

## Propuesta de un Catálogo de Patrones de Escenario para la Definición de Requisitos

Carlos Alberto Cortés Bravo  
Instituto Tecnológico de Orizaba  
[cortesbravocarlos@gmail.com](mailto:cortesbravocarlos@gmail.com)

María Antonieta Abud Figueroa  
Instituto Tecnológico de Orizaba  
[mabud@ito-depi.edu.mx](mailto:mabud@ito-depi.edu.mx)

Celia Romero Torres  
Instituto Tecnológico de Orizaba  
[cromero@ito-depi.edu.mx](mailto:cromero@ito-depi.edu.mx)

Celia Romero Torres  
Instituto Tecnológico de Orizaba  
[ujuarez@ito-depi.edu.mx](mailto:ujuarez@ito-depi.edu.mx)

Gustavo Pelaez Camarena  
Instituto Tecnológico de Orizaba  
[gpelaez@ito-depi.edu.mx](mailto:gpelaez@ito-depi.edu.mx)

**Resumen:** La definición de los requisitos de software consiste en la especificación y validación de las funcionalidades que el sistema a desarrollar debe proporcionar, así como de las restricciones que el sistema debe cumplir. Para el análisis de los requisitos existen varias técnicas, siendo los escenarios una técnica que permite describir el comportamiento esperado del software en un lenguaje reconocido por los involucrados con lo que se logra una buena comunicación con el cliente. Por otra parte, los patrones permiten reutilizar elementos previamente comprobados funcionando como un registro de experiencias que ayudan a agilizar y mejorar los procesos. En este artículo se presenta una propuesta para la definición de patrones de escenario y un catálogo de patrones de escenario en el dominio de la telemática que permiten una descripción clara y precisa de los requisitos en este dominio.

**Palabras clave:** patrones, escenarios, requisitos de software.

## Catalogue for Scenario Patterns Requirements Elicitation

**Abstract:** Software requirements elicitation is the process of identifying needs for a software systems as well as the constraints that the system must meet. There are several techniques for the analysis of the requirements, the scenarios being a technique to describe the expected behavior of the software in terms understood by all stakeholders with what a good customer communication is achieved. Moreover, the patterns allow reuse previously tested elements functioning as a record of experiences that help streamline and improve processes. This article presents a proposal for defining patterns of scenarios and a catalog of patterns in the domain of telematics that enable clear and precise descriptions of the requirements in this domain.

**Keywords:** patterns, scenarios, software requirements.

## 1. Introducción

Un punto prioritario en el desarrollo de software es la definición de lo que se quiere producir, ya que el desarrollo del software inicia hasta el momento en el que se tienen establecidas a detalle las características del software que se va a desarrollar,

situación que es más crítica cuando se trata de sistemas complejos, por lo cual es necesario conocer distintas técnicas de recopilación y definición de requisitos y aplicar la que ofrezca mejores resultados.

La ingeniería de requisitos es el área de investigación que atiende este punto fundamental en el proceso de producción, proponiendo métodos, técnicas y herramientas que facilitan el trabajo de definición de lo que se quiere de un producto de software. Su objetivo es aumentar el conocimiento del dominio del problema para así representar las necesidades de un cliente en función de una aplicación de software de una manera adecuada y entendible tanto para los usuarios finales como para el equipo de desarrollo.

Es de suma importancia asegurar una buena comunicación con los clientes y/o usuarios, ya que de ello depende un futuro éxito o fracaso, debido a que a partir de que se obtienen y se definen los requisitos expresados por el cliente, se comienza con las demás etapas en el proceso de desarrollo y es de vital importancia comenzar el desarrollo de software a partir de una lista de requisitos concisos, completos, consistentes, no ambiguos y verificables.

Existen diversas técnicas para documentar los requisitos, como son casos de uso, prototipos, puntos de vista, escenarios, entre otras. De éstas se reporta como una de las más exitosas los escenarios, ya que éstos permiten mantener gran cantidad de información en una forma que los involucrados reconocen (Ridao, Doorn y Sampaio, 2001). Son muchas las definiciones de la palabra escenario (Ridao, et al. 2001; Sutcliffe, 2003; Hadad, Doorn y Kaplan, 1996), una de las cuales es: “un escenario es una descripción parcial y concreta del comportamiento de un sistema en una determinada situación” (Ridao, 2000).

Por otra parte, los patrones permiten, en diferentes áreas del conocimiento, utilizar soluciones previas a un problema en nuevas situaciones, ofreciendo un mecanismo de reutilización de experiencia. En este trabajo, se propone el uso de patrones en la definición de escenarios con el fin de aprovechar las situaciones recurrentes en la definición de requisitos, para lo cual se establece un formato para su definición. En la sección 2 de este artículo se presentan los antecedentes en los que se basa este trabajo, incluyendo una revisión de diferentes propuestas para la representación de

requisitos a través de escenarios; posteriormente en la sección 3 se presenta una propuesta para la definición de patrones de escenarios en la especificación de requisitos y un conjunto de patrones para el área de telemática, detallando uno de ellos. Finalmente se presentan las conclusiones.

## 2. Antecedentes

Los requisitos especifican qué es lo que el sistema debe hacer (funciones) y sus propiedades esenciales y deseables. La captura y el análisis de los mismos son actividades de mucha importancia para proporcionar una visión amplia sobre lo que se pretende resolver con el desarrollo de una aplicación de software

En la fase de definición de requisitos, el ingeniero de requisitos se enfrenta a distintas dificultades (Hadad et al 1996). La elección de una buena técnica para recopilar requisitos, como el uso de un lenguaje común tanto para los desarrolladores como para los clientes y usuarios, es vital para garantizar un buen proceso de definición y validación de requisitos. De igual manera es muy importante comprender el dominio del problema, por lo cual se necesita que el ingeniero de requisitos se vuelva experto en el dominio para así comprender de la manera más amplia las necesidades del cliente, ya que de no ser así es muy probable que se recopilen requisitos incompletos y erróneos.

### 2.1 Lenguaje para Definición de Requisitos

Capturar el dominio del problema es una de las formas mediante las cuales se garantiza una mejor comprensión en la descripción de un escenario, tanto por clientes como por el equipo de desarrollo. El vocabulario del UdeD (Universo de Discurso) es el que refleja las palabras peculiares y más usadas en el mismo (Ridao, et al. 2001). Para la recopilación del conocimiento del UdeD es necesario revisar la documentación, realizar entrevistas o aplicar otras técnicas.

Son diversos los casos donde la construcción de escenarios se basa en el UdeD como en (Ridao, et al. 2001; Ridao, 2000; Doorn, Hadad y Kaplan, 2002), con el propósito de mantener una buena comunicación con el cliente, ya que se logra que

el ingeniero de requisitos maneje un vocabulario entendible tanto por los integrantes del equipo de desarrollo así como por los clientes y/o usuarios. Uno de los métodos que permite mantener un lenguaje homogéneo para ambos es el LEL (Léxico Extendido del Lenguaje), el cual es una representación de los símbolos en el lenguaje del dominio del problema (do Prado Leite, Rossi, Balaguer, Maiorana, Kaplan, Hadad, Oliveros, 1997). Combinar el uso de escenarios y el LEL facilita la comprensión por parte de las personas no especializadas en el desarrollo del software (Hadad, 2010).

Los objetivos del LEL son conocer el vocabulario del usuario y contar con un instrumento simple de trazabilidad (Hadad, et al. 1996). Los elementos para la descripción de un símbolo LEL son: nombre, noción e impacto; el nombre identifica el símbolo del LEL, la noción indica qué es el símbolo y el impacto cómo repercute en el sistema. En la Figura 1 se presenta un ejemplo de un símbolo del LEL que se construye para un caso de estudio de una biblioteca, en la cual se observa que el nombre que se le da está formado por dos palabras siendo estas sinónimos mediante las cuales se identifica el símbolo (Kaplan, Hadad, Doorn y Do Prado Leite, 2000).

#### Obra/Publicación

##### Noción

- Es un ítem de la colección
- Es un libro, folleto, tesis, publicación o periódico
- Puede ser una obra de referencia
- Puede ser una obra de tapa roja
- Puede tener más de un ejemplar

##### Impacto

- Después de la adquisición, el bibliotecario realiza el registro y el procesamiento

técnico

- Después del procesamiento técnico, se puede localizar, consultar, prestar, devolver, renovar, reservar y prestar para fotocopiar

**Figura 1** Ejemplo simbología del LEL.

Las heurísticas que son utilizadas para la construcción del LEL son: identificar fuentes de información, identificar símbolos, clasificar símbolos, describir los símbolos, verificar el LEL y validarlo. Una vez que se obtiene el LEL es necesario actualizarlo para obtener la mejor comprensión del UdeD. Así como esta técnica mejora la comprensión del dominio del problema, también es posible que tenga defectos los cuales surgen cuando no se construye un LEL de manera correcta. Los principales defectos que pueden contener el LEL son: de descripción, de clasificación, de identificación y de referencia los cuales están descritos en (Kaplan, et al. 2000).

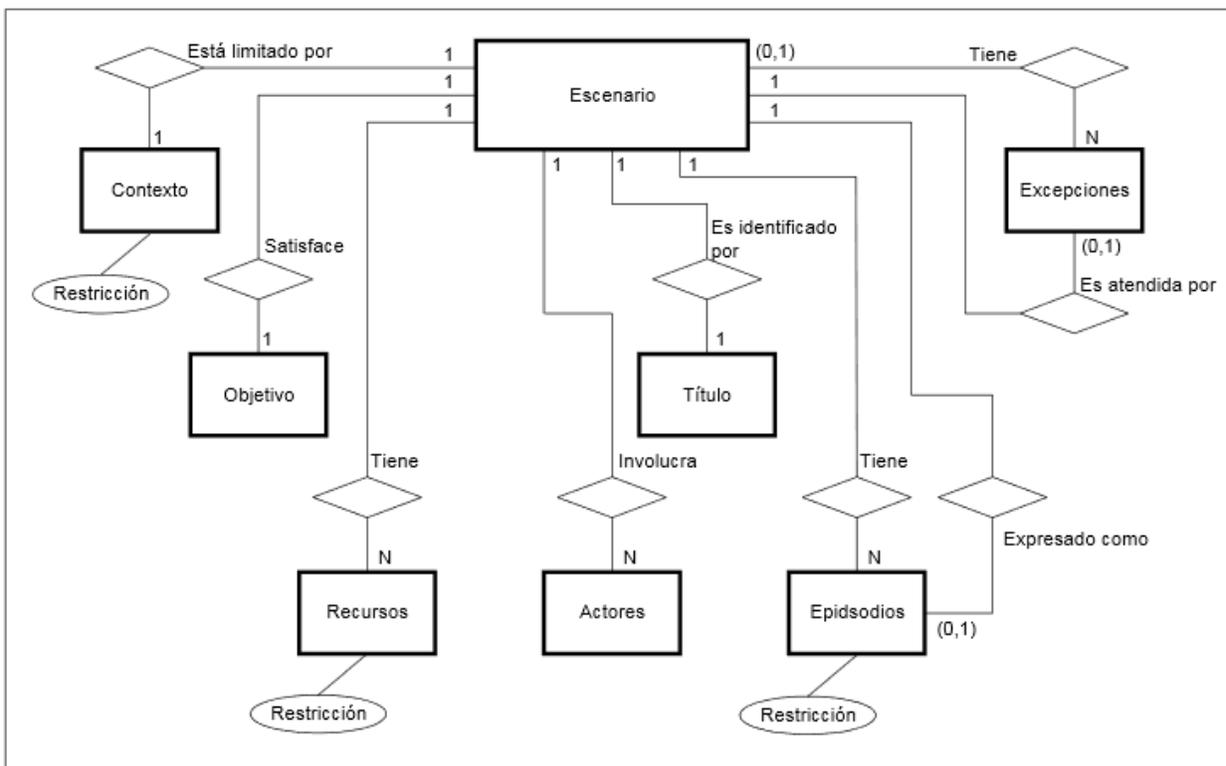
## 2.2 Representación de un Escenario

En Ryser y Glinz (1999) se define un procedimiento para la recopilación de los requisitos y su documentación a través de escenarios, en el cual destacan 15 pasos para la creación de un escenario:

- Encontrar todos los actores que tengan interacción con el sistema.
- Encontrar todos los eventos relevantes al sistema.
- Determinar entradas, resultados y salidas del sistema.
- Determinar límites del sistema.
- Crear descripciones de escenarios amplias.
- Priorizar los escenarios de acuerdo a la importancia, asegurar que los escenarios cubran la funcionalidad del sistema.
- Crear una descripción paso a paso de eventos y acciones para cada escenario •
- Crear un gráfico general y un diagrama de dependencia.
- Tener opinión y comentarios de los usuarios sobre los diagramas y escenarios.

- Extender escenario al refinar su descripción, dividir las tareas en pasos de trabajo individuales.
- Flujo de acciones del modelo alternativo, especificar excepciones y cómo reaccionar a las excepciones.
- Factorizar escenarios abstractos (secuencias de interacciones que aparecen en más de un escenario).
- Incluir requisitos no funcionales y cualidades en escenarios.
- Revisar el diagrama de información general y diagrama de dependencia.
- Pedir a los usuarios comprobar y validar los escenarios (revisiones formales).

Para la construcción de un escenario es preciso considerar los elementos por los que este escenario estará formado. En (Hadad, G et al 1996) se muestran algunos ejemplos de escenarios, en los cuales se observan los siguientes elementos como parte de un escenario: nombre, objetivo, contexto, recursos, actores, conjunto de episodios y casos alternativos. En (Hadad, Kaplan, Oliveros y Leite, 1997) se propone otra representación de un escenario que se ilustra como un diagrama E-R y se observa en la Figura 2.

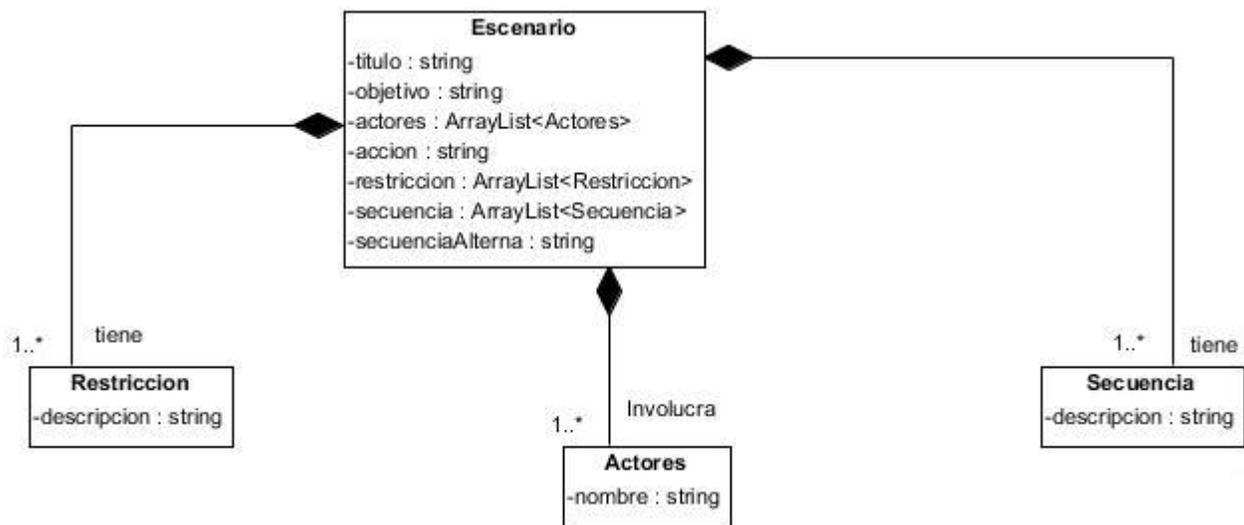


**Figura 2.** Diagrama Entidad-Relación para la construcción de un escenario

### 3. Patrones de Escenario

El uso de patrones en la construcción de escenarios permite ahorrar tiempo para la elaboración de los mismos, ofreciendo una visión estructural de las situaciones, y con esto determinar el tipo de escenario que corresponda a cada situación. En este sentido existen algunas propuestas como la de (Ridao 2001), sin embargo ofrecen situaciones muy abstractas y difíciles de utilizar.

Tomando en cuenta los trabajos revisados, se propone incluir en la representación de un escenario los elementos: título, objetivo, acción, recursos, actores, episodios y excepciones, mismos que se representan en un diagrama de clases UML como se muestra en la Figura 3.



**Figura 3.** Elementos Escenario

Además, se propone el uso de patrones para la representación de escenarios basados en la notación LEL, compuesto por la siguiente simbología:

- + significa composición,
- {x} significa cero o más ocurrencias de x,
- ( ) es usado para agrupamiento,
- | significa or, y
- [x] denota que x es opcional

El patrón se forma con los siguientes elementos:

- Título

- El título identifica de forma única un escenario para una acción específica del sistema.
  - Sintaxis: Frase
    - Objetivo
  - El objetivo describe de forma breve lo que se logra con implementación de este escenario.
  - Sintaxis: [Actor | Recurso] + Verbo + Predicado
    - Actores
  - Se mencionan los involucrados que participan en este escenario en particular.
  - Sintaxis: Nombre
    - Acción
  - Describe la actividad que se ejemplifica este escenario.
  - Sintaxis: [Sujeto | Actor | Recurso] + Verbo + Predicado + {Restricción}
    - Restricción
  - Describe la situación en la que este escenario tendrá lugar (opcional)
  - Sintaxis: [Actor | Recurso] + Verbo + Predicado
    - Secuencia
  - Contiene los pasos que describen de forma clara los pasos que sigue este escenario, para llegar a un fin.
  - Sintaxis:
    - <sentencia básica> ::= <sentencia simple> | <sentencia condicional> | <sentencia opcional>
    - <sentencia simple> ::= <sentencia episodio>
    - <sentencia condicional> ::= IF <condición> THEN <sentencia episodio>
    - <sentencia opcional> ::= [ <sentencia episodio > ]
- donde
- <sentencia episodio > es descrita como:
- (([Actor | Recurso] + Verbo + Predicado) | ([Actor | Recurso] + [Verbo] + Título)) + {Restricción}
- Excepción

- En caso de que durante el proceso de esta actividad no se logre completar satisfactoriamente en este apartado se describe lo que sucede en esos casos.
- Sintaxis:  
Causa [(Solución)] donde Causa es:  
Frase | ([Sujeto | Actor | Recurso] + Verbo + Predicado) donde Solución es:  
Título

### 3.1 Catálogo de Patrones Propuesto

El catálogo propuesto se basa en el análisis realizado a distintas fuentes de información y problemas de definición de estos requisitos en el dominio de empresas que se dedican al desarrollo de software en el área de telemática, específicamente orientado al área de transporte, y que presentan requisitos donde sus clientes desea obtener información sobre sus unidades en tiempo real. Este catálogo desarrollado consta de los siguientes patrones de escenario:

- Alarma por email: Describe una acción, en la cual se requiere que el sistema genere una alerta vía email para notificar algún evento que requiera de atención inmediata.
- Alerta por email: Describe una situación, en la cual se requiere que el sistema genere una alerta vía email para notificar algún evento que requiera atención.
- Alta motor: Describe los pasos requeridos para dar de alta un motor para una unidad (transporte).
- Consultar: Describe la situación cuando se requiere exportar cierta información del sistema a un archivo (Excel, PDF, entre otros)
- Exportar: Describe una acción, en la cual se requiere consultar cierta información requerida por el cliente
- Integración proveedor: Describe cuando se requiere la integración de sistemas a través de Servicios Web (Web Services), conexión directa a base de datos o por importación de un archivo externo, proporcionando un Servicio Web para que otro sistema lo utilice
- Integración consumidor: Describe una acción, en la cual se requiere la integración

de sistemas a través de Servicios Web (conexión directa a base de datos o por importación de un archivo externo a la empresa) para que una aplicación lo utilice. Cada uno de estos patrones se describe detalladamente con los términos mencionados antes. En la Tabla 1 se muestra la definición detallada del patrón de escenario Exportar.

**Table 1.** Patrón de Escenario Exportar

<b>Título</b>	Exportar
<b>Objetivo</b>	Exportar información solicitada de las unidades en un archivo externo.
<b>Actor (es)</b>	Al menos un usuario
<b>Acción</b>	Exportar información del sistema en un archivo externo
<b>Restricción</b>	Una o ninguna
<b>Secuencia</b>	<p>POR LO MENOS UNO COMO EL SIGUIENTE:</p> <ul style="list-style-type: none"> <li>• El sistema debe contar con la funcionalidad de exportar en un archivo externo la siguiente información:</li> <li>• (Escribir los datos que son necesarios en el reporte)</li> </ul> <p>Si se describe el formato para el archivo de exportación anotarlo en esta sección.</p> <p>UNO COMO EL SIGUIENTE</p> <ul style="list-style-type: none"> <li>• La opción de reporte se debe encontrar en la siguiente sección:</li> <li>o (Escribir la sección en la que se tendrá la opción de exportación )</li> </ul>
<b>Excepción</b>	(Circunstancia que obstaculiza que se pueda exportar información de las unidades en un

archivo externo)

Para ejemplificar el uso de este patrón, a continuación se muestra un requisitos donde se ilustra su aplicación.

## Requisitos

Después de una entrevista con el cliente, se establecieron las siguientes necesidades:

- Se requiere un reporte denominado análisis distribución, el cual muestra la información de los viajes de la flota.
- Se desea que este reporte se exporte a formato Excel.
- Solo los usuarios del departamento de logística y usuarios con rol de administrador tendrán permisos para generar este reporte.
- La opción de exportar debe encontrarse en el menú Estadísticas, en el Sub-menú Reportes. • El reporte debe contener la siguiente información:
  - o Unidad -> nombre unidad
  - o Viaje -> número identificación de viaje
  - o Combustible del viaje -> combustible consumido en el viaje
  - o Distancia de viaje -> distancia total recorrida del viaje
  - o Excesos de velocidad -> cantidad de excesos de velocidad (+ 90 km/h)
  - o Velocidad máxima -> velocidad máxima de la unidad registrada en el viaje
  - o Combustible ralentí -> combustible usado en ralentí.

## Identificación del patrón a utilizar

Se sugiere leer con detenimiento la descripción del requisitos e identificar si dentro de la misma aparece el nombre (o un sinónimo) de alguno de los patrones propuestos. Además es conveniente revisar la sección de objetivo de los posibles patrones que se aplican al problema para así determinar cuál patrón aplicar. Para el caso del ejemplo se identifica que es aplicable el patrón Exportar.

## Identificación de los Símbolos LEL

Revisando la descripción del requisitos se deben identificar aquellos términos propios del problema que es necesario detallar, y entonces identificarlas en el diccionario de símbolos LEL; de no encontrarlas se deberán agregar.

Para el caso que se ilustra, se identifican los términos Flota y Ralentí, como palabras que tienen un alto impacto en el dominio del problema y son usadas con frecuencia en dicho dominio, y deben incluirse en el LEL. En las tablas 2 y 3 se presenta la definición de éstos símbolos.

**Table 2.** Símbolo LEL Flota

<b>Símbolo</b>	<b>Flota</b>
<b>Conjunto de medios de transporte que realizan la misma actividad.</b>	Exportar información solicitada de las unidades en un archivo externo.
<b>Noción</b>	Conjunto de medios de transporte que realizan la misma actividad.
<b>Impacto</b>	<ul style="list-style-type: none"><li>• Controlar el combustible gastado para ciertas flotas de transporte de acuerdo a su trayecto</li><li>• Mejorar la productividad</li><li>• Definir ciclos de mantenimiento de acuerdo al desgaste de determinada flota.</li></ul>

**Table 3.** Símbolo LEL Ralentí

<b>Símbolo</b>	<b>Ralentí</b>
----------------	----------------

<b>Objetivo</b>	Exportar información solicitada de las unidades en un archivo externo.
<b>Noción</b>	Régimen mínimo de revoluciones por minuto (giros o vueltas por minuto) a las que se ajusta un motor de combustión interna para permanecer en funcionamiento de forma estable sin necesidad de accionar un mecanismo de aceleración o entrada de carburante.
<b>Impacto</b>	<ul style="list-style-type: none"> <li>• Se utiliza para calcular la cantidad de litros de combustible gastados por una unidad entando en ralentí.</li> </ul>

**Table 4..** Implementación Patrón Exportar

<b>Título</b>	Reporte “Análisis Distribución”
<b>Objetivo</b>	Exportar información solicitada de las unidades en un archivo externo.
<b>Actor (es)</b>	Sistema MotumWeb, Usuario Dep. Log, Administrador
<b>Acción</b>	Exportar información del sistema en un archivo externo
<b>Restricción</b>	Solo usuarios departamento de logística y usuario con rol de administrador pueden exportar la información
<b>Secuencia</b>	<p>El sistema tendrá la funcionalidad de exportar en un archivo externo la siguiente información:</p> <ul style="list-style-type: none"> <li>• Los datos que requiere el usuario en el reporte “Análisis Distribución” son los siguientes:</li> <li>• Unidad</li> </ul>

- Viaje
- Combustible viaje
- Distancia viaje
- Excesos velocidad
- Velocidad máxima
  
- Combustible ralentí

**Secuencia** El formato del reporte debe ser en Excel.

- La opción de reporte debe encontrarse en la siguiente sección:

Menú: Estadísticas  
Sub menú: Reportes

**Excepción** Error de comunicación con la base de datos. La unidad no ha reportado viajes realizados

## 4. Conclusiones

El brindar las directrices necesarias para implantar una mejora fue de gran ayuda para las organizaciones, ya que, como se demostró en la sección de experimentación, en un primer ciclo las empresas realizaron muchas de las actividades que no se llevaban a cabo hasta antes del uso de Kaizen, lo cual tuvo como resultado una disminución en los tiempos y esfuerzo de desarrollo, obteniendo un producto que cumpla con las expectativas marcadas al inicio del ciclo de desarrollo. Lo anterior establece que una herramienta con las características de Kaizen es una opción recomendable para aquellas pequeñas organizaciones con poca o nula experiencia en iniciativas de mejora y que buscan la implantación de un modelo de procesos dentro de su proceso actual, debido al marco de trabajo controlado y colaborativo que ofrece.

Una de las ventajas que tienen los escenarios es la posibilidad de construirlos a partir de patrones, de tal manera que es posible reutilizar las soluciones a situaciones similares que se presentan en diversos proyectos de software. El contar con una representación formal para un escenario facilita la creación de un catálogo de patrones que ayude a agilizar y hacer más segura la definición de requisitos.

En comparación con las propuestas de patrones de escenario revisadas, ésta se enfoca en describir escenarios específicos a un área de desarrollo y son útiles para la descripción precisa de necesidades del software para telemática, a diferencia del trabajo de Ridao (2001) quien en su propuesta describe situaciones muy abstractas y difíciles de reconocer en primera instancia.

Como trabajo futuro se realizarán las pruebas de campo que permitan exponer las ventajas que trae consigo el uso de un catálogo de patrones para la construcción de escenarios.

## Agradecimientos

El autor corresponsal agradece al Consejo Nacional de Ciencia y Tecnología CONACYT por el apoyo brindado para la realización de los estudios de postgrado.

## Referencias

Do Prado Leite, J. C. S., Rossi, G., Balaguer, F., Maiorana, V., Kaplan, G., Hadad, G., & Oliveros, A. (1997) Enhancing a requirements baseline with scenarios. *Requirements Engineering*, 2(4), 184—198

Doorn, J. H., Hadad, G. D., & Kaplan, G. N. (2002). Comprendiendo el Universo de Discurso Futuro con Escenarios. In *WER*, 117--131.

Hadad, Graciela D.S., Doorn, Jorge., Kaplan, Gladys. (1998). Enfoque middle-out en la Construcción e Integración de Escenarios.

Hadad, G., Kaplan, G., Oliveros, A., & Sampaio do Prado Leite, J. C. (1996). Integración de Escenarios con el Léxico Extendido del Lenguaje en la elicitación de requisitos\*: Aplicación a un caso real. Universidad de Belgrano-Facultad de Ingeniería y Tecnología Informática. 1-21

Hadad, G.; Kaplan, G.; Oliveros, A.; Leite, J.C.S.P. (1997). Construcción del Léxico Extendido del Lenguaje y derivación de Escenarios para la elicitación de requisitos.

Hadad, G. D. S. (2010). Uso de Escenarios en la Derivación de Software. In XII Workshop de Investigadores en Ciencias de la Computación.

Kaplan, G. N., Hadad, G. D., Doorn, J. H., & do Prado Leite, J. C. S. (2000). Inspección Del Lexico Extendido Del Lenguaje. In WER, 70—91.

Ridao, Marcela. (2000). Uso de Patrones en el Proceso de Construcción de Escenarios. Workshop de Engenharia de Requisitos. 140—157.

Ridao, Marcela. (2001). Uso de Patrones en el Proceso de Construcción de Escenarios. Tesis Doctoral no publicada. Facultad de Informática, Buenos Aires, Argentina.

Ridao, Marcela and Jorge Doorn., and Julio Cesar Sampaio (2001). Incorporación de Patrones al Proceso de Construcción de Escenarios. Workshop em Engenharia de Requisitos, Buenos Aires, Argentina, s.f. 107—123.

Ryser, J., & Glinz, M. (1999). A scenario-based approach to validating and testing software systems using statecharts. In Proc. 12th International Conference on Software and Systems Engineering and their Applications.

Sutcliffe, A. (2003). Scenario-based requirements engineering. In Requirements engineering conference., Proceedings. 11th IEEE international 320—329.

## Notas biográficas:



**Carlos Alberto Cortés Bravo** Carlos Alberto Cortés Bravo es Ingeniero en Sistemas Computacionales, egresado del Instituto Tecnológico de Orizaba (ITO), actualmente estudia la Maestría en Sistemas Computacionales en la misma institución (ITO), sus áreas principales de interés son: Ingeniería de Software y el desarrollo de Software Web, ha desarrollado módulos de sistemas Web de administración para empresas mediante el uso de varios lenguajes, ha sido parte de proyectos de sistemas Web de telemetría como ingeniero de software. ISC Cortés es miembro de la ACM



**Abud Figueroa María Antonieta** Abud Figueroa María Antonieta, nació en la ciudad de Orizaba, Ver. Es ingeniero en electrónica por la UAM-Iztapalapa, México DF en el año 1984, y maestra en ciencias en sistemas de información por el ITESM-Morelos, en la ciudad de Cuernavaca, Mor. en el año 1991.

Ella fue profesora de tiempo completo en el ITESM Campus Central de Veracruz entre los años 1985 y 1993; desde el año 1995 es profesora-investigadora en el área de posgrado del Instituto Tecnológico de Orizaba, en la ciudad de Orizaba, Ver. México. Su línea de investigación es la Ingeniería de Software. La M.C. Abud es miembro del ACM y del IEEE



**Romero Torres Celia** Romero Torres Celia nació en Orizaba, Veracruz. Se graduó como Licenciada en Informática, en la Universidad Veracruzana en 1989, en febrero de 2005 obtiene el grado de Maestra en Ciencias en Ciencias Computaciones en el instituto Tecnológico de Orizaba. Ella ejerce como profesor de tiempo completo en el Instituto Tecnológico de Orizaba ubicado en la ciudad de Orizaba, Veracruz. Actualmente es la

coordinadora del programa de Maestría en Sistemas Computacionales La MC Romero es miembro del ACM.



**Ulises Juárez Martínez** Ulises Juárez Martínez es Doctor en Ciencias en la especialidad de Computación por parte del CINVESTAV. Sus áreas de interés comprenden la adaptación en vivo de sistemas de software, desarrollo de software orientado a aspectos, programación generativa, líneas de productos de software y lenguajes de programación. Actualmente es profesor investigador del Grupo de Ingeniería de Software en el Instituto Tecnológico de Orizaba.



**Gustavo S. Peláez Camarena** Gustavo S. Peláez Camarena Ingeniero Industrial egresado del Instituto Tecnológico de Orizaba en el año 1976; Obtención del grado de Maestro en Ciencias en Cómputo Estadístico en el Colegio de Postgraduados en Chapingo, México en el mes de junio del 1980. Especialidad en educación a distancia en la Universidad Veracruzana en el año 2004. Actualmente se desempeña como Profesor Investigador de tiempo completo en el Instituto Tecnológico de Orizaba; sus áreas principales de interés son: Ingeniería de Software (Desarrollo de aplicaciones Web y desarrollo de aplicaciones para apoyo a la educación).



Esta obra está bajo una licencia de Creative Commons Reconocimiento-NoComercial-CompartirIgual 2.5 México.