

HERRAMIENTA PARA IMPLEMENTAR LEL Y ESCENARIOS (TILS)

Gustavo Daniel Gil

Director: Lic. Alejandro Oliveros

Co-director: Dr. Gustavo Rossi

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

**Facultad de Informática
Universidad Nacional de La Plata – Argentina
Año 2002**

1. INTRODUCCIÓN.....	1
1.1. MOTIVACIÓN.....	4
1.2. PUBLICACIONES Y TRABAJOS RELACIONADOS.....	5
1.3. ESTRUCTURA DE LA TESIS.....	6
INGENIERÍA DE REQUERIMIENTOS.....	7
2.1. PROCESO DE ANÁLISIS DE REQUERIMIENTOS.....	9
2.2. ÁREAS DE PROBLEMAS.....	13
2.3. TÉCNICAS DE ELICITACIÓN.....	16
3. LÉXICO EXTENDIDO DEL LENGUAJE.....	25
3.1. OBJETIVO.....	25
3.2. CARACTERÍSTICAS DEL LEL.....	26
3.3. PROCESO DE CONSTRUCCIÓN DEL LEL.....	28
3.3.1. Entrevistas.....	29
3.3.2. Generación de la lista de símbolos.....	29
3.3.3. Clasificación de los símbolos.....	30
3.3.4. Descripción de los símbolos.....	30
3.3.5. Validación con los clientes.....	31
3.3.6. Control del LEL.....	31
3.4. CASO DE ESTUDIO DE CONSTRUCCIÓN DEL LEL.....	32
3.4.1. Pasos previos a la construcción del LEL.....	33
3.4.2. Proceso de Construcción del LEL.....	34
3.4.3. Entrevistas y generación de símbolos.....	34
3.4.4. Clasificación y descripción de símbolos.....	36
3.4.5. Validación con los clientes.....	41
3.4.6. Control del LEL.....	42
3.4.7. Conclusiones.....	42
4. ESCENARIOS.....	43
4.1. INTRODUCCIÓN.....	43
4.2. MODELOS EXISTENTES.....	44
4.2.1 Uses Cases de Jacobson.....	44
4.2.2. OBA.....	45
4.2.3. ICM, una nueva propuesta [Potts 95].....	45
4.2.4. El modelo de Carroll.....	46
4.2.5. La propuesta de Booch.....	46
4.2.5. EL enfoque de Escenarios de Leite.....	47
4.2.5. Comparación de los distintos modelos.....	48
4.2.6. Uso de Escenarios en la Practica.....	49
4.3. ESQUEMA DE DESCRIPCIÓN DE ESCENARIOS.....	51
4.4. PROCESO DE CONSTRUCCIÓN DE ESCENARIOS.....	53
4.4.1. Identificación de los actores de la aplicación.....	55
4.4.2. Generación de la lista de Escenarios candidatos, a partir de los actores principales.....	56
4.4.3. Descripción de los Escenarios candidatos, provenientes de los actores principales.....	56
4.4.4. Ampliación de la lista de Escenarios candidatos, a partir de los actores secundarios.....	57
4.4.5. Descripción de los Escenarios candidatos, provenientes de los actores secundarios.....	57
4.4.6. Revisión de los Escenarios.....	57
4.4.7. Validación de Escenarios.....	58
4.5. ALGORITMO PARA GENERAR ESCENARIOS.....	59
5. HERRAMIENTA PARA IMPLEMENTAR LEL Y ESCENARIOS (TILS).....	61
5.1. GENERALIDADES.....	61

5.1.1. Motivación.....	61
5.1.2. Implementación.....	62
5.2. FUNCIONALIDAD DE TILS.....	63
5.2.1 El entorno de trabajo.....	63
5.2.2. Proyectos.....	65
5.2.3. Universo del discurso.....	74
5.2.4. Versiones.....	75
5.2.5. Construcción del LEL.....	78
5.2.6. Construcción de Escenarios.....	90
5.2.7. Exportar Documentos.....	99
5.2.8. El Navegador.....	100
5.3 ARQUITECTURA DE TILS.....	107
5.3.1 Introducción.....	107
5.3.2 Universo del Discurso.....	108
5.3.3. Lista de Símbolos.....	110
5.3.4. Descripción de Símbolos.....	110
5.3.5. Lista de Escenarios.....	117
5.3.6. Descripción de Escenarios.....	117
6. UTILIZACIÓN DE LA HERRAMIENTA.....	123
6.1. CASO DE USO: BANCO DE SANGRE.....	123
6.1.1. Generación de Escenarios.....	124
6.1.2. Identificación de los actores de la aplicación.....	125
6.1.3 Generación de la lista de Escenarios candidatos.....	126
6.1.4. Descripción de los Escenarios candidatos.....	128
6.1.5. Vista gráfica de la construcción de un escenario.....	132
6.2. CONCLUSIONES.....	133
7. CONTRIBUCIONES.....	135
8. REFERENCIAS.....	138

1. INTRODUCCIÓN.

No existe un único enfoque para mejorar el mal del software. Sin embargo, mediante la combinación de métodos completos para todas las fases del desarrollo del software, mejores herramientas para automatizar estos métodos, bloques de construcción más potentes para la implementación del software, mejores técnicas para garantía de calidad del software y una filosofía predominante para coordinación, control y gestión, podemos conseguir una disciplina para el desarrollo del software – una disciplina llamada ingeniería del software.

Roger S. Pressman

La ingeniería de software abarca un conjunto de tres elementos claves [Pressman]: 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.

De acuerdo con Boehm [Boehm79], la ingeniería de Software incluye la aplicación práctica del conocimiento científico en el diseño y construcción de programas para computadoras y la documentación asociada requerida para desarrollarlos, operarlos y mantenerlos. Otra definición de Ingeniería de Software es la entregada en el glosario de la IEEE, titulado "Standar Glossary of Software Engineering Terminology" [IEEE 90], en el cual se define como el enfoque sistemático para el desarrollo de operación, mantenimiento y eliminación del software donde software se define como aquellos programas, procedimientos, reglas y documentación posible asociada con la computación, así como los datos pertenecientes a la operación de un sistema de cómputo.

La Ingeniería de Software surge como disciplina en la década de los sesenta, cuando se hacen presentes los graves problemas existentes en la producción y, sobre todo, mantención de software, situación que se conoce como la "crisis del software". Las prácticas "artesanales" de programación y la escasa documentación que se utilizaban en ese entonces hicieron crisis y determinaron que se pensara en el desarrollo de software como un problema que necesariamente debería ser abordado de manera más sistemática. El modelo básico de esta sistematización corresponde a una variación de un modelo desarrollado en la década de los treinta en los Laboratorios Bell, método que en el ámbito del software se conoce como el ciclo de vida tradicional. Históricamente han surgido varios enfoques que buscan abordar de manera sistemática, la planificación, análisis, diseño e implementación de los proyectos de desarrollo de software, sean estos de gran escala y pequeñas aplicaciones, software a medida o productos de software. Cada uno de estos enfoques tiene su raíz en las preconcepciones dominantes en su época y, sobre todo, en la búsqueda incesante de mejoras a los enfoques precedentes.

Una constante que se observa en esta búsqueda, es que siempre incluyen una fase para alcanzar un acuerdo explícito sobre responsabilidades del cliente y del desarrollador, fase que conocemos como Especificación de Requerimientos.

Entre las tareas que hay que realizar en esta fase, está la definición de los requerimientos del software y del sistema del que el software forma parte, esta tarea debe realizarse al comienzo del proyecto, pero el principal problema que se nos presenta es que, en estos momentos iniciales, es difícil tener una

idea clara (o al menos, es difícil llegar a expresarla), de cuáles son los requerimientos del sistema y del software, y llegar a comprender en su totalidad la función que el software debe realizar. Por esto, algunos de los modelos de ciclo de vida proponen enfoques cíclicos de refinamiento de los requerimientos o incluso de todo el proceso de desarrollo de software.

El análisis de requerimientos es el primer paso en el proceso de ingeniería del software. Es aquí donde se refina la declaración general del ámbito del software en una *especificación* concreta que se convierte en la base de todas las actividades de ingeniería del software que siguen cumpliendo con los siguientes dos roles [Norris]:

- Proporciona la primera entrada para la fase de diseño.
- Da un lineamiento contra el cual las pruebas de aceptación son llevadas a cabo.

La forma de especificar un sistema tiene una gran influencia en la calidad de la solución implementada finalmente. Es ampliamente aceptado que el costo de corregir-modificar un sistema después de su instalación, o incluso después de las primeras etapas de la fase de diseño, es mucho mayor que el costo de preparar una especificación de requerimientos inicial [Davis].

Tradicionalmente los ingenieros de software han venido trabajando con especificaciones incompletas, inconsistentes o erróneas, lo que invariablemente lleva a la confusión y a la frustración en todas las etapas del ciclo de vida. Como consecuencia de esto, la calidad, la corrección y la completitud del software disminuyen.

Existen varios enfoques para analizar los requerimientos, dependiendo del ciclo de vida de desarrollo que se esté utilizando. Cada enfoque implica diferentes necesidades en términos del tipo de información que se captura. El diseño centrado en el punto de vista del usuario para definir los requerimientos es el enfoque al que adhieren recientes metodologías, por ejemplo en [Hsia94], [Rubin92] y [Jacobson92], que proponen la especificación de "casos de uso" los cuales son una descripción de las acciones de un sistema desde el punto de vista del usuario, los cuales complementan satisfactoriamente las características que propone el mencionado enfoque. Durante esta etapa la interacción con el usuario es indispensable, por lo cual estas herramientas deben ser relativas al lenguaje que ellos manejan para facilitar la comunicación.

En [Leite90], [Leite95], [Leite97] se propuso una metodología que propone trabajar con una documentación integrada en una estructura llamada Requirements Baseline, que acompaña al proceso de desarrollo de software. El elemento central de esta metodología es el LEL (Language Extended Lexicon) para modelar el vocabulario de un macrosistema y propone el uso de Escenarios para representar el comportamiento.

1.1. Motivación.

La Ingeniería de Requerimientos es el primer paso esencial para entregar lo que el cliente desea, ya que enfoca un área fundamental: la definición de lo que se desea producir. Como ya se mencionó, proporciona la primera entrada para la fase de diseño y da un lineamiento contra el cual las pruebas de aceptación son llevadas a cabo.

Se debe involucrar efectivamente a los usuarios, para conseguir identificar necesidades y/o problemas específicos y se puedan establecer mecanismos de resolución adecuados y apoyar cada una de las fases en sólidos principios de comunicación humana. La ingeniería de software como disciplina ha evolucionado significativamente en lo que se refiere a modelos conceptuales y herramientas de trabajo, que hacen del proceso de desarrollo y mantenimiento de software una actividad cada vez menos dependiente del arte de quienes llevan a la práctica un diseño elaborado.

Un enfoque novedoso para sistematizar el proceso de análisis de requerimientos, se propone en **A client oriented requirements baseline** [Leite 90], que está compuesto por el LEL (Language Extended Lexicon) que permite representar y documentar, con tecnología hipertextual un conjunto de símbolos que representan el lenguaje de la aplicación y por los Escenarios que hacen una descripción parcial del comportamiento de la aplicación en un momento específico.

El objetivo principal de este Trabajo de Tesis esta centrado en la creación de una herramienta automatizada para soportar este enfoque, es decir, que permita implementar el LEL, con tecnología de hipertexto, que soporte las relaciones dinámicas entre los símbolos. También realizará una derivación de escenarios sobre la base de las heurísticas desarrolladas en el trabajo de investigación de [Hadad96], cuyas conclusiones denotan la necesidad de una herramienta de este tipo.

Con el propósito de asistir al desarrollador en el proceso de documentación de requerimientos, la herramienta también incorpora un editor de texto para documentar el Universo del Discurso, crea link a símbolos y escenarios en forma automática, realiza controles de integridad cuando se modifica o elimina algún símbolo o escenario existente, permite consultas de símbolos por categoría y estados, administra distintas versiones para un proyecto, exporta la información de la versión a formato html para que pueda ser recorrida por medio de navegadores de internet ordinarios o en el navegador incorporado en ella, todas estas prestaciones, se analizara en detalle en el capítulo 5 de esta tesis.

Este trabajo se centró en la fase de elicitación de requerimientos, utilizando una metodología basada en el uso del Léxico Extendido del Lenguaje (LEL) y Escenarios. Utilizó como caso de estudio, para ejemplificar y verificar las heurísticas existentes, el Sistema de Registración y Producción del Instituto de Hemoterapia de la provincia de Buenos Aires, el cual resume una experiencia en

la que el Lenguaje del dominio contiene un gran volumen de palabras de uso muy específico que se podría denominar “lenguaje técnico”, de difícil comprensión para los desarrolladores.

1.2. Publicaciones y trabajos relacionados.

Los siguientes artículos publicados y trabajos de investigación son algunos de los resultados obtenidos respecto al tema de esta tesis:

Director en el Trabajo de Investigación “ESTUDIO DE UN CASO REAL APLICANDO LEL Y ESCENARIOS EN EL ANÁLISIS ORIENTADO A OBJETOS”.

Consejo de Investigaciones de la UNSa – Nota Nro. 113 C.I./2000

Período: 01/01/2000 al 31/12/2000.

“Léxico Extendido del Lenguaje y Escenarios del Instituto de Hemoterapia de la Provincia de Buenos Aires”. CIUNSa – Consejo de Investigación de la Universidad de Salta, 2000. Gil, Gustavo, Arias Figueroa, D.

“Gestión General de un Banco de Sangre. Universo del Discurso”, CIUNSa – Consejo de Investigación de la Universidad de Salta, 2000.

Gil, Gustavo, Arias Figueroa, D.

“PRODUCCIÓN DEL LEL EN UN DOMINIO TÉCNICO. INFORME DE UN CASO”. Presentado en WER 2000 – Workshop en Ingeniería de Requerimientos – Río de Janeiro – Brasil – 13 y 14 de julio del 2000.

L.I.D.T.I. – Gustavo D. Gil – Daniel Arias Figueroa – Alejandro Oliveros

“ADQUISICIÓN DE REQUISITOS USANDO LEL Y ESCENARIOS”. 2das. JORNADAS DE INFORMÁTICA DEL NEA. Del 25 al 27 de octubre de 2001 - Corrientes, Argentina. Invitado como Disertante.

“TOOLS FOR THE IMPLEMENTATION OF LEL AND SCENARIOS (TILS)”. Presentado en el marco de la 31 JAIIO – ASSE 2002 (Simposio Argentino de Ingeniería de Software) Santa Fe – Argentina – 9 al 11 de septiembre de 2002.

Gustavo D. Gil – Daniel Arias Figueroa – Loraine Gimson.

“EVALUACIÓN DE UNA HERRAMIENTA PARA IMPLEMENTAR LEL Y ESCENARIOS”. Presentado en el marco del CACIC 2002 (VIII Congreso Argentino de Ciencias de la Computación). Buenos Aires – Argentina. 15 al 18 de octubre de 2002.

Gustavo D. Gil – Daniel Arias Figueroa – Loraine Gimson.

Director en el Trabajo de Investigación “LEL Y ESCENARIOS EN EL ANÁLISIS Y DISEÑO ESTRUCTURADO”.

Consejo de Investigaciones de la UNSa – Nota Nro. 502 C.I./2002

Período: 01/07/2002 al 30/06/2003.

1.3. Estructura de la Tesis.

La tesis se divide básicamente en seis grandes partes: Introducción, Estado del Arte: Ingeniería de requerimientos, Léxico Extendido del Lenguaje, Escenarios, Funcionalidad y Arquitectura de la Herramienta, Utilización de la Herramienta y Conclusiones.

A continuación se describe las restantes partes de lo que sigue a esta tesis.

ESTADO DEL ARTE

Que incluyen los siguientes capítulos:

El capítulo 2, llamado "INGENIERÍA DE REQUERIMIENTOS", en donde se analiza algunos modelos para el análisis de requerimientos, áreas de problemas y técnicas de elicitación.

En los capítulo 3 y 4, "LA INGENIERÍA DE SOFTWARE" se describe cada uno de los pilares del modelo propuesto: Léxico Extendido del Lenguaje y Escenarios, respectivamente.

FUNCIONALIDAD Y ARQUITECTURA DE LA HERRAMIENTA

En capítulo 5 se describe la funcionalidad y arquitectura de la herramienta desarrollada, que automatiza el modelo presentado.

UTILIZACIÓN DE LA HERRAMIENTA Y CONCLUSIONES

En el capítulo 6 se muestra la utilización de la herramienta, se realiza una evaluación contextualizada, contrastando un caso de estudio realizado con la herramienta para implementar LEL y escenarios, con otro desarrollo echo manualmente.

Finalmente se presentan las conclusiones a esta propuesta y experiencia.

CONTRIBUCIONES

Por último, en el capítulo 7 se describe las contribuciones de esta tesis y se dejan bosquejadas algunas posibles líneas de investigación futuras.

2. 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. Existen varias definiciones de requerimientos, de entre las cuales podemos citar las siguientes:

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. Rrepresentación documentada de una condición o capacidad como en 1 o 2.

Según Zave:

- Rama de la ingeniería del software que trata con el establecimiento de los objetivos, funciones y restricciones de los sistemas software.
- Asimismo, se ocupa de la relación entre estos factores con el objeto de establecer especificaciones precisas.

Según Boehm:

- Ingeniería de Requerimientos es la disciplina para desarrollar una especificación completa, consistente y no ambigua, la cual servirá como base para acuerdos comunes entre todas las partes involucradas y en dónde se describen las funciones que realizará el sistema.

Según Loucopoulos:

- Trabajo sistemático de desarrollo de requisitos, a través de un proceso iterativo y cooperativo de análisis del problema, documentando los resultados en una variedad de formatos y probando la exactitud del conocimiento adquirido.

Según Leite:

- Es el proceso mediante el cual se intercambian diferentes puntos de vista para recopilar y modelar lo que el sistema va a realizar. Este proceso utiliza una combinación de métodos, herramientas y actores, cuyo producto es un modelo del cual se genera un documento de requerimientos.

Esta tesis adhiere a esta última definición [Leite89]. Se desarrollan sistemas de software para satisfacer una necesidad percibida por un cliente. Pueden formularse las necesidades reales del cliente como requerimientos. Los requerimientos son desarrollados conjuntamente por el cliente, usuario y diseñadores del sistema de software. La ingeniería de requerimientos es un proceso de descubrimiento, refinamiento, modelización, especificación y validación de lo que se desea construir. En este proceso tanto el cliente como el analista juegan un papel muy importante.

Cuando nos encontramos al frente de un proyecto de desarrollo de sistemas es importante dejar claramente definidos los requerimientos del software, en forma consistente y compacta, esta tarea es difícil básicamente porque consiste en la traducción de unas ideas vagas de necesidades de software en un conjunto concreto de funciones y restricciones. Además el analista debe extraer información dialogando con muchas personas y cada una de ellas se expresará de una forma distinta, tendrá conocimientos informáticos y técnicos distintos, y tendrá unas necesidades y una idea del proyecto muy particulares.

Es justamente este último punto uno de los que se atacará en esta tesis: que el usuario y el desarrollador compartan el mismo lenguaje asegura la comunicación entre ambos. [Leite89] sugiere que en particular el uso del lenguaje propio del usuario mejora considerablemente esta comunicación. En el proceso de Ingeniería de Requerimientos la validación de los diferentes productos requiere una fuerte interacción con el usuario [Loucopoulos], la que se ve facilitada por el vocabulario común usuario-desarrollador. Las diferentes representaciones que se construyen en el proceso de desarrollo de software encuentran en el vocabulario del usuario un marco referencial que permite al desarrollador, obtener un vocabulario de trabajo que es un subconjunto de la terminología del cliente, lo que a su vez facilita el acceso a la documentación por parte de todos los participantes en el desarrollo. En muchos casos es difícil especificar los requerimientos de un problema, con la mayor calidad posible en una etapa temprana del proceso de desarrollo.

La construcción de prototipos constituye un método alternativo, que da como resultado un modelo ejecutable del software a partir del cual se pueden refinar los requerimientos. Para poder construir el prototipo se necesitan técnicas y herramientas especiales.

El análisis de requerimientos establece el proceso de definición de requerimientos como una serie de tareas o actividades mediante las cuales se

busca ganar conocimientos relevantes del problema, que se utilizarán para producir una especificación formal del software necesario para resolverlo. En este proceso se deben conciliar diferentes puntos de vista y utilizar una combinación de métodos, personas y herramientas. El resultado final constituye la documentación de los requerimientos. Éstos deben expresarse de forma clara y estructurada de manera que puedan ser entendidos tanto por expertos como por el usuario, quien deberá participar en la validación, [Rumbaugh].

Como resultado del análisis se desarrolla la especificación de requerimientos del software. La revisión es esencial para asegurar que el realizador del software y el cliente tengan la misma percepción del sistema.

Una vez finalizado nuestro proyecto de desarrollo de sistemas, y contando con un buen análisis de requerimientos, podremos evaluar la calidad del producto terminado.

2.1. Proceso de Análisis de Requerimientos.

El proceso del establecimiento de requerimientos de un sistema de software, como ya mencionamos, es el primer paso esencial en entregar lo que el cliente desea. A pesar de esto, la insuficiencia de tiempo y esfuerzo son a menudo encontrados en esta actividad y existen pocos métodos sistemáticos para soportarlo. Entre los métodos conocidos se puede citar a los siguientes:

Para Pressman, en el proceso de análisis de requerimientos del software se puede identificar cinco tareas o etapas fundamentales:

1. *Reconocimiento del problema*

Se deben de estudiar inicialmente las especificaciones del sistema y el plan del proyecto del software. Realmente se necesita llegar a 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. Todas las tareas anteriores conducen fácilmente a la determinación del problema de forma sintetizada.

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 intenta 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.

Método CORE

El método Controlled Requirements Expression (CORE) [Norris] es un conjunto de notaciones textuales y gráficas, con guías especificadas para la captura y validación de requerimientos del sistema, en las etapas iniciales del diseño del sistema. CORE ha sido, por tradición, pensado como puramente una técnica de captura y análisis de requerimientos (RCA), aunque soporta algunos aspectos de diseño tales como estructuras de datos. CORE está basada en el principio de primero definir el problema a ser analizado (definición del problema), y luego dividirlo en unidades o puntos de vista a considerar.

El método CORE consiste en siete etapas. Cada una produce salidas que alimentan a la etapa subsecuente como entrada o que forman parte de la especificación de requerimientos final. CORE pretende examinar el sistema y su ambiente en un número de niveles, con detalles más finos progresivamente en cada nivel. Las siete etapas se presentan a continuación:

1. *Definición del problema.*

El propósito de la definición del problema es identificar los límites del mismo. Contiene detalles de los objetivos de la empresa de los usuarios del sistema, la base para la necesidad de un nuevo sistema, limitaciones de costo y tiempo, y quién va a ser el responsable de la revisión y aceptación de los resultados finales.

2. *Estructuración del punto de vista.*

El propósito de esta etapa es descomponer el ambiente del sistema en los elementos para que el sistema propuesto pueda ser analizado desde los puntos de vista de todas las entidades que se comunican con él, la más importante de las cuales son los usuarios. Durante esta etapa, todas las entidades que son fuentes potenciales de información deben ser identificadas.

3. *Colección tabular.*

Esta etapa es cuando la información sobre los flujos de datos entre los puntos de vista y el procesamiento de éstos son reunidos. Esto ayuda a establecer la totalidad y consistencia.

4. *Estructuración de datos.*

En la etapa previa, los elementos de información que pasan entre los puntos de vista son referidos por sus nombres generales. En esta etapa, se da una vista más cercana al contenido, a la estructura y a la derivación de datos, al producir diagramas de estructura de datos.

5. *Modelación individual de puntos de vista.*

Esta etapa puede dividirse en dos partes. Lo único concerniente a la primera es convertir las TCF'S en una notación diferente para producir los diagramas individuales del modelo de punto de vista. La segunda parte se refiere a agregar alguna información nueva perteneciente a flujos de datos internos, control de acciones y tiempo de acciones.

6. *Modelación combinada de punto de vista.*

Esta etapa facilita el análisis de una secuencia de eventos de más de un punto de vista. Cada diagrama de modelo combinado de punto de vista producido durante esta etapa es una representación del procesamiento de información que ocurre entre puntos de vista.

7. *Análisis de restricciones.*

En esta etapa, se consideran restricciones adicionales tales como desempeño y seguridad. Éstas pueden afectar los diagramas de puntos de vista ya producidos. Las restricciones se documentan en una especificación de restricción del sistema.

Los métodos analizados, como gran parte de los existentes, conciernen a la producción del documento de especificación de requerimientos y no direcciona el proceso de captura de requerimiento en detalle, a menudo sólo proporcionan guías simples de técnicas de entrevista y algunas veces áreas claves que deben investigarse pero que a menudo se olvidan. Cada método individualmente, dará diferentes soluciones para diferentes proyectos, incluyendo aquellos casos en los que el dominio y el área del problema son el mismo. Por esta razón, considero que no existe un modelo de proceso ideal para la Ingeniería de Requerimientos; encontrar el método o la técnica perfecta es una ilusión, pues cada método y técnica ofrece diferentes soluciones ante un problema, no obstante la aplicación de alguno de ellos nos ayuda a la producción de especificación de requerimientos.

El proceso de ingeniería de requerimientos en el que se basara este trabajo de tesis es el que propone Loucopoulos [Loucopoulos], en el que se

plantea que en esta fase hay que considerar por lo menos tres aspectos fundamentales:

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

Esto nos llevaría a simplificar el proceso a tres etapas para obtener los requerimientos del problema que estamos atacando, estas etapas son las siguientes [Loucopoulos]:

- **Elicitación de requerimientos**
- **Especificación**
- **Validación**

Elicitación de requerimientos

El propósito de la elicitación de requerimientos es ganar conocimientos relevantes del problema, que se utilizarán para producir una especificación formal del software necesario para resolverlo. “Un problema puede ser definido como la diferencia entre las cosas como se perciben y las cosas como se desean”¹. Aquí vemos nuevamente la importancia que tiene una buena comunicación entre desarrolladores y clientes; de esta comunicación con el cliente depende que entendamos sus necesidades.

Al final de la fase de análisis de requerimientos el analista podría llegar a tener un conocimiento extenso en el dominio del 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.

Validación

La validación es el proceso que certifica que el modelo de los requerimientos es consistente con las intenciones de los clientes y los usuarios; ésta 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 necesidad de validación aparece cuando:

- Se incorpora una nueva pieza de información al modelo actual.

¹ Gause & Weinberg. Turn your lights on, 1989

- Cuando diferentes piezas de información se incorporan en un todo coherente.

La validación no sólo se aplica al modelo final de los requerimientos, sino también a los modelos intermedios.

En la figura 1 podemos ver el esquema del proceso planteado

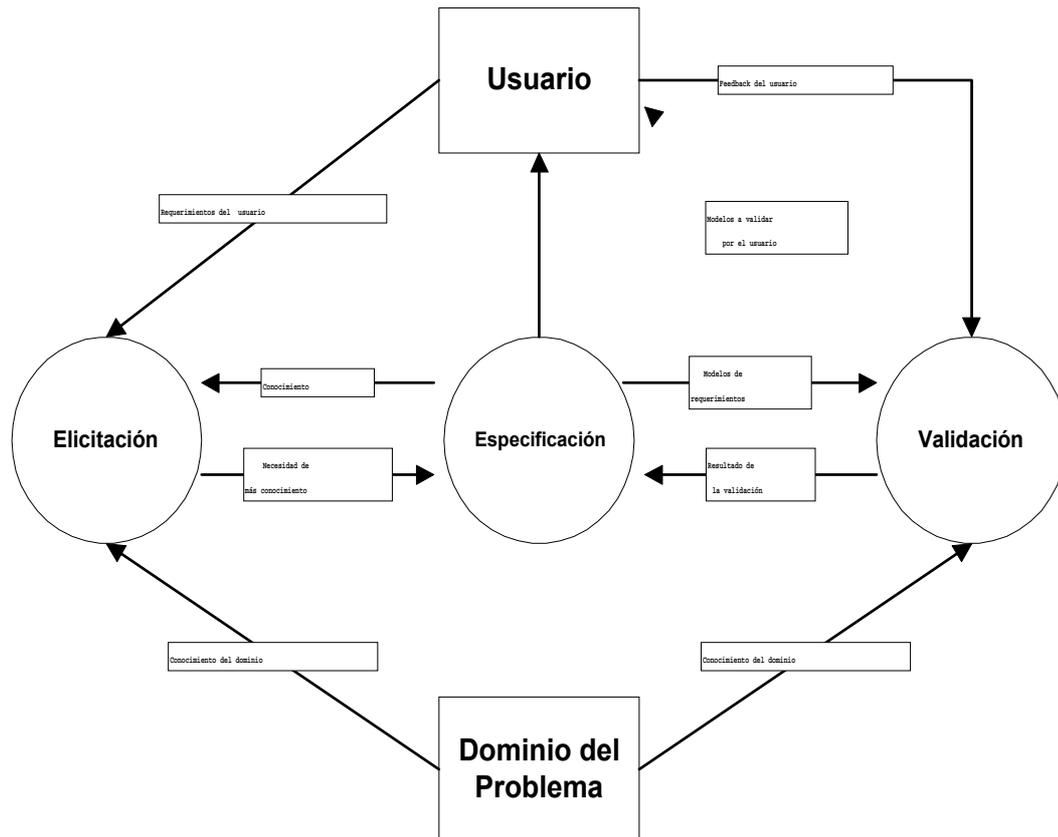


Figura1: Esquema del proceso de ingeniería de requerimientos (Fuente: Loucopoulos)

2.2. Áreas de problemas.

Uno de los problemas principales que podemos encontrar a la hora de especificar los requerimientos, es la necesidad de un entendimiento de los problemas del cliente, deseos y ambiente en el cual el sistema será instalado. Para [Booch], uno de los factores principales de problemas es la complejidad del dominio, en el cual se encuentran los requerimientos. También tenemos que tener en cuenta, que el análisis de requerimientos es una actividad de intensa comunicación. Siempre que existe comunicación, se pueden producir malas interpretaciones, omisiones, ect. que pueden producir problemas tanto al analista como al cliente [Pressman]. Estos problemas se producen por la difícil iteración entre los clientes de un sistema y sus desarrolladores. Vemos que por un lado

hay un cliente que necesita resolver un problema, y por otro, hay un técnico que tratará de darle la solución desarrollando un sistema, tanto los clientes como los desarrolladores a menudo utilizan terminología diferente provocando problemas de comunicación. Los usuarios encuentran generalmente muy difícil dar precisión sobre sus necesidades de forma que los desarrolladores puedan comprender.

Normalmente, los clientes y desarrolladores tienen diferentes perspectivas de la naturaleza del problema y hacen suposiciones diferentes sobre la naturaleza de la solución. Por lo tanto, el alcance de la especificación de requerimientos hace de su producción una tarea especializada que es más difícil de hacer por ciertos factores, entre los cuales se incluye:

1. Poca calidad en la comunicación
2. Uso de herramientas y/o técnicas no adecuadas
3. La tendencia a reducir el tiempo del análisis de requerimientos
4. La no-consideración de alternativas antes de especificar el software
5. Cambios de los requerimientos debido a las variaciones en el ambiente y en la comprensión tanto del cliente como del desarrollador conforme progresa el desarrollo.

Un enfoque organizado para el desarrollo de software es esencial para reducir estos factores de dificultad. Hay muchas oportunidades para cometer errores por requerimientos mal interpretados en el camino hacia una implantación incorrecta.

Con el fin de servir como guía y agilizar el proceso de desarrollo de software, se desarrollaron varios modelos de ciclo de vida. No hay un modelo correcto para todos los propósitos, aun así, el uso de un modelo apropiado puede ayudar de manera considerable en el control de un proyecto de software. Entre los más populares y por todos conocidos podemos nombrar: El modelo de cascada, modelo en espiral de Boehm y Prototipado Evolutivo.

El modelo de cascada o ciclo de vida clásico

La versión original del modelo en cascada, fue presentada por Royce en 1970, aunque son más conocidos los refinamientos realizados por Boehm (1981), Sommerville (1985) y Sigwart et al. (1990).

En este modelo, el producto evoluciona a través de una secuencia de fases ordenadas en forma lineal, permitiendo iteraciones al estado anterior.

El número de etapas suele variar, pero en general suelen ser:

- Análisis de requerimientos del sistema.
- Análisis de requerimientos del software.
- Diseño preliminar.

- Diseño detallado.
- Codificación y pruebas.
- Explotación (u operación) y mantenimiento.

Modelo en espiral de Boehm

En 1988 Boehm propone el modelo en espiral, para superar las limitaciones del modelo en cascada. La espiral se forma a partir de una serie de ciclos de desarrollo y va evolucionando. Los ciclos internos del espiral denotan análisis y prototipado y los externos el modelo clásico. En la dimensión radial están los costos acumulativos y la dimensión angular representa el progreso realizado en cada etapa.

En cada ciclo se empieza identificando los objetivos, las alternativas y las restricciones del mismo. Se deben evaluar las alternativas de solución respecto de los objetivos, considerando las restricciones en cada caso. Es en este momento en que se puede llevar a cabo el siguiente ciclo.

Prototipado Evolutivo

El uso de prototipos se centra en la idea de ayudar a comprender los requerimientos que plantea el usuario, sobre todo si este no tiene una idea muy acabada de lo que desea. También pueden utilizarse cuando el ingeniero de software tiene dudas acerca de la viabilidad de la solución pensada.

Esta versión temprana de lo que será el producto, con una funcionalidad reducida, en principio, podrá incrementarse paulatinamente a través de refinamientos sucesivos de las especificaciones del sistema, evolucionando hasta llegar al sistema final.

Al usar prototipos, las etapas del ciclo de vida clásico quedan modificadas de la siguiente manera:

- Análisis de requerimientos del sistema.
- Análisis de requerimientos del software.
- Diseño, desarrollo e implementación del prototipo
- Prueba del prototipo.
- Refinamiento iterativo del prototipo.
- Refinamiento de las especificaciones del prototipo.
- Diseño e implementación del sistema final.
- Explotación (u operación) y mantenimiento.

Todos los ciclos de vida para el desarrollo de software descriptos antes incluyen una fase para el análisis de los requerimientos del sistema.

Uno de los mayores problemas que surgen en la fase de análisis y definición de los requerimientos en una etapa temprana del desarrollo, es cómo organizar toda la información que adquiere el analista en sus entrevistas con las personas involucradas en un proyecto y cómo poner de acuerdo a todas estas

personas sobre cuál es la solución más adecuada. Mientras que los métodos clásicos se basan en entrevistas bilaterales (el analista y cada una de las partes) con lo que dejan para el analista toda la labor de organización y obtención de un consenso, recientemente se tiende a prácticas más relacionadas con el *brainstorming* en el que cada parte expone sus ideas y propuestas y se produce un debate de forma que las posiciones vayan acercándose sucesivamente hasta que se llegue a un consenso [Raghavan].

2.3. Técnicas de elicitación.

El análisis de requerimientos siempre comienza con una comunicación entre dos o más partes. En el libro *Ingeniería de Software* de R. Pressman [Pressman], nos sugiere que un cliente tiene un problema al que puede encontrar una solución basada en computadora. El desarrollador responde a la petición del cliente. La comunicación ha comenzado. Pero, el camino entre la comunicación y el entendimiento está lleno de baches.

Antes de mantener las reuniones con los clientes y usuarios e identificar los requerimientos es fundamental conocer el dominio del problema. Enfrentarse a un desarrollo sin conocer las características principales ni el vocabulario propio de su dominio suele provocar que el producto final no sea el esperado por clientes ni usuarios. Por otro lado, mantener reuniones con clientes y usuarios sin conocer las características de su actividad hará que probablemente no se entiendan sus necesidades y que su confianza inicial hacia el desarrollo se vea deteriorada enormemente.

Para conocer el dominio del problema se puede obtener información de fuentes externas al negocio del cliente: folletos, informes sobre el sector, publicaciones, consultas con expertos, etc. En el caso de que se trate de un dominio muy específico puede ser necesario recurrir a fuentes internas al propio negocio del cliente, en cuyo caso pueden utilizarse las técnicas de elicitación de requerimientos como el estudio de documentación, observación *in situ*, cuestionarios, etc.

En realidad una primera reunión entre el cliente y el analista servirá como un período corto de preguntas y respuestas, el cual, en adelante debe sustituirse por reuniones que busquen entender el problema del usuario.

Normalmente encontramos que los clientes y analistas se enfrascan en el proyecto de forma unilateral y no en equipo. Cada parte define su propio "territorio" y se comunica a través de una serie de notas, impresos formales, documentos y sesiones de preguntas y respuestas. Este enfoque no es muy efectivo, abundan los malentendidos, se pierde información importante y nunca se establece una relación de trabajo satisfactoria.

Con estos problemas presentes, se desarrollaron numerosas técnicas para tratar de superar este difícil momento, que es el inicio del proceso. Cada técnica puede aplicarse en una o más actividades de la ingeniería de requerimientos; en la práctica, la técnica más apropiada para cada actividad dependerá del proyecto que esté desarrollándose.

A continuación se resumen algunas de las más conocidas:

Entrevistas

Las entrevistas son la técnica de elicitación más utilizada, y de hecho son prácticamente inevitables en cualquier desarrollo. En las entrevistas se pueden identificar claramente tres fases [Piattini]: preparación, realización y análisis, que se describen a continuación.

Preparación de entrevistas

Las entrevistas no deben improvisarse, por lo que conviene realizar las siguientes tareas previas:

- **Estudiar el dominio del problema:** se debe conocer la terminología básica del dominio del problema, evitando que el cliente tenga que explicar términos que para él son obvios. Para ello se puede recurrir a la técnica auxiliar de estudio de documentación, a bibliografía sobre el tema, documentación de proyectos similares realizados anteriormente, etc.
- **Seleccionar a las personas a las que se va a entrevistar:** se debe minimizar el número de entrevistas a realizar, por lo que es fundamental seleccionar a las personas a entrevistar. El orden de realización de las entrevistas también es importante. Normalmente se aplica un enfoque *top-down*, comenzando por los directivos, que pueden ofrecer una visión global, ayudar a determinar los objetivos y reducir ciertas *reticencias* en sus subordinados, y terminando por los futuros usuarios, que pueden aportar información más detallada.
- **Determinar el objetivo y contenido de las entrevistas:** para minimizar el tiempo de la entrevista es fundamental fijar el objetivo que se pretende alcanzar y determinar previamente su contenido.
- **Planificar las entrevistas:** la fecha, hora, lugar y duración de las entrevistas deben fijarse teniendo en cuenta siempre la agenda del entrevistado.

Realización de entrevistas

Dentro de la realización de las entrevistas se distinguen tres etapas, tal como se expone en [Piattini]:

- **Apertura:** el entrevistador debe presentarse e informar al entrevistado sobre la razón de la entrevista, qué se espera conseguir, cómo se utilizará la información, la mecánica de las preguntas, etc.
- **Desarrollo:** la entrevista en sí no debería durar más de dos horas, distribuyendo el tiempo en un 20% para el entrevistador y un 80% para el entrevistado. Se deben evitar los monólogos y mantener el control por parte del entrevistador, contemplando la posibilidad de que una

tercera persona tome notas durante la entrevista o grabar la entrevista en cinta de vídeo o audio, siempre que el entrevistado esté de acuerdo.

- **Terminación:** se debe recapitular sobre la entrevista para confirmar que no ha habido confusiones en la información recogida, agradecer al entrevistado su colaboración y citar para una nueva entrevista si fuera necesario, dejando siempre abierta la posibilidad de volver a contactar para aclarar dudas que surjan al estudiar la información o al contrastarla con otros entrevistados.

Análisis de las entrevistas

Una vez realizada la entrevista es necesario leer las notas tomadas, pasarlas a limpio, reorganizar la información, contrastarla con otras entrevistas o fuentes de información, etc. Una vez elaborada la información, se puede enviar al entrevistado para confirmar los contenidos. También es importante evaluar la propia entrevista para determinar los aspectos mejorables.

Brainstorming

El *brainstorming* o tormenta de ideas es una técnica de reuniones en grupo cuyo objetivo es la generación de ideas en un ambiente libre de críticas o juicios [Raghavan].

Este método comenzó en el ámbito de las empresas, aplicándose a temas tan variados como la productividad, la necesidad de encontrar nuevas ideas y soluciones para los productos del mercado, encontrar nuevos métodos que desarrollen el pensamiento creativo a todos los niveles, etc. Pero pronto se extendió a otros ámbitos, incluyendo el mundo de desarrollo de sistemas; básicamente se busca que los involucrados en un proyecto desarrollen su creatividad, promoviendo la introducción de los principios creativos.

Las sesiones de brainstorming suelen estar formadas por un número de cuatro a diez participantes, uno de los cuales es el *jefe* de la sesión, encargado más de comenzar la sesión que de controlarla.

Como técnica de elicitación de requerimientos, el brainstorming puede ayudar a generar una gran variedad de vistas del problema y a formularlo de diferentes formas, sobre todo al comienzo del proceso de elicitación, cuando los requerimientos son todavía muy difusos.

Fases del brainstorming

En el brainstorming se distinguen las siguientes fases [Raghavan]:

Preparación: la preparación para una sesión de brainstorming requiere que se seleccione a los participantes y al jefe de la sesión, citarlos y preparar la sala donde se llevará a cabo la sesión. Los participantes en una sesión de brainstorming para elicitación de requerimientos son normalmente clientes, usuarios, ingenieros de requerimientos,

desarrolladores y, si es necesario, algún experto en temas relevantes para el proyecto.

Generación: el jefe abre la sesión exponiendo un enunciado general del problema a tratar, que hace de *semilla* para que se vayan generando ideas. Los participantes aportan libremente nuevas ideas sobre el problema *semilla*, bien por un orden establecido por el jefe de la sesión, bien aleatoriamente. El jefe es siempre el responsable de dar la palabra a un participante. Este proceso continúa hasta que el jefe decide parar, bien porque no se están generando suficientes ideas, en cuyo caso la reunión se pospone, bien porque el número de ideas sea suficiente para pasar a la siguiente fase. Durante esta fase se deben observar las siguientes reglas:

- Se prohíbe la crítica de ideas, de forma que los participantes se sientan libres de formular cualquier idea.
- Se fomentan las ideas más *avanzadas*, que aunque no sean factibles, estimulan a los demás participantes a explorar nuevas soluciones más creativas.
- Se debe generar un gran número de ideas, ya que cuantas más ideas se presenten más probable será que se generen mejores ideas.
- Se debe alentar a los participantes a combinar o completar las ideas de otros participantes.

Consolidación: en esta fase se deben organizar y evaluar las ideas generadas durante la fase anterior. Se suelen seguir tres pasos:

- **Revisar ideas:** se revisan las ideas generadas para clarificarlas. Es habitual identificar ideas similares, en cuyo caso se unifican en un solo enunciado.
- **Descartar ideas:** aquellas ideas que los participantes consideren excesivamente *avanzadas* se descartan.
- **Priorizar ideas:** se priorizan las ideas restantes, identificando las absolutamente esenciales, las que estarían bien pero que no son esenciales y las que podrían ser apropiadas para una próxima versión del sistema a desarrollar.

Documentación: después de la sesión, el jefe produce la documentación oportuna conteniendo las ideas priorizadas y comentarios generados durante la consolidación.

Casos de Uso

Los casos de uso son una técnica para la especificación de requerimientos funcionales propuesta inicialmente en [Jacobson] y que actualmente forma parte de la propuesta de UML [Booch99].

Un caso de uso es la descripción de una secuencia de interacciones entre el sistema y uno o más *actores* en la que se considera al sistema como una caja negra.

Los casos de uso son una técnica para especificar el comportamiento de un sistema: “Un caso de uso es una secuencia de interacciones entre un sistema y actores que usan alguno de sus servicios” [Schneider].

Los actores son personas u otros sistemas que interactúan con el sistema cuyos requerimientos se están describiendo. Un actor puede participar en varios casos de uso y un caso de uso puede estar relacionado con varios actores.

Los casos de uso presentan ciertas ventajas sobre la descripción meramente textual de los requerimientos funcionales [Firesmith97], ya que facilitan la elicitación de requerimientos y son fácilmente comprensibles por los clientes y usuarios. Además, pueden servir de base a las pruebas del sistema y a la documentación para los usuarios.

La idea de especificar un sistema a partir de su interacción con el ambiente es original de McMenamin y Palmer, dos precursores del análisis estructurado, que en 1984 definieron un concepto muy parecido al del caso de uso: el *Evento*. Para McMenamin y Palmer, un evento es algo que ocurre fuera de los límites del sistema, ante lo cual el sistema debe responder.

Sin embargo, existen algunas diferencias entre los casos de uso y los eventos. Las principales son:

- Los eventos se centran en describir qué hace el sistema cuando el evento ocurre, mientras que los casos de uso se centran en describir cómo es el diálogo entre el usuario y el sistema.
- Los eventos son “atómicos”: se recibe una entrada, se la procesa, y se genera una salida, mientras que los casos de uso se prolongan a lo largo del tiempo mientras dure la interacción del usuario con el sistema. De esta forma, un caso de uso puede agrupar a varios eventos.
- Para los eventos, lo importante es qué datos ingresan al sistema o salen de él cuando ocurre el evento (estos datos se llaman datos esenciales), mientras que para los casos de uso la importancia del detalle sobre la información que se intercambia es secundaria. Según esta técnica, ya habrá tiempo más adelante en el desarrollo del sistema para ocuparse de este tema.

Los casos de uso combinan el concepto de evento del análisis estructurado con otra técnica de especificación de requerimientos bastante poco difundida: aquella que dice que una buena forma de expresar los requerimientos de un sistema es escribir su manual de usuario antes de construirlo [Howes]. Esta técnica, si bien ganó pocos adeptos, se basa en un concepto muy interesante: al definir requerimientos, es importante describir al sistema desde el punto de vista de aquél que lo va a usar, y no desde el punto de vista del que lo va a construir. De esta forma, es más fácil validar que los requerimientos documentados son los verdaderos requerimientos de los usuarios, ya que éstos comprenderán fácilmente la forma en la que están expresados.

Los Casos de Uso y UML

A partir de la publicación del libro de Jacobson, gran parte de los más reconocidos especialistas en métodos Orientados a Objetos coincidieron en considerar a los casos de uso como una excelente forma de especificar el comportamiento externo de un sistema. De esta forma, la notación de los casos de uso fue incorporada al lenguaje estándar de modelado UML (Unified Modeling Language) propuesto por Ivar Jacobson, James Rumbaugh y Grady Booch, tres de los precursores de las metodologías de Análisis y Diseño Orientado a Objetos, y avalado por las principales empresas que desarrollan software en el mundo. UML va en camino de convertirse en un estándar para modelado de sistemas de software de amplia difusión.

A pesar de ser considerada una técnica de Análisis Orientado a Objetos, es importante destacar que los casos de uso poco tienen que ver con entender a un sistema como un conjunto de objetos que interactúan, que es la premisa básica del análisis orientado a objetos "clásico". En este sentido, el éxito de los casos de uso no hace más que dar la razón al análisis estructurado, que propone que la mejor forma de empezar a entender un sistema es a partir de los servicios o funciones que ofrece a su entorno, independientemente de los objetos que interactúan dentro del sistema para proveerlos.

Como era de esperar, es probable que en el futuro los métodos de análisis y diseño que prevalezcan hayan adoptado las principales ventajas de todos los métodos disponibles en la actualidad (estructurados, métodos formales, métodos orientados a objetos, etc.).

Otras técnicas

Un ejemplo son las *técnicas para facilitar la especificación de la aplicación* (TFEA) [Pressman], cuyos dos enfoques más populares son Joint Application Development (JAD – Desarrollo Conjunto de Aplicaciones), desarrollado por IBM, y The METHOD (EL METODO), desarrollado por Performance Resources, Inc., Falls Church, VA.

La técnica denominada *JAD (Joint Application Development)*, desarrollada por IBM en 1977, es una alternativa a las entrevistas individuales que se desarrolla a lo largo de un conjunto de reuniones en grupo durante un periodo de 2 a 4 días. En estas reuniones se ayuda a los clientes y usuarios a

formular problemas y explorar posibles soluciones, involucrándolos y haciéndolos sentirse partícipes del desarrollo.

Esta técnica se base en cuatro principios [Raghavan]: dinámica de grupo, el uso de ayudas visuales para mejorar la comunicación (diagramas, transparencias, multimedia, herramientas CASE, etc.), mantener un proceso organizado y racional y una filosofía de documentación *WYSIWYG* (*What You See Is What You Get, lo que se ve es lo que se obtiene*), por la que durante las reuniones se trabaja directamente sobre los documentos a generar.

Debido a las necesidades de organización que requiere y a que no suele adaptarse bien a los horarios de trabajo de los clientes y usuarios, esta técnica no suele emplearse con frecuencia, aunque cuando se aplica suele tener buenos resultados, especialmente para elicitar requerimientos en el campo de los sistemas de información [Raghavan].

En comparación con las entrevistas individuales, presenta las siguientes ventajas:

- Ahorra tiempo al evitar que las opiniones de los clientes se contrasten por separado.
- Todo el grupo, incluyendo los clientes y los futuros usuarios, revisa la documentación generada, no sólo los ingenieros de requerimientos.
- Implica más a los clientes y usuarios en el desarrollo.

Podríamos analizar muchos métodos que ofrece esta técnica, pero todos enfocan hacia las siguientes directrices básicas:

Se efectúa una reunión en un lugar neutral, se establecen reglas para la preparación y participación, se elabora una agenda, se elige un facilitador, se utiliza un mecanismo de comunicación y primeramente el objetivo principal debe ser identificar el problema.

En el cuadro 1 resumimos Como resumen podemos presentar las principales ventajas y desventajas de cada una de las técnicas utilizadas en las etapas de la Ingeniería de Requerimientos.

Técnica	Ventajas	Desventajas
Entrevistas y Cuestionarios	<ul style="list-style-type: none">• Mediante ellas se obtiene una gran cantidad de información adecuada a través del usuario.• Pueden ser usadas para obtener una visión general del dominio del problema.• Son flexibles.• Permiten combinarse con	<ul style="list-style-type: none">• La información obtenida al principio puede ser redundante o incompleta.• Si el volumen de información manejado es alto, requiere mucha organización de parte del analista, así como la habilidad para tratar y comprender el

	otras técnicas.	comportamiento de todos los involucrados.
Tormenta de Ideas	<ul style="list-style-type: none"> • La producción de ideas en grupos puede ser más efectiva que la individual. • Aflora una gran cantidad de ideas sin ataduras. 	<ul style="list-style-type: none"> • Es necesaria una buena compenetración del grupo participante.
Casos de Uso	<ul style="list-style-type: none"> • Representan los requerimientos desde el punto de vista del usuario. • Permiten representar más de un rol para cada afectado. • Identifica nuevos requerimientos , dentro de un conjunto de requerimientos. 	<ul style="list-style-type: none"> • En sistemas grandes, toma mucho tiempo definir todos los casos de uso. • El análisis de calidad depende de la calidad con que se haya hecho la descripción inicial.
JAD	<ul style="list-style-type: none"> • Ahorra tiempo, elimina retrasos del proceso y mejora la calidad del sistema. • Es una de las mejores maneras de reducir los errores arrastrados de los resultados de los requerimientos iniciales. • Con la participación de gerentes y usuarios apropiados, el riesgo cultural típico se mitiga. • Evita funcionalidad sobredimensionada. • Evita que los requerimientos sean demasiado específicos o demasiados vagos, que son dos problemas comunes en el análisis. 	<ul style="list-style-type: none"> • Es costoso Involucrar al patrocinador del proyecto y gerente, experto en la tecnología, y expertos de la materia, como parte del equipo del proyecto.

Cuadro 1: Principales ventajas y desventajas de las técnicas de elicitación.

Es importante destacar, que independientemente a la técnica de elicitación que se utilice y durante todo el proceso de desarrollo del software es indispensable mantener una adecuada comunicación entre los participantes.

Para esto, un enfoque valioso es la utilización del mismo léxico. En la fase de elicitación, será tarea del analista ocuparse de homogeneizar el vocabulario, centrándolo en el léxico propio del cliente [Leite89], esto evitará que el analista utilice el lenguaje común en su medio ambiente. Simultáneamente, el cliente transmitirá sus necesidades en su lenguaje habitual.

3. LÉXICO EXTENDIDO DEL LENGUAJE.

El LEL (Language Extended Lexicon) [Leite90] 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.

El lenguaje de la aplicación es el lenguaje usado por los actores² 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.

Desde el punto de vista de la construcción de los escenarios, el LEL es un tamiz del universo del discurso. Los signos identificados forman la materia prima para los escenarios.

3.1. Objetivo.

El principal objetivo del LEL es conocer el vocabulario del usuario. Que el usuario y el desarrollador compartan el mismo lenguaje asegura la comunicación entre ambos en particular, el uso del lenguaje propio del usuario mejora considerablemente esta comunicación. En el proceso de Ingeniería de Requerimientos, la validación de los diferentes productos requiere una fuerte interacción con el usuario [Loucopoulos], la que se ve facilitada por el vocabulario común usuario-desarrollador. Las diferentes representaciones que se construyen en el proceso de desarrollo de software, encuentran en el vocabulario del usuario, un marco referencial que permite mantener la continuidad del vocabulario, lo que a su vez facilita el acceso a la documentación por parte de todos los participantes en el desarrollo.

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. En [Hadad97] podemos encontrar un resumen de las relaciones establecidas en términos de objetivos a alcanzar y las consecuencias que se

² Los actores son todas las personas involucradas en el proceso de desarrollo: los usuarios y los ingenieros de software.

derivan de la satisfacción de esos objetivos relacionados con el uso del LEL en un proyecto de desarrollo. Este resumen lo podemos ver en el cuadro 2:

Objetivo	Consecuencia
CONOCER EL VOCABULARIO DEL USUARIO	<ul style="list-style-type: none">• Asegurar la comunicación• Facilitar la validación de los requerimientos con el usuario• 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

Cuadro2: Objetivo-Consecuencia del LEL (fuente Hadad 97)

En [Leite89] se propone construir el LEL como primer paso en la fase de elicitación de los requerimientos: “despreocuparse durante este procedimiento de entender el problema y abocarse sólo a conocer el vocabulario que utiliza el usuario en su mundo real”. Una vez familiarizados con ese léxico, el camino para comunicarse con el usuario y comprender el problema tendrá un obstáculo menos: el vocabulario.

3.2. Características del LEL.

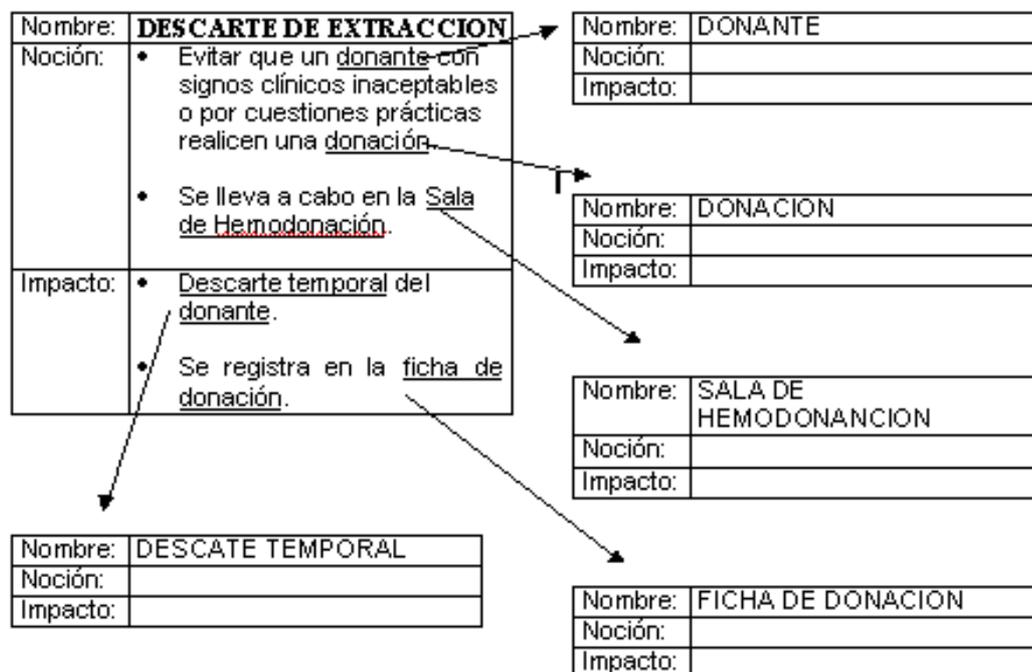
El LEL está compuesto por un conjunto de símbolos que identifican el lenguaje de la aplicación. Los *símbolos* son, en general, las palabras o frases utilizadas por el usuario y que repite con más frecuencia. También se incluyen aquellas palabras o frases que son relevantes para el dominio del problema, más allá de su frecuencia de repetición. Los símbolos se adquieren, por ejemplo, de entrevistas, observaciones, lectura de documentos [Leite90]. Se genera una lista con todos los símbolos reconocidos. Durante el proceso de recolección, el ingeniero de software procura entender el significado de cada símbolo. La semántica de cada símbolo se representa con una o más nociones y uno o más impactos. El impacto puede no existir. La *noción* indica qué es el símbolo y el *impacto*, cómo repercute en el sistema. Por lo tanto, cada símbolo tiene un “nombre” que lo identifica, una “noción” y un “impacto” que lo describen [Leite93].

A continuación podemos ver un ejemplo de un símbolo del LEL:

Nombre:	DESCARTE DE EXTRACCION
Noción:	<ul style="list-style-type: none"> • Evitar que un <u>donante</u> con signos clínicos inaceptables o por cuestiones prácticas realicen una <u>donación</u>. • Se lleva a cabo en la <u>Sala de Hemodonación</u>.
Impacto:	<ul style="list-style-type: none"> • <u>Descarte temporal</u> del <u>donante</u>. • Se registra en la <u>ficha de donación</u>.

En la descripción de las nociones e impactos existen dos reglas básicas [Hadad97] [Leite93], que se deben cumplir simultáneamente: una tiende a maximizar el uso de símbolos en el significado de otros símbolos, esto es acotar el lenguaje externo al dominio de la aplicación, utilizando para la descripción de la noción y el impacto símbolos definidos en el mismo LEL. Esto se denomina “principio de circularidad”. La otra requiere que se minimice el uso de símbolos externos al lenguaje de la aplicación. Este principio se denomina “principio del vocabulario mínimo”.

Con estos principios se tiende a que se utilice lo más posible el lenguaje de la aplicación. Como resultado se obtiene un conjunto de símbolos que forman una red que permite representar al LEL en un hipertexto [Leite90]. Los links lo determinan los símbolos que tienen una entrada en el LEL y se utilizan para definir otros. Lo que permite navegar entre ellos para conocer todo el vocabulario del dominio. Esto se ilustra en el siguiente ejemplo:



3.3. Proceso de construcción del LEL.

Se ha propuesto en el trabajo de investigación “Integración de Escenarios con el Léxico Extendido del Lenguaje en la elicitación de requerimientos: Aplicación a un caso real” [Hadad99], que el proceso de construcción del LEL consta de 6 etapas interdependientes, que en algunos casos se desarrollan simultáneamente, las cuales podemos ver a continuación:

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.

En la Figura 2 se presentan gráficamente las etapas mencionadas y luego se describen y ejemplifican con el caso de estudio:

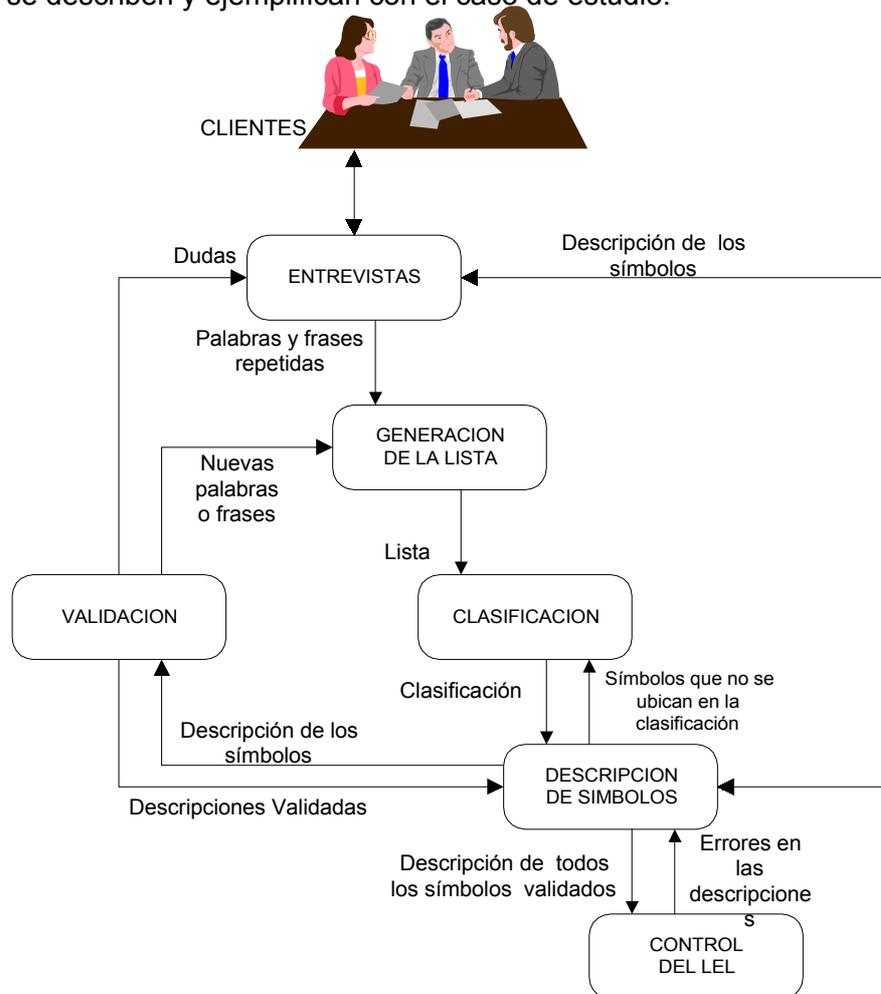


Figura. 2 - Etapas para la construcción del LEL. (Fuente Hadad 97)

3.3.1. Entrevistas.

Permiten 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 que el ingeniero de requerimientos tiene en construir un LEL;
- los conocimientos que el cliente entrevistado tiene del tema en estudio.

Estas entrevistas, en primera instancia suelen ser reuniones con los usuarios. La cantidad de información que se puede extraer de una entrevista es grande. 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. Contrariamente al análisis de requerimientos tradicional, donde el analista identifica las funciones a partir de las entrevistas, el objetivo es familiarizarse con el lenguaje del dominio de la aplicación.

3.3.2. Generación de la lista de símbolos.

La lista de símbolos es el conjunto de símbolos obtenidos durante las entrevistas. Pueden elegirse 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.

Primero se genera una **lista de símbolos candidatos**, que es la lista inicial obtenida de las primeras entrevistas, en las que se registran las palabras o frases que el cliente repite o enfatiza. Esta lista se modifica con frecuencia después de las primeras validaciones, que es cuando se incorpora mayor cantidad de símbolos y sinónimos. Se eliminan 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.

Durante la generación de la lista candidata se determina el punto de vista que tendrá el LEL. Esto permite homogeneizar las descripciones, más aún cuando participan varios ingenieros de requerimientos en la construcción del LEL [Hadad97].

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.

3.3.3. Clasificación de los símbolos.

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

Partimos de la clasificación general del LEL definida en [Leite90], donde se dividen los símbolos en **Sujeto**, **Verbo**, **Objeto** y **Estado**.

En este trabajo propongo realizar una subclasificación en función del Universo del Discurso de la aplicación en estudio, como veremos más adelante.

3.3.4. Descripción de los símbolos.

Describir los símbolos es determinar su noción y su impacto. Durante la generación de la lista candidata se percibe inicialmente el significado de los símbolos, ya que las palabras o frases significativas del dominio están dentro de un contexto que el ingeniero de requerimientos está escuchando. Esto permite esbozar una primera descripción de algunos símbolos. Luego de validar con los clientes y determinar la clasificación, se describen los restantes símbolos.

A continuación se detallan una serie de reglas que pueden ayudar en este proceso, [Antonelli]:

- 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.
- Para los símbolos que son sujeto de una oración, los impactos deben indicar las acciones que realiza.
- Para los símbolos que cumplen el rol de verbo, las nociones deben decir quién ejecuta la acción, cuándo sucede y el proceso involucrado en la acción. Para los impactos se deben identificar las restricciones sobre la realización de la acción, qué es lo que origina esta acción y qué es lo que causa esta acción.
- Para los símbolos que son objetos de una oración, la noción debe identificar otros objetos con los cuales se relaciona y los impactos serán las acciones que se pueden realizar con este signo.

3.3.5. Validación con los clientes.

La validación permite rectificar o ratificar el conocimiento del vocabulario del dominio obtenido durante las entrevistas. 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.

3.3.6. Control del LEL.

Este proceso consiste en controlar 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. O sea, 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.

3.4. Caso de estudio de construcción del LEL

Para experimentar en el uso del LEL, se utilizó como caso de estudio, el Sistema de Registración y Producción del Instituto de Hemoterapia de la provincia de Buenos Aires. Este trabajo se centró en la fase de elicitación de requerimientos, utilizando la metodología basada en el uso del Léxico Extendido del Lenguaje (LEL) y Escenarios. Se trabajó con usuarios cuya característica sobresaliente fue justamente el léxico que utilizan en sus tareas cotidianas, razón por la cual resultó adecuado para aplicar la metodología.

El Instituto de Hemoterapia es una institución que depende del Ministerio de Salud de la Provincia de Buenos Aires. Tiene como misión asistencial, suministrar sangre y hemocomponentes a centros sanitarios de la zona, públicos y privados. Actualmente el Instituto recibe alrededor de 40.000 donaciones anuales, esto implica que es un sistema que maneja un gran volumen de información y requiere muchas verificaciones y controles internos.

Operativamente, el Instituto consta de los siguientes Sectores: Sala de Recepción, Consultorios, Sala de Hemodonación, Laboratorio de Inmunoserología, Laboratorio de Inmunoematología, Sector de Fraccionamiento o Producción y Sector Banco de Sangre. Las principales tareas que se realizan en estos sectores son las siguientes:

- En la Sala de Recepción se solicitan los datos personales al donante y se verifica si realizó donaciones anteriores.
- En Consultorios, un médico hematólogo realiza una entrevista profesional con el objeto de comprobar si el donante se encuentra “medicamente” apto para realizar una donación. Es decir, comprobar que el donante no pertenezca a algún grupo de riesgo.
- En la Sala de Hemodonación se verifica si el donante esta “clínicamente” apto para realizar una donación, y a continuación se procede con la extracción de sangre.
- El Laboratorio de Inmunoserología se encarga de realizar estudios a la sangre con el objeto de detectar enfermedades de transmisión como HIV, sífilis, chagas, etc.
- En el Laboratorio de Inmunoematología se realizan estudios para determinar la estructura orgánica de la sangre tal como grupo, factor, etc.
- En el Sector de Fraccionamiento o Producción se realizan procesos de fraccionamiento a la sangre para obtener los hemocomponentes o componentes de la sangre que son los que finalmente son almacenados en el Sector Banco de Sangre para posteriormente ser distribuidos a clínicas y hospitales.

3.4.1. Pasos previos a la construcción del LEL

Como se vió anteriormente, para la elaboración del LEL se dispone de un modelo de seis etapas [Hadad97] y [Hadad99]. En la estrategia utilizada para la construcción del LEL, en el punto de comienzo se encuentra el Universo del Discurso³. Con el objetivo de construir el UD se desarrollaron diversas entrevistas con el usuario principal, con diversos integrantes de la institución y sesiones de trabajo de los ingenieros de requerimientos para la producción de material.

- **Entrevista preliminar.** El objetivo de esta entrevista fue que los desarrolladores tuvieran una visión global de las actividades que realiza el Instituto de Hemoterapia, principales funciones y esquema de organización institucional. Esta entrevista se realizó con el Gerente Administrativo del Instituto (el cargo lo desempeña un médico hematólogo).

En la reunión se revisó el circuito que realiza un donante y luego el circuito que realiza la donación (producto). Durante la revisión, el entrevistado explicó las tareas que se realizan en cada sector, los entrevistadores tomaron notas y se elaboró una primera visión del Universo del Discurso.

- **Primera entrevista sectorial.** El objetivo fue conocer el funcionamiento detallado de cada sector del Instituto y para ello se hicieron entrevistas en dos etapas. En la primera se asumió el rol de un donante⁴ desde el momento que ingresa al Instituto y hasta que termina su donación. En la segunda etapa se realizó un seguimiento del producto donado y hasta el momento que queda listo para su distribución.

Los que participaron de esta reunión fueron los encargados de cada sector. Se utilizó un grabador (con conocimiento del entrevistado) para registrar lo que explicaban, dejando que el entrevistado describa el funcionamiento del sector.

No se prepararon preguntas para estas entrevistas, pero surgieron algunas en el transcurso de la misma, generalmente para aclarar los términos técnicos que utilizaba el usuario.

La mecánica fue de una reunión única, en la que estuvieron todos los entrevistados simultáneamente, pero se interactuó con uno por vez en los temas específicos del sector que se tratara. Esta entrevista duró aproximadamente dos horas.

- **Universo del discurso.** Con el material obtenido se completó el Universo del Discurso [Gil99b], confeccionándose un documento en un editor de texto, transcribiendo y editando la cinta grabada. El documento resultante tiene una extensión de nueve páginas.

Si bien la construcción previa del Universo del Discurso, difiere con el modelo propuesto por [Hadad97] y [Hadad99] en el proceso de construcción del LEL, lo

³ “La producción de una descripción del comportamiento del sistema que sea completa, sin inconsistencias ni ambigüedades requiere establecer el contexto: lugar en el que se desarrollarán las tareas de ingeniería, recursos disponibles, objetivos del producto y sus límites. A este contexto lo denominamos *Universo del Discurso* (UD) y no debe establecerse una visión estática del mismo [...] En el UD residen las fuentes de información a ser utilizadas: el proceso de elicitación busca la información necesaria para conocer el sistema.” ([Hadad99] pp 83-84)

⁴ En este punto se siguió la experiencia de [Hadad99] de asumir un rol simulado, en este caso el de donante.

consideramos de gran importancia, pues permitió conocer los sectores que componen el Instituto de Hemoterapia y los recursos disponibles en cada uno de ellos.

3.4.2. Proceso de Construcción del LEL

En cuanto al proceso específico de construcción del LEL se siguió el modelo propuesto en [Hadad97] y [Hadad99]. Dicho modelo consta de 6 etapas como ya se presentó en el punto 3.3 de este informe. En el cuadro 3 podemos ver estas etapas con un resumen de los objetivos de cada una de ellas:

Etapas	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.

Cuadro 3. Etapas - Objetivos

En este caso de estudio, se trabajó con el esquema del Cuadro 2 como modelo de etapas a desarrollar. Si bien el modelo prevé fuertes interacciones a través de un desarrollo interdependiente de las etapas e incluso en paralelo, en el proceso de construcción del LEL se pudo observar que las fronteras que existen entre las cuatro primeras son muy difusas, esta es una de las principales conclusiones de nuestra investigación.

En todo el proceso de producción del LEL participaron dos investigadores como ingenieros de requerimientos o desarrolladores.

3.4.3. Entrevistas y generación de símbolos

Una vez obtenido el UD, comenzó el trabajo específico de producción del LEL.

- **Trabajo con el UD.** Se identificaron las entradas candidatas para el LEL. Para esto se buscó las palabras o frases utilizadas por el usuario y las que repitió con más frecuencia tal como se sugiere en [Hadad 99]. En este momento surgió claramente que en el Universo del Discurso existían palabras y frases extremadamente específicas del ámbito de la medicina, lo que ratificó el hecho de que los investigadores y clientes hablaban idiomas diferentes. En ese momento se resolvió incorporar dichas palabras y frases a las entradas candidatas.

En este punto, cuyo objetivo es conocer el vocabulario de la aplicación, se decidió organizar la siguiente entrevista con un esquema de trabajo en donde los investigadores desempeñaran distintos roles, uno a cargo de mejorar la comprensión del problema siguiendo la entrevista con el cliente y el otro a cargo de registrar palabras o frases que el cliente repetía o las que considerara importantes, teniendo en cuenta el contexto.

- **Segunda entrevista sectorial.** El objetivo de esta entrevista fue refinar y profundizar el conocimiento que los investigadores tenían en ese momento del problema. También fue grabada, y en cuanto al recorrido, el criterio fue similar al de la primera, pero en esta oportunidad un investigador se dedicó a tratar de entender el problema, mientras que el otro tomaba notas. Estas notas registraban potenciales símbolos candidatos de entradas al LEL, tomando nota de las palabras, frases y términos específicos que utilizaban los encargados de cada sector.
- **Lista de símbolos y LEL preliminar.** Se elaboró una lista de candidatos a símbolos del LEL y se esbozaron posibles contenidos de las nociones e impactos para tener un cierto “recordatorio” acerca del símbolo. Esta lista, como todo el material del proceso, se registró en un editor de textos de uso comercial.
- **Validación con el cliente.** Se realizó una reunión con el usuario (Gerente Administrativo del Instituto) con una lista de entradas candidatas y sus posibles Nociones e Impactos, el propósito de la reunión fue clarificar el significado de algunas entradas del LEL. Previamente se le hizo llegar una copia del documento elaborado, (LEL preliminar), para que lo leyera e hiciera las observaciones que creyera conveniente. Se le explicó que la idea detrás de la elección de símbolos es entender el lenguaje, de ningún modo pretendíamos elegir símbolos para determinar las funciones principales del sistema. El punto a destacar es que el usuario no tuvo inconvenientes para comprender el LEL preliminar, lo que permitió completar varias entradas que hasta antes de esta reunión se tenían marcadas como dudosas. Además, esta entrevista dió la posibilidad de detectar aquellos símbolos que los ingenieros de requerimientos consideraron diferentes, cuando en realidad eran sinónimos. Por ejemplo: el Instituto cuenta con dos laboratorios, el de Inmunoematología y el de Inmunoserología. Internamente, el personal menciona la palabra “inmuno” para referirse al Laboratorio de Inmunoserología, esto recién se descubrió en esta reunión con el usuario, lo que nos llevó finalmente a incluirla como un sinónimo de Laboratorio de Inmunoserología en el LEL.

Otro punto a destacar fue que el usuario tendió a ampliar las nociones e impactos con detalles extremadamente técnicos propios de su conocimiento que no hacían a la comprensión de los símbolos. Esto contrasta con la actitud habitual de los usuarios en cuanto a la primera presentación de un trabajo de análisis, en el cual la predisposición es normalmente retraída y de poca participación.

Realizar la validación con el cliente en esta etapa, no está previsto el modelo que se siguió para construir el LEL. Sin embargo fue de mucha utilidad, dado que gracias a la realización de la misma, se aclararon muchos puntos necesarios para realizar la clasificación y descripción de los símbolos.

3.4.4. Clasificación y descripción de símbolos

Sobre la base de la Lista Preliminar de símbolos, se elaboró una clasificación específica para el dominio considerado. Partimos de la clasificación general del LEL definida en [Leite90] donde se dividen los símbolos en **Sujeto**, **Verbo**, **Objeto** y **Estado**. En este trabajo proponemos realizar una subclasificación en función al Universo del Discurso de la aplicación en estudio. Esta subclasificación se basa en la clasificación que se utilizó en [Hada 99] con pequeñas modificaciones. En el Cuadro 4 se reproduce dicha clasificación.

Tipo	Apertura
Sujeto	Sectores Personas
Objeto	Formularios Índices Depósitos
Verbo	Acciones

Cuadro 4. Subclasificación de Símbolos

En el caso desarrollado, la problemática del Lenguaje Técnico utilizado se resolvió en esta etapa del proceso. En el lenguaje se identificaban una serie de términos, potencialmente candidatos a símbolos del LEL, que fueron denominados **Conceptos Específicos**, estos eran un conjunto de palabras que requerían a los investigadores esfuerzos especiales para comprender su significado, tales como: Hemocomponente, Resultado no Reactivo, Confirmación, etc. Si bien algunos como “Serología Positiva” permitían a los investigadores intuir su contenido, no resultaba atinado asignar el contenido proveniente de la cultura general a esas palabras. Inicialmente se consideraron como términos extremadamente específicos y que su desconocimiento no influiría en la comunicación usuario-desarrollador, sin embargo, a lo largo de la interacción con el usuario quedó claro que su incorporación al LEL era clave para entender su lenguaje. De allí que se resolviera su incorporación al LEL.

Dentro de los Conceptos Específicos, se remarcó la diferencia entre algunos que identifican solamente conceptos, y otros que identifican procesos. Se designó como **Términos Específicos (TE)** a los primeros y **Métodos Específicos (ME)** a los segundos. Un ejemplo de los primeros es “Hemocomponente” y de los segundos es “Screening”.

En el Cuadro 5 se reproduce la versión final de la clasificación adoptada.

Tipo	Apertura
Sujeto	Personas (S) Sectores (P)
Objeto	Almacenamiento (AI) Formularios (F) Índices (I) Términos Específicos (TE)
Verbo	Acciones (A) Métodos Específicos (ME)

Cuadro 5. Clasificación adoptada.

Analizando las entradas por grupo, se identificaron ciertos aspectos comunes que simplificaban el proceso de definición de los símbolos y nociones. Esto sugirió la existencia de un cierto “template” común que facilita la definición de los símbolos y su control, esto es coincidente con resultados que ya fueron comprobados y reportados por varios autores en trabajos realizados sobre dominios diferentes. Este template para cada tipo de entrada se estableció definiendo para cada una de ellas el contenido correspondiente a la Noción y el Impacto.

Para los Términos Específicos, en la noción se definió su significado, sus características y en qué sector o proceso se lo encuentra; en el impacto se describió que se hace con él o que se le hace.

Los Métodos Específicos, son muy similares a la subclasificación Acciones, pero se agregaron en la noción sus características particulares para facilitar la comprensión.

Es posible confundir las Acciones con los Métodos Específicos. Pero los Métodos Específicos son aquellos que están estrechamente ligados al contexto del Universo que se está estudiando. La mera lectura del nombre de la entrada no sugiere el contenido del símbolo, por ejemplo, el símbolo “screening” posee un significado específico tal, que inclusive los que están en el ámbito de la medicina no lo conocen, es necesario estar dentro de la especialidad de hemoterapia para conocerlo.

A continuación se detallan los templates establecidos y ejemplos de aplicación.

TEMPLATES Y EJEMPLOS

Como se mencionó, se elaboró un template para cada tipo de entrada que se utilizó en la descripción de los símbolos de ese tipo. En lo que sigue, se reproducen los template definidos y se da un ejemplo de uso.

➤ **Tipo Verbo:**

Template de Acciones (A)

Noción: identifica el sector *donde* se lleva a cabo y el *objetivo* de la acción.

Impacto: describe los *efectos sobre* las personas, los formularios, los depósitos y la generación o uso de índices (contenido similar a Sectores).

Ejemplo de Símbolo del tipo “Acción”:

DESCARTE DE EXTRACCION (A)

Noción:

- Evitar que un donante con signos clínicos inaceptables o por cuestiones prácticas realicen una donación.
- Se lleva a cabo en la Sala de Hemodonación.

Impacto:

- Descarte temporal del donante.
- Se Registra en la ficha de donación.

Template de Métodos Específicos (ME).

Noción: identifica el sector *donde* se lleva a cabo, el *objetivo* y características particulares del método

Impacto: describe los *efectos sobre* las personas, los formularios, los depósitos y la generación o uso de índices.

Ejemplo de Símbolo del tipo “Métodos Específicos”:

PLAQUETOAFERESIS (ME)

Noción:

- Técnica para separar las plaquetas de la sangre entera.
- Se extrae la sangre de un brazo al donante, se separan las plaquetas y el resto se repone al donante mediante un flujo continuo.
- El proceso dura dos horas.
- Es utilizada en la sala de transfusión y aféresis.

Impacto:

- El donante necesita preparación previa.
- El donante es inhabilitado por 48 horas para una nueva donación.
- Registrar en la planilla de existencias.

➤ Tipo Objetos:

Template de Almacenamiento (AI).

Noción: identifica su *contenido* y *ubicación*.

Impacto: describe *en qué acciones* se lo utiliza.

Ejemplo de Símbolo del tipo “Almacenamiento”:

FICHERO DE ASEGURADOS (AI)

Noción:

- Conjunto de fichas de donantes que pertenecen al seguro de sangre
- Se encuentran en el Sector de Recepción.

Impacto:

- Es utilizado para seguir la historia de las donaciones.
- Es consultado para realizar notificaciones al asegurado, en caso de necesidades extraordinarias.

Template de Formularios (F).

Noción: identifica la información que *contiene* y su *finalidad*.

Impacto: describe todas las *acciones que sobre él se efectúan*.

Ejemplo de Símbolo del tipo “Formulario”:

FICHA DE DONACION (F)

Noción:

- Es un papel preimpreso, obligatorio para todos los donantes.
- Sirve para que el médico pueda determinar si el donante se encuentra en condiciones de realizar una donación.
- Detectar si el donante pertenece a algún grupo de riesgo.
- Informar al donante sobre las condiciones de extracción.
- Se escriben las verificaciones que se realizan en consultorio y en la sala de hemodonación.

Impacto:

- Recepción lo llena con los datos personales del donante y con el tipo de donación.
- En consultorio el médico completa la parte de admisión y lo firma.
- Lo firma el donante aceptando condiciones de extracción.
- En la Sala de Hemodonación un técnico completa los signos clínicos y lo firma.
- Un técnico completa las características de la extracción y lo firma.

Template de Índices (I).

Noción: identifica cuál es su *objetivo*, *quién* lo genera y sus *características*.

Impacto: describe *dónde* se lo utiliza, *en qué* formularios y *para qué* acciones.

Ejemplo de Símbolo del tipo “Índices”:

NÚMERO DE DONACION (I)

Noción:

- Lo genera el Sector de Recepción.
- Es un número secuencial que comienza en 1 cada año, único para cada donación.
- Sirve para identificar interna y externamente a la donación.

Impacto:

- Se lo utiliza en la elaboración de la etiqueta inicial y la etiqueta final.

Template de Términos Específicos (TE).

Noción: define su significado, sus características y en que sector o proceso se lo identifica.

Impacto: describimos que se hace con él o que se le hace.

Ejemplo de Símbolo del tipo “Término Específicos”:

SEROLOGIA POSITIVA (TE)

Noción:

- Es un resultado reactivo en alguna de las patologías estudiadas en el Laboratorio de Inmunoserología.

Impacto:

- Es causa del descarte del producto.
- Es causa de descarte temporario o descarte definitivo del donante.

➤ **Tipo Sujetos:**

Template de Personas (P).

Noción: identifica las *características* y *condiciones* que debe cumplir.

Impacto: describe todas las *acciones* que pueden realizar.

Ejemplo de Símbolo del tipo “Personas”:

MÉDICO (P)

Noción:

- Médico especialista en hematólogo.
- Encargado de administrar cada sector del Instituto de Hemoterapia.

Impacto:

- Realizar la encuesta al donante en el consultorio.
- Firmar la ficha de donación.
- Capacitar al personal técnico de cada sector.
- Realizar control de calidad del producto.
- Supervisar el funcionamiento de equipos.
- Evaluar el cumplimiento de normas.

Template de Sectores (S).

Noción: identifica *dónde* se desarrolla y las *acciones* principales que cumple.

Impacto: los *efectos* sobre las personas, los formularios, los depósitos y la generación o uso de índices.

Ejemplo de Símbolo del tipo “Sectores”:

LABORATORIO DE INMUNOSEROLOGIA (S)

Noción:

- Sector del Instituto de Hemoterapia.
- Realizan estudios a las muestras de sangre para detectar patologías.
- Utilizan dos métodos, screening y confirmación.

Impacto:

- A partir de la muestra de sangre, se detectan las siguientes patologías: sífilis, chagas, brucelosis, hepatitis-a, hepatitis-b, hepatitis-c, sida.
- Completan y firman la planilla de serología.

3.4.5. Validación con los clientes

El proceso de validación de las entradas definidas fue similar al descrito en el punto 3.4.3. La diferencia fundamental con respecto a la primera validación

con el cliente, fue que el estado de las entradas era prácticamente el definitivo, echo que se refleja en las pocas modificaciones realizadas.

3.4.6. Control del LEL

En esta actividad el uso del template contribuyó muy efectivamente para facilitar el control. El Control del LEL fue ejecutado por los dos Ingenieros de Requerimientos que trabajaron en el caso. El enfoque utilizado fue la revisión individual y conjunta del producto terminado a través de la lectura exhaustiva de la versión final del LEL.

La utilización de los templates de tipos de entradas resultó de utilidad en cuanto a asegurar la utilización consistente del enfoque y permitió detectar inconsistencias en las definiciones de algunas entradas. Si bien aún persiste el problema de asegurar la completitud del LEL, la utilización intensiva de los tipos de entradas permitió detectar algunas carencias.

La versión final del LEL quedó formada por 51 símbolos, de los cuales el 22% son Conceptos o Términos específicos.

3.4.7. Conclusiones

La experiencia obtenida en la construcción del Universo del Discurso y LEL para el caso de estudio (Banco de Sangre), deja una sensación de lentitud en cuanto al desarrollo de las etapas desarrolladas para construirlos. Pero esto se debe a que el desarrollador, invierte demasiado tiempo en cuestiones que no hacen a la elicitación de requerimientos propiamente dicha, dado que necesita realizar tareas de administración de la información obtenida, como por ejemplo: insertar nociones e impacto a símbolos ya existentes, ordenar todas las entradas alfabéticamente por el símbolo, marcar símbolos en distintos colores, etc.

Este problema justifica uno de los propósitos de esta tesis, que es desarrollar una herramienta para soportar la producción de LEL, incorporando en la misma un entorno integrado entre el Universo del Discurso y LEL con tecnología de hipertexto, para facilitar la navegación entre los distintos elementos y de esta forma interactuar de una manera mucho más flexible en la navegación y administración de la información obtenida, para conocer todo el vocabulario del dominio.

4. ESCENARIOS

4.1. 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 mismos son de gran ayuda en el momento de especificar requerimientos; y su rol principal es el de permitir la comunicación entre expertos de software y del dominio, y analizar aspectos específicos de un sistema, describiéndolo en forma concreta. 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.

Los escenarios describen actores, objetivos y episodios. Un actor no necesariamente es una persona o agente físico, un actor representa un rol dentro del sistema, por lo tanto, los actores son las entidades que hacen uso del sistema para satisfacer cierta necesidad, estas necesidades son los objetivos, que representan las condiciones a ser alcanzadas. Los objetivos están representados por episodios. Un episodio es un conjunto de acciones asignadas a determinados actores. Están formados por un conjunto de oraciones en concordancia a un lenguaje natural simple que hace posible la descripción operacional del comportamiento, las cuales involucran la actividad de alguna función del sistema.

Varios autores han estudiado esta técnica como por ejemplo [Potts], [Jacobson], [Carrol], [Zorman]. Esto lleva aparejado que el formato en el cual se generan es muy variado. Pueden ser hechos en lenguaje natural, como los uses cases en [Jacobson], pueden ser storyboards, una descripción gráfica en donde las acciones se identifican con distintos cuadros como si fuera una historieta, o pueden ser diagramas de interacción entre objetos.

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].

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 cumplen distintos objetivos, dependiendo de la fase en que se describen durante el ciclo de desarrollo del software. En la fase de producción 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.

4.2. Modelos existentes.

A continuación se analizan algunos modelos de escenarios propuestos por diferentes autores.

4.2.1 Uses Cases de Jacobson

Para ver las similitudes entre escenarios y use cases, analicemos el planteo de Jacobson de los use cases, En el cual se observa que el modelo de use cases define el comportamiento del sistema (una de las dos componentes fundamentales de los escenarios). Se describe el entorno del sistema (situación como se llama en los escenarios) a través de los distintos usuarios que operan el sistema a través de los use cases.

El modelo de use cases es un grafo con dos tipos de nodos, nodos actores y nodos use cases, y con un nombre, el nombre del sistema. Cada nodo actor tiene un nombre y una clase. Los nombres de los nodos actores son únicos. Por su parte, cada nodo use case tiene también un nombre y una clase. Los nombres de los nodos use cases, también son únicos.

Un nodo actor tiene al menos un arco hacia un nodo use case, y un nodo use case tiene al menos un arco hacia un nodo actor. Estos arcos se denominan arcos de comunicación.

Una instancia de un actor puede crear instancias de use cases, y una instancia de use cases obedece a su clase. Un arco de comunicación entre un nodo actor y un nodo use case significa que un estímulo ha sido enviado entre una instancia de la clase actor y una de la clase use case o entre instancias de la clase use case.

Los actores son objetos que residen fuera del modelo del sistema. Representan todo lo que necesita intercambiar información con el sistema. Nada más fuera del sistema tiene algún impacto en él. Los actores pueden ser humanos u otro sistema.

Se hace una distinción entre actores y usuarios. Un usuario es un humano que usa el sistema, en cambio, un actor representa el rol específico que

un usuario puede jugar. Los actores son instancias de una clase, y los usuarios son algún tipo de recursos que implementan estas instancias. El mismo usuario puede así actuar como instancias de diferentes actores.

4.2.2. OBA

Se propone la descripción de escenarios a través de scripts. Un script es una descripción estructurada de un uso típico del sistema. Se forman realizando un contrato entre dos roles. El primer rol, iniciador, colabora con el segundo participante para realizar un paso de la tarea completa. El iniciador realiza una acción, responsabilidad y el participante responde con otra acción, el servicio correspondiente.

Cada script contiene: nombre, autor, versión, precondition (estado del sistema para que ese script suceda), postcondición (estado final del sistema al finalizar el script), trace (área de actividad del dominio a la que pertenece ese script).

Estos scripts surgen en la etapa de análisis de requerimientos para capturar la funcionalidad del sistema. En la etapa de diseño se utilizan los scripts para encontrar los objetos del sistema, sus responsabilidades y colaboraciones con el resto de los objetos. Finalmente se usan las pre y post condiciones para determinar el ciclo de vida completo de un objeto.

4.2.3. ICM, una nueva propuesta [Potts 95]

En el trabajo de Potts 95, se especifica en forma más detallada el concepto de escenarios, se identifican partes que los componen y se presentan estrategias para definirlos.

Un escenario es una descripción narrativa de un uso concreto del sistema. Describe la ejecución de una parte de la funcionalidad del mismo. Los escenarios tienen actores con objetivos (condiciones a ser alcanzadas). Estos objetivos pueden no alcanzarse por determinados obstáculos. Estos obstáculos pueden ser condiciones del sistema o porque se ha alcanzado algún otro objetivo (conflictivo). Los objetivos y obstáculos están representados por episodios. Un episodio es un conjunto de acciones asignadas a determinados actores. Un actor no necesariamente es una persona o agente físico, un actor representa un rol dentro del sistema.

El esquema propuesto de escenarios es el siguiente:

Escenario: nombre del escenario

Settings:

Background: información, estado o situación del sistema que indica los objetivos de los actores.

Roles(actores): Cada uno de los actores intervinientes.

Narrativa: Conjunto de episodios: Cada episodio está descrito por: objetivos, obstáculos (opcional), acciones y logros.

El hecho de considerar obstáculos, fuerza a los diseñadores a pensar en soluciones flexibles y robustas para situaciones del sistema no idealizadas (por ejemplo: errores cometidos por el usuario o usos del sistema en una forma no prevista durante el análisis y diseño).

4.2.4. El modelo de Carroll

Este autor proviene del área hci⁵, motivo por el cual considera un escenario como el uso (real o imaginario) que tiene el usuario de un sistema, analizando el comportamiento deseado o no deseado del individuo frente al mismo.

Un escenario describe en forma textual una situación particular de un usuario interactuando con el sistema. Esta información sirve como fuente para el diseño del sistema. A través del escenario se puede observar que es lo que hace el usuario con el sistema, como interactúa, como reacciona ante las respuestas y que problemas tiene. El conjunto de escenarios permite razonar sobre el comportamiento de los usuarios ante determinadas situaciones del sistema.

La creación de escenarios no puede basarse en la simple observación del sistema. A veces pueden necesitarse escenarios de sistemas que aún no existen o de los cuales no se conoce su uso. Esto se puede hacer por analogía con sistemas ya existentes (el nuevo sistema puede ser similar a otro ya existente o ser descendiente de otro sistema).

Si bien se menciona el uso de escenarios para la etapa de adquisición de requerimientos iniciales, el trabajo está enfocado en el uso de escenarios para diseño. El conjunto de escenarios muestra como actúan los usuarios ante determinadas situaciones, y sobre esta base se pueden discutir alternativas de diseño. Propone el diseño racional, a partir de un escenario analizar las relaciones causales del mismo: ante determinada situación se puede desencadenar una reacción favorable o no en el usuario. Cada situación del escenario debe ser analizada de la siguiente forma:

En <situación> <una expresión> causa <consecuencias deseables> pero puede causar <consecuencias indeseables>

A partir de aquí se pueden analizar escenarios alternativos en donde distintas condiciones del sistema intentan obviar las consecuencias no deseadas.

4.2.5. La propuesta de Booch

Según Booch, los escenarios cumplen tres principios fundamentales. En primer lugar los escenarios son una parte esencial para capturar los requerimientos. Los escenarios hablan el lenguaje del usuario final y del experto del dominio, por lo tanto proveen un medio para que ellos expliquen sus expectativas sobre el comportamiento del sistema. Segundo, los escenarios

⁵Human Computer Interfase

proveen un vehículo de comunicación. Llevan al usuario final y experto del dominio al nivel del problema, exigiendo al desarrollador a adquirir el dominio del problema, forzándolo a considerar una distribución inteligente de responsabilidades dentro del sistema. Y tercero, a medida que el proyecto avanza, los escenarios sirven como instrucciones tanto a los desarrolladores individuales, como al equipo de pruebas.

¿Qué es un escenario? Un escenario provee un esbozo del comportamiento del sistema. Los escenarios documentan decisiones de requerimientos o diseño, proveen un punto de comunicación sobre la semántica del sistema y pueden servir como punto de partida para la implementación detallada.

Luego, el comportamiento del sistema de software puede ser capturado a través de una red de escenarios de la misma forma que lo hace los storyboards con respecto a una película. Aunque, la analogía entre el proceso de desarrollo de soft y filmar una película se quiebra en el punto en que las películas tienen sólo un camino de acción y pueden ser representadas por un largo y continuo storyboard. En cambio, todas las aplicaciones de soft interesantes, raramente tienen un camino simple de comportamiento.

Booch utiliza distintos métodos para representar escenarios. En primer lugar usa tarjetas CRC. El probó que son una buena manera de abordar la construcción de escenarios. Su mayor atractivo como técnica de desarrollo es que son totalmente libres. Lamentablemente, las tarjetas CRC padecen de una gran limitación: no pueden considerar aspectos temporales de un escenario.

4.2.5. EL enfoque de Escenarios de Leite.

La metodología de Leite [Leite 95] es a la que se adhiere este trabajo de tesis, la misma propone trabajar con documentos en lenguaje natural altamente vinculados entre sí, centrados en una estructura orientada al cliente llamada Requirements Baseline, siendo está un metamodelo que contiene descripciones sobre el dominio del problema y el artefacto de software que ha de ser construido dentro de ese dominio. 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.

Los escenarios tienen una estructura compuesta por el Título que lo identifica, el Objetivo a lograr en el macrosistema, el Contexto que describe la ubicación geográfica y temporal del escenario, así como un estado inicial o precondition, también se especifican los Recursos necesarios que estén disponible, los Actores que tienen un rol en el escenario y por ultimo los Episodios que son una serie ordenada de sentencias escritas en lenguaje natural.

El Objetivo, el Contexto, los Recursos y los Actores son sentencias declarativas, mientras que los Episodios son un conjunto de sentencias con un lenguaje muy simple que hace posible la descripción operativa de comportamientos.

Un episodio puede concebirse como un escenario en sí mismo, esto posibilita la descomposición de un escenario en subescenarios.

En estudios anteriores [Hadad 97] se han propuesto heurísticas para derivar los escenarios desde el LEL. Estas heurísticas permiten generar una primera versión de los escenarios. Se detectan los actores en el LEL y se utilizan sus impactos y sus vínculos para describir parcialmente los escenarios. Luego, se completan los escenarios validándolos con el usuario y ampliando la información faltante o confusa.

4.2.5. Comparación de los distintos modelos.

Como los enfoques basados en escenarios atraen un interés creciente entre los ingenieros de requerimientos, la literatura sobre métodos, modelo y notaciones de escenarios prolifera. En función de las definiciones vistas y teniendo en cuenta la metodología a la que adhiere esta tesis (ver cuadro 5), podemos considerar que una de las más coincidentes es el modelo ICM. En él, podemos ver que se tienen en cuenta los actores, sus objetivos, las acciones que realizan para alcanzarlos y los posibles obstáculos que impiden alcanzar esos objetivos. Un elemento que aparece en casi todos los trabajos, son las pre y post condiciones. Las mismas, dan una idea de estado inicial y final de un escenario como se hace en el modelo de OBA. Los distintos trabajos, si bien lo denominan de otra forma, también tienen presente esta idea.

Con respecto a los actores, si bien en todas las representaciones se mencionan, en el único trabajo en que se los identifican como entidades externas del sistema es en el modelo de use cases. Esto significa que es necesario conocer los límites del sistema, los que en realidad se están tratando de definir. No siempre es fácil identificar en un primer momento, qué entidades son externas y cuáles internas al sistema.

La propuesta de Booch y los use cases destacan los beneficios de crear un sistema de descripción concreta orientada al uso, antes de modelar la función, los datos y el comportamiento. Las extensiones y alternativas propuestas incluyen el agregado de estructura a los use cases; el tratamiento formal de escenarios y el uso de escenarios durante la documentación, discusión y evolución de los requerimientos.

Los escenarios también son populares en otros campos, como en la interacción hombre-computadora [Carroll] y planeamiento estratégico.

	Objetivo	Contexto	Recursos	Actores	Episodios
Use Case	Sí	No	No	Sí	No
OBA	Sí	Sí	No	Sí	No
ICM	Sí	Sí	No	Sí	Sí
Carroll	Sí	No	No	Sí	Sí
Booch	Sí	No	No	No	Sí

Cuadro 6. Comparación de distintos modelos de Escenarios.

4.2.6. Uso de Escenarios en la Practica

Mientras los escenarios permiten un aprendizaje interdisciplinario en ingeniería de requerimientos, ellos también sirven como medio para divisiones de trabajo, con consecuencias significantes para el proyecto de administración e integración del artefacto de software. De la investigación que se informa en el artículo Scenarios in System Development: Current Practice publicado en [IEEE 98], se puede resumir algunas de las observaciones a las que llegaron luego de evaluar 15 proyectos desarrollados en 4 países europeos.

- ◆ **Uso de escenarios cuando falla el modelado abstracto**

Cuando algunos proyectos fracasaron al desarrollar los modelos abstractos conceptuales como los modelos de clase, debido a la complejidad del dominio del problema, los desarrolladores utilizaron escenarios para elicitación y documentar los requerimientos del cliente, satisfactoriamente en ambos casos.

- ◆ **Los Escenarios requieren la coexistencia de prototipos.**

En las dos terceras partes de los proyectos, la generación y uso de escenarios se vinculó con una rápida prototipación o aún construcción de las primeras versiones del nuevo sistema.

Los escenarios iniciales sirvieron para validar los prototipos e, indirectamente, la especificación de requerimientos. La evaluación de los prototipos condujo a la detección de malentendidos entre los expertos de dominio y los desarrolladores. Resolver tales malentendidos se torna más fácil con escenarios como una base común para la comunicación.

- ◆ **Los Escenarios facilitan el acuerdo parcial y la consistencia**

Los involucrados en un sistema bajo desarrollo tienen diferentes objetivos, aún sus percepciones de la realidad varían significativamente. Juntar a todos los involucrados y alcanzar un acuerdo general en los aspectos de un sistema consume demasiado tiempo y a veces es imposible. Similarmente, asegurar que el sistema a construirse está conforme con todos los aspectos de los sistemas existentes en la organización es impracticable. No obstante, casi todos los proyectos usaron escenarios para manejar el proceso de acuerdo y establecer consistencia parcial entre los sistemas existentes.

◆ **Escenarios de enlace y glosarios**

En varios proyectos, el vincular escenarios con un glosario amplio del proyecto estableció un entendimiento común de los términos usados entre diferentes grupos: desarrolladores, expertos de dominio y administradores.

Se notó una relación bidireccional interesante entre escenarios y glosarios. Los desarrolladores establecieron una infraestructura de hipertexto en una intranet de proyecto amplio que enlaza las partes correspondientes al escenario con los ítems del glosario.

Estos enlaces ayudaron a los desarrolladores y expertos del dominio a ajustar sus interpretaciones a los términos claves usados y así alcanzar un entendimiento común del proyecto. Más aún, estas relaciones ayudaron a nuevos miembros del proyecto a familiarizarse con la terminología del proyecto.

Podemos apreciar que estas conclusiones tienen una gran similitud con las características del uso del Léxico Extendido del Lenguaje.

◆ **Evolución de Escenarios**

En todos los proyectos, la definición de escenarios no fué una actividad corta, los escenarios evolucionaron a través del tiempo.

◆ **Derivando casos de prueba de los escenarios**

Casi todos los desarrolladores que se entrevistaron en este estudio, mencionaron la necesidad de basar las pruebas de sistema en los escenarios definidos con el cliente durante la etapa de Ingeniería de requerimiento. Esto significa apoyar al desarrollador para que pueda probar al cliente, que el sistema implementado alcanza los requerimientos.

◆ **Traceability**

Muchos desarrolladores también mencionaron la necesidad de un mayor apoyo de traceability, con traceability visto como un prerequisite para establecer el amplio ciclo de vida del uso de escenarios definidos.

Los escenarios desactualizados fueron inconsistentes con el diseño actual de versiones de prototipo y además no pudieron ser usados como base para casos de prueba. La Traceability posibilita la integración de cambio, ayudando a usuarios a mantener escenarios actualizados. Los desarrolladores deben establecer traceability entre niveles de abstracción de los escenarios, versiones de escenarios, escenarios y prototipos, escenarios y la especificación, y escenarios y casos de prueba. Establecer traceability requiere entender la relación entre los artefactos producidos durante el desarrollo del proyecto y los escenarios.

Por ultimo, podemos decir que del trabajo de investigación también se desprende que los usuarios necesitan mejores herramientas para manejar los escenarios y sus relaciones. Observaron una frecuente falta de apoyo apropiado de herramientas y la falta de capacidad para asegurar manualmente la consistencia entre escenarios o entre escenarios y otros artefactos. Difícilmente

dos proyectos cualquiera usen las mismas herramientas para el manejo de escenarios, excepto por el procesador de texto. Esto indica que no existe ninguna herramienta aceptada generalmente.

4.3. Esquema de descripción de Escenarios

En esta sección se describe la estructura que se adopta en este trabajo para representar los escenarios.

La vista del modelo de escenarios aplicada es una estructura compuesta por el **nombre**, el **objetivo**, el **contexto**, los **recursos**, los **actores** y los **episodios**. El objetivo, el contexto, los recursos y los actores son sentencias declarativas, mientras que los episodios son un conjunto de sentencias con un lenguaje muy simple que hace posible la descripción operativa de comportamientos.

Para la descripción de escenarios se utilizó el template descrito en el Cuadro 7 [Leite]:

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.

Sintaxis:

Frase | ([Actor | Recurso] + Verbo + Predicado)

Objetivo: finalidad a ser alcanzada en el contexto del problema. El escenario describe el logro del objetivo.

Sintaxis:

[Sujeto] + Verbo + Predicado

Contexto: ubicación geográfica y temporal del escenario, y/o estado inicial del mismo.

Sintaxis:

Ubicación + Estado

donde Ubicación es:

Nombre

donde Estado es:

[Actor | Recurso] + Verbo + Predicado + {Restricciones}

Recursos: medios de soporte, dispositivos u otros elementos pasivos necesarios para estar disponibles en el escenario.

Sintaxis:

Nombre + {Restricciones}

Actores: personas o estructuras organizacionales que tienen un rol en el escenario.

Sintaxis:

Nombre

Episodios: conjunto de acciones que detallan el escenario y proveen su comportamiento.

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}

Cuadro 7 - Esquema para la descripción de escenarios

En la Figura 3, se describe el modelo de escenarios usando el diagrama de entidad-relación. Se observa en dicho diagrama que un episodio puede

concebirse como un escenario en sí mismo, esto posibilita la descomposición de un escenario en sub-escenarios.

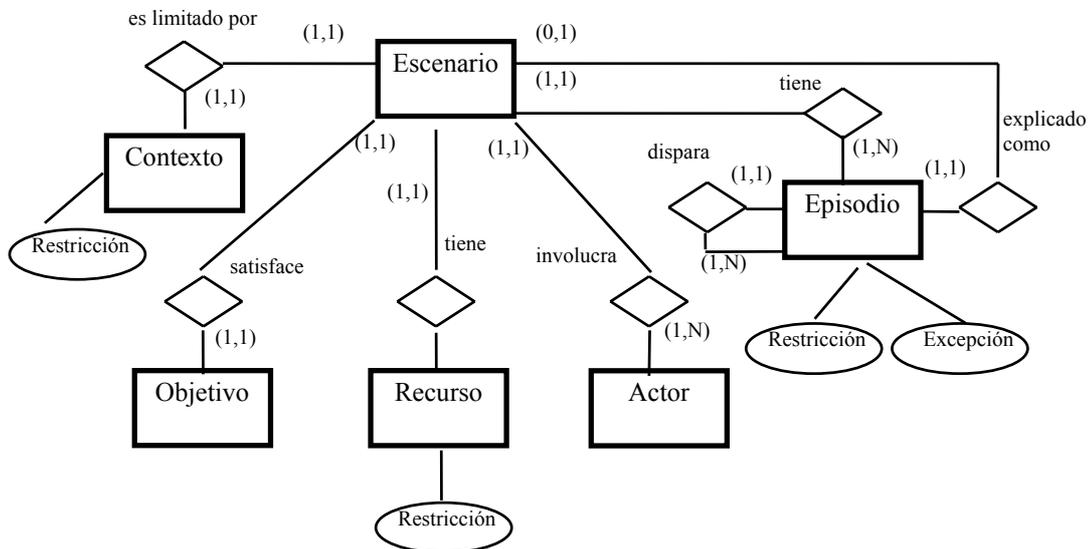


Figura.3: Diagrama de Entidad-Relación para el Modelo de Escenarios.

4.4. Proceso de construcción de Escenarios

Algunos trabajos de investigación han profundizado en el estudio del LEL y Escenarios aplicándolos a varios casos de estudio [Hadad96] y [Hadad97], lo que ha permitido desarrollar heurísticas para la construcción de escenarios, las cuales serán implementadas en la herramienta que presenta esta tesis.

La construcción de los escenarios se basa exclusivamente en la información contenida en el LEL, al que se le aplican una serie de heurísticas, que se detallan más adelante en esta sección, para obtener una primera lista de escenarios candidatos.

Posteriormente se realizan nuevas entrevistas con los clientes, con el fin de ampliar información sobre aquellos escenarios cuya descripción resultara incompleta y/o confusa a partir de la información obtenida del LEL.

Esto implica que la derivación de escenarios que se detalla en el punto siguiente es el comienzo de la descripción de los escenarios, siendo luego necesario retornar al Universo de Discurso para concluirla.

Los escenarios se describen utilizando el esquema informado en la sección 4.3. y se emplea el **punto de vista** de la aplicación.

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 4 se presentan gráficamente las etapas mencionadas y a continuación se detalla cada una de estas etapas.

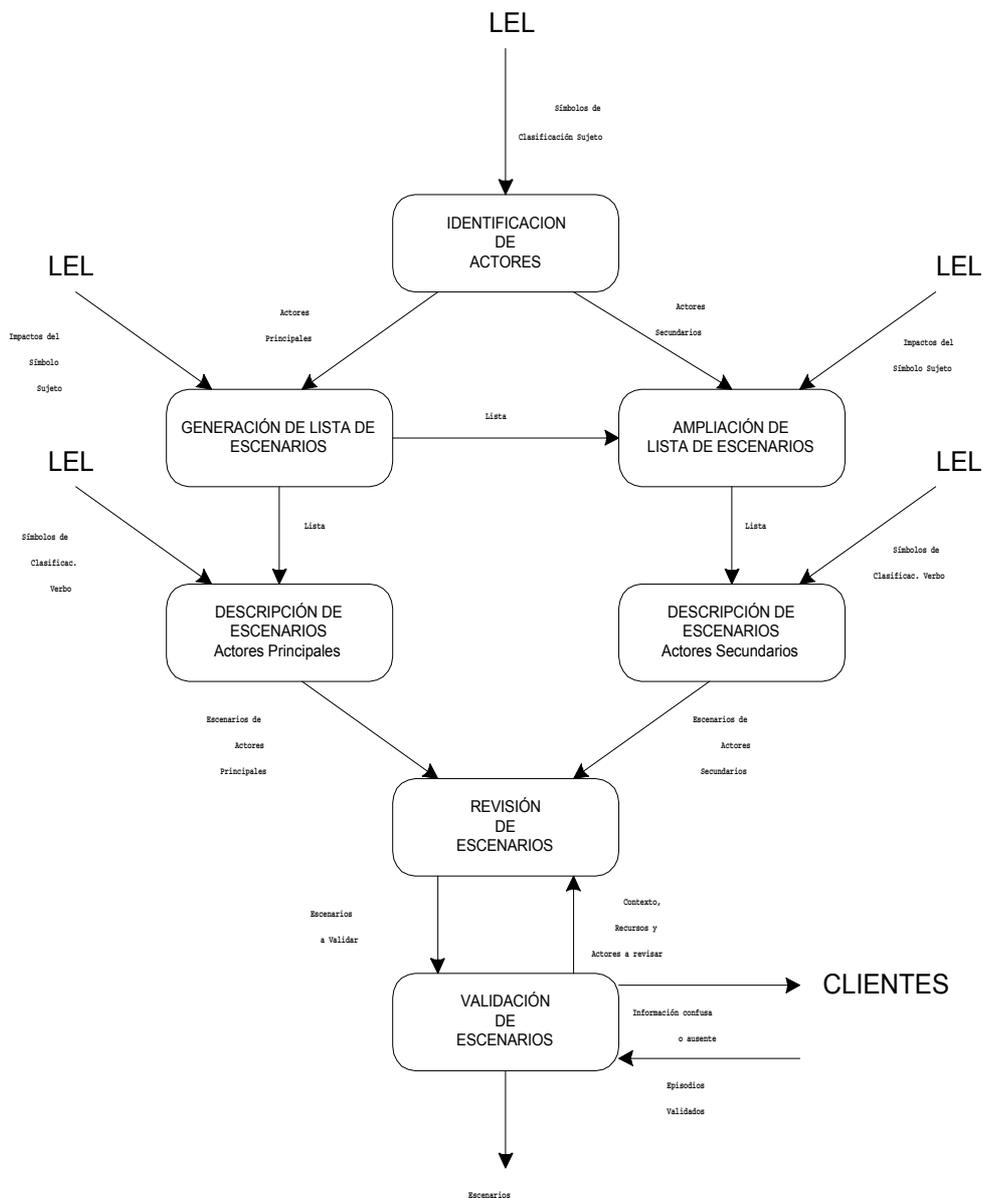


Figura. 4:- Etapas para la construcción de escenarios

4.4.1. 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.

4.4.2 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.

4.4.3. Descripción de los Escenarios candidatos, provenientes de los actores principales.

A partir de la lista candidata, se describe cada uno de los escenarios, utilizando el esquema descrito en el Cuadro 5. 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**.

4.4.3.1 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.

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

Se identifican **símbolos** dentro del **impacto**.

Se buscan dichos símbolos en el LEL para obtener mayor información sobre el posible escenario.

Se define el **objetivo** en función del nombre del escenario y el punto de vista de la aplicación.

Se identifican posibles **actores** y **recursos** del escenario, a través de los símbolos asociados, pertenecientes a la **clasificación Sujeto y Objeto** respectivamente.

No se describen episodios partiendo del LEL, se deja su definición para un paso posterior.

Para la definición del **contexto** se observa si el impacto del símbolo que lo originó tenía alguna secuencia con respecto a los restantes impactos, en caso afirmativo éstos se incorporan como precondition.

Se completa el ítem **dudas** con información confusa o ausente.

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

4.4.4. 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 4.4.2, para obtener otros escenarios candidatos e incorporarlos a la lista.

4.4.5. Descripción de los Escenarios candidatos, provenientes de los actores secundarios.

Se aplica el procedimiento descrito en el punto 4.4.3 a los nuevos escenarios incorporados a la lista en el punto anterior.

4.4.6. Revisión de los Escenarios.

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

4.4.6.1 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, ver punto 4.4.3.2.

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 **sub-escenarios**.

En caso contrario, se elimina de la lista el escenario candidato.

4.4.6.2 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 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 **sub-escenarios**.

4.4.6.3 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.

4.4.6.4 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.

4.4.6.5 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.

4.4.7. Validación de Escenarios.

4.4.7.1 Se **valida** con los **clientes** cada uno de los escenarios incluidos en la lista definitiva, prestando especial atención al ítem dudas que 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.

4.4.7.2 Se realizan las **correcciones** necesarias en cada ítem del escenario.

En el caso de ampliación de información para un episodio, según su complejidad, 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.

4.4.7.3 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.

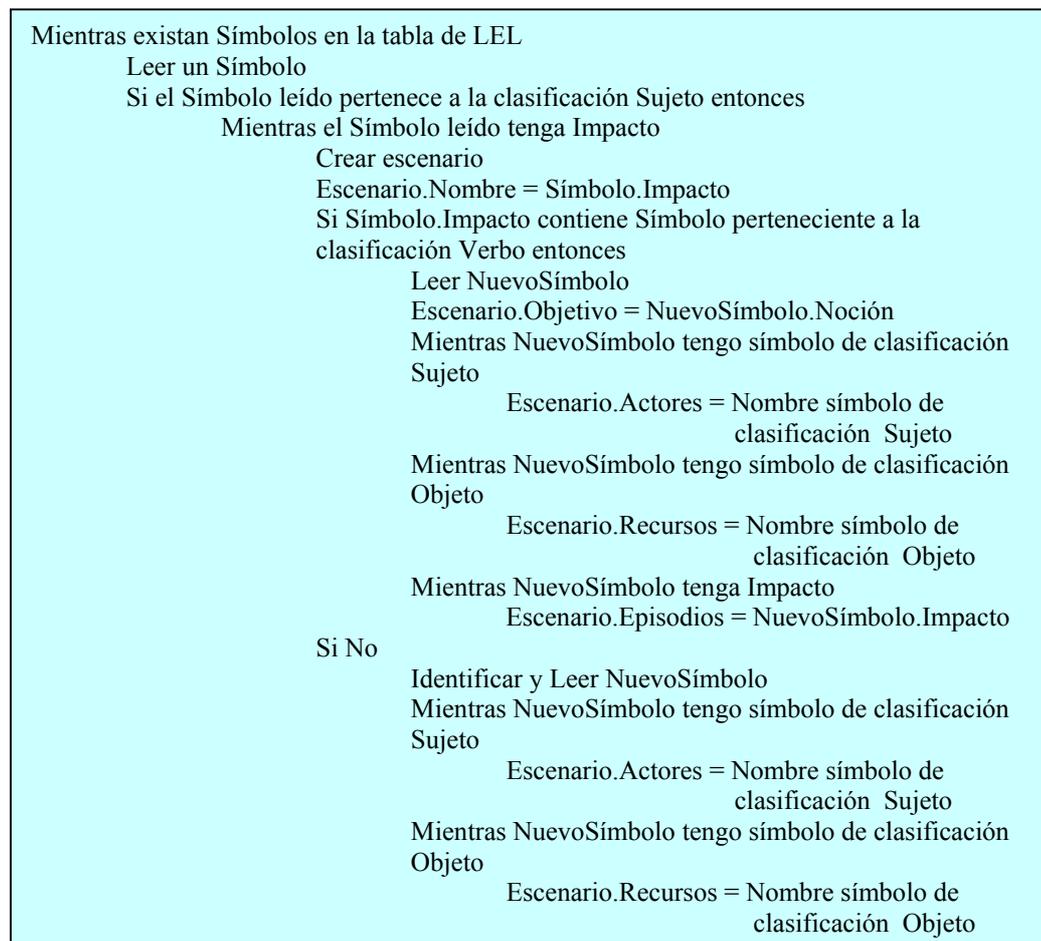
A partir de este punto, se obtiene la lista definitiva de escenarios y sus descripciones, correspondientes a la fase de elicitación de requerimientos.

4.5. Algoritmo para generar Escenarios

Desde el punto de vista informático, podemos pensar que la herramienta contara con un Repositorio en el que se almacenara toda la información relacionada con cada proyecto que el usuario esté desarrollando. Entre los componentes de este repositorio habrá una Base de Datos formada por una tabla de LEL, una tabla de Escenarios y un documento que contiene el Universo del Discurso obtenido.

La herramienta, a partir de la base de datos del LEL, identificara los símbolos que representan a los actores del Universo de Discurso. Estos símbolos pertenecen a la clasificación Sujeto y cada impacto representa un posible escenario, que se incorpora a la lista de escenarios candidatos.

La herramienta, a medida que genera la lista de escenarios candidatos, simultáneamente, describirá cada uno de los escenarios, aplicando las reglas que se detallaron en los puntos 4.4.2 y 4.4.5 de este informe. TILS verifica si el impacto que genera un escenario candidato contiene un símbolo del LEL que pertenezca a la clasificación Verbo, si ocurre esto entonces es posible generar la descripción de dicho escenario. A continuación en el cuadro 8 se expone el algoritmo que representa lo expuesto:



Cuadro 8: Algoritmo para generar escenarios candidatos.

5. HERRAMIENTA PARA IMPLEMENTAR LEL Y ESCENARIOS (TILS).

5.1. Generalidades.

5.1.1. Motivación.

En la asignatura Tópicos de Ingeniería I de esta Maestría, se utilizó la Metodología de LEL y Escenarios para apoyar la fase de ingeniería de requerimientos, en un caso de estudio real. La experiencia adquirida fue muy positiva y se pudo verificar que si los desarrolladores comparten el mismo lenguaje que los usuarios, se facilita notablemente el conocimiento del dominio de cualquier aplicación, evitando con esto que se produzcan subconjuntos de afirmaciones contradictorias entre sí, en la Especificación de Requerimientos de Software [IEEE 84], fortaleciendo a su vez, la relación usuario-desarrollador en una etapa muy temprana de desarrollo.

La construcción del LEL fue hecha manualmente y luego transcrita en un editor de texto. Cuando se definían nociones e impacto había que realizar el marcado de los símbolos ya existentes de forma manual. Este tipo de proceso es lento y entorpece la labor del desarrollador, sobre todo en la etapa de validación de símbolos, donde es necesario completar información, hacer cambios, eliminar y agregar nuevas entradas, lo cual insume mucho tiempo en tareas que no hacen a la elicitación de requerimientos.

En la construcción de los escenarios ocurre algo similar, agravado por el hecho de que es posible generar automáticamente una versión preliminar de Escenarios a partir de las heurísticas propuestas en el trabajo de Hadad 96.

Todo esto justificó el desarrollo de una herramienta CASE para soportar la producción de LEL, incorporando en la misma tecnología de hipertexto para facilitar la navegación entre los distintos símbolos que forman el léxico. De esta forma es posible interactuar de una manera mucho más flexible en la navegación y administración de la información obtenida para conocer todo el vocabulario del dominio.

La herramienta también puede generar una primera versión de los escenarios involucrados en el proyecto en desarrollo, permitiendo luego depurarlos, para lograr así una versión definitiva. Esto origina cambios en el proyecto, por lo que es necesario que TILS proporcione una forma simple de traceability, consiguiendo con esto: una documentación consistente, facilidad para capacitar a nuevos miembros que se incorporen al equipo de desarrollo en la terminología empleada y generar versiones del proyecto a medida que evoluciona el proceso de desarrollo.

5.1.2. Implementación.

El ser humano siempre se ha distinguido del resto de los seres terrestres por su capacidad de crear elementos no existentes en la naturaleza. Ciertamente es que algunos animales superiores son capaces de utilizar como herramientas algunos elementos, pero se desconoce que ninguno de ellos sea capaz de elaborar la herramienta más simple que el hombre utilizaba hace un millón de años.

Desde entonces hasta ahora ha pasado mucho tiempo, suponemos que un millón de años, y la capacidad de creación del hombre no se ha detenido en ningún momento. En este instante, sin embargo, esta capacidad se ve potenciada de forma extraordinariamente fuerte por la tecnología, y particularmente por la computación. En una computadora, esto es posible. Se pueden crear elementos intangibles, que se diferencian por completo de cualquier otro elemento material que el hombre haya podido construir anteriormente, pero que sin embargo nos facilitan y hacen más cómoda nuestra vida.

Todos sabemos lo que es un lenguaje, un conjunto de palabras con significado que, unidas apropiadamente según unas reglas sintácticas previamente establecidas y aceptadas por todos los usuarios, dan origen a construcciones que facilitan la comunicación. Los lenguajes informáticos son una versión muy simplificada y restringida del lenguaje humano, ya que, por ahora, las computadoras no tienen nuestra capacidad intelectual. Mediante un lenguaje de programación podemos comunicarnos con el ordenador, lo que nos permite instruirle acerca de los pasos que debe dar para realizar un cierto proceso.

TILS está implementado en Visual Basic 5.0. En la actualidad, Visual Basic se configura como una potente herramienta de desarrollo dotada de una gran versatilidad que permite la creación de avanzadas aplicaciones en entornos distribuidos (llámense cliente/servidor de varias capas o Internet/Intranet), trabajos con clases, la creación de Servidores de Automatización, los Objetos de Datos Remotos (RDO), el uso del API de Windows, el control de dispositivos, y las últimas tecnologías para el desarrollo en Internet/Intranet.

5.2. Funcionalidad de TILS.

5.2.1 El entorno de trabajo

El entorno de trabajo de TILS es bastante sencillo y fácil de usar una vez que se está familiarizado con él. Cuando se ejecuta TILS aparece la pantalla principal que se muestra en la figura 5:

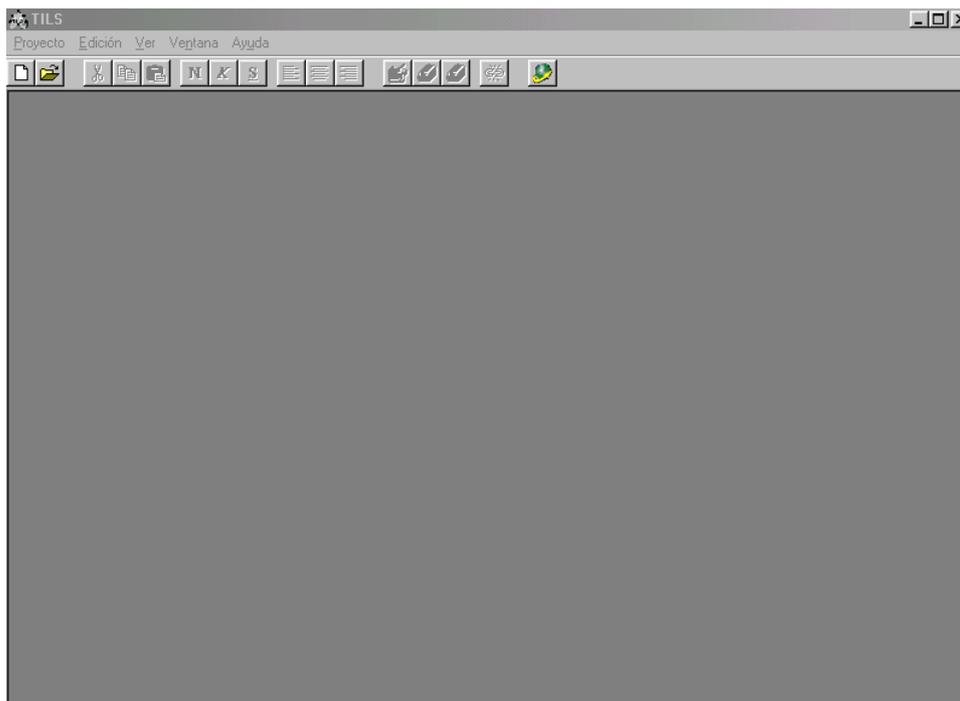


Figura 5: Pantalla principal de TILS

5.2.1.1 Barra de título

En la parte superior de la pantalla se encuentra la barra de título, que muestra inicialmente el nombre de la aplicación. Luego, cuando se abre un proyecto existente o se cree uno nuevo se agregará en esta barra el nombre de dicho proyecto.

5.2.1.2 Barra de menús

Debajo de la barra de título se encuentra la barra de menús, con el que se accede a los comandos que permiten construir el proyecto. La barra de menús de TILS consta de los menús Proyecto, Edición, Ver y Ventana.

5.2.1.3 Barras de herramientas

Proporcionan un rápido acceso a los comandos usados normalmente en el entorno de programación. Haciendo clic en un botón de la barra de herramientas se lleva a cabo la acción que representa ese botón.

5.2.1.4 Área de trabajo

TILS fue creada al estilo de las aplicaciones MDI, o aplicaciones cuya interfaz es capaz de manejar múltiples documentos. Las MDI mantienen varios formularios "hijos" dentro de un único formulario (el "padre"). Las aplicaciones SDI (Single Document Interface), o aplicaciones de interfaz mono-documento, por otra parte, están basadas en una única ventana que se utiliza para todas las interacciones con el usuario.

Las aplicaciones Bloc de Notas y WordPad que se entregan con Windows 95 son ejemplos de aplicaciones SDI. En cambio Word y Excel son aplicaciones MDI clásicas. En las aplicaciones MDI el usuario puede trabajar fácilmente con varios grupos de datos al mismo tiempo, puede arrastrar y soltar información entre las diferentes ventanas hijas. Empleando el menú Ventana para moverse entre las ventanas hijas.

En Visual Basic las aplicaciones MDI contienen un único formulario padre, llamado formulario MDI.

Los formularios de una aplicación MDI tienen características especiales en tiempo de ejecución, que pasamos a detallar:

- Todos los formularios hijos están restringidos al área cliente de su formulario padre.
- El icono de un formulario MDI hijo minimizado aparece dentro de la ventana de su formulario padre en lugar de aparecer en el escritorio como es normal. De la misma manera, al minimizar el formulario MDI padre se minimizan al mismo tiempo todos sus formularios hijo. Todo este conjunto "familiar" se representa con un único icono. De la misma manera, al restaurar el tamaño del formulario padre, se restauran todos sus formularios hijo.
- El título de un formulario MDI hijo maximizado se combina con el título de su formulario padre y se muestra como título del formulario.

5.2.2. Proyectos.

El usuario necesariamente deberá ingresar en este menú para poder comenzar a trabajar en un proyecto TILS, las opciones que se encuentran activas son: Nuevo, Abrir y Salir, como podemos ver en la Figura 6:

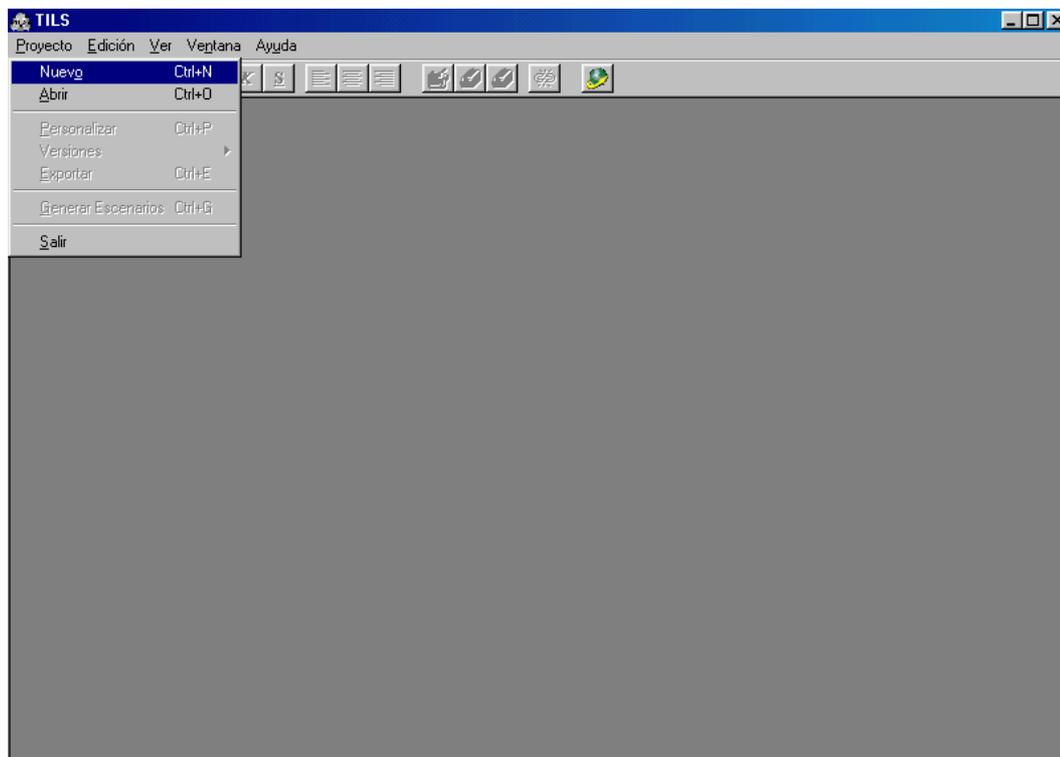


Figura 6: El menú Proyecto

Las funciones de estas opciones son:

5.2.2.1 Opción Nuevo

Permite crear un nuevo Proyecto, cuando se selecciona esta opción, se genera una base de datos con toda la información necesaria para contener los datos del nuevo proyecto en el que se va a trabajar. En primer lugar se presenta una ventana de diálogo como la que se puede ver en la siguiente figura:



Figura 7: Pantalla de diálogo para crear un proyecto

Este cuadro de diálogo es el mismo que utiliza Windows para este tipo de operaciones. Las personas que trabajan con este sistema operativo están familiarizados con el mismo, con lo cual no es necesario detallar en profundidad su funcionamiento. La característica de este cuadro de diálogo es que permite al usuario seleccionar una unidad de disco, un directorio y un nombre de Proyecto.

5.2.2.2 Opción Abrir

Permite abrir un proyecto existente. Cuando se selecciona esta opción, se despliega un cuadro de diálogo similar al de la opción Nuevo, como se puede ver en la figura 8:



Figura 8: Pantalla de diálogo para abrir un proyecto

Una vez seleccionado el proyecto que se desea abrir, se presiona el botón abrir o bien, hacer doble click en el nombre del proyecto para que se ejecute la acción.

5.2.2.3 Opción Salir

Esta opción finaliza la aplicación.

Como se expuso antes, estas son las opciones habilitadas cuando se ejecuta la aplicación. Una vez que se haya creado un proyecto nuevo o abierto uno existente, el menú proyecto cambia su apariencia habilitando cuatro nuevas opciones, como se ve en la figura 9:

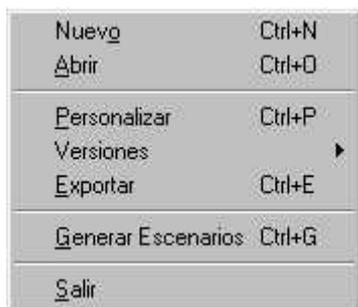


Figura 9: Menú proyecto

Ahora están disponibles cuatro nuevas opciones: Personalizar, Versiones, Exportar y Generar Escenarios. Por el momento solo se analizará la opción Personalizar, dejando para más adelante el estudio de las otras tres.

5.2.2.4 Opción Personalizar

Esta opción está relacionada con la clasificación de las entradas al LEL, tema visto en el punto 3.3.4 de este mismo informe.

Por defecto, TILS propone una clasificación de símbolos como se puede ver en el cuadro 9:

Tipo	Categoría
Sujeto	Personas (S) Sectores (P)
Objeto	Almacenamiento (AI) Formularios (F) Índices (I) Términos Específicos (TE)
Verbo	Acciones (A) Métodos Específicos (ME)

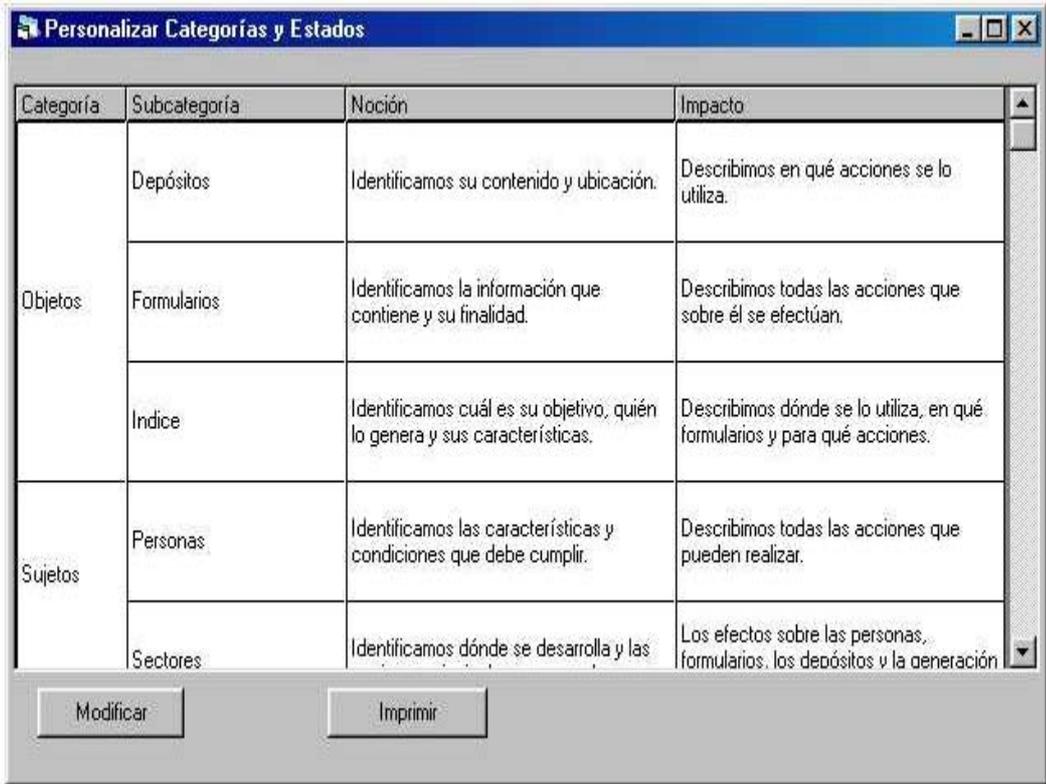
Cuadro 9: Clasificación de los Símbolos

A su vez, también incorpora una definición para la Noción y el Impacto de cada subclasificación, como se ve en el cuadro 10:

Clasificación	Noción	Impacto
Persona	Indicamos las características y condiciones que debe cumplir.	Describimos todas las acciones que pueden realizar.
Sector	Identificamos dónde se desarrolla y las acciones principales que cumple.	Los efectos sobre las personas, formularios, los almacenamientos y la generación o uso de índices.
Almacenamiento	Indicamos su contenido y ubicación.	Describimos en qué acciones se utiliza.
Formulario	Identificamos la información que contiene y su finalidad.	Describimos todas las acciones que sobre él se efectúan.
Índice	Identificamos cuál es su objetivo, quién lo genera y sus características.	Describimos dónde se utiliza, en qué formularios y para qué acciones.
Término Específico	Definimos su significado, sus características y en que sector o proceso se lo identifica.	Describimos que se hace con él o que se le hace.
Acción	Identificamos el sector donde se lleva a cabo y el objetivo de la acción.	Describimos los efectos sobre las personas, los formularios, los depósitos y la generación o uso de índices.
Método Específico	Identificamos el sector <i>donde</i> se lleva a cabo, el <i>objetivo</i> y características particulares del método	Describimos los <i>efectos sobre</i> las personas, los formularios, los depósitos y la generación o uso de índices.

Cuadro 10: Definición de Noción e Impacto de los Símbolos

Cuando se elige la opción Personalizar del menú Proyecto, se puede ver una ventana con toda la información antes mencionada, como muestra la figura 10:



Categoría	Subcategoría	Noción	Impacto
Objetos	Depósitos	Identificamos su contenido y ubicación.	Describimos en qué acciones se lo utiliza.
	Formularios	Identificamos la información que contiene y su finalidad.	Describimos todas las acciones que sobre él se efectúan.
	Índice	Identificamos cuál es su objetivo, quién lo genera y sus características.	Describimos dónde se lo utiliza, en qué formularios y para qué acciones.
Sujetos	Personas	Identificamos las características y condiciones que debe cumplir.	Describimos todas las acciones que pueden realizar.
	Sectores	Identificamos dónde se desarrolla y las	Los efectos sobre las personas, formularios, los depósitos y la generación

Modificar Imprimir

Figura 10: Ventana personalizar

Si bien TILS propone por defecto esta subclasificación con la definición de la Noción e Impacto, el usuario puede realizar las modificaciones que crea conveniente, en función del proyecto que está desarrollando. Esto se logra presionando el botón Modificar, el cual activa la ventana que vemos en la figura 11:



Figura 11: Opción modificar

En esta ventana se puede agregar, modificar o eliminar subcategorías. La forma de realizar estas acciones es la siguiente:

Agregar subcategoría:

1. Se selecciona la categoría a la que pertenecerá la nueva subcategoría.
2. Se ubica el cursor en la primer celda vacía de la grilla de subcategoría y escribimos el nombre que tendrá dicha subcategoría.
3. Si está definida la Noción e Impacto, es posible incorporarlos presionando el botón de la celda actual, lo que provocará la aparición de una nueva ventana, como se puede ver en la figura 12:

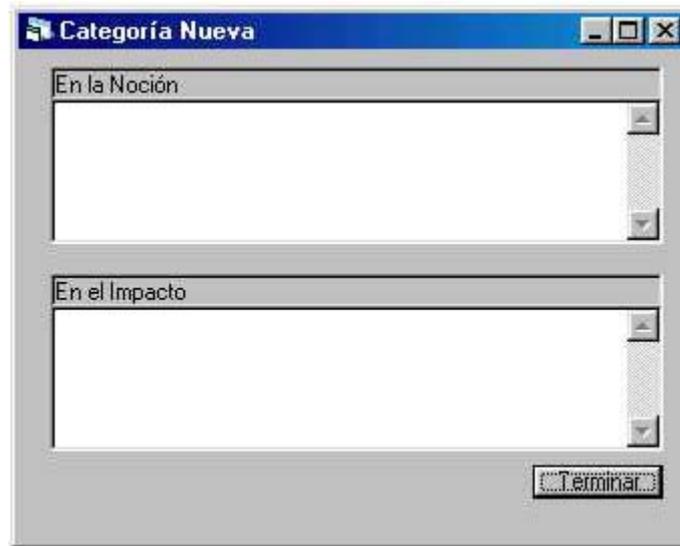


Figura 12: Ventana para agregar Noción e Impacto a una subcategoría nueva

4. Una vez definida la Noción e Impacto, se presiona el botón Terminar y se regresa a la ventana principal.
5. Si el usuario necesita agregar nuevas subcategorías, se repiten los pasos 1 al 4.
6. Si no se desea agregar más subcategorías, se presiona el botón Terminar de la ventana Personalizar.

Modificar una Subcategoría existente

1. Se selecciona la subcategoría que se desea modificar.
2. Se concreta la modificación.
3. Si es necesario modificar su Noción e Impacto, se presiona el botón de la celda, y aparece nuevamente la figura 12.
4. Se ubica el cursor en el punto donde se desea realizar la modificación y se la concreta.
5. Se presiona el botón Terminar.
6. Nuevamente, si es necesaria otra modificación, se repiten los pasos 1 al 5.
7. En caso contrario se presiona el botón terminar.

Eliminar una Subcategoría

1. Se selecciona la subcategoría que se desea eliminar.
2. Se presiona la tecla Del o Supr.
3. La aplicación mostrará un mensaje solicitando confirmación para esta acción.
4. Si se presiona Aceptar, la subcategoría será eliminada permanentemente.
5. Si se desea eliminar más Subcategorías, se repiten los pasos 1 al 4.
6. En caso contrario se presiona el botón terminar.

En la figura 11, se puede notar que existe una segunda tarjeta cuyo título es Estados, como lo indica la figura 13:



Figura 13: Estados para los símbolos del LEL y los Escenarios.

Estos estados también son definidos por defecto por TILS, y su función es facilitar la construcción del LEL y los Escenarios, permitiendo al usuario definir por ejemplo, si un Símbolo está completo, con dudas, etc.

Para los estados, se pueden agregar nuevos, modificar o eliminar

existentes según el usuario crea conveniente.

Agregar un Estado nuevo

1. Se posiciona el cursor en la última celda libre.
2. Se escribe el nombre del nuevo estado.
3. Si se desea agregar más estados, se repiten los pasos 1 y 2.
4. En caso contrario se pulsa el botón terminar.

Modificar un Estado existente

1. Se selecciona el Estado que es necesario modificar.
2. Se realiza la modificación.
3. Si es necesaria otra modificación, se repiten los pasos 1 y 2.
4. En caso contrario se pulsa el botón terminar.

Eliminar una Subcategoría

1. Se selecciona el Estado que se desea eliminar.
2. Se presiona la tecla Del o Supr.
3. La aplicación mostrará un mensaje solicitando confirmación para esta acción.
4. Si se presiona Aceptar, el Estado será eliminado permanentemente.
5. Si se desea eliminar más Estados, se repiten los pasos 1 al 3.
6. En caso contrario se presiona el botón terminar.

5.2.3. Universo del discurso.

El Universo del Discurso se puede traducir en un documento que contenga la información que se obtuvo mediante las entrevistas o documentos que se hayan consultado para tener más conocimiento del dominio del problema.

En este documento podremos encontrar las palabras o frases que el cliente repite o enfatiza. Estas palabras o frases son la fuente para construir la primer lista de símbolos candidatos a ser entradas al LEL. Para ello, TILS incorpora un pequeño editor de texto para poder transcribir dicha información.

Cuando el usuario crea un proyecto nuevo, o abre uno existente, se abre esta ventana editor, como podemos ver en la figura 14:

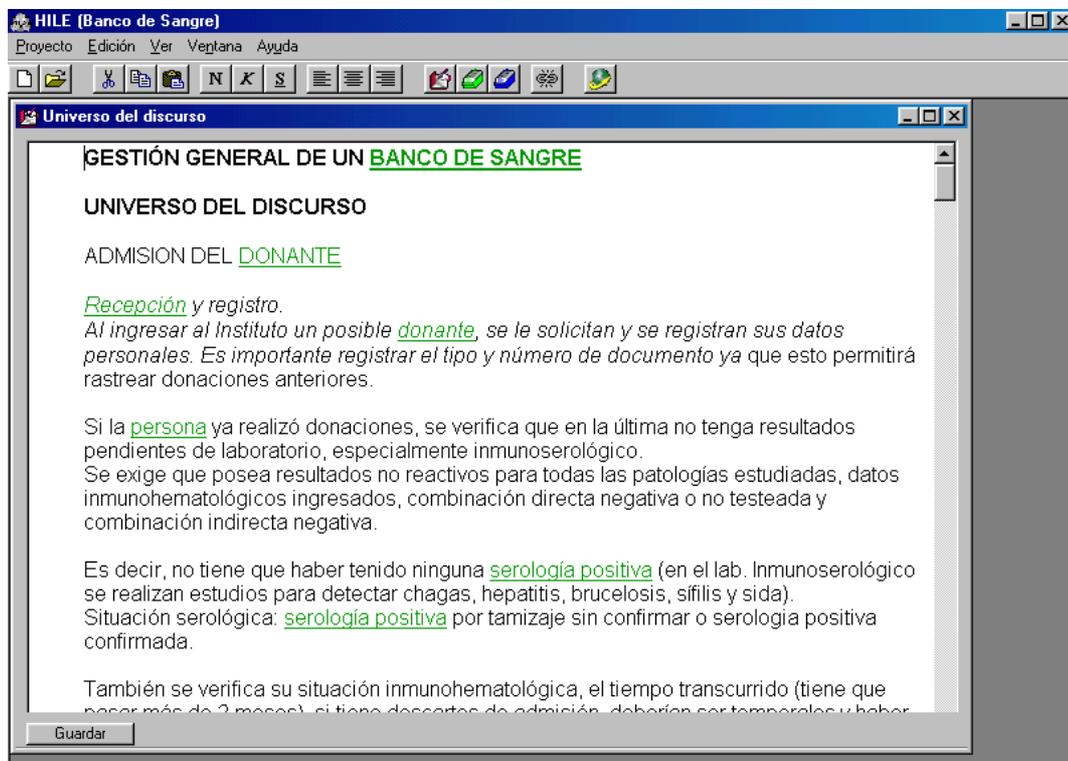


Figura 14: Editor de TILS.

También es posible acceder al editor con el menú Ver, cuya primer opción es Universo del Discurso, como lo indica la figura 15:



Figura 15: Opción Universo del Discurso del menú Ver.

Esta opción es necesaria, ya que si bien el editor se activa automáticamente cuando se comienza a trabajar en un proyecto, también es cierto que es posible minimizarlo o cerrarlo en el momento que sea necesario.

Otra forma de abrir el editor, es con la barra de herramientas, con el icono correspondiente al editor, Figura 16.



Figura 16: Botón para activar el editor.

5.2.4. Versiones.

En el punto 3.1 se trató el tema de las relaciones establecidas en términos de objetivos a alcanzar con la construcción del LEL y las consecuencias que se derivan de la satisfacción de esos objetivos relacionados con el uso del LEL en un proyecto de desarrollo.

El segundo objetivo planteado era el de contar con un instrumento simple de traceability, logrando con esto una documentación consistente, poder capacitar fácilmente nuevos miembros que se incorporen al equipo de desarrollo en la terminología empleada y generar versiones del LEL a medida que evoluciona el proceso de desarrollo. Además esta misma idea es aplicable al Universo del Discurso y a los escenarios.

El criterio implementado en la herramienta para crear una versión, es el de permitir al usuario obtener un documento con toda la información almacenada hasta un momento específico en la tarea de análisis de requerimientos que está efectuando. Esto implica que un proyecto puede almacenar n versiones diferentes V_1, V_2, \dots, V_n , en donde cada nueva versión se puede representar como:

$$V_{i+1} = V_i + \lambda$$

siendo λ los cambios que se fueron produciendo en cada uno de los elementos que forman el proyecto, estos cambios pueden ser por ejemplo, modificaciones en el Universo del Discurso, incorporación, modificación o eliminación de algún símbolo del LEL o Escenario.

Cuando el usuario trabaja en una V_i determinada, TILS solo permite consultar las versiones anteriores, permitiendo de esta manera poder comparar los cambios que se hubieran efectuado de una versión a otra.

5.2.4.1 Crear una versión nueva

TILS genera versiones del proyecto en desarrollo en el momento que el usuario crea necesario. La forma de hacerlo es la siguiente:

1. En el menú Proyecto, seleccionar la opción Versiones.

Nota: El aspecto que el submenú muestra, dependerá de si ya se creó una versión anterior o no. En el último caso el submenú tendrá el aspecto de la figura 17. Si existía una versión previa veremos el submenú con todas sus opciones activadas, figura 18.

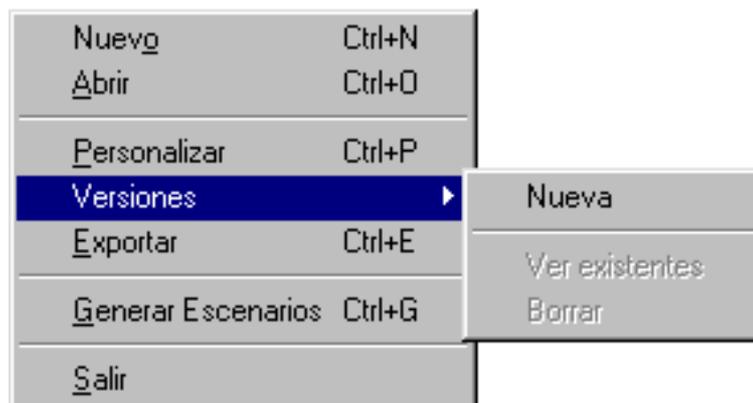


Figura 17: Aspecto del submenú Versiones cuando no existe una versión creada

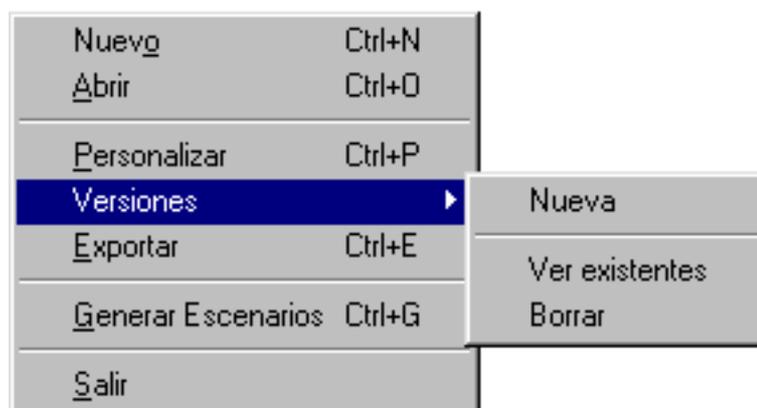


Figura 18: Aspecto del submenú Versiones cuando existe una versión creada

2. Seleccionar la opción Nueva, para activar la ventana de diálogo que muestra la figura 19:

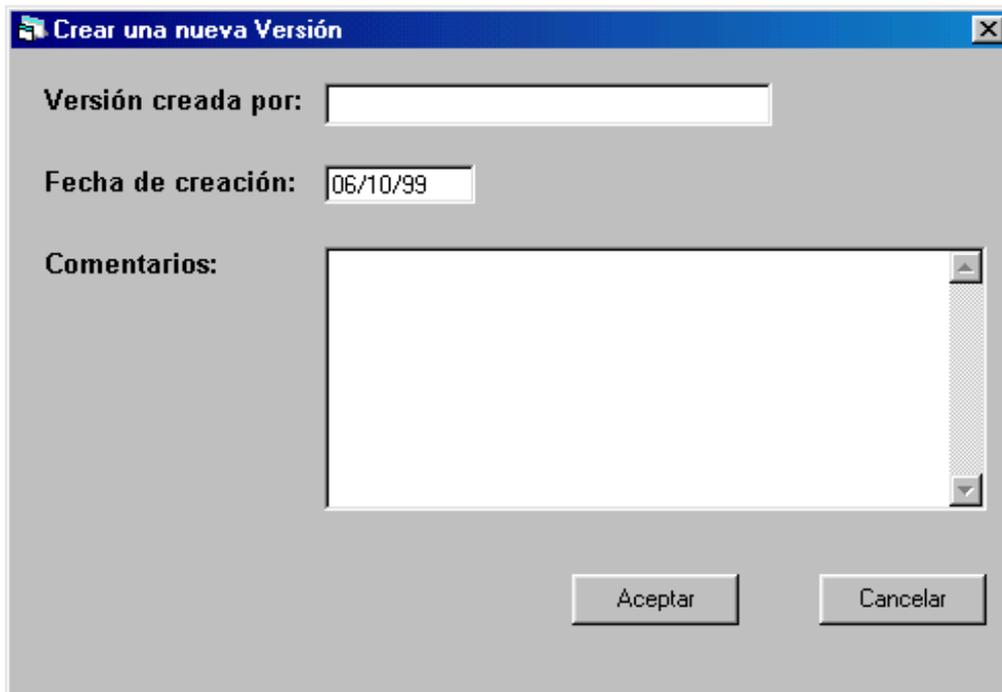


Figura 19: Ventana de dialogo para crear una nueva versión

Ingresar los datos que el usuario crea necesario, excepto fecha de creación, que es un dato obligatorio, tomándose por defecto la fecha actual del sistema.

Al presionar el botón Aceptar, TILS toma unos segundos para crear esta nueva versión, la cual es en realidad un documento HTML que contiene los datos que hasta ese momento almacenaba las bases de datos que maneja el sistema.

La estructura del documento es la siguiente:

Titulo del proyecto.
Información de la versión: Autor, Fecha de creación, Número de versión y Comentarios.
Universo del Discurso.
Lista de símbolos del LEL.
Lista de Escenarios.
Descripción de símbolos del LEL.
Descripción de Escenarios.

Este documento es de sólo lectura, y se puede consultar desde cualquier explorador de la Web, incluso desde el explorador que TILS tiene incorporado, que será descripto más adelante.

5.2.4.2 Consultar versiones existentes.

Para consultar las versiones existentes del proyecto en desarrollo, se debe seleccionar la opción **Ver existentes** (figura 18). Esto provocará la activación de la ventana que muestra la figura 20:



Autor	Fecha de creación	Comentarios
Gustavo Daniel Gil	15/09/99	Prueba 1
Gustavo Daniel Gil	15/10/99	Prueba 2
Gustavo Daniel Gil	15/02/00	Versión 3
Gustavo Daniel Gil	05/03/00	Versión final

Figura 20: Ventana de selección de versiones

En esta ventana es posible seleccionar con el mouse, la versión que se quiere visualizar, luego, se hace doble Click en la misma para activar el explorador de TILS que mostrará el contenido de dicha versión.

5.2.5. Construcción del LEL.

Para comenzar la construcción del LEL, se debe ir al menú Ver, en el que como ya se expuso, posibilita el acceso a las opciones Universo del Discurso, LEL, Escenarios, Barra de Herramientas y Explorador del Web, (Figura 15). Si se selecciona la opción LEL, se despliega un submenú como lo muestra la figura 21:



Figura 21: Opción LEL del Menú Proyectos

Este submenú, permite tener tres vistas diferentes de los símbolos existentes del LEL, a continuación se describen cada una de ellas:

5.2.5.1 Opción Todas las Entradas:

Esta opción, muestra todas las entradas en orden alfabético en el caso que ya se hayan creado algunos símbolos, en caso contrario se mostrará la ventana de trabajo en blanco como lo indica la figura 24.

Esta es la opción más conveniente para comenzar a crear la lista candidata de símbolos y trabajar en la construcción de los mismos.

5.2.5.2 Opción Por Clasificación:

La opción Por Clasificación muestra otro submenú con todas las categorías que se definieron para el proyecto en particular que se está implementando, esto es posible ver en la figura 22:



Figura 22: Opción Por Clasificación

El submenú Por Clasificación se genera dinámicamente en función de las clasificaciones que existan en el momento en que se accede a él.

Cuando se selecciona una clasificación en particular, TILS mostrará todos los símbolos que pertenezcan a dicha clasificación.

Esta consulta de los símbolos es especialmente importante en la etapa de revisión.

5.2.5.3 Opción Por Estado:

La última opción es Por Estado. Funciona de manera similar a la anterior, pero ahora se pueden consultar los símbolos según el estado en el que se encuentran con respecto a su construcción, esto lo muestra la figura 23:



Figura 23: Opción Por Estado

Este submenú es también de generación dinámica, por lo cual siempre se reflejarán en él los estados actuales que se hayan definido para el proyecto.

Este tipo de consulta es importante cuando se interactúa con los clientes, dado que es posible, por ejemplo, seleccionar las entradas que se consideren con dudas y es posible revisarlas conjuntamente con ellos para de esta manera completarla y convertirla en un símbolo completo.

5.2.5.4 Ventana de trabajo para el LEL.

El aspecto de esta ventana corresponde a lo que se vió en Características del LEL, en este informe, como se puede ver en la figura 24:

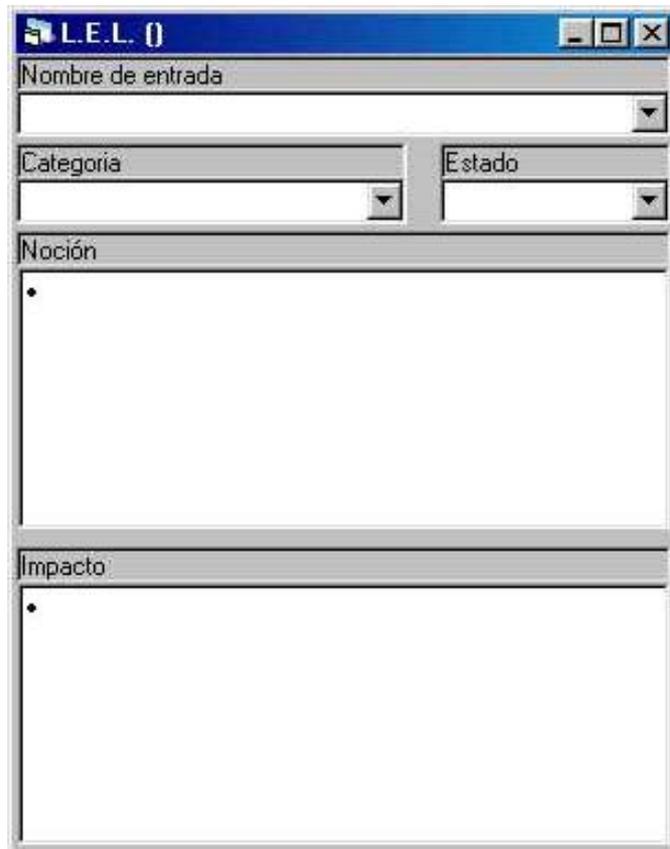


Figura 24: Ventana de trabajo para el LEL

En dicho punto se menciona que la semántica de cada símbolo se representa con una o más nociones y uno o más impactos. El impacto puede no existir. La *noción* indica qué es el símbolo y el *impacto* cómo repercute en el sistema. Por lo tanto, cada símbolo tiene un “nombre” que lo identifica, una “noción” y un “impacto” que lo describen, en TILS además se incorporan dos nuevos elementos, la clasificación del símbolo y el estado del mismo.

La clasificación del símbolo es muy importante para el usuario, en cuanto permite realizar revisiones en forma más estructurada de los símbolos creados. Otro aspecto relevante es que la generación de Escenarios se fundamenta en esta clasificación, como veremos más adelante.

Con esta ventana de trabajo podremos realizar la creación, modificación o eliminación de símbolos del LEL.

5.2.5.5 Creación de un nuevo símbolo

La forma de crear un nuevo símbolo es muy sencilla. Una vez activa la ventana LEL, solamente es necesario escribir el nombre que va a tener el nuevo símbolo en el cuadro Nombre de la Entrada. TILS realiza los controles de integridad en línea, es decir que si se escribe un nombre de símbolo que ya existe, automáticamente lo mostrará, evitando de esta manera que haya nombres duplicados.

Hay que tener en cuenta que la navegación por la ventana LEL se realiza con la tecla TAB o con el mouse, como en cualquier aplicación bajo entorno Windows.

Cuando el cuadro Nombre de Entrada pierde el foco, se produce la incorporación del nuevo símbolo a la base de datos que los almacena. Otro evento que se produce en este momento es el marcado de este nuevo símbolo, en la *noción* e *impacto* de todos los símbolos existentes hasta ese momento. Una vez escrito el nombre del nuevo símbolo, es posible completar los datos restantes asignándole una Clasificación, un Estado, la *Noción* y el *Impacto*.

Cuando se selecciona una clasificación para el símbolo, se puede consultar las características de la *Noción* e *Impacto* correspondiente a dicha clasificación, esto se logra haciendo doble click con el mouse sobre el título del cuadro de texto, como lo indica la figura 25:

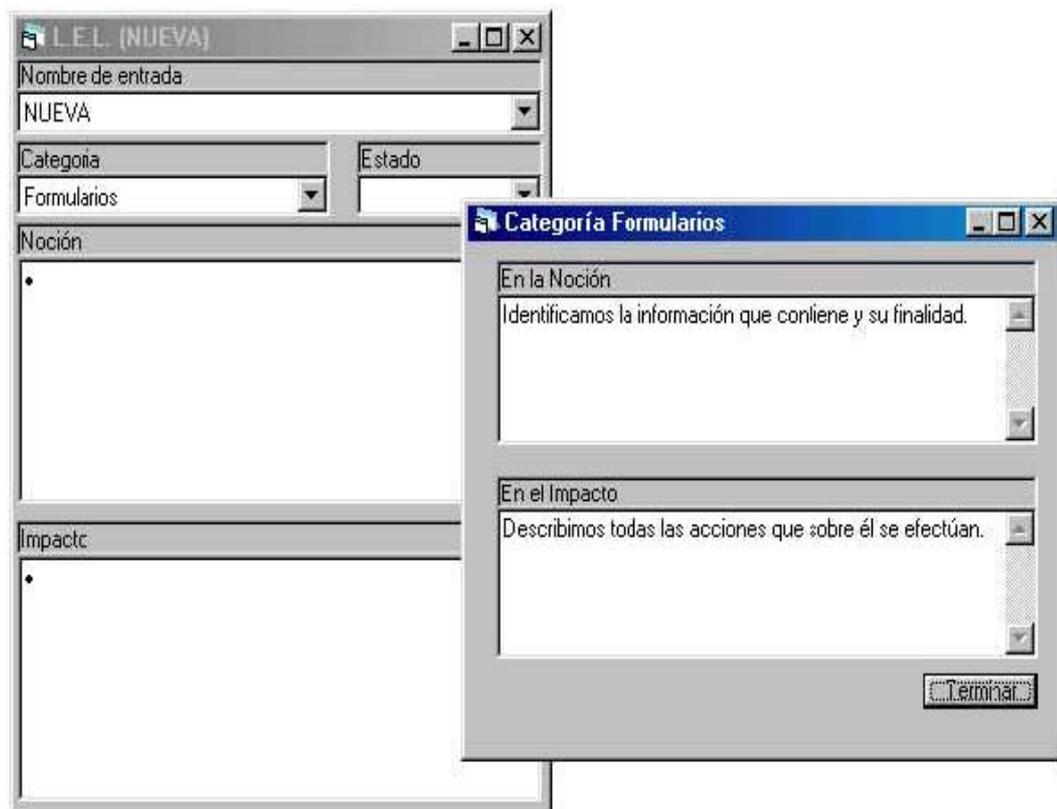


Figura 25: Ventana para agregar Noción e Impacto

Esta ventana tiene el propósito de ser una guía para el usuario, tanto en la descripción de la Noción e Impacto como en la etapa de revisión del mismo, pues se puede comparar si lo que escribió el usuario se encuentra en el contexto de su categoría. Cuando se llena la casilla de estado de un símbolo, se está consignando información del estado de construcción del mismo. Este estado solo le sirve al usuario para tener un control de la entrada que está construyendo, por ejemplo podría seleccionar que dicho símbolo está Incompleto para luego seguir trabajando en él, que está Completo cuando lo haya validado con el cliente, o bien que tiene Dudas por lo que será necesario una revisión más profunda. Este último caso permite hacer las anotaciones correspondientes a esas dudas, haciendo doble click sobre el ícono que aparece en el borde derecho del título, como se puede ver en la figura 26:

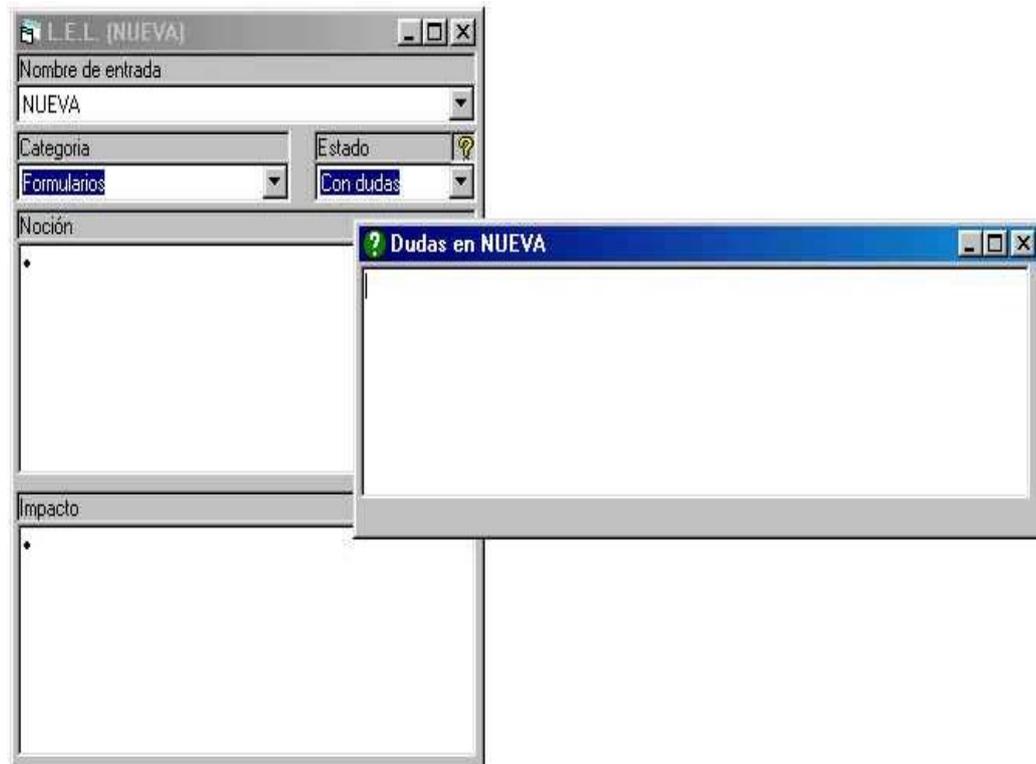


Figura 26: Ventana para ingresar dudas correspondiente a un símbolo.

Esta ventana permite almacenar las dudas referente al símbolo que se está construyendo, el acceso a las anotaciones realizadas en esta ventana se puede lograr en cualquier etapa de construcción del mismo.

Para realizar las anotaciones de la Noción o Impacto del símbolo, solamente es necesario ubicar el cursor en el cuadro correspondiente y escribir la información pertinente. Cuando se escribe alguna palabra o frase que ya haya sido definida como entrada del LEL, TILS la marcará automáticamente de color verde claro y subrayada.

5.2.5.6 Modificación del nombre de un símbolo

En la ventana LEL se puede modificar la clasificación, el estado, la Noción o el Impacto en cualquier momento, dado que éstos están en edición permanente, pero el Nombre de la entrada tiene un tratamiento distinto, puesto que cualquier modificación podría atentar contra la integridad de las referencias que otros símbolos puedan tener con el que se intenta modificar.

Para modificar el nombre de un símbolo existente, es necesario realizar los siguientes pasos:

1. Seleccionar de la lista desplegable de Nombre de Entrada, el símbolo que se desea modificar.

2. Hacer Click con el botón secundario del mouse (botón Derecho), sobre el cuadro de texto, en este momento aparecerá un menú contextual como lo indica la figura 27:

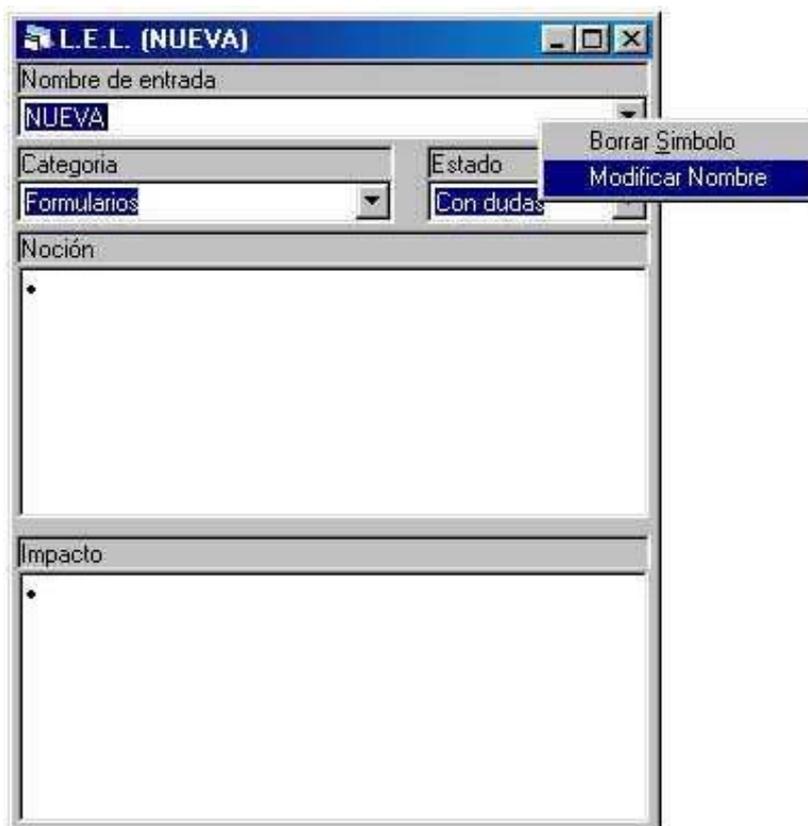


Figura 27: Menú contextual

Se selecciona la opción modificar nombre y se desplegará una ventana de diálogo como el de la figura 28:

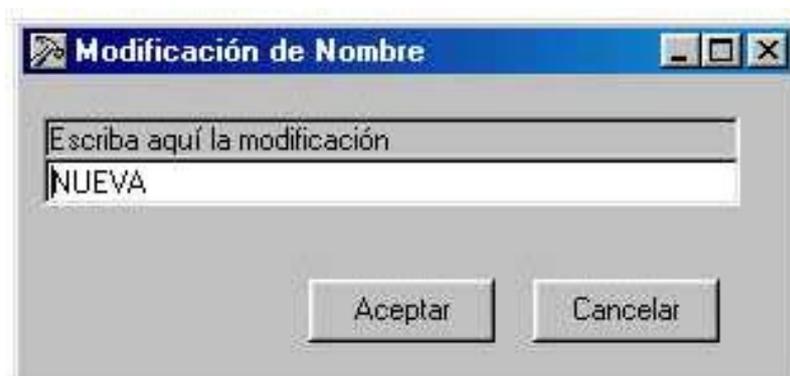


Figura 28: Ventana para modificar nombre

En esta ventana es posible realizar la modificación y luego presionar el botón aceptar.

Cuando se realizan modificaciones de nombre de símbolos, TILS muestra un mensaje en el cual indica que está revisando la integridad de los datos.

5.2.5.7 Eliminar un símbolo.

El proceso de eliminación de símbolos es similar al de modificación, los pasos a seguir son los siguientes:

1. Seleccionar de la lista desplegable de Nombre de Entrada, el símbolo que se pretende modificar.
2. Hacer Click con el botón secundario del mouse (botón Derecho), sobre el cuadro de texto, en este momento aparecerá en menú contextual que se puede ver en la figura 27:
3. Seleccionar la opción Borrar Símbolo, esta acción activa una ventana de diálogo que da la opción de cancelar o aceptar la acción, figura 29.

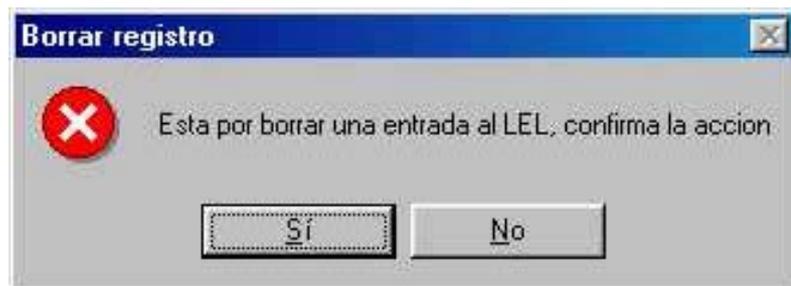


Figura 29: Ventana de diálogo para confirmar la eliminación de un símbolo.

Si se acepta borrar el símbolo, esto será permanente, o sea que no hay forma de recuperarlo. También se produce una validación de los datos y las referencias, como se mencionó en el caso de modificar un nombre.

5.2.5.8 Links entre símbolos.

En párrafos anteriores se hizo referencia a la integridad de los datos. Esta integridad es necesaria por el uso del principio de circularidad, que asevera que en la descripción de la noción y el impacto se debe maximizar el uso de símbolos del LEL. El segundo principio es el de vocabulario mínimo, el cual establece que hay que minimizar el uso de signos externos al domino. Cuando se los usa, se debe asegurar que estos términos pertenezcan al vocabulario natural en uso dentro del Universo del Discurso. Ambos principios tienen como fin que se utilice lo más posible el lenguaje de la aplicación.

Lo dicho anteriormente da origen a uno de los motivos para la creación de TILS. El hacer uso de estos dos principios, provoca que el LEL posea un aspecto de hipertexto. Cada nodo está representado por el símbolo y su descripción. Los

links lo determinan los símbolos que hacen referencia a una entrada ya creada en el LEL para definir su Noción e Impacto.

Para comprender mejor lo expuesto, se analizará el siguiente caso tomado del proyecto Banco de Sangre. En este caso se define el símbolo Plaquetoféresis, que una vez terminado tiene el aspecto que se muestra en la figura 30:

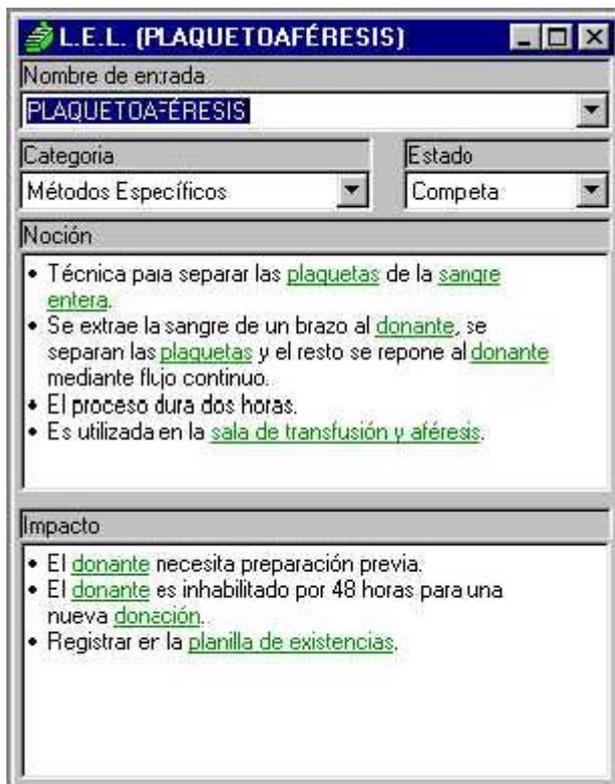


Figura 30: Ejemplo de símbolo del LEL

Se puede ver en este ejemplo, que tanto la noción como el impacto contienen otros símbolos ya definidos previamente:

- ✓ donante,
- ✓ plaqueta,
- ✓ sangre entera,
- ✓ sala de transfusión y aféresis,
- ✓ donación,
- ✓ planilla de existencias,

Para poder consultar estos símbolos a los que se hace referencia, simplemente se accede al menú Edición de TILS y se activa la opción Hipertexto, o bien se presiona, en la barra de herramientas, el ícono correspondiente:



Una vez activada esta opción, hacemos Click sobre el símbolo que se desea consultar, con lo que se activará una nueva ventana en el sector derecho de la pantalla, figura 31:

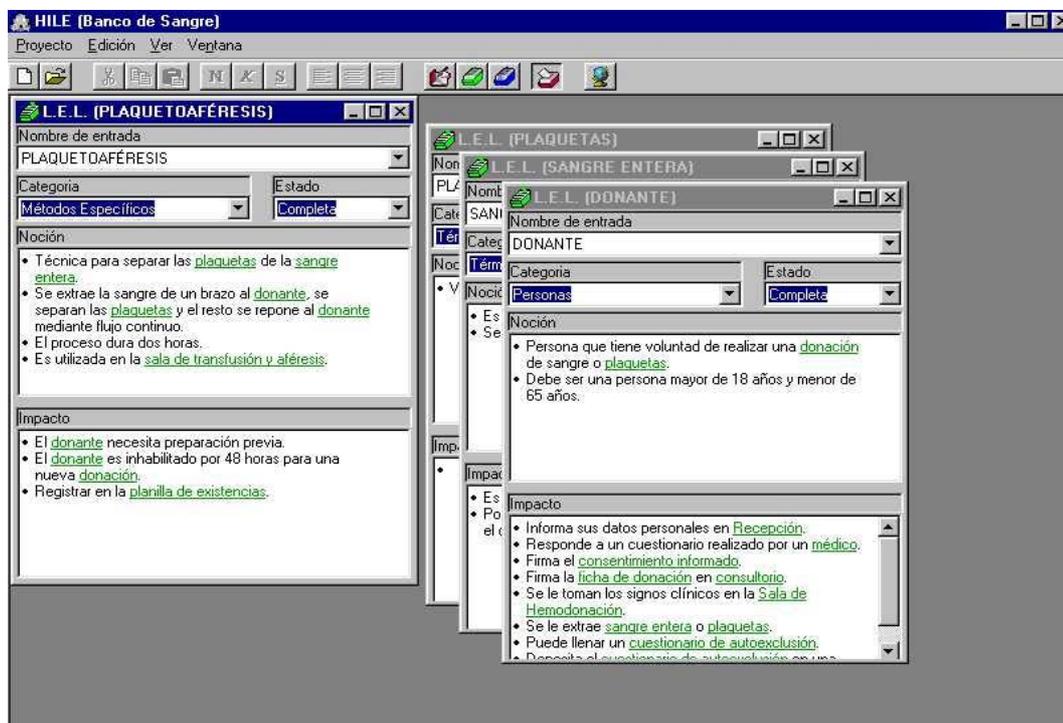


Figura 31: Navegación entre símbolos

5.2.5.9 Links entre el Universo del Discurso y símbolos del LEL.

Como ya se mencionó en el punto 3.2, en el que se trató las características del LEL, éste está compuesto 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 adquieren, por ejemplo, de entrevistas, observaciones, lectura de documentos. Toda esta información se refleja en el Universo del Discurso. A partir de allí se genera una lista con todos los símbolos reconocidos. Durante el proceso de recolección, el ingeniero de software procura entender el significado de cada símbolo describiendo la Noción y el Impacto del mismo. Por esta razón es cómodo poder ver simultáneamente las ventanas de trabajo del LEL y el Universo del Discurso, y que cada nueva entrada que el usuario vaya definiendo se marque como tal en este último. Para comprender mejor esto analicemos el siguiente ejemplo:

El analista tiene el Universo del Discurso completo y se dispone a comenzar a crear el LEL, para eso abre la ventana de trabajo del LEL y el editor, figura 32:

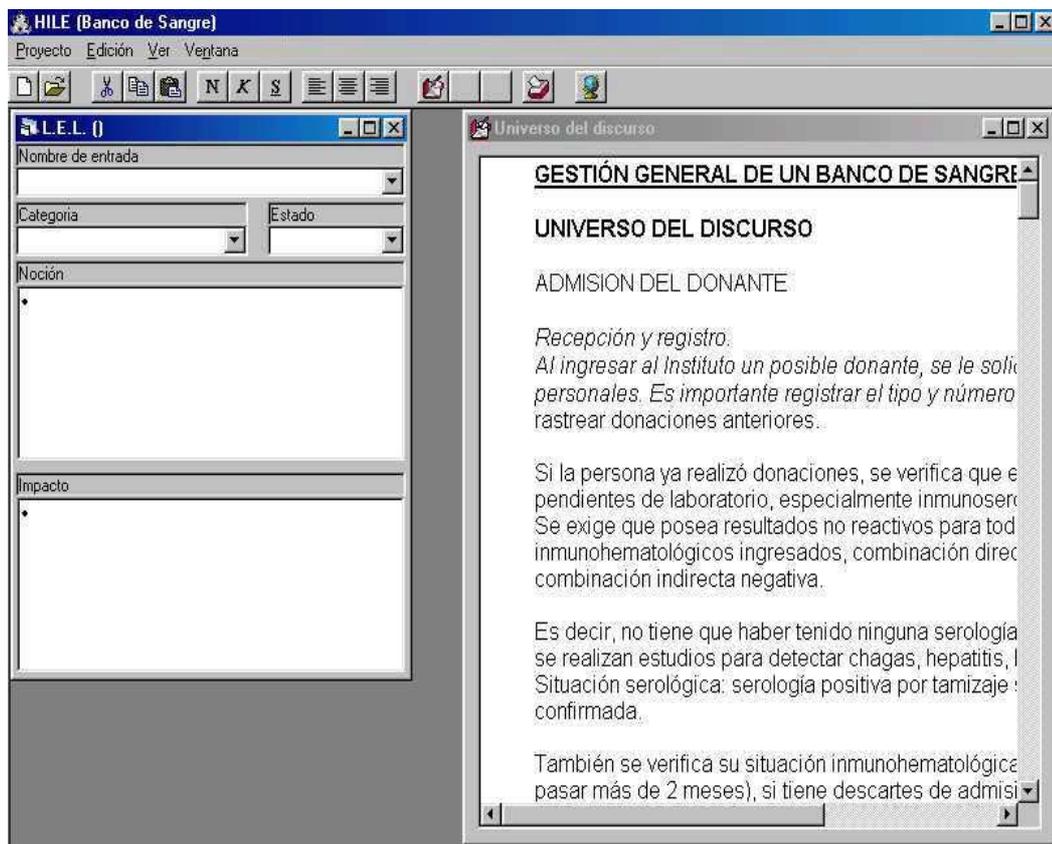


Figura 32: Ventas de LEL y Universo del Discurso.

Luego de analizar el Universo del Discurso decide incorporar Donante como símbolo del LEL, al hacerlo podrá observar que en el editor el nuevo símbolo está marcado, figura 33.

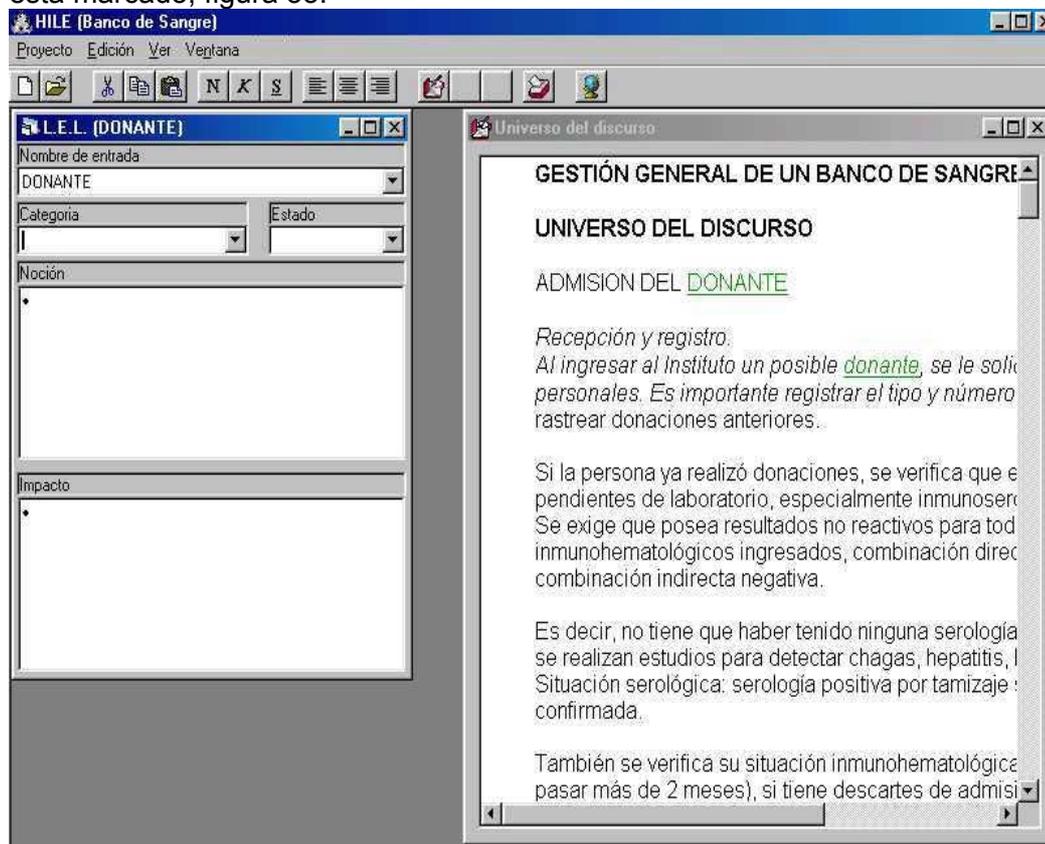


Figura 33: Links de símbolos en el Universo del Discurso.

Esto ocurrirá con todos los símbolos que incorpore al LEL, lográndose así que el Universo del Discurso tenga Links con los símbolos del LEL, lo cual permite consultarlos desde éste.

5.2.6. Construcción de Escenarios.

En el punto 4.4. de este informe, se describió el proceso de construcción de Escenarios, el cual se basa exclusivamente en la información contenida en el LEL, a la que se le aplican una serie de heurísticas. Estas heurísticas están implementadas en TILS de manera tal que el usuario puede acceder al comienzo de la descripción de los escenarios de forma automatizada, logrando con esto un ahorro considerable de tiempo y esfuerzo. Como se dijo anteriormente, el proceso de construcción consta de las siguientes etapas:

8. Identificación de los actores de la aplicación;
9. Generación de la lista de escenarios candidatos a partir de los actores principales;
10. Descripción de los escenarios candidatos provenientes de los actores principales;
11. Ampliación de la lista de escenarios candidatos a partir de los actores secundarios;
12. Descripción de los escenarios candidatos provenientes de actores secundarios;
13. Revisión de los escenarios;
14. Validación de escenarios.

En TILS se hallan implementadas las etapas 1 a 5, siendo luego necesario retornar al Universo de Discurso para concluir la descripción de los mismos. Posteriormente se realizan nuevas entrevistas con los clientes, con el fin realizar las etapas 6 y 7, para poder ampliar la información sobre aquellos escenarios cuya descripción resultara incompleta y/o confusa a partir de la información obtenida del LEL.

Si bien las etapas 6 y 7 no se realizan automáticamente, el usuario puede concluir las en TILS en forma manual, de manera muy similar a la construcción de los símbolos del LEL.

De lo antes expuesto, podemos concluir que TILS separa la construcción de escenarios en dos etapas bien definidas:

1. Generación de la lista de escenarios candidatos y su descripción.
2. Revisión y Validación de escenarios.

5.2.6.1. Generación de la lista de escenarios candidatos y su descripción.

Para realizar la generación de los escenarios candidatos y su descripción, se debe seleccionar el menú Proyecto de TILS, luego presionar la opción Generar Escenarios, como lo indica la figura 34:



Figura 34: Opción Generar Escenarios del Menú Proyectos.

Al seleccionar esta opción se despliega una ventana para que se confirme esta acción y muestre el progreso de la generación, figura 35:

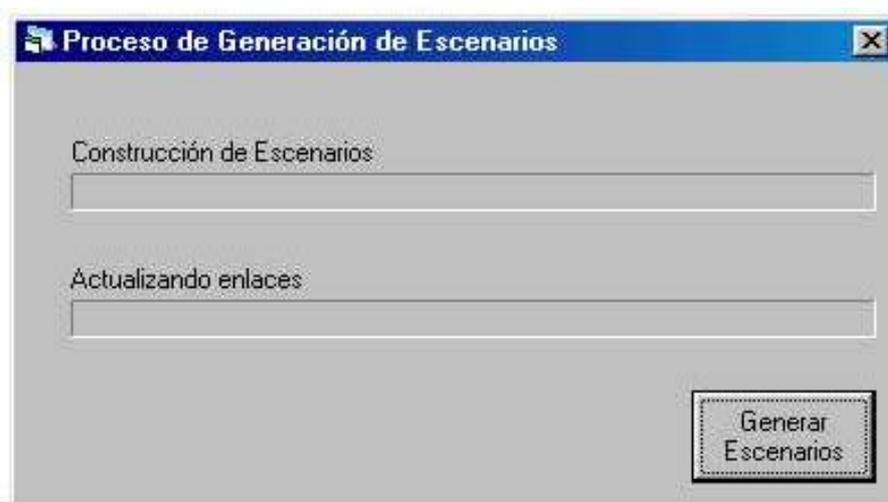


Figura 35: Ventana de generación de Escenarios.

Al presionar el botón **Generar Escenarios** de esta ventana, TILS comienza el proceso de construcción de los escenarios candidatos.

El hecho de crear los escenarios lleva aparejado el uso de símbolos que se encuentran en el LEL. Esto obliga a disparar un segundo proceso encargado de actualizar los nuevos enlaces que se produjeron en la descripción de cada escenario.

Es importante destacar que la generación de los escenarios candidatos debería realizarse una sola vez en la construcción de un proyecto, cuando el LEL se encuentre totalmente terminado. Sin embargo, es posible que sea necesario realizar una nueva generación, ya sea porque incorporamos nuevos símbolos al LEL, o porque realizamos modificaciones importantes en algunos de ellos. En este caso TILS permitirá generar nuevamente los escenarios candidatos, pero antes nos preguntará sobre lo que estamos tratando de hacer, figura 36:

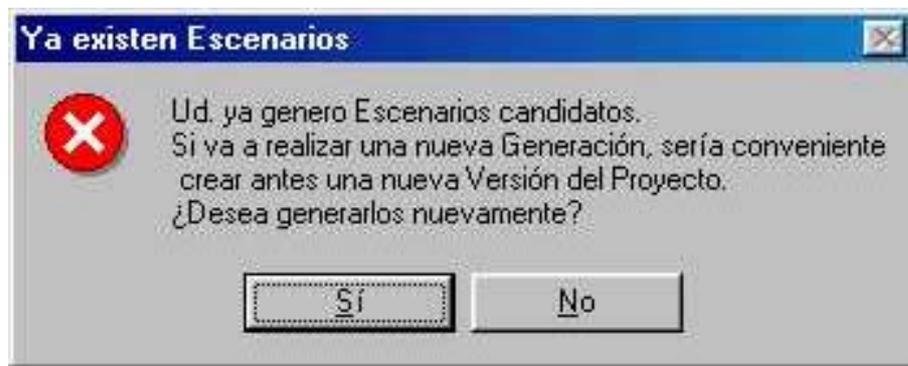


Figura 36: Ventana para confirmar la generación de Escenarios.

Todos los escenarios que existían serán eliminados, quedando solamente los que se crearon en la última generación.

5.2.6.2. Revisión y Validación de escenarios.

Para realizar esta segunda etapa se debe seleccionar la opción Escenarios del menú Ver, figura 37:



Figura 37: Menú Ver.

logrando así acceso a la ventana de trabajo para los escenarios, figura 38:

The image shows a software window titled 'Escenario (ACTUALIZAR EXISTENCI...'. The window contains several input fields and dropdown menus. At the top is a 'Nombre' field with a dropdown arrow. Below it are two dropdown menus: 'Clasificación' and 'Estado'. The main body of the window is divided into seven sections, each with a header and a text area: 'Objetivo', 'Contexto', 'Recursos', 'Actores', and 'Episodios'. The 'Episodios' section is the largest and is currently empty.

Figura 38: Ventana de trabajo para los Escenarios.

5.6.2.3 Ventana de trabajo para los Escenarios.

El aspecto de esta ventana corresponde a lo visto en el punto 4.3. en donde se describió el esquema de descripción de los escenarios. La vista del modelo de escenarios aplicada es una estructura compuesta por el **nombre**, el **objetivo**, el **contexto**, los **recursos**, los **actores** y los **episodios**. El objetivo, el contexto, los recursos y los actores son sentencias declarativas, mientras que los episodios son un conjunto de sentencias con un lenguaje muy simple que hace posible la descripción operativa de comportamientos. Además TILS incorpora dos elementos nuevos al mismo, la clasificación, que puede ser de tipo general o particular y el estado, que cumple la misma función que en el LEL.

Con esta ventana de trabajo se puede realizar la creación, modificación o eliminación de escenarios.

5.6.2.4 Creación de un nuevo escenario.

La forma de crear un nuevo escenario es escribir el nombre que va a tener el nuevo escenario, en el cuadro Nombre. Al igual que con los símbolos del LEL, los controles de integridad se realizan en línea, es decir que si se escribe

un nombre que ya existe, automáticamente mostrará todo su contenido, evitando de esta manera que haya nombres duplicados.

Cuando el cuadro Nombre pierde el foco, se produce la incorporación del nuevo escenario a la base de datos que los almacena. Otro evento que se produce en este momento es el marcado de este nuevo escenario, en la noción e impacto de todos los símbolos del LEL existentes hasta este momento, como así también en todas las componentes existentes de los escenarios ya creados.

Una vez escrito el nombre del nuevo escenario, se pueden completar los datos restantes.

5.2.6.5 Modificación del nombre de un escenario.

Normalmente luego de la generación de los escenarios candidatos, se deseará realizar modificaciones a medida que se efectúe la revisión y validación de los mismos. Estas modificaciones en lo que respecta a la clasificación, el estado, objetivo, contexto, recursos, actores y episodios se pueden realizar en cualquier momento, dado que estos están en edición permanente, pero el nombre del escenario tiene un tratamiento distinto, puesto que cualquier modificación podría atentar contra la integridad de las referencias que otros escenarios puedan tener con el que se intenta modificar.

Para modificar el nombre de un escenario existente, se deben realizar los siguientes pasos:

1. Seleccionar de la lista desplegable de Nombre, el escenario que se pretende modificar.
2. Hacer Click con el botón secundario del mouse (botón Derecho), sobre el cuadro de texto. En este momento aparecerá en menú contextual que se puede ver en la figura 39:



Figura 39: Menú para Modificar, Borrar o ver el Origen de un Escenario.

Seleccionando la opción modificar nombre, se desplegará una ventana de diálogo como el de la figura 40:

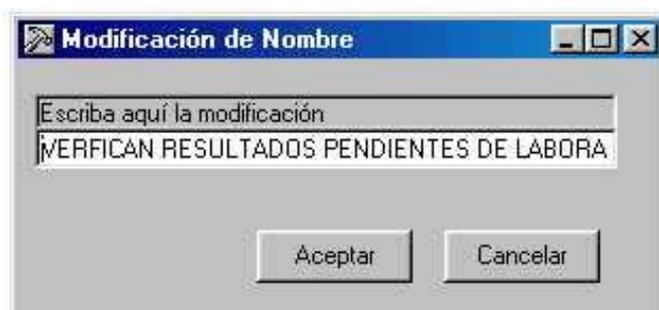


Figura 40: Ventana para modificar el nombre de un Escenario

En esta ventana se puede realizar la modificación y luego presionar el botón aceptar.

Cuando se llevan a cabo modificaciones de nombre de escenarios, TILS mostrará un mensaje en el cual nos indica que está revisando la integridad de los datos.

5.2.6.6 Eliminar un escenario.

El proceso de eliminación de escenarios es similar al de modificación. Los pasos a seguir son los siguientes:

1. Seleccionar de la lista desplegable de Nombre, el escenario que queremos modificar.
2. Hacer Click con el botón secundario del mouse (botón Derecho), sobre el cuadro de texto. En este momento aparecerá en menú contextual que se puede ver en la figura 39.

Seleccionar la opción Borrar Escenario, esta acción activa una ventana de diálogo que brinda la posibilidad de cancelar o aceptar la acción, figura 41.

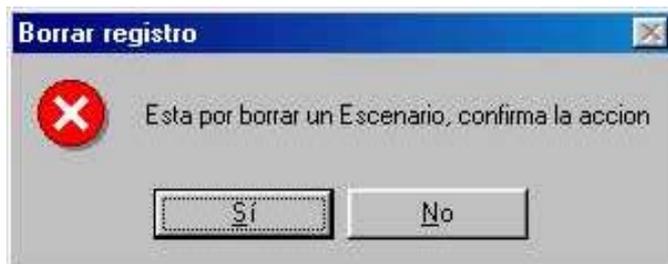


Figura 41: Ventana para confirmar la eliminación de un Escenario

Si se acepta borrar el escenario, esto será permanente, o sea que no hay forma de recuperarlo. También se produce una validación de los datos y las referencias, como se mencionó en el caso de modificar un nombre.

5.2.6.7 Origen de un escenario.

La tercera opción del menú contextual (figura 39), permite ver cual es el símbolo del LEL que dió origen al escenario, figura 42.

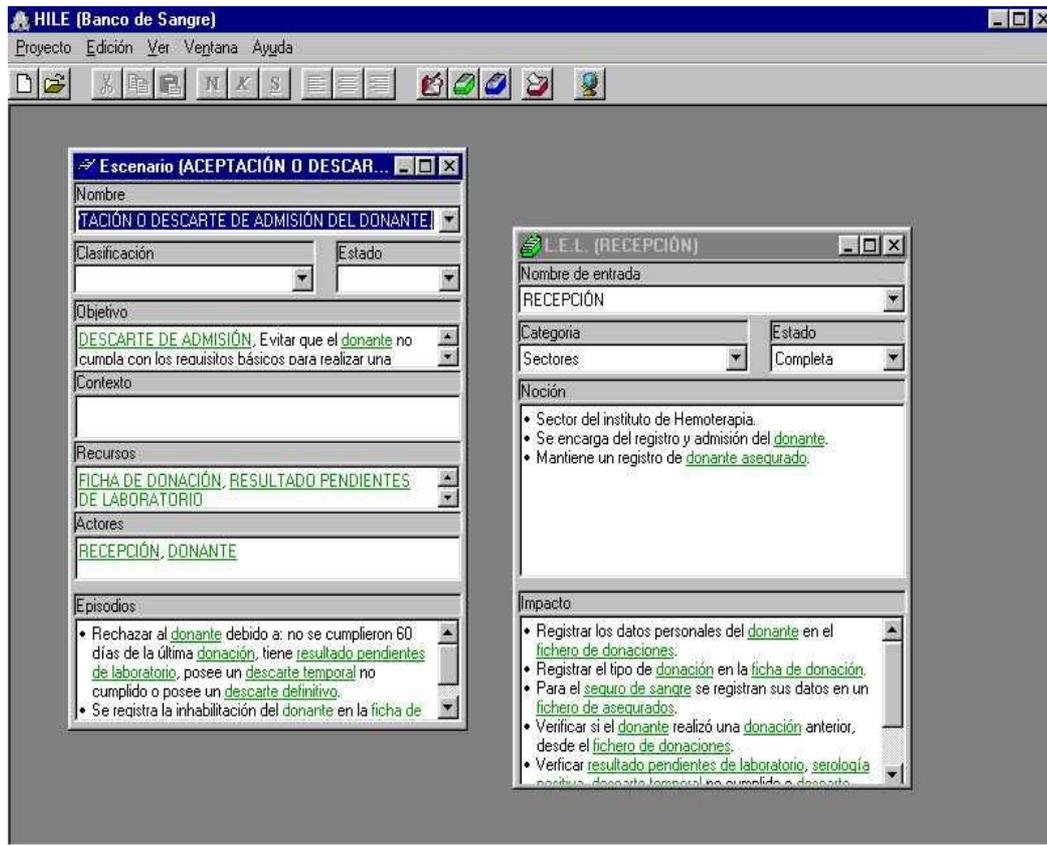


Figura 42: Origen de un Escenario.

Como se explicó en las heurísticas, para obtener un escenario cuando se realiza la generación de los mismos, siempre tendrá como origen un símbolo del LEL, el cual lo es posible ver con esta opción del menú contextual.

5.2.6.8 Links entre escenarios y símbolos.

En la ventana de trabajo de los escenarios, también se producen links a los símbolos del LEL existentes. Esto permite realizar consultas a los símbolos para poder completar un escenario en construcción o simplemente consultar los símbolos involucrados para comprender mejor el contenido del escenario. Si se utiliza algún escenario para describir otro, se verá que éstos se marcan en azul y subrayados, para identificarlos fácilmente.

En resumen se puede ver que un escenario tiene links a símbolos del LEL y a otros escenarios produciendo nuevamente un aspecto hipertextual. Para comprender mejor lo expuesto, se analizará el siguiente ejemplo, figura 43:

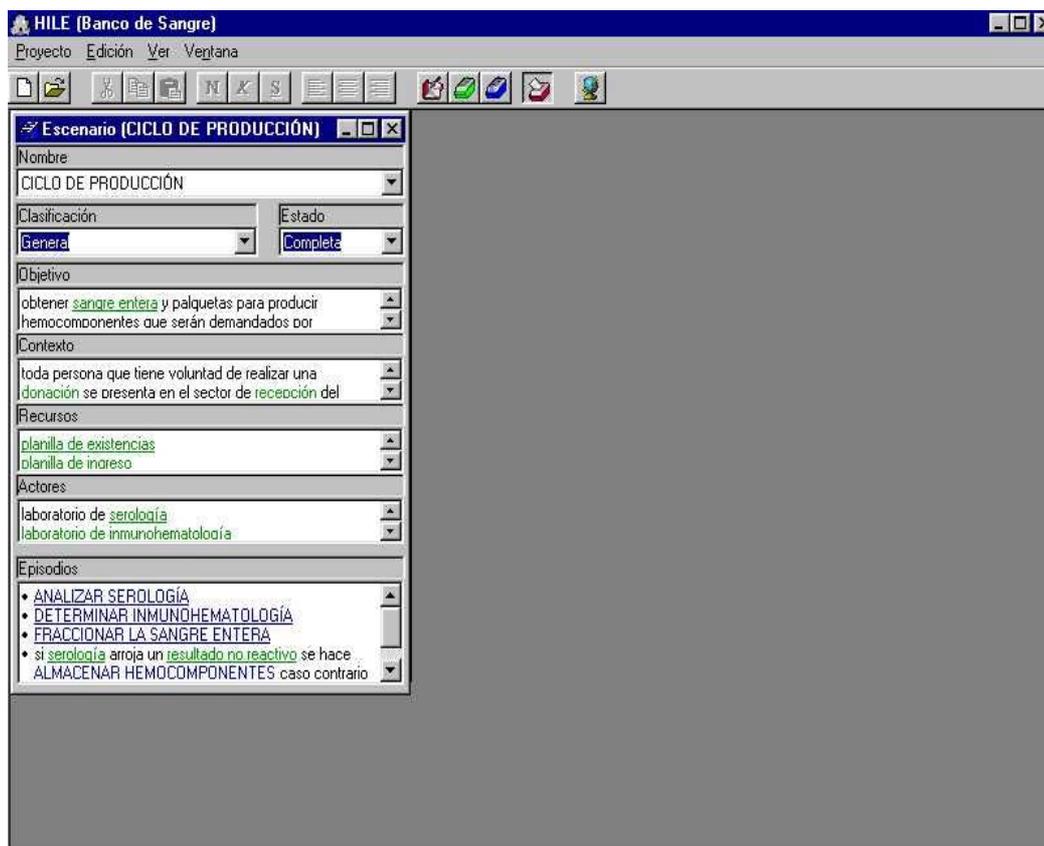


Figura 43: Ejemplo de Links a Símbolos y a Escenarios

En este ejemplo se puede ver que en los cuadros de objetivo, contexto, recursos y actores se hace referencia a símbolos del LEL ya definidos previamente, y en los episodios se encuentran referencias a escenarios. Para poder consultar estos símbolos y escenarios a los que se hace referencia, simplemente se accede al menú Edición de TILS y se activa la opción Hipertexto, o bien se presiona en la barra de herramientas, el ícono correspondiente:



Una vez activada esta opción, se hace Click sobre el símbolo o escenario que se quiere consultar, con lo que se activará una nueva ventana en el sector derecho de la pantalla, figura 44:

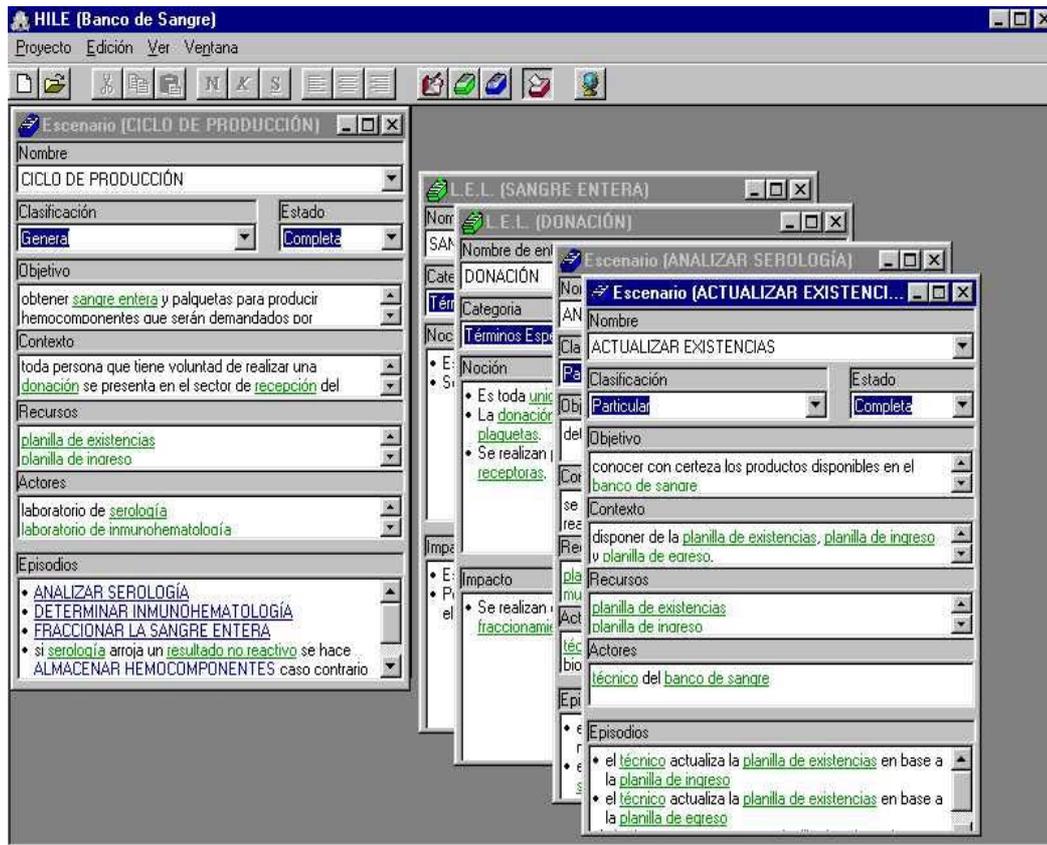


Figura 44: Navegación entre Links.

5.2.7. Exportar Documentos.

La exportación de documentos es otra utilidad incorporada a TILS, con la cual el usuario podrá documentar los símbolos del LEL y los escenarios en tres formatos distintos según crea conveniente. Para acceder a esta utilidad, se selecciona la opción Exportar del menú Proyectos, figura 45:



Figura 45: Opción exportar documentos

Una vez hecha la selección, se verá la ventana de diálogo de la opción exportar, figura 46:

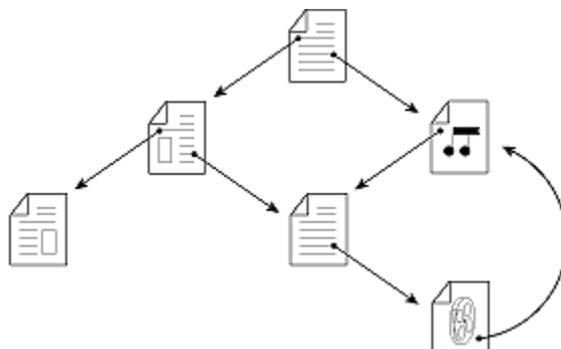


Figura 46: Ventana de diálogo de la opción exportar

En esta ventana el usuario podrá exportar el proyecto a un formato de sólo texto, RTF o bien HTML.

5.2.8. El Navegador.

Los navegadores son los programas que permiten visualizar la World Wide Web (el Web o WWW) la cual proporciona una interfaz gráfica por la que es fácil desplazarse para buscar documentos en Internet o en una Intranet. Estos documentos, así como los vínculos entre ellos, componen una red de información.



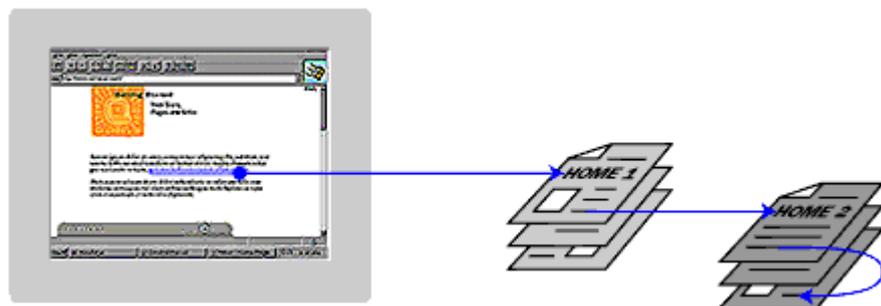
El Web permite saltar mediante un "hipervínculo" de una página a otra. Imagine que el Web es una gran biblioteca. Los sitios Web son los libros y las "páginas" de Web páginas concretas de los libros. Las páginas pueden contener noticias, imágenes, películas, sonidos, gráficos en tres dimensiones, casi cualquier cosa. Estas páginas pueden estar situadas en sistemas en cualquier lugar del mundo. Al conectarse al Web, tiene igual acceso a información en cualquier lugar del mundo.

El Web está cambiando la forma en que las personas se comunican en todo el mundo. Este nuevo medio global está siendo aceptado más rápidamente que ningún otro medio de comunicación en la historia. En los dos últimos años, ha crecido hasta incluir una vasta gama de información: desde cotizaciones bursátiles hasta ofertas de trabajo, boletines de noticias, preestrenos de películas, revistas literarias y juegos. La gama de información oscila desde los temas más desconocidos, hasta los de importancia mundial. La gente suele hablar de "explorar" el Web y visitar nuevos sitios. "Explorar" significa seguir los hipervínculos entre páginas y temas sobre los que es posible que nunca se haya oído hablar.

Una "página principal" es el punto de inicio de un sitio Web. Sería el equivalente a la cubierta o el índice de un libro.

Cada página de Web, incluida la página principal de un sitio Web, tiene una dirección exclusiva llamada URL (Universal Resource Locator).

Las páginas están interconectadas en el Web. La conexión a otras páginas se realiza haciendo click sobre texto o gráficos llamados hipervínculos.



5.2.8.1 Los hipervínculos

Son palabras y gráficos subrayados o con un recuadro, que tienen direcciones de Web (también llamadas direcciones URL -- Universal Resource Locator) incrustadas. Al hacer click en un hipervínculo, se salta a una página concreta de un determinado sitio Web. Los hipervínculos pueden diferenciarse fácilmente, ya que aparecen en un color distinto al resto del texto del sitio Web.

5.2.8.2 Direcciones URL

Las páginas Web que encontramos en Internet se designan mediante lo que se llama la dirección URL (Universal Resource Locator: Localizador Universal de Recursos). Cuando se pretende cargar de la red una página, se escribe su dirección URL en la celda dispuesta para este propósito en el navegador. Así, por ejemplo, si se quiere ver la página de la Universidad Nacional de Salta, se escribe la dirección <http://www.unsa.edu.ar>.

Una URL tiene tres partes:

- ◆ **Siglas del Protocolo:** indican qué protocolo se va a utilizar para la transmisión de datos. Lo normal es usar el protocolo de hipertexto, o sea, páginas Web, que es el HTTP (HyperText Transfer Protocol: Protocolo de Transferencia de Hipertexto). Normalmente, se pone después del protocolo los siguientes caracteres: “://”, por lo que las siglas del protocolo de páginas Web serán: “http://”.
- ◆ **Nombre de Dominio del servidor:** indica qué servidor de Internet va a brindar la información solicitada. Normalmente, los servidores de páginas Web tienen por nombre de dominio algo parecido a lo siguiente: *www.nombre.ar*. Las tres letras “w” del principio indican que la computadora está destinada a servir páginas Web en Internet.
- ◆ **Ruta en el servidor:** indica la posición del fichero que concretamente se ha solicitado dentro del servidor. Los ficheros están en el servidor ordenados jerárquicamente en una estructura de directorios o carpetas, como si fuera un disco duro. Una carpeta se separa de su sucesora mediante una barra de dividir. Así, por ejemplo, la siguiente ruta: */personal/nombre/imagen*, significa que la información que solicitamos se encuentra en la carpeta principal llamada *personal*, y dentro de ésta en la carpeta *nombre*, y por último en la carpeta *imagen* que está, a su vez, dentro de la anterior. La ruta no sólo se compone de la posición de los datos en el sistema de ficheros, como se vio recientemente, sino que también hay que poner el nombre del fichero donde se encuentra la información que se pretende visualizar. Estos ficheros suelen tener por extensión las siglas *.htm* o bien *.html* (HTML: HyperText Mark-up Language: Lenguaje de Hipertexto a base de Marcas: Lenguaje de descripción de páginas Web). De este modo, una ruta completa podría ser: */personal/nombre/imagen/galeria.htm*.

Juntando estos tres elementos, formamos una URL completa. Un ejemplo, sería:

“http://unsa.edu.ar/personal/nombre/imagen/galeria.htm”

5.2.8.3 ¿Cómo Navegar?

Para empezar, se debe seleccionar la opción Explorador del Web del menú Ver, figura 47, o bien presionar el ícono correspondiente en la barra de

herramienta, figura 48. TILS introduce una dirección de partida en el navegador, en este caso es la de la Universidad Nacional de Salta.



Figura 47: Opción Explorar.



Figura 48: Acceso directo de la barra de herramientas.

De esta manera si la computadora del usuario está conectada a Internet, se podrá ver al explorador de TILS como lo muestra la figura 49:

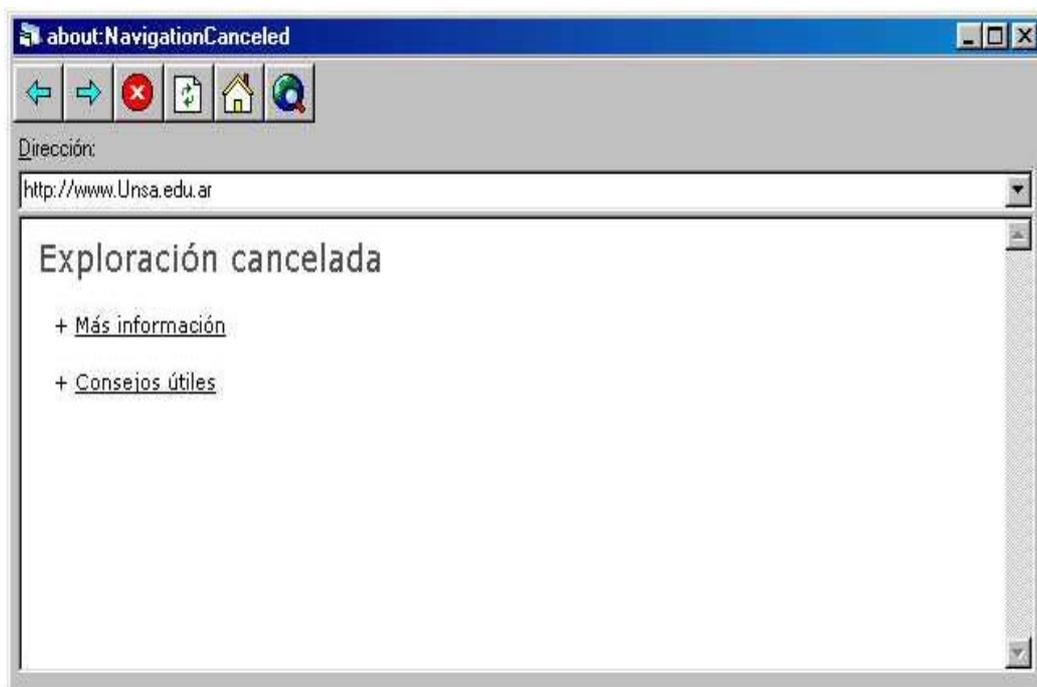


Figura 49: Explorador de TILS.

Nota: en caso de que el usuario no este conectado a Internet, TILS avisará que no puede establecer la conexión, quedando a cargo del usuario hacerla manualmente.

La barra de herramientas proporciona una serie de funciones y comandos detallados para la gestión del explorador. La barra de direcciones, situada debajo de la barra de herramientas, muestra la dirección del sitio Web al que se ha tenido acceso.

Para ir a otro sitio, se escribe su dirección URL directamente en el espacio blanco de esta barra. Cuando haya terminado se presiona *Entrar* en el teclado. También se puede realizar esta tarea con el botón correspondiente a buscar de la barra de herramientas, figura 50:



Figura 50: Botón buscar

si se utiliza este botón, primero se verá una ventana de dialogo para buscar el documento que se pretende abrir. Este es el procedimiento más común para ver las versiones de un proyecto.

Botones Atrás y Adelante

El botón Atrás le devuelve a una página ya vista, normalmente la más reciente, figura 51



Figura 51: Botón atrás.

El botón Adelante le vuelve a llevar a una página ya visitada. Si no ha utilizado el botón Atrás en la sesión actual, el botón Adelante permanece inactivo, figura 52.



Figura 52: Botón adelante.

Botón Detener

Este botón detiene inmediatamente el acceso del explorador a un vínculo, figura 53.



Figura 53: Botón detener.

Botón Actualizar página

Este botón actualiza la página activa, figura 54:



Figura 54: Botón actualizar.

Botón Abrir página de inicio

Este botón devuelve a la página de inicio predeterminada. La página predeterminada es la que aparece al abrir el explorador.



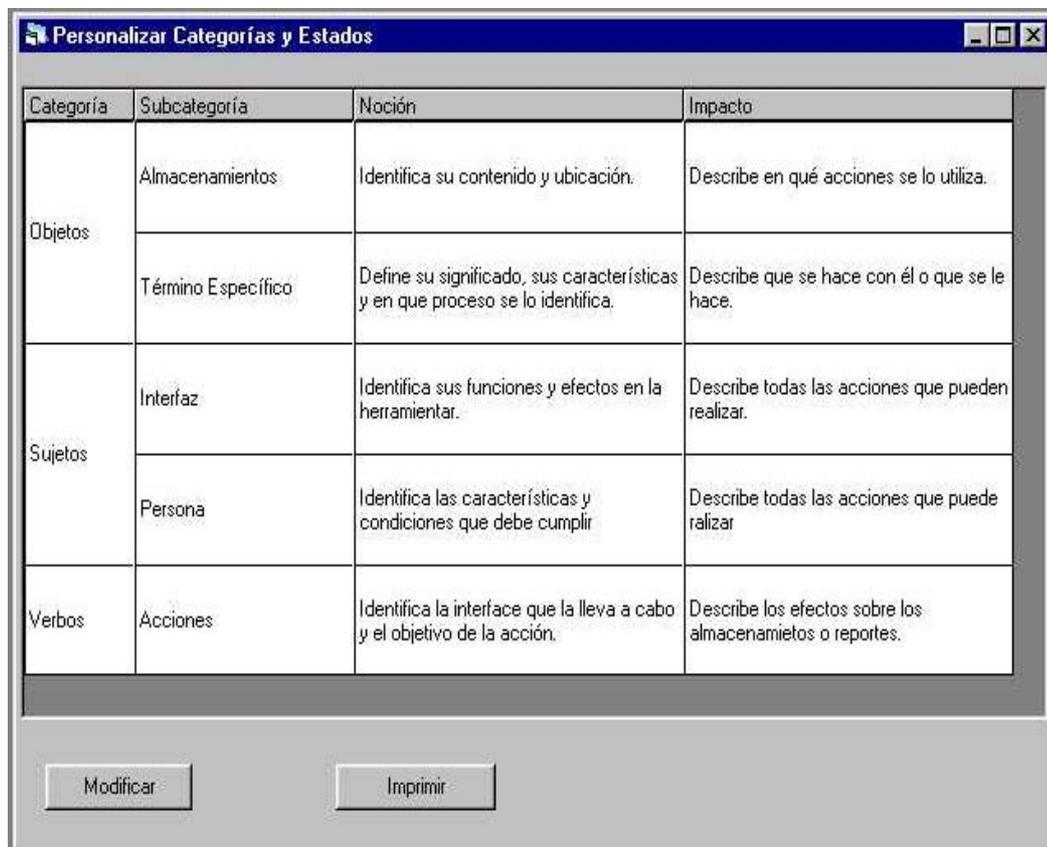
Figura 55: Botón inicio.

5.3 Arquitectura de TILS

5.3.1 Introducción

Para definir la arquitectura de la herramienta se puede utilizar la metodología que está consignada en el presente informe, es decir un Universo del Discurso, un LEL y los Escenarios. Para apoyar la construcción de este sistema se utilizará a TILS para administrar el proyecto y generar un informe.

Previamente definimos las categorías y subcategorías para los símbolos del LEL, como se puede ver en la figura 56, quedando de esta manera formado el template que se utiliza en la construcción de los símbolos.



Categoría	Subcategoría	Noción	Impacto
Objetos	Almacenamientos	Identifica su contenido y ubicación.	Describe en qué acciones se lo utiliza.
	Término Específico	Define su significado, sus características y en que proceso se lo identifica.	Describe que se hace con él o que se le hace.
Sujetos	Interfaz	Identifica sus funciones y efectos en la herramienta.	Describe todas las acciones que pueden realizar.
	Persona	Identifica las características y condiciones que debe cumplir	Describe todas las acciones que puede realizar
Verbos	Acciones	Identifica la interface que la lleva a cabo y el objetivo de la acción.	Describe los efectos sobre los almacenamientos o reportes.

Modificar Imprimir

Figura 56: Ventana de categorías y subcategorías de TILS.

Con estas categorías para los símbolos del LEL, y luego de definir los símbolos y generar los escenarios, se llega al informe que se detalla a continuación.

Proyecto: Arquitectura de la Herramienta

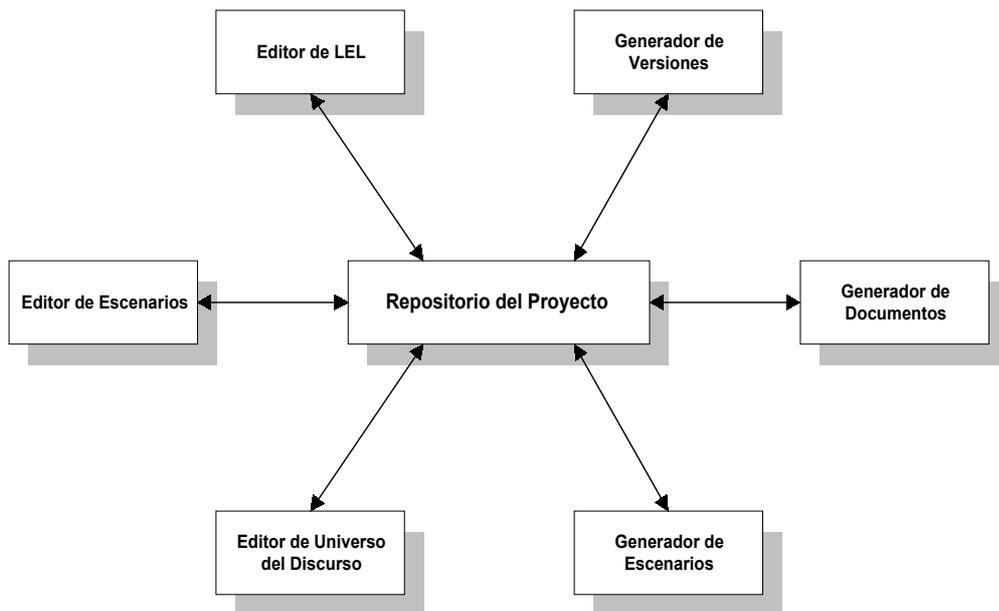
Autor: Gustavo Gil

Fecha de Creación: 15/02/2000

Comentarios: Versión final

5.3.2 Universo del Discurso

El objetivo principal de la herramienta es el de automatizar y simplificar el uso del modelo presentado en los primeros cuatro puntos de este informe. Para lograr este objetivo, la herramienta es capaz de administrar los distintos proyectos que el usuario esté desarrollando. Se puede esquematizar a TILS de la siguiente manera:



En este esquema básico, se puede ver que existen los siguientes subsistemas:

- Repositorio del Proyecto: es el encargado de almacenar toda la información relacionada con cada proyecto que el usuario esté desarrollando. Los componentes de este repositorio son: una Base de Datos formada por una tabla de LEL, una tabla de escenarios, una tabla de versiones, una tabla de estado, una tabla de categoría y la tabla de clasificación. Otro componente de

este repositorio es un documento con formato RTF, que contiene el [Universo del Discurso](#) obtenido por el usuario de la herramienta.

- [Editor de LEL](#): administra la interacción del [usuario](#) con los [símbolos](#) que se incorporarán al [proyecto](#). Las principales funciones de este editor son:
 - ✓ Crear un [símbolo](#) nuevo, realizando las validaciones correspondientes para no permitir la duplicación de nombres.
 - ✓ Editar y modificar los [símbolos](#) existentes.
 - ✓ Eliminar [símbolos](#) a petición del [usuario](#).
 - ✓ Crear [enlaces](#) automáticamente a [símbolos](#) existentes.
 - ✓ Permitir la navegación entre [símbolos](#).
 - ✓ En la navegación, reconocer los contextos navegacionales.
 - [Editor de Escenarios](#): administrar los [escenarios](#) creados por el [usuario](#) o generados por la herramienta. Las principales funciones de este editor son las mismas que las del [editor de LEL](#), pero referidas a los [escenarios](#). Una función adicional con respecto al editor del LEL, es que este editor es capaz de mostrar cuál es el [símbolo](#) que dió origen a un [escenario](#) generado.
 - [Editor de Universo del Discurso](#): administra la interacción del [usuario](#) con el documento que representa el universo del discurso. Este es un editor especializado que además de realizar las funciones mínimas de un editor de texto, como cortar, copiar, pegar, cambiar atributos de tipo de letra, etc., también genera automáticamente [enlaces](#) a [símbolos](#) existentes en el LEL, de manera que el [usuario](#) pueda navegar entre los mismos.
 - [Generador de Escenario](#): es el encargado de generar [escenarios](#) candidatos cuando el [usuario](#) lo solicita. Para esto se basa en las heurísticas vistas en el punto 4 de este informe, y obtiene la información necesaria consultando el [repositorio del proyecto](#).
 - [Generador de Documentos](#): tiene la responsabilidad de exportar información contenida en la [tabla de LEL](#) y la [tabla de escenarios](#), a documentos con formatos a elección del usuario para permitirle a éste poder interactuar de forma cómoda con el cliente.
 - [Generador de Versiones](#): exporta la información de todo el [repositorio del proyecto](#) en una etapa a determinar por el [usuario](#), a formato html para que pueda ser recorrida por medio de navegadores de internet ordinarios o el navegador propio de la herramienta.
-

5.3.3. Lista de Símbolos

- EDITOR DE ESCENARIOS
- EDITOR DE LEL
- EDITOR DE UNIVERSO DEL DISCURSO
- ENLACE
- ESCENARIO
- EXPORTAR INFORMACIÓN
- EXPORTAR VERSIÓN
- GENERADOR DE DOCUMENTOS
- GENERADOR DE ESCENARIOS
- GENERADOR DE VERSIONES
- GENERAR ESCENARIOS CANDIDATOS
- NAVEGACIÓN ENTRE SÍMBOLOS
- PROYECTO
- REGISTRAR DUDAS
- REPOSITORIO DEL PROYECTO
- SÍMBOLO
- TABLA DE CATEGORÍA
- TABLA DE CLASIFICACIÓN
- TABLA DE ESCENARIOS
- TABLA DE ESTADO
- TABLA DE LEL
- TABLA DE VERSIONES
- UNIVERSO DEL DISCURSO
- USUARIO

5.3.4. Descripción de Símbolos

Símbolo: EDITOR DE ESCENARIOS

Categoría: Interfaz

Noción

- Subsistema que permite realizar el mantenimiento de los [escenarios](#) en el [repositorio del proyecto](#).

Impacto

- Crear un [escenario](#).
- Modificar un [escenario](#) existente.
- Borrar [escenario](#).
- Crear [enlaces](#).
- Permitir la [navegación entre enlaces](#).
- [Registrar dudas](#).
- Mostrar el [símbolo](#) que originó el [escenario](#).

Símbolo: EDITOR DE LEL

Categoría: Interfaz

Noción

- Subsistema que permite realizar el mantenimiento de [símbolos](#) en el [repositorio del proyecto](#).

Impacto

- Crear un [símbolo](#).
- Modificar [símbolo](#).
- Borrar [símbolo](#).
- Crear [enlaces](#).
- Permitir la [navegación entre enlaces](#).
- [Registrar dudas](#).

Símbolo: EDITOR DE UNIVERSO DEL DISCURSO

Categoría: Interfaz

Noción

- Subsistema que permite realizar el mantenimiento del [Universo del discurso](#) en el [repositorio del proyecto](#).

Impacto

- Modificar [Universo del Discurso](#).
- Crear [enlaces](#).
- Permitir la [navegación entre enlaces](#).

Símbolo: ENLACE

Categoría: Acciones

Noción

- Referencia a otro [símbolo](#) o [escenario](#).
- Se origina en el [editor de LEL](#), [editor de escenarios](#) y [editor de universo del discurso](#).

Impacto

- Permitir al [usuario](#) la navegación hipertextual del [proyecto](#).

Símbolo: ESCENARIO

Categoría: Término Específico

Noción

- Descripción parcial del funcionamiento del sistema, focaliza un momento específico de la aplicación.
- Cada [escenario](#) está compuesto por un Nombre: identifica al [escenario](#). Objetivo: establece la finalidad del [escenario](#). Contexto: describe las acciones previas para iniciar el [escenario](#), las precondiciones, la ubicación física y temporal. Recursos: identifica los objetos pasivos con los cuales los actores trabajan. Actores: detalla las entidades que se involucran activamente en el [escenario](#). Set de Episodios: cada episodio representa una acción realizada por un actor, donde participan otros actores y se utilizan

recursos. Dudas: describe los puntos pendientes a clarificar con el [usuario](#).

Impacto

- Brindar una visión más amplia del comportamiento del macro sistema.
- Proveer un medio de comunicación con el [usuario](#).
- Contar con un instrumento de traceability.

Símbolo: EXPORTAR INFORMACIÓN

Categoría: Acciones

Noción

- Exportar la información que se encuentra en el [repositorio del proyecto](#), a formato TXT, RTF o HTML, según la solicitud del [usuario](#).
- Se realiza en el [generador de documentos](#).

Impacto

- Recorrer las tablas del [repositorio del proyecto](#) recopilando la información necesaria.

Símbolo: EXPORTAR VERSIÓN

Categoría: Acciones

Noción

- Representa un momento determinado de la evolución del [proyecto](#) en desarrollo.
- Se realiza en el [generador de versiones](#).

Impacto

- Obtener del [repositorio del proyecto](#) toda la información almacenada hasta ese momento.

Símbolo: GENERADOR DE DOCUMENTOS

Categoría: Interfaz

Noción

- Responsable de exportar información contenida en la [tabla de LEL](#) y la [tabla de escenarios](#), a documentos con formatos a elección del usuario para permitirle a éste poder interactuar de forma cómoda con el cliente.

Impacto

- Realiza [exportar información](#).

Símbolo: GENERADOR DE ESCENARIOS

Categoría: Interfaz

Noción

- Administra el proceso de generación de [escenario](#).

Impacto

- [Generar escenarios candidatos](#).

Símbolo: GENERADOR DE VERSIONES

Categoría: Interfaz

Noción

- Obtiene información específica para crear una nueva versión del [proyecto](#).

Impacto

- Realiza [exportar versión](#).

Símbolo: GENERAR ESCENARIOS CANDIDATOS

Categoría: Acciones

Noción

- Genera una lista preliminar de [escenario](#).
- Se realiza en el [generador de escenarios](#).

Impacto

- Construir [escenarios](#) candidatos en base a la información obtenida de la [tabla de LEL](#).
- Registrar el [símbolo](#) que originó al [escenario](#).

Símbolo: NAVEGACIÓN ENTRE SÍMBOLOS

Categoría: Acciones

Noción

- Mostrar el contenido del [símbolo](#) o [escenario](#) seleccionado por el [usuario](#).
- Lo realiza el [editor de LEL](#), el [editor de escenarios](#) y el [editor de universo del discurso](#).

Impacto

- Crear una nueva instancia del [editor de LEL](#) o el [editor de escenarios](#), que muestra el contenido del [símbolo](#) o [escenario](#) seleccionado por el [usuario](#).

Símbolo: PROYECTO

Categoría: Término Específico

Noción

- Una instancia particular de LEL, escenarios y versiones, que un [usuario](#) desarrolla para un sistema dado.

Impacto

- Servir como guía para agilizar el proceso de desarrollo de software.

Símbolo: REGISTRAR DUDAS

Categoría: Acciones

Noción

- Permitir al [usuario](#) agregar anotaciones referentes a dudas relacionadas con un [símbolo](#) o [escenario](#) para luego aclararlas con el cliente.
- Lo realiza el [editor de LEL](#), y el [editor de escenarios](#).

Impacto

- Agregar en la [tabla de LEL](#) o la [tabla de escenarios](#) las dudas del [usuario](#).

Símbolo: REPOSITORIO DEL PROYECTO

Categoría: Almacenamientos

Noción

- Toda la información relativa a un [proyecto](#).
- Esta compuesta por una Base de Datos que contiene la [tabla de LEL](#), la [tabla de escenarios](#), la [tabla de versiones](#), la [tabla de estado](#), la [tabla de categoría](#) y la [tabla de clasificación](#).
- También tiene referencia al documento que contiene al [universo del discurso](#).

Impacto

- Es utilizado para guardar, organizar, buscar y recuperar datos acerca de un [proyecto](#).

Símbolo: SÍMBOLO

Categoría: Término Específico

Noción

- 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.
- Cada [símbolo](#) tiene un Nombre que lo identifica, una Noción que indica qué es el [símbolo](#) y el Impacto que es cómo repercute en el sistema. Si es necesario se agregan las Dudas que describen los puntos pendientes a clarificar con el [usuario](#).

Impacto

- Asegurar la comunicación con el [usuario](#).
- Facilitar la validación de los requerimientos.
- Mantener el mismo vocabulario durante todo el proceso de desarrollo.

Símbolo: TABLA DE CATEGORÍA

Categoría: Almacenamientos

Noción

- Relación que almacena información de las categorías definidas por el [usuario](#) para cada [símbolo](#) capturados para el [proyecto](#).
- Cada tupla esta compuesta por los atributos Código de Categoría, Nombre de Categoría, Nombre de Subcategoría, Noción, Impacto.

Impacto

- Se la utiliza almacenar información de la categoría de [símbolo](#) en el [repositorio del proyecto](#).

Símbolo: TABLA DE CLASIFICACIÓN

Categoría: Almacenamientos

Noción

- Relación que almacena información de las clasificaciones para los [escenarios](#) del [proyecto](#).
- Cada tupla esta compuesta por los atributos Código de clasificación y Nombre.

Impacto

- Se la utiliza almacenar información de la clasificación de [escenario](#) en el [repositorio del proyecto](#).

Símbolo: TABLA DE ESCENARIOS

Categoría: Almacenamientos

Noción

- Relación que almacena información de los [escenarios](#) que describen el funcionamiento de la aplicación.
- Cada tupla está compuesta por los atributos Nombre, Objetivo, Contexto, Recursos, Actores, Episodios, Código de Clasificación, Código de Estado y Dudas.

Impacto

- Se la utiliza para almacenar información de cada [escenario](#) en el [repositorio del proyecto](#).

Símbolo: TABLA DE ESTADO

Categoría: Almacenamientos

Noción

- Relación que almacena información de los estados definidos por el [usuario](#) para cada [símbolo](#) o [escenario](#) perteneciente al [proyecto](#).
- Cada tupla está compuesta por los atributos Código de Estado y Nombre del Estado.

Impacto

- Se la utiliza almacenar información de los estados en el [repositorio del proyecto](#).

Símbolo: TABLA DE LEL

Categoría: Almacenamientos

Noción

- Relación que almacena información de los [símbolos](#) capturados para el [proyecto](#).
- Cada tupla esta compuesta por los atributos Nombre, Noción, Impacto, Código de Categoría, Código de Estado y Dudas.

Impacto

- Se la utiliza almacenar información de cada [símbolo](#) en el [repositorio del proyecto](#).

Símbolo: TABLA DE VERSIONES

Categoría: Almacenamientos

Noción

- Relación que almacena información de las versiones existentes de cada [proyecto](#).
- Cada tupla esta compuesta por los atributos Autor, Fecha de creación, Número de versión, Lugar físico de almacenamiento, y Observaciones.

Impacto

- Se la utiliza almacenar información de cada versión en el [repositorio del proyecto](#).

Símbolo: UNIVERSO DEL DISCURSO

Categoría: Término Específico

Noción

- Documento que contiene la información que se obtuvo mediante las entrevistas o documentos que se hayan consultado para obtener conocimiento del dominio del problema.
- Contiene palabras o frases que el cliente repite o enfatiza, las cuales son la fuente para construir la primera lista de [símbolos](#).

Impacto

- Especificar con claridad, sin ambigüedad, en forma consistente y compacta, el comportamiento del sistema.

Símbolo: USUARIO

Categoría: Persona

Noción

- Persona que interactúa con la herramienta para construir un [proyecto](#).

Impacto

- Realiza peticiones a la herramienta.
-

5.3.5. Lista de Escenarios

- ELIMINAR ESCENARIO.
 - ELIMINAR SÍMBOLOS.
 - CREAR ENLACE.
 - CREAR ESCENARIO.
 - CREAR SÍMBOLOS.
 - EDITAR Y ALMACENAR DUDAS.
 - GENERAR DOCUMENTOS
 - GENERAR ESCENARIOS.
 - GENERAR VERSIÓN.
 - LINKEAR
 - MODIFICAR ESCENARIO.
 - MODIFICAR SÍMBOLOS.
 - MODIFICAR UNIVERSO DEL DISCURSO.
-

5.3.6. Descripción de Escenarios

Escenario: ELIMINAR ESCENARIO.

Objetivo:

Eliminar de forma definitiva un [escenario](#) del [repositorio del proyecto](#).

Contexto:

El [usuario](#) debe activar la opción eliminar del [editor de escenarios](#).

Recursos:

[Tabla de escenarios](#), [repositorio del proyecto](#), [escenario](#)

Actores:

[Usuario](#), [editor de escenarios](#)

Episodios:

- [El [usuario](#) confirma si quiere borrar el [escenario](#).]
- Si la confirmación es afirmativa entonces buscar y eliminar los [Enlaces](#) que apunten al [escenario](#).
- Eliminar el [escenario](#) de la [tabla de escenarios](#).
- Caso contrario, anular.

Escenario: ELIMINAR SÍMBOLOS.

Objetivo:

Eliminar de forma definitiva un [símbolo](#) del [repositorio del proyecto](#).

Contexto:

El [usuario](#) debe activar la opción eliminar del [editor de LEL](#).

Recursos:

[Tabla de LeI](#), [repositorio del proyecto](#), [símbolo](#)

Actores:

[Usuario](#), [editor de LeI](#)

Episodios:

- [El [usuario](#) confirma si quiere borrar el [símbolo](#).]
- Si la confirmación es afirmativa entonces buscar y eliminar los [enlaces](#) que apunten al [símbolo](#).
- Eliminar el [símbolo](#) de la [tabla de LEL](#).
- Caso contrario anular.

Escenario: CREAR ENLACE.

Objetivo:

Crear referencia a otro [símbolo](#) o [escenario](#).

Contexto:

El usuario incorpora un nuevo símbolo o escenario, o bien modifica el nombre de uno ya existente.

Recursos:

[Tabla de escenarios](#), [repositorio del proyecto](#), [tabla de LeI](#), [universo del discurso](#)

Actores:

[Usuario](#), [editor de universo del discurso](#), [editor de LeI](#), [editor de escenarios](#)

Episodios:

- [Capturar palabra o frase ingresada por el [usuario](#).]
- Buscar en la [tabla de LEL](#) o la [tabla de escenarios](#), según corresponda, las palabras o frases capturada.
- Si la palabra o frase existe en el [repositorio del proyecto](#) entonces cambiarle atributos del tipo de letra.

Escenario: CREAR ESCENARIO.

Objetivo:

Incorporar un nuevo [escenario](#) al [repositorio del proyecto](#).

Contexto:

El [usuario](#) tipeo un nombre en el [editor de escenarios](#).

Recursos:

[Tabla de escenarios](#), [repositorio del proyecto](#)

Actores:

[Usuario](#), [editor de escenarios](#)

Episodios:

- [Verificar que el nombre del [escenario](#) que se pretende agregar no exista en la [tabla de escenarios](#).]
- Si existe, entonces mostrarlo, en caso contrario incorporarlo a la [tabla de escenarios](#).
- [[Crear enlace](#) con el nuevo [escenario](#).]

Escenario: CREAR SÍMBOLOS.

Objetivo:

Incorporar un nuevo [símbolo](#) al [repositorio del proyecto](#).

Contexto:

El [usuario](#) tipea un nombre para un nuevo símbolo en el [editor de LEL](#).

Recursos:

[Tabla de Lel](#), [repositorio del proyecto](#), [símbolo](#)

Actores:

[Usuario](#), [editor de LEL](#)

Episodios:

- [Verificar que el nombre del [símbolo](#) que se pretende agregar no exista en la [tabla de LEL](#).]
- Si existe, entonces mostrarlo, en caso contrario incorporarlo a la [tabla de LEL](#).
- [[Crear enlace](#) con el nuevo [símbolo](#).]

Escenario: EDITAR Y ALMACENAR DUDAS.

Objetivo:

Permitir al [usuario](#) agregar anotaciones referentes a dudas relacionadas con un [símbolo](#) o [escenario](#) para luego aclararlas con el cliente.

Contexto:

El [usuario](#) debe activar la opción Dudas del [editor de LEL](#), o del [editor de escenarios](#).

Recursos:

[Tabla de escenarios](#), [tabla de LEL](#), [símbolo](#), [escenario](#)

Actores:

[Usuario](#), [editor de Lel](#), [editor de escenarios](#)

Episodios:

- Seleccionar un [símbolo](#) o [escenario](#).
- Permitir editar dudas.
- Agregar en la [tabla de LEL](#) o la [tabla de escenarios](#).

Escenario: GENERAR DOCUMENTOS

Objetivo:

Exportar la información que se encuentra en el [repositorio del proyecto](#), a formato TXT, RTF o HTML, según la solicitud del [usuario](#).

Contexto:

El usuario selecciona la opción [generar documentos](#).

Recursos:

[Repositorio del proyecto](#)

Actores:

[Usuario](#), [generador de documentos](#)

Episodios:

- Solicitar al [usuario](#) el formato de preferencia.
- Obtener información del [repositorio del proyecto](#).
- Generar el documento.

Escenario: GENERAR ESCENARIOS.

Objetivo:

Genera una lista preliminar de [escenarios](#).

Contexto:

El usuario selecciona la opción [generar escenarios](#).

Recursos:

[Tabla de escenarios](#), [tabla de Lel](#)

Actores:

[Usuario](#), [generador de escenarios](#)

Episodios:

- Construir [escenario](#) candidatos con la información obtenida del LEL, aplicando heurísticas de Hadad.
- Almacenar los [escenario](#) en la [tabla de escenarios](#).
- [Crear enlace](#).

Escenario: GENERAR VERSIÓN.

Objetivo:

Representar un momento determinado de la evolución del [proyecto](#) en desarrollo.

Contexto:

El [usuario](#) selecciono la opción [generar versiones](#).

Recursos:

[Repositorio del proyecto](#)

Actores:

[Usuario](#), [generador de versiones](#)

Episodios:

- Solicitar al [usuario](#) nombre del autor, fecha de creación y observación.
- Obtener del [repositorio del proyecto](#) información almacenada hasta ese momento.
- Generar un documento HTML con dicha información.

Escenario: LINKEAR

Objetivo:

Mostrar el contenido del [símbolo](#) o [escenario](#) seleccionado por el [usuario](#).

Contexto:

Debe estar activa la opción de navegación

Recursos:

[Tabla de Lel](#), [tabla de escenarios](#), [universo del discurso](#)

Actores:

[Usuario](#), [editor de universo del discurso](#), [editor de Lel](#), [editor de escenarios](#)

Episodios:

- El usuario selecciona un símbolo o un escenario.
- Buscar el [símbolo](#) o el [escenario](#) seleccionado en la [tabla de LEL](#) o la [tabla de escenarios](#) según corresponda.
- Crear una nueva instancia del [editor de LEL](#) o el [editor de escenarios](#).
- [Mostrar el contenido del [símbolo](#) o [escenario](#).]

Escenario: MODIFICAR ESCENARIO.

Objetivo:

Modificar el contenido de alguno de los componentes de un escenario existente en el [repositorio del proyecto](#).

Contexto:

El usuario selecciona la opción modificar en el [editor de Escenarios](#).

Recursos:

[Tabla de escenarios](#), [repositorio del proyecto](#)

Actores:

[Usuario](#), [editor de escenarios](#)

Episodios:

- Editar elementos del [escenario](#).
 - Aceptar modificaciones.
 - Almacenar elementos modificados.
- Caso alternativo: Modificar Nombre
- Eliminar los [enlaces](#).

- Realizar [crear enlace](#).

Escenario: MODIFICAR SÍMBOLOS.

Objetivo:

Modificar el contenido de alguno de los componentes de un [símbolo](#) existente en el [repositorio del proyecto](#).

Contexto:

El usuario selecciona la opción modificar en el [editor de LEL](#).

Recursos:

[Tabla de Lel](#), [repositorio del proyecto](#)

Actores:

[Usuario, editor de Lel](#)

Episodios:

- Editar elementos del [símbolo](#).
- Aceptar modificaciones.
- Almacenar elementos modificados.
- Caso alternativo: Modificar Nombre
- Eliminar los [enlaces](#).
- Realizar [crear enlace](#).

Escenario: MODIFICAR UNIVERSO DEL DISCURSO.

Objetivo:

Permitir actualizar la información recopilada durante todo el proceso de desarrollo.

Contexto:

El usuario modifica el contenido del [universo del discurso](#).

Recursos:

[Universo del discurso](#)

Actores:

[Usuario, editor de universo del discurso](#)

Episodios:

- Crear una nueva instancia del [editor de Universo del Discurso](#).
- [Mostrar el contenido del documento.]
- Permitir edición.

Almacenar la nueva información.

6. UTILIZACIÓN DE LA HERRAMIENTA

6.1. Caso de uso: Banco de Sangre

La construcción de la primera versión del Universo del Discurso y LEL para el Sistema de Registro y Producción del Instituto de Hemoterapia (Banco de Sangre), se realizó en forma manual, con la ayuda de un editor de texto común, con el cual se facilitaron las tareas comunes de edición como por ejemplo: insertar líneas de descripciones entre otras ya existentes, ordenar todas las entradas alfabéticamente por el símbolo, marcar símbolos en distintos colores, etc. Sin embargo, el conjunto de símbolos forman una red, que permite representar al LEL en un hipertexto [Leite90] y navegar en él para conocer todo el vocabulario del dominio, lo cual es muy difícil, pues si mientras se lee una descripción aparece un término, el cual también está definido, hay que buscar entre todas las entradas al símbolo para ver su descripción. Otro inconveniente se presenta en el proceso de validación, el desarrollador debe releer el texto para verificar que la consistencia se mantenga. También es necesario llevar un estado de los símbolos para saber cuándo está completamente desarrollado, o incompleto o si quedan dudas para poder evacuarlas con los usuarios.

Para la construcción de escenarios, se utilizó el mismo editor de texto con lo cual se tuvieron las mismas falencias que las mencionadas para el LEL. Además, el enfoque empleado para generar esta versión de escenarios fue analizar las funciones que se cumplían en los distintos sectores del Instituto, lo que hacía ver la necesidad de poder clasificar los símbolos para poder facilitar el proceso de búsqueda.

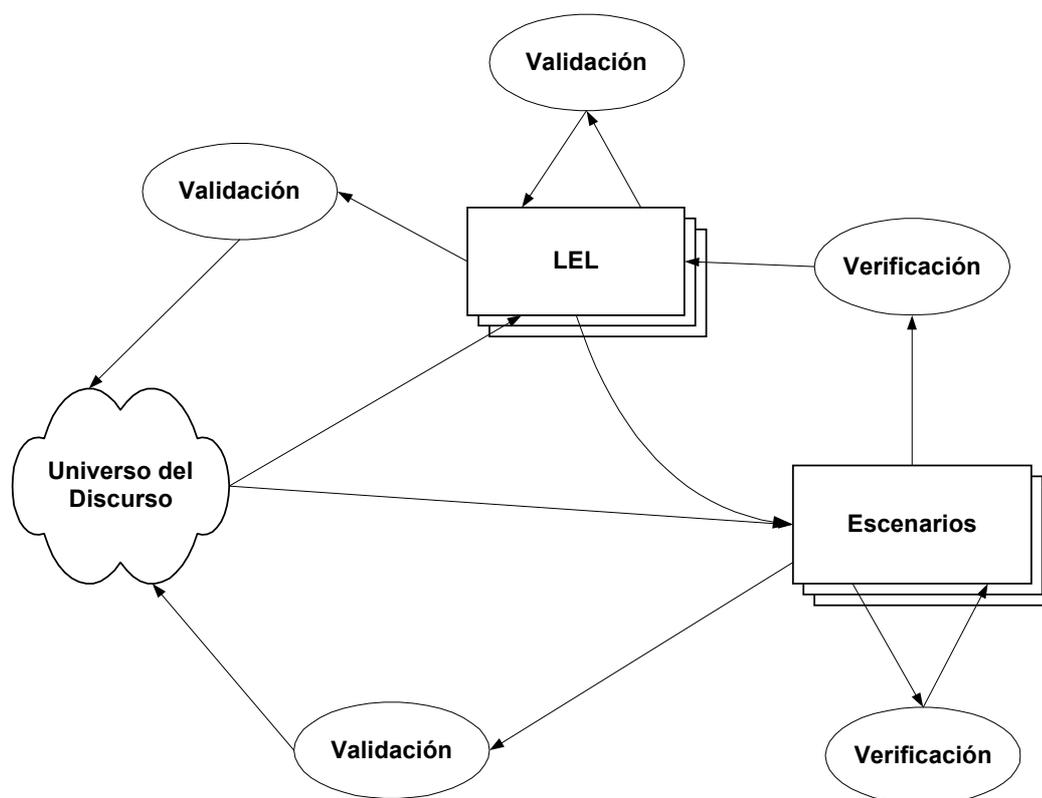
Estos inconvenientes hacen que la elicitación de requerimientos, con la metodología de LEL y escenarios, sea lenta y además la labor del desarrollador, sobre todo en la etapa de validación de símbolos, donde es necesario completar información, hacer cambios, eliminar y agregar nuevas entradas, tarea que se ve entorpecida por el consumo de tiempo en cuestiones que no hacen a la elicitación en sí.

El propósito de esta tesis fue desarrollar una herramienta para soportar la producción de LEL, incorporando en la misma un entorno integrado entre el Universo del Discurso, LEL y Escenarios con tecnología de hipertexto para facilitar la navegación entre los distintos elementos y de esta forma interactuar de una manera mucho mas flexible en la navegación y administración de la información obtenida, para conocer todo el vocabulario del dominio y la generación de escenarios.

La estrategia de trabajo que usa la herramienta es la propuesta en [Hadad96] y [Hadad97], que consiste en construir el LEL partiendo del universo del Discurso, y luego generar los Escenarios basados exclusivamente en la

información contenida en el LEL, al que se le aplican una serie de heurísticas (punto 4.4. vistas en este mismo informe). Posteriormente se realizan nuevas entrevistas con los clientes, con el fin de ampliar información sobre aquellos escenarios cuya descripción resultara incompleta y/o confusa, a partir de la información obtenida del LEL. Esto implica que la derivación de escenarios es el comienzo de la descripción de los escenarios, siendo luego necesario retornar al Universo de Discurso para concluirla.

En la siguiente figura se puede ver en forma esquemática dicha estrategia:



Estrategia utilizada en la elicitación de los requerimientos [Hadad96]

El objetivo de brindar un entorno de trabajo integrado, se cumple ampliamente con esta herramienta, permitiendo al analista trabajar de una manera más cómoda y ordenada, logrando así evitar distracciones en tareas que no son relevantes y poder prestar atención a la elicitación de requerimientos propiamente dicha.

6.1.1. Generación de Escenarios

Las heurísticas programadas en la herramienta para la construcción de los escenarios se basa exclusivamente en la información que el analista

almacenó en la tabla de LEL. Posteriormente se realizan nuevas entrevistas con los clientes, con el fin de ampliar información sobre aquellos escenarios cuya descripción resultara incompleta y/o confusa a partir de la información obtenida del LEL.

Los escenarios se describen utilizando el esquema informado en la sección 4.3., con una estructura compuesta por el **nombre**, el **objetivo**, el **contexto**, los **recursos**, los **actores** y los **episodios**, además TILS agrega el elemento **estado** con el cual el analista puede guiarse durante el desarrollo del proyecto para saber si un determinado escenario está completo, incompleto o con dudas. También es necesario que se emplee el punto de vista de la aplicación para llegar a una versión final de los mismos. Esto implica que la derivación de escenarios que se obtiene es el comienzo de la descripción de los escenarios, siendo luego necesario retornar al Universo de Discurso que también está disponible en TILS, para concluirla.

6.1.2. Identificación de los actores de la aplicación.

La herramienta, a partir de la base de datos del LEL, identifica los símbolos que representan a los actores del Universo de Discurso. Estos símbolos pertenecen a la clasificación Sujeto.

Se hace necesario recordar que la propuesta de este trabajo es realizar una subclasificación en función del Universo del Discurso de la aplicación en estudio y mantener fija la clasificación definida en [Leite90] donde se dividen los símbolos en Sujeto, Verbo, Objeto y Estado, obteniendo para este caso particular de estudio, el siguiente resultado:

Clasificación	Subclasificación	Actores
Sujeto	Personas	Asegurado / Donante Asegurado / Seguro de Sangre
		Donante / Donante Voluntario
		Donante de Reposición
		Médico
		Técnico Hematólogo / Técnico
	Sectores	Banco de Sangre
		Colecta Móvil
		Consultorio
		Fraccionamiento / Producción
		Laboratorio de Inmunoematología / Inmuno
		Instituciones Receptoras
		Laboratorio de Inmunoserología / Serología
		Recepción
		Sala de Hemodonación
		Sala de Transfusión y Aféresis

La identificación de cuáles son los actores principales y cuáles los actores secundarios no es realizada por la aplicación. 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.

6.1.3 Generación de la lista de Escenarios candidatos.

La segunda acción que realiza la herramienta es la de obtener los impactos del símbolo del LEL correspondiente a cada actor. Cada impacto representa un posible escenario, que se incorpora a la lista de escenarios candidatos. Para este caso de estudio se obtuvo la siguiente lista de 53 escenarios candidatos:

LISTA DE ESCENARIOS
1. A PARTIR DE LA MUESTRA DE SANGRE, SE DETECTAN LAS SIGUIENTES PATOLOGÍAS: SÍFILIS, CHAGAS, BRUCELOSIS, HEPATITIS-A, HEPATITIS-B, HEPATITIS-C, SIDA.
2. A PARTIR DE LA MUESTRA DE SANGRE, SE DETERMINAN SUS CARACTERÍSTICAS, GRUPO Y FACTOR ENTRE OTROS (COOMB DIRECTA, INDIRECTA, ANTÍGENO DU Y FENOTIPO CDE/CDE).
3. ACEPTA EL CONSENTIMIENTO INFORMADO.
4. ACEPTACIÓN O DESCARTE DE ADMISIÓN DEL DONANTE.
5. AÍSLAN LOS COMPONENTES EN LAS BOLSA SATÉLITE.
6. APORTAR NUEVOS MÉTODOS Y TÉCNICAS.
7. CAPACITAR AL PERSONAL TÉCNICO DE CADA SECTOR.
8. COMPLETA LA FICHA DE DONACIÓN CON LAS CARACTERÍSTICAS DE LA EXTRACCIÓN Y LA FIRMA.
9. COMPLETA LA FICHA DE DONACIÓN CON LOS SIGNOS CLÍNICOS Y LA FIRMA.
10. COMPLETAN LA PLANILLA DE PRODUCCIÓN CON LOS HEMOCOMPONENTE QUE SE OBTIENEN.
11. COMPLETAN LAS CARACTERÍSTICAS DE LA EXTRACCIÓN EN LA FICHA DE DONACIÓN Y LA FIRMAN.
12. COMPLETAN Y FIRMAN LA PLANILLA DE SEROLOGÍA.
13. DEMANDAN PRODUCTO.

14. DEPOSITA EL CUESTIONARIO DE AUTOEXCLUSIÓN EN UNA URNA.
15. DEPOSITAN LA SANGRE EN UNA BOLSA.
16. EL DONANTE ES SOMETIDO A LA TÉCNICA PLAQUETOAFÉRESIS.
17. EL DONANTE FIRMA ACEPTACIÓN DE CONDICIONES DE EXTRACCIÓN EN LA FICHA DE DONACIÓN.
18. EL MÉDICO COMPLETA INFORMACIÓN DE ADMISIÓN Y FIRMA LA FICHA DE DONACIÓN.
19. EL TÉCNICO DE LABORATORIO COMPLETA Y FIRMA LA PLANILLA DE INMUNO.
20. EN CASO DE NECESITAR EL ASEGURADO DONACIONES PARA ÉL O PARA ALGÚN FAMILIAR, NO DEBE REPONER EL PRODUCTO.
21. ES EL RESPONSABLE DE TOMAR LOS SIGNOS CLÍNICOS AL DONANTE.
22. ES ESTIMULADO POR EL INSTITUTO A REALIZAR DONACIÓN.
23. EVALUAR EL CUMPLIMIENTO DE NORMAS.
24. EXTRAEN SANGRE ENTERA O PLAQUETAS.
25. FIRMA LA FICHA DE DONACIÓN EN CONSULTORIO.
26. FIRMAR LA FICHA DE DONACIÓN
27. INFORMA SUS DATOS PERSONALES EN RECEPCIÓN.
28. LAS UNIDAD DE SANGRE SON ALMACENADAS PARA SU POSTERIOR PROCESAMIENTO.
29. PARA EL SEGURO DE SANGRE SE REGISTRAN SUS DATOS EN UN FICHERO DE ASEGURADOS.
30. PUEDE LLENAR UN CUESTIONARIO DE AUTOEXCLUSIÓN.
31. REALIZA EL ETIQUETADO INICIAL A LA DONACIÓN.
32. REALIZAN EL CALCULAR VENCIMIENTO DE LOS HEMOCOMPONENTE.
33. REALIZAN EL ETIQUETADO FINAL DE HABILITACIÓN A LOS HEMOCOMPONENTES EN BASE A LA PLANILLA DE PRODUCCIÓN, A LA PLANILLA DE SEROLOGÍA Y AL LA PLANILLA DE INMUNO.
34. REALIZAN UNA ENTREVISTA AL DONANTE EN BASE AL CONSENTIMIENTO INFORMADO.
35. REALIZAR CONTROL DE CALIDAD DEL PRODUCTO.

36. REALIZAR LA ENCUESTA AL DONANTE EN EL CONSULTORIO.
37. REGISTRA LA INSTITUCIONES RECEPTORAS DONDE SE ENCUENTRA INTERNADO EL PACIENTE.
38. REGISTRAN APELLIDO Y NOMBRE PARA EL CUAL REPONE.
39. REGISTRAN EL INGRESO Y EGRESO DE LOS HEMOCOMPONENTE.
40. REGISTRAN EL TIPO DE DONACIÓN EN LA FICHA DE DONACIÓN.
41. REGISTRAN EN LA PLANILLA DE EXISTENCIAS.
42. REGISTRAN LOS DATOS PERSONALES DEL DONANTE EN EL FICHERO DE DONACIONES.
43. REGISTRAN LOS DATOS PERSONALES DEL DONANTE EN UNA PLANILLA.
44. REGISTRAN SUS DATOS PERSONALES EN EL FICHERO DE ASEGURADOS.
45. RESPONDE A UN CUESTIONARIO REALIZADO POR UN MÉDICO.
46. ROTULAN LAS BOLSAS CON EL NOMBRE DEL DONANTE.
47. SUPERVISAR EL FUNCIONAMIENTO DE EQUIPOS.
48. TOMAN LOS SIGNOS CLÍNICOS AL DONANTE, REGISTRAN EN LA FICHA DE DONACIÓN Y LA FIRMAN.
49. TOMAN LOS SIGNOS CLÍNICOS EN LA SALA DE HEMODONACIÓN
50. TOMAN UNA MUESTRA DE SANGRE AL DONANTE.
51. VERIFICAN RESULTADO PENDIENTES DE LABORATORIO, SEROLOGÍA POSITIVA, DESCARTE TEMPORAL NO CUMPLIDO O DESCARTE DEFINITIVO EN LA ÚLTIMA DONACIÓN DESDE EL FICHERO DE DONACIONES.
52. VERIFICAN DONACIONES ANTERIORES DESDE EL FICHERO DE DONACIONES.
53. VERIFICAN VENCIMIENTOS DE HEMOCOMPONENTES.

De esta lista, se eliminan los escenarios repetidos, que se generan en el caso de que distintos actores ejecuten la misma acción en el sistema en estudio, por lo que en sus impacto se refleja esta repetición.

6.1.4. Descripción de los Escenarios candidatos.

La herramienta, a medida que genera la lista de escenarios candidatos, simultáneamente, describe cada uno de los escenarios, aplicando las reglas que se detallaron en los puntos 4.4.2 y 4.4.5 de este informe. TILS verifica si el impacto que genera un escenario candidato contiene un símbolo del LEL que pertenezca a la clasificación Verbo, si ocurre esto entonces es posible generar la descripción de dicho escenario. Se obtienen como resultado de este proceso siete escenarios provenientes de impactos que contenían un símbolo del LEL.

perteneciente a la clasificación Verbo, estos escenarios se detallan a continuación:

Nombre: ACEPTACIÓN O DESCARTE DE ADMISIÓN DEL DONANTE

Objetivo:

DESCARTE DE ADMISIÓN, evitar que los donante que no cumplen los requisitos básicos realicen una donación se lleva a cabo en el sector de recepción

Contexto:

Recursos:

DONACIÓN, FICHA DE DONACIÓN

Actores:

DONANTE, RECEPCIÓN

Episodios:

rechazar al donante debido a: no se cumplieron 60 días de la última donación, tiene resultados pendientes de laboratorio, posee un descarte temporal no cumplido o posee descarte definitivo
se registra la inhabilitación del donante en la ficha de donación

Nombre: EL DONANTE ES SOMETIDO A LA TÉCNICA PLAQUETOAFÉRESIS

Objetivo:

PLAQUETOAFÉRESIS, el donante necesita preparación previa técnica para separar las plaquetas de la sangre entera se extrae la sangre de un brazo al donante, se separan las plaquetas y el resto se repone al donante mediante flujo continuo el proceso dura dos horas es utilizada en la sala de transfusión y aféresis

Contexto:

Recursos:

DONACIÓN, PLANILLA DE EXISTENCIAS, PLAQUETAS, SANGRE ENTERA

Actores:

BANCO DE SANGRE, DONANTE, SALA DE TRANSFUSIÓN Y AFÉRESIS

Episodios:

el donante es inhabilitado por 48 horas para una nueva donación
registrar en la planilla de existencias al sector banco de sangre donde no se le aplica nada

Nombre: REALIZA EL ETIQUETADO INICIAL A LA DONACIÓN

Objetivo:

ETIQUETADO INICIAL, identificar la donación internamente permite relacionar las muestras con el producto durante el ciclo de producción se lleva a cabo en la sala de hemodonación

Contexto:

Recursos:

DONACIÓN, MUESTRA

Actores:

PRODUCCIÓN, SALA DE HEMODONACIÓN

Episodios:

se confeccionan a lo sumo ocho etiquetado inicial

Nombre: REALIZAN EL CALCULAR VENCIMIENTO DE LOS HEMOCOMPONENTE

Objetivo:

CALCULAR VENCIMIENTO, se realiza en el sector banco de sangre para determinar la fecha de vencimiento del producto en base a: anticoagulante, calidad, tipo de hemocomponente y fecha de extracción, realizar controles de validez de producto

Contexto:

Recursos:

HEMOCOMPONENTE, PLANILLA DE EXISTENCIAS

Actores:

BANCO DE SANGRE

Episodios:

registrar vencimiento en la planilla de existencias

Nombre: REALIZAN EL ETIQUETADO FINAL DE HABILITACIÓN A LOS HEMOCOMPONENTES EN BASE A LA PLANILLA DE PRODUCCIÓN, A LA PLANILLA DE SEROLOGÍA Y AL LA PLANILLA DE INMUNO

Objetivo:

ETIQUETADO FINAL DE HABILITACIÓN, dar de alta producto al stock de banco de sangre rastrear en cualquier momento los resultados de laboratorio de una donación con posterioridad a la distribución se realiza en el sector banco de sangre

Contexto:

Recursos:

DONACIÓN, ETIQUETA FINAL, HEMOCOMPONENTE, PLANILLA DE EXISTENCIAS

Actores:

BANCO DE SANGRE

Episodios:

los hemocomponente se almacenan en el banco de sangre en base a la etiqueta final registrar alta de productos en planilla de existencias

Nombre: REALIZAR CONTROL DE CALIDAD DEL PRODUCTO

Objetivo:

CALIDAD, es un proceso para modificar la excelencia del producto para usos específicos se puede realizar antes o después del almacenamiento del hemocomponente es una de las variables que interviene en el calcular vencimiento de un hemocomponente.

Contexto:

Recursos:

HEMOCOMPONENTE

Actores:

Episodios:

se registra en la planilla de existencias.

Nombre: VERIFICAN RESULTADO PENDIENTES DE LABORATORIO, SEROLOGÍA POSITIVA, DESCARTE TEMPORAL NO CUMPLIDO O DESCARTE DEFINITIVO EN LA ÚLTIMA DONACIÓN DESDE EL FICHERO DE DONACIONES

Objetivo:

DESCARTE DEFINITIVO, evitar que el donante realice donaciones futuras son descartes de consultorio fijados para un período indeterminado de tiempo se lleva a cabo en consultorio, DESCARTE TEMPORAL, evitar que el donante realice donación sin estar en condición son descarte de admisión, descarte de consultorio y descarte de extracción fijados para un período de tiempo se lleva a cabo en admisión, consultorio o sala de hemodonación

Contexto:

Recursos:

DONACIÓN, FICHA DE DONACIÓN

Actores:

CONSULTORIO, DONANTE, SALA DE HEMODONACIÓN

Episodios:

inhabilita al donante definitivamente

se registra en la ficha de donación, inhabilitar al donante temporalmente

registrar en la ficha de donación

6.1.5. Vista gráfica de la construcción de un escenario.

TILS permite ver gráfica y simultáneamente, los Escenarios generados, los símbolos del LEL y el Universo del discurso, lo cual facilita las tareas de revisión y validación de los escenarios, como se puede ver en la figura N° 57. En ella hay un posible Escenario (ACEPTACIÓN O DESCARTE DE ADMISIÓN DEL DONANTE), que se origina en un símbolo que representa a un actor del Universo de Discurso (Recepción). Estos símbolos pertenecen a la clasificación Sujeto, que agrupa a los individuos que interactúan en el Universo del Discurso. El impacto que representa un posible escenario en el símbolo Recepción es ACEPTACIÓN O DESCARTE DE ADMISIÓN DEL DONANTE, que contiene un símbolo del LEL que pertenece a la clasificación Verbo (Descarte de Admisión), de cuyos impactos se definen los episodios del escenario.

Esta vista gráfica también permite visualizar fácilmente el origen del objetivo del escenario, los actores y los recursos, que surgen de lo expuesto en el punto 4.4.3.1 de este informe.

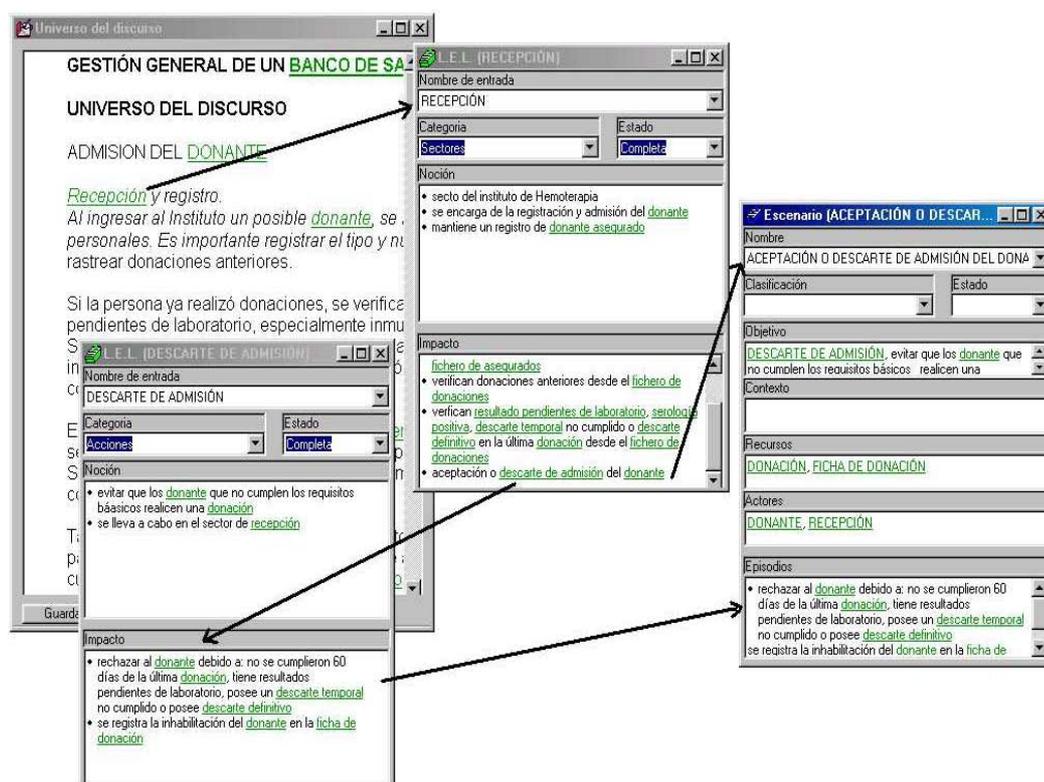


Figura N° 57: vista gráfica de la construcción de un Escenario.

6.2. Conclusiones.

La generación de la lista de escenarios candidatos a partir del LEL, es un excelente inicio para detectar escenarios y a partir de allí completar su descripción, esto se puede verificar en los casos de estudio del Banco de Sangre, Sistema Nacional para la Obtención de Pasaportes y Agenda de Reuniones. En estos dos últimos, los resultados obtenidos fueron similares al primero, como por ejemplo, el número de escenarios generados por la aplicación es siempre superior al número de escenarios finales del proyecto, esto se refleja en la siguiente tabla:

Proyecto	Nº de escenarios generados	Nº de escenarios del proyecto terminado
Banco de Sangre	53	20
Obtención de Pasaportes	63	24
Agenda de Reuniones	30	13

de estos resultados, se puede ver que para los dos primeros casos se generan aproximadamente un 62 % más de los escenarios que tiene cada proyecto en sus listas definitivas, y para el caso de la agenda de Reuniones un 57 % más, con la característica de que los escenarios sobrantes son en realidad subescenarios o episodios de los escenarios terminados.

Sin embargo, es útil recordar que la herramienta no realiza automáticamente la revisión y validación tratados en los puntos 4.4.6. y 4.4.7. de este informe, y que son de fundamental importancia dado que todavía se puede detectar escenarios candidatos como episodios simples dentro de otros escenarios. Como ya se mencionó, 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, ver punto 4.4.3.2.

Estos tipos de escenarios son los que se tendrán que discutir con el cliente para determinar si involucra más de un episodio. En caso afirmativo, se mantiene el escenario en la lista y, en los escenarios que figuran como una acción simple, se reemplaza el episodio por el nombre del escenario, es decir, ocurre la detección de subescenarios. En caso contrario, se elimina de la lista el escenario candidato.

Otra manera de detectar subescenarios es cuando existen escenarios candidatos como un conjunto de episodios dentro de otros escenarios. En algunos escenarios puede ocurrir que, en la descripción de episodios, un conjunto de éstos correspondan a un escenario proveniente de un actor. En estos casos se reemplaza el conjunto de episodios por el escenario proveniente del actor.

Es común también realizar la 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.

El momento adecuado para definir las restricciones a los episodios es en la etapa de revisión de los escenarios, que luego son verificadas con los clientes. La validación con los clientes de cada uno de los escenarios generados, es fundamental para obtener la lista definitiva, esta es una tarea que se facilita notablemente con la utilización de TILS.

7. Contribuciones.

Este trabajo de tesis presenta una herramienta para implementar Lel y Escenarios (TILS) que soporta satisfactoriamente la producción del Universo del Discurso, LEL y Escenarios, incorporando en la misma, tecnología de hipertexto para facilitar la navegación entre los distintos elementos que forman el léxico en un dominio particular, permitiendo así, interactuar de una manera mucho más flexible en la navegación y administración de la información obtenida para conocer todo el vocabulario del dominio.

TILS fue probada en diversos casos de estudio, de los cuales se desarrolló uno en particular, presentado en la sección 3, el cual es un estudio de campo real en un dominio particular caracterizado por la especificidad del lenguaje utilizado. En primer lugar se verificó que se puede implementar el Universo del Discurso, LEL y Escenarios de un dominio con un lenguaje altamente especializado utilizando las herramientas de la Ingeniería de Requerimientos y apoyados en una herramienta CASE que soporta una estrategia ya aceptada dentro de la comunidad de requisitos. Se verificó que si los desarrolladores comparten el mismo lenguaje que los usuarios, se facilita notablemente el conocimiento del dominio, evitando con esto que se produzcan subconjuntos de afirmaciones contradictorias entre sí en la Especificación de Requerimientos de Software [IEEE 84], fortaleciendo a su vez la relación usuario-desarrollador en una etapa muy temprana.

En las pruebas que se realizaron a TILS, se aplicó el proceso establecido para construir el LEL presentado en la sección 3.3. de esta tesis con ligeras variantes que sugieren la necesidad de definir la interacción específica entre las etapas propuestas, como por ejemplo: En las etapas iniciales se detectó un fuerte nivel de interacción entre ellas, el acoplamiento entre las tareas de las etapas 1 y 2 resultó elevado, el mismo efecto se detectó entre las etapas 3 y 4. En la labor de construcción del LEL resultó recurrente que, cuando se estaba conociendo el vocabulario de la aplicación se retornaba una y otra vez a la lista de símbolos ya elaborada. En las etapas de Clasificación y Descripción de los Símbolos, al trabajar con símbolos no habituales para los analistas (los Términos y Métodos Específicos) era imperioso la consulta y procesamiento continuo del material de ambas etapas. Este fuerte acoplamiento puede que se origine en la no habitualidad en el manejo de ciertos tecnicismos por parte de los ingenieros de requerimientos.

Si bien el modelo prevé fuertes interacciones a través de un desarrollo interdependiente de las etapas e incluso en paralelo, en el desarrollo del experimento las cuatro primeras etapas se agruparon en dos: *Entrevistas y generación de símbolos* (agrupando etapas 1 y 2) y *Clasificación y descripción de símbolos* (agrupando etapas 3 y 4), quedando el proceso de construcción del Lel de la siguiente manera:

1. Entrevistas y generación de símbolos
2. Clasificación y descripción de símbolos
3. Validación con los clientes
4. Control del LEL

Todo indica que esta necesidad de un proceso de cuatro etapas deriva de la problemática particular del uso de un Lenguaje Técnico.

La incorporación de templates en TILS, para los tipos de símbolos resultó útil para asegurar la homogeneidad del LEL y facilitar su revisión. Al mismo tiempo parece necesario desarrollar un cuerpo teórico más importante para los conceptos de tipo de entrada, subclasificación y template. Las experiencias de desarrollo del LEL han desarrollado clasificaciones ad-hoc, pero hasta ahora se carece de un cuerpo de fundamentos del proceso de definición de estas clasificaciones especiales.

La experiencia de construir el LEL en un dominio del problema altamente complejo desde el punto de vista del lenguaje utilizado y enteramente ajeno al área de conocimiento de los ingenieros de requerimientos puede considerarse exitosa. La extensión propuesta de considerar tipos de entradas particulares para los términos específicos del campo en consideración demostró ser un recurso eficiente para la comprensión del problema por parte de los usuarios. Si bien se carece de una medición objetiva, corresponde mencionar que los usuarios afirmaron en varias oportunidades que encontraban en los investigadores un gran conocimiento del problema.

Desde el punto de vista práctico, se puede resumir algunos de los aspectos más importantes de este trabajo de tesis:

- **Disponer de una herramienta que soporta el proceso presentado en los primeros capítulos de esta tesis.**
- **TILS soporta derivación automática de Escenarios.**
Uno de los aspectos más relevantes de TILS, es la de administrar la clasificación de símbolos del LEL, por ejemplo, a partir de una clasificación general y en base a ésta, el usuario puede hacer una subclasificación particular para el problema que quiere atacar, de esta manera se puede generar automáticamente una versión preliminar de Escenarios a partir de las heurísticas propuestas en el trabajo de [Hada 99], las cuales están implementadas en TILS. La generación automática de los escenarios permite ahorrar mucho tiempo, liberando al desarrollador de las tareas de administración por lo cual puede dedicarse más a tareas específicas de ingeniería de requerimientos como por ejemplo, completar los escenarios ya generados por la herramienta, permitiendo determinar el símbolo del LEL que le dio origen al escenario para ampliar información y, por

supuesto, también para eliminar escenarios repetidos, o escenarios que no merecían ser tales sino simplemente episodios de otros.

- **TILS provee de soporte de administración para los usuarios.**
Normalmente la construcción del LEL y Escenarios se realizan manualmente y luego se transcriben en un editor de texto. Este tipo de proceso es lento y entorpece la labor del desarrollador, sobre todo en las tareas de validación de símbolos, donde es necesario completar información, hacer cambios, eliminar y agregar nuevas entradas, lo que consume mucho tiempo en tareas que no hacen a la elicitación de requerimientos. TILS, permite interactuar de una manera mucho más flexible en la navegación y administración de la información obtenida para conocer un dominio específico. Para lo cual esta herramienta incorpora para cada símbolo un estado; por ejemplo: Completo, Incompleto o con dudas, lo que permitiría una administración de los símbolos más eficiente.
- **TILS facilita las etapas de Revisión y Validación.**
La navegación que soporta TILS entre Universo del Discurso, Lel y Escenarios, facilita la depuración de todos los elementos, permitiendo aclarar dudas y revisar que no hayan contradicciones en la definición de LEL y escenarios.
- **Se verifico en la experimentación la utilidad de TILS como herramienta CASE.**

Es factible pensar en nuevas versiones de TILS en las cuales se pueda definir el modelo de objetos del proyecto. TILS ya cuenta con una iteración entre el LEL que define, a través de sus impactos, el comportamiento global que tiene una entidad dentro del macrosistema y los Escenarios que pueden ser utilizados para encontrar los objetos, delinear sus responsabilidades, sus colaboraciones con otras clases y sus asociaciones, permitiendo construir un modelo estático del sistema.

También es inminente pensar en migrar TILS a una arquitectura Cliente/Servidor, logrando de esta manera repartir el tratamiento de la información y los datos por todo el sistema informático, permitiendo mejorar el rendimiento global de un equipo de desarrollo.

8. REFERENCIAS

- [Pressman] Pressman, R.S., *Ingeniería del Software. Un enfoque práctico*, McGraw-Hill, Madrid, 1993.
- [Boehm79] Boehm, B. W. *Characteristics of Software Quality* – Nueva York, North Holland, 1979.
- [Hsia94] Hsia, Pei et. al. *Formal Approach to Scenario Analysis*. IEEE Software, march 1994. (33-40)
- [Rubin92] Rubin, K. and Goldberg, A. *Object Behavior Analysis*. CACM, Vol.35, No. 9, September 1992.
- [Jacobson92] Jacobson I. et al. *Object Oriented Software Engineering: A Use-Case Driven Approach*. Addison Wesley, 1992
- [Leite90] Leite J.C.S.P., Franco A.P., *O so de Hipertexto na Elicitação de Linguagens da Aplicação*". IV Simpósio Brasileiro de engenharia de software Sau Pedro, 1990
- [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.
- [Norris] Norris N. Y Rigby P. et. al. *Ingeniería de Software Explicada*. Megabyte – Grupo Noriega Editores, 1994.
- [Kaplan] Kaplan,V.,Hadad, G., Oliveros, A., *Uso de Lexico Extendido del Lenguaje (LEL) y de Escenarios para la Elicitacion de Requerimientos. Aplicación a un Caso Real*, Informe de Investigación Dpto. de Investigación de la Universidad de Belgrano, Argentina
- [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
- [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.

- [Leite'89] Leite, J.C.S.P., *Application Languages: A Product of Requirements Analysis*, Departamento de Informática, PUC-RJ, January 1989.
- [Loucopoulos] Loucopoulos, P., Karakostas, V., *System Requirements Engineering*, McGraw-Hill, London, 1995
- [Rumbaugh] Rumbaugh, J. *Getting Started: Using the Use Cases to Capture Requirements*. JOOP, sept. 1994
- [Pressman] Pressman, R.S., *Ingeniería del Software. Un enfoque práctico*, McGraw-Hill, Madrid, 1993
- [Booch] Booch, G., *Object Oriented Design with Applications*, The Benjamin/Cumming Publishing Company, Inc., Redwood City, 1991
- [Booch99] G. Booch, J. Rumbaugh, y I. Jacobson. *The Unified Modeling Language User Guide*. Addison–Wesley, 1999.
- [Jackson] Jackson, M., *Software Requirements & Specifications. A lexicon of practice, principles and prejudices*, Addison-Wesley, Cambridge, 1995
- [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.
- [Leite'93] Leite, J.C.S.P., "Eliciting Requirements Using a Natural Language Based Approach: The Case of the Meeting Scheduler Problem", March 1993.
- [Antonelli] Rubén Leandro Antonelli, "Herramienta asistente para el análisis de requerimientos", Trabajo de Grado, Universidad Nacional de La Plata.
- [Hadad99] Hadad, G., Kaplan, G., Oliveros, A., Leite, J.C.S.P., "Integración de Escenarios con el Léxico Extendido del Lenguaje en la elicitación de requerimientos: Aplicación a un caso real", *Revista de Informática Teórica y Aplicada*, Vol 6, Nro. 1, Julho 1999.
- [Gil99a] Gil, G.D, Arias Figueroa, D, *Léxico Extendido del Lenguaje y Escenarios del Instituto de Hemoterapia de la Provincia de Buenos Aires*. CIUNSa – Consejo de Investigación de la Universidad de Salta, 1999.
- [Gil99b] Gil, G.D, Arias Figueroa, D, *Gestión General de un Banco de*

- [Potts] *Sangre. Universo del Discurso*, CIUNSa – Consejo de Investigación de la Universidad de Salta, 1999.
Potts, C., Takahashi, K., Antón, A.I., “Inquiry-Based Requirements Analysis”, *IEEE Software*, Vol. 11, n.2, Mar. 1994.
- [Jacobson] I. Jacobson, M. Christerson, P. Jonsson, y G. Övergaard.
Object-Oriented Software Engineering: A Use Case Driven Approach. Addison-Wesley, 4a edición, 1993.
- [Howes] Howes, Norman, *On Using the Users’ Manual As the Requirements Specification II*, en IEEE Tutorial on Software Engineering Project Management, 1990.
- [Carrol] Carrol, J. (ed.), *Scenario Based Design: Envisioning Work and Technology in System Development*, Wiley, New York, 1995.
- [Zorman] Zorman, L., *Requirements Envisaging by Utilizing Scenarios (Rebus)*, Ph.D. Dissertation, University of Southern California, 1995.
- [Booch] Booch, G., “Scenarios”, in *Report on Object Oriented Analysis and Design*, Vol 1, 3, Sep-Oct 1994
- [Potts 95] Potts Colin. Using schematic scenarios to understand user needs. Diciembre 1995.
- [Leite’97] Leite, J.C.S.P., Rossi, G., Balaguer, F., Maiorana, V., Kaplan, G., Hadad, G., Oliveros, A., “Enhancing a Requirements Baseline with Scenarios”, in *Proceedings of the Third IEEE International Symposium on Requirements Engineering*, IEEE Computer Society Press, Los Alamitos, 1997
- [IEEE 84] Institute of Electrical and Electronic Engineers, *IEEE Guide to Software Requirements Specifications*. ANSI/IEEE Standard 830-1984. New York 1984. En Thayer, R. y Dorfman, M., *System and*
- [Sommerville] IAN SOMMERVILLE. *Software Engineering*. Addison-Wesley. 1995.
- [Gil2000] Gustavo D. Gil – Daniel Arias Figueroa – Alejandro Oliveros “Producción del LEL en un Dominio Técnico. Informe de un caso”. Presentado en WER 2000 – Workshop en Ingeniería de Requerimientos – Río de Janeiro – Brasil – 13 y 14 de julio del 2000.
- [DAVIS] DAVIS WILLIAM S. *Herramientas Case*. Paraninfo. 1992.
- [Piattini] M. G. Piattini, J. A. Calvo-Manzano, J. Cervera, y L. Fernández. *Análisis y Diseño Detallado de Aplicaciones Informáticas de Gestión*. Rama, 1996.
- [Raghavan] S. Raghavan, G. Zelesnik, y G. Ford. *Lecture Notes on*

- Requirements Elicitation. Educational Materials CMU/SEI-94-EM-10, 1994. Disponible en <http://www.sei.cmu.edu>.
- [Schneider] G. Schneider y J. P. Winters. *Applying Use Cases: a Practical Guide*. Addison-Wesley, 1998.
- IEEE 90 IEEE Std 610.12-1990, *IEEE Standard Glossary of Software Engineering Terminology*. http://standards.ieee.org/reading/ieee/std_public/
- [Firesmith 97] D. G. Firesmith. *Uses Cases: the Pros and Cons*, 1997. Disponible en: <http://www.ksscary.com/usecjrnl.html>.
- ESSENTIAL SYSTEMS ANALYSIS, Stephen M. McMenamin & John F. Palmer, prefacio por Tom De Marco- Yordon Press- Prentice Hall Building. Impreso en 1984 por Prentice-Hall, inc. una division de Simon & Shuster. ISBN 0-13-287905-0 025.