

Administración y trazabilidad inteligente de requerimientos

Bersano, Diego Matías

Universidad Católica de Santiago del Estero, Departamento Académico Rafaela

diegobersano33@gmail.com

Resumen

La realización de un proyecto para lograr un software es una tarea compleja, debido a las diversas actividades que deben realizarse para lograr el objetivo. Una de las principales tareas para la correcta finalización del mismo es la gestión de requerimientos, debido a que a partir de esto se definirá el alcance final y preciso del software a construir. Uno de los resultados de esta gestión es la matriz de trazabilidad de los requerimientos, la cual brinda las relaciones existentes entre los mismos. A medida que la cantidad de los requerimientos aumenta se hace más dificultosa su gestión sin una asistencia adecuada. Las numerosas relaciones entre los requerimientos deben gestionarse manualmente, tanto sobre los iniciales del proyecto como los que surjan a lo largo de la vida del software. En el marco de estas problemáticas y como Trabajo Final de Grado de la carrera Ingeniería en Informática de la Universidad Católica de Santiago del Estero, Departamento Académico Rafaela, se decidió desarrollar una herramienta que brinde una ayuda semi-automática en la realización de la gestión de los requerimientos y trazabilidad, mediante la utilización de técnicas de búsqueda y recuperación de información (Information Retrieval).

Palabras claves: gestión de requerimientos de software, trazabilidad, gestión de cambios, pruebas por requerimientos.

Justificación del proyecto

Las problemáticas inherentes a la gestión de requerimientos han sido destacadas a lo largo de distintas cátedras correspondientes a la carrera Ingeniería en Informática. En las mismas se la destacaba en repetidas ocasiones como una tarea compleja e importante, no solo por la cantidad de trabajo necesario a realizar y su complejidad inherente, sino también el impacto que tiene en el resultado final. La correcta

realización de la gestión de requerimientos permite que ante algún cambio se pueda tener un panorama claro del impacto que tendría el mismo en los demás requerimientos relacionados. La falta de relaciones entre requerimientos puede llegar a tener un impacto muy alto en el proyecto, ya que se podrían producir incumplimientos en los requisitos del usuario, lo cual implica realizar nuevamente diversas tareas. Hasta potencialmente estas situaciones podrían poner en riesgo el éxito final del proyecto.

La mayor parte de las herramientas de gestión de requerimientos de tipo comercial se enfocan en otros aspectos (los cuales también son útiles y necesarios para el éxito de un proyecto) como lo son la colaboración, modelado de artefactos que surgen del análisis de los requerimientos del usuario, importación del contenido de otros formatos, diferenciación visual de los cambios que se producen entre determinadas versiones, entre otros, no considerando como aspecto funcional la asistencia en el establecimiento de relaciones de trazabilidad.

No obstante, se han realizado diversos avances en los últimos años en el reconocimiento automático de relaciones de trazabilidad de requerimientos a partir de diversos métodos, los cuales generalmente están fundamentados en la técnica *Information Retrieval*. Dicha técnica se base en la comparación del contenido de los documentos. A continuación se detallan algunos proyectos realizados sobre esta área del conocimiento, los cuales se utilizaron como referencia para la realización de este trabajo final:

- ADAMS (ADvanced Artefact Management System) [1]: es un sistema de soporte de procesos basado en artefactos. Plantea un gran énfasis en el ciclo de vida de los artefactos, asociando ingenieros de software con las diferentes operaciones que se pueden realizar en cada uno.
- RETRO (REquirements TRacing On-target) [2]: es una herramienta de propósito especial diseñada exclusivamente para las necesidades de seguimiento. Puede ser utilizada como una herramienta independiente para descubrir matrices de trazabilidad, aunque también se puede utilizar en conjunto con otro software de gestión de proyectos. En el núcleo de RETRO se encuentra una colección de las

implementaciones de los métodos de IR adaptados a los efectos de los requisitos de seguimiento de tareas. Se accede a estos métodos el bloque de GUI (Java) para filtrar y analizar los documentos de requisitos de entrada y construir criterios de juicio.

Objetivos

El objetivo de este trabajo final es lograr una herramienta que consiga combinar características de los dos ámbitos descritos anteriormente, tanto las funcionalidades de las herramientas de tipo comercial como nuevas características innovadoras que se aplicaron en los proyectos investigativos mencionados anteriormente. Esto da como resultado la realización de una aplicación web para la gestión de requerimientos de forma inteligente, la cual se denomina ReqT. El propósito de la misma es poder administrar los requerimientos de los proyectos software que lleve adelante una empresa, permitiendo gestionar el cambio y evolución de los mismos, los casos de pruebas necesarios para validar la correctitud, y las relaciones de trazabilidad que se produzcan entre sí. Para este cometido en particular es donde se han aplicado las técnicas de recuperación de la información (*Information Retrieval*).

Se espera que esta aplicación web logre de forma efectiva ser una herramienta de uso real en tareas asociadas al análisis de software para cualquier profesional del área, dentro del alcance definido para su realización.

A su vez desde el aspecto técnico de desarrollo de la aplicación, se plantea la realización de un software escalable, con una arquitectura planificada desde el momento inicial en la cual se tiene en consideración el funcionamiento global que desea lograrse. Así mismo, debido a la utilización de metodologías ágiles, dicha arquitectura debe ser susceptible a las modificaciones que surjan a medida que el proyecto avance.

Marco teórico

A continuación se exponen las definiciones que se consideran como base para la realización y correcta comprensión de este trabajo:

Requerimiento [3, pp. 108-110]: los requerimientos para un sistema son la descripción de los servicios proporcionados por el sistema y sus restricciones

operativas. Estos requerimientos reflejan las necesidades de los clientes de un sistema que ayude a resolver algún problema como el control de un dispositivo, hacer un pedido o encontrar una información. Podemos diferenciar dos tipos de requerimientos según su nivel de descripción: requerimientos del usuario y del sistema. A su vez, los podemos clasificar como requerimientos funcionales, no funcionales y del dominio.

Trazabilidad o rastreo [3, pp. 148-149]: existen muchas relaciones entre los requerimientos y entre éstos y el diseño del sistema. También existen vínculos entre los requerimientos y las razones fundamentales por las que éstos se propusieron. Cuando se proponen cambios, se debe rastrear el impacto de estos cambios en los otros requerimientos y en el diseño del sistema. El rastreo es una propiedad de la especificación de requerimientos que refleja la facilidad de encontrar requerimientos relacionados.

Pruebas basadas en requerimientos [3, pp. 505-506]: un principio general de los requerimientos es que deberían poder probarse. Es decir, los requerimientos deberían ser escritos de tal forma que se pueda diseñar una prueba para que un observador pueda comprobar que los requerimientos se satisfacen. Generalmente las pruebas de un requerimiento no significan escribir sólo una única prueba.

Recuperación de información (Information Retrieval) [4, p. 1]: la recuperación de información (IR) es el material encontrado (generalmente documentos) de naturaleza no estructurada (generalmente texto) que satisface una necesidad de información tomando como origen grandes colecciones de la misma (normalmente almacenada en computadoras).

Matriz de trazabilidad o matriz de rastreo [3, p. 149]: la información de rastreo se representa utilizando matrices de rastreo, las cuales relacionan los requerimientos con los stakeholders, con los módulos del diseño o los requerimientos entre ellos.

Gestión del cambio de los requerimientos [3, pp. 150-151]: la gestión del cambio de los requerimientos se debe aplicar a todos los cambios propuestos en los requerimientos. La ventaja de utilizar un proceso formal para gestionar el cambio es que todos los cambios propuestos son tratados de forma consistente y que los cambios

en el documento de requerimientos se hacen de forma controlada. Existen tres etapas principales en un proceso de gestión de cambio: análisis del problema y especificación del cambio, análisis del cambio y cálculo de costes e implementación del cambio.

Metodología utilizada

Debido a las características de este proyecto, el cual persigue como objetivo principal la construcción de un Producto de Software e implica fundamentalmente la utilización de una metodología de desarrollo, no es posible realizar una clasificación en todas las categorías de proyectos de investigación que remarca [5]. Sin embargo se pueden establecer las siguientes:

- Aplicada: se caracteriza por ser un proyecto orientado a los conceptos de una investigación aplicada, porque se pretende desarrollar una solución a un determinado problema práctico, definido y delimitado.
- Documental: debido a que la base de los conocimientos se realiza a través de la consulta de documentos (libros, papers, publicaciones y documentación de proyectos realizados).
- Exploratoria: “Los estudios exploratorios se efectúan, normalmente, cuando el objetivo es examinar un tema o problema de investigación poco estudiado o que no ha sido abordado antes” [5, p. 49]. Si bien el objetivo de este proyecto es trabajar sobre un tema que ya ha sido abordado anteriormente por investigadores y cuyos resultados se utilizan como referencia en la realización de este trabajo, considero que es posible realizar un aporte o avance sobre la problemática, realizando una aplicación que presenta una perspectiva distinta a la ya abordada.
- Transversal: la realización de este proyecto está vinculada con un momento específico del tiempo, no con una observación a lo largo del tiempo de determinados sucesos.

Para la realización de este trabajo final se ha elegido Scrum como marco de trabajo. “Scrum es un marco de trabajo dentro del cual las personas pueden afrontar problemas complejos adaptativos, a la vez que entregan productos del máximo valor posible de forma productiva y creativa.” [6, p. 3] El motivo de la elección del mismo

se basa en su enfoque, ya que plantea que las actividades a realizar giran en torno al producto. A su vez, la delimitación de Sprints en los cuales se debe entregar un incremento plenamente funcional, es muy útil para este trabajo final, ya que además de plantear un ritmo de trabajo constante en el cual se debe cumplir con entregas de calidad, se obtiene una retroalimentación constante del director y de los tutores del proyecto. Esto sin duda es muy provechoso, ya que evita el re-trabajo.

Productos generados

Debido a que el objeto de realización de este trabajo final es un producto software de calidad, ha sido necesaria la realización de un plan de proyecto. De esta forma se establece un alcance delimitado, y funcionalidades necesarias y realizables para los requisitos de un trabajo final de carrera. A su vez se cuenta con un documento claro y definido para comunicar el producto a realizar tanto al director del trabajo final como a los tutores. En el mismo se han especificado los siguientes aspectos, utilizando como base las prácticas de la administración de proyectos:

- Ciclo de vida
- WBS del producto
- Calendario de actividades
- Plan de recursos humanos
- Plan de costos
- Gestión de riesgos
- Plan de gestión de las comunicaciones
- Plan de adquisiciones
- Gestión de la calidad
- Monitoreo y control del proyecto

Si bien la utilización de Scrum no prescribe la utilización de determinados documentos de análisis y diseño, debido a las características del trabajo final se ha optado por realizar los documentos correspondientes que den soporte en este aspecto. Para ello se ha trabajado con el lenguaje unificado del modelado (UML) [7]. La elección de este lenguaje de modelado está basada en que al momento de realizar este proyecto se lo considera como la referencia para los procesos de desarrollo orientados

a objetos, situación que se da con este proyecto final. Adicionalmente de que es ampliamente utilizado en la industria del software, reconocido por sus amplias cualidades.

Según lo definido, se han realizado los diagramas UML que se detallan a continuación:

- diagramas de casos de uso para detallar las funcionalidades que deberá brindar cada módulo de la aplicación.
- diagramas de actividades utilizados para detallar los flujos más complejos de la aplicación.
- diagramas de estados para reflejar el ciclo de vida de determinadas entidades de la aplicación.
- diagrama de clases para detallar las entidades que se utilizan en la aplicación.
- diagramas de secuencias para mostrar la relación e intercambio de mensajes entre las distintas capas que componen la aplicación.
- diagrama de componentes para reflejar la estructura de los distintos componentes que componen la aplicación.

Bajo la utilización de dicho lenguaje, se realizó el diagrama de clases representado en la *Fig. 1*, el cual muestra las entidades y relaciones que se consideraron necesarias para la resolución de la problemática planteada con el alcance definido.

Para la codificación del software se utilizó el lenguaje C# y se diseñó una aplicación web bajo la utilización del framework ASP.NET MVC con una arquitectura de 3 capas (datos – servicios – web) utilizando para ello 4 proyectos dentro de una solución de Visual Studio, el cual se utilizó como entorno de desarrollo para este trabajo final:

- Entidades: contiene las clases de dominio de la aplicación (las cuales se graficaron en el diagrama de clases).
- Datos: contiene la conexión con la base de datos mediante el uso de la librería EntityFramework como ORM.

- Servicios: contiene la implementación de la lógica de negocio, siendo la interfaz entre la capa de datos y la web. Es esta capa la que contiene los algoritmos para la trazabilidad de requerimientos.
- Web: es el sitio ejecutable de la aplicación, el cual interactúa con los servicios.

La particularidad de la estructura en capas es que establece como restricción que cada capa puede acceder solo a la que la precede (ej.: la capa web puede acceder a la de servicios, y la de servicios a la de datos). En este caso particular, el proyecto de entidades será compartido por todas las capas, por eso su no inclusión en ninguna de las capas mencionadas anteriormente. De esta forma se logra un alto grado de independencia entre cada una de las capas, siendo escalable, modularizable y tolerante a los cambios.

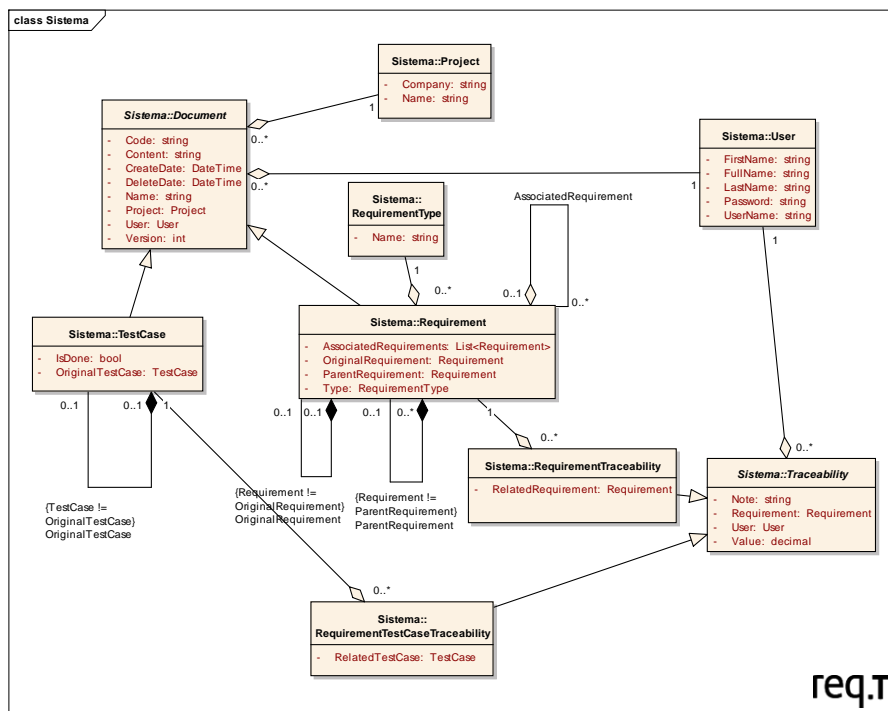


Fig. 1. Diagrama de clases de la aplicación

Como estándar de desarrollo interno se ha decidido la utilización del idioma inglés, bajo la premisa de obtener un código más legible y prolijo. A su vez se han aplicado los principios SOLID en varios casos donde se lo consideraba necesario.

Uno de los objetivos de este trabajo final es la implementación de un algoritmo que permita la trazabilidad inteligente de requerimientos, la cual sea una asistencia para el profesional del área que deba llevar adelante tal actividad. Para llevar adelante este cometido se partió de una base existente de código que aplica el algoritmo LSI (*Latent Semantic Index*), propiedad de Anup Shinde [8]. El mismo se encuentra publicado de forma gratuita, con permisos de que dicho código sea utilizado y modificado.

En esta implementación en particular, lo que se ha realizado fue adaptar la organización de este código fuente para que responda al nivel de calidad que se desea lograr con este trabajo, modularizando y separando funcionalidades para lograr una mayor independencia entre las partes que lo componen. Una de las características que requiere este algoritmo es la comparación de las raíces de las palabras, obteniendo de esta forma una mayor precisión en los resultados que, luego del procesamiento, se mostrarán al usuario de la aplicación. Para tal tarea se ha decidido utilizar el algoritmo de Porter, el cual es el más utilizado para el cumplimiento de esta funcionalidad. Si bien el código original utilizaba una implementación de este algoritmo en inglés, a partir de la implementación realizada por Paolo Ragone [9] del algoritmo en español y en el lenguaje de programación PHP, se ha desarrollado una equivalencia del mismo al lenguaje C#, la cual ha resuelto de forma correcta un conjunto de prueba de 28390 palabras. En el **Apéndice B** se incluye una explicación del método LSI utilizado para la recuperación de la información, junto a sus fundamentos matemáticos.

Como resultado de los puntos descriptos anteriormente, se ha logrado la realización de la aplicación resultante. La misma se compone de 3 módulos, cada uno de los cuales engloba un conjunto de funcionalidad relacionada:

- **Requerimientos:** este módulo es el principal de la aplicación. El mismo permite la administración general de los requerimientos.

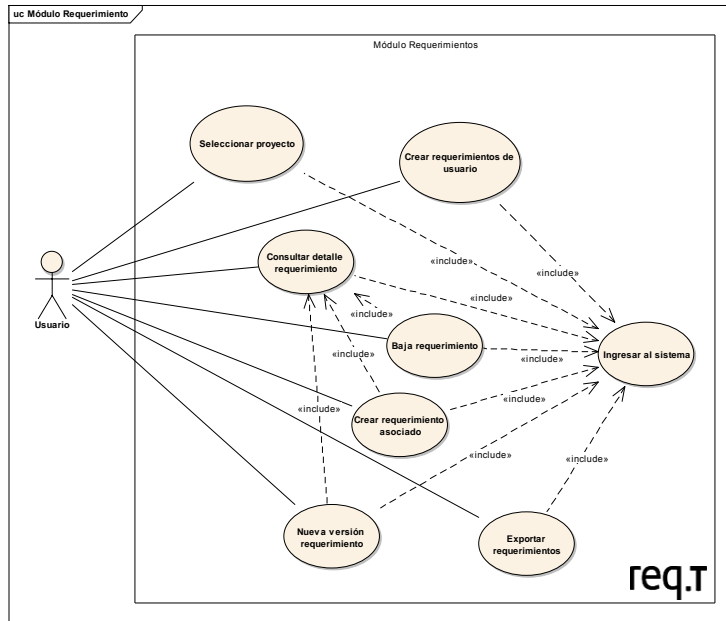


Fig. 2. Diagrama de casos de uso del módulo de requerimientos

- Casos de test: este módulo es el complemento sobre los requerimientos, lo cual permite que a partir de las relaciones de trazabilidad se pueda determinar cuándo un determinado requerimiento ha sido verificado.

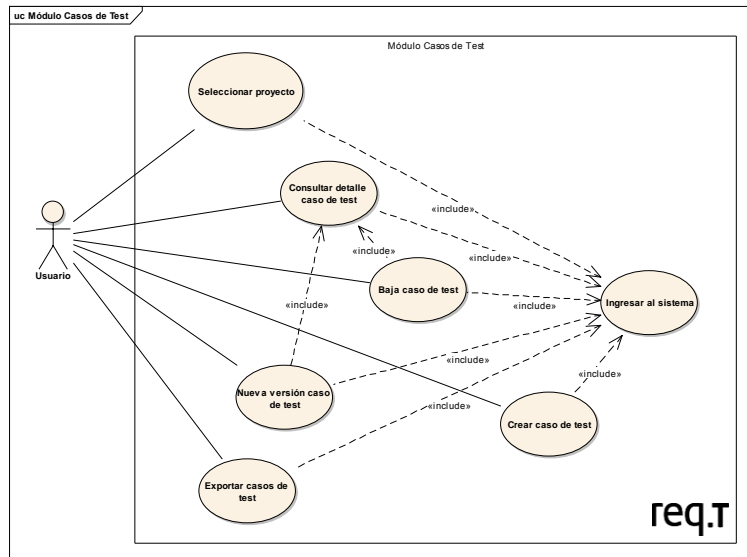


Fig. 3. Diagrama de casos de uso del módulo de casos de test

- Trazabilidad: este módulo contiene el componente innovador que se ha planteado en este trabajo final. En el mismo se podrán establecer las relaciones de trazabilidad de requerimientos, contando con la asistencia de documentos relacionados (tanto casos de test como otros requerimientos) a partir de su contenido textual.

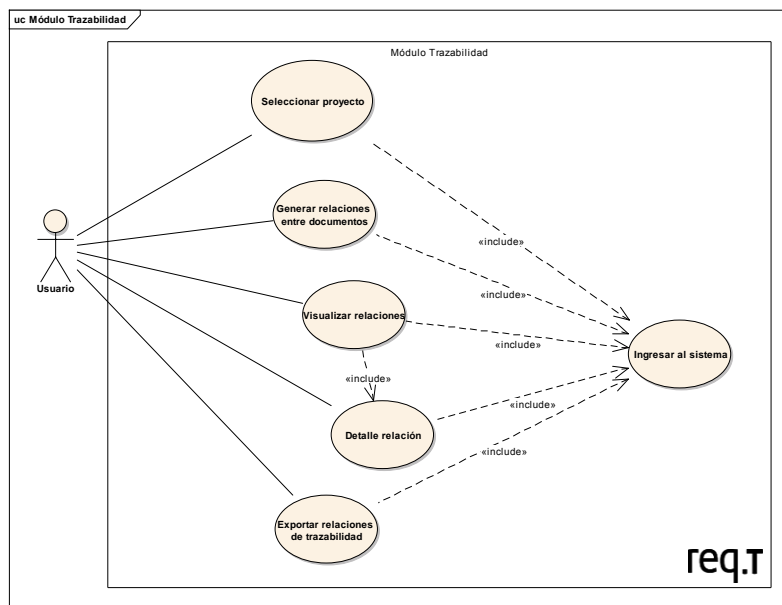


Fig. 4. Diagrama de casos de uso del módulo de trazabilidad

Características de la aplicación

En la Fig. 5 se muestra el flujo básico de funcionamiento que tiene la aplicación, comenzando desde el punto inicial que para la misma se considera como base, que es un requerimiento de usuario. A partir del mismo, se podrán crear los requerimientos funcionales y no funcionales asociados y realizar la búsqueda de documentos relacionados (tanto otros requerimientos funcionales, no funcionales o casos de test), previo ingreso del contenido del requerimiento en cuestión.

Las relaciones de trazabilidad se podrán crear en conjunto con el requerimiento, como en cualquier momento posterior (incluyendo la edición de las relaciones ya existentes). En la búsqueda de documentos relacionados se incluye tanto los

resultados encontrados de forma asistida mediante las técnicas de *Information Retrieval*, como las relaciones que el usuario establece de forma manual.

Contando con la información de trazabilidad, se hace posible realizar un análisis de impacto sobre cada requerimiento funcional o no funcional, visualizando de forma jerárquica y recursiva a partir de un documento (de los tipos mencionados anteriormente), cuales son todos los artefactos relacionados de forma directa o indirecta.

En el módulo de casos de test, podrán crearse los mismos, además de marcarlos como realizados. Esto permite que para un requerimiento dado (funcional o no funcional) se pueda saber si está validado. Dicha situación se dará si todos los casos de test relacionados están realizados.

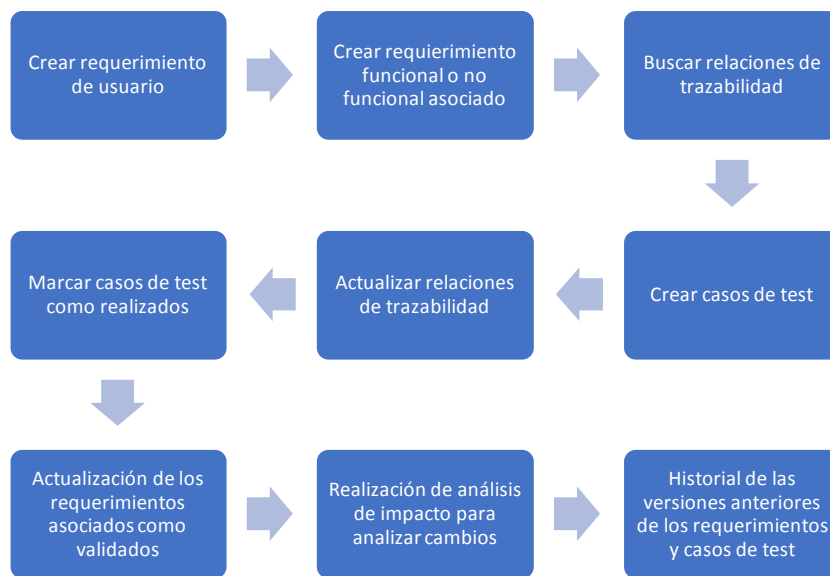


Fig. 5. Diagrama de funcionamiento de la aplicación

Para cada documento de la aplicación (todos los tipos de requerimientos y los casos de test) podrán visualizarse las versiones previas si existiesen, sin limitación en su cantidad.

La información que se generará en los módulos descritos anteriormente se englobará en torno a proyectos. El usuario que esté utilizando la aplicación podrá seleccionar el proyecto sobre el cual trabajar, viendo en cada módulo la información correspondiente. A su vez se brindan las posibilidades de administración tanto de usuarios como de proyectos. Estas son dos funcionalidades anexas, que sirven de apoyo a las funcionalidades específicas que se han realizado, logrando el objetivo de una aplicación utilizable en casos reales.

Finalmente, al realizar un aplicativo se consideró necesaria la realización de un manual de usuario. La forma en la que se ha decidido implementarlo es mediante una documentación online, la cual estará disponible al acceder a la aplicación. En la misma se explica de forma simple y con imágenes de la aplicación cuales son las funcionalidades que ofrece, el acceso a las mismas y las consideraciones básicas a tener en cuenta en su uso. En el *Apéndice A* se explica de forma más detallada el funcionamiento general de la aplicación, además de la inclusión de capturas de la misma.

Conclusiones

Luego de finalizar el desarrollo de este trabajo final, es posible concluir que los objetivos planteados se han podido cumplir de forma satisfactoria.

En primera medida, uno de los puntos más críticos como proyecto de desarrollo de software era el tiempo estipulado para la finalización del mismo, cumpliendo todo lo definido en el alcance inicial y manteniendo la calidad en el producto resultante. Este objetivo ha sido exitosamente alcanzado, ya que adicionalmente como producto de la retroalimentación con el director y tutores planteada en el plan de proyecto, se han agregado funcionalidades (generalmente de soporte a las funciones principales de la aplicación) que inicialmente no estaban planeadas.

La meta principal del proyecto fue el lograr una herramienta para la administración y trazabilidad asistida de requerimientos de software. Esto también pudo ser logrado, ya que se desarrolló una herramienta que permite gestionar los distintos tipos de requerimientos de software, sus cambios, relaciones de trazabilidad y verificación mediante casos de pruebas asociados de forma sencilla para un usuario profesional del

área. Enfocándonos en la trazabilidad, aspecto clave en este trabajo final de carrera, al aplicar los algoritmos necesarios se ha podido ver que los resultados obtenidos eran acordes a lo esperado si la realización de tal tarea hubiera sido de forma manual. Esto permite demostrar que los algoritmos de *Information Retrieval*, aplicado como ámbito delimitado a las relaciones de trazabilidad de un producto software, son implementables y usables en una aplicación real, logrando una ayuda efectiva en la gestión de requerimientos de software.

Finalmente, se pueden establecer las siguientes líneas futuras a partir de lo realizado en este trabajo final de carrera:

- Mejora de los resultados: los algoritmos aplicados para la recuperación y posterior comparación de documentos para obtener las relaciones de trazabilidad son susceptibles a dos tipos de errores: no recuperar documentos que si están relacionados y recuperar documentos que no están relacionados como si realmente lo estuvieran (esto último denominado comúnmente como falsos positivos). Ambos errores son propios de los algoritmos utilizados, por lo que se propone la realización de un trabajo investigativo, cuyo objetivo sea determinar cómo lograr una reducción de ambos.
- Realizar la trazabilidad con el código fuente: en el desarrollo de este trabajo final se ha limitado el alcance a la trazabilidad entre requerimientos y casos de test. Sin embargo, también resultaría factible realizar esta misma trazabilidad con el código fuente del proyecto, ampliando de forma significativa la cantidad de información disponible y lo que se podría realizar con ella.
- Implementar otro algoritmo para la trazabilidad: podría utilizarse como base la aplicación desarrollada e implementar otro algoritmo para la trazabilidad, el cual se considere que sea más apropiado o de mejores resultados.

Bibliografía

- [1] A. De Lucia, F. Fasano, R. Oliveto y G. Tortora, Recovering Traceability Links in Software Artefact Management Systems using Information Retrieval Methods, Fisciano: Universidad de Salerno, 2001.

- [2] J. Huffman Hayes, A. Dekhtyar y S. Karthikeyan Sundaram, Advancing Candidate Link Generation for Requirements Tracing: the Study of Methods, Universidad de Kentucky, 2006.
- [3] I. Sommerville, Ingeniería del Software (séptima edición), Madrid: Pearson Educación, 2005.
- [4] C. Manning, P. Raghavan y H. Schütze, An Introduction to Information Retrieval, Cambridge: Cambridge University Press, 2009.
- [5] R. Hernández Sampieri, C. Fernández Collado y P. Baptista Lucio, Metodología de la Investigación, McGraw-Hill, 1991.
- [6] K. Schwaber y J. Sutherland, «Scrum.org,» Octubre 2011. [En línea]. Available: http://scrumguides.squarespace.com/s/Scrum_Guide-2011-ES.pdf. [Último acceso: 04 11 2013].
- [7] J. Rumbaugh, I. Jacobson y G. Booch, El lenguaje unificado de modelado - Manual de referencia, Madrid: Pearson Educación S.A., 2000.
- [8] A. Shinde, «Anup Shinde,» [En línea]. Available: <http://www.anupshinde.com/posts/latent-semantic-indexing/>. [Último acceso: 10 10 2013].
- [9] P. Ragone, «Stemmer-es,» [En línea]. Available: <http://stemmer-es.sourceforge.net/>. [Último acceso: 10 10 2013].
- [10] B. Michael, D. Susan y O. Gavin, Using Linear Algebra for Intelligent Information Retrieval, Escuela Politécnica Federal de Lausana - Departamento de las Ciencias de la Computación, 1994.

Apéndice A: descripción del producto software realizado

Anteriormente se ha incluido brevemente una descripción de la aplicación realizada como resultado del trabajo final. El objetivo de este apéndice es explicar con mayor detalle el producto software realizado.

La aplicación resultante se compone de 3 módulos, cada uno de los cuales engloba un conjunto de funcionalidad relacionada:

- *Requerimientos:* este módulo es el principal de la aplicación. El mismo permite la administración general de los requerimientos.
- *Casos de test:* este módulo es el complemento sobre los requerimientos, lo cual permite que a partir de las relaciones de trazabilidad se pueda determinar cuándo un determinado requerimiento ha sido verificado.
- *Trazabilidad:* este módulo contiene el componente innovador que se ha intentado lograr con este trabajo final. En el mismo se podrán establecer las relaciones de trazabilidad de requerimientos, contando con la asistencia de documentos relacionados (tanto casos de test como otros requerimientos) a partir de su contenido textual.

La información que se generará en dichos módulos se englobará en torno a proyectos. El usuario que esté utilizando la aplicación podrá seleccionar el proyecto sobre el cual trabajar, viendo en cada módulo la información correspondiente.

A su vez, más allá de los módulos descritos anteriormente, se brindan las posibilidades de administración tanto de usuarios como de proyectos. Estas son dos funcionalidades anexas, que permiten el apoyo a las funcionalidades específicas que se han realizado.

Módulo de Requerimientos

Como se comentó de manera breve anteriormente, este módulo es el central de la aplicación resultante en cuanto a las funcionalidades que provee.

El planteo que se ha decidido adoptar para la realización de este módulo es englobar tanto los requerimientos funcionales como los no funcionales bajo requerimientos de

usuario. El motivo de esta elección es la consideración de que toda necesidad de implementar una determinada funcionalidad en un proyecto de software debería surgir del planteamiento del cliente para la satisfacción de una necesidad propia bien definida.

req.T | Requerimientos | Casos de Test | Trazabilidad | Agenda Contactos | Diego Bersano

Requerimientos de usuario

+ Nuevo requerimiento de usuario | Descargar especificación de requerimientos

Filtrar resultados

Nombre de usuario: | Fecha desde: | Fecha hasta: | Validados: |

Código	Nombre	Acciones
RU001	Mantenimiento y Gestión de nuestros contactos personales	<input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>
RF001	El usuario podrá agregar un nuevo contacto a la agenda	Requerimiento funcional Verificado <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>
RF002	Podrá eliminar contactos existentes en la agenda	Requerimiento funcional <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>
RF003	Podrá modificar la información almacenada para un contacto determinado	Requerimiento funcional <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>
RF004	Podrá consultar la información de un contacto concreto almacenado	Requerimiento funcional <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>
RU002	Gestión de préstamos a los contactos	<input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>
RU003	Impresión de una lista de contactos	<input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>

Fig. 6. Módulo Requerimientos

Sobre todo tipo de requerimiento es posible la creación de nuevas versiones, debido a que las necesidades pueden ir cambiando a lo largo del tiempo, y consecuentemente debe hacerlo el producto que se desea crear. Esto se realiza manteniendo un historial de la versión anterior a medida que el usuario guarda los cambios sobre dichos artefactos (sin establecer un límite en la cantidad de versiones anteriores que pueden mantenerse). Ante esto, se asocian las relaciones de trazabilidad de la versión original a la nueva, sin mantenerse un historial de las mismas.

Módulo de Casos de Test

Este módulo es complementario al de requerimientos. Su función específica es poder integrar mediante los aspectos de trazabilidad la indicación de cuando un determinado requerimiento ha alcanzado su verificación. Esto se producirá cuando todos los casos de test relacionados por trazabilidad a un determinado requerimiento hayan sido

marcados como realizados. Cabe aclarar que la verificación de la realización real del caso de test queda fuera del ámbito de esta aplicación, en la cual solo se le indicará cuando se ha producido esta situación para reflejar la información resultante.

Fig. 7. Módulo Casos de Pruebas

Módulo de Trazabilidad

Este módulo es el encargado de gestionar las relaciones de trazabilidad desde los requerimientos hacia otros requerimientos y casos de test.

	RF001	RF002	RF003	RF004	RF005	RF006	RF007	RF008	RF009	RF010	RF011	RN001	TC001	TC002	TC003
RF001				✓	✓								✓		
RF002															
RF003				✓								✓			
RF004															
RF005	✓	✓	✓					✓							
RF006															
RF007															
RF008															
RF009	✓														
RF010	✓	✓	✓	✓											
RF011					✓										
RN001															
TC001															

Fig. 8. Módulo de Trazabilidad de Requerimientos

Para cada proyecto se podrá visualizar la matriz de trazabilidad, en la cual se visualizan de forma clara las relaciones existentes.

La creación de las relaciones específicas de un requerimiento se puede realizar tanto en el alta del mismo como en una pantalla dedicada a tal motivo.

Documentos encontrados:

Código	Nombre	Coincidencia	
<input type="checkbox"/> RF003	Podrá modificar la información almacenada para un contacto determinado	98,64%	<input type="text" value="Nota..."/>
<input type="checkbox"/> RF004	Podrá consultar la información de un contacto concreto almacenado	94,77%	<input type="text" value="Nota..."/>

Busque los requerimientos asociados que quiera agregar por nombre o código.

Fig. 9. Resultado búsqueda relaciones de trazabilidad

Como se puede apreciar en la **Fig. 9**, se muestran los documentos que se han encontrado como relacionados con el porcentaje de coincidencia que resultó en dicha búsqueda. Para cada caso podrá agregarse una nota descriptiva, en la cual se detalle algún aspecto particular de la relación que deba tenerse en cuenta.

A su vez, debido a que no se puede garantizar que se recuperen todos los documentos que realmente están relacionados, se ha agregado una función de autocompletar. La misma le permite al usuario agregar manualmente las relaciones que se consideran necesarias y no han sido encontradas por esta aplicación.

Como resultado del establecimiento de las relaciones de trazabilidad, para cada requerimiento se brinda la funcionalidad de visualizar todas las relaciones que el mismo tiene directamente, las relaciones de dichas relaciones y así recursivamente. De esta forma se puede obtener un análisis del impacto que podría llegar a tener un cambio en un determinado requerimiento, mostrando dicha información en forma de árbol como se puede apreciar en la **Fig. 10**. En el mismo se ven nodos reducidos, los cuales se pueden ampliar seleccionándolos.

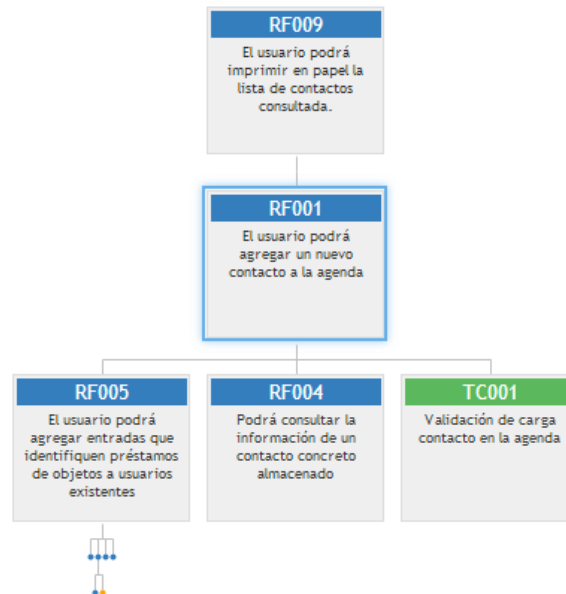


Fig. 10. Análisis de impacto de un requerimiento

Para mayor información acerca de las funcionalidades de la aplicación y su explicación detallada, puede accederse a la documentación online realizada para esta aplicación, disponible en <http://reqt.apphb.com/Documentation>

req.T

Bienvenido!
Descripción
Funcionalidades generales

REQUERIMIENTOS
Explicación general
Funciones en el listado
Requerimientos de usuario
Requerimientos funcionales y no funcionales
Crear requerimiento asociado
Crear nuevas versiones
Eliminar requerimientos de asociados
Ver detalle
Crear relaciones
Análisis de impacto
Descargar especificación de requerimientos

CASOS DE TEST
Explicación general
Funciones en el listado
Creación de caso de test
Crear nuevas versiones
Eliminar

Requerimientos funcionales y no funcionales

ReqT administra los requerimientos funcionales y no funcionales subordinados a un requerimiento de usuario a partir del cual tienen origen.

Crear requerimiento asociado
Para cada requerimiento de usuario está disponible la opción "Nuevo requerimiento asociado", la cual permite crear los requerimientos funcionales y no funcionales derivados del mismo.

Filtrar resultados

Código	Nombre	Acciones
R000	Mantenimiento y Gestión de nuestros contactos personales	[Iconos de acción]
R000	Gestión de préstamos a los contactos	[Iconos de acción]

Posteriormente se abrirá la pantalla correspondiente al nuevo requerimiento asociado. En ella estarán disponibles los siguientes campos de carga, los cuales son obligatorios:

- Código
- Nombre
- Tipo de requerimiento
- Contenido

Luego el usuario podrá guardar el requerimiento como está actualmente, o podrá buscar documentos relacionados al mismo para establecer las relaciones de trazabilidad.

req.T | Requerimientos | Casos de Test | Trazabilidad | Proyecto 1 | Competencia 1 | Diego Bencomo

Fig. 11. Manual de usuario online de la aplicación

También si se desea probar las funcionalidades provistas se ha creado un usuario a tal propósito. Los datos de acceso son los siguientes:

- Url: <http://reqt.apphb.com>
- Usuario: demojaiio
- Contraseña: demojaiio

Apéndice B: explicación de los métodos de recuperación de la información

Introducción

Para lograr la meta de que la aplicación a desarrollar brinde una asistencia efectiva más allá de lo que brinda una herramienta convencional es necesario justamente ir un paso adelante en lo que incluyen las mismas. Sin duda que son necesarios muchos aspectos relacionados al mantenimiento y organización de los requerimientos (los cuales también se incluyen en esta aplicación), pero también se considera útil realizar un agregado en el aspecto planteado en este trabajo final de carrera.

Con las técnicas de recuperación de la información es posible lograr este cometido. Su función es la búsqueda de información en documentos que están en formato electrónico, a partir de una determinada necesidad de información. En este caso la necesidad de la información será el requerimiento sobre el cual se desee buscar otros documentos relacionados.

El algoritmo que se ha decidido utilizar para llevar a cabo esta tarea es Latent Semantic Index (LSI). El motivo de esta elección entre otras alternativas se basa principalmente en que durante la investigación que se ha realizado para el armado del marco teórico, los correspondientes documentos destacan que dicho algoritmo presenta las siguientes ventajas con respecto a otras alternativas:

- Sinónimos: es capaz de considerar diferentes formas de tratar a una palabra en particular en la búsqueda de información.
- Polisemia: es capaz de superar la situación de cuando una palabra tiene más de un significado en particular, lo cual permite recuperar solo los documentos pertinentes.

Una de las críticas que recibe este método es que cuando se aplica a textos en lenguaje natural no hace uso del orden de las palabras, las relaciones sintácticas o morfología. Sin embargo, se pueden obtener muy buenos los resultados incluso sin la consideración de esta información.

Este algoritmo propone dos medidas que se utilizan para resumir el rendimiento de los resultados: precisión y recuperación.

- Precisión es la proporción de los documentos pertinentes en el conjunto devuelto al usuario.
- Recuperación es la proporción de todos los documentos relevantes de la colección que se recuperan por el sistema

Según estudios realizados, como los nombrados en [10], demuestran que el método LSI consigue los mejores resultados en relación con otros métodos vectoriales estándares, cuando las consultas y los documentos pertinentes no comparten muchas palabras.

Reseña de LSI

A continuación se expondrá una breve reseña del planteo matemático detrás del algoritmo LSI utilizado para la trazabilidad de requerimientos [1].

Modelo de espacio vectorial (VSM):

En el modelo de espacio vectorial (VSM), los documentos y las consultas se representan como vectores de los términos que aparecen en los documentos de una colección. Por lo tanto, un espacio en el documento VSM se describe por una matriz $m \times n$, donde m es el número de términos, y n es el número de documentos en la colección. A menudo, esta matriz se conoce como la matriz *término-por-documento*.

Una entrada genérica $a_{i,j}$ de esta matriz denota una medida del peso del término i en el documento j -ésimo. Estos se construyen sobre la base de la frecuencia de los términos en los documentos. En particular, estas medidas se aplican tanto a los coeficientes locales y globales para aumentar o disminuir la importancia de los términos dentro de los documentos. En concreto, se puede escribir:

$$a_{i,j} = L(i,j)G(i)$$

Donde $L(i,j)$ es el peso local del término i -ésimo en el documento j -ésimo y $G(i)$ es el peso global del i -ésimo término en todo el espacio de documentos. En general, el peso local aumenta con la frecuencia del término i en el documento j -ésimo, mientras que el peso global disminuye tanto como el término i -ésimo se extiende a través de los documentos del espacio.

Se considera que los mejores resultados en este apartado se han logrado mediante la ampliación de la frecuencia de los términos por un factor logarítmico para el peso local y el uso de la entropía del término dentro del espacio de documento para el peso global, como se ejemplifica en las siguientes fórmulas:

$$L(i,j) = \log(tf_{ij} + 1)$$

$$G(i) = 1 - \sum_{j=1}^n \frac{p_{ij} \log(p_{ij})}{\log(n)}$$

Donde tf_{ij} es la frecuencia del término i en el documento j -ésimo y p_{ij} se define como:

$$p_{ij} = \frac{tf_{ij}}{\sum_{k=1}^n tf_{ik}}$$

Desde un punto de vista geométrico, cada vector de documento (columnas de la matriz *término-por-documento*) representa un punto en el m -espacio de los términos. Por lo tanto, la similitud entre dos documentos en este espacio se mide por el coseno del ángulo entre los vectores correspondientes, lo que aumenta a medida que más términos se comparten. En general, dos documentos se consideran similares si sus correspondientes vectores apuntan en la misma dirección.