

## Aplicación de un Método para Especificar Casos de Prueba de Software en la Administración Pública

Edumilis Méndez, María Pérez, Luis E. Mendoza

Departamento de Procesos y Sistemas, Edificio de Matemáticas y Sistemas, Laboratorio de Investigación en Sistemas de Información (LISI), Universidad Simón Bolívar, Apartado 89000, Baruta, Caracas, 1080-A, Venezuela  
{emendez, movalles, lmendoza}@usb.ve

### Resumen

Una diferencia interesante entre las Pruebas y otras disciplinas del proceso de desarrollo de software es que esencialmente es una tarea que encuentra y pone de manifiesto las debilidades del producto de software. Existen cuatro elementos que son relevantes al momento de definir las pruebas: Confiabilidad, Costo, Tiempo y Calidad. En la medida que se deseen pruebas confiables y un software de calidad, el tiempo y el costo se incrementarán. Pero ¿qué se puede hacer para que los involucrados comprendan que las pruebas deben ser vistas como una red de seguridad? Si la Calidad no está ahí antes de comenzar las pruebas, entonces no estará cuando se terminen. En base a esto, ¿cómo se puede hacer una traza entre las Pruebas y los requerimientos funcionales y no funcionales del sistema de software? El objetivo de este artículo es proponer un método que permite especificar Casos de Prueba a partir de Casos de Uso incorporando elementos que permiten verificar y validar la trazabilidad entre la Gestión de Requerimientos, el Análisis y Diseño y las Pruebas. Esta iniciativa se originó como respuesta a una necesidad del sector público venezolano que desarrolla software.

### 1. Introducción

La Disciplina de Pruebas de Rational Unified Process (RUP) se aborda desde la fase de inicio [3]. Esto quiere decir que las pruebas se deben comenzar a planificar y además se debe establecer cuál es la estrategia de pruebas a seguir una vez que han aprobado todos los Casos de Uso (CU) correspondientes a la iteración. Esta disciplina amerita un mayor esfuerzo en la fase de construcción ya que es el momento en el que se valida (¿se está construyendo el producto correcto?) el producto de software. Antes, durante

y después de esta fase se realizan actividades de verificación (¿se está construyendo el producto correctamente?).

El objetivo de la Disciplina de Pruebas es evaluar la calidad del producto a lo largo de todo el ciclo de vida apoyándose en un conjunto de buenas prácticas, entre las que destacan [4]:

- Verificar que el producto de software trabaja según el diseño.
- Validar que los requerimientos son implementados apropiadamente.

En [2] indican que esta disciplina, generalmente no se implementa de forma organizada y sistemática. Además, algunos autores como [1, 3, 5, 6, 7] afirman que el proceso de ejecución de Pruebas debe ser considerado durante todo el ciclo de vida de un proyecto, para así obtener un producto de alta calidad. Su éxito dependerá del seguimiento de una Estrategia de Prueba adecuada. La Estrategia de Prueba de software integra un conjunto de actividades que describen los pasos que hay que llevar a cabo en un proceso de prueba, tomando en consideración cuánto esfuerzo y recursos se van a requerir, con el fin de obtener como resultado una correcta construcción del software [6].

Las empresas desarrolladoras de software invierten en las pruebas entre el 30% y 50% del costo total del software [6]. Esto representa un esfuerzo considerable indicando que es una disciplina cuyos resultados (confiabilidad de la prueba) pueden impactar sobre la percepción del cliente o usuario en cuanto a la calidad del software que se le está entregando.

En el año 2004 el Estado Venezolano promulgó el decreto 3.390, que establece el uso prioritario de Software Libre (SL), basado en estándares abiertos en los servicios y sistemas informáticos de los organismos pertenecientes a la Administración Pública Nacional (APN). Por lo

tanto, la responsabilidad recayó en el Ministerio del Poder Popular para las Telecomunicaciones y la Informática (MPPTI), tomando como responsabilidad definir los lineamientos y políticas para llevar a cabo los procesos institucionales de migración gradual y progresiva a SL a través de proyectos de desarrollo de software. Para ello, se propuso una metodología basada en Unified Process (UP).

Pero, como empresa, ¿qué estrategia se puede utilizar, qué metodología o método se debe seguir para establecer una trazabilidad entre las pruebas y los requerimientos y garantizar un mejoramiento de las actividades de verificación y validación? Garantizándose así la entrega de un producto de software de Calidad.

En respuesta a ello, el Laboratorio de Investigación en Sistemas de Información (LISI) atendió al llamado del MPPTI al proponer un método que permite especificar CP a partir de los CU como punto de partida para la estandarización y la trazabilidad de todo el proceso de desarrollo de software y de forma eficaz, obteniendo productos de calidad con una alta productividad.

Este artículo presenta la siguiente estructura: inicialmente se tratan los Casos de Prueba y la relación de la disciplina de pruebas con otras disciplinas presentes en el proceso de desarrollo de software. En la sección 3, se muestra el Método propuesto indicando los roles, actividades y artefactos asociados. Por último, las conclusiones y próximos pasos.

## 2. Casos de prueba

Un CP es una especificación, usualmente formal, de un conjunto de entradas de prueba, condiciones de ejecución y resultados esperados, identificados con el propósito de hacer una evaluación de aspectos particulares de un elemento objeto de prueba [4]:

- Los CP reflejan trazabilidad con los CU (Funcionalidad), ya que estos muestran una secuencia ordenada de eventos, al describir flujos básicos, flujos alternos, precondiciones y postcondiciones.
- Las especificaciones suplementarias de requerimientos ya que existen otras características de calidad a evaluar, además de la funcionalidad, como Usabilidad,

Confiablez, Eficiencia, Mantenibilidad y Portabilidad.

- Las especificaciones de diseño del Sistema, ya que se debe verificar que el software fue implementado según el diseño y que los elementos arquitectónicos garantizan la calidad del software.

Esto garantiza que los procedimientos de pruebas sean compatibles con las necesidades de los usuarios/clientes. En la práctica se tiende a asumir que un CU en si mismo es un CP y que el equipo del proyecto trabaje correctamente sobre los CU sin planificar los CP. Cuando se sienta a probar los CU, intuitivamente asume datos y procedimientos de pruebas sin que estas queden documentadas. Esto es un error ya que el CP extiende o amplía la información contenida en un CU; por ejemplo, en los CU no indicamos valores ni condiciones para la pruebas.

Los CP son esenciales para todas las actividades de pruebas [4]:

- Son la base para diseñar y ejecutar los procedimientos de pruebas.
- La profundidad de las pruebas es proporcional al número de casos de pruebas.
- El diseño y desarrollo, y los recursos necesarios son gobernados por los casos de pruebas requeridos.

Si los CP no son correctos, la Calidad del Sistema se pone en duda y las pruebas dejan de ser confiables.

La Figura 1 muestra un modelo conceptual sobre los conceptos que están asociados a los CP. En ella se puede observar que los CU son la fuente principal de los CP, estos se encuentran como entregables del documento Plan de Pruebas de RUP. Es importante señalar que en la plantilla de este documento no se encuentra un formato establecido para especificar los CP. Los CP se pueden agrupar mediante Suite de Pruebas. Los CP están relacionados con el nivel de la prueba y con el tipo de prueba, la cual a su vez contiene la técnica que permite ejecutar el tipo de prueba. Los CP proveen las instrucciones para el procedimiento de las pruebas. En el método que se propone en este artículo se indica que el procedimiento de la prueba se conforma de pasos, condiciones, valores y resultados esperados y obtenidos. A su vez, el procedimiento de las pruebas puede ser automatizado a través de los script de pruebas. Todos los conceptos indicados

anteriormente permiten visionar el enfoque de pruebas: ¿qué se probará, cómo, quién, cuándo, para qué? Una vez ejecutados todos los CP, estos resultados deben reflejarse de manera global. Con ello, se establece si al validar el sistema se cumplió con los criterios de aceptación definidos con el usuario.

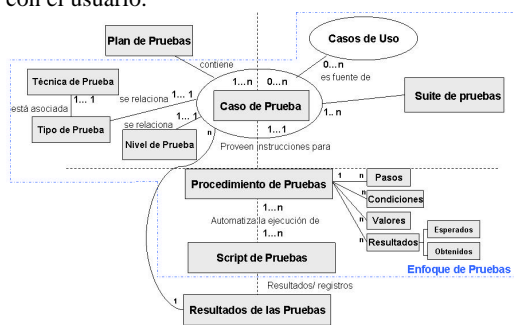


Figura 1. Modelo Conceptual asociado a los CP.

En la siguiente sección se explica en detalle el método propuesto.

### 3. Método para especificar casos de prueba (MECAP)

El método propuesto consiste en construir CP a partir de CU ya que se parte del supuesto que se debe probar el comportamiento del software en base a las solicitudes o requerimientos. El moverse de un CU a un CP es un proceso razonablemente amplio y no trivial. En [4] señalan cuatro (4) pasos para lograrlo. Estos pasos indican qué debe hacerse pero no se enseña el cómo de una forma detallada; es decir, hay cosas que no están escritas y que después de la revisión bibliográfica y de la experiencia con el MPPTI y otras empresas fueron propuestas y recogidas a través de MECAP. Tomando como base los pasos que establecen [4] se propone un método para especificar CP a partir de CU, constituido por cuatro fases principales:

1. Identificar Escenarios.
2. Identificar CP.
3. Especificar los CP.
4. Ejecución y Aprobación del CP.

*El aporte de MECAP es que incorpora elementos de trazabilidad de las pruebas con respecto a todo el proceso de desarrollo, y a su vez mejora la estrategia de las pruebas al disciplinar este proceso, estructurándolo en fases, actividades que*

*las componen, los roles o personas que participan en ella y el artefacto que se genera. Así mismo, ayuda a documentar las ideas previas a las pruebas, los CP y cómo estos fueron generados, todo esto se utiliza para verificar la trazabilidad de las pruebas con respecto a las fases previas y las disciplinas de Requerimientos, Análisis y Diseño. Además, se garantiza la gestión del conocimiento de la organización en lo que a Aseguramiento de la Calidad se refiere.*

*Este método incorpora los 4 roles que se proponen dentro de la Disciplina de Pruebas: Gerente de Pruebas, Diseñador de las Pruebas, Analista de Pruebas y Probador. A continuación se describe cada fase. Por limitaciones de espacio, se hará notar los aportes de los autores destacando el elemento mediante letra cursiva.*

*Cabe resaltar que MECAP es activado por el analista de pruebas una vez que se hayan verificado las narrativas de los CU; al estar aprobadas las funcionalidades del sistema por parte de los stakeholders.*

#### 3.1. Fase 1: Identificar escenarios.

Las actividades de esta fase, a ser realizadas por el Analista de Pruebas son las siguientes:

1. Se identifican los escenarios tomando como base las narrativas de los CU y considerando cada uno de los escenarios específicos que ocurren para cada CU. El flujo normal, cada flujo alterno o la combinación de ellos es un escenario, que puede ser ejecutado y probado. *Esto deriva que siempre el primer escenario sea el que evoca todo el flujo normal de ese CU en particular y que la relación entre CU y escenarios sea de uno a muchos.*
2. *Presentar gráficamente la secuencia de eventos que se plantea en cada CU:* esto permite, como lo muestra la Figura 2 abstraer los eventos que ocurren en un CU: el flujo normal o básico y los flujos alternos, *y sirve de apoyo para visualizar fácilmente las posibles combinaciones que representarían un escenario ya que establece en qué punto del flujo básico ocurre y además qué sucede después que se activa ese flujo alterno: finaliza el CU o retorna al flujo básico.*
3. *Chequear que se hayan representado gráficamente todos los Flujos alternos con su acción de finalización o retorno.*
4. Establecer a través de una tabla (como lo refleja la Figura 3), todos los escenarios asociados a un

CU de la Figura 2.

5. Identificar cada escenario del CU indicando el flujo normal y/o el o los flujo(s) alternativo(s) que lo componen. La Figura 4 constituye el primero de los 3 artefactos que se generan en MECAP: Tabla de Escenarios por CU. En esta se puede observar que se incorpora el ID del Escenario con el propósito de establecer un elemento de trazabilidad de las pruebas lo que facilita, a su vez, las actividades de verificación y aprobación de las pruebas, así como el CU asociado a este. Como se puede observar en la Figura 4 el ID puede estar conformado por el Nro. CU y Nro. Escenario.

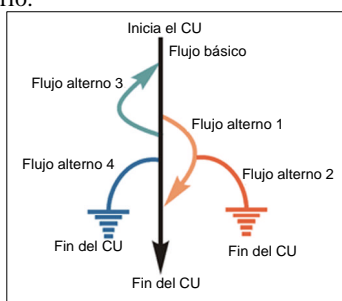


Figura 2. Visualización de los Flujos en un CU [4]

Nro. Escenario	Flujo Originario	Flujo alternativo	Próximo alternativo	Próximo alternativo
Escenario 1	Flujo Básico			
Escenario 2	Flujo Básico	Flujo alternativo 1		
Escenario 3	Flujo Básico	Flujo alternativo 1	Flujo alternativo 2	
Escenario 4	Flujo Básico	Flujo alternativo 3		
Escenario 5	Flujo Básico	Flujo alternativo 3	Flujo alternativo 1	
Escenario 6	Flujo Básico	Flujo alternativo 3	Flujo alternativo 1	Flujo alternativo 2
Escenario 7	Flujo Básico	Flujo alternativo 4		
Escenario 8	Flujo Básico	Flujo alternativo 3	Flujo alternativo 4	

Figura 3. Escenarios por CU [4]

ID Escenario	Flujo Originario	Flujo alternativo	Próximo alternativo	Próximo alternativo
02-01	Flujo Básico			
02-02	Flujo Básico	Flujo alternativo 1: Error Login		
02-03	Flujo Básico	Flujo alternativo 1: Error Password	Flujo alternativo 3: Presionar Cancelar	
02-04	Flujo Básico	Flujo alternativo 2: Error Password		
02-05	Flujo Básico	Flujo alternativo 2: Error Password	Flujo alternativo 3: Presionar Cancelar	
02-06	Flujo Básico	Flujo alternativo 1: Error Login	Flujo alternativo 2: Error Password	
02-07	Flujo Básico	Flujo alternativo 1: Error Login	Flujo alternativo 2: Error Password	Flujo alternativo 3: Presionar Cancelar
02-08	Flujo Básico	Flujo alternativo 3: Presionar Cancelar		

Figura 4. Escenarios para CU002

Esto nos indicaría, por ejemplo, que si se tiene 03-02, se traduce como el escenario 02 del CU 03.

Esta nomenclatura debe ser establecida por la organización. Además, es importante que se agregue un nombre corto durante la identificación de los flujos alternos que representan o conforman un escenario. La información de los escenarios que se muestran en la Figura 4 está asociada a la narrativa de un CU (Validar Usuario del Sistema Tarjeta Académica Inteligente –TAI–), el cual sirve de ejemplo para explicar el método propuesto.

6. Verificar que se hayan identificado y descritos todos los escenarios posibles para cada CU.

En síntesis, cada escenario representa el número de posibilidades o formas a través de las cuales se puede ejecutar el CU. Esto evita que sólo se prueben algunas de las combinaciones posibles.

### 3.2. Fase 2: Identificar los CP

El proceso de pruebas varía según la empresa y el proyecto, pero en cada situación un CP documenta un número de elementos comunes.

De un escenario de un CU pueden derivarse varios CP. Después que se han identificado todos los escenarios de un CU, se analizan cada uno de ellos en la búsqueda de los posibles CP. Esto se traduce en las siguientes actividades a ser realizadas por el Diseñador de las Pruebas:

1. Se organizan las ideas de pruebas en base a los elementos que se desean probar: funcionalidad (CU), atributos de calidad, validaciones de entradas y salidas, base de datos, interfaces, etc. Esto va a depender del tipo de aplicación, de las restricciones tecnológicas, del alcance del proyecto, del propósito y la motivación de las pruebas, y experiencia del equipo de pruebas (sobre todo, del probador).

2. Se comienza a llenar la Tabla que se muestra a través de la Figura 5, la cual representa el segundo artefacto del método propuesto y cuyos datos están asociados a considerar los CP para el Escenario 02-02 (Error en Login). Con la información proveniente de las ideas de pruebas se puede decir que para este caso se podría tener un CP para validar “error en login” cuando se introducen caracteres inválidos, un CP cuando el login está en minúsculas, un CP cuando el login es mayor a 10 caracteres, un CP cuando el login está en blanco. Se completa la información asociada a cada CP identificado: ID Caso de Prueba, Nombre del CP, Resultados esperados (valores,

mensajes de error, etc.), Nivel de prueba y Tipo de prueba. Con respecto al ID del CP se propone incluir en la nomenclatura estándar que determine la organización una estructura que refleje CU-Escenario-CP para así saber por ejemplo que 02-02-01 significa el CP 01 del Escenario 02 del CU 02.

3. Se verifica que se hayan identificado todos los CP para cada escenario. Después de esto, se procede a la siguiente fase.

ID Caso de Prueba	Nombre del Caso de Prueba	Resultados esperados	Nivel de Prueba	Tipo de Prueba
02-02-01	Login con caracteres inválidos	Mensaje Error 20: Login inválido	Sistema/Aceptación	Función/Control de acceso
02-02-02	Login con minúsculas	Mensaje Error 20: Login inválido	Sistema/Aceptación	Función/Control de acceso
02-02-03	Login con longitud mayor a 10 caracteres	Mensaje Error 20: Login inválido	Sistema/Aceptación	Función/Control de acceso
02-02-04	Login en Blanco	Mensaje Error 20: Login inválido	Sistema/Aceptación	Función/Control de acceso

Figura 5. Casos de Prueba para el Escenario 02-02

### 3.3. Fase 3: Especificaciones de los CP

Uno de los aportes más importantes de esta investigación es el tercer artefacto (*su diseño*) que se utiliza para especificar con detalle el CP y que se muestra a través de la Figura 6: Tabla de Especificaciones de CP (ECP). Su llenado es también, responsabilidad del diseñador de las Pruebas. Para cada CP, se deben llevar a cabo las siguientes actividades:

1. Identificar el nombre del Sistema, ID CU, ID de Requerimiento, ID Escenario, ID del CP y versión del CP. Esto permite hacer una traza bidireccional entre estos elementos: por ejemplo, se puede conocer si se especificaron todos los CP para todos los CU y se probaron todos los CU (cobertura de las pruebas).
2. Identificar el nivel y tipo de prueba asociada al CP y cuya información proviene de la Tabla CP por Escenario.
3. Indicar el ambiente de las pruebas. Se puede señalar el nombre de la empresa, si es el ambiente de desarrollo o producción, o si tengo varios ambientes en la empresa se indica en cuál de ellos se ejecutará ese CP en particular.
4. Identificar el nombre del autor del CP y del Probador. Se recomienda que sean personas diferentes las que ocupen estos roles a fin de darle seriedad y confiabilidad al proceso de las pruebas.
5. Señalar la fecha de creación de la versión de ese

CP y la fecha en la que fue ejecutado.

6. Identificar las condiciones que deben estar presentes para ejecutar el CP. ¿Cuáles son las condiciones que originan o causan que un usuario ejecute un evento específico o una secuencia de eventos? En la Figura 6, se tiene el CP 02-02-02 en cual se propone el CP para Login con minúsculas. En esta figura se observa que se deben haber implementado todas las funcionalidades asociadas con Validar Usuario, así mismo, los Datos que se utilizaran en este CP hayan sido validados y aprobados por la instancia correspondiente, etc.

ID/Nombre Sistema/Proyecto: SIS-PROY	Nivel de Prueba: Sistema/Aceptación			
ID Caso de Uso: CU-02 Validar Usuario	Tipo(s) de Prueba(s): Función/Control de acceso			
ID Requerimiento: <i>(solo para Caso de Uso No Funcional)</i>	Ambiente de Prueba: <b>AMBENTEX</b>			
ID/Nombre Escenario: 02-02 Error en Login	Autor del Caso de Prueba: LISI			
ID/Nombre Caso de Prueba: 02-02-02 Login con minúsculas	Nombre del Probador: Probador 1			
Versión del Caso de Prueba: v.1.	Fecha de Creación: 10.01.07	Fecha de Ejecución: 15.03.07		
<b>Condición(es) para que se ejecute el Caso de Prueba:</b>				
El usuario desea acceder al sistema. Se hayan implementado todas las funciones que se relacionan con validar usuario. Datos a utilizar para las pruebas hayan sido validados y aprobados. Deben existir usuarios registrados como válidos.				
<b>Para la Ejecución del ID Caso de Prueba:</b>				
<b>Paso</b>	<b>Condición</b>	<b>Valor(es)</b>	<b>Resultado Esperado</b>	<b>Resultado Obtenido</b>
Introduce login presiona Ok	intento de acceso < 3	adm22	Mensaje Error 20 Login inválido	✓
Introduce login presiona Ok	intento de acceso < 3	administrador	Mensaje Error 20 Login inválido	✓
Introduce login presiona Ok	intento de acceso < 3	AdminiSTRADOR	Mensaje Error 20 Login inválido	✓
<b>Criterios de Aprobación del Caso de Prueba: Si se cumplen en un 100% los resultados esperados</b>				
Decisión de Aprobación del Caso de Prueba: Aprobó: <input checked="" type="checkbox"/> Fallo: <input type="checkbox"/> <i>(marque con una X el resultado obtenido)</i>				
Fecha de Aprobación del Caso de Prueba: 15.03.07				

Figura 6. Especificación del CP (ECP) 02-02-02

7. Describir el procedimiento del CP. Este procedimiento está compuesto de los pasos que se deben hacer para probar el Escenario del CU a través de lo que plantea el CP, de las condiciones particulares que pudieran aplicar para un determinado paso, los valores utilizados, los resultados esperados y los resultados obtenidos. Cabe resaltar, que este último se incluye en la Tabla ECP una vez que el CP es ejecutado.

Sin la incorporación de datos no es posible ejecutar las pruebas y determinar los resultados. Se deben observar las especificaciones suplementarias para encontrar medidas de desempeño (mínimo y máximo), rangos válidos de entrada, protocolos de interfaces, entre otros.

Como se puede observar en la Figura 6, se tiene como pasos: 1) introducir login y 2) presionar Ok. Para que se ponga en práctica el paso 1 se tiene como condición que el usuario debe haber hecho menos de 3 intentos. De lo contrario el paso no puede activarse. Así mismo, se indican los valores que se deben introducir en el login, para cuáles se debe tratar de utilizar datos aplicables al negocio y a la lógica de lo que propone el CP. Por ejemplo:

si el CP plantea login con minúsculas, el valor debe tener al menos una letra en minúsculas y el resultado esperado (mensaje de error). Este paso puede repetirse varias veces a fin de comprobar que el sistema da un mensaje de error si el login tiene en alguna de sus caracteres una letra en minúsculas, lo que representaría repetir este paso al menos 11 veces dentro del CP.

8. *Se indica cuál es el Criterio de Aprobación del CP.* Como se observa en el ejemplo de la Figura 6. El Criterio es que se deben cumplir en un 100% todos los resultados esperados. También puede indicarse a nivel de pasos cuando el CP involucra pasos diferentes y alguno(s) de ellos son más importante que otros; p.e., El CP está superado si se cumplen en un 100% los resultados esperados del Paso 2, 4 y 5.

9. *El analista y diseñador de las pruebas verifican que todos los CP se hayan especificado correctamente.*

#### 3.4. Fase 4: Ejecución y aprobación del CP

Esta fase consiste en poner en práctica el procedimiento de los CP que se encuentran en la Tabla ECP. A continuación se indican las actividades de esta fase:

1. *Debe comprobarse que está dado el ambiente y las condiciones para ejecutar los CP.* Los involucrados en las pruebas deben cooperar en esto.
2. *El Gerente y Diseñador de las pruebas deben autorizar que se active el ciclo de prueba.*
3. *El Probador ejecuta todos los CP e incorpora los datos de los resultados obtenidos en cada Tabla ECP.*
4. *El Gerente de Pruebas toma la decisión de si aprobó o falló el CP en base al criterio de aprobación y, además, indica la fecha de la aprobación y en algunos casos, stampa su firma.*
5. *El equipo de pruebas verifica si se cumplió el criterio de salida del ciclo de prueba para decidir si se aprueba el ciclo de prueba o hay que hacer pruebas de regresión y pruebas adicionales a ciertos CP en un próximo ciclo de prueba hasta que se obtengan los criterios de aceptación.*
6. *El equipo de pruebas guarda todos estos entregables y publica los resultados de los ciclos de prueba, cambios, etc. que se obtuvieron durante el proceso de pruebas.*

#### 4. Discusión de resultados y próximos pasos

El método propuesto MECAP fue utilizado en el MPPTI para 4 proyectos obteniendo resultados significativos en cuanto a 1) la calidad de los productos desarrollados: al momento de culminar la fase de construcción los sistemas de software ya habían alcanzado un 90% de calidad esperada; 2) el proyecto más grande llegó a implementar 51 CU lo que se tradujo en documentar, ejecutar y aprobar 460 CP y la Tabla de ECP puede ser utilizada para definir procedimientos de CP asociados a requerimientos no funcionales; 3) la experiencia sentó un precedente y será replicada para futuros proyectos y además estableciendo indicadores de gestión que pueden reflejar por ejemplo: el número promedio de CP por aplicación. Actualmente, MECAP está siendo utilizado por otras organizaciones del sector público y privado que recogen 16 proyectos por lo que un próximo paso es publicar los resultados detallados de la aplicación de este método en cada una de estas organizaciones.

#### 5. Conclusiones

En este artículo se describe un método que permite especificar Casos de Prueba a partir de Casos de Uso incorporando elementos que permiten verificar y validar la trazabilidad entre la Gestión de Requerimientos, el Análisis y Diseño y las Pruebas. Esta iniciativa se originó como respuesta a una necesidad del sector público venezolano que desarrolla software. Adicionalmente, se puso en evidencia que los costos de las pruebas pueden verse reducidos a mediano y largo plazo ya que se pueden utilizar Probadores que no sean Especialistas si se tiene definido con anterioridad ¿qué se debe probar, cuándo y cómo?.

#### Referencias

- [1] Cardoso, R. (2001). Pruebas del Software. Mérida, Venezuela.
- [2] Grimán A., Pérez, M. & Mendoza, L. (2003). Estrategia de pruebas para software OO que garantiza requerimientos no funcionales. III

Workshop de Ingeniería de Software, Chillán,  
Chile.

- [3] Kruchten, P. (2000). The Rational Unified Process as Introduction. 2nd Edition. Addison – Wesley.
- [4] Leffingwell D. and Widrig D. (2006). Managing Software Requirements, a Use Case Approach. Second Edition. Addison-Wesley, Pearson Education.
- [5] Pfleeger, S. (1998). Software Engineering.
- [6] Pressman, R. (2002). Ingeniería del Software: Un enfoque Práctico. McGraw Hill.
- [7] Sommerville, I. (2000). Software Engineering. Pearson Education.