



UNIVERSIDAD DE CHILE  
FACULTAD DE CIENCIAS FÍSICAS Y MATEMÁTICAS  
DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN

REINGENIERÍA DE UNA HERRAMIENTA DE ADMINISTRACIÓN DE  
REQUISITOS: REQADMIN

MEMORIA PARA OPTAR AL TÍTULO DE INGENIERO CIVIL EN COMPUTACIÓN

FRANCISCO YENSUNG HAFON ACUÑA

PROFESOR GUÍA:

SERGIO OCHOA DELORENZI

MIEMBROS DE LA COMISIÓN:

SERGIO OCHOA DELORENZI

PABLO GONZÁLEZ JURE

AGUSTÍN VILLENA MOYA

SANTIAGO DE CHILE

2014

## RESUMEN

La administración de requisitos de un proyecto de software puede parecer un trabajo fácil, pero en realidad es una labor compleja, puesto que los requisitos están sujetos a cambios, ambigüedades, malas interpretaciones y pérdida de información. Además, la cantidad de requisitos y las posibles relaciones entre ellos pueden obstaculizar su visibilidad, especialmente al momento de determinar el impacto de un cambio en las especificaciones de requisitos.

Los alumnos del curso CC5401: Ingeniería de Software II, deben aprender a resolver estos y otros problemas rápidamente, sobre proyectos reales. Afortunadamente cuentan con una herramienta de administración de requisitos, la cual fue desarrollada específicamente para dar visibilidad y mantener la trazabilidad de los requisitos involucrados en los proyectos manejados por los estudiantes. La herramienta también ofrece instrumentos automatizados que permiten detectar tempranamente riesgos e inconsistencias sobre las especificaciones de los requisitos. Sin embargo, la interfaz y otras características de esa herramienta presentan problemas de usabilidad. Esto hace que los usuarios pierdan tiempo valioso al utilizarla, teniendo además que contrarrestar errores presentes en la implementación actual. Para resolver dicho problema, este trabajo de memoria desarrolló una nueva versión de la herramienta.

Para la nueva versión de la herramienta se reutilizaron sólo los conceptos de la herramienta original. La nueva herramienta tiene una interfaz totalmente renovada y fue implementada usando un framework Web que permite una mayor extensibilidad y mantenibilidad del software. Además, incluye el feedback de los usuarios en el proceso de desarrollo, e incorpora nuevos servicios útiles para los usuarios.

Las principales ventajas que ofrece esta nueva solución son las siguientes: una navegación intuitiva, generación rápida de documentos, fácil reconocimiento de potenciales conflictos en la especificación de requisitos, evita la pérdida de datos y los usuarios requieren menos tiempo para usar la herramienta.

Pruebas de usuario basadas en metodologías de usabilidad han mostrado, luego de varias iteraciones, que el software es estable y que los requisitos planteados fueron alcanzados de manera satisfactoria. Además, las encuestas finales revelan un mejor nivel de aceptación y facilidad de uso para los alumnos.

## TABLA DE CONTENIDO

1. Introducción .....	1
1.1. Justificación de la Propuesta.....	2
1.2. Objetivos de la Memoria .....	3
2. Trabajos Relacionados .....	4
3. Diagnóstico de la Solución Anterior.....	8
3.1. Diseño de una Nueva Interfaz .....	13
3.2. Evaluación de la Nueva Interfaz .....	14
3.2.1. Elección de usuarios para las pruebas .....	15
3.2.2. Pruebas de usuario .....	16
3.2.3. Resultados de evaluación de interfaz propuesta .....	17
3.3. Reimplementación .....	17
4. Concepción de la Solución .....	19
4.1. Principales Requisitos de la Solución .....	19
4.2. Arquitectura de la Solución .....	20
4.3. Modelo de Datos .....	22
4.4. Componentes de la Solución .....	24
5. Implementación de la Solución.....	27
5.1. Proceso de Especificación de Requisitos.....	28
5.2. Mapa de navegación .....	30
5.3. Administración .....	32
5.4. Proyecto.....	34
5.4.1. Hitos .....	37
5.4.2. Tipos de Usuario .....	39
5.4.3. Requisitos de Usuario .....	40
5.4.4. Requisitos de Software .....	43
5.4.5. Casos de Prueba .....	46
5.4.6. Módulos .....	48
5.5. Documentos .....	50

5.6. Herramientas .....	51
5.6.1. Estadísticas .....	52
5.6.2. Matrices de Trazado .....	53
5.6.3. Consistencia de Relaciones .....	55
5.6.4. Bitácora .....	57
5.6.5. Tareas .....	57
6. Evaluación Preliminar de la Solución .....	60
7. Conclusiones y Trabajo a Futuro.....	63
8. Bibliografía y Referencias .....	65

## ÍNDICE DE FIGURAS

Figura 3.1: Menú de navegación de ReqAdmin, Requisitos de Usuario. ....	9
Figura 3.2: Menú de navegación de ReqAdmin, Indicadores. ....	10
Figura 3.3: Interfaz original de ReqAdmin (problema con la distribución de información en la vista de requisitos de usuario). ....	13
Figura 4.1: Arquitectura de la solución. ....	21
Figura 4.2: Relaciones entre elementos de un proyecto. ....	22
Figura 4.3: Relaciones entre los documentos, las tareas y los elementos de un proyecto. ....	24
Figura 4.4: Componentes de la solución. ....	27
Figura 5.1: Proceso de especificación de requisitos. ....	29
Figura 5.2: Vista de ingreso. ....	31
Figura 5.3: Selección de proyecto a revisar. ....	31
Figura 5.4: Barra de navegación. ....	32
Figura 5.5: Asignación de permisos sobre un grupo de usuarios (Grupo: Analista). ....	32
Figura 5.6: Creación de un Proyecto. ....	33
Figura 5.7: Incorporar usuario dentro de un grupo. ....	33
Figura 5.8: Autogenerar nueva clave de acceso, y enviarla por correo electrónico. ....	34
Figura 5.9: Vista de requisitos de usuario en formato de elementos de contenido expandible. ....	35
Figura 5.10: Mensajes informan la acción realizada por el sistema. ....	35
Figura 5.11: Edición de elementos dentro de la lista, con sistema de pre validación de campos de formularios. ....	36
Figura 5.12: Alarma automática sobre un Requisito de Usuario. ....	37
Figura 5.13: Formulario de creación de Hitos. ....	38
Figura 5.14: Vista comprimida y expandida de Hitos, con atributos principales en barra de ordenamiento. ....	38
Figura 5.15: Formulario de creación de Tipos de Usuario. ....	39

Figura 5.16: Vista comprimida y expandida de Tipos de Usuario, con atributos principales en barra de ordenamiento. ....	40
Figura 5.17: Formulario de creación de Requisitos de Usuario. ....	42
Figura 5.18: Vista comprimida y expandida de Requisitos de Usuario, con atributos principales en barra de ordenamiento. ....	43
Figura 5.19: Formulario de creación de Requisitos de Software. ....	45
Figura 5.20: Vista comprimida y expandida de Requisitos de Software, con atributos principales en barra de ordenamiento. ....	46
Figura 5.21: Formulario de creación de Casos de Prueba. ....	47
Figura 5.22: Vista comprimida y expandida de Casos de Prueba, con atributos principales en barra de ordenamiento. ....	48
Figura 5.23: Formulario de creación de Módulos. ....	49
Figura 5.24: Vista comprimida y expandida de Módulos, con atributos principales en barra de ordenamiento. ....	50
Figura 5.25: Sección de introducción del Documento de Requisitos. ....	50
Figura 5.26: Selección de secciones del Documento de Requisitos. ....	51
Figura 5.27: Selección de versiones anteriores de la sección del documento. ....	51
Figura 5.28: Estadísticas de Requisitos de Usuario para la Primera Iteración. ....	52
Figura 5.29: Estadísticas de Casos de Prueba para la Primera Iteración. ....	52
Figura 5.30: Estadísticas de Módulos. ....	53
Figura 5.31: Matriz de Trazado muestra cinco Requisitos de Software sin Casos de Prueba asociados. ....	54
Figura 5.32: Matriz de Trazado de un documento PDF (Requisitos de Software vs. Casos de Prueba). ....	55
Figura 5.33: Vista de Consistencia de Relaciones para Requisitos de Usuario y Requisitos de Software. ....	56
Figura 5.34: Bitácora muestra los cambios realizados en los requisitos. ....	57
Figura 5.35: Vista del Jefe de Proyecto sobre las Tareas asignadas a un usuario. ....	58
Figura 5.36: Ciclo de vida de las Tareas. ....	59

# 1. INTRODUCCIÓN

ReqAdmin es una herramienta Web destinada a la administración de requisitos de un proyecto de software. La herramienta es capaz de llevar a cabo una generación automática de métricas e indicadores que dan visibilidad al estado y avance de un proyecto. ReqAdmin fue desarrollado en el año 2007 como Tesis de Magíster en Ciencias de la Computación [1], y ha sido usado principalmente como herramienta de apoyo a la administración de proyectos del curso de Ingeniería de Software II (CC5401).

Dada la alta complejidad que representa el correcto seguimiento y administración de requisitos y funcionalidades asociadas a un proyecto de software, se hace evidente la necesidad de contar una herramienta como ReqAdmin, que se encargue de asistir a las personas involucradas en el desarrollo de éste.

Cabe señalar que ReqAdmin es un software implementado hace más de 6 años y que por tanto, su aspecto general (tanto visual como estructural) no se encuentra del todo alineado con los modelos actualmente más recomendados. Especialmente en lo que se refiere a las interfaces de usuario y su facilidad de uso.

Además, este software ha sido extendido en diferentes oportunidades por parte de distintos equipos de desarrollo, quienes no se preocuparon de dejar la nueva funcionalidad bien integrada al producto de software, por lo tanto el software tiene problemas de usabilidad. Esto representa una de las principales limitaciones para los usuarios nuevos (usualmente estudiantes). Por ejemplo, muchos usuarios no saben cómo acceder a una sección del software, o tienen dificultades y necesitan ayuda para usar ciertas opciones que ofrece la herramienta. Algunas de las funcionalidades disponibles en ReqAdmin para la visualización de, por ejemplo, las relaciones entre los distintos tipos de requisitos asociados al proyecto, son las matrices de trazado y el árbol de relaciones, los cuales permiten verificar la consistencia del matching entre requisitos de usuario y de software.

Sin embargo, a pesar del propósito inicial de la herramienta y su enfoque orientado a la generación automática de métricas e indicadores, los usuarios pierden tiempo valioso averiguando la manera correcta de utilizarla, teniendo que luchar además contra problemas de usabilidad y funcionales causados en parte por las sucesivas modificaciones a la versión original de ReqAdmin. Para corregir los problemas planteados anteriormente, este

trabajo de memoria pretende realizar una reingeniería a ReqAdmin, enfocándose principalmente en mejorar la usabilidad y utilidad de la herramienta. Se pretende también establecer un orden adecuado a los distintos componentes y funcionalidades del sistema, y así disponer de una herramienta de administración de requisitos más intuitiva y provechosa.

### 1.1. JUSTIFICACIÓN DE LA PROPUESTA

Hasta hoy aún existen reclamos de alumnos sobre el mal funcionamiento de algunos módulos de ReqAdmin. Siendo, por ejemplo, víctimas de pérdida de información importante al momento de crear informes o introducir datos en la interfaz actual. Frecuentemente recurren a soluciones “parche”, trabajando de manera paralela con otro software de edición. Se puede decir que la herramienta actual no ofrece la confiabilidad necesaria para su propósito. Sin embargo ha sido de gran ayuda para los alumnos al momento de realizar informes correspondientes a las distintas etapas de un proyecto de software. Ofrece un nivel adecuado de automatización que permite al alumno concentrarse en el contenido, por ejemplo, automatizando y estandarizando el formato final del documento impreso.

En un entorno de trabajo real, los alumnos pueden verse enfrentados a una gran cantidad de problemas que surgen en la realización de un proyecto de software, causados en parte por una mala o insuficiente administración de requisitos.

Con respecto a ReqAdmin, la problemática explicada anteriormente se podría atribuir a una falta de documentación introductoria para el usuario, pero en realidad una buena interfaz y un correcto orden en los componentes disponibles deberían ser suficientes para hacer de ReqAdmin una herramienta más intuitiva y útil para el usuario. Siempre contemplando que los cambios en la interfaz y refinamiento de funcionalidades deben ser llevados a cabo de manera sistemática, haciendo uso de, por ejemplo, técnicas preestablecidas por la ingeniería en usabilidad.

Es necesario destacar que a pesar de los problemas mencionados, la utilidad potencial de este software es sumamente valorable, principalmente desde el punto de vista de los alumnos que finalmente hacen uso del sistema. Los alumnos generalmente disponen de poco tiempo para formalizar y especificar los requisitos, y la herramienta entrega una valiosa ayuda a la hora de administrar tanto los requisitos como el avance de los proyectos.



## 1.2. OBJETIVOS DE LA MEMORIA

Este trabajo de memoria busca principalmente realizar un proceso de reingeniería a ReqAdmin, para obtener una nueva versión de la herramienta más robusta, extensible y con una interfaz ordenada, intuitiva y usable. La idea es que de esta manera pueda tornarse en una herramienta realmente útil para la administración de requisitos de usuario y de software, sobre todo para los distintos proyectos de software desarrollados en el curso de Ingeniería de Software II (CC5401), y también aumentar su potencial como una herramienta de apoyo para otros proyectos guiados por los requisitos.

Para alcanzar el objetivo anterior existen distintos objetivos específicos relacionados que deben ser cumplidos:

- Determinar las **funcionalidades definitivas** (principales y secundarias) que se deberán conservar, reparando las que se encuentren defectuosas, y considerando eventualmente adicionar funcionalidades que se consideren necesarias de acuerdo a las necesidades de los usuarios finales.
- Diseñar las nuevas **interfaces de usuario** de acuerdo a las necesidades y funcionalidades principales, las cuales deben ser validadas y depuradas según metodologías de ingeniería en usabilidad.
- Crear las distintas pruebas y entrevistas que se realizarán durante la recopilación de **feedback de los usuarios** sobre los distintos prototipos a generar durante el desarrollo del trabajo. Estas interacciones con los usuarios deben poder realizarse de manera rápida, de bajo costo, eficiente y estructurada para recopilar una cantidad suficiente y confiable de datos útiles.
- **Implementar** la solución propuesta, incluyendo su nueva apariencia, y la reestructuración de los diversos componentes de la herramienta. Este trabajo deberá estar debidamente validado y probado por usuarios reales, aumentando las probabilidades de que el producto desarrollado sea exitoso.

## 2. TRABAJOS RELACIONADOS

Actualmente se encuentran disponibles muchas alternativas de software para la administración de requisitos de un proyecto. Estas herramientas comparten características en común, y a continuación se listan algunas de ellas para comprender mejor el proceso de administración de requisitos:

- Permite definir atributos y estructura de requisitos.
- Permite asociarlos con otros elementos del proyecto, como casos de prueba y otros requisitos según algún criterio de dependencia.
- Permite trazar la relación de requisitos con otros elementos del proyecto, por ejemplo para realizar un análisis de impacto ante modificaciones.
- Permite tener control sobre los cambios en los requisitos (control de versiones).
- Permite priorizar los requisitos.
- Permite administrar versiones de documentos o reportes de requisitos (estadísticas y otros indicadores de avance del proyecto).
- Permite realizar el seguimiento del estado del requisito dentro del proyecto.
- Permite obtener vistas de los requisitos filtradas por atributos.

La gran variedad de herramientas presentes en el mercado para la administración de requisitos ofrecen además muchas otras características que son incorporadas para abarcar otros procesos relacionados con el desarrollo de una aplicación o producto de software, desde la toma y creación de requisitos hasta la administración de su ciclo de vida, incluyendo producción y mantenimiento. Además tienen un enfoque más visual y colaborativo sobre las primeras etapas de la ingeniería de requisitos. Durante estas etapas los requisitos no están bien definidos aún y son modificados constantemente de acuerdo a las necesidades del producto final y del proyecto. Los integrantes encargados del desarrollo de requisitos deben tomar en cuenta todos los cambios rápidamente para tomar decisiones a tiempo y una representación visual puede ser de gran ayuda para el proceso completo. Imágenes, sketches, storyboards, mockups y otros tipos de visualizaciones son incorporadas cada vez más en herramientas de modelamiento de requisitos. Incluso como manera de complementar las visualizaciones incorporan características colaborativas al proceso de

modelamiento y desarrollo de requisitos que faciliten la interacción y comunicación entre las personas a cargo de discutir y definir la forma final de los requisitos y otros elementos del proyecto, logrado generalmente a través de herramientas web con espacios de discusión y vistas generales fácilmente accesibles.

Algunas de estas herramientas son CaliberRM, Analyst Pro, Visure Requirements, Jira, HP Agile Manager, Innovator for Business Analysts y OverTI Requirements Quality Analyzer.

CaliberRM facilita la comunicación entre los integrantes principalmente a través de un sistema de modelamiento visual de requisitos colaborativo. Hace uso de diagramas UML y prototipos visuales, incorporando además espacios de discusión sobre el diseño actual y levantando notificaciones ante cambios en los requisitos para evitar pasar por alto el impacto de esa modificación sobre el resto de los elementos del proyecto. Soporta metodologías ágiles y tradicionales (waterfall, iterativa).

De manera muy similar a CaliberRM, las herramientas de administración y desarrollo de requisitos Analyst Pro y Visure Requirements permiten personalizar los atributos de los elementos del proyecto (requisitos, casos de prueba, scripts de prueba) y relacionarlos de manera jerárquica, permitiendo visualizar la trazabilidad de éstos de manera gráfica (por ejemplo con matrices de trazado). Incluyen un editor de diagramas UML y son compatibles con MS Word y Excel para la generación de reportes gráficos y estadísticos a través de plantillas personalizables. Incluyen módulos para el control de versiones de los reportes generados y almacenan la evolución de los distintos elementos. Permiten establecer permisos de edición y revisión sobre los integrantes ofreciendo interfaces diferenciadas por rol. Además en los reportes escritos se pueden embeber links hacia requisitos y otros elementos del proyecto.

Jira es una herramienta de administración de proyectos que incluye soporte para el desarrollo de requisitos. Está enfocada al uso de metodologías ágiles poniendo mucho énfasis en la priorización de los requisitos y la comunicación dentro del equipo, ofreciendo módulos de administración de tareas asignadas a los integrantes. De esta manera mantiene a cada integrante del equipo al tanto de sus tareas por hacer, visualizando su prioridad, plazos límites y el avance con respecto al resto del grupo. Incorpora un módulo de priorización y administración de problemas y cambios solicitados por resolver en el producto como manera de apoyar las distintas etapas de desarrollo. HP Agile Manager es otra herramienta que usa

enfoque en metodologías ágiles, facilitando características para la planificación del equipo y de pequeños entregables del producto, como el uso de scrum y kanban para mantener el foco en los principales requisitos de la solución.

También existen algunas alternativas que destacan por sus características innovadoras, como Innovator for Business Analysts, que no sólo incluye edición de diagramas sino que incorpora animaciones para ver en movimiento los procesos descritos por los requisitos. Otra herramienta particular es OverTI Requirements Quality Analyzer, la cual apoya la validación de requisitos y entrega métricas de calidad de éstos basado en el análisis del contenido textual. Es capaz de analizar el texto descriptivo de los requisitos e identificar ambigüedades, abusos de conectores, oraciones subjetivas, nivel de legibilidad y otras características propias de la redacción.

El diseño de la nueva herramienta toma en consideración varias de las funcionalidades y características descritas en los casos anteriores, pero está basado principalmente en ReqAdmin, que es la herramienta actualmente utilizada en el curso de Ingeniería de Software II (CC5401) y ha sido actualizada con múltiples funcionalidades orientadas exclusivamente a los requerimientos particulares del tipo de proyecto que los alumnos abordan durante el semestre. ReqAdmin permite al cuerpo docente administrar múltiples proyectos, estableciendo las fechas o hitos principales y asociar los integrantes (alumnos), con permisos de acceso y edición determinados por su rol asociado (jefe de proyecto, analista, implementador, diseñador y téster), que serán responsables del desarrollo de éste.

Dentro de las características principales se encuentra la creación y edición de elementos del proyecto (requisitos de usuario, requisitos de software, módulos y casos de prueba), considerando las posibles relaciones de dependencia que se pueden dar entre éstos. Se pueden obtener estadísticas de avance y realizar un seguimiento al estado de cumplimiento de los requisitos usando árboles de relaciones o descargando las matrices de trazado. Además permite la edición de documentos escritos y reportes descargables. Los documentos son generados con un formato específico usado en el curso de Ingeniería de Software II y son usados como manera oficial de presentar el avance del proyecto. Una bitácora de cambios se encarga de registrar las modificaciones realizadas a las descripciones de los requisitos, módulos y casos de prueba. También incorpora un sistema de alarmas de consistencia que advierte automáticamente al usuario de posibles conflictos en la especificación de requisitos. Con la intención de organizar y controlar mejor el trabajo de los integrantes durante el desarrollo del

proyecto, incorpora un módulo que permite al jefe de proyecto asignar tareas a los integrantes y revisar el nivel de cumplimiento.

Si bien existe una gran variedad de productos que permiten la administración de requisitos que a la vez son candidatos para reemplazar a la actual herramienta (ReqAdmin), no es evidente que un reemplazo por un producto existente sea un real aporte al curso de Ingeniería de Software II, considerando que es muy probable que la herramienta reemplazante sea demasiado compleja como para que los alumnos puedan aprovecharla sin gastar demasiado tiempo en aprender a manejarla, o no disponga de todas las características que el proyecto y el curso requieren. La estrategia actual es usar una herramienta que sea simple y esté específicamente diseñada de acuerdo a las necesidades de los alumnos y del curso de Ingeniería de Software II.

### 3. DIAGNÓSTICO DE LA SOLUCIÓN ANTERIOR

Para poder realizar un diagnóstico correcto de la herramienta actual se comienza examinando las funcionalidades implementadas, revisando principalmente la interfaz del usuario y el flujo de navegación.

Con la intención de identificar y formalizar los problemas actuales de ReqAdmin se realizan encuestas online y presenciales a usuarios que hayan usado la herramienta durante el curso de Ingeniería de Software II. Estas encuestas son realizadas con los siguientes objetivos: averiguar cuáles son los principales problemas de la herramienta; obtener sugerencias para corregir las funcionalidades problemáticas; considerar maneras diferentes de realizar alguna tarea actual; identificar nuevas funcionalidades que serían deseables o necesarias; determinar posibles funcionalidades innecesarias o redundantes actualmente implementadas para reducir la complejidad de la herramienta; averiguar el nivel de satisfacción actual.

El feedback obtenido revela que los principales problemas de la herramienta percibidos por los usuarios están relacionados con comportamientos inesperados, como la pérdida de información ingresada a formularios o la des-configuración de permisos de edición. También representa un problema la navegación poco intuitiva y la visualización incómoda de los documentos y reportes generados. Éste último punto se refiere específicamente a que existen problemas de codificación sobre algunos caracteres y problemas de ajuste de dimensiones en algunas imágenes que son embebidas en el documento, imágenes que con frecuencia no aparecen en el documento si se descarga y visualiza con el editor MS Word.

Una de las sugerencias recibidas para mejorar la experiencia, considera reemplazar la descarga de documentos en formato RTF (de texto enriquecido) visibles desde editores locales, como MS Word, por la opción de descarga en formato PDF, visible directamente desde el navegador y con mínimos cambios a través de plataformas. Un defecto incómodo al que hacen referencia los usuarios, es la dificultad para abrir dos pestañas con ReqAdmin en el navegador (por ejemplo para comparar listas de requisitos), provocado por el uso incorrecto de Javascript en el flujo de navegación. Además se sugiere añadir una visualización online de matrices de trazado como una funcionalidad deseada y potencialmente útil para trazar las relaciones en requisitos sin tener que descargar las matrices en un archivo en formato XLS y visualizarlas con un editor local, como MS Excel. Esta visualización además

puede incorporar elementos interactivos como apoyo, por ejemplo, para detectar conflictos en la matriz de trazado.

Una característica que resulta ser muy útil y no se encuentra presente en ReqAdmin es la capacidad de generar un documento que permita comparar visualmente la consistencia del contenido de elementos que estén relacionados, permitiendo verificar, por ejemplo, si un requisito de usuario se encuentra razonablemente descrito por el contenido de sus requisitos de software asociados.

Para complementar la información obtenida en las encuestas y en las entrevistas, y considerar la calidad de ReqAdmin en términos de usabilidad, se presenta a continuación una revisión sobre su funcionamiento en base a reglas y principios conocidos de usabilidad [5].

El “principio 7+-2” hace referencia a que la memoria de corto plazo es capaz de retener de 5 a 9 cosas a la vez, argumento usado generalmente para limitar el número de opciones en los menús de navegación a 7. Según este principio ReqAdmin tiene problemas en su barra de navegación, al no tener una clara separación entre categorías y sub-categorías se requiere un escaneo visual cada vez que se busca alguna funcionalidad, y presenta más de 13 opciones que pueden ser difíciles de memorizar (ver Figura 3.1).

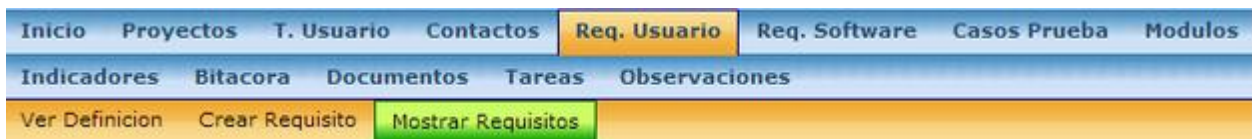


Figura 3.1: Menú de navegación de ReqAdmin, Requisitos de Usuario.

La “regla de los 3 clics” está muy relacionada con la problemática anterior y hace referencia a una navegación clara y una estructura y jerarquía lógica del sitio. En realidad lo importante no es la cantidad de clics sino que los usuarios siempre sepan dónde se encuentran y dónde pueden ir. En este caso el problema de las opciones de navegación que ofrece ReqAdmin es que a los usuarios (según las encuestas) no les hace sentido la ubicación de algunas funcionalidades dentro de sus categorías. Un ejemplo de esta situación se da cuando un usuario desea descargar un documento e intenta accediendo a la sección “Documentos”, cuando en realidad la descarga se realiza en la sección “Indicadores” en la opción “descarga de reportes” junto a otras opciones no del todo relacionadas, como la “vista rápida de requisitos” (ver Figura 3.2). Por lo tanto, una nueva interfaz debería considerar un menú de navegación más consistente en cuanto a una clasificación clara y un número reducido de opciones simultáneas.



Figura 3.2: Menú de navegación de ReqAdmin, Indicadores.

Como resultado de estudios en diseño de interfaces, Ben Shneiderman [6] propone una colección de principios que son derivados heurísticamente de su experiencia, aplicables en sistemas interactivos, principios conocidos como las ocho reglas doradas del diseño de interfaces.

La primera regla puede expresarse como “esforzarse por ser consistente”, y enfatiza que para situaciones similares deberían requerirse acciones similares, utilizando términos y nombres únicos para referirse a las opciones, pop-ups y alertas. La interfaz actual utiliza ocasionalmente nombres distintos para referirse a un mismo elemento. Por ejemplo, algunos términos que se confunden constantemente son: documentos/reportes, iteración/incremento/hito y tipos de usuario/roles. Los títulos de algunas opciones de la barra de navegación muestran una aparente falta de consistencia con su contenido, como la opción de “indicadores” anteriormente mencionada que muestra indicadores y otros tipos de herramientas como la “descarga de documentos” y la “vista rápida de requisitos”. La opción de “documentos” muestra las distintas secciones editables de los documentos a generar, sin embargo no señala cuáles de estas secciones se incluyen en los distintos tipos de documentos descargables (Documento de Requisitos, Documento de Diseño y Documento Histórico). Entonces, considerando esta primera regla el nuevo diseño debe referirse a los distintos elementos de la interfaz mediante un único nombre (como “documentos”, “hitos” y “tipos de usuario”) consistente con su contenido, por ejemplo, procurar que todo el módulo relacionado con los documentos se encuentre bajo una misma categoría.

La segunda regla hace referencia a habilitar atajos para facilitar el trabajo de usuarios expertos.

La tercera regla se puede denominar como “ofrecer retroalimentación informativa” e indica que por cada acción del usuario el sistema debería emitir una respuesta, la cual debe ser modesta en caso de acciones frecuentes, y más notoria en caso de acciones menos frecuentes. En el caso de ReqAdmin una acción frecuente es la edición de requisitos, la cual hace que simplemente se recargue la página, pero la edición del estado de los casos de prueba asociados a esos requisitos desencadena una actualización del estado de los mismos requisitos, actualización que pasa inadvertida luego



de simplemente recargar la página de los casos de prueba. Este simple problema es causa frecuente de confusiones y produce desconfianza sobre la estabilidad de los datos ingresados en los formularios, problema originado por ausencia de una respuesta más consistente del sistema.

La cuarta regla está muy relacionada con este problema y enfatiza el uso de diálogos para mostrar al usuario un punto de satisfacción al momento de completar una etapa del proceso. En el caso de borrar algún requisito, ReqAdmin pide correctamente una confirmación y una justificación de la acción con un diálogo. Sin embargo al intentar borrar un caso de prueba el mensaje de confirmación es ambiguo y advierte que "al borrar este caso de prueba además se eliminarán las relaciones existentes", mensaje poco claro que se presta a confusiones. En base a estas dos reglas el nuevo diseño debe advertir al usuario la acción realizada por el sistema tras cada edición o creación de un elemento del proyecto (requisito, caso de prueba, módulo, hito y tipo de usuario), y debe mostrar alertas o diálogos de confirmación ante acciones de mayor impacto, como borrar un elemento del proyecto, procurando que el mensaje sea claro e informe de todos los riesgos involucrados en la decisión.

La quinta regla consiste en "ofrecer alternativas simples ante errores", de tal manera que el diseño de la interfaz no debería permitir al usuario cometer errores muy graves, y en caso de un error el sistema debería ofrecer una alternativa simple para manejarlo. Ingresar datos directamente a ReqAdmin es riesgoso, según los propios usuarios, y cuando existe un error en algún formulario se advierte al usuario con una alerta pero se recarga la página perdiendo la información que se había ingresado en el formulario. Esto es un problema para el ingreso de información relativa a elementos del proyecto (requisitos, casos de prueba, etc.) en cuanto a errores por información requerida no ingresada o conflictos con otros elementos asociados, pero principalmente representa un problema para la sección de edición de párrafos o secciones de un documento, pues en el estado actual de la herramienta y deja de ser factible editar el documento sobre el mismo formulario ante errores relacionados con caracteres no soportados, forzando al usuario a ingresar nuevamente toda la información junto con las correcciones respectivas. Entonces, como medida para atender esta necesidad en el nuevo diseño se incorpora al flujo de ingreso de datos en formularios de elementos del proyecto una pre-consulta de validación y verificación de los datos ingresados con el objetivo de señalar los errores sin perder la información ingresada a causa de una recarga de la página completa.

La sexta regla toma en cuenta que el usuario desconoce la interfaz y generalmente asume que las acciones son reversibles, y a modo de aprendizaje prueba opciones y funcionalidades desconocidas sin anticipar errores. Por lo tanto, el sistema debería ofrecer un método simple para revertir las últimas acciones. La bitácora de cambios implementada en ReqAdmin sólo despliega la descripción del elemento modificado y no el resto de su información, imposibilitando una recuperación completa ante una modificación o eliminación de un elemento del proyecto. Tampoco existe una bitácora o un controlador de versiones que maneje las modificaciones realizadas sobre algún párrafo o sección de un documento, y por tanto no es factible (salvo usando respaldos manuales) recuperar versiones pasadas de un documento de ReqAdmin. Por lo tanto, una mejora en el nuevo diseño contempla un sistema de registro de cambios (con autor y fecha), en formato de bitácora para almacenar la evolución de los elementos del proyecto, y en formato de historial de cambios para cada sección o párrafo de los documentos.

Tras cada creación, modificación o eliminación de algún elemento, se debe registrar su estado completo (incluyendo su descripción, atributos y posibles relaciones con otros elementos) en la bitácora permitiendo la recuperación de información ante algún error. Particularmente a través del historial de cambios es posible recuperar con un sólo clic el estado de alguna sección de un documento.

La séptima regla indica que los usuarios esperan poder tener el control sobre la herramienta y no al revés, entonces el diseño debe permitirles ser los que inician la acción y no los que reciben acciones del sistema.

Por último la octava regla establece como objetivo reducir el esfuerzo sobre la memoria de corto plazo, por lo que las visualizaciones deben ser simples y claras, procurando reducir el movimiento del usuario sobre la página. Este punto está relacionado fuertemente con el problema de distribución de información en la visualización de requisitos y otros elementos del proyecto (casos de prueba, módulos, etc.) y su formato tabular que dificulta la lectura de múltiples elementos y crea espacios con márgenes muy grandes sobre algunos atributos (ver Figura 3.3).

La alternativa a esta visualización muestra los elementos del proyecto en listas verticales, en donde los principales atributos son presentados de manera resumida en una fila y el resto de los atributos junto con la descripción principal se encuentran disponibles una vez que se expande el contenido del elemento. De esta manera se ahorra espacio, se simplifica la

búsqueda visual y se facilita la comparación entre dos o más elementos cercanos.

ReqAdmin - Administración de Requisitos [Sistema de Pre-Inscripcion Academica para el DCC] | [fhafon] | [Logout]

Inicio Proyectos T. Usuario Contactos **Req. Usuario** Req. Software Casos Prueba Modulos

Indicadores Bitacora Documentos Tareas Observaciones

Ver Definicion Crear Requisito **Mostrar Requisitos**

Req. Usuario/ Mostrar Requisitos/

**Req. Usuario**

Tipo : Todos [Mostrar] Se han encontrado 28 registros de este tipo Requisitos: (Cumple) (No Cumple) (Ambiguo)

ID	Nombre	Descripcion	Fuente	Prioridad	Estabilidad	Estado	Tipo	Inc.	Actualizacion	Palabras Ambiguas	Opciones
RU0001	Usuarios Sistema	El sistema debe poder identificar a los usuarios Administrativos y Alumnos segun el permiso de acceso que posean, que son editables y unicos, es decir, solo se puede ser un tipo de usuario. Tambien debe existir un usuario Super-Admin, unico en el sistema, y que administra los permisos de los otros usuarios mediante la manipulacion de una tabla con RUTs que tengan permisos distintos de alumno.	Equipo	Critica	Intransable	Cumple	Funcional	Primera Iteracion	2013-06-04 15:09:00	0	
	T.U. Asoc.	Administrativo -Alumno -									[Editar]
RU0002	Usuario Profesor	El sistema debe poder identificar a los usuarios Profesores, que es tambien un tipo de usuario disjunto a los Administrativos y Alumnos, y	Cliente, Equipo	Deseable	Transable	Cumple	Funcional	Segunda Iteracion	2013-07-22 08:50:00	0	?

Figura 3.3: Interfaz original de ReqAdmin (problema con la distribución de información en la vista de requisitos de usuario).

### 3.1. DISEÑO DE UNA NUEVA INTERFAZ

Cada una de las reglas y principios de usabilidad que se mencionan son considerados en el proceso de diseño de la nueva propuesta, incorporando a la vez el feedback de los usuarios finales y las mejoras reveladas durante el proceso de diagnóstico. Las principales características que diferencian a esta nueva propuesta de ReqAdmin son:

- Menú de navegación simplificado y cuidadosamente diseñado para ser intuitivo.

- Los elementos del proyecto (Hitos, Tipos de Usuario, Requisitos de Usuario, Requisitos de Software, Módulos y Casos de Prueba) originalmente eran visualizados a través de tablas extensas con todo el contenido, ahora son listados como elementos de contenido expandible, los cuales pueden ser ordenados de acuerdo a sus atributos principales. De esta manera cada elemento muestra sus atributos principales en la vista compacta y el resto de sus atributos están disponibles en la vista extendida.
- Permite ver las matrices de trazado directamente desde el navegador sin tener que descargar un archivo en formato XLS, destacando automáticamente posibles conflictos en alguna de sus filas o columnas.
- Se puede generar documentos en formato PDF, visibles y descargables directamente desde la herramienta.
- Evita la pérdida de datos ingresados en formularios con conflictos en alguno de sus campos (pre validación), además integra una bitácora de cambios sobre los elementos del proyecto, incluyendo también un historial de cambios sobre los documentos permitiendo la recuperación de versiones pasadas.
- Aspecto visual claro y simple, reduciendo la cantidad de bordes y detalles, usando colores claros y consistente con su significado, como la representación de los estados de cumplimiento (verde cumple, rojo no cumple, morado ambiguo) o los botones azules para las posibles acciones del usuario.
- Se incorpora una nueva opción que permite comparar visualmente la consistencia del contenido de elementos que se encuentran relacionados.

### 3.2. EVALUACIÓN DE LA NUEVA INTERFAZ

Una de las buenas prácticas de usabilidad que se aplica en este proyecto es la evaluación del diseño de la interfaz previa a la implementación. Una estrategia de desarrollo precipitado suele dejar las pruebas de usuarios para el final, esperando primero a tener un producto funcional mínimo. Postergando decisiones de diseño y de usabilidad para las últimas iteraciones, y por lo tanto aumentando el riesgo de que la solución no cumpla con las expectativas de usuarios finales.

La interfaz propuesta es puesta a prueba de manera preliminar a la implementación, usando una estrategia basada en Paper Prototyping [3]. Esta estrategia permite poner a prueba la interfaz con usuarios reales,

usando representaciones fáciles de generar y de bajo costo de las distintas vistas en la herramienta. La idea original de esta estrategia es crear físicamente las distintas vistas y componentes de la interfaz, usando prototipos de papel, simulando un ambiente web interactivo frente al usuario y examinando su comportamiento ante pruebas específicas diseñadas para resolver las principales dudas que se tienen sobre la nueva propuesta. Sin embargo, simular las interacciones del usuario puede llegar a ser complejo, y por lo general requiere a un equipo detrás de la simulación que se encargue de realizar tareas simultáneas. Durante la simulación se deben mostrar las vistas y componentes de papel en el orden y tiempo adecuado a medida que el usuario realiza las pruebas asignadas y se examina su comportamiento de acuerdo a pautas de evaluación.

En esta oportunidad, se utilizan prototipos hechos en HTML con estilos CSS (Cascading Style Sheets) e hipervínculos, para obtener una navegación simple, de tal manera de agilizar y simplificar la simulación, realizándose directamente en un navegador web. El resto del procedimiento consiste en la elección y diseño de las pruebas de usuario y pautas de evaluación usadas para rescatar información útil de las pruebas.

Cabe destacar que no sólo se evalúa la propuesta previa a la implementación, sino que durante el desarrollo de la nueva herramienta también se realizan pruebas de usuario encargadas de obtener feedback sobre versiones incrementales del producto. Estas pruebas permiten comprobar la utilidad de la solución frente a usuarios reales, modificando el diseño de acuerdo a sus requerimientos e incluso incorporando nuevas características.

### *3.2.1. ELECCIÓN DE USUARIOS PARA LAS PRUEBAS*

En este caso, se puede identificar dos tipos de usuarios sobre el producto final: administradores e integrantes del equipo de desarrollo. Los administradores corresponden al equipo docente del curso de Ingeniería de Software II, encargados de administrar los proyectos del semestre, establecer las fechas o hitos asociados a los proyectos, asignar permisos de edición a integrantes del equipo de desarrollo y monitorear el estado de los proyectos. Los usuarios que integran el equipo de desarrollo (alumnos) son, por lo general, clasificados según su rol específico relativo al proyecto designado: Jefe de Proyecto, Analista, Diseñador, Implementador y Téster. En esta oportunidad las pruebas de usabilidad son enfocadas sólo sobre los usuarios del equipo de desarrollo, puesto que son los usuarios más afectados

por problemas de usabilidad y deben ser capaces de aprender a usar la herramienta rápidamente.

### *3.2.2. PRUEBAS DE USUARIO*

El esquema sugerido por Paper Prototyping para evaluación de vistas y componentes de la interfaz establece que las pruebas de usuarios deben ser creadas a partir de objetivos y preguntas orientadas a resolver las principales dudas que se tienen sobre el diseño. Sin embargo las pruebas de usuario también deben corresponder a objetivos válidos y con importancia real para el usuario. La duración de cada prueba no debería ser mayor a los 10 minutos, y debe ser útil para resolver una o más dudas importantes. Estas pruebas deben preferentemente hacer que el usuario navegue por más de una vista para disminuir la cantidad de supuestos y aumentar la probabilidad de detectar problemas conceptuales y de navegación. El usuario debe ser el que decida si ha completado la prueba, por lo tanto ésta debe tener un fin claro y fácil de reconocer. Cuando no existe un fin bien definido y fácil de identificar podría ocurrir que el usuario no sepa cuándo terminar y no alcance el objetivo deseado o realice más acciones de las planificadas, y deba ser interrumpido para pasar a la siguiente prueba. Cualquier duda que le quede al evaluador mientras el usuario realiza las pruebas sólo debe ser resuelta una vez que se han completado todas las pruebas, al igual que las posibles sugerencias o apreciaciones que el usuario desee aportar en base a su experiencia.

La creación de pruebas de usuario es uno de los puntos más importantes, éstas son diseñadas a partir de la intersección entre posibles objetivos del usuario (problemas que desee resolver usando la herramienta dada) y las respectivas dudas que el equipo desarrollador tiene sobre la interfaz y el funcionamiento general de la herramienta. La creación de este tipo de pruebas comienza con la creación de una lista con objetivos que el usuario pueda alcanzar con la interfaz. Luego se listan las dudas, riesgos y otros aspectos que el equipo desarrollador considere relevante evaluar. Se priorizan las dudas escogiendo, para las de mayor importancia, objetivos de usuario con los cuales sea posible resolverlas.

A partir de los objetivos de usuario escogidos se formalizan las pruebas en una plantilla especificando: el objetivo a cumplir, los supuestos previos a la prueba, una lista de los pasos que realizaría un experto para completar el objetivo y las instrucciones que serán dictadas al usuario al momento de la prueba. Existen consideraciones que deben ser tomadas en cuenta al momento de redactar las instrucciones, como el hecho de que éstas no

deben usar los mismos términos que se muestran en las opciones de navegación, ya que se estaría facilitando en demasía la labor del usuario, el cual sólo tendría que explorar visualmente las palabras de las opciones para lograr el objetivo.

### *3.2.3. RESULTADOS DE EVALUACIÓN DE INTERFAZ PROPUESTA*

Los resultados que se obtienen con las pruebas de usuario sirven para resolver las principales dudas de diseño y refinar algunos aspectos no funcionales de la interfaz. La mayoría de los aspectos que son refinados, corresponden al posicionamiento y visibilidad de botones y espacios de información. Por ejemplo, se comprueba que el usuario presta atención a los iconos de alerta (advertencias automáticas de posibles conflictos) cuando inspecciona visualmente la descripción de los requisitos y otros elementos del proyecto. También se descubre que el usuario ignora frecuentemente el botón de ayuda presente en la barra de herramientas.

El botón de ayuda se propone para aportar información adicional que el usuario podría necesitar sobre la vista en la que se encuentra trabajando, como la definición de los tipos de requisitos en la vista de especificación de requisitos. Sin embargo, el usuario suele preferir explorar las opciones buscando una solución rápida en lugar de leer instrucciones. La barra de navegación también se pone a prueba, verificando que las categorías y opciones son efectivamente intuitivas y coherentes para el usuario. Las pruebas también comprueban la utilidad del formato de lista de elementos expandibles usado en las vistas que muestran datos de requisitos, casos de prueba, módulos y otros elementos del proyecto. En estas vistas se verifica que el usuario entiende el funcionamiento general de estas listas, las cuales permiten realizar búsquedas sobre la descripción de los elementos expandidos y ordenar la lista en función de los atributos principales. La vista de edición de documentos es reajustada para un mejor aprovechamiento del espacio de edición, según recomendaciones de los mismos usuarios.

### 3.3. REIMPLEMENTACIÓN

Una vez teniendo un diseño de interfaz y navegación validado con los usuarios, la siguiente etapa es implementar las funcionalidades propuestas. Lamentablemente es difícil reemplazar la interfaz y navegación original sobre el código existente de ReqAdmin sin primero reparar su funcionamiento realizando un fuerte trabajo de refactoring. Es un trabajo costoso en términos de tiempo y complejidad, además es difícil garantizar la integridad de todas las funcionalidades originales luego de una modificación destinada a

extender la herramienta con nuevas características. Razón por la cual se escoge realizar una reimplementación de ReqAdmin, dando paso a una nueva versión de la herramienta de administración de requisitos llamada MainReq.

En este caso, en reemplazo de la implementación original basada en PHP, se opta por el uso de un framework web que garantice una mayor extensibilidad y reutilización de componentes, además de un desarrollo rápido y amigable con el diseñador, para facilitar la implementación de la nueva interfaz propuesta.



## 4. CONCEPCIÓN DE LA SOLUCIÓN

En esta sección se presentan los principales requisitos de la solución, describiendo lo que se espera del producto final. Luego se describe el ambiente de operación y la estructura física en la sección de arquitectura de la solución. Se presenta el modelo de datos y la relación entre sus principales entidades. En la sección de componentes de la solución se explica de manera general, cómo se relacionan las diferentes partes del sistema.

### 4.1. PRINCIPALES REQUISITOS DE LA SOLUCIÓN

A continuación se presenta un listado con los principales requisitos involucrados en la solución:

- La herramienta debe soportar la **administración de elementos del proyecto**, éstos son Requisitos de Usuario, Requisitos de Software, Módulos, Casos de Prueba, Tipos de Usuario (usuarios del producto final) e Hitos (iteraciones o incrementos en los que se divide el proyecto).
- La herramienta debe generar **mensajes de alerta** en caso de posibles problemas con la especificación de los elementos del proyecto.
- Debe ofrecer **indicadores estadísticos** y de avance del proyecto.
- Debe ofrecer herramientas que permitan **revisar la consistencia y trazabilidad** entre requisitos y otros elementos del proyecto.
- Debe permitir la **edición y descarga de documentos** asociados al proyecto.
- Debe permitir revisar el **historial de cambios** sobre cada elemento del proyecto y sobre los documentos.
- Debe presentar una **interfaz clara y usable**, diseñada en base a principios y reglas de usabilidad, y validada usando pruebas con usuarios. Otorgando un nivel aceptable de confiabilidad.
- Debe permitir la **creación de proyectos y asociarlos a usuarios** del equipo de trabajo.
- Debe permitir asignar **tareas a cumplir por los integrantes** del equipo de desarrollo del proyecto, las cuales serán revisadas dentro de una fecha límite.
- El software debe ser desarrollado en un framework web que otorgue una **mayor extensibilidad** y reutilización de componentes.

- Debe ser **robusto**, para soportar errores en el ingreso de datos a través de formularios, sin experimentar pérdidas importantes de información.

#### 4.2. ARQUITECTURA DE LA SOLUCIÓN

El desarrollo de la nueva herramienta se realiza sobre el framework web Django, servidor Apache y base de datos MySQL. Se escoge Django por ofrecer un entorno de desarrollo rápido y de alto nivel, permitiendo una mayor extensibilidad y reutilización de componentes, dividiendo la lógica de la aplicación en módulos e incorporando un sistema de plantillas HTML amigable con el diseñador. Además provee otras facilidades como la autogeneración de un módulo para la administración de la herramienta o la capacidad de especificar la asociación entre la URL que ve el usuario y el controlador de la vista respectiva.

Para la generación de documentos en formato PDF desde plantillas HTML se utiliza un módulo integrado a Django llamado "xhtml2pdf", de código abierto disponible en github.com. Se utiliza una versión adaptada a Django de TinyMCE como editor de texto enriquecido para la sección de documentos, junto con un plugin para cargar imágenes al servidor. Las versiones de las herramientas utilizadas en el desarrollo son las siguientes:

- Python 2.7.4
- Django 1.6.2
- MySQL 5.5.2
- Apache 2.2.22
- xhtml2pdf 0.0.6
- Mozilla Firefox 26.0
- TinyMCE 3.5.8

Django es un framework web en Python que opera según un esquema Model-View-Template (MVT), en el cual la petición HTTP del usuario desencadena una respuesta de un método en el módulo "View", mapeado según la dirección URL, que realiza las consultas necesarias al "Model" y le entrega los datos necesarios a un "Template" HTML determinado que se retorna como respuesta al usuario cliente. De esta manera gran parte de la lógica de la herramienta se encuentra en el módulo "View" (ver Figura 4.1).

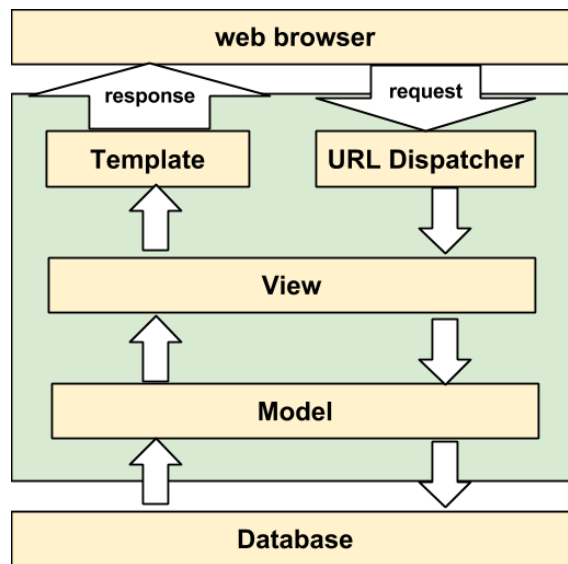


Figura 4.1: Arquitectura de la solución.

El código se encuentra estructurado en directorios con archivos Python bajo un esquema predeterminado por Django. Los directorios más importantes son los siguientes:

- **MainRec:** Aquí se encuentra el archivo de configuración del proyecto Django (`settings.py`), responsable de los parámetros de conexión con la base de datos, la declaración de las aplicaciones que se usan en este proyecto y otras opciones particulares de estas mismas aplicaciones.
- **MainApp:** Este directorio contiene la aplicación principal de la herramienta, la cual contiene los módulos "View" y "Model". El módulo "Model" se encuentra en el archivo "models.py", el cual contiene definiciones de las clases y métodos que se utilizan como medio de acceso a las entidades almacenadas en la base de datos. El módulo "View", a través de métodos en archivos "views.py", accede a datos del modelo y prepara la información requerida por la plantilla HTML que se usa para mostrar la respuesta al usuario. También se encuentra el archivo que se encarga de asociar las URL a las que acceden los clientes/usuarios (`urls.py`) con los métodos del módulo "View" respectivos. El archivo "admin.py" permite personalizar la vista de administración de la herramienta.
- **Templates:** Tal como su nombre lo indica, este directorio contiene todos los archivos HTML que son enviados al cliente en respuesta a su petición http. El sistema de herencia de plantillas de Django permite extender estas plantillas y reutilizarlas como componentes modulares en las vistas.

- **Media:** Este directorio alberga los archivos de imágenes que los usuarios incorporan a los documentos de su proyecto.

#### 4.3. MODELO DE DATOS

El modelo de datos se puede dividir en dos grupos, el primer grupo (ver Figura 4.2) consta de las tablas UserRequirement, SoftwareRequirement, Module, TestCase, UserType e Increment. Este grupo describe cómo se relacionan los principales elementos de un proyecto. El segundo grupo (ver Figura 4.3) consta de las tablas Project, Registry, DocumentSection y Task. Este grupo relaciona los documentos, las tareas de los integrantes y los elementos descritos en el primer grupo, con un proyecto en particular.

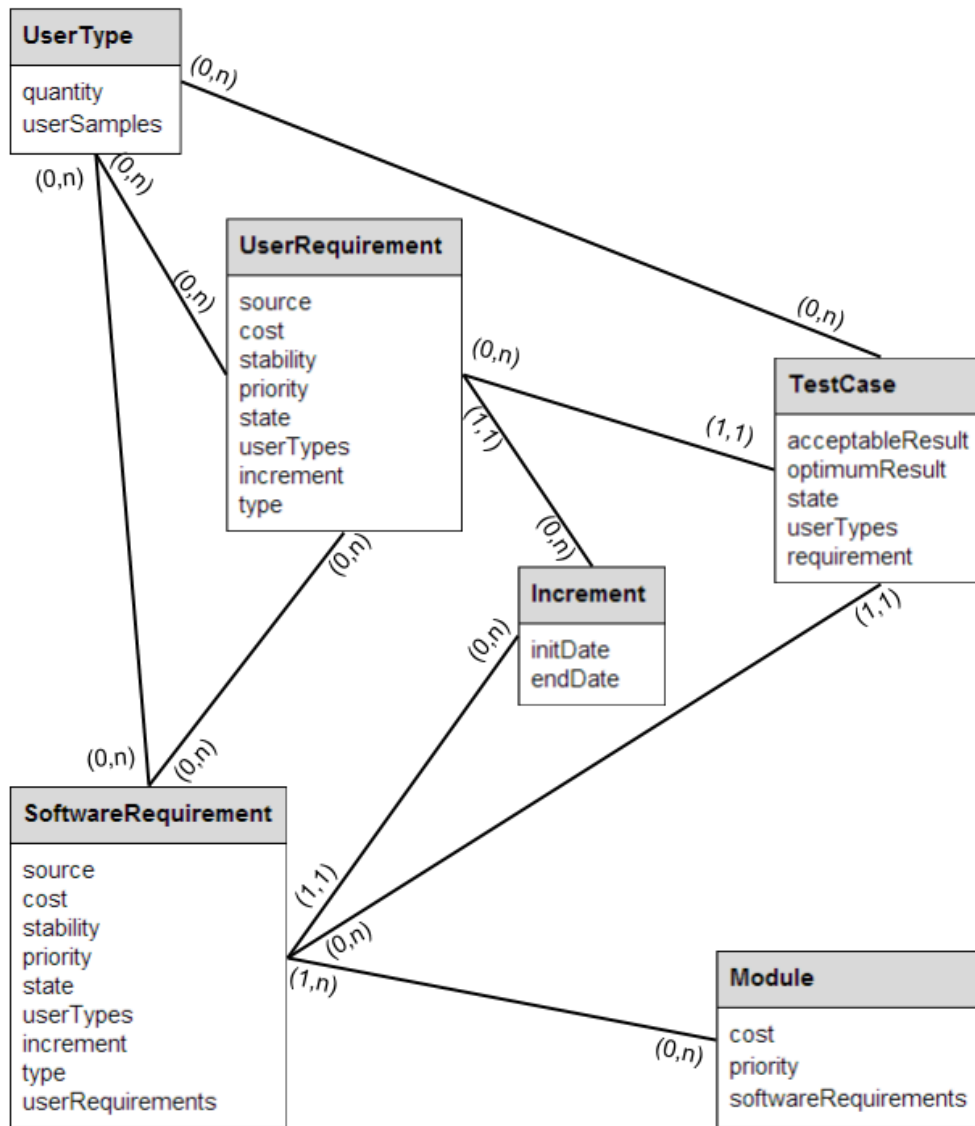


Figura 4.2: Relaciones entre elementos de un proyecto.

A continuación se describen las principales entidades que conforman el modelo de datos de la nueva herramienta de administración de requisitos:

- **Project:** almacena la información básica que determina la identidad de un proyecto.
- **Registry:** almacena la información necesaria para el funcionamiento de la bitácora respectiva a cada elemento del proyecto. Todas las tablas del primer grupo de entidades del modelo (ver Figura 6) heredan de la tabla Registry, por lo tanto se encuentran asociado a un único proyecto. Además dado que ningún registro es borrado realmente de la base de datos se incorpora un campo que indica la vigencia de éste
- **DocumentSection:** un documento se conforma a partir de secciones de distinto tipo, y esta entidad almacena la evolución de cada tipo de sección. Cada sección está relacionado con un único proyecto.
- **Task:** esta entidad da soporte a la funcionalidad que se encarga de administrar las tareas asignadas a los integrantes del equipo de desarrollo. Cada tarea está relacionada con un único proyecto.
- **Increment:** almacena hitos como un rango de fechas significativas para el proyecto asociado. Todo requisito debe tener un hito asociado.
- **UserType:** almacena los tipos de usuarios asociados al producto de software del proyecto. Puede estar relacionado con requisitos o casos de prueba.
- **UserRequirement:** almacena la información que define un requisito de usuario. Deben estar asociados a un hito (de la tabla Increment), y pueden estar asociados a casos de prueba, requisitos de software y tipos de usuario.
- **SoftwareRequirement:** almacena la información que define un requisito de software. Deben estar asociados a un hito (de la tabla Increment), y pueden estar asociados a casos de prueba, requisitos de usuario y tipos de usuario.
- **Module:** almacena la información que define un módulo. Puede tener requisitos de software asociados.
- **TestCase:** almacena la información que define un caso de prueba. Debe tener un requisito asociado y puede tener tipos de usuario relacionados.

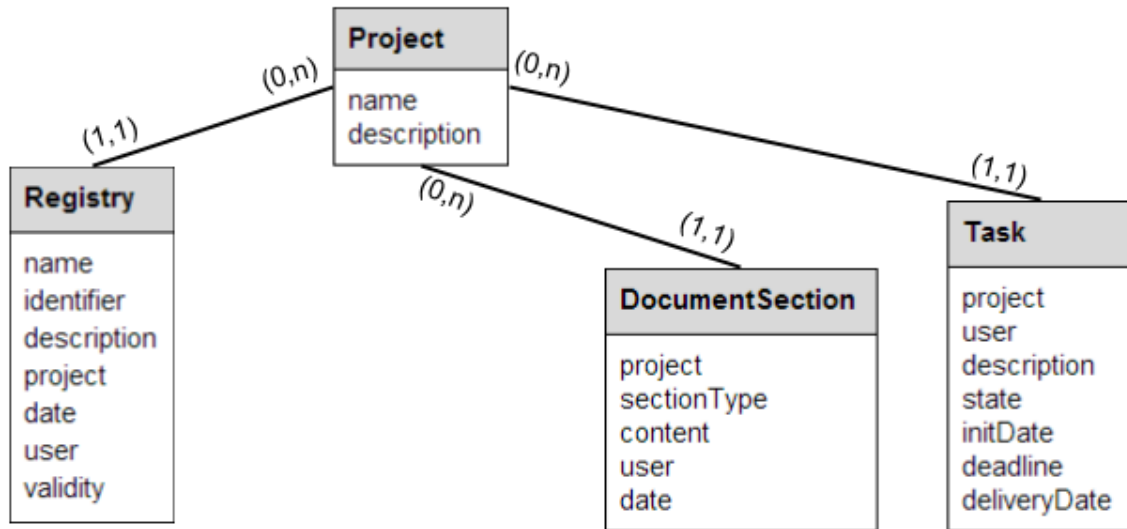


Figura 4.3: Relaciones entre los documentos, las tareas y los elementos de un proyecto.

#### 4.4. COMPONENTES DE LA SOLUCIÓN

La solución se encuentra estructurada en 10 componentes (ver Figura 4.4), clasificados en 4 tipos: los componentes de administración general, los componentes relacionados con el ingreso y visualización de datos del proyecto (edición de elementos y documentos del proyecto), los componentes que operan de manera transversal en la aplicación (alarmas y generación de documentos PDF) y los componentes de indicadores y herramientas.

- **Administración general:** Este es el componente desde el cual un usuario administrador puede crear un proyecto y asociar integrantes (generalmente alumnos), incluyendo la asignación de permisos de edición para los usuarios dentro del proyecto. La asignación de permisos es relevante cuanto a los distintos roles que pueden tener los integrantes del equipo de alumnos que trabajará en el proyecto.
- **Administración de elementos del proyecto:** Este es el componente que se encarga del ingreso de datos y relaciones de los elementos del proyecto: Hitos, Tipos de Usuario, Requisitos de Usuario, Requisitos de Software, Módulos y Casos de Prueba. Los atributos de estos elementos y sus relaciones con otros elementos son editados a partir de formularios y una visualización en formato de lista de elementos expandibles permite revisar la información de cada elemento asociado al proyecto. En éste componente se realiza la validación de formularios necesarios y se almacenan todas las modificaciones sobre algún

elemento en registros nuevos, permitiendo el funcionamiento de la Bitácora.

- **Documentos del Proyecto:** Este componente es el encargado de presentar en formato de informe el estado del proyecto, permitiendo la edición de los principales documentos requeridos: Documento de Requisitos, Documento de Diseño, Documento de Casos de Prueba y el Documento Histórico. Consta básicamente de un editor de texto enriquecido para cada sección del documento respectivo capaz de incorporar imágenes y se establece un formato de informe a partir de una plantilla HTML (incluye portada, índice de contenido automático, número de página y estilos personalizables). Cada edición sobre alguna sección de un documento se almacena en la base de datos (no se sobre escribe), permitiendo revisar y recuperar una versión pasada desde un historial de modificaciones disponible para el usuario editor.
- **Generador PDF:** Un componente muy importante para los Documentos del Proyecto consiste en un generador de archivos en formato PDF. Esto permite la visualización y descarga directamente desde el navegador, no sólo de los informes anteriormente mencionados, sino que también permite visualizar en un documento los distintos elementos del proyecto en listados detallados, junto con otros indicadores del proyecto, como las matrices de trazado y estadísticas. El archivo PDF es generado a partir de plantillas HTML con información proveniente de otros componentes, como los Documentos del Proyecto o los Elementos del Proyecto.
- **Alarmas:** Este componente permite detectar tempranamente ciertas inconsistencias en la especificación de los elementos del proyecto, principalmente requisitos. Dado que las relaciones que se establecen entre elementos del proyecto, particularmente entre requisitos de usuario y de software, pueden generar inconsistencias causadas por la compleja visibilidad de conflictos al momento de ingresar datos, es de gran utilidad automatizar la mayor cantidad posible de chequeos de consistencia y aumentar así la visibilidad de los conflictos a tiempo. Se analizan posibles Alarmas para cada elemento desplegado por los componentes de Administración de Elementos del Proyecto y de Consistencia de Elementos Relacionados, mostrando un icono específico al lado de la descripción del elemento respectivo.
- **Estadísticas:** El componente encargado de las Estadísticas despliega información de avance de Requisitos de Usuario, Requisitos de

Software, Casos de Prueba y Módulos. Más específicamente, realiza un conteo de elementos separado por atributos (como el nivel de cumplimiento) e Hito del proyecto y los despliega de manera resumida en tablas. Además se realiza un conteo de elementos conflictivos, como requisitos sin casos de prueba asociados.

- **Matrices de Trazado:** El componente encargado de las Matrices de Trazado sirve para visualizar las relaciones entre dos tipos de elementos como intersecciones en una matriz. Por ejemplo, sirve para detectar si un requisito de usuario tiene demasiados requisitos de software asociados o si no tiene ninguno asociado, significando que ese requisito de usuario debería ser dividido en partes menos complejas o que no sería considerado al momento de la implementación respectivamente. Este componente obtiene la información de dependencias entre elementos desde el modelo de datos y crea una tabla que se incluye en la interfaz del usuario dependiendo del tipo de matriz que haya solicitado, destacando automáticamente los posibles conflictos.
- **Consistencia de Elementos Relacionados:** Este componente opera de manera similar a las Matrices de Trazado, mostrando cómo se relacionan dos tipos de elementos, sin embargo su diseño está enfocado en comparar lado a lado el contenido (descripción y otros atributos) de dos elementos relacionados permitiendo verificar la consistencia entre éstos. Utiliza el mismo formato de "lista de elementos con contenido expandible" que el componente de Administración de Elementos del Proyecto para visualizar toda la información necesaria, incluyendo las Alarmas correspondientes sobre cada elemento.
- **Bitácora:** Este componente rescata la evolución de cada elemento del proyecto directamente desde el modelo de datos (entidad "Registry"), mostrando en una lista las distintas versiones de éstos, junto con la fecha de modificación, el usuario responsable y el estado del registro ("creado", "modificado" y "actual").
- **Tareas:** Finalmente tenemos el componente encargado de las Tareas asignadas a los integrantes encargados del proyecto, incorporando a la herramienta un modo de monitorear y organizar el trabajo designado por los jefes de proyecto y realizado por los integrantes.



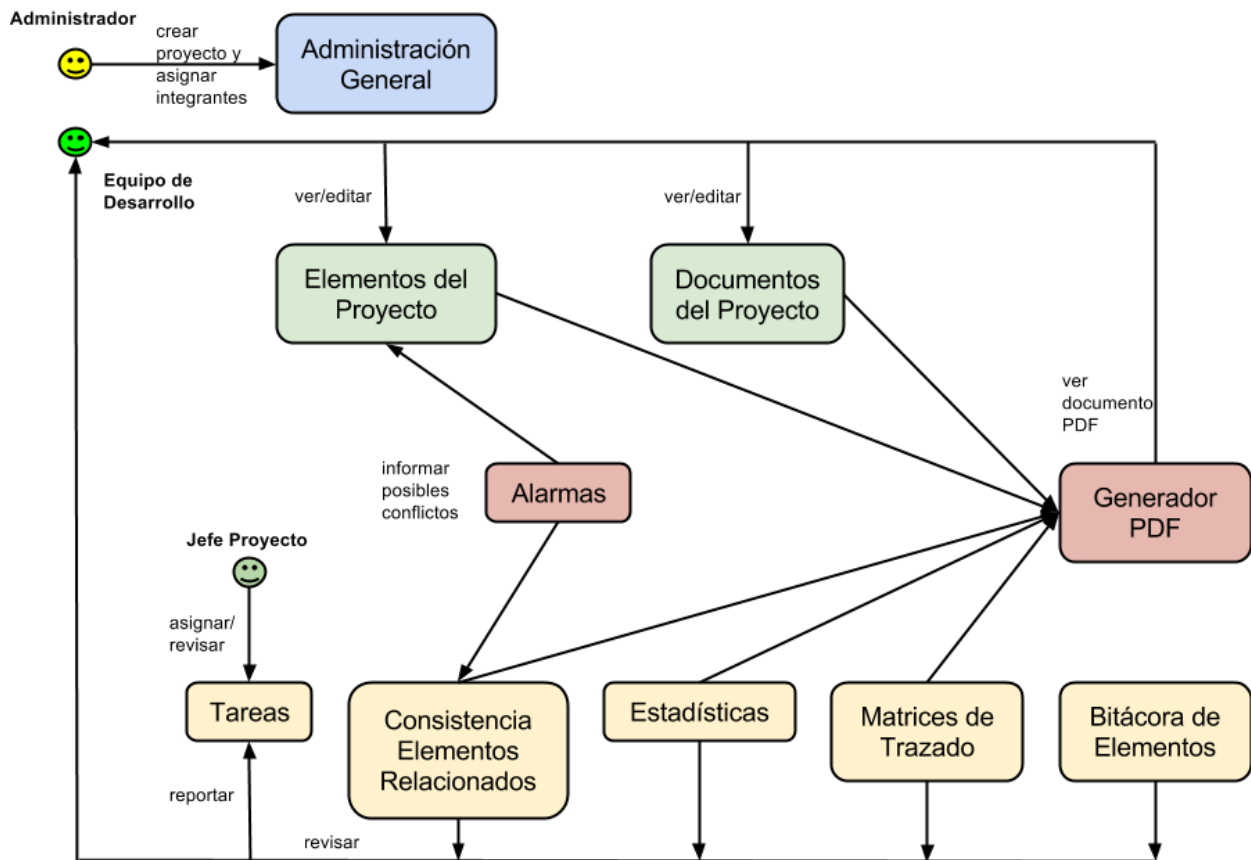


Figura 4.4: Componentes de la solución.

## 5. IMPLEMENTACIÓN DE LA SOLUCIÓN

El propósito de esta herramienta es facilitar el proceso de especificación de los requisitos de un proyecto de software para todos los involucrados en el desarrollo de éste. Este trabajo se encuentra principalmente basado en ReqAdmin, y por lo tanto incorpora gran parte de sus características como herramienta de administración de requisitos. Al igual que ReqAdmin, las actividades que un usuario de la herramienta puede realizar se pueden clasificar en dos tipos: actualización de datos (ingreso y edición) y obtención de reportes e indicadores de un proyecto. Sin embargo un proceso de diagnóstico de la situación actual y rediseño enfocado en resolver los problemas de los usuarios (principalmente alumnos) ha resultado no sólo en un nuevo aspecto visual sino que también en modificaciones y adición de varias características funcionales, a modo de ofrecer un producto más útil y centrado en el usuario final.

## 5.1. PROCESO DE ESPECIFICACIÓN DE REQUISITOS

El proceso de especificación de requisitos heredado de ReqAdmin consta de las siguientes etapas (ver Figura 5.1):

- **Creación del proyecto y designación de un equipo de desarrollo:** Un administrador de MainReq utiliza el componente de Administración General para crear un proyecto y designar un equipo de desarrollo a cargo de éste.
- **Definición de hitos y tipos de usuarios del proyecto:** Un analista se encarga de especificar los hitos del proyecto, correspondientes a fechas de entrega de avance del sistema desarrollado, y además se encarga de establecer los tipos de usuario que están relacionados con el sistema final.
- **Definición de requisitos de usuario:** El analista debe ingresar los requisitos de usuario especificando sus atributos, asociándolos con los tipos de usuario respectivos y con un hito específico del proyecto para distribuir trabajo durante el semestre. Al final de esta etapa los requisitos de usuario son formalizados en un documento.
- **Definición de requisitos de software:** El analista ingresa los requisitos de software en base al documento generado en la etapa anterior, salvo que éstos además deben estar vinculados con sus respectivos requisitos de usuario. Al final de esta etapa los requisitos son formalizados en un documento.
- **Definición de casos de pruebas para los requisitos:** Se utilizan casos de prueba como método de revisión de cumplimiento de los requisitos sobre el sistema desarrollado, y por lo tanto en esta etapa el téster debe ingresar los casos de prueba vinculados a cada requisito formalizado en las etapas anteriores.
- **Diseño arquitectónico y diseño detallado:** En base a la formalización de los requerimientos especificados en las etapas anteriores, el diseñador debe generar un documento con el diseño arquitectónico, especificando además un diseño detallado separando las características del producto de software propuesto e ingresando los módulos que estructuran la solución a implementar. Cada módulo ingresado debe estar vinculado con sus requisitos de software respectivos. Al final de esta etapa se formaliza el diseño de la solución en un documento.

- **Evaluar y actualizar el nivel de cumplimiento del trabajo desarrollado:** Luego de que los implementadores presenten versiones del software desarrollado hasta la fecha (Hito) se procede a probar su funcionamiento según los casos de prueba especificados anteriormente, actualizando el estado de cumplimiento de éstos y por consiguiente el estado de los requisitos asociados.
- **Documentación final del proyecto:** El trabajo realizado por el equipo de desarrollo debe ser finalmente registrado en un documento histórico.

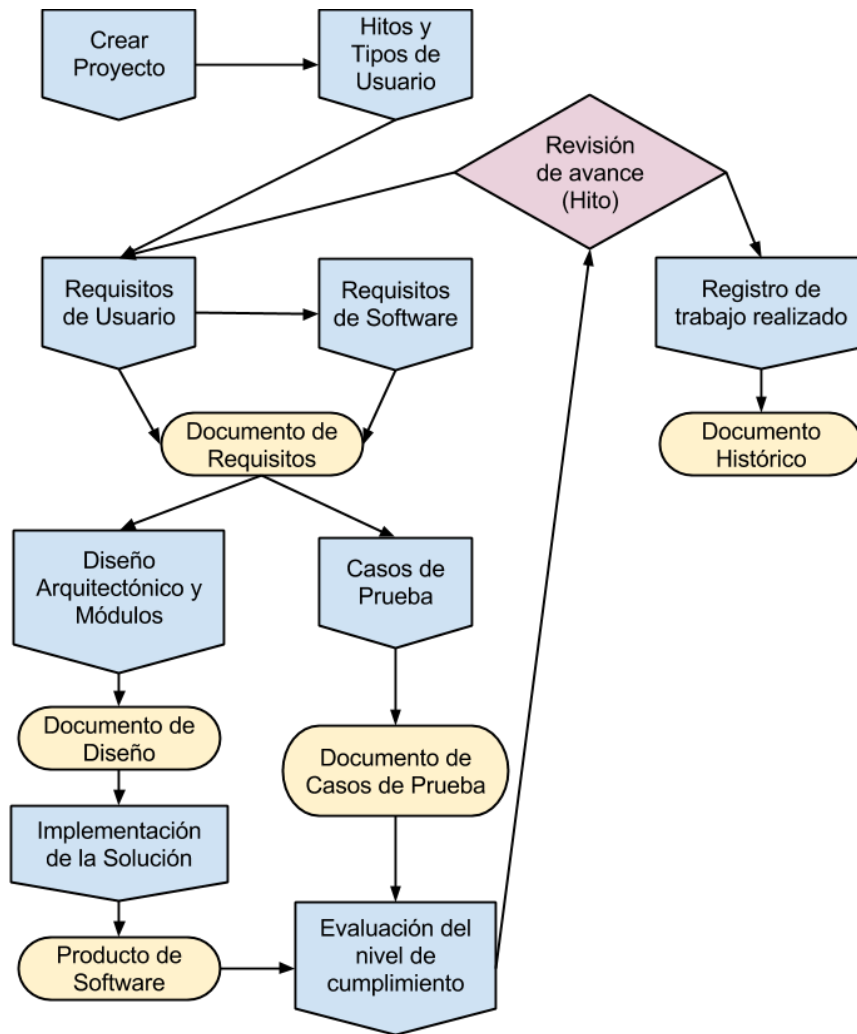


Figura 5.1: Proceso de especificación de requisitos.

Las siguientes secciones presentan las principales interfaces y características de MainReq. Se muestra el propósito y el funcionamiento de cada módulo de las vistas, utilizando ejemplos de uso y especificando el formato de ingreso de datos para requisitos y otros elementos relacionados con el desarrollo de un proyecto de software.

## 5.2. MAPA DE NAVEGACIÓN

A continuación se presenta el mapa de navegación de la nueva herramienta de administración de requisitos, MainReq. La herramienta tiene dos tipos de acceso, uno de administración para la creación de proyectos y equipos de desarrollo, y otro para que un equipo de desarrollo pueda trabajar sobre el proyecto asignado:

- Administración
  - Proyectos
  - Usuarios
  - Permisos y Grupos de Usuarios
- Inicio
  - Seleccionar Proyecto
    - Proyecto
      - Hitos
      - Tipos de Usuario
      - Requisitos de Usuario
      - Requisitos de Software
      - Módulos
      - Casos de Prueba
      - Documento de Requisitos
      - Documento de Diseño
      - Documento de Casos de Prueba
      - Documento Histórico
      - Tareas
      - Estadísticas
      - Matrices de Trazado
      - Consistencia de Relaciones
      - Bitácora



**mainReq**

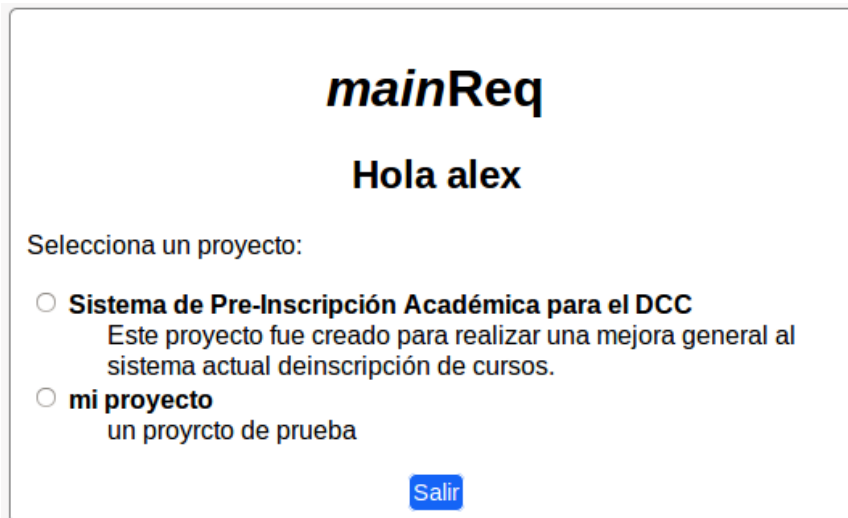
Usuario

Contraseña

Ingresar

Figura 5.2: Vista de ingreso.

Los usuarios pueden acceder a sus proyectos asociados después del ingreso (ver Figura 5.2) a través de la vista de selección de proyectos (ver Figura 5.3), la que los muestra ordenados por fecha de creación junto con la descripción de cada uno. En el caso de que sólo tenga un proyecto asociado se omite la selección de proyectos, y se le informa su situación en caso de no tener ningún proyecto asociado aún.



**mainReq**

**Hola alex**

Selecciona un proyecto:

- Sistema de Pre-Inscripción Académica para el DCC**  
Este proyecto fue creado para realizar una mejora general al sistema actual de inscripción de cursos.
- mi proyecto**  
un proycto de prueba

Salir

Figura 5.3: Selección de proyecto a revisar.

La barra de navegación pone a disposición de los usuarios tres secciones principales, correspondientes a la visualización y edición de elementos del proyecto, vista y edición de los principales documentos asociados al proyecto y una sección con herramientas e indicadores (ver Figura 5.4). En la esquina superior se pueden revisar los datos del usuario y botón para cerrar la sesión.



Figura 5.4: Barra de navegación.

### 5.3. ADMINISTRACIÓN

Al inicio del curso de Ingeniería de Software II se designan proyectos de software reales a grupos de alumnos, los cuales asumen roles y responsabilidades específicas dentro del grupo, típicamente con los siguientes cargos: Jefe de Proyecto, Analista, Diseñador, Implementador y Téster.

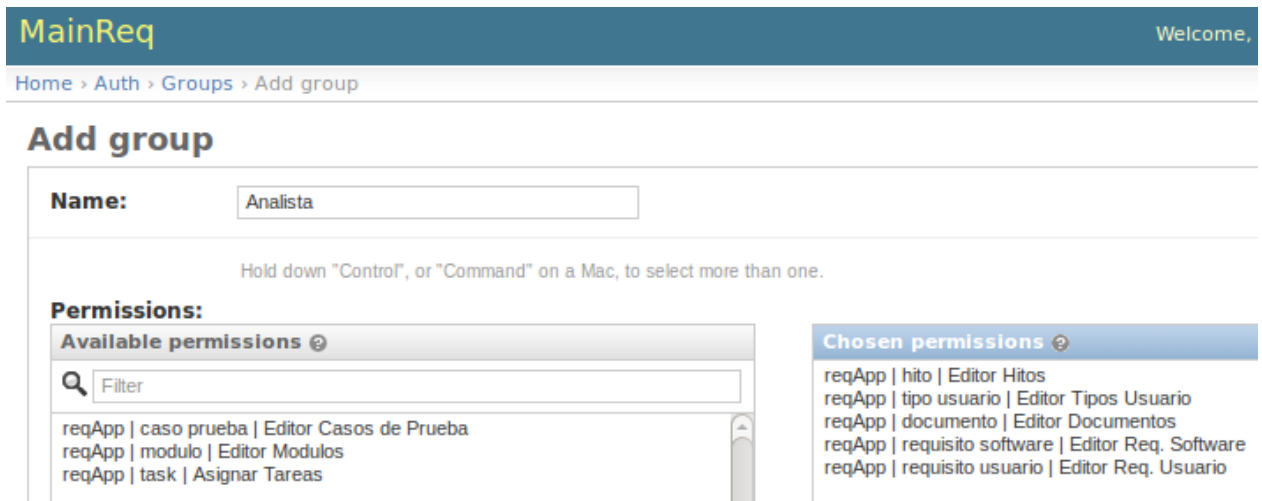


Figura 5.5: Asignación de permisos sobre un grupo de usuarios (Grupo: Analista).

En la vista de administración es posible implementar estos cargos utilizando grupos de usuarios con permisos de edición específicos (ver Figura 5.5). En esta instancia creamos los siguientes grupos con sus permisos asociados:

- Jefe Proyecto
  - Asignar Tareas
- Analista
  - Editor Hitos
  - Editor Tipos Usuario
  - Editor Requisitos de Usuario

- Editor Requisitos de Software
- Editor Documentos
- Diseñador
  - Editor Módulos
  - Editor Documentos
- Téster
  - Editor Casos de Prueba

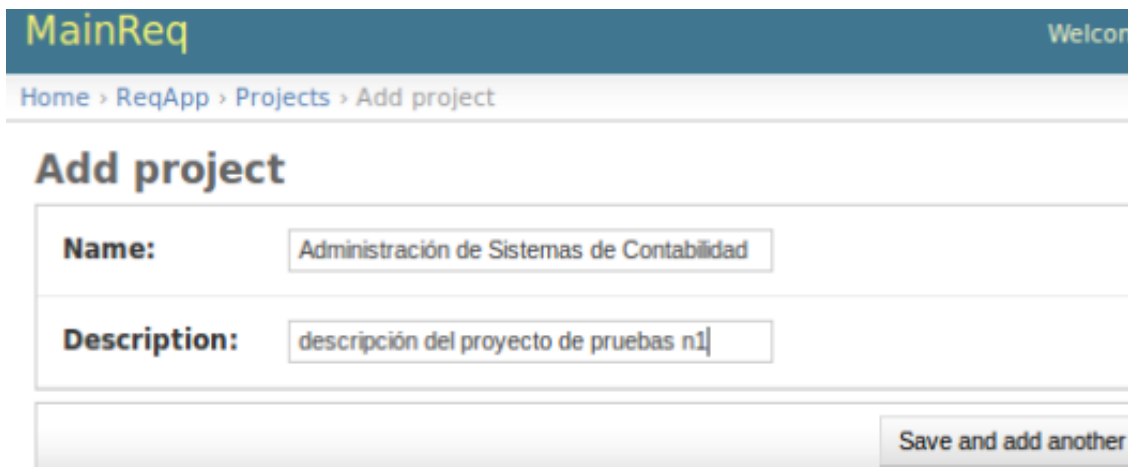


Figura 5.6: Creación de un Proyecto.

Es posible crear un proyecto simplemente especificando su nombre y una descripción (ver Figura 5.6), descripción visible en la vista de selección de proyecto. Además se pueden crear usuarios ingresando sus datos esenciales: usuario, nombre y correo electrónico, y asignarles grupos con permisos de edición (ver Figura 5.7).

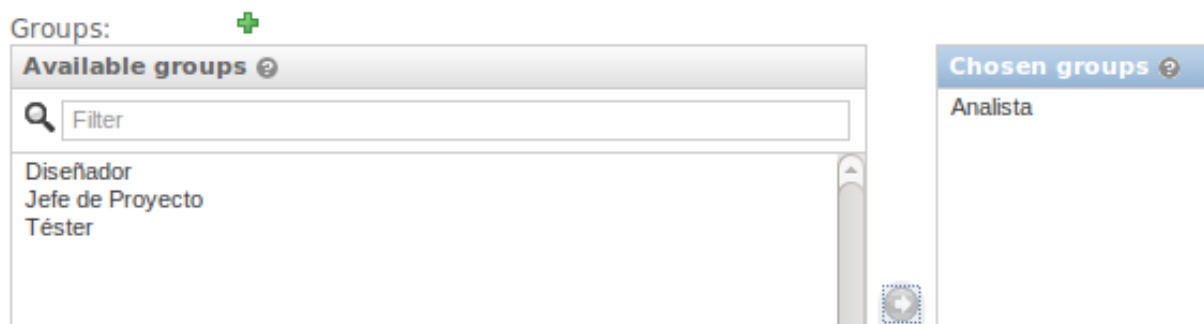


Figura 5.7: Incorporar usuario dentro de un grupo.

El correo electrónico asociado se usa para enviar mensajes desde la herramienta, en este caso se puede enviar el usuario y clave de acceso

(autogenerada) a usuarios de MainReq (ver Figura 5.8), lo que es útil en el caso de olvido de contraseña, pues con un clic se autogenera una nueva contraseña y se envía al correo asociado.

The screenshot shows the MainReq interface. At the top, there is a blue header with the text 'MainReq'. Below it, a breadcrumb trail reads 'Home > Auth > Users'. A list of four yellow notification boxes with green checkmarks indicates successful password changes and email notifications for users 'alex', 'rod', and 'tod', plus a summary for '3 passwords changed & sent by email.'. Below the notifications is a section titled 'Select user to change'. This section includes a search bar with a magnifying glass icon and a 'Search' button. Below the search bar is an 'Action:' dropdown menu set to 'Re-assign password & notify by email', a 'Go' button, and the text '3 of 5 selected'. A table with columns 'Username', 'Email address', and 'First name' lists five users. The first two, 'admin' and 'alejandro', are unselected. The last three, 'alex', 'rod', and 'tod', are selected with red checkmarks. At the bottom of the table area, it says '5 users'.

<input type="checkbox"/>	Username	Email address	First name
<input type="checkbox"/>	admin	admin@admin.com	
<input type="checkbox"/>	alejandro		john
<input checked="" type="checkbox"/>	alex	el.lokillo@gmail.com	Alex
<input checked="" type="checkbox"/>	rod	AThown2247@fleckens.hu	5818
<input checked="" type="checkbox"/>	tod	mainreq.90@gmail.com	5556

Figura 5.8: Autogenerar nueva clave de acceso, y enviarla por correo electrónico.

Las secciones de MainReq disponibles para el equipo de desarrollo son presentadas a continuación simulando los pasos necesarios para el desarrollo de un proyecto de software.

#### 5.4. PROYECTO

En esta sección se pueden revisar y editar los elementos del proyecto, éstos son: Hitos, Tipos de Usuario, Requisitos de Usuario, Requisitos de Software, Módulos y Casos de Prueba. Éstos son visualizados con un formato de lista de elementos de contenido expandible (ver Figura 5.9), es decir, cada elemento presenta su título identificador junto a sus principales atributos en un registro horizontal, el cual despliega o contrae con una animación el resto de sus atributos y descripción una vez que se hace un clic en el título identificador. Incluido en el título se encuentra un código único



con un formato específico usado para referenciar formalmente al elemento dentro del proyecto. De este modo se ofrece una vista de lista simplificada, para realizar revisiones o comparaciones visuales rápidas sobre los principales atributos de los elementos, y también se ofrece una vista detallada de los objetos para poder revisarlos en mayor profundidad.

identificador	nombre	estado	costo	prioridad	tipo	hito
RU0001	Usuarios Sistema		0	urgente	funcional	Primera iteración
RU0002	Usuario Profesor		0	normal	funcional	Segunda iteración
RU0003	Login U-Pasaporte		0	urgente	funcional	Primera iteración
RU0004	Permisos Super-Admin		0	urgente	funcional	Primera iteración
RU0005	Permisos Profesor Super-Admin		0	normal	funcional	Segunda iteración
RU0006	Catálogo de Cursos		0	urgente	funcional	Primera iteración
RU0007	Catálogo Historico		0	urgente	funcional	Primera iteración

Estado: no cumple

El sistema debe contar con un Catalogo Historico [ver sección 1.4] que un Administrativo pueda actualizar de manera manual sobre cada curso en el sistema. Se debe poder seleccionar de entre el total de cursos cuales se desea considerar en el Catalogo de Resta para ofrecer a los Alumnos.

Fuente: equipo, cliente  
Estabilidad: intransable

Usuarios asociados(1):  
o TU0002 Administrativo

Figura 5.9: Vista de requisitos de usuario en formato de elementos de contenido expandible.

Una barra con botones permiten realizar acciones sobre estas listas, como ofrecer la opción de expandir o contraer todos los elementos listados con un clic, permitiendo realizar búsquedas de conceptos (función para “buscar” en navegadores web) sobre las descripciones de los elementos. También se puede crear un nuevo elemento en el caso de tener los permisos de edición correspondientes.

En el caso de que el usuario requiera revisar la lista de elementos “offline” en un archivo o documento impreso, puede descargar un documento en formato PDF presionando el botón de “documento PDF”, con una portada que indica el proyecto, la fecha y el tipo de documento generado. El usuario puede ordenar la lista de acuerdo a los atributos principales. Los estados de cumplimiento de requisitos y casos de prueba son mostrados con colores especificados sobre la lista.

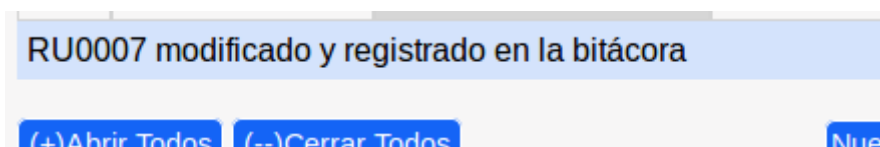


Figura 5.10: Mensajes informan la acción realizada por el sistema.

En caso de tener permisos de modificación, el usuario puede acceder al formulario de edición del elemento presionando el botón de edición en la esquina inferior izquierda del contenido del elemento. Acá también se puede eliminar el elemento de la lista sólo una vez que el usuario confirme la acción, tomando en cuenta que toda acción del usuario debe generar una respuesta del sistema, junto con mensajes que adviertan debidamente las consecuencias.

El botón "Nuevo" posiciona la vista al final de la lista y despliega el formulario para crear un nuevo elemento. Cada creación, modificación o eliminación muestra un mensaje al usuario con la acción realizada sobre el elemento respectivo, y además informa del registro de la acción en la bitácora de cambios (ver Figura 5.10). En el caso de que el formulario enviado tenga algún error sobre un campo (como campos requeridos no especificados, descripciones muy largas o rangos de fechas inválidos) se indica el conflicto sobre el campo respectivo (ver Figura 5.11) sin perder los datos ingresados, añadiendo un paso de pre validación sobre el ingreso de datos.

<b>RU0005 Permisos Profesor Super-Admin</b>	0	normal	funcional	Segunda iteración
<b>RU0006 Catálogo de Cursos</b>	0	urgente	funcional	Primera iteración

**RU0007**

\*Nombre:

\*Descripción: Ensure this value has at most 5000 characters (it has 7776).

\*Hito:

Usuarios asociados:  (presionar "Ctrl" para seleccionar)

\*Prioridad:

\*Estabilidad:

\*Tipo:

\*Fuente:

\*Estado:

Costo:

<b>RU0008 Catalogo de un proceso</b>	0	urgente	funcional	Primera iteración
<b>RU0009 Proceso de Principiación</b>	0	urgente	funcional	Primera iteración

Figura 5.11: Edición de elementos dentro de la lista, con sistema de pre validación de campos de formularios.

La especificación de elementos del proyecto puede llegar a ser compleja y garantizar la consistencia de éstos es una tarea difícil, por lo que

se dispone de un sistema de alarmas automáticas que advierten al usuario sobre posibles conflictos en la especificación de algún elemento. En el caso de que sea necesario mostrar una alarma, ésta se ubica a la derecha de los atributos principales en la vista "comprimida", facilitando su pronta detección. Mediante un clic sobre el ícono de alarma el usuario puede revisar la lista de posibles conflictos relacionados a ese elemento (ver Figura 5.12).

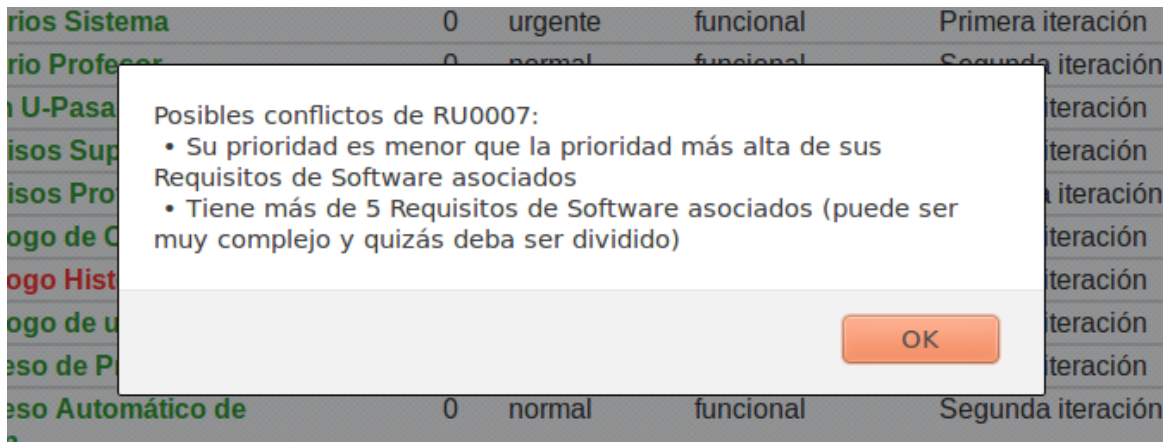


Figura 5.12: Alarma automática sobre un Requisito de Usuario.

La consistencia se considera como un factor importante para el diseño de la interfaz final. Por ejemplo, siempre se utilizan botones azules para indicar las posibles acciones del usuario. Siempre se muestra el significado de los colores usados sobre los elementos de lista (verde, rojo y morado para indicar el estado de cumplimiento de requisitos y casos de prueba). Otra buena característica que ayuda a una mejor experiencia de usuario, es que todas las modificaciones se realizan sobre la misma lista de elementos, permitiendo tener siempre disponible la información del resto de la lista al momento de editar el formulario. Un botón de ayuda presente en todas las secciones de la herramienta permite al usuario informarse sobre las definiciones y especificaciones necesarias para entender el funcionamiento específico de la herramienta.

#### 5.4.1. HITOS

Para la etapa de "Definición de hitos y tipos de usuarios del proyecto" un analista ingresa a la sección "Proyecto/Hitos", luego abre un formulario para crear un nuevo Hito ingresando:

- **Identificador:** Código único que sirve para identificar y referenciar formalmente este elemento dentro del proyecto. Es asignado automáticamente de manera incremental utilizando el formato "HTXXXX".

- **Nombre:** Nombre en lenguaje normal del requisito. El formulario ofrece un campo de texto para ingresar este atributo.
- **Descripción:** Descripción del período marcado por este hito. El formulario ofrece un área de texto para ingresar este atributo.
- **Fecha inicio y Fecha fin:** Las fechas (con el formato "AAAA-MM-DD hh:mm") de inicio y fin que contemplan la etapa de desarrollo definida por este Hito. El formulario ofrece un campo de texto para ingresar estos atributos.

Figura 5.13: Formulario de creación de Hitos.

Durante el curso de Ingeniería de Software II típicamente se realizan dos instancias de evaluación del proyecto realizado, denominadas generalmente como "Primera iteración" y "Segunda iteración".

La sección de ayuda correspondiente a los Hitos muestra como se define un hito del proyecto. Las alarmas automáticas programadas para los Hitos son las siguientes:

- "Este Hito no ha sido asignado a ningún Requisito de Usuario"
- "Este Hito no ha sido asignado a ningún Requisito de Software"

identificador	nombre
HT0001	Primera iteración
HT0002	Segunda iteración
Esta es la última iteración del proyecto. debe presentarse al final del semestre junto con la documentación necesaira para la implantación del producto y el documento histórico respectivo firmado por el cuerpo docente a cargo y el cliente.	
Fecha inicio:	2014-06-06 17:28
Fecha fin:	2014-07-06 23:59

Figura 5.14: Vista comprimida y expandida de Hitos, con atributos principales en barra de ordenamiento.

#### 5.4.2. TIPOS DE USUARIO

Para la etapa de "Definición de hitos y tipos de usuarios del proyecto" un analista ingresa a la sección "Proyecto/Tipos de Usuario", luego abre un formulario para crear un nuevo Tipo de Usuario ingresando:

- **Identificador:** Código único que sirve para identificar y referenciar formalmente este elemento dentro del proyecto. Es asignado automáticamente de manera incremental utilizando el formato "TUXXXX".
- **Nombre:** Nombre en lenguaje normal del Tipo de Usuario. El formulario ofrece un campo de texto para ingresar este atributo.
- **Descripción:** Aquí se describen las características de este Tipo de Usuario. El formulario ofrece un área de texto para ingresar este atributo.
- **Cantidad:** Número entero positivo correspondiente a una estimación de la cantidad de usuarios del sistema que son de este tipo. El formulario ofrece un campo de texto para ingresar este atributo.
- **Usuarios Localizables:** Listado de nombre y datos de contacto (ej.: correo electrónico y teléfono) de usuarios conocidos y localizables durante el desarrollo del proyecto. Se ofrece un área de texto para especificar este listado, cuyos elementos son separados por un punto y coma ";".

The screenshot shows a web form for creating a user type. It contains the following fields and content:

- \*Nombre:** A text input field containing the word "Administrativo".
- \*Descripción:** A text area containing the text: "Usuario con permisos para iniciar, cerrar o reabrir una preinscripcion academica, actualizar el Catalogo de Cursos, modificar el Catalogo Historico y obtener reportes."
- \*Cantidad:** A text input field containing the number "2".
- Usuarios contactables:** A text area containing the text: "Angelica Aguirre [angelica@dcc.uchile.cl]; Sandra Gaez [sandra@dcc.uchile.cl]". To the right of this field is the instruction "Usar semicolon (;) para separar."

At the bottom of the form are two buttons: "Cancelar" and "Guardar".

Figura 5.15: Formulario de creación de Tipos de Usuario.

La sección de ayuda muestra cómo se definen los Tipos de Usuario de un proyecto. Las alarmas automáticas programadas para los Tipos de Usuario son las siguientes:

- “Este Tipo de Usuario no ha sido asociado a ningún Requisito de Usuario”
- “Este Tipo de Usuario no ha sido asociado a ningún Requisito de Software”
- “Este Tipo de Usuario no ha sido asociado a ningún Caso de Prueba”

identificador	nombre
TU0001	Super-Admin
TU0002	Administrativo
Usuario con permisos para iniciar, cerrar o reabrir una preinscripcion academica, actualizar el Catalogo de Cursos, modificar el Catalogo Historico y obtener reportes.	
Cantidad de usuarios: 2	Usuarios contactables: <ul style="list-style-type: none"> <li>o Angelica Aguirre [angelica@dcc.uchile.cl]</li> <li>o Sandra Gaez [sandra@dcc.uchile.cl]</li> </ul>

Figura 5.16: Vista comprimida y expandida de Tipos de Usuario, con atributos principales en barra de ordenamiento.

#### 5.4.3. REQUISITOS DE USUARIO

Los requisitos en general deben presentar una serie de características tanto individualmente como en grupo. Algunas de las más importantes son:

- **Necesario:** Un requisito es necesario si su omisión provoca una deficiencia en el sistema a construir.
- **Conciso:** Un requisito es conciso si es fácil de leer y entender. Su redacción debe ser simple y clara para aquellos que lo consulten en el futuro.
- **Completo:** Un requisito está completo si no necesita ampliar detalles en su redacción, es decir, si muestra información suficiente para su comprensión.
- **Consistente:** Un requisito es consistente si no se contradice con otro requisito.
- **No ambiguo:** Cuando el lenguaje usado en su definición no causa confusiones al lector y tiene una única interpretación.
- **Verificable:** Un requisito es verificable cuando puede ser cuantificado de manera que permita hacer uso de algún método de verificación.

En la etapa de “Definición de requisitos de usuario” el analista ingresa los requisitos acordados con el usuario/cliente en la sección

“Proyecto/Requisitos de Usuario”. Se requiere ingresar los siguientes atributos para especificar un requisito de usuario:

- **Identificador:** Código único que sirve para identificar y referenciar formalmente este requisito dentro del proyecto. Es asignado automáticamente de manera incremental utilizando el formato “RUXXXX”.
- **Nombre:** Nombre en lenguaje normal del requisito. El formulario ofrece un campo de texto para ingresar este atributo.
- **Descripción:** Descripción del requisito (los aspectos que involucra, en qué consiste, etc.). El formulario ofrece un área de texto para ingresar este atributo.
- **Prioridad:** Nivel de urgencia asociada al requisito. La prioridad de un requisito puede ser “urgente”, “pronto”, “normal” o “cuando haya tiempo”. El formulario ofrece una lista seleccionable.
- **Estabilidad:** Este atributo determina si el requisito puede o no estar sujeto a cambios durante el ciclo de vida del producto de software. La estabilidad puede ser “transable” o “intransable”. El formulario ofrece una lista seleccionable para este atributo.
- **Tipo:** El formulario ofrece una lista seleccionable. Los requisitos de usuario se pueden clasificar en 3 tipos:
  - **Funcional:** Describe una capacidad que debe ser soportada por el producto de software.
  - **De Calidad:** Representan un atributo de calidad del producto, y se pueden expresar en una escala de medición.
  - **De Restricción:** Indican con qué restricciones se desarrollará y/o funcionará el software, en función de costos, tiempos, personal, ambiente operacional, hardware, redes, etc.
- **Fuente:** Documento o persona desde la cual surge el requisito. El formulario ofrece un campo de texto para ingresar este atributo.
- **Estado:** El estado de cumplimiento actual del requisito dentro del desarrollo, puede ser “Cumple”, “No Cumple” o “Ambiguo”. El estado predeterminado al momento de la creación es “No Cumple”. El formulario ofrece una lista seleccionable para este atributo.
- **Costo:** Número entero positivo que corresponde, por lo general, a los puntos de función asociados al requisito. Contabilizando funciones

básicas, como inputs, outputs, consultas e interacciones del usuario, para estimar el costo del desarrollo.

Además debe estar asociado con un Hito del proyecto (mediante una lista de selección), y puede estar asociado a varios Tipos de Usuario (mediante una lista de selección múltiple).

\*Nombre:

\*Descripción:

\*Hito:

Usuarios asociados:  (presionar "Ctrl" para seleccionar)

\*Prioridad:

\*Estabilidad:

\*Tipo:

\*Fuente:

\*Estado:

Costo:

Figura 5.17: Formulario de creación de Requisitos de Usuario.

La sección de ayuda muestra cómo se especifican los Requisitos de Usuario de un proyecto y muestra la definición de los tipos de requisitos de usuario presentada anteriormente. Las alarmas automáticas programadas para los Requisitos de Usuario son las siguientes:

- "El Hito asignado a este Requisito de Usuario ha sido eliminado"
- "No tiene ningún Requisito de Software asociado"
- "Tiene más de 5 Requisitos de Software asociados (puede ser muy complejo y quizás deba ser dividido)"
- "Su estabilidad es intransable, sin embargo todos sus Requisitos de Software asociados son transables"
- "Su prioridad es menor que la prioridad más baja de sus Requisitos de Software asociados"
- "Su estado es No Cumple, pero todos sus Casos de Prueba asociados Cumplen"
- "Su estado es No Cumple, pero todos sus Requisitos de Software asociados Cumplen"



- "Su estado es Cumple, pero ninguno de sus Casos de Prueba asociados Cumplen"
- "Su estado es Cumple, pero ninguno de sus Requisitos de Software asociados Cumplen"
- "Uno de sus Tipos de Usuario no se encuentra asociado a ninguno de sus Requisitos de Software"

identificador	nombre	estado	costo	prioridad	tipo	hito
RU0001	Usuarios Sistema	Cumple	0	urgente	funcional	Primera iteración
RU0003	Login U-Pasaporte	Cumple	0	urgente	funcional	Primera iteración

Colores: **cumple** **no cumple** **ambiguo**

Una persona que puede identificarse correctamente con U-Pasaporte puede ingresar al sistema y obtener la interfaz correspondiente a su tipo de usuario, sea Alumno, Administrativo o potencialmente Profesor.

Fuente: equipo  
Estabilidad: intransable

Usuarios asociados(3):

- TU0002 Administrativo
- TU0003 Alumno
- TU0004 Profesor

Figura 5.18: Vista comprimida y expandida de Requisitos de Usuario, con atributos principales en barra de ordenamiento.

#### 5.4.4. REQUISITOS DE SOFTWARE

En la etapa de "Definición de requisitos de software" el analista revisa los requisitos de usuario especificados anteriormente, asistido por conocimientos más técnicos del implementador, para establecer formalmente qué necesita el software a desarrollar para satisfacerlos. Los requisitos de software son ingresados en la sección "Proyecto/Requisitos de Software", y los atributos son prácticamente los mismos que los de requisitos de usuario, salvo algunas diferencias:

- El formato del identificador asignado automáticamente es "RSXXXX".
- Se asocia con los requisitos de usuario correspondientes mediante una lista de selección múltiple.
- Los requisitos de software se pueden clasificar en tipos más específicos:
  - **Funcionales:** Indican cuáles deben ser las capacidades del software. Se derivan del modelo lógico.
  - **Interacción:** Especifican el hardware, software o elementos de bases de datos con los que el sistema o sus componentes interactúan o se comunican.

- **Operacionales:** Especifican la forma en que correrá el sistema y cómo se comunicará con los operadores humanos. Incluyen todas las interfaces de usuario, interacción humano-computador, y requisitos logísticos y organizacionales.
- **Recursos (Ambiente Operacional):** Especifican los límites superiores de los recursos físicos, tales como capacidad de procesamiento, uso de memoria, espacio en disco, etc.
- **Usabilidad:** Están relacionados con el esfuerzo de uso y la evaluación del uso realizada por los usuarios.
- **Mantenibilidad:** Relacionados con el esfuerzo de hacer modificaciones, especificando cuán fácil es la reparación de fallas y adaptación del software a nuevos requisitos.
- **Portabilidad:** Relacionados con la habilidad del sistema de ser transferido de un ambiente a otro.
- **Confiabilidad:** Están relacionados con la capacidad de mantener un nivel adecuado de servicio, bajo ciertas condiciones y por cierto tiempo. Especificando los tiempos medios entre fallas aceptables.
- **Interoperabilidad:** Habilidad de interactuar con determinados sistemas.
- **Rendimiento:** Establecen valores numéricos para variables medibles relacionadas con el rendimiento del sistema.
- **Documentación:** Especifican requisitos particulares del proyecto para la documentación.
- **Escalabilidad:** Especifica la capacidad del sistema para mantener o mejorar su rendimiento medio conforme al aumento del número de usuarios.

\*Nombre: U-Pasaporte por defecto

\*Descripción: La manera de identificar a un usuario es utilizando su cuenta de U-Pasaporte, tal que el usuario realice su login, se le solicite por parte del ADI entregar sus datos de carrera y cursos aprobados al sistema, y luego se realice un match del rut del usuario con algun tipo de usuario, si es que tiene permisos especiales, o con ninguna, si es que es Alumno por defecto. Si no se autentifica correctamente, o si no otorga los datos de U-Pasaporte que el sistema solicita, el usuario no podra ingresar, volviendo a la pagina de inicio con un mensaje de error.

\*Hito: Primera iteración

RU asociados:
 

- RU0001 Usuarios Sistema (presionar "Ctrl" para seleccionar)
- RU0002 Usuario Profesor
- RU0003 Login U-Pasaporte
- RU0004 Permisos Super-Admin
- RU0005 Permisos Profesor Super-Admin
- RU0006 Catálogo de Cursos
- RU0007 Catálogo Historico
- RU0008 Catalogo de un proceso
- RU0009 Proceso de Preinscripción
- RU0010 Proceso Automático de Preinscripción

Usuarios asociados:
 

- TU0001 Super-Admin (presionar "Ctrl" para seleccionar)
- TU0002 Administrativo
- TU0003 Alumno
- TU0004 Profesor

\*Prioridad: normal

\*Estabilidad: intransable

\*Tipo: funcional

\*Fuente: analista

\*Estado: no cumple

Costo: 16

Cancelar Guardar


Figura 5.19: Formulario de creación de Requisitos de Software.

La sección de ayuda muestra cómo se especifican los Requisitos de Software de un proyecto y muestra la definición de los tipos de requisitos de software presentada anteriormente. Las alarmas automáticas programadas para los Requisitos de Software son las siguientes:

- "El Hito asignado a este Requisito de Software ha sido eliminado"
- "No tiene ningún Requisito de Usuario asociado"
- "No tiene ningún Módulo asociado"
- "Tiene más de 3 Requisitos de Usuario asociados (puede ser muy complejo y quizás deba ser dividido)"
- "Su estabilidad es intransable, sin embargo todos sus Requisitos de Usuario son transables"
- "Su prioridad es mayor que la prioridad más alta de sus Requisitos de Usuario asociados"

- “Su estado es No Cumple, pero todos sus Casos de Prueba asociados Cumplen”
- “Su estado es Cumple, pero ninguno de sus Casos de Prueba asociados Cumplen”
- “Uno de sus Tipos de Usuario no se encuentra asociado a ninguno de sus Requisitos de Usuario”
- “Su Hito no corresponde con ningún Hito asociado a sus Requisitos de Usuario”
- “Un Requisito de Usuario asociado ha sido modificado (su fecha de modificación es posterior a la de este Requisito de Software, favor revisar y actualizar)”

Colores: **cumple** **no cumple** **ambiguo**

identificador	nombre	estado	costo	prioridad	tipo	hito
RS0001	Inicio General del Sistema		0	urgente	funcional	Primera iteración
RS0002	U-Pasaporte por defecto		0	normal	funcional	Primera iteración 

Estado: **cumple**

La manera de identificar a un usuario es utilizando su cuenta de U-Pasaporte, tal que el usuario realice su login, se le solicite por parte del ADI entregar sus datos de carrera y cursos aprobados al sistema, y luego se realice un match del rut del usuario con algun tipo de usuario, si es que tiene permisos especiales, o con ninguna, si es que es Alumno por defecto. Si no se autentifica correctamente, o si no otorga los datos de U-Pasaporte que el sistema solicita, el usuario no podra ingresar, volviendo a la pagina de inicio con un mensaje de error.

Fuente: **analista**  
Estabilidad: **intransable**

Usuarios asociados(3):

- o TU0002 Administrativo
- o TU0003 Alumno
- o TU0004 Profesor

RU asociados(2):

- o **RU0001 Usuarios Sistema**
- o **RU0003 Login U-Pasaporte**

Figura 5.20: Vista comprimida y expandida de Requisitos de Software, con atributos principales en barra de ordenamiento.

#### 5.4.5. CASOS DE PRUEBA

En la etapa de “Definición de casos de pruebas” el téster debe ingresar los casos de prueba que se usarán para detectar errores en el sistema desarrollado y como medio para validar que el producto de software ha sido concebido e interpretado correctamente según lo especificado en los requisitos. Se debe acceder a la sección “Proyecto/Casos de Prueba” para crear un caso de prueba a partir de los siguientes atributos:

- **Identificador:** Código único que sirve para identificar y referenciar formalmente este caso de prueba dentro del proyecto. Es asignado automáticamente de manera incremental utilizando el formato “CPXXXX”.

- **Nombre:** Nombre simple del caso de prueba. El formulario ofrece un campo de texto para ingresar este atributo.
- **Descripción:** Acá se describe el procedimiento para evaluar el caso de prueba. El formulario ofrece un área de texto para ingresar este atributo.
- **Resultado Aceptable:** Se describe el mínimo resultado aceptable que hace exitosa a esta prueba.
- **Resultado Óptimo:** Se describe el mejor resultado esperable tras realizar la prueba.
- **Estado:** El estado de cumplimiento actual del caso de prueba, puede ser "Cumple", "No Cumple" o "Ambiguo". El estado predeterminado al momento de la creación es "No Cumple". El formulario ofrece una lista seleccionable para este atributo.

Un caso de prueba debe estar asociado a un Requisito de Usuario o bien a un Requisito de Software seleccionable desde una lista, y puede estar asociado a varios Tipos de Usuario (mediante una lista de selección múltiple), indicando las personas que deberían realizar esta prueba.

Figura 5.21: Formulario de creación de Casos de Prueba.

La sección de ayuda muestra cómo se definen los Casos de Prueba de un proyecto. Las alarmas automáticas programadas para los Casos de Prueba son las siguientes:

- "El Requisito asignado a este Casos de Prueba ha sido eliminado"

- “El Requisito asociado ha sido modificado (su fecha de modificación es posterior a la de este Caso de Prueba, favor revisar y actualizar)”

Colores: **cumple** **no cumple** **ambiguo**

identificador	nombre	estado	requisito
CP0001	Quitar permiso de administrativo		RS0003 Existencia Super-Admin
CP0002	Cambiar correo de contacto en login	no cumple	RU0003 Login U-Pasaporte

Estado: no cumple

El Super-Admin ingresa a la interfaz que le ofrece cambiar el mail de contacto del login, y cambia el correo de contacto por otro. Luego se ingresa a la pagina de login y se observa que correo se ofrece como contacto

Resultado aceptable: el correo de contacto es el mismo que se ingreso  
 Resultado óptimo: el correo de contacto es el mismo que se ingreso

Usuarios asociados( 1 ):  
 o TU0001 Super-Admin

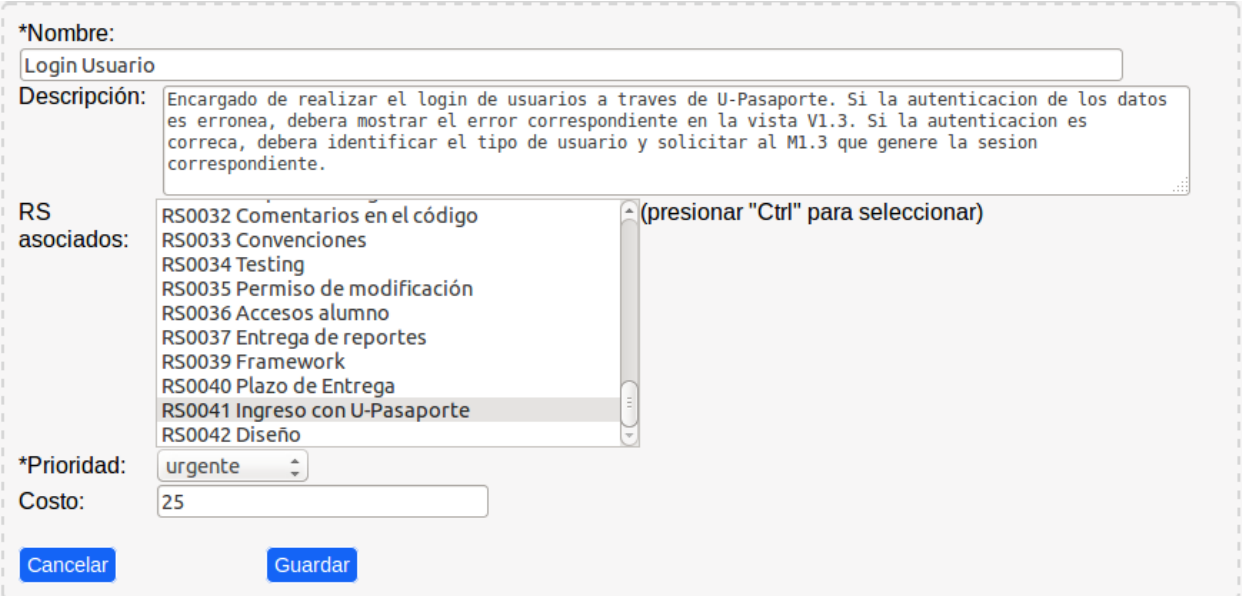
Figura 5.22: Vista comprimida y expandida de Casos de Prueba, con atributos principales en barra de ordenamiento.

#### 5.4.6. MÓDULOS

Para la etapa de “Diseño arquitectónico y diseño detallado” el diseñador debe ingresar los módulos del sistema a desarrollar, los que corresponden a componentes de alto nivel que encapsulan los requisitos de software descritos en etapas anteriores. En la práctica éstos se pueden definir como interfaces, páginas web o procedimientos. Se debe acceder a la sección “Proyecto/Módulos” para ingresar un módulo del sistema a partir de los siguientes atributos:

- **Identificador:** Código único que sirve para identificar y referenciar formalmente este módulo dentro del proyecto. Es asignado automáticamente de manera incremental utilizando el formato “MDXXXX”.
- **Nombre:** Nombre simple del módulo. El formulario ofrece un campo de texto para ingresar este atributo.
- **Descripción:** Se describe la función que encapsula este módulo. El formulario ofrece un área de texto para ingresar este atributo.
- **Prioridad:** Nivel de urgencia asociada al módulo. La prioridad de un módulo puede ser “urgente”, “pronto”, “normal” o “cuando haya tiempo”. El formulario ofrece una lista seleccionable.
- **Costo:** Número entero positivo que corresponde, por lo general, a los puntos de función asociados al módulo. Contabilizando funciones básicas, como inputs, outputs, consultas e interacciones del usuario, para estimar el costo del desarrollo.

Se pueden agregar los requisitos de software asociados al módulo mediante una lista de selección múltiple.



\*Nombre: Login Usuario

Descripción: Encargado de realizar el login de usuarios a través de U-Pasaporte. Si la autenticación de los datos es errónea, deberá mostrar el error correspondiente en la vista V1.3. Si la autenticación es correcta, deberá identificar el tipo de usuario y solicitar al M1.3 que genere la sesión correspondiente.

RS asociados: (presionar "Ctrl" para seleccionar)

- RS0032 Comentarios en el código
- RS0033 Convenciones
- RS0034 Testing
- RS0035 Permiso de modificación
- RS0036 Accesos alumno
- RS0037 Entrega de reportes
- RS0039 Framework
- RS0040 Plazo de Entrega
- RS0041 Ingreso con U-Pasaporte
- RS0042 Diseño

\*Prioridad: urgente

Costo: 25

Cancelar Guardar

Figura 5.23: Formulario de creación de Módulos.

La sección de ayuda muestra cómo se definen los Módulos de un proyecto. Las alarmas automáticas programadas para los Módulos son las siguientes:

- “No tiene ningún Requisito de Software asociado”
- “Ningún Requisito de Software asociado tiene por lo menos la misma prioridad que este módulo”
- “Un Requisito de Software asociado ha sido modificado (su fecha de modificación es posterior a la de este Módulo, favor revisar y actualizar)”

Colores: **cumple** **no cumple** **ambiguo**

identificador	nombre	costo	prioridad
MD0017	Actualización Datos Contacto	0	normal
MD0016	Administración de Permisos	0	urgente

Debera retornar una tabla con los usuarios Administrativos y otra con los usuarios Profesores. Adicionalmente debe permitir borrar o agregar usuarios a estas tablas. Se debera ofrecer un label y dos botones; en el label se ingresa el RUT del usuario, un boton para agregar y el otro para borrar. Si el RUT es invalido mostrar error. Si se agrega un usuario ya agregado, notificar el hecho sin realizar cambios. Si se elimina un usuario que no existia, notificar el hecho sin realizar cambios. Un usuario puede pertenecer a una o a ninguna de estas tablas, pero nunca puede pertenecer a ambas, en caso de conflicto, notificar el hecho sin realizar cambios. Finalmente, todo cambio realizado en esta tabla debe ser persistida.

RS asociados( 1 ):  
 ◦ RS0004 Dar permiso Administrativo

Figura 5.24: Vista comprimida y expandida de Módulos, con atributos principales en barra de ordenamiento.

## 5.5. DOCUMENTOS

La sección de “Documentos” sirve para editar y descargar los documentos resultantes de las etapas de “Definición de Requisitos”, “Definición de Casos de Pruebas”, “Diseño Arquitectónico y Diseño Detallado” y “Documentación Final del Proyecto” (ver Figura 5.1).

Sección: **Introducción** PDF ?

2014-06-11 20:06 [alejandro]

---

En la actualidad, el Departamento de Ciencias de la Computacion (DCC) cuenta con un problema concreto en la oferta de ramos de cada semestre, puesto que no existe ninguna manera de preveer las demandas efectivas de los cursos ofrecidos, obligando asi a eliminar cursos una vez comenzado cada semestre academico por poca demanda -perjudicando a aquellos que escogieron tomar estos cursos-, a abrir secciones nuevas en cursos con demandas inesperadamente altas -generando problemas de horarios y debiendo llegar a acuerdos en tiempo limitado con profesores que puedan dictarlos-, y restringen las ofertas de electivos de los profesores por desconocimiento del quorum que obtendran en la inscripcion, lo que genera que no exista una rotacion grande de los cursos que se dictan, y acota cada vez mas los electivos.

El Director del departamento, Nelson Baloian, acude con la inquietud de realizar un sistema de Preinscripcion Academica, basado en el prototipo desarrollado anteriormente para el CADCC 2012, y que permita a los alumnos simular una inscripcion academica con el fin de obtener retroalimentacion tempranamente de las tendencias de los cursos, pudiendo enviar una mejor oferta de cursos a la Escuela para el periodo de inscripcion oficial, beneficiando asi tanto al departamento como a los alumnos y profesores, que pueden tener una comunicacion relativamente confiable antes de que comience el semestre.

Figura 5.25: Sección de introducción del Documento de Requisitos.

Un usuario puede seleccionar la sección del documento que desea ver o modificar (ver Figura 5.26). En el caso que disponga de permisos de edición se le ofrece un editor de texto enriquecido, con capacidad de carga y edición de imágenes, junto a un historial de modificaciones que permite recuperar versiones pasadas de la sección (ver Figura 5.27).



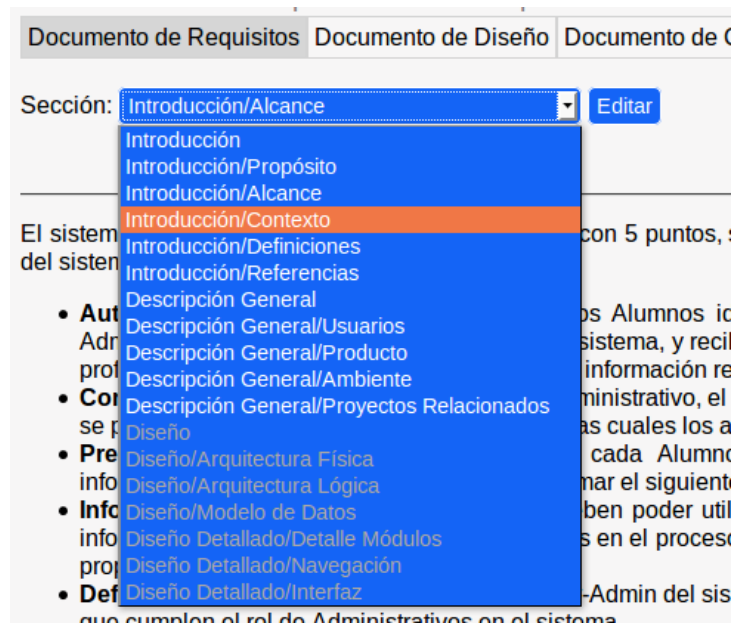


Figura 5.26: Selección de secciones del Documento de Requisitos.

Se puede descargar el resultado con el botón "PDF". El informe generado incluye una portada, un índice de contenido, las secciones editadas, las listas de elementos de proyecto respectivas (ej.: listado de requisitos) y las matrices de trazado correspondientes. El botón de ayuda muestra la definición de las secciones y la estructura de cada documento.

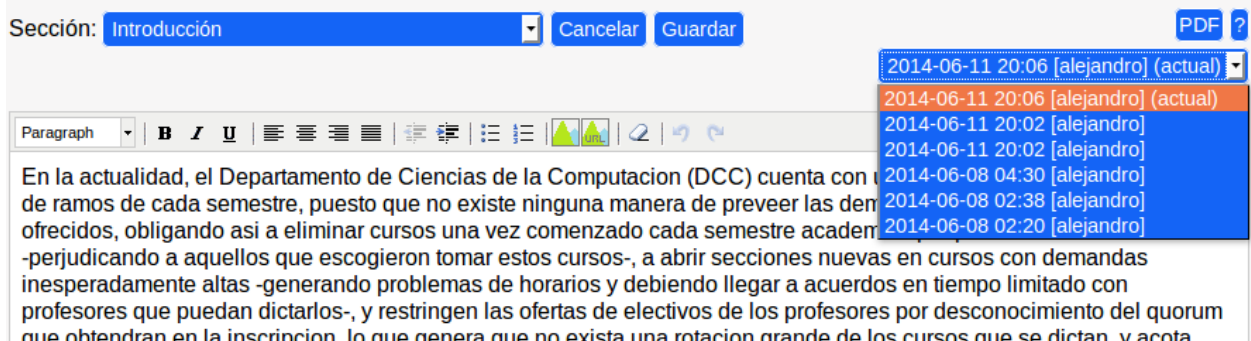


Figura 5.27: Selección de versiones anteriores de la sección del documento.

## 5.6. HERRAMIENTAS

Esta sección de MainReq tiene como propósito asistir a los integrantes del equipo de desarrollo a lo largo del proyecto ofreciendo indicadores que permitan identificar riesgos y resolver inconsistencias de manera temprana. La sección de Herramientas se compone de cinco módulos: Estadísticas, Matrices de Trazado, Consistencia de Relaciones, Bitácora de Cambios y Tareas.

### 5.6.1. ESTADÍSTICAS

En esta sección se puede revisar el avance del proyecto en términos de cumplimiento y cantidad de elementos abarcados por Hito.

<b>Requisitos de Usuario</b>									
<b>RU</b>		<b>cumple</b>	<b>no cumple</b>	<b>ambiguo</b>	<b>total</b>				
<b>Prioridad</b>	urgente	16	1	5	22	78%			
	normal	5	1	0	6	21%			
	pronto	0	0	0	0	0%			
	cuando haya tiempo	0	0	0	0	0%			
<b>Estabilidad</b>	intransable	17	1	5	23	82%			
	transable	4	1	0	5	17%			
<b>Tipo</b>	funcional	18	2	2	22	78%			
	calidad	1	0	2	3	10%			
	restricción	2	0	1	3	10%			
		21	75%	2	7%	5	17%	<b>28</b>	100%
<b>Sin RS asoci.</b>	0								
<b>Sin TU asoci.</b>	5								
<b>Costo total</b>	35								

Figura 5.28: Estadísticas de Requisitos de Usuario para la Primera Iteración.

Se ofrece un selector para revisar las estadísticas de elementos asociados a un Hito particular del proyecto. Las tablas pueden ser descargadas en un documento PDF con portada, índice y separadas por hito para su posterior revisión. Con el botón de ayuda el usuario puede revisar la definición de las tablas estadísticas.

<b>Casos de Prueba, Primera iteración</b>								
<b>CP</b>	<b>cumple</b>	<b>no cumple</b>	<b>ambiguo</b>	<b>total</b>				
<b>asociados a RU</b>	1	3	0	4	6%			
<b>asociados a RS</b>	33	16	5	54	93%			
	34	58%	19	32%	5	8%	<b>58</b>	100%
<b>Sin TU asoci.</b>	3							

Figura 5.29: Estadísticas de Casos de Prueba para la Primera Iteración.

Las estadísticas son desplegadas en cuatro tablas:

- **Requisitos de Usuario:** Se despliega la cantidad separada por atributos (prioridad, estabilidad y tipo) y por el nivel de cumplimiento (cumple, no cumple y ambiguo). Contabilizando el costo total y cuántos de ellos no tienen Requisito de Software o Tipo de Usuario asociados (ver Figura 5.28).
- **Requisitos de Software:** Muestra la misma información que la tabla anterior, pero considerando los tipos específicos de los Requisitos de Software.
- **Casos de Prueba:** Se contabiliza la cantidad de Casos de Prueba asociados a Requisitos de Usuario y los asociados a Requisitos de Software, separados por estado de cumplimiento. Además se cuentan los que no tienen Tipos de Usuarios asociados (ver Figura 5.29).
- **Módulos:** Se separan por prioridad, sumando el costo total y contabilizando cuántos módulos no tienen Requisitos de Software asociados (ver Figura 5.30).

<b>Módulos</b>			
		<b>MD</b>	<b>total</b>
<b>prioridad</b>		urgente	17 100%
		normal	0 0%
		pronto	0 0%
		cuando haya tiempo	0 0%
			<b>17 100%</b>
<b>Sin RS asoci.</b>	2		
<b>Costo total</b>	0		

Figura 5.30: Estadísticas de Módulos.

### 5.6.2. MATRICES DE TRAZADO

Las matrices de trazado sirven para revisar las relaciones establecidas entre dos tipos de elementos. Ubicando un tipo de elemento en las filas y el otro tipo en las columnas se puede evaluar la cobertura de requisitos, casos de prueba y módulos. De esta manera se puede realizar un análisis de impacto ante alguna modificación de un elemento en particular.

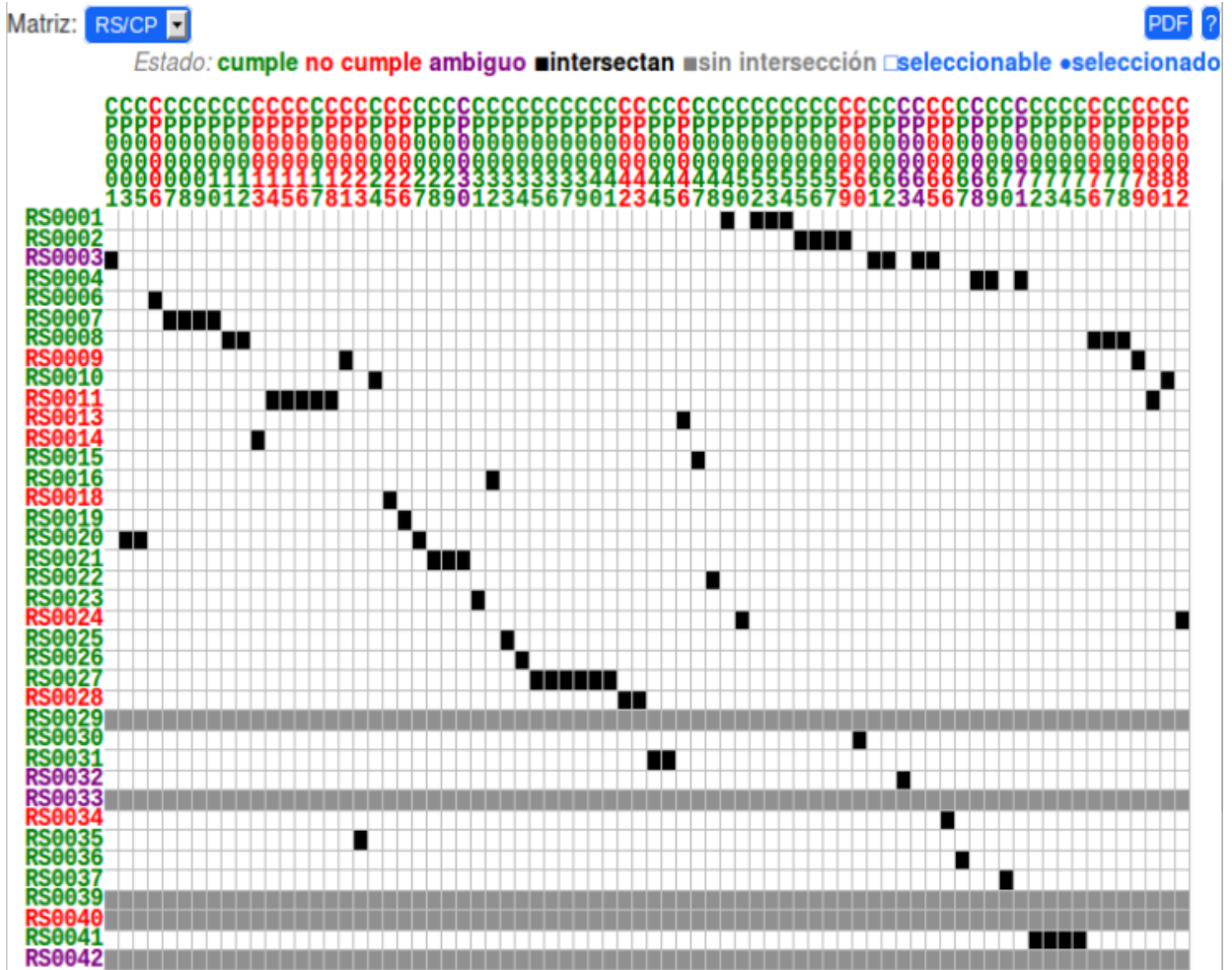


Figura 5.31: Matriz de Trazado muestra cinco Requisitos de Software sin Casos de Prueba asociados.

La matriz interactiva (ver Figura 5.31) permite revisar con un clic el contenido del par de elementos en una intersección. Se destacan con color gris los elementos que no intersecan. Se puede descargar un documento todas las matrices en un archivo PDF, las cuales son particionadas en cuadrantes si sobrepasan el tamaño de la hoja (ver Figura 5.32).

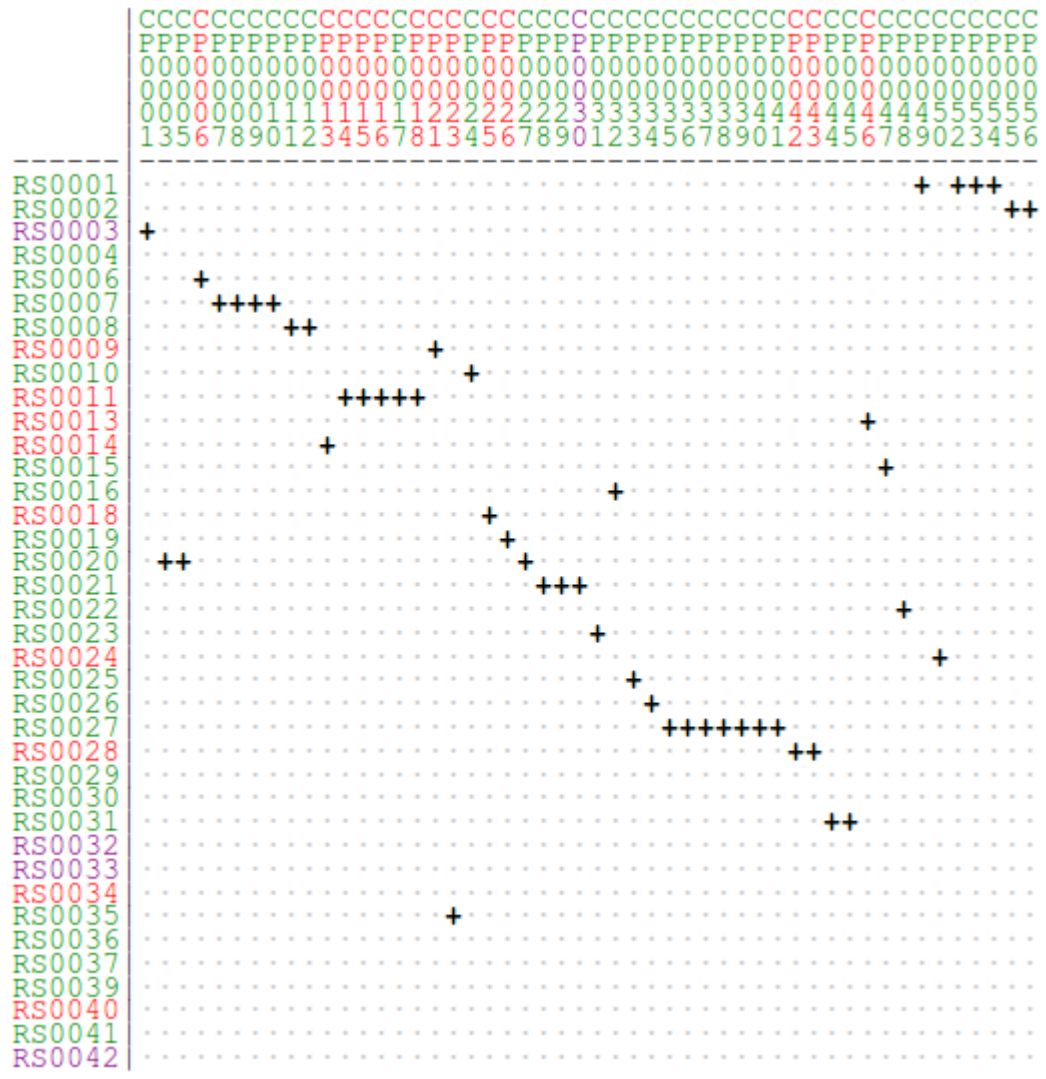


Figura 5.32: Matriz de Trazado de un documento PDF (Requisitos de Software vs. Casos de Prueba).

Las matrices disponibles son las siguientes:

- Requisitos de Usuario vs. Requisitos de Software
- Módulos vs. Requisitos de Software
- Requisitos de Usuario vs. Casos de Prueba
- Requisitos de Software vs. Casos de Prueba

5.6.3. CONSISTENCIA DE RELACIONES

Mientras las matrices de trazado permiten rápidamente observar dónde se encuentran las relaciones entre dos tipos de elementos, la sección de Consistencia de Relaciones muestra el contenido completo de los elementos relacionados, complementando la información de las matrices.

El usuario escoge dos tipos de elementos, y la herramienta enlista los del segundo tipo bajo los del primer tipo (ver Figura 5.33). También puede seleccionar un elemento del primer tipo desde una lista. Se utiliza el formato de elementos de contenido expandible y la barra de ordenamiento usados en el ingreso de datos del proyecto. Se puede descargar una versión PDF como un documento para su revisión “offline”.

Consistencia RU/RS PDF ?

(+)Abrir Todos (-)Cerrar Todos

Colores: cumple no cumple ambiguo

identificador	nombre	estado	costo	prioridad	tipo	hito	
<b>RU0001</b>	<b>Usuarios Sistema</b>		0	urgente	funcional	Primera iteración	!
	<b>RS0002 U-Pasaporte por defecto</b>		0	urgente	funcional	Primera iteración	
	<b>RS0003 Existencia Super-Admin</b>		0	urgente	funcional	Primera iteración	
	<b>RS0004 Dar permiso Administrativo</b>		0	urgente	funcional	Primera iteración	
<b>RU0002</b>	<b>Usuario Profesor</b>		0	normal	funcional	Segunda iteración	
	<b>RS0004 Dar permiso Administrativo</b>		0	urgente	funcional	Primera iteración	
<b>RU0003</b>	<b>Login U-Pasaporte</b>		0	urgente	funcional	Primera iteración	!
	<b>RS0001 Inicio General del Sistema</b>		0	urgente	funcional	Primera iteración	
	<b>RS0002 U-Pasaporte por defecto</b>		0	urgente	funcional	Primera iteración	!
	<b>RS0019 Inicio Profesor</b>		0	normal	funcional	Segunda iteración	!
	<b>RS0020 Inicio Alumno</b>		0	urgente	funcional	Primera iteración	

Figura 5.33: Vista de Consistencia de Relaciones para Requisitos de Usuario y Requisitos de Software.

Esto permite comparar lado a lado el contenido de los elementos facilitando la tarea de los usuarios al momento de comprobar la consistencia de las relaciones. Los tipos de relaciones que se pueden revisar son:

- Requisitos de Usuario vs. Requisitos de Software
- Requisitos de Usuario vs. Casos de Prueba
- Requisitos de Software vs. Requisitos de Usuario
- Requisitos de Software vs. Casos de Prueba
- Requisitos de Software vs. Módulos
- Módulos vs. Requisitos de Software
- Casos de Prueba vs. Requisitos de Usuario
- Casos de Prueba vs. Requisitos de Software

#### 5.6.4. BITÁCORA

A modo de evitar la pérdida de datos se incorpora una bitácora que registra la evolución de cada elemento de proyecto creado, junto con la fecha y el usuario responsable de la modificación. Indica si el registro es creado, modificado, borrado o corresponde al registro actualmente visible. El usuario puede seleccionar el tipo y el elemento específico que desea revisar (ver Figura 5.34). Los registros son mostrados con el formato de elementos de contenido expandible presente en el resto de la herramienta y son ordenados cronológicamente.



The screenshot shows the 'Herramientas / Bitácora' interface. At the top, there are two dropdown menus: 'Bitácora: Requisitos de Usuario' and 'Seleccione Elemento Especifico'. Below these are two buttons: '(+)Abrir Todos' and '(--)Cerrar Todos'. The main table has a header row for 'Ordenamiento' with options: 'fecha', 'Estado: cumple', 'no cumple', 'ambiguo', 'actual', 'creado', 'editado', and 'eliminado'. The table contains five rows of requirement data:

	Ordenamiento	fecha	Estado: cumple	no cumple	ambiguo	actual	creado	editado	eliminado
<b>RU0020 Catálogo de Cursos</b>	0	urgente	funcional			Primera iteración			
						<b>Elemento actual</b>	2014-08-13 00:29	[tom]	
<b>RU0032 Permisos especiales</b>	0	normal	restricción			Tercera iteración			
						<b>Elemento actual</b>	2014-08-12 23:56	[tom]	
<b>RU0032 loloes</b>	0	normal	restricción			Primera iteración			
						<b>Elemento creado</b>	2014-07-31 19:07	[alex]	
<b>RU0031 Tormenta solar</b>	0	normal	calidad			Primera iteración			
						<b>Elemento eliminado</b>	2014-07-31 18:59	[alex]	
<b>RU0031 Tormenta solar</b>	0	normal	calidad			Primera iteración			

Figura 5.34: Bitácora muestra los cambios realizados en los requisitos.

#### 5.6.5. TAREAS

Esta sección incorpora a la herramienta un modo de monitorear y organizar el trabajo del equipo de desarrollo. Las tareas se presentan como elementos de contenido expandible y se pueden ordenar por estado (ver Figura 5.35). El jefe de proyecto tiene la responsabilidad de asignar tareas y aprobar o reprobar el trabajo realizado. En caso de que la tarea presente inconsistencias en su especificación puede ser descartada. Cada vez que se crea una tarea se da la opción de enviar un correo electrónico al integrante con copia al jefe de proyecto. Para crear una tarea el jefe de proyecto debe especificar:

- Integrante designado
- Nombre de Tarea
- Descripción/Instrucciones
- Fecha de Entrega

fhafon [ifracke.95@gmail.com] ?

(+)Abrir Todos (-)Cerrar Todos Nuevo

Colores: por hacer realizada no realizada aprobada reprobada descartada

estado	fecha
Actualizar estado requisitos	2014-06-20 23:59 (quedan 2 dias)
<b>Realizar los primeros Casos de Prueba</b>	2014-06-15 23:59
Estado: realizada	
Responsable: fhafon [ifracke.95@gmail.com]	
Fecha inicio: 2014-06-07 21:00	
Debes realizar los casos de prueba de los requisitos de software con prioridad 'urgente'.	
Aprobar	Reprobar
Descartar	

Figura 5.35: Vista del Jefe de Proyecto sobre las Tareas asignadas a un usuario.

El integrante sólo puede revisar sus tareas y tiene un plazo definido para reportar la realización de su tarea. El botón de ayuda muestra el ciclo de vida de las tareas (ver Figura 5.36), cuyos estados son presentados a continuación:

- **Por Hacer:** La Tarea ha sido creada y aún queda tiempo para realizarla.
- **Realizada:** El integrante responsable reporta la realización de la Tarea dentro del plazo.
- **No Realizada:** Se acaba el plazo y el integrante no ha reportado su realización.
- **Aprobada:** El jefe de proyecto considera válida la Tarea reportada por el integrante.
- **Reprobada:** El jefe de proyecto considera inválida la Tarea reportada por el integrante.
- **Descartada:** El jefe de proyecto decide que la especificación de la Tarea es inconsistente o ambigua.



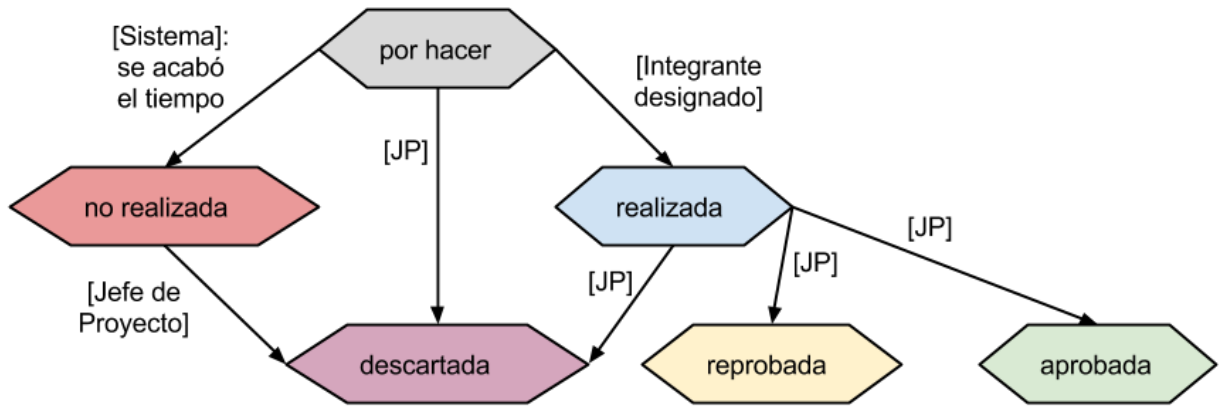


Figura 5.36: Ciclo de vida de las Tareas.

## 6. EVALUACIÓN PRELIMINAR DE LA SOLUCIÓN

La nueva solución es evaluada de manera previa a la implementación, durante la etapa de desarrollo y sobre el producto final. Para cada sesión de evaluación se escoge un usuario con conocimientos adquiridos en el curso de Ingeniería de Software II (CC5401). El usuario debe completar las pruebas diseñadas de acuerdo a la estrategia definida en la sección "3.2.2. Pruebas de Usuario". Por mientras un evaluador registra los errores de uso, es decir, cada vez que el usuario no utiliza de manera esperada alguna opción de la herramienta.

La interfaz inicial es sometida a pruebas de usuario sobre prototipos no funcionales, enfocando los esfuerzos en comprobar que la interfaz propuesta sea intuitiva y que no existan problemas de navegación o secciones ambiguas. Se realizaron 3 sesiones de este tipo, evaluando de 4 a 6 pruebas o tareas por cada una. Las correcciones se realizan principalmente sobre el nombre y la distribución de opciones de navegación. Dejando, por ejemplo, todas las opciones de administración en un módulo separado. Agrupando opciones con propósitos similares, como las herramientas, y normalizando la posición y nombres de botones de acción para que sean consistentes a través de todos los formularios.

Las pruebas de usuario realizadas durante el desarrollo, están enfocadas a depurar las características funcionales de la nueva herramienta, permitiendo detectar y corregir problemas a tiempo, dentro del flujo de acciones del usuario. Los usuarios escogidos para estas pruebas corresponden a ex alumnos del curso de Ingeniería de Software II, por lo que se evalúa sólo la parte que permite la administración de un único proyecto (ingreso de datos del proyecto y revisión de documentos e indicadores). Se realizaron 7 sesiones de este tipo, evaluando de 8 a 11 pruebas de usuario por cada una. Cabe destacar que para cada sesión de pruebas se usa una versión incremental del software, incluyendo mejoras identificadas en la sesión previa. La mayoría de las modificaciones son realizadas durante esta etapa del desarrollo. Hacia el final de esta etapa de evaluación se puede observar que la cantidad de errores de uso disminuye de 20 a 5 errores promedio por sesión (aproximadamente), y los usuarios logran cumplir exitosamente casi el 90% de las tareas esperadas por sesión. Al final de esta etapa las principales necesidades de los usuarios alumnos son satisfechas y sólo se requieren mínimas modificaciones adicionales para un mejor desempeño.

Luego del refinamiento realizado durante la etapa de desarrollo, se evalúa el producto final considerando dos tipos de usuarios: alumnos con poca experiencia y docentes con experiencia en este tipo de herramienta. Estos usuarios exploran la herramienta, guiados por tareas con un propósito definido. Las sesiones de pruebas abordadas por usuarios alumnos son similares a las realizadas durante la etapa de desarrollo. Las tareas desarrolladas por usuarios docentes, están enfocadas a la administración de varios proyectos, y por lo tanto evalúan el flujo completo de actividades, desde la creación de proyectos y asignación de equipos de trabajo, hasta la revisión de documentos e indicadores. La mayoría de las modificaciones sugeridas en esta etapa corresponden a ajustes para facilitar la administración de varios proyectos a través de los semestres. Por ejemplo, añadir un filtro para obtener los proyectos realizados en un semestre determinado. En esta oportunidad se realizan 3 sesiones adicionales con usuarios alumnos y 2 sesiones con usuarios docentes, incorporando las últimas modificaciones requeridas al producto final. Luego de estas pruebas se estima que la nueva herramienta podría ser usada de manera experimental como apoyo al curso de Ingeniería de Software II.

De manera similar al proceso de diagnóstico realizado sobre la solución anterior, se crea una encuesta para evaluar la usabilidad y el nivel de satisfacción brindado por la nueva herramienta. Evaluando los siguientes aspectos:

- Navegación fácil e intuitiva.
- Fácil búsqueda de información.
- Velocidad de carga.
- Uso aceptable de colores.
- Organización adecuada de información y secciones.
- Adecuado manejo y recuperación de errores.
- Muestra contenido relevante.
- Funcionalmente completo.

Estas encuestas son realizadas durante la etapa de desarrollo y en la etapa final de evaluación, luego de cada sesión de pruebas. El usuario califica el desempeño de cada uno de los aspectos mencionados anteriormente, en 5 niveles de satisfacción, desde "Muy de acuerdo" a "Muy en desacuerdo". Al comienzo del desarrollo, aspectos importantes, como "Fácil búsqueda de información" y "Funcionalmente completo", presentan problemas siendo

calificados con "En desacuerdo" y "Muy en desacuerdo". Pero ya en iteraciones finales, la mayoría de los aspectos son corregidos y evaluados como "De acuerdo" o "Muy de acuerdo". Salvo la "Velocidad de carga" y un "Adecuado manejo y recuperación de errores", los que aún tienen bajas calificaciones. Este resultado se atribuye a que efectivamente la velocidad del computador portátil utilizado en las pruebas no permite una navegación fluida. Lo que a su vez causa frustración en los usuarios, quienes realizan acciones repetitivas sobre la interfaz esperando obtener respuestas inmediatas. Estos comentarios son incluidos en la encuesta, y permiten determinar de mejor manera las modificaciones necesarias para un mejor desempeño en una próxima iteración de la solución. Adicionalmente se le pide al usuario que evalúe con una nota de 1 a 7 la nueva herramienta, obteniendo un 4,9 y un 6,0 como nota promedio en la etapa de desarrollo y la etapa final respectivamente. Lo que denota un progreso en la aceptación de la solución.

## 7. CONCLUSIONES Y TRABAJO A FUTURO

Este trabajo logra cumplir con el objetivo principal de desarrollar una nueva versión de ReqAdmin. La nueva herramienta cumple con los principales requisitos propuestos en un comienzo (ver sección 4.1). La utilidad y usabilidad de esta nueva versión es verificada usando pruebas de usuarios y encuestas, las cuales sirven también durante el desarrollo, para detectar y resolver problemas a tiempo, rediseñando e incorporando mejores características al producto final.

El proceso de diagnóstico de la herramienta anterior revela múltiples problemas funcionales y de usabilidad en ReqAdmin. Su implementación actual presenta comportamientos inesperados, como la pérdida de información ingresada a formularios o la des-configuración de permisos de edición. Una navegación poco intuitiva y una edición incómoda de los documentos generados, también representan un obstáculo para alumnos y docentes que utilizan la herramienta.

MainReq ofrece gran parte de las características presentes en la herramienta anterior e incorpora modificaciones y nuevas funcionalidades necesarias para mejorar el desempeño y la facilidad de uso. Permite el ingreso, clasificación, validación y revisión de requisitos de un proyecto de software, facilitando su gestión. Muestra indicadores estadísticos de cumplimiento con los cuales se pueden identificar riesgos y conflictos de manera temprana, anticipando situaciones no deseadas con respecto a la administración de requisitos. Dispone de visualizaciones que permiten revisar la trazabilidad y consistencia de los requisitos. Incorpora un sistema de alarmas que advierten automáticamente al usuario sobre algún problema en la especificación de algún elemento del proyecto. Ofrece la capacidad de editar y descargar documentos en formato PDF. Evita la pérdida de datos mediante el uso de una bitácora que almacena los cambios realizados sobre el proyecto. Además incluye un módulo que permite asignar y revisar las tareas realizadas por los integrantes del equipo de desarrollo responsable del proyecto de software.

Las múltiples pruebas de usuario, realizadas durante el diseño y desarrollo de la solución, permitieron forjar una herramienta estable, que actualmente no requiere de mayores cambios para satisfacer las necesidades de los usuarios finales. De tal manera que podría ser usada de manera experimental en un semestre normal con proyectos reales, en el curso de Ingeniería de Software II. Una creciente cantidad de pruebas exitosas y un reducido número de errores por sesión de evaluación, dan cuenta de una

mayor facilidad de uso. Estos resultados se pueden atribuir a una interfaz cada vez más ordenada, intuitiva y usable.

Las encuestas realizadas tras las pruebas finales mostraron un mayor nivel de satisfacción en comparación a ReqAdmin y a las primeras versiones de la nueva herramienta. Sin embargo, comentarios presentes en las encuestas revelaron que aún quedan aspectos por mejorar y probar de manera más rigurosa en un ambiente de producción. Aspectos como la eficiencia (velocidad de carga y uso de recursos), la robustez (comportamiento razonable ante eventos inesperados) y la confiabilidad percibida por los alumnos. Estos últimos aspectos deberán ser evaluados luego de que la herramienta se encuentre implantada en un ambiente de producción bajo un régimen de uso normal.

Algunos desafíos futuros que quedan pendientes a partir de este trabajo son los siguientes:

- Adaptar la interfaz para soportar dispositivos móviles.
- Incorporar a las estadísticas información de avance de las tareas realizadas por los integrantes del equipo de desarrollo.
- Añadir gráficos que faciliten la lectura de tablas estadísticas.
- Agregar una opción para ocultar las alarmas ya revisadas.
- Optimizar el sistema de descarga de documentos PDF, generando archivos nuevos sólo si el contenido ha sido modificado.

## 8. BIBLIOGRAFÍA Y REFERENCIAS

- [1] Vergara Astudillo, Andrés (2007). *Generación automática de métricas en proyectos de software, a partir de la especificación de requisitos*, Tesis de Magíster en Ciencias de la Computación, Departamento de Ciencias de la Computación, FCFM, Universidad de Chile.
- [2] Nielsen, J. (1993). *Usability Engineering*. California: Morgan Kaufmann.
- [3] Snyder, C. (2003). *The Fast and Easy Way to Design and Refine User Interfaces* (Interactive Technologies). San Francisco: Morgan Kaufmann  
URL: [www.paperprototyping.com](http://www.paperprototyping.com). Última visita: Diciembre de 2013.
- [4] Nielsen, J. (1995). *Usability Testing for the 1995 Sun Microsystems' Website*. URL: <http://www.nngroup.com/articles/usability-testing-1995-sun-microsystems-website/>. Última visita: Julio de 2014.
- [5] Friedman, V. (2007). *30 Usability Issues to Be Aware Of*. URL: <http://www.smashingmagazine.com/2007/10/09/30-usability-issues-to-be-aware-of/>. Última visita: Junio de 2014.
- [6] Shneiderman, B. (1997). 3<sup>o</sup>ed. Boston: Addison-Wesley Publishing. URL: <http://faculty.washington.edu/jtenenbg/courses/360/f04/sessions/schneidermanGoldenRules.html>. Última visita: Junio de 2014.
- [7] Nielsen, J. (1996). *Inverted Pyramids in Cyberspace*. URL: <http://www.nngroup.com/articles/inverted-pyramids-in-cyberspace/>. Última visita: Mayo de 2014.
- [8] Westfall, L. (2006). *Bidirectional Requirements Traceability*. URL: [www.westfallteam.com](http://www.westfallteam.com). Última visita: Julio de 2014.
- [9] Gould, J., Lewis, C. (1985). *Designing for Usability: Editor Key Principles and What Designers Think*. ACM volume 28 Number 3.
- [10] Garrido, A., Rossi, G., Medina, N., Grigera, J., Firmenich, S. (2013). *Improving accessibility of Web interfaces: refactoring to the rescue*. Berlin Heidelberg: Springer-Verlag.
- [11] Birk, A., Heller, G. (2014). *List of Requirements Management Tools*. URL: <http://makingofsoftware.com/resources/list-of-rm-tools>. Última visita: Julio de 2014.
- [12] Blueprint Software Systems, Inc. *Blueprint Requirements Center (v5.4)*. URL: <http://www.blueprintsys.com/>. Última visita: Julio de 2014.
- [13] Borland (Micro Focus). *CaliberRM (v11.2)*. URL: <http://www.borland.com/products/caliber/>. Última visita: Julio de 2014.

- [14] Atlassian. *Jira (6.1)*. URL: [www.atlassian.com/software/jira](http://www.atlassian.com/software/jira). Última visita: Julio de 2014.
- [15] Visure. *Visure Requirements (v4.5)*. URL: <http://www.visuresolutions.com/>. Última visita: Julio de 2014.
- [16] Hewlett-Packard. *HP Agile Manager (2014Jan26)*. URL: <http://www.pronq.com/software/agile-manager>. Última visita: Julio de 2014.
- [17] MID GmbH. *Innovator for Business Analysts (11 R5)*. URL: <http://www.mid.de/produkte/innovator-for-business-analysts/>. Última visita: Julio de 2014.
- [18] OverTI. *Requirements Quality Analyzer*. URL: <http://www.overti.es/software/requirements-quality-analyzer.aspx>. Última visita: Julio de 2014.
- [19] Heller. G. (2013). *RM Tools – what are they anyway?*. URL: <http://makingofsoftware.com/2013/rm-tools-what-are-they-anyway>. Última visita: Julio de 2014.
- [20] Django Software Foundation. (2005). *Django documentation*. URL: <https://docs.djangoproject.com/en>. Última visita: Julio de 2014.
- [21] IBM Rational RequisitePro. URL: <http://www03.ibm.com/software/products/es/reqpro>. Última visita: Julio de 2014.
- [22] Marini, E. (2012). *20 Software gratuitos para la Gestión de Proyectos*. URL: <http://blog.masterinprojectmanagement.net/20-software-gratuitos-para-la-gestion-de-proyectos/>. Última visita: Julio de 2014.
- [23] *CASE Spec Requirements & life-cycle Management Tool*. URL: <http://www.analysttool.com/>. Última visita: Julio de 2014.