

# Enfoque Middle-Out en la Construcción e Integración de Escenarios

Graciela D. S. Hadad<sup>1</sup>, Jorge H. Doorn<sup>2</sup>, Gladys N. Kaplan<sup>3</sup>,  
Julio Cesar Sampaio do Prado Leite<sup>4</sup>

<sup>1</sup>Universidad de Belgrano y LINTI, Universidad Nacional de La Plata  
E-mail: ghadad@ub.edu.ar

<sup>2</sup>Univ. de Belgrano y Univ. Nacional del Centro de la Prov. Buenos Aires  
E-mail: jdoorn@exa.unicen.edu.ar

<sup>3</sup>Universidad de Belgrano  
E-mail: gkaplan@ub.edu.ar

<sup>4</sup>Pontificia Universidade Católica do Rio de Janeiro  
E-mail: julio@inf.puc-rio.br

**Abstract:** En la literatura referida a escenarios existen varios autores que proponen un proceso de construcción de escenarios basado en un enfoque top-down<sup>1</sup>, es decir partiendo de uno o más Escenarios Generales y construyendo los siguientes escenarios en una forma de refinamiento por pasos. Por otro lado, existen otros autores que consideran que los escenarios deberían construirse desde lo particular. Nuestra experiencia con escenarios señala que un enfoque middle-out es más conveniente. En este artículo detallamos dicha estrategia y describimos cómo esta estrategia es importante para integrar escenarios y subescenarios. Nuestras conclusiones están basadas en varios casos de estudio.

**Palabras Claves:** Escenarios, Ingeniería de Requisitos, Elicitación, Léxico, Top-down, Bottom-up, Middle-Out.

## 1 Introducción

La necesidad de asegurar una buena comprensión entre los ingenieros de requisitos y los clientes/usuarios ha inducido al desarrollo de métodos que permiten la colaboración entre todos los participantes del proceso de definición de requisitos. Los ingenieros de requisitos deben comprender, modelar y analizar el macrosistema donde el software correrá y los clientes/usuarios deben confirmar que la visión de los ingenieros es correcta. Para lograr esto se requiere una técnica de comunicación atractiva para todos los participantes. Es aquí donde los escenarios se vuelven importantes, dado que pueden mantener mucha información de una forma que todos pueden utilizar. Todo escenario describe una situación teniendo en cuenta aspectos de uso, permitiendo conocer el problema, unificar criterios, obtener el compromiso de los clientes/usuarios, organizar los detalles involucrados y entrenar nuevos participantes (Carroll 1995). El uso de escenarios como una técnica para comprender un problema a ser resuelto mediante un sistema de software ha sido recomendado por

---

<sup>1</sup> Se utilizan los términos *top-down*, *bottom-up* y *middle-out* considerando que son conceptos establecidos en la comunidad informática.

varios autores (Potts 1995), (Booch 1991), (Jacobson et al. 1992), (Zorman 1995) y dichas propuestas son muy importantes para extender el uso de escenarios en la práctica real. Sin embargo, un análisis detallado de las recomendaciones y conclusiones muestra algún grado de dispersión y contradicciones. Esta falta de precisión sobre cuándo y cómo los escenarios deberían usarse ha sido extendida a quienes usan estas técnicas en la práctica. Por eso la mayoría de los desarrolladores ven la creación de escenarios como un arte más que como una tarea de ingeniería. Estudios recientes relativos al uso real de escenarios en la Ingeniería de Requisitos (Weidenhaupt et al. 1998) (Rolland et al. 1998) han probado claramente este hecho y señalado la necesidad de más definiciones detalladas acerca de la construcción de escenarios como una contribución ineludible para incrementar su uso en situaciones reales. Esto está en completo acuerdo con Sutcliffe (1997a), quien piensa que hay poca comprensión acerca de cómo los escenarios deberían construirse, poca evidencia acerca de su efectividad y menos aún de por qué funcionan. La variedad de interpretaciones, sintaxis y mecanismos de construcción de escenarios va más allá hasta mostrar contradicciones básicas, dado que no existe acuerdo entre los autores acerca de qué enfoque debe recomendarse para el proceso de construcción de escenarios: top-down o bottom-up. Este artículo se basa en la hipótesis que la diversidad de los criterios proviene precisamente de esta alternativa, y creemos que es una falsa opción en el sentido que no debe seguirse ni un enfoque top-down ni un enfoque bottom-up, sino uno “middle-out”.

Este artículo está organizado de la siguiente manera: en la sección siguiente se comparan los enfoques de distintos autores en la construcción de escenarios y casos de uso, en la sección 3 se describe nuestra propuesta middle-out, en la sección 4 detallamos nuestra experiencia y finalmente presentamos las conclusiones.

## **2 Comparando Enfoques en la Construcción de Escenarios**

Existen varios estilos para construir escenarios, tales como narrativa textual, storyboards, video mock-ups, prototipos escritos y otros (Potts et al. 1994) (Booch 1991) (Jacobson et al. 1992) (Carroll 1995) (Zorman 1995). Algunos de ellos no recomiendan un enfoque específico pero generalmente existe un principio subyacente top-down o bottom-up. Es fácil encontrar dicho enfoque observando cuidadosamente las heurísticas que se proponen. Algunas de las propuestas más conocidas se analizan brevemente en los siguientes párrafos.

Aunque no está claramente establecido, Booch (1994) parece adherirse al enfoque top-down dado que: *“La aplicación más compleja puede caracterizarse en términos de unas pocas docenas de escenarios primarios. Los Escenarios Primarios modelan el problema central. Un Escenario Primario representa alguna función fundamental del sistema. Los Escenarios Secundarios representan alguna variación en el tema de un Escenario Primario. El comportamiento deseado íntegro de un sistema de software puede ser capturado a través de una red de escenarios, de la misma forma que un storyboard hace a una película.”* Los escenarios primarios de Booch pueden verse como los escenarios relevantes de Firesmith (1994), los cuales se interconectan por un diagrama de ciclo de vida de escenarios: *“El diagrama de ciclo de vida de*

escenarios se utiliza para documentar las interacciones válidas entre los escenarios de mayor nivel de un sistema o ensamblado.” Y “Como los escenarios son abstracciones funcionales, existe a menudo una fuerte tendencia a descomponerlos funcionalmente.” La propuesta de Sutcliffe (1997b) también puede incluirse en el enfoque top-down teniendo en cuenta la heurística: “1. Captura inicial de requisitos y familiarización con el dominio. Un escenario fundamental se desarrolla basado en el análisis preliminar del dominio, 2. Especificación y desarrollo de un demostrador de conceptos (prototipo temprano). Diagramas de justificación de diseño se usan para explicar opciones de diseño en los puntos claves, 3. Análisis de requisitos - sesión de validación para criticar el demostrador de conceptos.” El proceso de Formalización de Escenarios usando gramáticas regulares en un árbol de escenarios descrito por HSIA et al. (1994) es también un ejemplo de enfoque top-down. La propuesta de Schneider et al. (1998) incluye cuatro fases primarias siguiendo un enfoque top-down: “1. Fase de concepción: identificar actores y se desarrollan casos de uso de nivel alto para ayudar a definir el alcance del proyecto; 2. Fase de elaboración: se desarrollan los casos de uso más detallados, se utilizan para crear el plan para la fase siguiente. El producto de esta fase: escenarios primarios detallados y escenarios secundarios; 3. Fase de construcción: los casos de uso se utilizan como punto de partida para diseñar y para desarrollar planes de prueba; 4. Fase de transición: los casos de uso se pueden usar para desarrollar guías de usuarios y entrenamiento.”

Por otro lado, la propuesta de Dano et al. (1997) está cercana al enfoque bottom-up, dado que proponen “recolectar y describir casos de uso basados en una notación tabular” y luego “crear redes de Petri para cada caso de uso para brindar el formalismo necesario al analista” y finalmente “establecer vínculos formales entre casos de uso para obtener una descripción global de la aplicación.” En el artículo de Dano los términos escenarios y casos de uso son usados como sinónimos. El rol de los vínculos formales entre casos de uso de Dano et al. es representado a través de una técnica de preguntas-respuestas sistemáticas por Robertson (1995) “dado que un sólo escenario da información parcial, el análisis basado en escenarios involucra la observación de un conjunto de escenarios.” y “La técnica sistemática de preguntas-respuestas ayuda a crear un puente entre los eventos de un escenario específico y las situaciones generales que posibilitan, causan o explican eventos.” El Modelo de Ciclo de Indagación de Potts et al. (1994) puede incluirse también en el enfoque bottom-up teniendo en cuenta: “Los escenarios se representan en dos niveles de detalle: 1. Episodios o fases que son secuencias de acciones de granularidad fina, 2. Familias de Escenarios usando relaciones de uso de Jacobson.”

Es además útil observar la construcción de casos de uso, donde enfoques top-down pueden encontrarse, tales como en Wirfs-Brock (1995) considerando los pasos recomendados: “1. Determinar el sistema de software (límites, alcance, identificar actores, identificar interfaces con el sistema, desarrollar el primer corte para el particionamiento en subsistemas), 2. Estereotipar los actores (activos o pasivos, roles), 3. Determinar los casos de uso del sistema (descomponer el sistema en porciones discretas significativas tanto para la gente de negocio y los desarrolladores), 4. Construir conversaciones (secuencia de interacciones entre los actores y el sistema para cada caso de uso)”. También la propuesta de Oberg et al. (1998) corresponde a un enfoque top-down, dado que los pasos propuestos involucran: “1. Análisis del Problema”, “2. Comprender las necesidades de los

involucrados”, que incluye “Encontrar los actores y los casos de uso” y “Delinear el modelo de casos de uso”, “3. Definir el sistema”, que incluye las tareas: “Refinar el modelo de casos de uso” y “Describir los casos de uso”, continúa con: “4. Administrar el alcance del proyecto”, que involucra “Dar prioridades a los casos de uso” y “Modelar una vista de casos de uso de la arquitectura del sistema”, sigue “5. Refinar la definición del sistema” que incluye: “Detallar los casos de uso”, y finalmente “6. Administrar cambios en los requisitos”. Un enfoque intermedio, más cercano al top-down, es el presentado por Constantine (1998) donde el proceso para la generación de usos casos comprende: “Identificar casos de uso candidatos basados en un modelo de roles de usuarios y brainstorming: generar una lista de casos de uso; Bosquejar un mapa de casos de uso basado en un entendimiento inicial (relaciones entre casos de uso); Desarrollar narrativas de casos de uso en una forma estructurada; Reducir casos de uso concretos a la forma esencial; Intercambiar entre narrativas y el mapa cuando sea necesario”.

Por otro lado, el método de Jacobson (1994) (1994b) sigue un enfoque estrictamente bottom-up dado que: “Primero describimos los casos de uso básicos y luego describimos cómo mejorarlos para permitir casos de uso más avanzados, algunos de los cuales pueden depender de los básicos. Los casos de uso abstractos deben evolucionar a partir de casos de uso concretos, no de la otra forma. Extensiones de asociación nos permiten capturar los requisitos funcionales de un sistema complejo, de la misma forma que aprendemos cualquier nuevo tema: Primero comprendemos las funciones básicas, luego introducimos complejidad.”

Así mismo Gough et al. (1995) siguen un enfoque cercano al propuesto en este artículo, dado que recomiendan: “1. Creación de documentos en lenguaje natural: documentos de alcance del proyecto, documentos de necesidades del cliente, documentos de necesidades de servicio, análisis de competencia y análisis de mercado. Obtener especificaciones funcionales de un sistema existente, 2. Identificar los actores principales en el sistema, 3. Derivar los encabezados de escenarios, 4. Completar la descripción de escenarios”. En las conclusiones establecen claramente que: “La necesidad de proveer una vista a nivel alto de la interacción de escenarios como medio para destacar la complejidad de un gran número de escenarios.”

Weidenhaupt et al. (1998) han encontrado cuatro tipos de evolución para la definición de escenarios en la práctica real: “i) Descomposición Top-down”, “ii) Desde escenarios de caja negra a escenarios de caja blanca”, “iii) Desde definiciones de escenarios informales a formales” y “iv) Desarrollo incremental de escenarios.” El primero, el segundo y el cuarto son enfoques top-down.

Desde un punto de vista más amplio Jackson (1995) analiza el enfoque top-down cuando se usa para obtener conocimiento del problema a ser resuelto, encontrando dos dificultades básicas: “La primera es que Top-Down obliga al ordenamiento más riesgoso posible en las decisiones. La decisión más grande es la subdivisión del problema completo: es el más grande en escala y el más grande en sus consecuencias. A pesar de ello, esta decisión se toma primero, cuando aún no se conoce nada y todo resta por descubrirse.” Y “La segunda dificultad es que el mundo real casi nunca tiene una estructura jerárquica. Por el contrario, hay muchas estructuras paralelas y superpuestas”. Su consejo final acerca del uso del enfoque top-down para comprender un problema es simplemente: “No lo use”.

### 3 Enfoque Middle-Out

La propuesta middle-out está basada en la simple idea de aprovechar al máximo el conocimiento del problema bajo estudio, antes de tomar una buena decisión. Este objetivo es fácil de enunciar pero no de satisfacer. Nuestra propuesta se basa en el uso de dos modelos: el Léxico Extendido del Lenguaje (LEL) y Escenarios, los cuales se describen brevemente en los siguientes párrafos.

#### 3.1 Léxico Extendido del Lenguaje (LEL)

La propuesta del léxico es conocer el vocabulario de la aplicación y su semántica, dejando para un siguiente paso la comprensión del problema (Franco y Leite 1992).

El LEL está compuesto por un conjunto de símbolos que representan el lenguaje de la aplicación. Los símbolos son, en general, palabras o frases que el cliente/usuario repite o enfatiza, también se incluyen aquellas palabras o frases relevantes para el dominio más allá de su frecuencia de repetición. Mientras se obtienen los símbolos, el ingeniero comprende el significado de los mismos, identifica el nombre o nombres (en el caso de sinónimos) de cada uno y describe la noción y su impacto. La *noción* define "lo que el símbolo es" y el *impacto* describe "cómo el símbolo repercute en el sistema" (Leite y Franco 1990). Mientras se describen los símbolos, dos principios deben cumplirse: el *principio de circularidad*, que tiende a maximizar el uso de símbolos en la descripción de otros símbolos y el *principio de vocabulario mínimo* que minimiza el uso de símbolos que son externos al vocabulario de la aplicación.

Los símbolos obtenidos se clasifican en Sujeto, Objeto, Verbo y Estado, indicando para cada tipo el contenido de las nociones y de los impactos. Esto facilita la descripción, integridad y homogeneidad del LEL obtenido.

#### 3.2 Escenarios

Nuestro modelo de escenarios (Leite et al. 1997) es una estructura compuesta de *título, objetivo, contexto, recursos, actores, episodios* y *excepciones*. Los recursos y los actores son enumeraciones; el título, el objetivo, el contexto y las excepciones son sentencias declarativas, mientras que los episodios son un conjunto de sentencias expresadas en un lenguaje simple, que muestran descripciones operacionales de comportamiento.

La Figura 1 muestra el modelo de escenarios basado en lenguaje natural. Se observa un elemento importante en el contexto, los recursos y los episodios que son las *restricciones*, las cuales muestran aspectos referidos a requisitos no funcionales. También se debe señalar que los episodios pueden expresarse en sí mismos como un escenario, posibilitando *relaciones de jerarquía* entre escenarios y subescenarios y *relaciones de integración* de escenarios.

Así mismo, los episodios se clasifican en: episodios simples, episodios condicionales y episodios opcionales. Por otro lado, el modelo de escenarios permite la descripción de *comportamientos secuenciales* y *no secuenciales* a través del agrupamiento de episodios.

#### 3.3 Derivación e Integración de Escenarios

Como ejemplo de la propuesta middle-out, se presenta la derivación de escenarios desde el Léxico Extendido del Lenguaje descripta en (Hadad et al. 1997) que ha sido aplicada a diferentes casos de estudio, los que se detallan en la próxima sección. La derivación de escenarios parte de la información elicitada durante el proceso de construcción del LEL y genera los escenarios retornando al Universo de Discurso<sup>2</sup> (UdeD) cuando es necesario. Este proceso no es ni top-down ni bottom-up. Un verdadero proceso bottom-up comenzaría descubriendo episodios, luego armando subescenarios y finalmente integrándolos en escenarios. Mientras un proceso top-down comenzaría armando uno o muy pocos escenarios representando todo el sistema, refinándolos más tarde para obtener una sucesión de escenarios con un creciente nivel de detalle, y eventualmente un creciente número de escenarios.

El proceso de derivación de Escenarios desde el LEL comienza con la construcción del LEL que no mira específicamente a todo el sistema ni ahonda en cada detalle de la funcionalidad del problema, por el contrario avanza y retrocede de las ideas generales al conocimiento detallado solamente para comprender el vocabulario usado en el UdeD. El proceso continúa con la construcción de escenarios siguiendo las heurísticas que extraen información del LEL y se obtienen escenarios cuyo nivel de detalle es mayor que el encontrado en el LEL. La información adicional se obtiene principalmente del UdeD. Durante este paso algunas partes comunes de diferentes escenarios se factorizan para crear subescenarios. Los subescenarios también son generados cuando uno o más episodios de un escenario merecen un tratamiento independiente. Este último paso muestra un comportamiento descendente, así esta parte del proceso puede verse como top-down. En este punto, surge como una fuerte necesidad de los ingenieros tener una visión global de los escenarios y un nuevo paso, ahora bottom-up, es requerido. En este paso se construyen los escenarios integradores, mencionando en sus episodios situaciones concretas descritas por los escenarios obtenidos previamente.

### 3.3.1 Proceso de Derivación de Escenarios

Este proceso comienza con la construcción del LEL, luego se aplica al mismo una heurística de derivación para generar escenarios y se retorna al UdeD para completar los escenarios.

Los aspectos principales de la heurística de este proceso se mencionan a continuación (Hadad et al. 1997):

1. Identificar los actores.
  - 1.1. Buscar en el LEL símbolos del tipo Sujeto para actores.
  - 1.2. Clasificar en actores principales y actores secundarios.
2. Generar la lista de escenarios candidatos.
  - 2.1. Incluir los impactos de cada actor principal como un potencial

---

<sup>2</sup> El contexto total en el cual el software se desarrollará y operará. El UdeD incluye todas las fuentes de información y toda la gente relacionada con el software. Es la realidad acotada por el conjunto de objetivos establecidos por aquéllos que demandan una solución de software (Leite et al. 1997).

- escenario en la lista de escenarios candidatos.
- 2.2. Ampliar la lista con los impactos de cada actor secundario.
- 3. Describir los escenarios candidatos a partir de información del LEL.
  - 3.1. Buscar en el LEL símbolos del tipo Verbo incluidos en el título del escenario candidato.
  - 3.2. Usar estos símbolos como principal fuente para describir los componentes de cada escenario.
- 4. Completar los escenarios con información proveniente del UdeD.
- 5. Analizar los escenarios.
  - 5.1. Verificar los escenarios, individualmente y en conjunto.
  - 5.2. Validar los escenarios con los clientes/usuarios.

### **3.3.2 Proceso de Construcción de Escenarios Integradores**

Una vez descriptos todos los escenarios que representan situaciones del problema en estudio, éstos son verificados y luego validados con los clientes/usuarios. Es en este momento cuando se requiere una visión global del problema en estudio y surge la necesidad de generar escenarios integradores.

El proceso de construcción de Escenarios Integradores consiste en:

1. Construir jerarquías de escenarios, considerando la existencia de escenarios y subescenarios.
2. Agrupar escenarios pertenecientes al nivel más alto de la jerarquía utilizando información del contexto y del objetivo, y generar una lista candidata de grupos de escenarios.
3. Analizar cada grupo de escenarios y generar la lista final de grupos que representarán cada uno un potencial escenario integrador. Tener en cuenta que un mismo escenario puede pertenecer a más de un grupo.

<p><b>Escenario:</b> descripción de una situación que ocurre en el contexto del problema.  Sintaxis: Título+ Objetivo+ Contexto+ {Recursos}<sub>1</sub><sup>N</sup>+{Actores}<sub>1</sub><sup>N</sup>+{Episodios}<sub>2</sub><sup>N</sup>+{Excepciones}</p> <p><b>Título:</b> identificación del escenario. En el caso de un subescenario, el título es el mismo que la sentencia episodio, sin las restricciones.  Sintaxis: Frase   ([Actor   Recurso] + Verbo + Predicado)</p> <p><b>Objetivo:</b> finalidad a ser alcanzada. El escenario describe la forma de lograr el objetivo.  Sintaxis: [Actor   Recurso] + Verbo + Predicado</p> <p><b>Contexto:</b> compuesto por al menos uno de los siguientes ítems: Ubicación Geográfica: lugar físico donde se produce el escenario; Ubicación Temporal: especificación de tiempo para el desarrollo del escenario; Precondición: estado inicial del escenario.  Sintaxis: {Ubicación Geográfica} + {Ubicación Temporal} + {Precondición}  donde Ubicación Geográfica es: Frase + {Restricción}  donde Ubicación Temporal es: Frase + {Restricción}  donde Precondición es: [Sujeto   Actor   Recurso] + Verbo + Predicado + {Restricción}</p> <p><b>Recursos:</b> elementos físicos o información relevantes que deben estar disponibles en el escenario.  Sintaxis: Nombre + {Restricción}</p> <p><b>Actores:</b> personas o estructuras organizacionales que tienen un rol en el escenario.  Sintaxis: Nombre</p> <p><b>Episodios:</b> conjunto de acciones que detallan al escenario y proveen su comportamiento. Un episodio también puede ser descrito como un escenario.  Sintaxis (usando BNF parcial):  &lt;episodios&gt; ::= &lt;serie grupo&gt;   &lt;serie episodios&gt;  &lt;serie grupo&gt; ::= &lt;grupo&gt; &lt;grupo&gt;   &lt; grupo no secuencial&gt;   &lt;serie grupo&gt; &lt;grupo&gt;  &lt;grupo&gt; ::= &lt;grupo secuencial&gt;   &lt; grupo no secuencial&gt;  &lt;grupo secuencial&gt; ::= &lt;sentencia básica&gt;   &lt;grupo secuencial &gt; &lt;sentencia básica&gt;  &lt;grupo no secuencial&gt; ::= # &lt;serie episodios&gt; #  &lt;serie episodios&gt; ::= &lt;sentencia básica&gt; &lt;sentencia básica&gt;   &lt;serie episodios&gt; &lt;sentencia básica&gt;  &lt;sentencia básica&gt; ::= &lt;sentencia simple&gt;   &lt;sentencia condicional&gt;   &lt;sentencia optativa&gt;  &lt;sentencia simple&gt; ::= &lt;sentencia episodio&gt; CR  &lt;sentencia condicional&gt; ::= Si &lt;condición&gt; &lt;conector&gt; &lt;sentencia episodio&gt; CR  &lt;conector&gt; ::= <b>entonces</b>   ,  &lt;sentencia optativa&gt; ::= [ &lt;sentencia episodio&gt; ] CR  donde &lt;sentencia episodio&gt; se describe:  ((([Actor   Recurso] + Verbo + Predicado)([Actor   Recurso] + [Verbo] + Título)) + {Restricción})</p> <p><b>Excepciones:</b> generalmente reflejan la falta o mal funcionamiento de un recurso. Una excepción impide el cumplimiento del objetivo del escenario. Eventualmente se puede indicar el tratamiento de la excepción a través de un escenario.  Sintaxis: Causa [(Solución)]  donde Causa es: Frase   ([Sujeto   Actor   Recurso] + Verbo + Predicado)  donde Solución es: Título</p> <p><b>Restricción:</b> es un requerimiento de calidad o alcance referido a una dada entidad. Es un atributo de Recurso, Episodio o de los subcomponentes del Contexto.  Sintaxis: ([Sujeto   Actor   Recurso] + [No] Debe + Verbo + Predicado)   Frase</p>
---

Fig. 1 - Modelo de Escenarios

4. Para cada grupo, establecer las relaciones de orden (tanto secuenciales como de paralelismo) entre los escenarios involucrados, teniendo en cuenta las precondiciones de cada escenario. Precondiciones similares pueden implicar escenarios no secuenciales. Convertir los escenarios en

- episodios del escenario integrador para cada grupo.
5. Por cada escenario integrador identificado:
    - 5.1. Definir un nombre significativo para el título del escenario integrador.
    - 5.2. Definir el objetivo analizando los objetivos de cada escenario del grupo, en concordancia con el título dado.
    - 5.3. Combinar las ubicaciones geográficas de cada escenario en el grupo para conformar la ubicación geográfica del escenario integrador.
    - 5.4. Analizar las ubicaciones temporales de cada escenario en el grupo para armar la ubicación temporal del escenario integrador.
    - 5.5. Usar la precondición del escenario listado como primer episodio del escenario integrador como precondición del escenario integrador. Analizar las precondiciones de los restantes escenarios del grupo e incorporar aquellas no cubiertas por el primer episodio siempre que no sean precondiciones internas al grupo.
    - 5.6. Identificar los actores del escenario integrador como la unión de los actores de cada escenario en el grupo.
    - 5.7. Identificar los recursos del escenario integrador como la unión de los recursos de cada escenario en el grupo, eliminando aquellos generados dentro de estos escenarios.
    - 5.8. Incorporar las restricciones de cada escenario del grupo como restricciones referidas al episodio que lo representa.
    - 5.9. Incorporar las excepciones de cada escenario del grupo como excepciones del escenario integrador mencionando los episodios que pueden dispararlas.
  6. Observar la existencia de escenarios “aislados” no utilizados en ningún escenario integrador, ya sea como episodio o como excepción. Analizar su contenido para determinar si corresponde a una excepción de algún escenario integrador o si está fuera del dominio de la aplicación.

#### **4 Casos de Estudio Desarrollados**

A continuación se presentan las experiencias recogidas por los autores, que justifican los beneficios del enfoque middle-out. Los casos de estudio se componen principalmente por:

1. Sistema Nacional para la Emisión de Pasaportes<sup>3</sup> (Leite et al. 1997) (Leite et al. 1996). Se obtuvo un conjunto de 24 escenarios desarrollados por un equipo de investigadores. Este sistema está centralizado en el Departamento de la Policía Federal que es la única organización autorizada para la emisión de pasaportes en todo el país. La complejidad del sistema surge de los estrictos controles requeridos y del gran volumen de información.
2. Sistema de Agenda de Reuniones<sup>4</sup> (Leite et al. 1997) (Hadad et al. 1998).

---

<sup>3</sup> Desarrollado por la Universidad de Belgrano.

<sup>4</sup> Desarrollado por la Universidad de Belgrano.

Tres conjuntos de escenarios fueron generados por dos grupos de estudiantes graduados y uno por un equipo de investigadores. Se utilizó la Universidad de Belgrano como organización destino, basándose en el caso de estudio propuesto por van Lamsweerde et al. (1993).

3. Sistema de Plan de Ahorro para la Adquisición de Automóviles<sup>5</sup> (Rivero et al. 1998) (Mauco et al. 1997). Nueve conjuntos de escenarios fueron generados por siete grupos de estudiantes y por dos equipos de miembros de la facultad. El objetivo del sistema es administrar los planes para la adquisición de vehículos 0km. Se constituye un grupo de personas físicas o legales que participan en un proceso de adjudicación organizado por el Administrador General para la entrega de un vehículo.
4. Sistema para el Control de Pos Graduación<sup>6</sup> (Breitman et al. 1997). Cinco conjuntos de escenarios fueron generados por grupos de estudiantes graduados. Este sistema fue modelado y desarrollado para el Departamento de Informática de PUC-Rio y maneja todo el funcionamiento concerniente a los estudiantes postgraduados.
5. Sistema de Biblioteca<sup>7</sup>. Cuatro conjuntos de escenarios fueron generados por grupos de estudiantes graduados. Este sistema consiste básicamente en el registro de libros y de usuarios, y en la verificación de préstamos de libros.
6. Sistema Editor de Texto<sup>8</sup>. Ocho conjuntos de escenarios fueron desarrollados por estudiantes avanzados.

Algunos de estos conjuntos de escenarios se desarrollaron intuitivamente mientras que otros siguieron las heurísticas descritas en la sección 3.3.

#### **4.1 Experiencia con desarrollos intuitivos**

Cuando el conjunto de escenarios se desarrolló intuitivamente, había un sentimiento común en todos los desarrolladores en relación con la necesidad de tener una apreciación global del problema, materializado en uno o más "Escenarios Generales" u otro modelo que ayudase a tener una primera idea del macrosistema. No siempre estos Escenarios Generales o representaciones equivalentes fueron construidas. Frecuentemente, cuando aún no se había escrito nada, los desarrolladores informaron que necesitaban por lo menos de un cuadro en su mente para enfrentar el "síndrome de la página en blanco". Esto es especialmente verdadero cuando:

- El equipo que desarrolla el conjunto de escenarios es diferente del que desarrolla el LEL.
- Existe una falta de información previa sobre el macrosistema.

Como ejemplo podemos decir que con el Sistema Nacional para la Emisión de Pasaportes se creó, al principio, cierto grado de confusión en el grupo de ingenieros.

---

<sup>5</sup> Desarrollado por la Universidad Nacional del Centro de la Provincia de Buenos Aires.

<sup>6</sup> Desarrollado por la Pontificia Universidade Católica do Rio de Janeiro.

<sup>7</sup> Desarrollado por la Pontificia Universidade Católica do Rio de Janeiro.

<sup>8</sup> Desarrollado por la Pontificia Universidade Católica do Rio de Janeiro.

Por lo tanto, en ese momento se determinó la conveniencia de un enfoque top-down. Al construirse los Escenarios Generales, se volvió evidente que éstos necesitaban un importante esfuerzo de mantenimiento, dado que al escribir los escenarios, se debió propagar los cambios a los Escenarios Generales para mantener la coherencia. Esto provocó una importante fuente de errores. Aún cuando se dedicó especial atención a los Escenarios Generales al llegar al final del proceso, éstos presentaban una cierta falta de precisión debido a la presencia de palabras colectivas representando acciones, actores y recursos. En este punto se detectaron los inconvenientes de un enfoque top-down dado que se requirió mucho trabajo para obtener un producto mediocre.

Título:	Solicitar un nuevo <u>pasaporte</u> .
Objetivo:	Entregar un <u>pasaporte</u> a un <u>solicitante</u> por primera vez.
Contexto:	Ubicación Geográfica: Departamento de Policía. Precondición: El <u>solicitante</u> nunca ha pedido un <u>pasaporte</u> .
Actores:	<u>Solicitante</u> Empleados del Departamento de Policía
Recursos:	<u>Formulario</u> <u>Documentación del Solicitante</u> <u>Pasaporte vacío</u> Fotografías del <u>Solicitante</u>
Episodios:	<ol style="list-style-type: none"><li>1. El <u>solicitante</u> completa el <u>formulario</u>.</li><li>2. La <u>División Documentos y Certificados</u> verifica el <u>formulario</u> y la <u>documentación del Solicitante</u>.</li><li>3. El fotógrafo toma las fotografías al <u>solicitante</u>.</li><li>4. El <u>solicitante</u> paga la tarifa del <u>pasaporte</u>.</li><li>5. El empleado de la <u>Cabina de Huellas Digitales</u> toma las huellas digitales del <u>solicitante</u>.</li><li>6. La <u>División Documentos y Certificados</u> prepara el nuevo <u>pasaporte</u>.</li><li>7. Verificación interna de los datos del <u>solicitante</u>.</li><li>8. El <u>solicitante</u> recibe el <u>pasaporte</u>.</li></ol>

Fig. 2 - Ejemplo de un Escenario General

Título:	Solicitar un nuevo <u>pasaporte</u> .
Objetivo:	Entregar un <u>pasaporte</u> a un <u>solicitante</u> por primera vez.
Contexto:	Ubicación geográfica: <u>División Documentos y Certificados</u> , <u>División Índice General</u> , <u>División Dactiloscopia</u> , <u>División Legajos y Antecedentes</u> . Precondición: El <u>solicitante</u> nunca ha pedido un <u>pasaporte</u> .
Actores:	<u>Solicitante</u> Cajero Empleado de la <u>Cabina Huellas Digitales</u> Fotógrafo Empleado de la <u>Cabina de Control de Documentación</u> Empleado de la <u>Cabina de Recepción</u> Empleado de la <u>División General de Índices</u> Empleado de la <u>División Dactiloscopia</u> Empleado de la <u>División Legajos y Antecedentes</u> Empleado de la <u>Cabina de Entrega de Pasaportes</u>
Recursos:	<u>Formulario</u> <u>Documentación del Solicitante</u> <u>Pasaporte vacío</u> <u>Fotografías del Solicitante</u> <u>Cámara fotográfica</u> <u>Máquina de estampillado</u> <u>Ficha dactiloscópica</u> <u>Talón</u> <u>Legajo del Solicitante</u> <u>Libro de control de numeración de <u>pasaporte</u></u>
Episodios:	<ol style="list-style-type: none"> <li>1. ENTREGA Y LLENADO DE <u>FORMULARIO</u>.</li> <li>2. CONTROL DE <u>DOCUMENTACIÓN</u>.</li> <li>3. #SACAR FOTOGRAFÍA.</li> <li>4. PAGAR TRÁMITE.</li> <li>5. Si el <u>solicitante</u> tiene un <u>número de identificación</u>, <b>Entonces</b> OBTENCIÓN DE HUELLAS DIGITALES PARA CONTROL.</li> <li>6. Si el <u>solicitante</u> no tiene un <u>número de identificación</u>, <b>Entonces</b> OBTENCIÓN DE HUELLAS DIGITALES PARA <u>LEGAJO</u>. #</li> <li>7. DERIVACIÓN A <u>CABINA DE RECEPCIÓN</u>.</li> <li>8. RECEPCIÓN Y ARMADO DE <u>PASAPORTE ORIGINAL</u>.</li> <li>9. CONTROL DE <u>PRONTUARIO</u>.</li> <li>10. CONTROL DACTILOSCÓPICO.</li> <li>11. VERIFICACIÓN FINAL Y CERTIFICACIÓN DE <u>PASAPORTE</u>.</li> <li>12. ENTREGA DE <u>PASAPORTE</u>.</li> <li>13. ENVÍO DEL <u>LEGAJO DEL SOLICITANTE</u> AL ARCHIVO GENERAL DE <u>LEGAJOS</u>.</li> </ol>
Excepciones:	El <u>solicitante</u> no retira el <u>pasaporte</u> en el período previsto, en episodio 12. (ENVÍO A INCINERACIÓN DE <u>PASAPORTES NO RETIRADOS</u> ). Un <u>pasaporte</u> es observado, en episodio 9 o 10. (DERIVACIÓN DE <u>PASAPORTES OBSERVADOS</u> ).

Fig. 3 - Ejemplo de un Escenario Integrador

#### 4.2 Experiencia con desarrollos basados en heurísticas

Cuando el conjunto de escenarios se desarrolló siguiendo las heurísticas descritas en la sección 3.3, no fue necesario tener una vista global del macrosistema al comienzo del proceso y los escenarios se desarrollaron sin problemas, pero cuando el número de escenarios aumentó los ingenieros perdieron el cuadro completo del conjunto de escenarios. La falta de una visión global es similar a la que dio origen a la creación de los Escenarios Generales, pero el conocimiento del macrosistema a esta altura es notoriamente mejor. En algunos de los casos de estudio, se construyeron Escenarios Integradores, los que agruparon varios escenarios dispersos. Usando un enfoque bottom-up, la construcción de los Escenarios Integradores se realizó rápidamente sin requerir mantenimiento. Estos escenarios carecieron de la falta de precisión que se observó en los Escenarios Generales ya que el vocabulario utilizado fue más preciso y los episodios correspondían a escenarios ya descriptos.

#### 4.3 Comparación entre los enfoques top-down y middle-out

La ventaja de los Escenarios Integradores sobre los Escenarios Generales fue confusa en el inicio y se entendió pobremente hasta que se llevó a cabo un proceso de Inspección de Escenarios (Doorn et al. 1998) aplicado a varios casos de la lista mencionada. En dicho proceso, el beneficio de tener Escenarios Integradores en lugar de Escenarios Generales se volvió evidente ya que los primeros son una importante fuente de verificación y permiten fácilmente detectar superposiciones y omisiones entre escenarios.

ESCENARIO	GENERAL	INTEGRADOR
Enfoque	TOP-DOWN	MIDDLE-OUT
Construcción	AL COMIENZO	AL FINAL
Necesidad de Mantenimiento	SI	NO
Vista total del Macrosistema	APROXIMADA	CASI REAL
Vista de las relaciones entre escenarios	POBRE	CLARA
Permite la división del problema	SI	NO
Permite verificar la consistencia	SOLAMENTE SI FUE REFINADO	SI
Conocimiento previo del macrosistema	SI	NO

Tabla 1 – Tabla Comparativa de la Construcción de Escenarios Generales e Integradores

Las Figuras 2 y 3 muestran respectivamente ejemplos de un Escenario General y un Escenario Integrador. Ambos ejemplos se basan en el caso de estudio Sistema

Nacional para la Emisión de Pasaportes.

Se muestra en la Tabla 1 las diferencias entre los Escenarios Generales y los Escenarios Integradores en el proceso de construcción de escenarios.

En un enfoque top-down los Escenarios Generales se describen al comienzo del proceso de construcción de escenarios partiendo de un conocimiento escaso y general del macrosistema. Por otro lado, en un enfoque middle-out, los Escenarios Integradores se describen como último paso del proceso de construcción, habiendo adquirido entonces un conocimiento amplio y detallado del macrosistema.

Los Escenarios Generales requieren de un mantenimiento constante durante todo el proceso de construcción para que los mismos sean consistentes con el resto de los escenarios, de lo contrario perderán su utilidad y deberán ser desechados. En contraste, los Escenarios Integradores por ser construidos al final del proceso son naturalmente consistentes con el resto de los escenarios, y tendrán utilidad a lo largo del ciclo de desarrollo del software. Nuestros escenarios son del tipo “escenarios persistentes” según la faceta referida a lapso de vida (Vista Ciclo de Vida) en la clasificación propuesta por CREWS en (Rolland et al. 1998).

Los Escenarios Generales dan una imagen aproximada del macrosistema al inicio del proceso, y con un gran esfuerzo de mantenimiento se logra una vista semejante a la que proveen los Escenarios Integradores. Este esfuerzo está directamente relacionado con la posibilidad de verificar la consistencia entre escenarios.

Los Escenarios Integradores muestran claramente a través de sus episodios las relaciones entre los escenarios, mientras que los episodios de los Escenarios Generales muestran actividades que luego podrán o no convertirse en escenarios.

En la práctica (Weidenhaupt et al. 1998), los Escenarios Generales se utilizan para dividir tareas, aún no bien conocidas, entre el equipo de ingenieros. Esto no es posible y tampoco deseable con los Escenarios Integradores dado el momento de su construcción.

## **5 Conclusiones**

Considerando varios de nuestros casos de estudio con escenarios, hemos detectado la ventaja de usar escenarios integradores frente a escenarios generales, teniendo como fuerte evidencia que la construcción de escenarios no debería usar una descomposición top-down ni tampoco es deseable una construcción estrictamente bottom-up. Nuestra estrategia para la construcción de escenarios por medio del enfoque middle-out produce un conjunto de escenarios que están naturalmente integrados. Muchas veces la cantidad de escenarios y su complejidad requieren que el conjunto de escenarios se organice como un modelo, aplicando para ello una estrategia de integración de escenarios.

Consideramos que nuestra contribución a la investigación en escenarios es develar algunos de los aspectos del proceso de construcción de escenarios. No cubrimos todos los aspectos de este proceso sino que enfatizamos en la organización del conjunto de escenarios como un modelo. Tratamos explícitamente con la composición/descomposición de escenarios tanto como una forma natural como una forma artificial de organizar el conjunto de escenarios. Aunque nuestro trabajo (Leite et al. 1997) está influenciado por nuestro propio modelo de representación (Figura 1), creemos que el aspecto de composición/descomposición es de interés general para la

construcción de escenarios. Vemos el tema de composición/descomposición relacionado con la faceta de abstracción<sup>9</sup> (Vista de Contenidos) propuesta por CREWS (Rolland et al. 1998), pero allí sólo se hace una distinción entre “escenario ejemplificado” y “escenario tipo”.

Destacamos que nuestros resultados se basan en experiencias reales con escenarios. Creemos firmemente que este debate sobre estrategias para la construcción de escenarios es importante para la investigación tanto como para su aplicación práctica.

## Referencias

- Booch, G., “Object Oriented Design with Applications”, The Benjamin Cumming Publishing Company, Inc., Redwood City (1991).
- Booch, G., “Scenarios”, Report on Object Analysis and Design, Vol.1 No.3 (1994), pp. 3-6.
- Breitman, K.K, Leite, J.C.S.P., "A Framework for Scenario Evolution", Proceedings of the IEEE Int. Conf. on Requirements Engineering, IEEE Computer Society Press (1998), pp. 214-221.
- Carroll, J., “Introduction: The Scenario Perspective on System Development”, Scenario-Based Design: Envisioning Work and Technology in System Development, J. Carroll, ed., John Wiley & Sons, New York (1995), pp. 1-18.
- Constantine, L., “Joint Essential Modeling, User Requirements Modeling for Usability”, International Conference on Requirements Engineering, Colorado Springs, Constantine & Lockwood, Ltd. (1998).
- Dano, B., Briand, H., Barbier, F., “An Approach Based on the Concept of Use Case to Produce Dynamic Object-Oriented Specifications”, Proceedings of the Third IEEE International Symposium on Requirements Engineering, IEEE Computer Society Press, Los Alamitos, California (1997), pp. 54-64.
- Dorn, J., Kaplan, G., Hadad, G., Leite, J.C.S.P., “Inspección de Escenarios”, Anais do WER'98 Workshop em Engenharia do Requisitos”, Maringá, Brasil (1998), pp. 57-69.
- Firesmith, D. G, “Modeling the Dynamic Behavior of Systems, Mechanisms, and Classes with Scenarios”, Report on Object Analysis and Design, Vol.1, No.2 (1994), pp. 32-36.
- Franco, A.P.M., Leite, J.C.S.P., “Uma estratégia de Suporte a Engenharia de Requisitos”, Anais do XIX Seminario Integrado de Software y Hardware, Rio de Janeiro (1992), pp.200-213.
- Gough, P. A., Fodemski, F. T., Higgins, S. A., Ray, S. J., “Scenarios - An Industrial Case Study and Hypermedia Enhancements”, RE95: Proceedings of the International Symposium on Requirements Engineering, IEEE Computer Society Press, Los Alamitos, California, 1995, pp. 10-17.
- Hadad, G., Kaplan, G., Leite, J.C.S.P., “Léxico extendido del lenguaje y escenarios del meeting scheduler”, Technical Report # 13, Departamento de Investigación, Universidad de Belgrano, Buenos Aires (1998).
- Hadad, G., Kaplan, G., Oliveros, A., Leite, J.C.S.P., “Construcción de Escenarios a partir del Léxico Extendido del Lenguaje”, 26 JAIIO, Buenos Aires (1997), pp. 65-77.
- Hsia, P., Samuel, J., Gao, J. Kung, D., Toyosima, Y., Chen, C., “Formal Approach to Scenario Analysis”, IEEE Software, V.11, No.2 (1994), pp. 33-41.
- Jackson, M., “Software Requirements & Specifications. A lexicon of practice, principles and prejudices”, Addison Wesley, ACM Press (1995).
- Jacobson, I., “Basic Use-Case Modeling”, Report on Object Analysis and Design, Vol.1, No.2 (1994a), pp. 15-19.

---

<sup>9</sup> Este aspecto nos fue señalado por Colette Rolland.

- Jacobson, I., "Basic Use-Case Modeling (Continued)", Report on Object Analysis and Design, Vol.1, No.3 (1994b), pp. 7-9.
- Jacobson, I., Christerson, M., Jonsson, P., Overgaard, G., "Object-Oriented Software Engineering - A Use Case Driven Approach", Reading, MA: Addison Wesley, New York: ACM Press (1992).
- Leite, J.C.S.P., Franco, A.P.M., "O Uso de Hipertexto na Elicitação de Linguagens da Aplicação", Anais de IV Simpósio Brasileiro de Engenharia de Software, SBC (1990), pp. 134-149.
- Leite, J.C.S.P., Oliveros, A., Rossi, G., Balaguer, F., Hadad, G., Kaplan, G., Maiorana, V., "Léxico extendido del lenguaje y escenarios del sistema nacional para la obtención de pasaportes", Technical Report # 7, Departamento de Investigación, Universidad de Belgrano, Buenos Aires (1996).
- Leite, J.C.S.P., Rossi, G., Balaguer, F., Maiorana, V., Kaplan, G., Hadad, G., Oliveros, A., "Enhancing a Requirements Baseline with Scenarios", Requirements Engineering Journal, Vol.2, No. 4 (1997), pp 184-198.
- Mauco, V., Ridao M., del Fresno, M., Rivero, L., Doorn, J., "Ingeniería de Requisitos, Proyecto: Sistema de Planes de Ahorro", Technical Report, ISISTAN, UNCPBA, Tandil (1997).
- Oberg, R., Probasco, L., Ericsson, M., "Applying Requirements Management with Use Cases", Rational Software Corporation (1998).
- Potts, C., "Using Schematic Scenarios to Understand User Needs", Proceedings of DIS'95 - Symposium on Designing Interactive Systems: Processes, Practices and Techniques, ACM Press, University of Michigan (1995), pp. 247-256.
- Potts, C., Takahashi, K., Antón, A. Y., "Inquiry-Based Requirements Analysis", IEEE Software, Vol. 11, No.2 (1994), pp.21-32.
- Rivero, L., Doorn, J., Del Fresno, M., Mauco, V., Ridao, M., Leonardi, C., "Una Estrategia de Análisis Orientada a Objetos basada en Escenarios: Aplicación en un Caso Real", Anais do WER'98 - Workshop em Engenharia do Requisitos, Maringá, Brasil (1998), pp. 79-90.
- Robertson, S.P., "Generating Object-Oriented Design Representations via Scenario Queries", Scenario-Based Design: Envisioning Work and Technology in System Development, J. Carroll, ed., John Wiley & Sons, New York (1995), pp. 279-306.
- Rolland, C., Ben Achour, C., Cauvet, C., Ralyté, J., Sutcliffe, A., Maiden, M., Jarke, M., Haumer, P., Pohl, K., Dubois, E., Heymans, P., "A Proposal for a Scenario Classification Framework", Requirements Engineering Journal, Vol.3, No.1 (1998), pp. 23-47.
- Schneider, G., Winters, J., "Applying Use Cases, A Practical Guide", Addison-Wesley (1998).
- Sutcliffe, A., "Workshop. Exploring Scenarios in Requirements Engineering", Proceedings of the Third IEEE International Symposium on Requirements Engineering, IEEE Computer Society Press, Los Alamitos, Colorado (1997a), pp. 180-182.
- Sutcliffe, A., "A Technique Combination Approach to Requirements Engineering", Proceedings of the Third IEEE International Symposium on Requirements Engineering, IEEE Computer Society Press, Los Alamitos, Colorado (1997b), pp. 65-74.
- van Lamsweerde, A., Darimont, R., Massonet, Ph., "The Meeting Scheduler System - Preliminary Definition", Internal Report, Université Catholique de Louvain (1993).
- Weidenhaupt, K., Pohl, K., Jarke, M., Haumer, P., "Scenarios in System Development: Current Practice", IEEE Software (1998), pp. 34-45.
- Wirfs-Brock, R., "Designing Objects and Their Interactions: A Brief Look at Responsibility-Driven Design", Scenario-Based Design: Envisioning Work and Technology in System Development, J. Carroll, ed., John Wiley & Sons, New York (1995), pp. 337-359.
- Zorman, L., "Requirements Envisaging by Utilizing Scenarios (Rebus)", Ph.D. Dissertation, University of Southern California (1995).