

# **Tool for Requirements Management in Cuban's Small and Medium Enterprises**

**Enrique Pérez Rodríguez**

**Universidad de las Ciencias Informáticas (UCI), La Habana,  
Cuba, enriquepr@uci.cu**

**Karina Pérez Teruel**

Universidad de las Ciencias Informáticas (UCI), La Habana, Cuba, karinapt@uci.cu

## **ABSTRACT**

In the current competitive merchandising of software, it is necessary that the enterprises dedicated to the development of these products make emphasis on quality. In all phases of the software construction, the requirement elicitation have the greatest impact on their quality, and for that reason special attention must be dedicated to it. A good practice is to use automated tools to facilitate and make more efficient the performance of that area. There are numerous tools for requirements management, however most of them do not meet the needs of small and medium enterprises, especially in Cuba, particularly given its economic and foreign pressure to which it is forced to be guided by specific policies. The aim of this paper is to propose an alternative tool for requirements management based on a study of different tools developed in Cuba and around the world, as well as the technologies that support them.

**Keywords:** requirements engineering, requirements management, traceability of requirements, functional requirements, nonfunctional requirements.

## **RESUMEN**

En el actual contexto competitivo de la comercialización de software, se hace necesario que las empresas dedicadas al desarrollo de estos productos hagan énfasis en su calidad. De todas las fases de la construcción del software, la de requisitos es la que mayor impacto tiene en su calidad, de ahí que se le deba prestar especial atención. Una buena práctica consiste en utilizar herramientas automatizadas que faciliten y hagan más eficiente su desempeño. Existen numerosas herramientas para la Gestión de Requisitos, sin embargo la mayoría no se ajustan a las necesidades de las pequeñas y medianas empresas, en especial en Cuba, dada su particular situación económica y la presión foránea a la que se encuentra sometida, que la obligan a regirse por políticas específicas. El objetivo del presente trabajo es hacer una propuesta de herramienta para la gestión de requisitos, tomando como base un estudio de las diferentes herramientas de este tipo que existen en Cuba y el mundo, así como de las tecnologías que la sustentan.

**Palabras claves:** Ingeniería de requisitos, Gestión de requisitos, Trazabilidad de requisitos, Requisitos funcionales, Requisitos no funcionales.

## **1. INTRODUCCIÓN**

La industria de software ha devenido un renglón importante en la economía de numerosos países. El hecho de que constituye una industria limpia cuyos pilares fundamentales son el capital humano que la desempeña y la tecnología, y que además no requiere grandes inversiones en materias primas, son algunos de los incentivos que

han provocado su creciente desarrollo. Cuba es uno de los países en vías de desarrollo que ha apostado por este sector.

La calidad es una necesidad prioritaria para lograr la inserción de los productos de software en el mercado. La etapa dedicada a la gestión de los requisitos es la de mayor importancia en el desarrollo de software, puesto que en ella se definen las necesidades de los clientes así como los requerimientos técnicos, normativos y legales que debe cumplir el producto final y por tanto son la base de su posterior desarrollo.

Una buena práctica para un mejor desempeño de esta fase es la utilización de herramientas que permitan automatizar la realización de las actividades que comprende. Existen diversas herramientas para la Gestión de Requisitos, tanto libres como propietarias pero ninguna de ellas satisface totalmente las necesidades de las pequeñas y medianas empresas, en especial las cubanas, ya sea porque tienen un limitado número de funcionalidades o porque no satisfacen los requisitos técnicos, legales o reglamentarios que establece el Ministerio de la Informática y las Comunicaciones (MIC), como órgano rector de la informática y las comunicaciones en Cuba.

La presente investigación pretende brindar una solución para mejorar el desempeño del área de procesos Administración de Requisitos, en las empresas cubanas de desarrollo de software. Para eso la investigación se centrará en el Proceso de Administración de Requisitos en proyectos de desarrollo de software y tendrá como objetivo la propuesta de una herramienta de software que permita automatizar las actividades de esta área de procesos. Para eso se hará una comparación de diversas herramientas de Gestión de Requisitos existentes en Cuba y el mundo así como las tecnologías que pudieran soportar la solución.

La herramienta en cuestión se encuentra en estos momentos en fase de implementación por parte de un equipo de proyecto de Calisoft (Centro de Calidad de Software).

## 1.1 MÉTODOS UTILIZADOS

Se usarán los métodos científicos empíricos:

- **Observación:** Observar el funcionamiento de diferentes herramientas existentes que permiten la Gestión de Requisitos e identificar funcionalidades y características a tener en cuenta para la posible construcción de un software con este propósito.

Se usarán además los métodos teóricos:

- **Analítico-Sintético:**

Para analizar los objetivos y actividades del área de procesos Administración de Requisitos, de diversos modelos de calidad y a partir de su descomposición obtener los requisitos necesarios para una herramienta que automatice la Gestión de Requisitos en las pequeñas y medianas empresas cubanas. Además para analizar las herramientas y plataformas existentes que pudieran soportar la implementación de la solución propuesta.

- **Análisis Histórico-Lógico:**

Para hacer un estudio del estado del arte de las herramientas existentes que permiten la gestión de requisitos, tanto en el ámbito nacional como internacional, así como las ventajas y desventajas que poseen.

Como posible resultado de la investigación se obtendrá una herramienta de software que permitirá la automatización de la Gestión de requisitos en las pequeñas y medianas empresas de Cuba.

## 1.2 CONCEPTOS Y CARACTERÍSTICAS ASOCIADOS AL DOMINIO DEL PROBLEMA

Existen múltiples conceptos asociados a los requisitos. Un requisito puede ser: "Una condición o capacidad que debe estar presente en un sistema o componente de un sistema para satisfacer un contrato, estándar, especificación u otro documento formal." (IEEE-SA 2006)

También puede ser identificado como: “Una declaración abstracta de alto nivel de un servicio que debe proporcionar el sistema o una restricción de este.” (Sommerville 2005)

A los efectos de la presente investigación un requisito será:

Una necesidad de los usuarios expresada como una condición o capacidad que debe cumplir un software.

Una de las prácticas más útiles de la Ingeniería de Requisitos es la **Trazabilidad de Requisitos**, que no es más que la habilidad para [1] representar las necesidades de los usuarios a través de requisitos [2] para identificar y seguir la instanciación de cada requisito en todos los productos de trabajo desde la fase de requisitos, diseño, desarrollo de los componentes del sistema hasta las pruebas y la documentación del sistema y [3] para establecer la correspondencia entre los requerimiento padre a uno hijo y viceversa. (Young 2004)

## **2. HERRAMIENTAS EXISTENTES PARA LA TRAZABILIDAD DE REQUISITOS**

En el mundo se han desarrollado diversas herramientas que facilitan la trazabilidad de requisitos, algunas de las cuales se caracterizan en las siguientes (Callejas, Castillo, y Fernández 2002). De ellas 4 son propietarias (IrqA, RequisitePro, Doors, Caliber RM) y 3 libres (REM, RETO y Controla).

**Tabla1: Comparación entre herramientas de gestión de requisitos**

Herramienta	Libre/propietaria	Plataforma	Extensibilidad	Trazabilidad	Integración con otras herramientas	Repositorio	Validación de la especificación	Generación de informes
<b>Irqa</b>	Propietaria	Windows	A través de APIs con Herramientas (COM, Java).	Entre requisitos y otros requisitos, elementos del dominio del problema, escenarios, elementos de la especificación de la solución, test, código fuente a través de la asociación con archivos externos, clases de implementación.	Together ControlCenter, Rational Rose Mercury TestDirector. Integración con MS Word, MS Excel.	MS-Access, SQL Server, Oracle, Informix, MySQL	Utiliza la matriz de trazabilidad	La presentación de informes puede generarse en diferentes formatos/plantillas. Informes Cristal Reports predefinidos, informes basados en plantillas MS Words.
<b>RequisitePro</b>	Propietaria	Windows	A través de APIs con Herramientas (COM, Java).	Trazabilidad entre los requisitos.	Integrada con las herramientas de IBM Rational: Rational Software Architect, Rational Software Modeler, Rational ClearCase, Rational ClearQuest, Rational SoDA, etc.	DB2, Microsoft SQL Server, Oracle, Microsoft Access	Se puede llevar a cabo con la matriz de trazabilidad entre la especificación de casos de uso (UCS) y documentos tipos Visión (VIS).	Crea y exporta matrices filtrables de trazabilidad y reportes de atributos para soportar los requisitos de documentación.

Herramienta	Libre/propietaria	Plataforma	Extensibilidad	Trazabilidad	Integración con otras herramientas	Repositorio	Validación de la especificación	Generación de informes
<b>DOORS</b>	Propietaria	Windows	Sí, API lenguaje DXL	Sí, entre cualquier elemento del repositorio	Mediante lenguaje DXL.	Propietario	Sí, con matriz de trazabilidad	La salida es adaptable a diferentes formatos
<b>CaliberRM</b>	Propietaria	Windows	Sí, API basado en COM y JAVA	Sí, entre los tipos de requisitos y otros elementos.	Office, Project, Modelling, Testing e IDE tools.	Microsoft Access, y SQL Server.	Sí, con matriz de trazabilidad	Sí, la salida puede adaptarse a diferentes formatos
<b>REM</b>	Libre	Windows	Sí, API, basada en XML y XSLT, genera HTML.	Sí, entre los tipos de requisitos y otros elementos.	----	Microsoft Access	Si, con matriz de trazabilidad.	Requisitos del sistema, análisis del sistema, registros y defectos, registro de peticiones de cambio.
<b>RETO</b>	Libre	Windows	Sí, API	Sí, entre modelo de requisitos y modelo conceptual	-----	-----	Sí, esquema conceptual de OO-Method	Especificación de requisitos del sistema.

Herramienta	Libre/propietaria	Plataforma	Extensibilidad	Trazabilidad	Integración con otras herramientas	Repositorio	Validación de la especificación	Generación de informes
<b>CONTROLA</b>	Libre	Windows	----	Trazabilidad entre los requisitos y otros elementos del sistema como casos de uso, casos de uso de prueba, implementaciones, liberaciones.	-----	Microsoft Access	Se realiza a través de una matriz de trazabilidad.	Genera documentos de plan de proyecto, descripción de los casos de uso, especificación de requisitos, casos de pruebas, indicadores, etc.

Como se puede apreciar en el análisis comparativo anterior, las herramientas propietarias presentan mayores funcionalidades y son más robustas, sin embargo sus altos costos hacen que no sean viables para su utilización en las pequeñas y medianas empresas cubanas. Por otra parte todas están sustentadas en la plataforma *windows*. Cuba sigue una política de migración a software libre, como lo expresa el acuerdo 84, de abril de 2004, del Consejo de Ministros. (Bencomo 2010) Siendo consecuentes con esta política es recomendable el uso de herramientas y plataformas libres para el desarrollo de la aplicación. Las herramientas libres analizadas tienen un conjunto más limitado de funcionalidades. Además solo funcionan sobre plataforma Windows. Por otra parte el repositorio del proyecto de las mismas está soportado sobre *Microsoft Access*, herramienta que es propietaria y de un limitado soporte para el almacenamiento de un número significativo de requisitos.

El análisis anterior conduce a la conclusión de que resulta necesario la implementación de una nueva herramienta para automatizar la Gestión de Requisitos, que satisfaga las necesidades propias de las empresas cubanas de desarrollo de software a la vez que cumpla con las disposiciones establecidas por el país, o sea una herramienta que sea multiplataforma y desarrollada según el paradigma de software libre.

### 3. ANÁLISIS DE LA SOLUCIÓN PROPUESTA

La siguiente lista de requisitos que deberá soportar la herramienta de Gestión de Requisitos propuesta, se hizo tomando como base las funcionalidades de las herramientas descritas en el epígrafe 2.

#### 3.1 REQUISITOS FUNCIONALES

- Gestionar proyectos (incluye insertar, modificar y eliminar proyecto)
- Gestionar usuarios

- Gestionar roles.
- Gestionar requisitos.
- Gestionar configuración (Incluye visualizar detalles de la configuración para un usuario determinado y establecer permisos a los roles para los diferentes flujos de trabajo).
- Generar matriz de trazabilidad.
- Gestionar cambios a los requisitos.
- Generar reportes.
- Gestionar glosario de términos.

### 3.2 REQUISITOS NO FUNCIONALES

- **Requisitos de Apariencia o Interfaz Externa:**
  - El sistema deberá poseer una interfaz web sencilla, amigable, lo más atractiva y clara posible para el usuario.
- **Requisitos de Uso:**
  - La aplicación garantiza una fácil interacción entre cliente y PC, de tal forma que no haya conflictos de usabilidad entre ambos. El sistema podrá ser usado por cualquier persona que posea conocimientos básicos en el manejo de la computadora y de un ambiente Web en sentido general.
- **Requisitos de Seguridad:**
- **Confidencialidad**
  - La información de un proyecto deberá ser confidencial. Solo tendrán acceso a la misma los usuarios involucrados en ese proyecto de acuerdo a los niveles de acceso que sean definidos en la configuración, así un usuario solo manejará la información correspondiente al rol que esté desempeñando.
- **Integridad**
  - El sistema debe garantizar que no haya corrupción de la información que soporta.
- **Disponibilidad**
  - El sistema debe estar disponible para su utilización las 24 horas del día, durante los siete días de la semana.
- **Requisitos de software:**
  - **Cliente**  
Debe tener instalado un navegador web que soporte JavaScript y Adobe Reader 5 o superior. Se recomienda Mozilla Firefox 3.5 o superior, Google Chrome 3.0 o superior, Internet Explorer 6 o superior, Opera 9 o superior o *Safari* 3 o superior. Debe tener instalado Sistema Operativo: Windows XP o superior, o GNU Linux.
  - **Servidor**  
Se debe instalar TOMCAT como servidor web y PostgreSQL como gestor de base de datos. Debe estar instalado el Java *Runtime Environment* (JRE) versión 1.6 o superior. Debe estar instalado Sistema Operativo: Windows XP o superior, o GNU Linux.
- **Requisitos de Hardware**
  - Para las PC clientes: procesador Pentium II o superior, 256 Mb de RAM. El servidor de aplicaciones debe tener las siguientes características: capacidad de disco duro superior a 80 GB, microprocesador Pentium IV superior a 2.0 GHz y como mínimo 1.0 GB de RAM.
  - El servidor de base de datos debe tener las siguientes características: capacidad de disco duro superior a 180 GB, microprocesador Pentium IV superior a 2.0 GHz y como mínimo 1.0 GB de RAM.

### 3.3 TECNOLOGÍAS A UTILIZAR PARA EL DESARROLLO DE LA HERRAMIENTA

### 3.3.1 LENGUAJE DE PROGRAMACIÓN

#### 3.3.1.1 PHP 5

PHP (*Hypertext Preprocessor*) es un lenguaje *script* para el desarrollo de páginas web dinámicas del lado del servidor, cuyos fragmentos de código se intercalan fácilmente en páginas HTML, debido a esto, y a que es de *Open Source* (código abierto), es el más popular y extendido en la web.

PHP tiene muchas ventajas algunas de ellas son:

- Multiplataforma: Inicialmente fue diseñado para entornos UNIX por lo que ofrece más prestaciones en este sistema operativo, pero es perfectamente compatible con Windows.
- Soporte para varios servidores Web. Mucha documentación (Ejemplos, manuales).
- Posee una sintaxis bastante clara.
- Fácil aprendizaje.
- Seguro.
- Capacidad de conexión con la mayoría de los motores de base de datos que se utilizan en la actualidad, destaca su conectividad con MySQL y PostgreSQL.
- Además PHP está orientado a objetos.

#### 3.3.1.2 JAVA

Es un lenguaje de programación sencillo, orientado a objetos, de propósito general e independiente de la plataforma de desarrollo. Java es muy amplio y variado. Este lenguaje posibilita la “Integración externa”, como el uso de herramientas, métodos y funcionalidades desarrolladas por otros programadores, esto supone una ventaja tanto en el ámbito del desarrollo como en la repercusión final de un proyecto. (García et al. 2000)

Las características principales que ofrece son:

- Es robusto: Realiza verificaciones en busca de problemas tanto en tiempo de compilación como en tiempo de ejecución. La comprobación de tipos ayuda a detectar errores, lo antes posible, en el ciclo de desarrollo.
- Es dinámico: Se beneficia todo lo posible de la tecnología orientada a objetos. No intenta conectar todos los módulos que comprenden una aplicación hasta el tiempo de ejecución.
- Es seguro: El sistema de Java tiene ciertas políticas que evitan que se puedan codificar virus con este lenguaje.

Por las potencialidades que ofrece, se recomienda utilizar como lenguaje de programación Java, además porque además de robusto y seguro se puede usar de manera fácil con el IDE seleccionado utilizando además los *frameworks* de desarrollo que se enunciarán a continuación.

### 3.4 FRAMEWORK DE DESARROLLO

#### 3.4.1 SPRING

Spring es un framework de aplicaciones Java/J2EE desarrollado usando licencia de OpenSource. Se basa en una configuración a base de JavaBeans bastante simple. Es potente en cuanto a la gestión del ciclo de vida de los componentes y fácilmente ampliable. (Sánchez 2006)

Características Básicas:



- Facilita una configuración y escritura de objetos muy centralizada y automatizada. Es compatible con ambientes de desarrollo J2SE y J2EE.
- Contiene una capa común de abstracción para el manejo de las transacciones que permite conectar los administradores de las mismas, haciendo que sea más fácil para demarcar las operaciones sin tratar con bajo nivel de cuestiones.

### 3.4.2 HIBERNATE

Hibernate es una solución ORM (Mapeo Objeto-Relacional) <sup>1</sup> para Java. Este framework que busca solucionar el problema de la diferencia entre el modelo orientado a objetos y el usado en las bases de datos modelo relacional mediante archivos declarativos, es una capa de persistencia objeto-relacional y un generador de sentencias SQL (Structured Query Language). Permite diseñar objetos persistentes que podrán incluir polimorfismos, relaciones, colecciones, y un gran número de tipos de datos. Hibernate convierte los datos entre los tipos utilizados por Java y los definidos por SQL. Permite una fácil integración con cualquier otro Framework de Java y para su uso no requiere de muchas reglas específicas o patrones. (Rosés 2003)

## 3.5 GESTOR DE BASE DE DATOS

### 3.5.1 MYSQL

MySQL es un sistema de gestión de base de datos relacional. Opera en una arquitectura cliente/servidor. Las principales características de este gestor de bases de datos son las siguientes:

- Soporta gran cantidad de tipos de datos para las columnas.
- Dispone de API's en gran cantidad de lenguajes (C, C++, Java, PHP).
- Relativamente sencillo de añadir otro sistema de almacenamiento.

Desventajas de MySQL:

- Carece de soporte para transacciones y subconsultas.
- No es viable para su uso con grandes bases de datos, a las que se acceda continuamente, ya que no implementa una buena escalabilidad. («MySQL... » 2012)

### 3.5.2 POSTGRESQL

Es un sistema de base de datos relacional perteneciente al ámbito del software libre que se destaca por su robustez, escalabilidad y cumplimiento de los estándares SQL. Cuenta con versiones para una amplia gama de sistemas operativos, entre ellos: Linux, Windows, Mac SX, Solaris y otros más.

Características de PostgreSQL

- Soporta distintos tipos de datos.
- Incluye herencia entre tablas (aunque no entre objetos, ya que no existen), por lo que a este gestor de bases de datos se le incluye entre los gestores objeto-relacionales.
- Permite la gestión de diferentes usuarios, así como también los permisos asignados a cada uno de ellos. («PostgreSQL: Welcome» 2012)

Para la implementación de la herramienta recomienda utilizar como sistema gestor de base de datos a PostgreSQL, además de por su demostradas fiabilidad, robustez y seguridad para todo tipo de sistemas

---

<sup>1</sup> **ORM (object/relationalmapping) Mapeo Objeto-Relacional:** El mapeo de objetos relacionales es la persistencia automática y transparente de objetos en una aplicación Java a tablas en una base de datos relacional, usando metadatos que describen el mapeo entre objetos y la base de datos, ORM, en esencia, trabaja transformando datos de una representación a la otra.

informáticos, es software libre, siendo coherentes con las políticas trazadas en el país. Por otra parte en la Universidad de las Ciencias Informáticas se cuenta con un grupo de desarrollo de este gestor, lo cual garantiza que continúe su desarrollo, perfeccionamiento y soporte.

### 3.5.3 CONSIDERACIONES DE LA SOLUCIÓN

La herramienta propuesta se encuentra enmarcada en el contexto del Modelo Cubano de Calidad de Software, una iniciativa llevada a cabo por un equipo de profesionales de la UCI y de Calisoft, que pretende crear un modelo de calidad específico para las necesidades de las Pymes (pequeñas y medianas empresas) cubanas, por lo que la extensibilidad de dicha herramienta puede estar limitada a organizaciones con características similares a estas empresas, tales como: cantidad de personas involucradas, roles y procesos de ingeniería de requisitos definidos, etc.

## 4. CONCLUSIONES

Se realizó un análisis de las diferentes herramientas, tanto propietarias como libres existentes para la Gestión de Requisitos y se corroboró que ninguna es adecuada para las necesidades de las pequeñas y medianas empresas de desarrollo de software en Cuba, ya sea por los costos de las mismas, cuestiones legales y reglamentarias o de políticas propias del entorno o por las escasas funcionalidades y limitaciones desde el punto de vista tecnológico. Por estos motivos se determinó que resulta necesaria la implementación de una herramienta que cumpla con las expectativas planteadas. Se determinaron los requisitos de dicha aplicación así como se analizaron las tecnologías y herramientas necesarias para su implementación.

## 5. REFERENCIAS

- Bencomo, Edel. 2010. «Legislación Informática de República de Cuba». *Informática-Jurídica*. Mayo. <http://www.informatica-juridica.com/legislacion/cuba.asp>.
- Callejas, M, L Y Castillo, y R M Fernández. 2002. «HELER: Una herramienta para la Ingeniería de Requisitos Automatizada.» *Entramado*, diciembre.
- García, Javier, José Ignacio Rodríguez, Iñigo Migno, Aitor Imaz, Alfonso Brazález, Alberto Larzabal, Jesús Calleja, y Jon García. 2000. «Aprenda JAVA como si estuviera en primero». San Sebastian.
- IEEE-SA. 2006. «IEEE Standard Glossary of Software Engineering Terminology». [http://standards.ieee.org/reading/ieee/std\\_public/description/se/610.12-1990\\_desc.html](http://standards.ieee.org/reading/ieee/std_public/description/se/610.12-1990_desc.html).
- «MySQL :: The world's most popular open source database». 2012. abril 13. <http://www.mysql.com/>.
- «PostgreSQL: Welcome». 2012. abril 13. <http://www.postgresql.org/>.
- Rosés, F. 2003. «Introducción a Hibernate».
- Sánchez, M.A. 2006. «Sistema de administración y control de renta de películas y libros vía web utilizando Spring». Escuela de Ingeniería y Ciencias Departamento de Computación, Electrónica, Física e Innovación: Universidad de las Américas Puebla.
- Sommerville, Ian. 2005. *Ingeniería de Software*. Mexico DF: Pearson.
- Young, Ralph R. 2004. *The Requirements Engineering Handbook*. London: Artech House.

## AUTORIZACIÓN Y RENUNCIA

Como autor autorizo a LACCEI a publicar este artículo en las conferencias del evento. Ni LACCEI ni los editores son responsables por el contenido expresado en el presente artículo ni las implicaciones que pueda tener el mismo.

### *Authorization and Disclaimer*

*Authors authorize LACCEI to publish the paper in the conference proceedings. Neither LACCEI nor the editors are responsible either for the content or for the implications of what is expressed in the paper.*