso de ampliación de información de un episodio, se puede detectar que episodios considerados simples involucran un conjunto de acciones.

- Se realizan las correcciones necesarias en cada ítem del escenario. En el caso de que se necesite agregar información a un episodio, se puede:
 - abrir el episodio en varios episodios dentro del mismo escenario, o
 - reemplazar el episodio por un nuevo escenario que involucre el conjunto de acciones detectado e incorporar el nuevo escenario a la lista.

- Se realiza una revisión del contexto, los actores y los recursos de cada escenario, en función de las correcciones introducidas en los episodios en los puntos precedentes. A partir de este punto, se obtiene la lista definitiva de escenarios y sus descripciones, correspondientes a la fase de elicitación de requerimientos.

Capítulo 4 – Estrategia de derivación de escenarios enriquecida con estados

4.1 Introducción

La derivación de los escenarios se basa exclusivamente en la información contenida en el LEL, al que se le aplican una serie de heurísticas. La intención de esta etapa es construir los escenarios aprovechando el conocimiento que el LEL nos da en la mayor medida posible. En los enfoques [Leite 95], [Hadad96] y [Hadad97] no utilizan los símbolos con la clasificación de estado para la derivación de los escenarios.

En consecuencia este trabajo propone un enriquecimiento de las estrategias de derivación actuales. Nuestro objetivo es enriquecer la construcción de los escenarios, utilizando los símbolos de tipo estado que no eran usados.

El capítulo se encuentra organizado de la siguiente manera: en la sección 4.2 se enumeran las consideraciones que se deben tomar al definir el LEL para poder aplicar la metodología propuesta. En la sección 4.3 se encuentran los pasos que se deben seguir para la identificación e inclusión de las nuevas heurísticas para la derivación de los escenarios. En la sección 4.4 daremos una serie de peculiaridades que han surgido al momento de definir la estrategia presentada.

4.2 Consideraciones para aplicar la metodología propuesta

El LEL es un modelo que permite representar y documentar un conjunto de símbolos que representan el lenguaje de la aplicación. Pero a veces no se puede utilizar la totalidad de la información para definir de manera correcta y detallada el proceso de derivación de escenarios al estar el LEL mal definido.

A continuación se señalan algunos de los aspectos que se han advertido referidos al LEL que se deben tener en cuenta al momento de definirlo para poder aplicar la metodología propuesta.

El LEL debería contar con al menos dos símbolos con la clasificación de estado para poder aplicar la estrategia que proponemos en este trabajo. Decimos que debería haber dos símbolos de tipo estado para que haya un diagrama de transición de estados. O sea para que exista al menos un estado inicial y un siguiente estado.

Con respecto a la definición de los símbolos de tipo estado, tienen que estar descriptos con una noción y uno o varios impactos [Leite 95]. Si en la definición de un impacto nombramos a otro símbolo con la clasificación de estado, este otro símbolo se debe encontrar definido entre los símbolos del LEL. Este es el principio de circularidad del LEL.

Lo que intentamos con las consideraciones descriptas en esta sección es mejorar y refinar la información con la que contamos en el LEL, es por este motivo que propusimos que se deberían tener en cuenta al momento de definir el LEL. Si el LEL no se especifica de la forma antes mencionada, se van a poder aplicar las heurísticas que se

tenían hasta ahora para la derivación de los escenarios pero no se va a poder aplicar la nueva regla propuesta en esta tesis.

4.3 Etapas generales

Esta sección se encuentra organizada de la siguiente manera: primero se muestra la estrategia de derivación de escenarios tradicional, y después se indica como se enriquece esta estrategia con los estados. Luego se muestra un ejemplo que contrasta como se obtenían los escenarios antes y con la nueva estrategia. Y finalmente se detallan los pasos de la estrategia enriquecida.

El objetivo de esta tesis fue la incorporación de una serie de nuevas reglas para la derivación de los escenarios. La identificación e inclusión de estas nuevas heurísticas para la derivación de los escenarios, permite utilizar información del LEL que hasta ahora no se estaba teniendo en cuenta al momento de realizar la derivación de los escenarios. Con esto mejoraremos la calidad de la información del LEL y refinaremos el proceso de derivación de los escenarios.

Nos proponemos encontrar un método que permita utilizar para la derivación de los escenarios los símbolos con la clasificación de estado. Basándonos en el modelo de dominio que resulta de utilizar LEL como herramienta formal para el análisis del dominio, realizaremos la construcción de los escenarios como lo explicamos en el capítulo anterior.

Presentaremos la explicación de nuestra estrategia con un ejemplo. Al final del capítulo mencionaremos las conclusiones obtenidas a partir de lo estudiado. En el capítulo siguiente desarrollaremos un caso de estudio completo.

Haremos la explicación basándonos en el ejemplo de una Agencia de autos que realiza círculos cerrados de compra, sorteo y licitación. El objetivo de la administradora es la conformación de grupos de adherentes, los cuales realizan la compra de automóviles 0km a través de un sistema de ahorro y licitación [Rivero 98] [Mauco 97]. Para simplificar el ejemplo no se muestra todo el análisis del LEL sino que se construye sólo lo que nos es útil para la explicación de nuestra estrategia. A continuación mostraremos la definición del LEL:

Símbolo	Solicitante	Categoría	Sujeto
Sinónimos			
Noción	Es el que solicita la compra de un automóvil 0km.		
Impacto:	Solicita planilla de adhesión.		

Tabla 5 - LEL Solicitante

Nombres	Administradora	Categoría	Sujeto
Sinónimos			
Noción	Es la encargada de administrar los planes para la adquisición de vehículos 0km.		
Impacto:	Ingresa a los planes de ahorro a un nuevo solicitante.		

Tabla 6 - LEL Administradora

Nombres	Planilla de adhesión	Categoría	Objeto
Sinónimos			
Noción	Formulario que el solicitante debe llenar al querer ingresar a un plan de ahorro.		
Impacto:	El solicitante completa la planilla de adhesión.		

Tabla 7 - LEL Planilla de Adhesión

Nombres	Cuota de ingreso	Categoría	Objeto
Sinónimos			
Noción	Cantidad fija con que solicitante debe pagar al ingresar al plan de ahorro		
Impacto:	El solicitante paga la cuota de ingreso al ingresar al plan de ahorro.		

Tabla 8 - LEL Cuota de ingreso

Nombres	Solicitar ingresar	Categoría	Verbo
Sinónimos			
Noción	La administradora ingresa un nuevo solicitante a los planes de ahorro		
Impacto:	El solicitante pide la planilla de adhesión.		
Impacto:	El solicitante completa dicha planilla y la entrega, abonando la cuota de ingreso.		

Tabla 9 - LEL Solicitar Ingresar

Nombres	Ingresado	Categoría	Estado
Sinónimos			
Noción	La planilla de adhesión se encuentra en estado ingresado una vez que el solicitante completa dicha planilla.		
Impacto:	Se puede pasar a autorizar el plan.		

Tabla 10 - LEL Ingresado

A continuación se muestra la derivación del escenario “Solicitar Ingresar” siguiendo con la estrategia de derivación presentada en el Capítulo 3 [Hadad96], [Hadad97]. Vemos gráficamente en la tabla 11 como se deriva el escenario:

Nombres	Solicitar ingresar	Categoría	Verbo
Sinónimos			
Noción	La administradora ingresa un nuevo solicitante a los planes de ahorro		
Impacto:	El solicitante pide la planilla de adhesión.		
Impacto:	El solicitante completa dicha planilla y la entrega, abonando la cuota de ingreso.		

Tabla 11 - LEL Solicitar Ingresar - Estrategia de derivación

Para realizar la explicación de la estrategia propuesta en esta tesis, inicialmente identificamos los actores de la aplicación, que son símbolos clasificados como Sujetos. En este ejemplo contamos con los símbolos de la tabla 5 y tabla 6.

Al generar la lista de escenarios candidatos a partir de los actores principales, contamos con un posible escenario “Solicitar ingresar”. Verificamos que ese posible escenario contiene un símbolo del LEL que pertenece a la clasificación Verbo. Lo podemos encontrar en la tabla 9.

Como se verificó que el posible escenario contiene un símbolo en el LEL con la clasificación Verbo, veamos paso a paso la derivación del escenario:

- a) Se define el objetivo del escenario basándose en el nombre del escenario, el punto de vista de la aplicación y la noción de este símbolo. En este caso para la definición del objetivo del escenario nos basamos en el nombre de nuestro escenario que es “Solicitar ingresar”.
- b) Se identifican los actores y los recursos que surgen de la información de este símbolo, y que pertenecen a la clasificación Sujeto y a la clasificación Objeto respectivamente. Para nuestro ejemplo los actores son Administradora, y Solicitante y como recursos tenemos planilla de adhesión y cuota de ingreso.
- c) Además, se definen los episodios del escenario a partir de cada impacto del símbolo “Solicitar ingresar”, este símbolo contaba con dos impactos como vimos en la tabla 11.

El escenario resultante nos queda:

Título:	Solicitar ingresar
Objetivo:	Una persona ingresa a un plan de ahorro.
Actores	Solicitante
Recursos:	Planilla de adhesión, Cuota de ingreso
Episodio 1:	El solicitante pide la planilla de adhesión.
Episodio 2:	El solicitante completa dicha planilla y la entrega, abonando la cuota de ingreso.

Tabla 12 - Escenario Solicitar Ingresar

Pero hemos observado que si el símbolo del cual se obtuvo este escenario, el impacto que le dio origen contiene en su definición un símbolo del tipo estado, la información de este símbolo que hasta ahora no era utilizada para la derivación de los escenarios nos puede ser útil para enriquecer el contexto y el objetivo.

En caso que se encuentre este símbolo con la clasificación de estado entre los símbolos definidos en el LEL, podemos describir la precondition o estado inicial del contexto del escenario basándonos en los estados anteriores al símbolo de tipo estado, y los impactos asociados a este símbolo lo podríamos usar para la definición del objetivo. De esta forma enriquecemos el modelo actual de derivación de los escenarios.

Entonces vamos a aplicar lo expuesto en el párrafo anterior al ejemplo de la Agencia de autos que realiza círculos cerrados de compra, sorteo y licitación. Como en nuestro ejemplo el símbolo del cual se obtuvo el escenario “Solicitar ingresar”, el impacto de donde se originó contiene un símbolo de tipo estado (Ingresado), podemos utilizar la información de este símbolo estado. Para describir la precondition del escenario usaremos los estados anteriores al símbolo de tipo estado. Primero veamos gráficamente como se obtiene la información:

Corresponde a un símbolo de tipo estado

Nombres	Solicitar ingresar	Categoría	Verbo
Sinónimos			
Noción	La administradora ingresa un nuevo solicitante a los planes de ahorro		
Impacto:	El solicitante pide la planilla de adhesión.		
Impacto:	El solicitante completa dicha planilla y la entrega, abonando la cuota de ingreso.		

Tabla 13 - LEL Solicitar Ingresar - Estrategia de derivación

El símbolo que originó el escenario
contiene un símbolo de tipo Estado

Nombres	Ingresado	Categoría	Estado
Sinónimos			
Noción	La planilla de adhesión se encuentra en estado ingresado una vez que el solicitante completa dicha planilla.		
Impacto:	Se puede pasar a autorizar el plan.		

Tabla 14 - LEL Ingresar - Estrategia de derivación

Lo que haremos es agregarle al escenario anterior una precondición al contexto. Basándonos en los estados anteriores al que se hace referencia en el impacto que le dio origen. En nuestro ejemplo el impacto que originó el escenario menciona al estado “Ingresar”, en nuestra precondición se va a nombrar a los estados anteriores a Ingresar, verificando el workflow de estados. En este caso se va a nombrar al estado “Inicial”, que es un estado anterior a Ingresar.

Nombres	Iniciar	Categoría	Estado
Sinónimos			
Noción	La planilla de adhesión se encuentra en estado inicial una vez que al solicitante se le entrega la planilla.		
Impacto:	Se puede pasar a ingresar la planilla de adhesión.		

Vemos que es un estado
anterior a Ingresar

Tabla 15 - LEL Iniciar – Estrategia de derivación

Se puede observar que uno de los impactos del estado Inicial es: “Se puede pasar a ingresar el plan”, el cual hace referencia al estado Ingresar, con lo que nos permite armar la transición de estados siguiente:

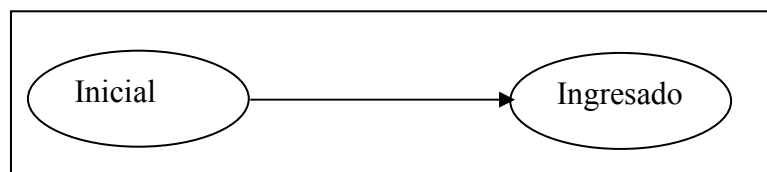


Figura 4 - Workflow de Inicial a Ingresado

Entonces la precondición para el contexto del escenario va a tener la forma: “Se debe encontrar en alguno de los siguientes estados:” “nombres de tipo estado”. En nuestro ejemplo nos va a quedar: “La planilla de adhesión se debe encontrar en alguno de los siguientes estados: inicial”. Vemos como queda el escenario resultante al utilizar esta información.

Título:	Solicitar ingresar
Objetivo:	Una persona ingresa a un plan de ahorro.
Contexto:	Precondición: La planilla de adhesión se debe encontrar en alguno de los siguientes estados: inicial.
Actores	Administradora, Solicitante
Recursos:	Planilla de adhesión Cuota de ingreso
Episodio 1:	El solicitante pide la planilla de adhesión.
Episodio 2:	El solicitante completa dicha planilla y la entrega, abonando la cuota de ingreso.

Tabla 16 - Escenario Solicitar ingresar- Paso 1

En el paso siguiente vamos a usar la información de los impactos asociados al símbolo de tipo estado. Siguiendo con nuestro ejemplo, el símbolo de tipo estado “Ingresar” contaba con un impacto que era “*Se puede pasar a autorizar el plan*”, como podemos ver en la tabla 17. Ahora lo que haremos es usar este impacto para enriquecer a la descripción del objetivo del escenario que estamos derivando. Vemos gráficamente de donde sacamos esta información:

Nombres	Ingresar	Categoría	Estado
Sinónimos			
Noción	La planilla de adhesión se encuentra en estado ingresado una vez que el solicitante completa dicha planilla		
Impacto:	Queda en condiciones de autorizarse el plan.		

Esta información la usaremos para enriquecer el objetivo

Tabla 17 - LEL Ingresar – Estrategia de derivación

Vemos como queda el escenario resultante al usar estos datos:

Título:	Solicitar ingresar
Objetivo:	Una persona ingresa a un plan de ahorro. <i>Se puede pasar a autorizar el plan.</i>
Contexto:	La planilla de adhesión se debe encontrar en alguno de los siguientes estados: inicial.
Actores	Administradora, Solicitante
Recursos:	Planilla de adhesión Cuota de ingreso
Episodio 1:	El solicitante pide la planilla de adhesión.
Episodio 2:	El solicitante completa dicha planilla y la entrega, abonando la cuota de ingreso.

Tabla 18 - Escenario Solicitar ingresar - Paso 2

Lo que hicimos en este paso fue agregarle al escenario anterior una frase al objetivo (*Se puede pasar a autorizar el plan*). Si el símbolo de tipo estado cuenta con más de un impacto, se deben agregar todos al objetivo del escenario.

En resumen, para comprender mejor lo expuesto, mostraremos los pasos que debemos seguir, una vez que tenemos el LEL corregido para aplicar la estrategia propuesta en este trabajo de tesis. En negrita se encuentran los pasos que fueron agregados para enriquecer la derivación de los escenarios. Estos son:

1. Realizar la lista de los posibles escenarios.
2. Por cada uno, verificar si contiene un símbolo del LEL que pertenece a la clasificación Verbo.
3. **Verificar si el impacto que originó el escenario contiene un símbolo del tipo estado.**
4. Realizar la derivación del escenario como se explico en el Capítulo 3.
5. **Agregarle al escenario derivado en el punto 4 una precondición al contexto, basándonos en los estados anteriores del símbolo de tipo estado.**
6. **Agregarle al escenario anterior una frase al objetivo, basándose en los impactos del símbolo de tipo estado.**
7. A partir de este punto, se obtiene la lista definitiva de escenarios y sus descripciones, correspondientes a la fase de elicitación de requerimientos.

Mostraremos como quedaba el escenario antes de usar la información clasificada como estado:

Título:	Solicitar ingresar
Objetivo:	Una persona ingresa a un plan de ahorro.
Actores	Administradora, Solicitante
Recursos:	Planilla de adhesión, Cuota de ingreso
Episodio 1:	El solicitante pide la planilla de adhesión.
Episodio 2:	El solicitante completa dicha planilla y la entrega, abonando la cuota de ingreso.

Tabla 19 - Escenario Solicitar ingresar - Derivado sin utilizar símbolos de tipo estado

Y ahora veremos como queda el escenario luego de usar en la derivación los símbolos de tipo estado:

Título:	Solicitar ingresar
Objetivo:	Una persona ingresa a un plan de ahorro. <i>Se puede pasar a autorizar el plan.</i>
Contexto:	La planilla de adhesión se debe encontrar en alguno de los siguientes estados: inicial.
Actores	Administradora, Solicitante
Recursos:	Planilla de adhesión Cuota de ingreso
Episodio 1:	El solicitante pide la planilla de adhesión.
Episodio 2:	El solicitante completa dicha planilla y la entrega, abonando la cuota de ingreso.

Tabla 20 - Escenario Solicitar ingresar - Derivado utilizando símbolos de tipo estado

Entonces, podemos concluir a partir del conocimiento obtenido de los símbolos clasificados como estado, que estos nos brindan información útil que usaremos para enriquecer el contexto y el objetivo del escenario. Lo cual nos permitirá comprender de una mejor manera las restricciones asociadas a un escenario, agregándole detalle a la restricción de los requisitos, permitiendo una mejor comunicación entre los expertos del software y del dominio.

Igualmente una vez terminada la derivación de los escenarios deben ser revisados con los clientes porque son ellos los expertos en el dominio y podrán aceptar o rechazar cada uno de los escenarios.

Se han desarrollado una serie de ejemplos para mostrar la ejecución del proceso aplicado a un caso de estudio. Esto se puede encontrar en el Capítulo 5.

4.4 Particularidades

A continuación daremos una serie de peculiaridades que han surgido al momento de definir la estrategia presentada en el punto anterior. Una de ellas es que si el impacto que originó el escenario contiene un símbolo con la clasificación de estado, pero este es un estado inicial del workflow, al contexto del escenario no lo derivaremos con la nueva regla propuesta en este trabajo, sino que lo haremos como se explicó en el Capítulo 3. Vamos a usar la información del estado para engrosar el objetivo del escenario. Daremos un ejemplo:

Tenemos el posible escenario “Solicitar Iniciar”, además existe un símbolo en el LEL con la clasificación de verbo y el símbolo con la clasificación de estado que es nombrado en el impacto que genera al escenario:

Nombres	Solicitar Iniciar	Categoría	Verbo
Sinónimos			
Noción	El solicitante se interesa sobre los planes de ahorro		
Impacto:	El solicitante pide información sobre un plan de ahorro de interés.		

Tabla 21 - LEL Solicitar iniciar

Nombres	Iniciar	Categoría	Estado
Sinónimos			
Noción	La planilla de adhesión se encuentra en estado inicial una vez que al solicitante se le entrega la planilla.		
Impacto:	Se puede pasar a ingresar la planilla de adhesión.		

Tabla 22 - LEL Iniciar

En nuestro ejemplo el impacto que originó el escenario menciona al estado “Iniciar”, pero como es un estado inicial del workflow no vamos a usar la información del estado

para enriquecer la precondition del contexto. Vamos a usar la información de ese estado para engrosar el objetivo del escenario. En este caso “*Se puede pasar a ingresar la planilla de adhesión*”. Veamos como queda el escenario derivado:

Título:	Solicitar iniciar
Objetivo:	Una persona desea informarse sobre un plan de ahorro. <i>Se puede pasar a ingresar la planilla de adhesión.</i>
Actores	Administradora, Solicitante
Recursos:	Planilla de adhesión Cuota de ingreso
Episodio 1:	El solicitante pide información sobre un plan de ahorro de interés.

Tabla 23 - Escenario Solicitar iniciar

Como podemos ver, no se uso la información del símbolo de tipo estado para enriquecer el contexto del escenario. No utilizamos esta información para agregarle al escenario anterior una precondition al contexto, porque nos estábamos basando en los estados anteriores del símbolo de tipo estado, y como es un estado inicial del workflow, no tiene estados anteriores. Pero si se uso la información de este símbolo para engrosar el objetivo.

Otra particularidad que debemos mencionar es que en el workflow a un estado le puede llegar más de una transición.

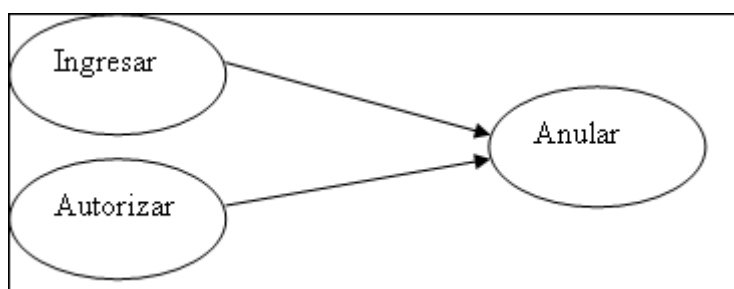


Figura 5 - Workflow de Ingresar a Anular y de Autorizar a Anular

Vamos a utilizar otro ejemplo para explicar esta particularidad. Tenemos el posible escenario “Solicitar Anular”, además existe un símbolo en el LEL con la clasificación de verbo:

Nombres	Solicitar Anular	Categoría	Verbo
Sinónimos			
Noción	El solicitante no desea tener mas un plan de ahorro.		
Impacto:	El solicitante pide la planilla de anulación.		
Impacto:	El solicitante completa dicha planilla y la entrega.		

Tabla 24 - LEL Solicitar Anular

En nuestro ejemplo el impacto que originó el escenario menciona al estado “Anular”, en nuestra precondition se va a nombrar a los estados anteriores a Anular, verificando el

workflow de estados. En este caso se va a nombrar al estado “Ingresar” y “Autorizar”, que son estados anteriores a Anular. A continuación mostraremos los dos estados.

Nombres	Ingresar	Categoría	Estado
Sinónimos			
Noción	La planilla de adhesión se encuentra en estado ingresado una vez que el solicitante completa dicha planilla.		
Impacto:	Se puede pasar a anular el plan.		

Tabla 25 - LEL Ingresar

Nombres	Autorizar	Categoría	Estado
Sinónimos			
Noción	La planilla de adhesión se encuentra en estado autorizado una vez que el solicitante cumpla con los requisitos pedidos por la administradora.		
Impacto:	Se puede pasar a anular el plan.		

Tabla 26 - LEL Autorizar

Entonces la precondition para el contexto del escenario va a tener la forma: “Se debe encontrar en alguno de los siguientes estados:” “nombres de tipo estado”. En nuestro ejemplo nos va a quedar: “La planilla de adhesión se debe encontrar en alguno de los siguientes estados: Ingresar, Autorizar”. Vemos como queda el escenario derivado:

Título:	Solicitar Anular
Objetivo:	El solicitante no desea tener más un plan de ahorro. Se puede pasar a anular el plan si se encuentra en estado Ingresado. Se puede pasar a anular el plan si se encuentra en estado Autorizar.
Contexto:	La planilla de adhesión se debe encontrar en alguno de los siguientes estados: Ingresar, Autorizar.
Actores	Administradora, Solicitante
Recursos:	Planilla de adhesión Cuota de ingreso
Episodio 1:	El solicitante pide la planilla de anulación.
Episodio 2:	El solicitante completa dicha planilla y la entrega.

Tabla 27 - LEL Solicitar Anular

O sea en la precondition del contexto se nombran los dos posibles estados anteriores a Anular, Ingresar y Autorizar. Y para la definición del objetivo usamos los impactos de los símbolos de tipo estado pero poniendo explícitamente al estado al que pertenecen: “Se puede pasar a Anular el plan, si se encuentra en estado ingresado.” “Se puede pasar a Anular el plan si se encuentra en estado autorizado.”

Capítulo 5 – Caso de estudio

5.1 Resumen del caso de estudio

El enunciado de este trabajo se utilizó durante la cursada de la materia Base de Datos 2. El dominio que se ha utilizado es el de una herramienta para el control de cambios y requerimientos de un proyecto de software.

Los requerimientos se clasifican utilizando un tipo de requerimiento (reporte de bug, ampliación, mejora, nuevo) para poder asignarle el equipo correcto. Estos tipos varían de proyecto en proyecto.

El ciclo de vida de un requerimiento está determinado por la secuencia de estados por los que pasa, por ejemplo, Creado, Análisis, Desarrollo, Testing, Evaluación Usuario, Aceptado. Los estados deben ser configurados por el líder de proyecto teniendo en cuenta los diferentes tipos de requerimientos.

De cada tipo de requerimiento se conocen las posibles secuencias que puede seguir un requerimiento clasificado con este tipo. Por ejemplo el siguiente gráfico describe las posibles secuencias de estados que puede seguir un requerimiento.

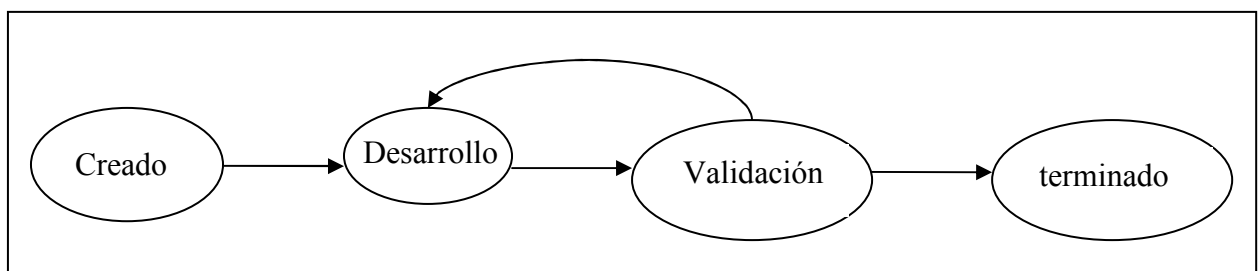


Figura 6 - Posible secuencia de estados que puede seguir un requerimiento

Cada estado definido para un tipo de requerimiento define las propiedades (textuales) que debe brindar un futuro requerimiento de este tipo bajo este estado.

De cada requerimiento, se conoce su prioridad, su tipo de requerimiento, su estado actual, y la secuencia de estados por la que pasó. Tanto el actual como los estados anteriores deben pertenecer a los estados definidos por su tipo de requerimiento. Cada requerimiento en un estado dado, deberá asignar valores a las propiedades definidas para ese estado.

Cuando un requerimiento pasa por un estado se debe asignar un responsable. Puede ocurrir que el responsable de un requerimiento en un estado sea reasignado, con lo cual otro miembro del proyecto deberá ocupar su puesto. Los responsables de los requerimientos siempre serán miembros del proyecto.

Existen dos tipos de usuarios del sistema:

1-Administradores del sistema de control de cambios. Sus actividades pueden ser:

- Configurar el sistema.
- Agregar un nuevo proyecto (estableciendo quién será el líder del proyecto).
- Agregar nuevos usuarios al sistema.

2-Miembros de proyecto. Sus actividades pueden ser:

- Agregar miembros al proyecto.
- Asignar roles a miembros.
- Crear tipo de requerimientos junto con sus estados y las posibles secuencias.
- Crear nuevos requerimientos.
- Pasar de un estado a otro un requerimiento.
- Listar los requerimientos asignados.
- Listar los requerimientos mediante filtros.

Los usuarios de proyectos tienen diferentes roles que pueden cumplir como por ejemplo Líder de Proyecto, Analista, Diseñador, etc. Estos posibles roles están definidos a nivel de proyecto, pues una misma persona puede participar en varios proyectos y su conjunto de roles en un proyecto y otro es diferente.

Las actividades que pueden realizar los miembros de proyecto sobre el sistema están dadas por el rol que cumplen en el proyecto. Es necesario aclarar que una persona puede tener más de un rol en un proyecto, del mismo modo que un rol puede ser cubierto por más de una persona.

5.2 LEL

El trabajo práctico definido en el punto anterior fue considerado como la fuente de información utilizada para la definición del modelo de LEL y luego para los escenarios.

Símbolo	<i>Administrador</i>	Categoría	Sujeto
Sinónimos			
Noción	Es el encargado de la configuración de los proyectos		
Noción	Es un usuario con el rol de administrador		
Impacto:	Configura el Sistema		
Impacto:	Crea proyectos		

Impacto:	Agrega un Project Leader a un proyecto
Impacto:	Cambia un Project Leader de un proyecto
Impacto:	Crea los tipos de requerimientos
Impacto:	Configura los tipos de requerimientos soportados para cada proyecto
Impacto:	Crea los estados
Impacto:	Configura los estados soportados para cada proyecto
Impacto:	Agrega usuarios
Impacto:	Asigna a un usuario como miembro de un proyecto
Impacto:	Agregar roles
Impacto:	Asigna roles a miembros del proyecto

Tabla 28 - LEL Administrador

Símbolo	<i>Project Leader</i>	Categoría	Sujeto
Sinónimos	Líder		
Noción	Es el encargado de guiar la gestión de equipos de un proyecto		
Noción	Es un usuario con el rol de Project Leader		
Impacto:	Crea equipos para un proyecto		
Impacto:	Elimina equipos para un proyecto		
Impacto:	Crea nuevos requerimientos		
Impacto:	Configura secuencias de estados para cada tipo de requerimiento		
Impacto:	Define propiedades para un tipo de requerimiento		
Impacto:	Reasigna un requerimiento		
Impacto:	Cambia al estado creado un requerimiento		
Impacto:	Cambia al estado desarrollo un requerimiento		
Impacto:	Cambia al estado validación un requerimiento		
Impacto:	Cambia al estado terminado un requerimiento		
Impacto:	Asigna miembros de equipo		
Impacto:	Asigna a un miembro de Proyecto como Coordinador de un equipo		

Impacto:	Asigna a un miembro de equipo como Responsable de un requerimiento
-----------------	--

Tabla 29 - LEL Project Leader

Nombres	<i>Coordinador</i>	Categoría	Sujeto
Sinónimos			
Noción	Es el encargado de la coordinación de un equipo del proyecto		
Noción	Es un miembro del proyecto		
Noción	Es un usuario con el rol de Coordinador		
Impacto:	Crea nuevos requerimientos		
Impacto:	Asigna a un Miembro de equipo como Responsable de un requerimiento		
Impacto:	Asigna miembros de equipo		
Impacto:	Reasigna un requerimiento		
Impacto:	Cambia al estado creado un requerimiento		
Impacto:	Cambia al estado desarrollo un requerimiento		
Impacto:	Cambia al estado validación un requerimiento		
Impacto:	Cambia al estado terminado un requerimiento		
Impacto:	Lista los requerimientos mediante filtros		
Impacto:	Lista los requerimientos asignados		

Tabla 30 - LEL Coordinador

Símbolo	<i>Miembro del Proyecto</i>	Categoría	Sujeto
Sinónimos			
Noción	Es un integrante de un proyecto		
Noción	Posee un rol		
Impacto:	Realiza actividades en el sistema de acuerdo al rol que cumple en el proyecto		

Tabla 31 - LEL Miembro del Proyecto

Nombres	<i>Miembro de Equipo</i>	Categoría	Sujeto
Sinónimos			

Noción	Es un integrante de un equipo del proyecto
Noción	Es un miembro del proyecto
Noción	Posee un rol
Impacto:	Lista los requerimientos asignados
Impacto:	Lista los requerimientos mediante filtros

Tabla 32 - LEL Miembro de Equipo

Símbolo	Responsable	Categoría	Sujeto
Sinónimos			
Noción	Es responsable de un requerimiento en un momento dado		
Noción	Es un miembro de un equipo		
Impacto:	Reasigna un requerimiento		
Impacto:	Crea nuevos requerimientos		
Impacto:	Cambia al estado creado un requerimiento		
Impacto:	Cambia al estado desarrollo un requerimiento		
Impacto:	Cambia al estado validación un requerimiento		
Impacto:	Cambia al estado terminado un requerimiento		

Tabla 33 - LEL Responsable

Símbolo	Secuencia de Estados	Categoría	Objeto
Sinónimos	Workflow		
Noción	Conjunto de estados relacionados		
Impacto:	El Project Leader configura la secuencia de estados		
Impacto:	Se conoce el conjunto de estados que la conforman.		

Tabla 34 - LEL Secuencia de Estados

Símbolo	Rol	Categoría	Objeto
Sinónimos			
Noción	Determina las tareas que puede realizar un usuario		
Impacto:	El Administrador configura las tareas que se pueden realizar con un determinado rol		

Impacto:	El Administrador asigna roles a miembros del proyecto
-----------------	---

Tabla 35 - LEL Rol

Símbolo	Sistema	Categoría	Objeto
Sinónimos			
Noción	Conjunto de Proyectos		
Impacto:	El Administrador configura el sistema		
Impacto:	El Administrador agrega proyectos		
Impacto:	El Administrador agrega usuarios		

Tabla 36 - LEL Sistema

Símbolo	<i>Usuario del Sistema</i>	Categoría	Objeto
Sinónimos	Usuario		
Noción	Persona que puede ser asignada a un proyecto		
Impacto:	El Administrador asigna a un usuario como miembro del proyecto		

Tabla 37 - LEL Usuario del Sistema

Símbolo	Requerimiento	Categoría	Objeto
Sinónimos	Actividad		
Noción	Tarea a realizar por un Responsable		
Noción	Se conoce su prioridad , su tipo de requerimiento , su estado actual, y la secuencia de estados por los que pasó		
Impacto:	El Coordinador asigna un miembro de proyecto como responsable		
Impacto:	El Coordinador le asigna un tipo de requerimiento		
Impacto:	El Coordinador le asigna prioridad		
Impacto:	El Project Leader / Responsable / Coordinador cambia de estado		

Tabla 38 - LEL Requerimiento

Símbolo	<i>Proyecto</i>	Categoría	Objeto
Sinónimos			
Noción	Planificación de conjuntos de requerimientos relacionados		

Noción	Dividido en equipos
Impacto:	El Administrador asigna un Project Leader
Impacto:	El Project Leader agrega un equipo
Impacto:	El Project Leader elimina un equipo
Impacto:	El Equipo de trabajo realiza el proyecto
Impacto:	El Administrador crea el proyecto
Impacto:	El Project Leader asigna una configuración previa de los posibles tipos de requerimientos
Impacto:	El Administrador agrega nuevos miembros del proyecto

Tabla 39 - LEL Proyecto

Símbolo	<i>Tipo de Requerimiento</i>	Categoría	Objeto
Sinónimos			
Noción	Clasificación que describe un requerimiento		
Noción:	Se conoce su secuencia de estados para un proyecto		
Impacto:	El Administrador asigna una configuración previa de los posibles tipos de requerimientos		
Impacto:	El Project Leader configura la secuencia de estados para el tipo de requerimiento		

Tabla 40 - LEL Tipo de Requerimiento

Símbolo	<i>Prioridad</i>	Categoría	Objeto
Sinónimos			
Noción	Define la importancia de un requerimiento		
Impacto:	El Project Leader / Coordinador especifica la prioridad de un requerimiento		

Tabla 41 - LEL Prioridad

Símbolo	<i>Propiedad</i>	Categoría	Objeto
Sinónimos			
Noción	Define características textuales que debe brindar un requerimiento en un estado		
Impacto:	El Administrador define propiedades para un tipo de requerimiento		

Tabla 42 - LEL Propiedad

Nombres	<i>Equipo</i>	Categoría	Objeto
Sinónimos	Equipo de trabajo		
Noción	Conjunto de miembros de proyecto con fines en común		
Impacto:	El Administrador asigna un Coordinador		
Impacto:	El Project Leader / Coordinador asigna miembros de equipo		

Tabla 43 - LEL Equipo

Símbolo	<i>Estado</i>	Categoría	Estado
Sinónimos			
Noción	Situación en la que se encuentra un requerimiento		
Noción	Situación en la cual el requerimiento posee un responsable		
Impacto:	El requerimiento posee un estado		

Tabla 44 - LEL Estado

Símbolo	<i>Creado</i>	Categoría	Estado
Sinónimos			
Noción	Es un estado		
Noción	Situación en la que se encuentra un requerimiento del tipo Bug		
Impacto:	La posible secuencia que puede seguir es al estado de desarrollo		

Tabla 45 - LEL Estado

Símbolo	<i>Desarrollo</i>	Categoría	Estado
Sinónimos			
Noción	Es un estado		
Noción	Situación en la que se encuentra un requerimiento del tipo Bug		
Impacto:	La posible secuencia que puede seguir es al estado de validación		

Tabla 46 - LEL Desarrollo

Símbolo	<i>Validación</i>	Categoría	Estado
Sinónimos			
Noción	Es un estado		

Noción	Situación en la que se encuentra un requerimiento del tipo Bug
Impacto:	La posible secuencia que puede seguir es al estado de desarrollo o terminado

Tabla 47 - LEL Validación

Símbolo	<i>Terminado</i>	Categoría	Estado
Sinónimos			
Noción	Es un estado		
Noción	Situación en la que se encuentra un requerimiento del tipo Bug		
Impacto:	Un requerimiento en este estado no puede pasar a ningún otro estado		

Tabla 48 - LEL Terminado

Nombres	<i>Crear Proyecto</i>	Categoría	Verbo
Sinónimos			
Noción	Agregar un nuevo proyecto al sistema		
Impacto:	El Administrador define el nombre del proyecto		
Impacto:	El Administrador agrega a un usuario como Project Leader		
Impacto:	El Administrador configura los tipos de requerimientos soportados por el proyecto		
Impacto:	El Project Leader agrega miembros de equipo al proyecto		
Impacto:	El proyecto no se debe encontrar en el sistema		

Tabla 49 - LEL Crear Proyecto

Nombres	Cambiar un Project Leader de un proyecto	Categoría	Verbo
Sinónimos			
Noción	El Administrador reasigna un nuevo usuario como Project Leader		
Impacto:	El usuario debe tener el rol de Project Leader		

Tabla 50 - LEL Cambiar un Project Leader de un proyecto

Nombres	<i>Agregar Equipo</i>	Categoría	Verbo
Sinónimos			
Noción	Agrega un equipo a un proyecto		
Impacto:	El Project Leader define un nombre de equipo		

Impacto:	El Project Leader agrega un usuario como Coordinador del equipo
Impacto:	El Coordinador agrega miembros al equipo

Tabla 51 - LEL Agregar Equipo

Nombres	<i>Agregar Miembros</i>	Categoría	Verbo
Sinónimos			
Noción	Agrega miembros de proyecto como miembro de un equipo		
Impacto:	Seleccionar un miembro de proyecto para que sea miembro del equipo		

Tabla 52 - LEL Agregar Miembros

Nombres	<i>Crear Requerimiento</i>	Categoría	Verbo
Sinónimos			
Noción	Se agrega al conjunto de requerimientos de un proyecto		
Impacto:	El Project Leader / Coordinador asocia una prioridad		
Impacto:	El Project Leader / Coordinador asocia el estado creado		
Impacto:	El Project Leader / Coordinador asocia un equipo		
Impacto:	El Project Leader / Coordinador asocia un responsable		
Impacto:	El Project Leader / Coordinador asocia un tipo de requerimiento		

Tabla 53 - LEL Crear Requerimiento

Nombres	<i>Cambiar el estado de un requerimiento que se encuentra en estado Creado</i>	Categoría	Verbo
Sinónimos			
Noción	Asignar un nuevo estado		
Impacto:	El Responsable reasigna el requerimiento		
Impacto:	El Responsable asigna valores a las propiedades asociadas al nuevo estado		
Impacto:	El Responsable / Project Leader / Coordinador cambia el estado del requerimiento		

Tabla 54 - LEL Cambiar el estado de un requerimiento que se encuentra Creado

Nombres	<i>Cambiar el estado de un requerimiento que se encuentra en estado Desarrollo</i>	Categoría	Verbo
Sinónimos			
Noción	Asignar un nuevo estado		

Impacto:	El Responsable reasigna el requerimiento
Impacto:	El Responsable asigna valores a las propiedades asociadas al nuevo estado
Impacto:	El Responsable / Project Leader / Coordinador cambia el estado del requerimiento

Tabla 55 - LEL Cambiar el estado de un requerimiento en Desarrollo

Nombres	<i>Cambiar el estado de un requerimiento que se encuentra en estado Validación</i>	Categoría	Verbo
Sinónimos			
Noción	Asignar un nuevo estado		
Impacto:	El Responsable reasigna el requerimiento		
Impacto:	El Responsable asigna valores a las propiedades asociadas al nuevo estado		
Impacto:	El Responsable / Project Leader / Coordinador cambia el estado del requerimiento		

Tabla 56 - LEL Cambiar el estado de un requerimiento en Validación

Nombres	<i>Cambiar el estado de un requerimiento que se encuentra en estado Terminado</i>	Categoría	Verbo
Sinónimos			
Noción	Asignar un nuevo estado		
Impacto:	El Responsable reasigna el requerimiento		
Impacto:	El Responsable asigna valores a las propiedades asociadas al nuevo estado		
Impacto:	El Responsable / Project Leader / Coordinador cambia el estado del requerimiento		

Tabla 57 - LEL Cambiar el estado de un requerimiento en Terminado

Nombres	<i>Agregar Usuario al sistema</i>	Categoría	Verbo
Sinónimos			
Noción	Agregar un usuario al sistema		
Impacto:	El nuevo usuario no debe ser un usuario del sistema		

Tabla 58 - LEL Agregar Usuario al sistema

Nombres	Asignar a un usuario como miembro del proyecto	Categoría	Verbo
Sinónimos			

Noción	Agregar un usuario como miembro del proyecto
Impacto:	El nuevo usuario debe ser un usuario del sistema
Impacto:	El nuevo usuario no debe ser un miembro del proyecto

Tabla 59 - LEL Asignar a un usuario como miembro del proyecto

Nombres	<i>Crear un Tipo de Requerimiento</i>	Categoría	Verbo
Sinónimos			
Noción	Se agrega un nuevo tipo de requerimiento al sistema		
Impacto:	El Administrador agrega el nuevo tipo de requerimiento al sistema		
Impacto:	El Project Leader configura la secuencia de estados		
Impacto:	El Project Leader define propiedades para un tipo de requerimiento		

Tabla 60 - LEL Crear un Tipo de Requerimiento

Nombres	Configurar la secuencia de estados	Categoría	Verbo
Sinónimos			
Noción	El Project Leader crea un grafo con los estados disponibles en el sistema		
Impacto:	El Project Leader asigna la secuencia de estados a un tipo de requerimiento		

Tabla 61 - LEL Configurar la secuencia de estados

Nombres	<i>Reasignar Requerimiento</i>	Categoría	Verbo
Sinónimos			
Noción	El Responsable del requerimiento asigna un nuevo responsable		
Impacto:	El Coordinador / Responsable del requerimiento asigna un nuevo responsable		
Impacto:	El nuevo responsable del requerimiento debe ser un miembro del proyecto		

Tabla 62 - LEL Reasignar Requerimiento

Nombres	<i>Agregar Rol</i>	Categoría	Verbo
Sinónimos			
Noción	Agregar tareas que se pueden realizar bajo este rol		
Impacto:	El Administrador le asigna un nombre al rol		

Tabla 63 - LEL Agregar Rol

Nombres	<i>Asignar un Rol a un miembro del proyecto</i>	Categoría	Verbo
Sinónimos			
Noción	Agregar un rol a un miembro del proyecto		
Impacto:	El rol no se debe encontrar ya asociado al miembro del proyecto		

Tabla 64 - LEL Asignar un Rol a un miembro del proyecto

Nombres	<i>Configurar Sistema</i>	Categoría	Verbo
Sinónimos			
Noción	Agregar recursos mínimos necesarios para el funcionamiento del software		
Impacto:	El Administrador es el encargado de agregar los recursos		

Tabla 65 - LEL Configurar Sistema

5.3 Derivación de escenarios

Siguiendo las heurísticas propuestas en [Leite 95], [Hadad96] y [Hadad97], se construyeron y documentaron los escenarios. Utilizando como fuente de información el LEL. A continuación se presentan los escenarios resultantes.

Actores Principales

- Administrador
- Project Leader
- Coordinador
- Desarrollador
- Responsable

Escenarios Candidatos

- Configurar el sistema
- Crear proyecto
- Agregar un Project leader a un proyecto
- Cambiar un Project leader de un proyecto
- Configurar los tipos de requerimientos soportados para cada proyecto
- Crear los tipos de requerimientos
- Configurar los estados soportados para cada proyecto
- Crear los estados
- Agregar usuarios
- Asignar a un usuario como miembro de un proyecto
- Agregar roles
- Asignar roles a miembros del proyecto
- Crear equipos para un proyecto

- Eliminar equipo de un proyecto
- Crear nuevos requerimientos
- Configurar secuencia de estados para cada tipo de requerimiento
- Definir propiedades para un tipo de requerimiento
- Reasignar un requerimiento
- Cambiar el estado de un requerimiento que se encuentra en estado Creado
- Cambiar el estado de un requerimiento que se encuentra en estado Desarrollo
- Cambiar el estado de un requerimiento que se encuentra en estado Validación
- Asignar miembros de equipo
- Asignar a un miembro de proyecto como coordinador de un equipo
- Asignar a un miembro de equipo como responsable de un requerimiento
- Listar los requerimientos mediante filtros
- Listar los requerimientos asignados

Título:	Crear Proyecto
Objetivo:	Agregar el nuevo proyecto al Sistema.
Contexto:	El proyecto no se debe encontrar en el sistema. El usuario que crea el proyecto debe tener el rol de administrador
Actores	Administrador, Project Leader
Episodio 1:	Definir un nombre para el nuevo proyecto
Episodio 2:	Agregar un usuario como Project Leader
Episodio 3:	Configurar los tipos de requerimientos soportados
Episodio 4:	Agregar equipos al proyecto

Tabla 66 - Escenario Crear Proyecto

Título:	Crear equipo para un proyecto
Objetivo:	Agregar un nuevo equipo a un proyecto
Contexto:	El equipo no se debe encontrar en la lista de equipos del proyecto. El usuario que realiza esta acción debe tener el rol de Project Leader
Actores	Project Leader, Coordinador, Miembros de proyecto
Episodio 1:	Definir un nombre para el nuevo equipo
Episodio 2:	Seleccionar un proyecto para agregarle el equipo
Episodio 3:	Asignar a un miembro de proyecto como coordinador de un equipo
Episodio 4:	Asignar a un miembros de proyecto como miembros de equipos
Episodio 5:	Agregar el nuevo equipo a la lista de equipos del proyecto

Tabla 67 - Escenario Crear equipo para un proyecto

Título:	Eliminar equipo de un proyecto
Objetivo:	Borrar un equipo de la lista de equipos de un proyecto
Contexto:	El equipo se debe encontrar en la lista de equipos del proyecto. El usuario que realiza esta acción debe tener el rol de Project Leader. El equipo no debe tener asignados miembros de equipo como tampoco un coordinador
Actores	Project Leader
Episodio 1:	Seleccionar un proyecto
Episodio 2:	Seleccionar el equipo a ser eliminado

Episodio 3:	Eliminar el equipo de la lista de equipos
--------------------	---

Tabla 68 - Escenario Eliminar equipo de un proyecto

Título:	Crear nuevos requerimientos
Objetivo:	Agregar un nuevo requerimiento a la lista de requerimientos del equipo
Contexto:	No se debe encontrar un requerimiento con las mismas características.
Actores	Project Leader, Coordinador, Responsable
Episodio 1:	Definir una descripción para el requerimiento
Episodio 2:	Seleccionar un tipo de requerimiento
Episodio 3:	Completar las propiedades que debe cumplir el requerimiento por pertenecer a un cierto tipo
Episodio 4:	Asociar un equipo
Episodio 5:	Asociar una prioridad
Episodio 6:	Asociar el estado creado
Episodio 7:	Asignar a un miembro de equipo como responsable del requerimiento
Episodio 8:	Agregar el requerimiento a la lista de requerimientos del equipo

Tabla 69 - Escenario Crear nuevos requerimientos

Título:	Configurar secuencia de estados para cada tipo de requerimiento
Objetivo:	Armar un workflow de estados por los que debe pasar un requerimiento clasificado con un cierto tipo
Contexto:	Se deben seleccionar estados que se encuentren soportados por el proyecto
Actores	Project Leader
Episodio 1:	Seleccionar un tipo de requerimiento
Episodio 2:	Definir un estado inicial
Episodio 3:	Definir una secuencia al siguiente estado
Episodio 4:	Realizar el paso 3 hasta que se llegue a un estado final

Tabla 70 - Escenario Configurar secuencia de estados para cada tipo de requerimiento

Título:	Definir propiedades para un tipo de requerimiento
Objetivo:	Configurar propiedades textuales que debe brindar un futuro requerimiento de un tipo bajo un estado
Actores	Project Leader
Episodio 1:	Seleccionar un tipo de requerimiento de un proyecto
Episodio 2:	Seleccionar un estado del workflow de estados soportados por ese tipo de requerimiento
Episodio 3:	Agregar una propiedad textual a la lista de propiedades del tipo de requerimiento bajo un estado

Tabla 71 - Escenario Definir propiedades para un tipo de requerimiento

Título:	Reasignar un requerimiento
Objetivo:	Cambiar de responsable un requerimiento
Contexto:	El nuevo responsable debe pertenecer al mismo proyecto que pertenece el requerimiento

Actores	Project Leader, Coordinador, Responsable
Episodio 1:	Seleccionar el requerimiento
Episodio 2:	Elegir el nuevo responsable para el requerimiento

Tabla 72 - Escenario Reasignar un requerimiento

Título:	Cambiar el estado de un requerimiento que se encuentra en estado Creado
Objetivo:	El Project Leader / Coordinador / Responsable cambia el estado de un requerimiento
Actores	Project Leader, Coordinador, Responsable
Episodio 1:	Reasignar el requerimiento, poniendo a otro miembro como responsable
Episodio 2:	Asignar valores a las propiedades asociadas al nuevo estado
Episodio 3:	Seleccionar un nuevo estado para el requerimien

Tabla 73 - Escenario Cambiar el estado de un requerimiento Creado

Título:	Cambiar el estado de un requerimiento que se encuentra en estado Desarrollo
Objetivo:	El Project Leader / Coordinador / Responsable cambia el estado de un requerimiento
Actores	Project Leader, Coordinador, Responsable
Episodio 1:	Reasignar el requerimiento, poniendo a otro miembro como responsable
Episodio 2:	Asignar valores a las propiedades asociadas al nuevo estado
Episodio 3:	Seleccionar un nuevo estado para el requerimiento

Tabla 74 - Escenario Cambiar el estado de un requerimiento en Desarrollo

Título:	Cambiar el estado de un requerimiento que se encuentra en estado Validación
Objetivo:	El Project Leader / Coordinador / Responsable cambia el estado de un requerimiento
Actores	Project Leader, Coordinador, Responsable
Episodio 1:	Reasignar el requerimiento, poniendo a otro miembro como responsable
Episodio 2:	Asignar valores a las propiedades asociadas al nuevo estado
Episodio 3:	Seleccionar un nuevo estado para el requerimiento

Tabla 75 - Escenario Cambiar el estado de un requerimiento en Validación

Título:	Asignar miembros de equipo
Objetivo:	Agregar a un miembro de proyecto a la lista de miembros de equipo de un equipo
Contexto:	El usuario a agregar debe ser un miembro de proyecto El miembro de proyecto no debe existir en la lista de miembros de equipo
Actores	Project Leader, Coordinador, Responsable
Episodio 1:	Seleccionar un equipo
Episodio 2:	Seleccionar un miembro de proyecto
Episodio 3:	Agregar el miembro de proyecto elegido a la lista de miembros de

	equipo
--	--------

Tabla 76 - Escenario Asignar miembros de equipo

Título:	Asignar a un miembro de proyecto como coordinador de un equipo
Objetivo:	Agregar a un miembro de proyecto como coordinador de un equipo
Contexto:	El usuario a agregar debe ser un miembro de proyecto. Debe haber un único coordinador por equipo
Actores	Project Leader, Coordinador
Episodio 1:	Seleccionar un equipo
Episodio 2:	Seleccionar un miembro de proyecto
Episodio 3:	Agregar el miembro de proyecto elegido como coordinador

Tabla 77 - Escenario Asignar a miembro de proyecto como coordinador de equipo

Título:	Asignar a un miembro de equipo como responsable de un requerimiento
Objetivo:	Agregar un responsable a un requerimiento
Contexto:	El responsable debe pertenecer al equipo al que pertenece el requerimiento
Actores	Project Leader, Coordinador
Episodio 1:	Seleccionar un requerimiento
Episodio 2:	Buscar entre los miembros de un equipo al responsable del requerimiento
Episodio 3:	Buscar el estado inicial en el workflow de estados

Tabla 78 - Escenario Asignar miembro de equipo como responsable de requerimiento

Título:	Listar los requerimientos mediante filtros
Objetivo:	Ver una lista de requerimientos filtrados por algún criterio
Actores	Coordinador, Responsable
Episodio 1:	Seleccionar el filtro a aplicar
Episodio 2:	Listar los requerimientos que resultaron seleccionados al aplicar el filtro

Tabla 79 - Escenario Listar los requerimientos mediante filtros

Título:	Listar los requerimientos asignados
Objetivo:	Ver la lista de los requerimientos asignados a un miembro del equipo
Contexto:	Los requerimientos listados deben tener como último responsable al usuario loggeado
Actores	Coordinador, Responsable
Episodio 1:	Seleccionar el filtro "requerimientos asignados"
Episodio 2:	Listar los requerimientos que resultaron seleccionados al aplicar el filtro

Tabla 80 - Escenario Listar los requerimientos asignados

Título:	Configurar el Sistema
Objetivo:	El Administrador inicializa los recursos mínimos necesarios para el

	funcionamiento del software
Contexto:	Es la primera tarea a realizar en el sistema.
Actores	Administrador
Episodio 1:	Agregar los tipos de requerimientos
Episodio 2:	Agregar los estados
Episodio 3:	Agregar usuarios
Episodio 4:	Configurar roles

Tabla 81 - Escenario Configurar el Sistema

Título:	Agregar un Project Leader a un proyecto
Objetivo:	Definir un Project Leader para un Proyecto
Contexto:	El usuario a asignar debe tener el rol de líder de proyecto. El proyecto elegido no puede tener asociado un Project Leader
Actores	Project Leader, Administrador
Episodio 1:	Seleccionar un Proyecto
Episodio 2:	Seleccionar un usuario
Episodio 3:	Asignar el usuario elegido como Project Leader
Excepciones:	Si el usuario elegido no tiene el rol de Project Líder se informa el error al Administrador.

Tabla 82 - Escenario Agregar un Project Leader a un proyecto

Título:	Cambiar un Project Leader de un proyecto
Objetivo:	Cambiar el Project Leader de un proyecto por otro
Contexto:	El usuario a asignar debe tener el rol de líder de proyecto. El proyecto elegido debe tener asociado un Project Leader
Actores	Project Leader, Administrador
Episodio 1:	Seleccionar un Proyecto
Episodio 2:	Seleccionar un usuario
Episodio 3:	Asignar el usuario elegido como Project Leader
Excepciones:	Si el usuario elegido no tiene el rol de Project Líder se informa el error al Administrador.

Tabla 83 - Escenario Cambiar un Project Leader de un proyecto

Título:	Crear los tipos de requerimientos
Objetivo:	Agregar nuevos tipos de requerimientos
Contexto:	El nuevo tipo de requerimiento no debe existir en el sistema
Actores	Administrador, Project Leader
Episodio 1:	Asignar un nombre
Episodio 2:	Configurar la secuencia de estados
Episodio 3:	Definir las propiedades
Episodio 4:	Agregar a la lista de tipos de requerimientos del sistema

Tabla 84 - Escenario Crear los tipos de requerimientos

Título:	Configurar los tipos de requerimientos soportados para cada proyecto
Objetivo:	Definir los tipos de requerimientos para un proyecto

Actores	Administrador
Episodio 1:	Elegir el Proyecto
Episodio 2:	Elegir los tipos de requerimientos
Episodio 3:	Agregar los tipos de requerimientos seleccionados a la lista de tipos de requerimientos del proyecto

Tabla 85 - Escenario Configurar tipos de requerimientos para cada proyecto

Título:	Crear los estados
Objetivo:	nuevos estados al sistema
Contexto:	El nuevo estado no debe existir en el sistema
Actores	Administrador
Episodio 1:	Asignar un nombre
Episodio 2:	Agregar a la lista de estados del sistema

Tabla 86 - Escenario Crear los estados

Título:	Configurar los estados soportados para cada proyecto
Objetivo:	Definir los estados para un proyecto
Actores	Administrador
Episodio 1:	Elegir el Proyecto
Episodio 2:	Elegir los estados
Episodio 3:	Agregar los estados seleccionados a la lista de estados del proyecto

Tabla 87 - Escenario Configurar los estados soportados para cada proyecto

Título:	Agregar usuarios
Objetivo:	Agregar los usuarios a la lista de usuarios del sistema
Contexto:	El nuevo usuario no debe existir en el sistema
Actores	Administrador
Episodio 1:	Definir propiedades
Episodio 2:	Agregar a la lista de usuarios

Tabla 88 - Escenario Agregar usuarios

Título:	Asignar a un usuario como miembro de un proyecto
Objetivo:	Agregar a un usuario a la lista de miembros de un proyecto
Contexto:	El usuario a agregar no debe ser un miembro de proyecto
Actores	Administrador, Miembro de proyecto
Episodio 1:	Seleccionar un usuario
Episodio 2:	Seleccionar un proyecto
Episodio 3:	Agregar el usuario elegido a la lista de miembros de proyecto

Tabla 89 - Escenario Asignar a un usuario como miembro de un proyecto

Título:	Agregar roles
Objetivo:	Agregar el nuevo rol a la lista de roles del sistema
Contexto:	El nuevo rol no se debe encontrar en el sistema
Actores	Administrador
Episodio 1:	Definir un nombre

Episodio 2:	Definir las acciones que se pueden realizar bajo este rol
Episodio 3:	Agregar el rol a la lista de roles del sistema

Tabla 90 - Escenario Agregar roles

Título:	Asignar roles a miembros del proyecto
Objetivo:	Agregar un rol a un miembro del proyecto
Contexto:	Si el rol a asignar ya se encuentra asociado al miembro del proyecto se informa al Administrador
Actores	Administrador, Miembro de Proyecto
Episodio 1:	Elegir un miembro de proyecto
Episodio 2:	Elegir un rol
Episodio 3:	Agregarle el rol a la lista de roles del miembro de proyecto

Tabla 91 - Escenario Asignar roles a miembros del proyecto

5.4 Aplicación de la estrategia propuesta

A continuación daremos una serie de ejemplos para mostrar la ejecución del proceso de la estrategia de derivación propuesta en esta tesis, aplicado a nuestro caso de estudio. Tenemos un posible escenario que es “Cambiar al estado creado un requerimiento”, verificamos que existe un símbolo del LEL que pertenece a la clasificación Verbo.

Nombres	<i>Cambiar el estado de un requerimiento que se encuentra en estado Creado</i>	Categoría	Verbo
Sinónimos			
Noción	Asignar un nuevo estado		
Impacto:	El Responsable reasigna el requerimiento		
Impacto:	El Responsable asigna valores a las propiedades asociadas al nuevo estado		
Impacto:	El Responsable / Project Leader / Coordinador cambia el estado del requerimiento		

Tabla 92 - LEL Cambiar el estado de un requerimiento Creado

Entonces, aplicando las heurísticas conocidas hasta este momento, nos queda el siguiente escenario:

Título:	Cambiar el estado de un requerimiento que se encuentra en estado Creado
Objetivo:	El Project Leader / Coordinador / Responsable cambia el estado de un requerimiento
Actores	Project Leader, Coordinador, Responsable
Episodio 1:	Reasignar el requerimiento, poniendo a otro miembro como responsable
Episodio 2:	Asignar valores a las propiedades asociadas al nuevo estado
Episodio 3:	Cambiar el estado del requerimiento

Tabla 93 - Escenario Cambiar el estado Creado

Pero en el LEL además contamos con el siguiente símbolo, que se encuentra clasificado bajo la categoría de estado:

Símbolo	<i>Creado</i>	Categoría	Estado
Sinónimos			
Noción	Es un estado		
Noción	Situación en la que se encuentra un requerimiento del tipo Bug		
Impacto:	La posible secuencia que puede seguir es al estado de desarrollo		

Tabla 94 - LEL Creado

Podemos utilizar la información de este símbolo para enriquecer el contexto del escenario antes derivado. Entonces para la precondition del contexto no nos vamos a basar en los estados anteriores al que se hace referencia en el impacto que le dio origen, en este caso “Creado”, porque es un estado inicial del workflow. Pero los impactos asociados a este símbolo lo podríamos usar para la definición del objetivo. Esto nos daría como resultado el siguiente escenario:

Título:	Cambiar el estado de un requerimiento que se encuentra en estado Creado
Objetivo:	El Project Leader / Coordinador / Responsable cambia el estado de un requerimiento. La posible secuencia que puede seguir es al estado de desarrollo.
Actores	Project Leader, Coordinador, Responsable
Episodio 1:	Reasignar el requerimiento, poniendo a otro miembro como responsable
Episodio 2:	Asignar valores a las propiedades asociadas al nuevo estado
Episodio 3:	Cambiar el estado del requerimiento

Tabla 95 - Escenario Cambiar el estado de un requerimiento Creado

Realizamos el mismo procedimiento con los demás escenarios que tienen las mismas características que el antes mencionado.

Nombres	<i>Cambiar el estado de un requerimiento que se encuentra en estado Desarrollo</i>	Categoría	Verbo
Sinónimos			
Noción	Asignar un nuevo estado		
Impacto:	El Responsable reasigna el requerimiento		
Impacto:	El Responsable asigna valores a las propiedades asociadas al nuevo estado		

Impacto:	El Responsable / Project Leader / Coordinador cambia el estado del requerimiento
-----------------	--

Tabla 96 - LEL Cambiar el estado de un requerimiento Desarrollo

Se utiliza además para la construcción del escenario el siguiente símbolo:

Símbolo	<i>Desarrollo</i>	Categoría	Estado
Sinónimos			
Noción	Es un estado		
Noción	Situación en la que se encuentra un requerimiento del tipo Bug		
Impacto:	La posible secuencia que puede seguir es al estado de validación		

Tabla 97 - LEL Desarrollo

Aplicando las heurísticas conocidas hasta este momento, y la estrategia propuesta, nos queda el siguiente escenario:

Título:	Cambiar el estado de un requerimiento que se encuentra en estado Desarrollo
Objetivo:	El Project Leader / Coordinador / Responsable cambia el estado de un requerimiento. La posible secuencia que puede seguir es al estado de validación.
Contexto:	Se debe encontrar en alguno de los siguientes estados: Creado, Validación
Actores	Project Leader, Coordinador, Responsable
Episodio 1:	Reasignar el requerimiento, poniendo a otro miembro como responsable
Episodio 2:	Asignar valores a las propiedades asociadas al nuevo estado
Episodio 3:	Cambiar el estado del requerimiento

Tabla 98 - Escenario Cambiar el estado de un requerimiento Desarrollo

En este caso, nos basamos en los estados anteriores del símbolo de tipo estado para engrosar el contexto del escenario, por no ser un este un estado inicial del workflow. En este ejemplo se puede encontrar en estado creado o de validación.

Daremos otro ejemplo encontrado analizando el caso de estudio:

Nombres	<i>Cambiar el estado de un requerimiento que se encuentra en estado Validación</i>	Categoría	Verbo
Sinónimos			
Noción	Asignar un nuevo estado		
Impacto:	El Responsable reasigna el requerimiento		

Impacto:	El Responsable asigna valores a las propiedades asociadas al nuevo estado
Impacto:	El Responsable / Project Leader / Coordinador cambia el estado del requerimiento

Tabla 99 - LEL Cambiar el estado de un requerimiento Validación

También contamos con un símbolo de tipo Estado relacionado con el símbolo anterior:

Símbolo	<i>Validación</i>	Categoría	Estado
Sinónimos			
Noción	Es un estado		
Noción	Situación en la que se encuentra un requerimiento del tipo Bug		
Impacto:	La posible secuencia que puede seguir es al estado de desarrollo o terminado		

Tabla 100 - LEL Validación

Realizando la derivación de escenarios no queda lo siguiente:

Título:	Cambiar el estado de un requerimiento que se encuentra en estado Validación
Objetivo:	El Project Leader / Coordinador / Responsable cambia el estado de un requerimiento. La posible secuencia que puede seguir es al estado terminado.
Contexto:	Se debe encontrar en alguno de los siguientes estados: Desarrollo
Actores	Project Leader, Coordinador, Responsable
Episodio 1:	Reasignar el requerimiento, poniendo a otro miembro como responsable
Episodio 2:	Asignar valores a las propiedades asociadas al nuevo estado
Episodio 3:	Cambiar el estado del requerimiento

Tabla 101 - Escenario Cambiar el estado de un requerimiento Validación

Veamos un último ejemplo, si tenemos el posible escenario “Cambiar el estado de un requerimiento que se encuentra en estado Terminado”. Contamos con un símbolo del LEL que pertenece a la clasificación verbo. Y este símbolo nombra a otro símbolo del tipo estado. Buscamos dichos símbolos en el LEL:

Nombres	<i>Cambiar el estado de un requerimiento que se encuentra en estado Terminado</i>	Categoría	Verbo
Sinónimos			
Noción	Asignar un nuevo estado		
Impacto:	El Responsable reasigna el requerimiento		

Impacto:	El Responsable asigna valores a las propiedades asociadas al nuevo estado
Impacto:	El Responsable / Project Leader / Coordinador cambia el estado del requerimiento

Tabla 102 - LEL Cambiar el estado de un requerimiento Terminado

Símbolo	<i>Terminado</i>	Categoría	Estado
Sinónimos			
Noción	Es un estado		
Noción	Situación en la que se encuentra un requerimiento del tipo Bug		
Impacto:	Un requerimiento en este estado no puede pasar a ningún otro estado		

Tabla 103 - LEL Terminado

Aplicando las heurísticas conocidas hasta este momento, y la estrategia propuesta, nos queda el siguiente escenario:

Título:	Cambiar el estado de un requerimiento que se encuentra en estado Terminado
Objetivo:	El Project Leader / Coordinador / Responsable cambia el estado de un requerimiento. Un requerimiento en este estado no puede pasar a ningún otro estado.
Contexto:	Se debe encontrar en alguno de los siguientes estados: Validación
Actores	Project Leader, Coordinador, Responsable
Episodio 1:	Reasignar el requerimiento, poniendo a otro miembro como responsable
Episodio 2:	Asignar valores a las propiedades asociadas al nuevo estado
Episodio 3:	Cambiar el estado del requerimiento

Tabla 104 - Escenario Cambiar el estado de un requerimiento Terminado

Capítulo 6 – Herramienta para implementar la estrategia de derivación

6.1 Motivación

La herramienta permite la creación del LEL y basándose en él permite generar una primera versión de los escenarios involucrados en el proyecto en desarrollo, permitiendo luego depurarlos, para lograr así una versión definitiva.

Existen otras herramientas desktop similares desarrolladas que permiten realizar una primera derivación de escenarios, pero creemos necesario también que la herramienta sea simple de usar y accesible desde cualquier sitio. Es por esto que optamos por el desarrollo de una aplicación Web que permitirá editar, registrar y gestionar el LEL, la derivación de escenarios con la incorporación de la nueva derivación propuesta en esta tesis.

6.2 Tecnologías utilizadas

Para realizar esta aplicación decidimos usar frameworks ampliamente utilizados en el desarrollo web que nos permitan desarrollar la herramienta de la manera más rápida y fácil posible, por ejemplo utilizando generación automática de Altas, Bajas, Modificaciones. Decidimos elegir un framework basado en PHP (CakePHP) y como base de datos utilizaremos MySQL debido a la excelente interacción que existe con PHP. Para la generación del esquema de base de datos y del diagrama entidad relación usamos MySQL Query Browser y MySQL Workbench respectivamente. A continuación describimos algunas características que respaldan nuestra elección.

PHP

PHP es un lenguaje interpretado de propósito general ampliamente usado y que está diseñado especialmente para desarrollo Web y puede ser incrustado dentro de código HTML. Generalmente se ejecuta en un servidor Web, tomando el código en PHP como su entrada y creando páginas Web como salida. Puede ser desplegado en la mayoría de los servidores Web y en casi todos los sistemas operativos y plataformas sin costo alguno.

Cuando el cliente hace una petición al servidor para que le envíe una página Web, el servidor ejecuta el intérprete de PHP. Éste procesa el script solicitado que generará el contenido de manera dinámica (por ejemplo obteniendo información de una base de datos). El resultado es enviado por el intérprete al servidor, quien a su vez se lo envía al cliente.

Características:

- Es un lenguaje multiplataforma.
- Completamente orientado a la Web.
- Capacidad de conexión con la mayoría de los motores de base de datos que se utilizan en la actualidad, destaca su conectividad con MySQL y PostgreSQL.
- Capacidad de expandir su potencial utilizando la enorme cantidad de módulos (llamados ext's o extensiones).
- Posee una amplia documentación en su página oficial, entre la cual se destaca que todas las funciones del sistema están explicadas y ejemplificadas en un único archivo de ayuda.
- Es libre, por lo que se presenta como una alternativa de fácil acceso para todos.
- Permite aplicar técnicas de programación orientada a objetos.
- Biblioteca nativa de funciones sumamente amplia e incluida.
- No requiere definición de tipos de variables aunque sus variables se pueden evaluar también por el tipo que estén manejando en tiempo de ejecución.
- Tiene manejo de excepciones.

CakePHP

Hoy en día raramente se arranca una aplicación desde cero, para eso se utilizan frameworks. Un framework, en el desarrollo de software, es una estructura de soporte definida, mediante la cual otro proyecto de software puede ser organizado y desarrollado. Típicamente, puede incluir soporte de programas, bibliotecas y un lenguaje interpretado entre otros software para ayudar a desarrollar y unir los diferentes componentes de un proyecto.

CakePHP es un Framework, no un conjunto de librerías, aunque el contiene docenas de funciones y métodos que simplifican el desarrollo Web tal como las librerías. Cake espera que el desarrollador extienda sus objetos y agregue sus propios recursos customizados.

Características:

Desarrollo Rápido: Cake hace fácil los pasos iniciales de construir aplicaciones Web. En vez de correr scripts de instalación desde la línea de comando, Cake viene pre-empaquetado como una carpeta que simplemente colocándola en un servidor Web esta lista para arrancar.

Model-View-Controller: Cake utiliza una estructura MVC para aplicaciones Web. Básicamente, separa las operaciones típicas en áreas específicas: los modelos para la interacción con bases de datos, visto para todas las salidas, y controladores para todos los comandos para la entrada y el flujo del programa. La típica aplicación PHP mezcla cada una de estas tres funciones en el mismo código, por lo que es difícil de mantener y depurar.

El objetivo principal de MVC es asegurarse de que cada función de la aplicación esta escrita una vez y sólo una vez, simplificando así el código y evitando redundancia.

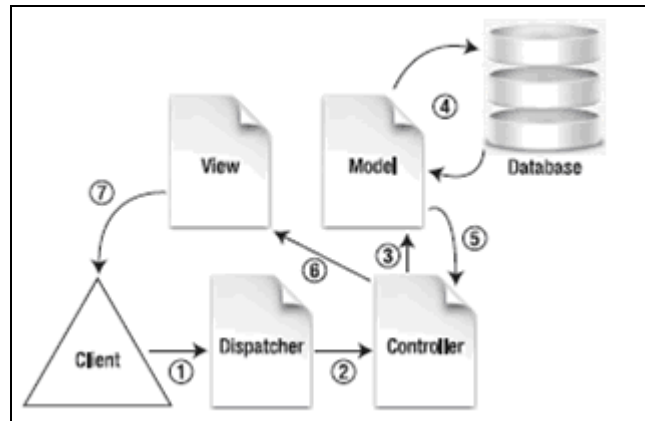


Figura 7 - Flujo MVC utilizado por Cake

1. El cliente envía un page request a la aplicación, por tipear una URL o por clicar algún enlace.
2. El dispatcher parsea la URL y determina cual controller ejecutar.
3. La función en el controller puede necesitar manejar más datos que solo los parámetros enviados por el dispatcher. Puede enviar database request al modelo.
4. El modelo determina como interactuar con la base de datos.
5. Una vez que el modelo ha terminado su interacción, retorna su salida al controller.
6. El controller procesa los datos y delega la salida a la vista.
7. La vista agrega diseño para mostrar datos y envía su salida al browser del cliente.

Ventajas:

- El beneficio de usar MVC para desarrollar Web sites es que funciones repetidas pueden ser separadas, permitiendo así rápida edición. Esto ayuda en el debug.
- *Operaciones CRUD*: Cake ofrece generación automática de CRUD basada en el esquema de la base de datos con validación integrada de campos.
- *Helpers*: Cake viene con HTML, Ajax y JavaScript helpers entre otros que permiten la creación de vistas mucho mas fácil.
- *Lista de Control de Acceso*: Cake permite configurar permisos para usuarios de una manera muy fácil.
- *Gran Comunidad*: Cake tiene una activa gran comunidad, que puede ser de gran ayuda.

6.3 Implementación

Diagrama Entidad Relación

El siguiente diagrama describe las tablas utilizadas para el desarrollo de la aplicación.

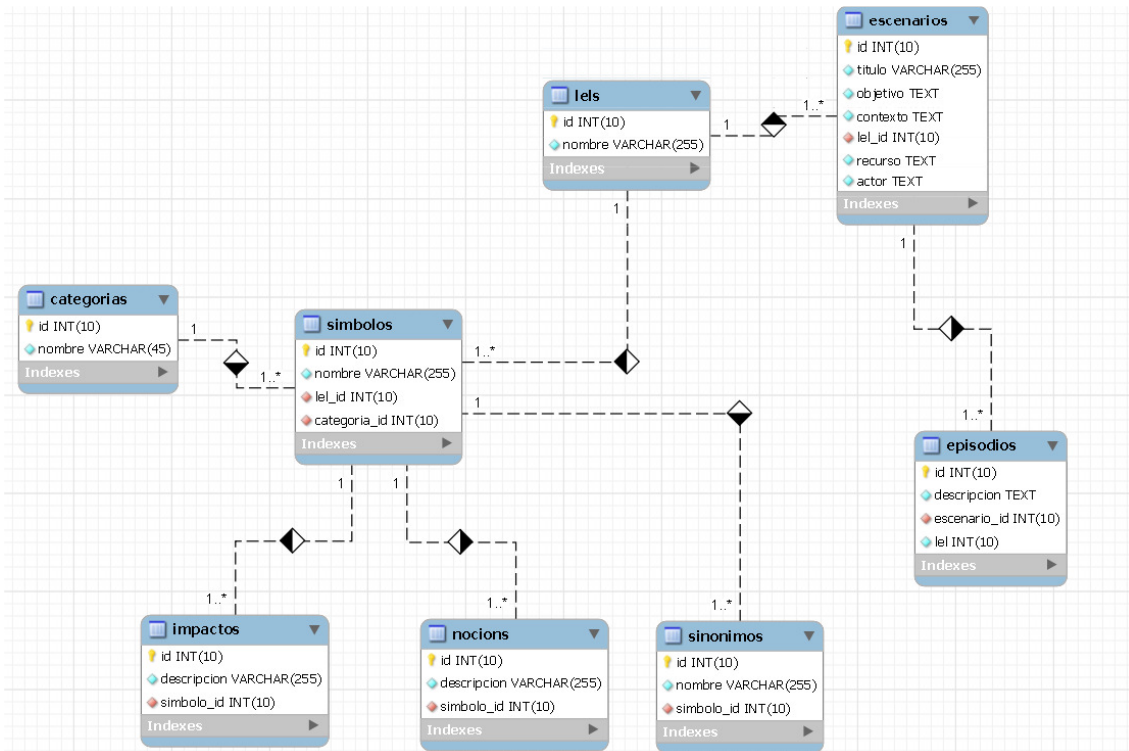


Figura 8 - Diagrama Entidad Relación de la aplicación desarrollada

6.4 Manual del usuario

En esta sección describiremos los pasos más importantes para la creación del LEL y la generación de escenarios. Para esto ejemplificaremos con la realización del LEL que utilizamos en el capítulo 5.

Referencia:

- (botón View) Permite visualizar el elemento asociado.
- (botón Edit) Permite pasar a editar el elemento asociado.
- (botón Delete) Permite eliminar el elemento asociado.
- (botón Escenarios) Permite generar escenarios para el LEL asociado.
- (botón Add) Permite agregar elementos.

Generación Del LEL y Escenarios			
Categorías Lels Simbolos Nociones Impactos Escenarios Episodios			
Lels			
Id	Nombre	Actions	
1	Item LEL	  	Crear
2	Ejemplo Tesis	  	Crear

<< previous next >>



Figura 9 - Herramienta de derivación - Pantalla inicial

En la figura 9 vemos lo que sería la pantalla inicial para empezar a trabajar. Vemos opciones de menús que estarán en cualquier página de la aplicación para brindar acceso a los distintos componentes de LEL y escenarios.

Los distintos componentes tanto del LEL (símbolos, nociones, impactos, etc.) como de los Escenarios (actores, episodios, etc.) siempre pueden ser editados, eliminados, etc. Ahora lo que haremos será crear un nuevo LEL, para esto haremos click en el (botón Add).

Generación Del LEL y Escenarios	
Categorías Lels Simbolos Nociones Impactos Escenarios Episodios	
Add Lel	
Nombre	<input type="text" value="Ejemplo LEL"/>
<input type="button" value="Submit"/>	
List Lels	List Simbolos New Simbolo

Figura 10 - Herramienta de derivación - Formulario de creación de LEL

La figura 10 nos muestra el formulario de creación de un LEL para esto simplemente hay que completar un nombre para el LEL.

Generación Del LEL y Escenarios			
Categorías Lels Simbolos Nociones Impactos Escenarios Episodios			
Lels			
Id	Nombre	Actions	
1	Item LEL	  	Crear
2	Ejemplo Tesis	  	Crear
4	Ejemplo LEL	  	Crear

<< previous next >>

Figura 11 - Herramienta de derivación - Visualización del LEL

En la figura 11 vemos al LEL que ingresamos anteriormente. El próximo paso es el más importante donde agregaremos los símbolos intervinientes en el LEL. Para esto iremos a la edición del LEL (botón Edit).

Generación Del LEL y Escenarios

Categorías: Lels Simbolos Nociones Impactos Escenarios Episodios

Edit Lel

Nombre
Ejemplo LEL

Submit

Delete List Lels

Simbolos

Id	Nombre	Categoria	Actions
Nombre:			
Administradora			
Clear Link Simbolos Disponibles			
Categoria:			
Sujeto			
Add Simbolo			

Figura 12 - Herramienta de derivación - Formulario de creación de símbolos

En la figura 12 vemos un formulario para agregar nuevos símbolos, en el que tendremos que indicar un nombre y una categoría. En este caso estamos creando un símbolo llamado “Administradora” que es de categoría Sujeto.

Id	Nombre
61	Administradora
62	Solicitante
63	Planilla de adhesion
64	Cuota de ingreso
65	Ingresar
66	Iniciar

Terminado

Figura 13 - Herramienta de derivación - Símbolos disponibles

En la figura 13 ya hemos creado algunos símbolos más, y se muestra un popup el que nos será de utilidad para saber los símbolos disponibles que tenemos para realizar links. A este popup se accede desde el botón Símbolos Disponibles.

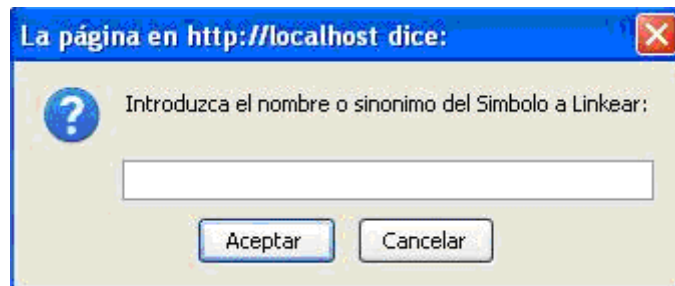


Figura 14 - Herramienta de derivación - Creación de un link

En la figura 14 estamos creando un símbolo llamado “Solicitar Ingresar” en el cual crearemos un link al símbolo “Ingresar”. Para realizar esto seleccionaremos la palabra ingresar y luego haremos click en el botón Crear Link, se abrirá el popup que se muestra en la figura, aquí podremos escribir el nombre o algún sinónimo del símbolo al que queremos hacer el link. En este caso pondremos “Ingresar”, luego Aceptar y posteriormente click en el botón “Add Simbolo”.

En la figura 15 podemos ver el símbolo que agregamos anteriormente, y vemos como se muestra que existe un vínculo en la palabra “ingresar”. Esto nos permitirá navegabilidad entre símbolos del LEL de una manera fácil.

Generación Del LEL y Escenarios

Categorías Lels **Símbolos** Nociones Impactos Escenarios Episodios

Edit Lel

Nombre
Ejemplo LEL

Submit

Delete List Lels

Símbolos

Id	Nombre	Categoría	Actions
61	Administradora	Sujeto	
62	Solicitante	Sujeto	
63	Planilla de adhesion	Objeto	
64	Cuota de ingreso	Objeto	
65	Ingresar	Estado	
66	Iniciar	Estado	
67	Solicitar ingresar	Verbo	

Figura 15 - Herramienta de derivación - Visualización de los símbolos

En el siguiente paso realizaremos la edición de los símbolos que creamos, le configuraremos las nociones y los impactos.

Generación Del LEL y Escenarios

Categorías LELs Símbolos Nociones Impactos Escenarios Episodios

Edit Símbolo

Nombre
Administradora

Clear Link | Símbolos Disponibles

Lel
Ejemplo LEL

Categoría
Sujeto

Submit

Delete | List Símbolos

Sinonimos

Nombre

Add Sinonimo

Nociones

Id	Descripcion	Actions
----	-------------	---------

Clear Link | Símbolos Disponibles

Add Nocion

Impactos

Id	Descripcion	Actions
----	-------------	---------

Clear Link | Símbolos Disponibles

Aplicacion para Trabajo de Tesis

Figura 16 - Herramienta de derivación - Edición de un símbolo

En la figura 16 vemos la pantalla de edición de un símbolo, podemos ver que podemos agregar Sinónimos, Nociones e Impactos. A su vez en cada uno de estos elementos se pueden crear links como vimos anteriormente y cualquier elemento creado siempre puede editarse.

En la figura 17 vemos al símbolo Administradora al que le hemos creado una noción y un impacto. También vemos como en el impacto hemos agregado un link al Símbolo “Solicitante”. Realizaremos la edición para cada uno de los símbolos pertenecientes al LEL.

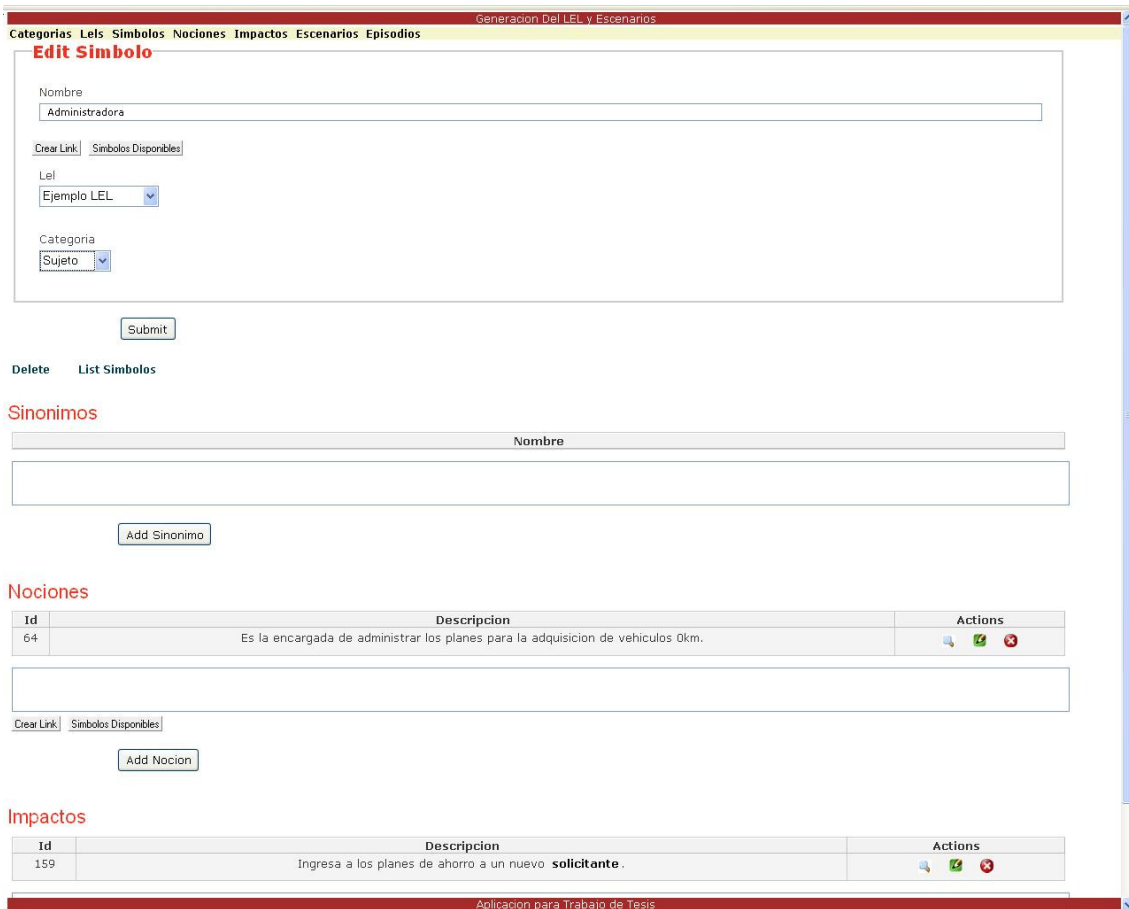


Figura 17 - Herramienta de derivación - Visualización de un símbolo

En la figura 18 vemos la vista en donde se muestran todos los símbolos del LEL con sus respectivos elementos (Nociones, Impactos,...) para poder tener una visión mas general del LEL que acabamos de construir. A esta vista se accede desde el botón View desde la pantalla de LELs.

Categorías Letra Símbolos Nombres Ingresos Escenarios Episodios		
Generación del LEL y Escenarios		
LEL		
Id	4	
Nombre	Ejemplo LEL	
Edt Lel	Delte Lel	Let Lel
		New Lel
Símbolos Asociados		
Id	Nombre	Categoría
61	Administradora	Sujeto
Nombres asociados al Símbolo: Administradora		
Id	Descripción	
64	Es la encargada de administrar los planes para la adquisición de vehículos Okm.	
Ingresos asociados al Símbolo: Administradora		
Id	Descripción	
159	Ingresa a los planes de ahorro a un nuevo solicitante .	
Id	Nombre	Categoría
62	Solicitante	Sujeto
Nombres asociados al Símbolo: Solicitante		
Id	Descripción	
67	Es la persona que desea adquirir un plan de ahorro Okm.	
Ingresos asociados al Símbolo: Solicitante		
Id	Descripción	
161	Solicita planilla de adhesión	
Id	Nombre	Categoría
63	Planilla de adhesión	Objeto
Nombres asociados al Símbolo: Planilla de adhesión		
Id	Descripción	
68	El solicitante completa la planilla de adhesión.	
Ingresos asociados al Símbolo: Planilla de adhesión		
Id	Nombre	Categoría
64	Cuota de ingreso	Objeto
Nombres asociados al Símbolo: Cuota de ingreso		
Id	Descripción	
66	Cantidad fija con que solicitante debe pagar al ingresar al plan de ahorro	
Ingresos asociados al Símbolo: Cuota de ingreso		
Id	Descripción	
162	El solicitante paga la cuota de ingreso al ingresar al plan de ahorro.	
Id	Nombre	Categoría
65	Ingresar	Estado
Nombres asociados al Símbolo: Ingresar		
Id	Descripción	
69	La planilla de adhesión se encuentra en estado ingresado una vez que el solicitante completa dicha planilla.	
Ingresos asociados al Símbolo: Ingresar		
Id	Descripción	
163	Se puede pasar a autorizar el plan.	
Id	Nombre	Categoría
66	Finalizar	Estado
Nombres asociados al Símbolo: Finalizar		
Id	Descripción	
70	La planilla de adhesión se encuentra en estado inicial una vez que al solicitante se le entrega la planilla.	
Ingresos asociados al Símbolo: Inicial		
Id	Descripción	
164	Se puede pasar a ingresar la planilla de adhesión.	
Id	Nombre	Categoría
67	Solicitar ingresar	Verbo
Nombres asociados al Símbolo: Solicitar ingresar		
Ingresos asociados al Símbolo: Solicitar ingresar		

Figura 18 - Herramienta de derivación - Vista de símbolos de un LEL

Ahora que ya tenemos el LEL configurado, podremos aplicar la realización de la derivación de escenarios. Para esto haremos click en el botón escenario correspondiente. Vemos en la figura 19 que luego del click nos avisa que en caso de existir escenarios ya creados para este LEL se reemplazarán.

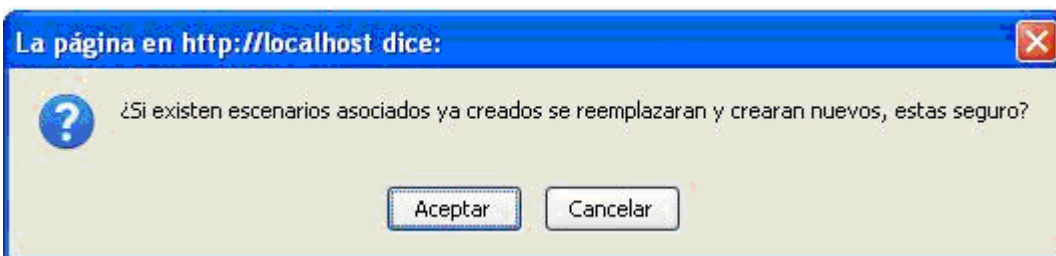


Figura 19 - Herramienta de derivación - Derivación de escenarios




Generación Del LEL y Escenarios						
Categorías Lels Símbolos Nociones Impactos Escenarios Episodios						
Escenarios						
Id	Título	Objetivo	Contexto	Recurso	Actor	Actions
151	Solicitar ingresar	La administradora ingresa un nuevo solicitante a los planes de ahorro . Se puede pasar a autorizar el plan. .	Se debe encontrar en alguno de los siguientes estados: Iniciar ,	Planilla de adhesion , Cuota de ingreso	Solicitante , Administradora	  

Figura 20 - Herramienta de derivación - Escenario derivado

La figura 20 nos muestra la pantalla luego de que se realizo la derivación de escenarios basado en las heurísticas. En este caso se creo un escenario llamado “Solicitar Ingresar” En esta vista preliminar observamos como se generaron el objetivo, contexto, recurso y Actor.







Generación Del LEL y Escenarios		
Categorías Lels Símbolos Nociones Impactos Escenarios Episodios		
Escenario		
Id	151	
Título	Solicitar ingresar	
Objetivo	La administradora ingresa un nuevo solicitante a los planes de ahorro . Se puede pasar a autorizar el plan. .	
Contexto	Se debe encontrar en alguno de los siguientes estados: Iniciar ,	
Lel	Ejemplo LEL	
Recurso	Planilla de adhesion , Cuota de ingreso	
Actor	Solicitante , Administradora	
Edit Escenario	Delete Escenario List Escenarios New Escenario	
Related Episodios		
Id	Descripción	Actions
182	El solicitante pide la planilla de adhesion	  
183	El solicitante completa dicha planilla y la entrega, abonando la cuota de ingreso .	  

Figura 21 - Herramienta de derivación - Vista detallada de un escenario

En la figura 21 podemos ver una vista más detallada que nos permite ver a su vez los Episodios que componen el Escenario. A esta vista se accede desde el botón View del Escenario.

En esta sección hemos mostrado la forma en que se crea un LEL con sus símbolos, impactos, nociones, etc. Hemos visto como crear links entre símbolos y a su vez mostramos como luego de tener el LEL configurado podemos aplicar una derivación para obtener escenarios preliminares para luego en caso necesario poder editarlos.

Capítulo 7 – Conclusiones

En este capítulo se van a desarrollar las conclusiones sobre el trabajo de tesis, como así también los trabajos que quedaron pendientes para su posterior investigación y desarrollo.

Existen varias metodologías que utilizan el concepto de escenarios o sus variantes integrados a estados, en esta sección se describen brevemente sus similitudes y diferencias con la estrategia presentada en esta tesis.

7.1 Trabajos relacionados

Uno de los posibles métodos para conseguir una mayor comprensión y una mayor colaboración entre todos los participantes del proceso de definición de requisitos son los casos de uso. Estos consisten en describir textualmente las interacciones entre un actor y el software. Los casos de uso brindan límites y restricciones sobre el sistema a ser construido. Sin embargo, las descripciones pueden ser poco específicas y esto hace que no se llegue a un único diseño [Wirfs 90].

Los escenarios son parecidos a los casos de uso ya que podemos decir que ambos capturan requisitos funcionales permitiendo manejar la complejidad de grandes sistemas, mejorando la comunicación entre expertos de software y del dominio.

A diferencia de los casos de uso, los escenarios son tipados, es decir que no se refieren a los actores concretos del Universo de Discurso sino a ‘tipo’ de actores. Por ejemplo, los escenarios se refieren a actores como Cliente, Escribano, Cajero y no a Pedro (un caso concreto de Cliente).

Podemos considerar al LEL como una o muchas máquinas de estados al estar conformado por símbolos de categoría “Estado”. El símbolo es el nodo / estado y podemos ver a los impactos como las transiciones entre estos.

En este trabajo de tesis usamos los símbolos de categoría estado para enriquecer el contexto y los objetivos de los escenarios generados. Se realiza la transición de estados al momento de realizar la derivación de los escenarios, valiéndonos de los estados previos y estados posteriores que obtenemos de los impactos.

Whittle [Whittle 06] parte de diagramas de secuencia obtenidos de casos de uso para, generar máquinas de estado. La importancia de esto es que permite automatizar una actividad clave de muchos procesos Object Oriented Analysis and Design (OOAD). Además, al generar máquinas de estado es una manera de simular requerimientos basados en escenarios, permitiendo realizar una validación de requerimientos.

Whittle muestra la forma de ir de escenarios a máquinas de estado, mientras que nuestro trabajo lo hace en sentido inverso, puesto que a partir de los estados (modelados en el LEL) se llega a los escenarios. Consideramos importante nuestro enfoque, puesto que:

1. Los estados surgen naturalmente en el LEL, por lo cual debemos aprovecharlos.
2. Hoy en día, el diseño es de muy alto nivel, y trabajamos con escenarios / use cases que necesitan ser guiados en la construcción y es necesario enriquecerlos puesto que son el producto mas importante que se obtiene en el análisis.

Otro trabajo similar es [Harel 05] en el cual el enfoque sigue la misma línea que [Whittle 06], sin embargo no usa diagramas de secuencia, sino Live Sequence Chart (LSC) que son muy parecidos a los diagramas de UML. Al igual que los diagrama de secuencia, LSC modelan el envío de mensajes a través del tiempo. La principal motivación para el uso de LSC es su poderoso poder expresivo. LSC son una extensión de Message Sequence Chart (MSC), o su variante UML, diagrama de secuencia.

Nuestra tesis se encuentra en la misma línea de [Behrens 02], es decir, obtiene escenarios a partir de maquinas de estados. Sin embargo, nuestra máquina de estado es mas fácil de construir (requiere menos esfuerzo), porque surge naturalmente del LEL. Y los escenarios están "integrados" en el LEL ya que al aplicarle las heurísticas de derivación la información se obtiene del LEL.

Glinz [Glinz 95] también se encuentra en nuestra misma línea. Utiliza los State Charts para poder componer e integrar un modelo de escenarios, otras aproximaciones tratan a los escenarios en forma aislada. En nuestro trabajo de tesis, los escenarios están naturalmente integrados, debido a la estrategia de derivación de escenarios a partir del LEL y a su vez están relacionados entre si porque el LEL esta basado en el principio de circularidad. En términos generales el LEL esta relacionado consigo mismo, por lo cual, los escenarios también están relacionados entre si.

Pudimos observar que tanto el enfoque propuesto en esta tesis, como los enfoques propuestos en los trabajos relacionados son similares. En todas las propuestas existen dos modelos: por un lado casos de uso / escenarios que representan la especificación de comportamiento de un sistema / dominio. Y por otro lado existen maquinas de estados / State Charts que representan la secuencia de transición de estados que un elemento del dominio sufre.

La diferencia es que en nuestra propuesta nosotros obtenemos escenarios enriquecidos a partir de los estados, mientras que en los trabajos relacionados los autores plantean como obtener maquinas de estado a partir de escenarios. Sin embargo, nosotros consideramos más conveniente nuestro enfoque, puesto que los estados surgen naturalmente en el LEL sin esfuerzo alguno. Luego, mediante las heurísticas, obtenemos escenarios enriquecidos. Y son estos escenarios enriquecidos los que el diseñador va a utilizar para hacer un diseño del sistema.

Sin embargo, el sentido inverso demandaría mas esfuerzo, debido a que se necesitaría información de transición de estados al escribir los casos de usos. Demandando un esfuerzo de análisis mayor.

En resumen, entendemos que es mas simple, crear el LEL, el cual tenga estados, y luego, aprovechar esa información y volcarla en escenarios, que hacer el proceso inverso, es decir, escribir escenarios para obtener a partir de ellos maquinas de estado.

Crear el LEL y derivar los escenarios es un proceso creciente en cuanto a complejidad, partimos de algo simple y lo enriquecemos para llegar a algo complejo. Sin embargo el proceso inverso construye primero algo complejo para llegar a una subparte simple.

7.2 Contribución del trabajo

En esta tesis se presenta una estrategia que pretende contribuir y enriquecer la derivación de los escenarios. Para esto utilizamos los símbolos de categoría estado del LEL con el propósito de generar escenarios más completos que describan mejor los requerimientos funcionales. Se agregaron un conjunto de heurísticas y pasos que guían la derivación a partir de los modelos de LEL.

En esta sección listaremos las tareas realizadas en este trabajo junto con un resumen del aporte realizado durante la elaboración del mismo. Podemos mencionar dos tipos de aportes que se hicieron en esta tesis:

- *Aportes teóricos:* El aporte que se introdujo en este trabajo es de suma utilidad por su simplicidad. Vemos que se realizó una contribución importante al utilizar los símbolos de tipo estado porque nos proveen una forma de complementar la derivación de los escenarios.
- *Aporte práctico:* se ha desarrollado la construcción de una herramienta Web que permite editar, registrar y gestionar el LEL y la derivación de escenarios sobre la base de las heurísticas tratadas en la presente tesis.

Como primera tarea, se realizó una investigación sobre el proceso de desarrollo de software y gestión de requerimientos, además del estudio sobre LEL y Escenarios, desde los conceptos más básicos, hasta los más complejos, viendo que con muchos de estos estábamos familiarizados. Se ha investigado y analizado acerca de las limitaciones que se encontraban en la derivación de los escenarios.

Al profundizar sobre los temas generales de LEL y Escenarios, nos permitió experimentar, para poder comprender como se aplican estos conceptos en la realidad. Esto nos sirvió como base para realizar el capítulo 5, el cual consiste en el proceso de construcción del LEL y luego la derivación de los escenarios.

Sobre la base de los resultados obtenidos a partir de los casos estudiados se pueden deducir las siguientes ventajas de la estrategia de derivación de LEL a Escenarios:

- Dada su naturaleza orientada al cliente, tanto de los modelos de escenarios y de LEL como también la estrategia de derivación de los escenarios, hace que sean fáciles de comprender. Luego de una corta explicación se puede comenzar a trabajar inmediatamente.
- El lenguaje natural facilita la validación con los stakeholders.
- Los símbolos clasificados como estado nos brindan información útil que usamos para engrosar el contexto y el objetivo del escenario. Esto nos ayuda a comprender de una mejor manera las restricciones asociadas a un escenario, agregándole detalle a la descripción de los requisitos, permitiendo una mejor comunicación entre los expertos del software y del dominio.

- Con estas nuevas reglas de derivación propuestas en esta tesis no quedan símbolos sin utilizar a la hora de realizar la derivación de LEL a Escenarios. Se logró mejorar la calidad de la información del LEL y refinar el proceso de derivación de los escenarios. En consecuencia se logró un enriquecimiento de las estrategias de derivación propuestas en los enfoques de [Leite 95], [Hadad96] y [Hadad97], especialmente en lo que se refiere al contexto y al objetivo.

Las principales dificultades que se evidenciaron en el uso de la estrategia son:

- Uno de los problemas más significativos con que se lidió fue el nivel de detalle con el que se debe describir el LEL. Según el nivel de detalle se puede aplicar o no la estrategia propuesta. En la sección 4.2 se explicaron las consideraciones que se deben tomar antes de aplicar las heurísticas propuestas en la tesis.
- También es necesario un análisis cuidadoso al momento de realizar la derivación de los escenarios para poder identificar símbolos de tipo estado en los impactos que los generaron.

El objetivo fundamental de la etapa de análisis del dominio es dar un marco para que permita obtener requerimientos claros, de alta calidad y que satisfagan de la mejor manera posible las necesidades del cliente.

En este trabajo quedó evidenciada la cantidad de información que puede obtenerse del dominio utilizando una herramienta tan simple como LEL, cuya construcción no consideramos compleja desde el momento en que está basada en lenguaje natural y ayuda a que las partes que integran un proyecto de software pueden llegar a un entendimiento óptimo.

Claramente la dificultad más grande de la primera etapa del desarrollo de Software es el entendimiento del dominio para localizar los requerimientos. Es un área de investigación difícil porque está sometida al contacto con el cliente. Creemos que un buen entendimiento del proceso de análisis de requerimientos es mucho más importante que las tecnologías subyacentes ya que ellas son volátiles y el análisis no.

Además analizamos las tecnologías que utilizamos para desarrollar la herramienta, focalizándonos al seleccionar las mismas en que sean open source, y por otro lado que el producto final sea lo más amigable y fácil de utilizar por el usuario.

La herramienta permite la creación del LEL y basándose en él permite generar una primera versión de los escenarios involucrados en el proyecto en desarrollo, permitiendo luego depurarlos, para lograr así una versión definitiva.

Existen otras herramientas desktop similares desarrolladas que permiten realizar una primera derivación de escenarios, pero creemos necesario también que la herramienta sea simple de usar y accesible desde cualquier sitio. Es por esto que optamos por el desarrollo de una aplicación Web que permitirá editar, registrar y gestionar el LEL, la derivación de escenarios con la incorporación de la nueva derivación propuesta en esta tesis.

7.3 Trabajos futuros

Los siguientes temas son aquellos en los que creemos que pueden existir puntos de interés para la ampliación de este trabajo:

- Trabajos teóricos:
 - Basándonos en nuestro caso de estudio investigar que sucedería si un requerimiento que pasa por distintos estados, a su vez el responsable de este requerimiento pasa por distintos roles. En ese caso, habría dos máquinas de estados disjuntas, por lo cual, sería interesante analizar como impacta el enfoque propuesto en este caso.

- Trabajos sobre la herramienta implementada:
 - Permitir generar versionado de LEL y Escenarios desde la herramienta implementada, para de esta manera proveer una forma de trazabilidad.
 - Mejorar la Interfaz de Usuario para mejorar la interacción con el usuario y que sea más amigable.

Capítulo 8 – Referencias

- [Antonelli 03] Antonelli L., “Traceability en la elicitación y especificación de requerimientos” Tesis de Magíster en Ingeniería de Software. Universidad Nacional de la Plata 2003.
- [Behrens 02] Behrens, H. “Requirements Analysis and Prototyping using Scenarios and Statecharts”. Proceedings of ICSE 2002 Workshop
- [Bertolami 08] Bertolami M., Oliveros A., “Evaluación Empírica de la Estimación de Tamaño Funcional de Escenarios” 11th. Workshop on Requirements Engineering, 2008.
- [Bézivin 08] Vallecillo-Moreno A., García-Molina J., y Rossi G.. The European Journal for the Informatics Professional Vol.IX, No 2, April 2008, UPGRADE Vol. IX.
- [Boehm 76] Software Engineering. IEEE Transactions on Computers, Diciembre 1976.
- [Chiarle 08] Chiarle L., Martinez A., “Aspectos en el modelo de dominio” – Tesis de grado. Universidad Nacional de la Plata. Marzo 2008
- [Cockton 01] 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.
- [Fairley 87] Fairley, R. Ingeniería del software. McGraw-Hill, 1987.
- [Fleurey 07] Fleurey, Baudry, France, Ghosh A Generic Approach for Automatic Model Composition *MoDELS'07*
- [Glinz 95] Glinz, M. An Integrated Formal Model of Scenarios Based on Statecharts. In Schäfer, W. and Botella, P. (eds.) (1995).
- [Hadad 96] Hadad, G., Kaplan, G., Maiorana, V., Balaguer, F., Oliveros, A., Leite, J.C.S.P., Rossi, G. Informe Técnico: “Léxico Extendido del Lenguaje y Escenarios del Sistema Nacional para la Obtención de Pasaportes”. *Proyecto de Investigación*, Departamento de Investigación, Universidad de Belgrano, Buenos Aires, 1996.
- [Hadad 97] Hadad, G., Kaplan, G., Oliveros, A., Leite, J., “Integración de Escenarios con el Léxico Extendido del Lenguaje en la elicitación de requerimientos: Aplicación a un caso real”. 1997.

- [Harel 05] Harel, D., Kugler, H., Pnueli, A. H.-J. Kreowski et al. (Eds.): Formal Methods (Ehrig Festschrift), LNCS 3393, pp. 309–324, 2005. Springer-Verlag Berlin Heidelberg 2005.
- [IEEE 83] IEEE Std 729 - "Glosario Estándar IEEE de Terminología de Ingeniería del Software", 1983.
- [IEEE 90] IEEE Std 610.12-1990, IEEE Standard Glossary of Software Engineering Terminology.
http://standards.ieee.org/reading/ieee/std_public/
- [Insfrán 02] Insfrán E., Díaz I. y Burbano M.. Modelado de Requisitos para la Obtención de esquemas conceptuales.
- [Jacobson 94] Jacobson, I., Basic use-case modeling, Revista Road, Volumen 1, número 2, Julio-Agosto, 1994, pp 15-19.
- [Jacobson 94b] Jacobson, I., Basic use-case modeling, Revista Road, Volumen 1, número 3, Septiembre-October, 1994, pp 7-9.
- [Jackson 99] Jackson, M., Problem Analysis and Structure, AT&T Research, Florham Park NJ, USA, y consultor independiente, London, England, 1999.
- [Kaplan 00] Kaplan G., Hadad G., Doorn J., Leite J.C.S.P “Inspección del Léxico Extendido del lenguaje”
- [Kotonya 98] Kotonya, G., Sommerville, I., “*Requirements Engineering. Processes and Techniquess*” . John Wiley & Sons, 1998.
- [Leite95] Leite J.C.S.P, Albuquerque Oliveira, A P. *A Client Oriented Requirements Baseline*. Proceedings of RE 95’: Second IEEE International Symposium on Requeriments Engineering. Inglaterra, Marzo 1995.
- [Leite 97] Leite J.C.S.P., Rossi G., et al. Enhancing a Requirements Baseline with Scenarios. Proceedings of RE 97’: International Symposium on Requeriments Engineering, IEEE. Enero 1997.
- [Leite 00] Leite J.C.S.P, Hadad G, Doorn J, Kaplan G. “A Scenario Construction Process” Requirements Engineering Journal, Springer-Verlag, 2000. Vol.5 N1, pp.38-61.
- [Leonardi 01] Leonardi C. “Una Estrategia de Modelado Conceptual de Objetos basada en Modelos de Requisitos en Lenguaje Natural”. Tesis de Magister en Ingeniería de Software. Universidad Nacional de la Plata, 2001.
- [Lientz 81] Lientz B.P., Swanson E. "Problems in application software maintenance", Communications of the ACM 24. 1981

- [Loucopoulos 95] Loucopoulos, P., Karakostas, V., *System Requirements Engineering*, McGraw-Hill, London, 1995
- [Martínez 08] Martínez L. “Componentes MDA para patrones de diseño”. 2008
- [Mauco 97] Mauco, V., Ridao M., del Fresno, M., Rivero, L., Doorn, J., “Ingeniería de Requisitos, Proyecto: Sistema de Planes de Ahorro”, Technical Report, ISISTAN, UNCPBA, Tandil (1997).
- [Naur 69] Naur, Peter and Brian Randell (eds.). 1969. *Software Engineering: Report on a conference sponsored by the NATO Science Committee, Garmisch, Germany, 7th to 11th October, 1968*. Brussels, Scientific Affairs Division, NATO. See Naur et al. 1976.
- [Pilgrim 07] Measuring the Level of Abstraction and Detail of Models in the Context of MDD. MoDELS 2007
- [Pressman 98] PRESSMAN Roger. *Un enfoque práctico*. Cuarta edición. Mexico. Editorial McGraw Hill. 1998.
- [Ramesh 98] Ramesh, B., “Factors Influencing Requirements Traceability Practice”, *Communications of the ACM* Vol 41 Nro 12, Diciembre, 1998, pp 37-44.
- [Rivero 98] Rivero, L., Doorn, J., Del Fresno, M., Mauco, V., Ridao, M., Leonardi, C., “Una Estrategia de Análisis Orientada a Objetos basada en Escenarios: Aplicación en un Caso Real”, *Anais do WER'98 - Workshop en Engenharia do Requisitos*, Maringá, Brasil (1998), pp. 79-90.
- [Rubin 92] Rubin, K. and Goldberg, A. *Object Behavior Analysis*. CACM, Vol.35, No. 9, September 1992.
- [Sutcliffe 01] Sutcliffe, A., “Requirements Engineering for Complex Collaborative Systems”, *Proceedings of Fifth IEEE International Symposium on Requirements Engineering*, Toronto, Canada, 2001, pp. 110-117.
- [TMC 04] “Model Driven Development for J2EE Utilizing a Model Driven Architecture (MDA) Approach - Maintainability Analysis” The Middleware Company 2004.
- [Thomas 05] Thomas, P. “Definición de un Proceso de Elicitación de Objetivos” Tesis de Magíster en Ingeniería de Software. Universidad Nacional de La Plata 2005.
- [Whittle 06] Whittle J., Jayaraman P. “Generating Hierarchical State Machines from Use Case Charts” *IEEE International Requirements Engineering Conference*. 2006.

- [Wirfs 90] Wirfs Brock, R. "Designing Object-Oriented Software". Prentice Hall 1990.
- [Wood 97] Wood L.E., Semi-structured interviewing for user-centered design, Interactions of the ACM, april-may 1997, pp 48-61.