

Universidad de las Ciencias Informáticas

Facultad 5

Laboratorio de Gestión de Proyectos



Título: Ingeniería de Requisitos del módulo Administración y Gestión de Usuarios del Sistema para la Dirección Integrada de Proyectos GESPRO.

**Trabajo de Diploma para optar por el título de
Ingeniero en Ciencias Informáticas**

Autor

Silvia Eugenia Torres Arrastre.

Tutora

MSc. Iliana Pérez Pupo.

Junio 2012

DECLARACIÓN DE AUTORÍA

DECLARACIÓN DE AUTORÍA

Declaro por este medio que yo, Silvia Eugenia Torres Arrastre, con carné de identidad 88072434970, soy la autora principal del trabajo final de tesis Ingeniería de Requisitos del módulo Administración y Gestión de Usuarios del Sistema para la Dirección Integrada de Proyectos GESPRO, y autorizo a la Universidad de las Ciencias Informáticas a hacer uso de la misma en su beneficio, así como los derechos patrimoniales con carácter exclusivo.

Y para que así conste, firmo la presente declaración jurada de autoría en Ciudad de La Habana a los ____ días del mes de _____ del año _____.

Silvia E. Torres Arrastre

Autora

Iliana Pérez Pupo

Tutora

"...si no existe la organización, las ideas, después del primer momento de impulso, van perdiendo eficacia..."

Che

*A mis papás, por estar siempre conmigo,
ayudarme y confiar siempre en mí.
A mis dos hermanas, por estar ahí en todo
momento y apoyarme.*

AGRADECIMIENTOS

A mi mamá, por ser mi guía, por su confianza, por su amor de manera incondicional y por demostrarme y enseñarme que todo lo que se propone se logra, que el problema no está en tener una recaída sino en saber sobreponerse y seguir adelante, y a pesar de las distancia siempre estuviste a mi lado, transmitiéndome tu fuerza, tu espíritu y tu amor. Gracias.

A mi hermana Yusmila, por ser como una segunda madre para mí, por ayudarme en todo momento, entenderme, aconsejarme y quererme del modo tan lindo que lo hace.

A mi papá por estar ahí de modo incondicional y a pesar de no vivir conmigo, nunca te separaste de mí, gracias por apoyarme durante mis años de estudio y por ser ese tipo de persona que eres. Te quiero mucho.

A mi hermana Anisley que a pesar de su carácter, supo darme siempre buenos consejos, las fuerzas y el ánimo en todo momento y más ahora por traer al mundo esa bella bebida, Shanel la cual te ha hecho cambiar mucho. Para ella también mis agradecimientos.

A quien quiero muchísimo Manuel, por compartir conmigo en cada momento, y lo siento también parte de mi vida, por quererme, apoyarme, comprenderme, hacerme reír y brindarme su amor. Gracias.

A mi Abuela Sonia, la única que me queda, que a pesar de no estar todo el tiempo conmigo. Siempre me hizo creer que podía lograrlo, solo había que proponérselo y tener paciencia.

A Ana, Miguel y toda la familia, gracias por quererme y tratarme como un miembro más de su familia.

A mis dos amigas del pre y de siempre, Kare y Lixi, gracias a las dos por considerarme como tal.

A mis compañeras de cuarto en especial a Yaniset y a Sonia, gracias por ayudarme cuando las necesité.

A la tutora Iliana, y a la profe Marielis por su dedicación, por la revisión y orientación en esta investigación.

A todos aquellos que de una forma u otra influyeron en mi formación profesional y dieron su aporte para que alcanzara la meta soñada.

A todos gracias por existir. Los quiero mucho.

RESUMEN

La presente investigación se basa en la aplicación de la Ingeniería de Requisitos al módulo Administración y Gestión de Usuarios del Sistema para la Dirección Integrada de Proyectos GESPRO. La Ingeniería de Requisitos está compuesta por cuatro fases: Identificación, Análisis, Especificación y Validación. La correcta aplicación de estas fases, contribuyen al mejoramiento del funcionamiento y la calidad de este módulo.

La propuesta realizada tiene un importante aporte práctico, pues a partir de ella se documentaron los requisitos del Redmine, del cual GESPRO tuvo base, así como las nuevas funcionalidades identificadas a partir de las necesidades existentes en la universidad. De este trabajo se obtuvo como resultado el Documento Especificación de Requisitos de Software y la propuesta del Diseño de los Casos de Prueba, para las pruebas de liberación del módulo Administración y Gestión de Usuarios.

Se realiza un análisis de los resultados obtenidos a partir de las Listas de Chequeo con el objetivo de comprobar que el Documento de Especificación de Requisitos de Software está correctamente redactado. Se analizó la evolución de este módulo con versiones anteriores evidenciándose la evolución del mismo. Finalmente a través de encuestas se definió el grado de aceptación de este módulo dentro del sistema mostrándose su efectividad en correspondencia a las necesidades para las que fue creado.

Palabras clave: Administración, Gestión de Proyectos, GESPRO, Gestión de Usuarios, Ingeniería de Requisitos.

ABSTRACT

This research is based on the application of Requirements Engineering on the module Administration and User Management of System for Integrated Project Management GESPRO. Requirements Engineering consists of four phases: Identification, Analysis, Specification and Validation. Proper application of these phases, contribute to improved functioning and quality of this module.

The proposal made has an important practical contribution, because it allows document the requirements of Redmine, that was the base of GESPRO, as well as new features identified from the needs in the university. Of this work was obtained Document Software Requirements Specification and the proposal of design of Test Cases for testing the release of module of Administration and User Management.

An analysis of the results obtained from the checklists was made in order to verify that the document Software Requirements Specification is properly drafted. It is also analyzed the evolution of this module in comparison with previous versions evidencing its progress. Finally, through surveys defined the degree of acceptance of this module within the system which shows its effectiveness in line with the needs for which it was created.

Key words: GESPRO, Management, Project Management, Requirements Engineering, User Management.

DATOS DE CONTACTO

DATOS DE CONTACTO

Tutora: MSc. Iliana Pérez Pupo.

Edad: 27.

Ciudadanía: Cubana.

Institución: Universidad de las Ciencias Informáticas (UCI).

Título(s): Ingeniero en Ciencias Informáticas y Máster en el área de Gestión de proyectos Informáticos.

Categoría Docente: Instructora.

E-mail: iperez@uci.cu

Graduada de la UCI, con cuatro años de experiencia como docente y en el desarrollo de software.

Índice

INTRODUCCIÓN	1
CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA.....	6
Introducción.....	6
1.1 Gestión de proyectos.	6
1.2 Herramientas de Gestión de proyectos.	7
1.3 Gestión de proyectos en la UCI.....	14
1.4 Ingeniería de Requisitos.	16
1.5 Requisitos de Software.....	23
1.6 Módulo Administración.....	24
Conclusiones parciales del Capítulo.....	26
CAPÍTULO 2: PROPUESTA DE SOLUCIÓN.....	27
Introducción.....	27
2.1 Propuesta de Solución para la Ingeniería de Requisitos del módulo Administración y Gestión de Usuarios.....	27
2.2 Flujo de Actividades para la propuesta de Solución.	29
2.3 Ingeniería de Requisitos del módulo Administración y Gestión de Usuarios del sistema GESPRO.	30
2.3.1 Etapa 1: Identificación de los requisitos.....	30
2.3.2 Etapa 2: Análisis de los requisitos.	31
2.3.3 Etapa 3: Especificación de requisitos.	31
2.3.4 Etapa 4: Validación de los requisitos.....	33
Conclusiones parciales del Capítulo.....	41
CAPÍTULO 3: VALIDACIÓN DE RESULTADOS.....	42
Introducción.....	42
3.1 Requisitos de Software.....	42
3.1.1 Requisitos Funcionales.....	42
3.1.2 Requisitos no Funcionales.....	44
3.2 Comparación de GESPRO con otros sistemas.	48
3.3 Análisis del resultado de las Listas de Chequeo.....	50
3.4 Resultado de los Casos de Prueba.	51
3.5 Evolución de los módulos Administración y Gestión de Usuarios del sistema GESPRO.	52
3.6 Análisis del impacto social del sistema GESPRO.....	54
Conclusiones parciales del Capítulo.....	57
CONCLUSIONES GENERALES.....	58

RECOMENDACIONES.....	59
REFERENCIAS BIBLIOGRÁFICAS.....	60
ANEXOS.....	63

Índice de Figuras

Figura 1. Resultados de la entrevista a Calisoft en el 2009.....	15
Figura 2. Etapas de la Identificación de Requerimientos.	22
Figura 3. Propuesta de Solución.	27
Figura 4. Vista del flujo de actividades de la propuesta de solución.....	29
Figura 6. Sección Presentación del diseño de Casos de Prueba.....	39
Figura 7. Sección Casos de Prueba.	40
Figura 8. Sección Variables del Diseño de Casos de Prueba.....	40
Figura 10. Iteraciones de las Listas de Chequeo.	51
Figura 11. Gráfica de Casos de Prueba con Requisitos Funcionales	52
Figura 12. Evolución del módulo Administración en cuanto a funcionalidades y módulos.	54
Figura 13. Grado de aceptación de sub módulos del módulo Administración en la comunidad.....	56

INTRODUCCIÓN

Actualmente es necesario para los países que deseen gestionar sus proyectos, contar con herramientas que les ayuden en la planeación, organización, control y seguimiento, de hecho, estas actividades se han convertido en procesos fundamentales para lograr los mismos (Hernández, 2005).

La gestión de proyectos se ha caracterizado desde la segunda mitad del siglo IXX por la aplicación sistemática de procesos repetibles y escalables, por tener un carácter predictivo: ejecución según el plan inicial, y su objetivo para el trabajo era desarrollar un plan y mantener el cronograma y los recursos planificados, entre otras. Hoy en día todos los aspectos antes mencionados caracterizan la gestión de proyectos (Alexander, 2002).

Existe en el mercado una gran cantidad de productos de software orientados a la gestión de proyectos, muchos de los cuales son ampliamente utilizados, tales como: Team Foundation Server, GForce, Redmine, Trac, DotProject, Internet Primavera y Jira. Todas estas herramientas ofrecen diversos módulos con capacidades para la gestionar proyectos como son gestión de tiempo, gestión de la calidad, alcance, costos, gestión de recursos humanos, gestión de riesgos, gestión documental, entre otros.

A raíz de la necesidad de cumplir con el concepto de universidad productiva y con el objetivo de lograr recursos humanos formados se produce en la universidad una modificación de su plan de estudio y del modelo de producción simultáneamente, El centro de estas modificaciones vista desde la producción se enfoca en la creación de los centros de desarrollo y desde entonces comenzaron a desarrollarse una serie de actividades en los proyectos encaminadas a resolver todas aquellas tareas que poco a poco conformaban un producto. Todo este proceso desenmascaraba una serie de problemas que no permitían que esto se realizara de la forma más práctica y fácil posible, dado por la inexperiencia que existía en este sentido.

- Primeramente no existían roles, por lo que todos los miembros del equipo debían asumir tareas de cualquier índole.
- La comunicación era personal y directa, influenciado porque los proyectos generalmente eran pequeños.
- El cumplimiento de las tareas asignadas eran revisadas a través de chequeos semanales y se probaba con los instaladores del producto, con el objetivo de comprobar que la calidad era

adecuada.

- El avance del proyecto se definía a partir de estimaciones que se le establecían a las tareas, de acuerdo a parámetros relacionados con la complejidad de las mismas.
- Se tenía en cuenta la posible desviación de los cronogramas, de acuerdo al fondo de tiempo de los miembros del proyecto y de las demás actividades que podían surgir.
- En cuanto a la información de los integrantes del proyecto la recopilaba el líder personalmente, este encuestaba a cada miembro y se quedaba con los datos que realmente necesitaba (Perez, 2010).

Ya para el 2009, en estos centros, comenzaron a utilizarse diversas herramientas de gestión de proyectos que ayudaban a gestionar los procesos que se llevaban a cabo en los mismos, entre las más utilizadas se encuentran Redmine, Trac y GForge. Esto trajo como consecuencia que incluso en los mismos centros, se utilizaban variedades de herramientas, trayendo como consecuencia que no se cubrieran las necesidades ni se gestionara la información a nivel de centro ni a nivel UCI. Es por esto que surge la necesidad de la estandarización y formalización de los procesos, a través de la verificación y validación de estos (Enamorado, 2009).

Una de las disciplinas de la Ingeniería de Software que ayuda a la estandarización de los requisitos es la Ingeniería de Requisitos, la cual tiene como objetivo principal darle a todas las partes una explicación escrita del problema y como propósito, hacer que los requisitos alcancen un estado óptimo, antes de alcanzar la fase de diseño en el proyecto. Para la gestión de proyectos se hace necesario el uso de esta disciplina (Márquez, 2007).

En el año 2010 se desarrolla GESPRO v1.0, que es un paquete para la Gestión de Proyectos desarrollado por la Universidad de las Ciencias Informáticas y comercializable desde todas las empresas comercializadoras asociadas a la Universidad. Este ha estado en ejecución en un modelo de desarrollo basado en prototipos. GESPRO centra su base en el Redmine, herramienta que ya existía y que según las necesidades existentes en la universidad se le crearon nuevos módulos y funcionalidades para la correcta gestión de los proyectos (Perez, 2010).

A partir de un análisis realizado por el equipo de proyecto sobre el módulo Administración y Gestión de Usuarios se obtienen las siguientes deficiencias:

- Los requisitos definidos en el documento de Especificación de Requisitos correspondiente al módulo Administración y Gestión de Usuarios no están totalmente documentados, afectando el desarrollo de las etapas de la Ingeniería de Requisitos.

INTRODUCCIÓN

- La especificación de los requisitos correspondiente al módulo Administración y Gestión de Usuarios no está en el formato establecido por Calisoft, lo que provoca que el producto no pueda ser liberado.
- No se realizan investigaciones que permita identificar nuevas funcionalidades del módulo Administración y Gestión de Usuarios para que esté a la altura de los principales competidores, lo que provoca que el sistema GESPRO no esté actualizado según las necesidades que existen mundialmente.
- No están diseñados los casos de prueba correspondientes a los requisitos del módulo Administración y Gestión de Usuarios, lo que afecta el proceso de validación y la liberación del producto, interfiriendo en el proceso de aseguramiento de la calidad.

Teniendo en cuenta lo expuesto anteriormente el **problema científico** que se requiere formular es que las deficiencias en los procesos de identificación, especificación y validación de requisitos del módulo Administración y Gestión de Usuarios del sistema para la Dirección Integrada de Proyectos GESPRO 12.05 está afectando la evolución y el proceso de aseguramiento de la calidad, siendo el **objeto de estudio**: La Ingeniería de Requisitos en sistemas de Gestión de Proyectos.

Como **objetivo general** se pretende: realizar la Ingeniería de Requisitos del módulo Administración y Gestión de Usuarios del Sistema para la Dirección Integrada de Proyectos GESPRO 12.05 para contribuir a mejorar la calidad y su evolución, donde el **campo de acción** es: la Ingeniería de Requisitos en el módulo Administración y Gestión de Usuarios de GESPRO.

Para dar cumplimiento al objetivo de investigación se proponen los siguientes **objetivos específicos**:

- ✓ Establecer el marco teórico de la investigación a través del estudio del estado del arte de las principales tendencias en cuanto a funcionalidades del módulo Administración y Gestión de Usuarios en los sistemas de gestión de proyectos.
- ✓ Realizar una propuesta de solución para la Ingeniería de Requisitos del módulo Administración y Gestión de Usuarios para la Dirección Integrada de Proyectos GESPRO 12.05.
- ✓ Validar la Ingeniería de Requisitos realizada al módulo Administración y Gestión de Usuarios.
- ✓ Analizar la evolución del sistema en cuanto al módulo Administración y Gestión de Usuarios.
- ✓ Analizar el impacto del módulo Administración y Gestión de Usuarios de la herramienta GESPRO en la comunidad.

INTRODUCCIÓN

La **idea a defender** es que: Si se desarrolla la Ingeniería de Requisitos del módulo Administración y Gestión de Usuarios dentro del sistema GESPRO, contribuirá a mejorar la calidad del producto y su evolución.

Con el objetivo de dar cumplimiento al objetivo general se definen como **Tareas de investigación:**

- ✓ Definición de la estrategia de trabajo (cronograma e hitos de la investigación).
- ✓ Selección y revisión de bibliografías que permitan actualizar los enfoques existentes sobre los sistemas para Gestión de Proyectos.
- ✓ Evaluación del contenido de la información obtenida en la bibliografía y establecer un diagnóstico de las tendencias actuales en las funcionalidades de las áreas de Administración y Gestión de Usuarios y tomar posición al respecto.
- ✓ Levantamiento de las principales herramientas de gestión de proyectos que son más fuertes en el área de Administración y Gestión de Usuarios, que puedan aportar al desarrollo de la investigación.
- ✓ Estudio a profundidad de las principales características del sistema GESPRO para entender su visión general y la necesidad de su desarrollo para la Universidad de las Ciencias Informáticas y el país.
- ✓ Comparación entre GESPRO y las demás herramientas identificadas como las más completas en las áreas de Administración y Gestión de Usuarios para identificar insuficiencias y posibilidades de mejoras.
- ✓ Levantamiento de los requisitos existentes en el módulo Administración y Gestión de Usuarios de GESPRO.
- ✓ Diseño de los Casos de pruebas necesarios para las pruebas de liberación del módulo Administración y Gestión de Usuarios de GESPRO 12.05.

Para todo el proceso de investigación y elaboración de este trabajo se tendrá en cuenta la utilización de varios **métodos científicos de investigación** como:

- ✓ **Histórico–Lógico:** Método teórico utilizado para analizar la evolución y las tendencias actuales de las herramientas de gestión de proyectos, así como de la ingeniería de requisitos.
- ✓ **Analítico–Sintético:** Método teórico utilizado para estudiar los sistemas de gestión de proyecto, con el objetivo de comparar las funcionalidades del sistema GESPRO con los requisitos de otras herramientas de gestión más conocidas con el objetivo de encontrar fortalezas y debilidades.

- ✓ **Consulta de fuentes de información:** Método empírico utilizado para la consulta de las fuentes bibliográficas durante la investigación.
- ✓ **Observación participativa:** Método empírico utilizado para apreciar los resultados obtenidos.

Estructura de la tesis

Para el desarrollo del tema de investigación se proponen tres capítulos, los cuales están estructurados de la siguiente manera:

En el **capítulo 1** se definen conceptos como Gestión de Proyectos, se realiza un estudio de las causas que propiciaron el desarrollo de una nueva herramienta en la universidad que estandarizara los procesos y que estuviese centralizada. Se define Ingeniería de Requisitos, las fases y las tareas a cumplir en cada una de ellas, se describe el módulo Administración y Gestión de Usuarios enfocándose fundamentalmente en los problemas que existían en ellos.

En el **capítulo 2** se define la propuesta de solución para el desarrollo del trabajo, con el objetivo de hacerlo más sencillo y eficiente, este consiste en el cumplimiento de las actividades que se desarrollarán sobre el módulo Administración y Gestión de Usuarios enfocadas en cada una de las etapas de la Ingeniería de Requisitos. Por la importancia de esta para el desarrollo de la investigación se realiza también el flujo de trabajo de dicha propuesta para un mayor entendimiento del mismo y finalmente se describirá cada una de las etapas de la Ingeniería de Requisitos, y el cumplimiento de cada una de las técnicas identificadas.

En el **capítulo 3** se realiza la validación de los resultados obtenidos a partir de lo realizado, Se listan el total de requisitos identificados, tanto funcionales como no funcionales. Con el objetivo de conocer el alcance competitivo y comercial del sistema en cuanto al módulo Administración y Gestión de Usuarios se realiza una comparación de este con otras herramientas con el objetivo de identificar fortalezas y debilidades. Con el objetivo de visualizar los resultados obtenidos se realiza el análisis de las listas de chequeo y del diseño de casos de prueba y finalmente se analiza la evolución y el impacto de este módulo en la comunidad.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

Introducción

En el presente capítulo se sientan las bases que fundamentan el desarrollo de la tesis. Se define la gestión de proyectos, así como una breve descripción de las herramientas más usadas en los inicios en la universidad. Partiendo de esto y de la necesidad de una nueva herramienta que fuera capaz de estandarizar los procesos y que estuviese centralizada a nivel de la universidad, se realiza una breve descripción de la herramienta GESPRO, surgimiento y funcionalidades que posee. Por la influencia que tiene la Ingeniería de Requisitos (IR) sobre la gestión de proyectos, se define y se aborda sobre las fases y las tareas a cumplir en cada una de ellas. Finalmente se describe el módulo Administración que es muy importante para la herramienta GESPRO y que es el fundamento de la tesis.

1.1 Gestión de proyectos.

La gestión de proyectos es la disciplina de organizar y administrar recursos de manera tal que se pueda culminar todo el trabajo requerido en el proyecto dentro del alcance, el tiempo, y coste definidos (Perez, 2010).

La dificultad de la gestión de un proyecto radica en gran medida en la cantidad de personas involucradas, pues el éxito del proyecto y la satisfacción de la necesidad por la que fue emprendido, generalmente es responsabilidad de diferentes personas, por lo que se considera necesario asegurarse que el producto que se está creando cumpla claramente con las expectativas del cliente (Capuz, 1999).

En la gestión de proyecto, surgen cambios en las planificaciones, configuraciones, en las herramientas que se usan, en los sistemas de trabajo, y esto acelera un proceso de investigación, trabajos de mesa, lluvia de ideas, entrevistas, cursos que tributan a la preparación y conocimiento de cada uno de los individuos que forman parte del equipo de trabajo, por lo que es necesario la centralización de las orientaciones para una mejor organización y los miembros que intervienen reciban las tareas e indicaciones necesarias para la entidad de manera centralizada (Varas, 2000).

Etapas de la Gestión de Proyectos.

La Gestión de Proyectos está compuesta por fases o etapas que de ejecutarlas correctamente, garantizarán una correcta gestión de proyectos, estas son:

Fase de iniciación.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

En esta fase se definen los objetivos del proyecto y de los recursos necesarios para su ejecución.

Fase de control.

Se refiere a la monitorización del trabajo realizado, analizando cómo el progreso difiere de lo planificado e iniciando las acciones correctivas que sean necesarias.

Fase de planificación.

Se trata de establecer cómo el equipo de trabajo deberá satisfacer las restricciones de prestaciones, planificación temporal y coste.

Fase de ejecución.

Representa el conjunto de tareas y actividades que suponen la realización propiamente dicha del proyecto, la ejecución de la obra que se trate.

Fase de Cierre.

El objetivo de esta actividad es concluir el proyecto, para lo que será necesario registrar toda la información que aún no lo esté, realizar el balance final del proyecto y extraer toda la documentación para archivarla adecuadamente (Cervantes, 2007).

1.2 Herramientas de Gestión de proyectos.

Sin entrar en metodologías de trabajo concretas, se puede decir que para gestionar adecuadamente un proyecto de desarrollo de software es recomendable disponer de las siguientes funcionalidades:

- ✓ **Sistema de planificación**, que nos permita organizar el proyecto en función de hitos, tareas y subtareas, con asignación y control de tiempos y recursos materiales y humanos.
- ✓ **Sistema de gestión documental**, que nos servirá para almacenar y mantener los documentos obtenidos o generados durante el desarrollo del proyecto y acceder a ellos cómodamente. Cada hito, tarea o subtareas puede implicar la obtención o generación de documentación, igualmente el sistema de gestión de proyectos debe permitir que almacenemos esa documentación en el propio sistema.
- ✓ **Sistema de control de versiones**, que se utilizará para permitir el desarrollo concurrente y para mantener la historia del código fuente y parte de la documentación producida en el proyecto.
- ✓ **Sistema de gestión de incidencias**, que se empleará para hacer el seguimiento de los errores detectados y sus correcciones, tanto aquellos reportados por los responsables de la prueba del software como por los desarrolladores o los usuarios finales (EUDE, 2001).

Existen en el mercado multitud de paquetes de software especializados que proporcionan estas y otras

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

funcionalidades que son necesarias para la gestión de proyectos, ya sea por separado o integrando múltiples funcionalidades en un único producto. Estos son ampliamente utilizados, tales como: Team Foundation Server, GForce, Redmine, Trac, DotProject, Internet Primavera y Jira. A continuación se describen cada una de estas herramientas.

1.2.1 Redmine.

Redmine es un gestor y planificador de proyectos con interfaz web, orientado a la coordinación de tareas, comunicación de participantes, y que puede especializarse en proyectos de desarrollo gracias a herramientas como la integración en un repositorio de código.

Permite gestionar múltiples proyectos desde una sola interfaz con una ventana de navegador. La navegación es muy sencilla y se puede cambiar de proyecto en cualquier momento. Además cada proyecto puede tener una configuración totalmente diferente y el usuario tener un rol distinto en cada uno. Los proyectos pueden definirse como privados, en los que el administrador debe dar acceso a cada miembro, o públicos, visibles para todo el mundo. En Redmine cada proyecto es totalmente personalizable, pudiendo encontrar proyectos muy distintos entre sí según sus objetivos. Si un proyecto está enfocado a notificar incidencias, se puede configurar para incluir solo peticiones, si se busca un proyecto más colaborativo, la wiki y las noticias son una buena opción, e incluso se puede habilitar un proyecto solo con un foro. Configurando previamente el servidor de correo SMTP, Redmine permite enviar notificaciones por correo electrónico en todos los proyectos, definiendo antes los eventos que activan estos avisos. Además cada usuario en su configuración puede elegir recibir notificaciones de cualquier evento, o solo las relacionadas con él.

Algunas otras funcionalidades que habría que destacar son la página personal de cada usuario, que ofrece una vista personalizable con información de todos los proyectos donde está participando, como un calendario global, o peticiones asignadas. También se pueden subir ficheros, definir campos personalizados para cada módulo, usar la barra de búsqueda global, y ampliar la funcionalidad con decenas de extensiones. Además cabe mencionar que admite como bases de datos MySQL, SQLite y PostgreSQL (CESLCAM, 2010).

Ventajas.

La principal ventaja que nos aporta como gestor de proyectos es poder tener toda la información asociada a un proyecto acotada dentro del mismo. Además permite el control de la ejecución del mismo, todo ello a través de una interfaz web que hace sencilla la gestión de los mismos (Arencibia,

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

2012).

Desventajas.

La mayor dificultad del Redmine queda sin duda en la instalación, que puede resultar muy compleja, no solo por los paquetes requeridos, sino por los pasos que hay que seguir. Existen paquetes **.deb** que facilitan el proceso, pero no están bien documentados, quedando un paso por realizar a mano para ejecutar la aplicación. Aunque hay mucha documentación al respecto, es demasiado informal y no siempre da resultados en todos los casos (Arencibia, 2012).

En resumen, Redmine ayuda en gran medida a gestionar de manera ágil todos los proyectos de una empresa, la documentación asociada, los archivos que anexan y todo ello controlando en cada momento la ejecución del mismo y su administración es muy sencilla (Arencibia, 2012).

1.2.2 DotProject.

Fue creado por *dotmarketing.org* en el año 2000, diseñada para gestionar proyectos y sus funciones de control. Está construido por aplicaciones de código abierto y es mantenida por un pequeño pero dedicado grupo de voluntarios. El entorno está programado en PHP y JavaScript como lenguajes en el lado del servidor y MySQL como base de datos, lo que lo convierte en un entorno muy accesible a una gran comunidad de programadores. La licencia de distribución, se engloba dentro de los términos de la GNU General Public License de la Free Software Foundation. Esto implica que todo el código se puede copiar y distribuir libremente mientras se reconozca la autoría del autor original. DotProject puede ser descargado gratuitamente desde su página oficial (Torres, 2008).

La última versión del programa es la 2.1, que fue lanzada en febrero de 2007, tres años después de la publicación de la primera versión estable del mismo. Desde entonces han sido muchas las empresas que de alguna forma u otra se han interesado por su uso, tanto empresas lucrativas como no lucrativas (Team, 2007).

Ventajas.

El programa es lo suficientemente robusto como para su implantación en un entorno real. Gracias a las ventajas que reporta su licencia de uso, los errores pueden ser corregidos por la propia empresa. Además de ser gratuito es ventajoso no sólo desde el punto de vista económico, sino que también como una reducción del riesgo que supone una eventual adaptación, especialmente si se lleva a cabo

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

como acuerdo de prácticas en la empresa. El hecho de que toda la base de datos emplee MySQL asegura tanto la portabilidad del contenido registrado en ella en caso de rechazo del sistema una vez implantado, como una garantía de robustez gracias a las facilidades de copias de seguridad y almacenamiento distribuido y replicado que ofrece este sistema (Sánchez, 2008).

Desventajas.

La compatibilidad de módulos entre versiones (1.0 a la 2.0) no está garantizada, ya que en la nueva versión se incluyen características, como la nueva gestión de permisos, que alteran seriamente su integración con el entorno. Existe una sección en el foro de soporte dedicada exclusivamente a solucionar las cuestiones relacionadas con los problemas de compatibilidad. Actualmente la información se encuentra distribuida entre dos wikis, lo que trae problema a la hora de encontrar documentación sobre cómo programar para DotProject. La documentación disponible en cuanto a soporte es muy escasa desde el momento en que se necesite información más técnica que los simples manuales de usuario. Las respuestas en el foro de soporte pocas veces son atendidas y demoran en ser resueltas (Team, 2007).

1.2.3 Team Foundation Server.

Team Foundation Server (TFS) no es simplemente una herramienta para el control de versiones de código fuente, su alcance va mucho más allá, facilitando la gestión completa del ciclo de vida de la aplicación, desde la fase de diseño hasta las pruebas, pasando por la integración continua o la calidad del código. TFS incorpora varios sistemas integrados: por un lado una base de datos en SQL Server, que contiene no sólo el código fuente de nuestras aplicaciones sino los elementos de trabajo que posibilitan el seguimiento del desarrollo; los datos se integran en un Data Warehouse¹ de SQL Server Analysis Services que proporciona información sobre el estado del proyecto mediante informes en Reporting Services o Excel, permite la compilación desatendida de los proyectos y genera informes de calidad de la compilación; y todo ello se puede integrar en portales de colaboración de Microsoft Sharepoint, para que todo el equipo pueda compartir información, documentos o calendarios asociados al proyecto (Macarrón, 2011).

¹ Almacén de datos centralizado. Es una colección de datos orientada a un determinado ámbito (empresa, organización), integrado, no volátil y variable en el tiempo, que ayuda a la toma de decisiones en la entidad en que la utilice.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

Ventajas.

La mayor ventaja es que no importa dónde esté físicamente sentado el integrante del equipo, ya sea desarrollador, tester, arquitecto, esto facilita el armado de equipos a distancia. No es necesario tener un gran equipo, se puede utilizar en equipos de 7 u 8 personas, o también de 1 ó 2. TFS no produce recarga, sólo ordena el trabajo. Es utilizado para proyectos de desarrollo, con muy diferentes tecnologías, pero también para proyectos de consultoría donde no se escribe ni una línea de código. También se pueden tener proyectos solo para control de versiones, o para capacitación. Alternar proyectos con diferentes metodologías según los equipos, o personas que pertenecen a diferentes proyectos. En fin, es aplicable en todos los escenarios (Scalzone, 2011).

Desventajas.

Además de ser muy práctica y fácil de usar, es considerada la principal herramienta para administración de la configuración usada por desarrolladores, pero desafortunadamente no es una herramienta que cumpla con todos los requerimientos de una herramienta de administración de la configuración y la desventaja más grande es que no está enfocada para equipos de desarrollo grandes sino para desarrolladores únicos o bien equipos de desarrollo muy pequeños, el método de almacenamiento es por medio de sistemas de archivos el cual es una gran desventaja, el control de versiones depende de cada punto de vista (Macarrón, 2011).

1.2.4 Trac.

Trac es un sistema que integra varios componentes con capacidades suficientes para la Gestión de Proyectos de desarrollo de software. Es un sistema web multiplataforma ligero y extensible. Trac es un programa pensado para desarrolladores que necesitan mantener un proyecto. Programado en python y ejecutado a través de *mod_python* usando un servidor web. Permite llevar una serie de utilidades propias para cualquier proyecto (López, 2010).

Trac ha sido concebido de forma modular donde se pueden añadir plugins que proporcionan distintas funcionalidades. La misma cuenta con un módulo administración, donde se personaliza el entorno, se manejan los usuarios así como permisos, plugins, etc. En cuanto a la autenticación, posee de tipo LDAP, BBDD o fichero y entre los servicios adicionales que presenta se encuentran los blogs, foros, etc.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

Ventajas.

Trac permite llevar control total de un proyecto, así como todas las tareas, la documentación y el código fuente. Permite además tener en un solo sitio, accesible vía web, todos los archivos que conforman un proyecto, las tareas, y el código fuente. Con Trac es muy fácil relacionar cambios en el código con tareas o con la documentación gracias a su sintaxis wiki, de este modo se puede tener una referencia para el futuro de qué cambios se hicieron en cada parte del código, cuánto tardó y los problemas que son encontrados. Trac utiliza una aproximación minimalista² para el manejo de los proyectos, ya que la misión fundamental es ayudar a los desarrolladores a escribir un buen software manteniendo la gestión de los cambios controlados. Trac provee una interfaz con Subversion, e integra un Wiki que provee herramientas para realizar gestión de metas y reportes de bugs (Edgewall, 2003).

Desventajas.

Entre las principales inconvenientes que se han identificado de la herramienta es que la misma no fue pensada para la gestión de proyectos en sí, sino desde el punto de vista del desarrollo software, por lo que cada proyecto es independiente y no se comparte la base de datos entre ellos, aunque existe un plugin multiproyecto que resuelve este defecto. El flujo de trabajo del sistema de tareas no es extensible desde la interfaz web a pesar de que hay desarrollos para resolver este defecto. El sistema de permisos no es actualmente muy flexible y tiene integrado motores de búsqueda avanzada (Norambuena, 2009).

1.2.5 Jira.

Jira es una aplicación basada en web para el seguimiento de errores, incidentes y para la gestión operativa de proyectos, es también utilizada en áreas no técnicas para la administración de tareas. La herramienta fue desarrollada por la empresa australiana Atlassian. Inicialmente se utilizó para el desarrollo de software, sirviendo de apoyo para la gestión de requisitos, seguimiento del estatus y más tarde para el seguimiento de errores. Jira puede ser utilizado para la gestión de procesos y para la mejora de procesos, gracias a sus funciones para la organización de flujos de trabajo.

Jira está basada en Java, esto hace que funcione en varios bancos de datos y sistemas operativos, dispone también de paneles de control adaptables, filtros de búsqueda, estadísticas y función de

² Significa minimista, o sea, que utiliza lo mínimo, este término se refiere a cualquier cosa que haya sido reducida a lo esencial, despojada de elementos sobrantes.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

correo electrónico. La flexible arquitectura de Jira permite al usuario crear ampliaciones específicas que pueden incluirse en la *Extensión de la biblioteca de Jira*. Entre las características encontramos que se pueden administrar las tareas, trámites, defectos, procesos, requerimientos, ideas, permite adjuntar documentos, es un poderoso sistema de búsqueda en lenguaje natural de actividades, es un poderoso sistema de reportes haciendo el uso de filtros, es compatible con casi todas las bases de datos y posee una fácil extensión e integración con otros sistemas (Lacoba, 2000).

A través de las variadas aplicaciones que Atlassian ofrece en combinación con Jira, más las diferentes licencias disponibles y sumada a la gran variedad de plugins, Jira puede ser utilizado por cualquier tipo de compañía que encare tanto proyectos grandes, como medianos y pequeños ya que es utilizado por más de 11.000 compañías.

Ventajas.

Además que los costes son razonables, su flexibilidad es tal que se puede decir que es una herramienta para todos los negocios. Jira brinda posibilidad de acceso y gestión web, creación de flujos de datos, múltiples proyectos, múltiples tipos de usos, generación de informes, un potente motor de búsqueda de información, creación de cuadros de mando personalizados, intuitivo en su uso, abierto a la creación de extensiones a través de su API, integrado con otras herramientas software: gestores de contenidos, gestores de proyectos, etcétera.

1.2.6 GForge.

GForge es una herramienta para el desarrollo de software en forma comunitaria que permite organizar y administrar gran cantidades de proyectos, proporciona un conjunto integrado de herramientas que facilitan el trabajo en colaboración, y en concreto, la gestión de proyectos de software libre que pueden acceder a diversos servicios. Entre las características encontramos que es un sistema abierto, libre, escalable, robusto, en cuanto a funciones permite la gestión, comunicación, coordinación, centralización y control de sus actividades (Altamiranda, 2000).

Ventajas.

Es una herramienta simple, poderosa y amigable que permite centralizar y homogeneizar la gestión de sus proyectos, posee un poderoso sistema de búsqueda en lenguaje natural de actividades, así como un sistema de reportes haciendo el uso de filtros, permite adjuntar documentos y brinda servicios como notificaciones vía correo.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

Con el uso de esta herramienta se logra aumentar la productividad, ya que posee herramientas comunes para toda la empresa, es compatible con casi todas las bases de datos, tiene fácil extensión e integración con otros sistemas y posee una adecuada centralización de los recursos técnicos en un servidor (Altamiranda, 2000).

Desventajas.

Persisten problemas en cuanto a usabilidad y lógica en la interfaz. Problemas en cuanto a menús, las autorizaciones pendientes. No es gratuito ya que la licencia cuesta 5000 dólares, lo que implica que los plugins son solo para la versión paga, por lo que solo una empresa bien montada puede utilizarlos (Altamiranda, 2006).

Resultado del estudio de las Herramientas.

El estudio de las distintas herramientas previamente definidas, permitió seleccionar cuáles serían las que se utilizarán para la comparación de GESPRO. Esta selección se realizó teniendo en cuenta aquellas que fueron más fuertes en cuanto a funcionalidades del proceso de Administración y Gestión de Usuarios. Luego del estudio fueron seleccionadas todas, puesto que de una forma u otra, todas tienen procesos que viéndolos desde el módulo Administración, podrían hacerle competencia al sistema GESPRO lo que posibilitaría la identificación de nuevas funcionalidades.

1.3 Gestión de proyectos en la UCI.

En el 2008 surgen en la UCI los Centros de Desarrollo de Software, en los que se concentraba toda la producción de la universidad. En estos centros se hacía necesario el uso de herramientas gestoras de proyectos, es por eso que se centran en usar la herramienta de gestión de proyecto que consideraron más conveniente. Lo negativo de esto fue, que cada proyecto por su parte hizo lo mismo, lo que trajo como consecuencia que incluso en el mismo centro se utilizaban variedades de herramientas.

En una entrevista realizada a especialistas de CALISOFT a finales del año 2009 arrojó que eran utilizadas en la universidad ocho herramientas para la gestión de proyectos. Con los porcentajes como se muestra en la figura.(Enamorado, 2009)

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

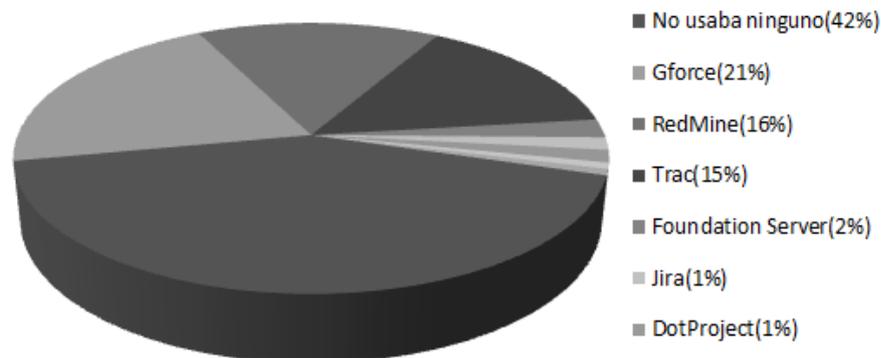


Figura 1. Resultados de la entrevista a Calisoft en el 2009.

Es por esto que se hace necesario el uso de una herramienta que esté centralizada a nivel de universidad y además que sea propia, que garantice soberanía tecnológica y estandarización. Esta herramienta es GESPRO.

Surgimiento de GESPRO.

Sin duda alguna existen múltiples soluciones para la gestión de proyectos en el mercado, pero no todas ofrecen soluciones a ciertas necesidades con un nivel adecuado de seguridad, con posibilidades de soporte y control de la tecnología a un costo razonablemente bajo.

GESPRO es un paquete para la Gestión de Proyectos desarrollado por la Universidad de las Ciencias Informáticas y comercializable desde las empresas comercializadoras asociadas a la Universidad. Este ha estado en ejecución en un modelo de desarrollo basado en prototipos. GESPRO para la gestión de proyectos de la UCI cuenta con una serie de herramientas que permiten la gestión de algunas variables distribuidas en diferentes módulos. En estos momentos el sistema cuenta con más de 6000 usuarios y se encuentra desplegado en 13 centros productivos y otras áreas administrativas de la Universidad y del país.

GESPRO centra su base en el REDMINE, herramienta que ya existía y que según las necesidades existentes en la universidad se le crearon nuevos módulos y funcionalidades para la correcta gestión de los proyectos como son, entre otros, la creación de perfiles o roles y la autenticación por LDAP en el caso del módulo Administración y Gestión de Usuarios.

Entre las funcionalidades que brinda GESPRO están:

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

- La posibilidad de desarrollar una solución integral, que incluye: la solución de software, el montaje de la infraestructura tecnológica requerida, tanto para un modelo de negocio basado en el hospedaje de las aplicaciones y acompañamiento, o como un modelo de negocio basado en la entrega de la aplicación informática.
- Garantía de un soporte sin límite.
- Seguridad en el manejo de su información más allá de los acuerdos de confiabilidad: existe la garantía real del uso que se le dará a la información, por eso nuestros principales clientes son entes de gobiernos.
- Un precio competitivo, siempre por debajo del competidor más cercano con facilidades de pago en dependencia de las formas en que se acceda al servicio o el producto.
- La posibilidad de un acompañamiento, asesoría y capacitación que va desde el uso de la herramienta hasta capacitación y formación en gestión de proyectos (PRODUCCIÓN, 2011).

1.4 Ingeniería de Requisitos.

La Ingeniería de Requisitos (IR) es una fase fundamental en la Ingeniería de Software, está presente en los inicios del proceso de desarrollo de cualquier sistema informático, jugando un papel primordial ya que se basa en transformar las necesidades en requisitos de software consistentes y entendibles, para que se haga más sencillo el análisis y diseño del producto que se desea (Sommerville, 2005).

A continuación se proponen algunos conceptos que han formulado diversos autores para describir qué es la Ingeniería de Requisitos y qué funciones cumple en el proceso de desarrollo de software.

- ✓ Rational Software Corporation la define como el enfoque sistémico para recolectar, organizar y documentar los requerimientos del sistema; es también el proceso que establece y mantiene acuerdos sobre los cambios de requerimientos, entre los clientes y el equipo del proyecto (danielvn7, 2008).
- ✓ Roger Pressman la define como la ayuda a los ingenieros de software a entender mejor el problema en cuya solución trabajarán. Incluye el conjunto de tareas que conducen a comprender cuál será el impacto del software sobre el negocio, qué es lo que el cliente quiere y cómo interactuarán los usuarios finales con el software (Pressman).

Luego de estudiar estas definiciones se concluye que la Ingeniería de Requisitos no es más que el conjunto de actividades, en las que utilizando técnicas y herramienta, se adapta el problema a las

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

necesidades que el cliente tiene, a través del análisis de estas y con el objetivo de satisfacer las expectativas y necesidades del mismo.

1.4.1 Fases o Etapas de la Ingeniería de Requisitos.

Diferentes autores estudiosos del tema describen y descomponen el proceso de IR de diversas maneras como se muestra a continuación.

- ✓ Rzepka lo define en tres actividades donde están: licitación, análisis y validación de los requerimientos que fueren obtenidos.
- ✓ Dorfman y Thayer describen la IR como un proceso de cuatro fases como son: Identificación, Análisis, Especificación y Validación de los Requerimientos (Escalona, 2004).
- ✓ Para otros autores la IR es un proceso que se describe en cinco actividades diferentes, las mismas son la identificación, análisis y negociación, especificación, modelado de sistema y validación de los requisitos, estas se documentan, analizan y definen los componentes que se van a producir, teniendo en cuenta las restricciones que tendrá el software (Chaves, 2006).

Muchos de estos autores reflejan sus criterios sobre la IR, todos con una perspectiva común en cuanto a métodos, técnicas y principios para identificar, analizar, especificar y validar los servicios que deben proporcionar el sistema y sus restricciones. Para un trabajo más fácil y sencillo se comparte la definición de Dorfman y Thayer, que definen la IR por cuatro fases como son identificación, análisis, especificación y validación. Estas técnicas son de gran importancia para el desarrollo de este trabajo. A continuación se describen cada una de ellas.

1.4.1.1 Etapa Identificación.

En esta etapa se obtiene gran cantidad de información para lograr un mejor entendimiento del negocio. Se busca la comprensión de las actividades que se realizan y las restricciones definidas para estas. Para la realización de las mismas se crea un equipo multidisciplinario que será el encargado de obtener la información del negocio, para posteriormente clasificarla y transformarla en requerimientos de software. Además estos requerimientos son evaluados para eliminar posibles falsos requisitos, se priorizan y finalmente son documentados en un lenguaje entendible para cliente, usuarios finales y desarrolladores (IEEE, 1990).

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

Técnicas de Captura de Requisitos.

La IR es un proceso complejo y engorroso, para ayudar a que este sea más eficiente y preciso se establecen métodos y técnicas que faciliten la identificación de los requerimientos de software. A continuación se hace referencias a las más conocidas y usadas.

1. Brainstorming (Tormenta de ideas): es una técnica donde se reúne un grupo de personas y exponen sus ideas sobre un problema y cómo este puede ser solucionado. La tormenta de ideas ayuda a generar una variedad de vistas del problema y esto aumenta las diferentes formas de modelarlo. Además es fácil de aprender y requiere menos organización que otras técnicas de identificación de requisitos.
2. Entrevistas: es la técnica más utilizada en la obtención de requisitos de software pues es la forma más natural que tiene el ser humano de intercambiar criterios durante la comunicación. El objetivo de ésta es obtener toda la información posible desde la visión que el entrevistado tiene de los requisitos. Los analistas necesitan ser sensibles a las dificultades que algunos entrevistados crean durante la entrevista y saber cómo tratar con problemas potenciales. Resulta útil además planificar la entrevista así como evaluar los resultados una vez realizada esta con el fin de evitar riesgos; no pueden ser improvisadas porque estaría expensa a olvidarse detalles importantes. Esta técnica depende en gran medida de las habilidades del entrevistador, la fluidez de su lenguaje y perspicacia para captar las ideas en un tiempo bastante limitado.
3. Encuesta: esta técnica requiere que el analista conozca el ámbito del problema en el que está trabajando. Consiste en redactar un documento con preguntas cuyas respuestas sean cortas y concretas, o incluso cerradas por unas cuantas opciones en el propio cuestionario, este cuestionario será cumplimentado por el grupo de personas entrevistadas o simplemente para recoger información en forma independiente de una encuesta.
4. Escenarios o casos de uso: son descripciones donde se incluyen actores, eventos y objetivos de las operaciones, esta técnica requiere de una alta preparación de los interesados.
5. Juego de roles: los miembros del equipo de trabajo toman la posición de los interesados, encontrándose así problemas relacionados al sistema. Esta técnica proporciona a los analistas una perspectiva de los requisitos del sistema a desarrollar.
6. Revisión de documentos: Esta técnica depende de la información almacenada por las entidades acerca de los procesos y términos que se manejan dentro de la misma. Las entidades guardan información referente a sus procesos, los modelos o informes necesarios para el desarrollo de la

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

misma o para rendir cuenta a los organismos superiores. Pueden ser además videoconferencias y otros dependiendo del grado de automatización de las mismas. Este cúmulo de información es estudiado por los analistas en busca de captar bien todos los procesos para determinar los requisitos asociados a estos, y que luego deben ser verificados por otras técnicas de especificación de requisitos. Esta técnica debe ser vinculada con otras para lograr un resultado efectivo (Barba, 2001).

Existen muchas otras técnicas como son: Proceso de Análisis Jerárquico (AHP), Comparación de terminología y otras que no deben restársele importancia. Lo significativo de todas estas técnicas es que no son efectivas, por si solas, se hace necesario combinarlas para obtener resultados con un mayor grado de certeza.

Luego del estudio de estas técnicas para el desarrollo de la investigación se realizó un análisis sobre las más usadas integrando diferentes conceptos abordados sobre las mismas. Para ello se estudiaron las características del cliente, con el objetivo de pronosticar qué técnicas pudieran ser más efectivas durante la captura. Teniendo en cuenta estas características se decidió aplicar las técnicas de Entrevistas y Revisión de Documentos.

1.4.1.2 Etapa Análisis.

En esta etapa se realiza un estudio detallado de los requerimientos obtenidos, buscando eliminar ambigüedades en los mismos y peticiones de poca relevancia para el sistema, con vistas a excluir las inconsistencias que hayan sido detectadas. Los requisitos comienzan a ser transformados a un lenguaje más técnico, llevando los términos del usuario a un lenguaje de sistema de software (IEEE, 1990).

1.4.1.3 Etapa Especificación.

Partiendo de lo elaborado en la etapa anterior tales como funciones, datos, requerimientos no funcionales, objetivos, restricciones de diseño/implementación o costos, e independientemente de la forma en que se realice. En esta etapa se realizan las descripciones detalladas de los requerimientos, teniendo en cuenta las diferentes categorías existentes para clasificarlos, centrándose en las características que debe tener el sistema para satisfacer las necesidades de los clientes. Estas especificaciones deben ser comprensibles y sin ambigüedades, expresadas de manera concisa y reflejando que cada requisito lleva a satisfacer una necesidad real. Si se presentan dificultades para especificar un requerimiento se debe volver a la etapa anterior que se crea conveniente (Quizhpe,

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

2002).

1.4.1.4 Etapa Validación.

Esta es una etapa que se nutre de las anteriores, se tienen en cuenta para la validación solamente aquellos requisitos que salieron de la etapa de especificación. Con la validación se evitan que malas interpretaciones y algunos errores puedan echar por tierra todo trabajo realizado en el trabajo realizado en etapas anteriores.

Una vez que han sido definidos los requisitos de software estos deben ser validados con el fin de demostrar que cumplen realmente las condiciones para el sistema que el usuario desea. Es necesario eliminar cualquier problema o inconsistencia que pueda existir en el documento de requerimientos.

Además los costos de errores en requerimientos son altos por lo cual es muy importante detectar y solucionar estos errores mediante el empleo de técnicas para la validación de requisitos (Quizhpe, 2002).

Técnicas de Validación de Requisitos.

A continuación se muestran algunas de las técnicas que pudieran emplearse para validar que los requisitos están correctamente especificados:

1. Listas de chequeo: Permite mediante una serie de preguntas, donde se comprueba la presencia de información en los documentos, detectar puntos que no se le hayan dado solución.
2. Prototipo: Con esta técnica se conoce claramente el nivel de aceptación del cliente ante los requisitos mostrados, le permite tener una idea de la estructura de la interfaz del sistema para los usuarios.
3. Matriz de trazabilidad: Se van marcando los objetivos del sistema y se chequea contra los requerimientos del mismo. Esto permite ir revisando los objetivos que cubre cada requerimiento, facilitando de esta manera encontrar inconsistencias y objetivos que no se vayan cumpliendo.
4. Diseño de casos de Prueba: Se realizan para determinar que un requisito es completamente satisfactorio (Barba, 2001).

Luego del análisis realizado de las técnicas de validación se concluye que esta etapa es de vital importancia, pues con ella se garantiza que el trabajo realizado no fue en vano. Por todo lo

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

anteriormente planteado en la presente investigación se decide utilizar las técnicas de Listas de Chequeo, la cual fue definido su uso, por Calisoft y la realización del diseño de los casos de prueba correspondientes a cada requisito.

1.4.2 Tareas por Etapas.

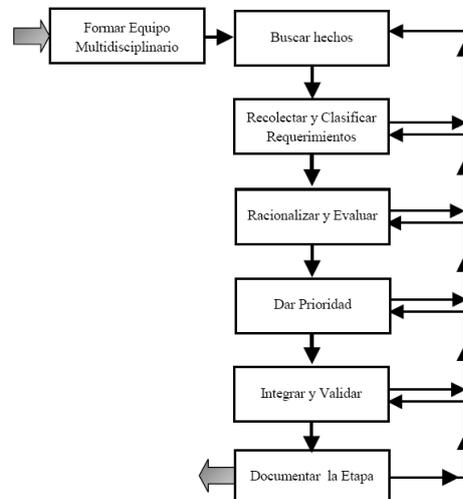
Etapa 1: Identificación.

En cuanto al proceso de Identificación se propone una solución metodológica, la cual consta de los siguientes pasos:

- 1.1 Formar el equipo multidisciplinario.
- 1.2 Buscar hechos.
- 1.3 Recolectar y clasificar requerimientos.
- 1.4 Evaluar y racionalizar.
- 1.5 Dar prioridad.
- 1.6 Integrar y validar.
- 1.7 Documentar la etapa.

Como es de esperar, a los efectos de obtener buenos requerimientos, todos estos pasos deben iterar ante la menor inconsistencia detectada, aconsejándose que la iteración se realice recurriendo al cliente/usuario, tantas veces como sea necesario, para garantizar una correcta depuración del producto final de la etapa de identificación (Barba, 2001).

Si se hiciera una representación gráfica de la propuesta metodológica para la etapa de identificación de requerimientos, la misma quedaría bosquejada de la manera que se puede apreciar en la figura 2.



CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

Figura 2. Etapas de la Identificación de Requerimientos.

Etapa 2: Análisis de Requerimientos.

Los pasos a realizar durante esta etapa tienen como objetivo la obtención de buenos requerimientos, para asegurar de esta manera que lo que se derivará a las etapas posteriores será de calidad tal que permita que disminuyan las fallas de los sistemas (Barba, 2001).

Las subetapas a contemplar durante esta etapa son:

2.1 Reducir ambigüedades en los requerimientos.

2.2 Traducir a lenguaje técnico los requerimientos.

2.3 Plantear un modelo lógico.

2.4 Documentar la etapa.

Etapa 3: Especificación de Requerimientos.

Recién superadas las etapas anteriores debe comenzar a pensarse en la forma de describir los requerimientos. Para ello, se deben seguir las subetapas planteadas a continuación (Barba, 2001).

3.1 Determinar el tipo de requerimiento.

3.2 Elegir la herramienta de especificación acorde al tipo de requerimiento.

3.3 Especificar de acuerdo a la herramienta seleccionada.

3.4 Documentar la etapa.

Etapa 4: Validación de los Requerimientos.

Todo el esfuerzo realizado durante las etapas anteriores puede darse por perdido si no es manifestado en forma correcta, ya que cualquier mala interpretación pueden echar por tierra hasta el proceso de Ingeniería de Requerimientos más consistente, desvirtuando la naturaleza de lo que fue, hasta el momento de su declaración, un buen requerimiento. Tratando de minimizar aún más la posibilidad de error, se proponen las siguientes subetapas para su elaboración (Barba, 2001).

4.1 Seleccionar las fuentes de información a partir de las cuales validar el documento de especificación.

4.2 Elegir o diseñar el modelo de documento acorde al grado de detalle requerido y al lector final.

4.3 Elegir la herramienta de documentación que mejor se aplica al modelo seleccionado.

4.4 Documentar respetando los estándares vigentes a la fecha de realización del documento de requerimientos.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

4.5 Validar.

4.6 Certificar.

1.5 Requisitos de Software.

Los requisitos de software deben ser especificados por escrito, como todo contrato o acuerdo entre dos partes, deben ser posibles de probar o verificar debido a que si un requisito no se puede comprobar no hay forma de saber si se cumplió con él o no, deben estar descritos como una característica del sistema a entregar, esto es, lo que el sistema debe hacer y no cómo debe hacerlo. Es importante destacar que deben estar redactados de la manera más abstracta y concisa posible para evitar malas interpretaciones.

1.5.1 Requisitos Funcionales del Sistema.

Los Requisitos Funcionales son capacidades o condiciones que el sistema debe cumplir. Es una tarea simple enunciada con un solo verbo y se corresponde con futuras opciones, acciones ocultas y condiciones extremas a determinar por el software.

1.5.2 Requisitos no Funcionales del Sistema.

Los Requisitos No Funcionales son propiedades o cualidades que el producto debe tener. Debe pensarse en estas propiedades como las características que hacen al producto atractivo, usable, rápido o confiable. Estos se dividen en grupos como se muestra a continuación:

- **Requisitos de Software:** debe mencionarse el software del que se debe disponer, después de implementado el sistema.
- **Requisitos de Hardware:** se deben enunciar los elementos de hardware que se necesitan para que el software cumpla sus funcionalidades.
- **Restricciones en el diseño y la implementación:** especifica o restringe la codificación o construcción de un sistema, son restricciones que han sido ordenadas y deben ser cumplidas estrictamente.
- **Requisitos de apariencia o interfaz externa:** este tipo de requisito describe la apariencia del producto. Es importante destacar que no se trata del diseño de la interfaz en detalle sino que especifican cómo se pretende que sea la interfaz externa del producto. También pueden ser necesidades de cumplir con normas estándares, o con los estándares de la empresa para la cual se esté desarrollando el software.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

- **Requisitos de Seguridad:** este es el tipo de requisito más difícil, que provocará los mayores riesgos si no se maneja correctamente. La seguridad puede ser tratada en tres aspectos diferentes:
 1. **Confidencialidad:** La información manejada por el sistema está protegida de acceso no autorizado y divulgación.
 2. **Integridad:** la información manejada por el sistema será objeto de cuidadosa protección contra la corrupción y estados inconsistentes.
 3. **Disponibilidad:** Significa que los usuarios autorizados se les garantizará el acceso a la información.
- **Requisitos de Usabilidad:** describen los niveles apropiados de usabilidad, dados los usuarios finales del producto, para ello deben revisarse las especificaciones de los perfiles de usuarios y las clasificaciones de sus niveles de experiencia.

1.6 Módulo Administración.

En la realización de todo software, un módulo para la administración de todos los procesos que se realicen es de vital importancia. Es desde este módulo donde se puede administrar todas las actividades que realicen los usuarios que ingresarán al software, ya sea para la creación de un proyecto, para la actualización de sus tareas o para cualquier otra actividad. En este módulo es donde se visualiza y controlan todas las operaciones que se realicen dentro del sistema.

GESPRO v1.0 cuenta con un módulo para la administración del sistema, este permite la administración centralizada de todos los recursos dispuestos en la red y que son explotados por la plataforma GESPRO. Permite además el monitoreo permanente de la plataforma, monitoreando más de 30 variables por cada servidor (Perez, 2010).

El módulo Administración está compuesto por 17 sub módulos como son:

1. Proyectos.
2. Usuarios.
3. Grupos.
4. Perfiles y Permisos.
5. Tipos de Peticiones.
6. Estado de las peticiones.
7. Flujos de Trabajo.
8. Campos personalizados.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

9. Listas de Valores.
10. Configurar el sistema.
11. Autenticación LDAP.
12. Configuración de despliegue.
13. Datos del Centro.
14. Sincronización pre-proyectos.
15. Importar recursos.
16. Extensiones.
17. Información.

Este módulo requiere de un nuevo levantamiento de requisitos pues hasta este punto no se cuenta con funcionalidades que ayuden a la gestión de tipo de peticiones, gestión de flujos de trabajo, campos personalizados, listas de valores entre otros.

1.6.1 Submódulo Gestión de Usuarios.

De igual modo se hace precisa la creación de un módulo que gestione toda la información referente a los usuarios que accedan al sistema, y es el caso del submódulo Usuarios. En él se gestionan todos los usuarios a través de sus datos de contacto.

GESPRO v1.0 cuenta con este submódulo con el objetivo de realizar un uso más eficaz de las personas involucradas con el proyecto. Esta consiste en la planificación organizacional, la adquisición de personal, y en el desarrollo del equipo (Perez, 2010).

Para la correcta gestión de usuarios se cuenta con las funcionalidades:

1. Registrarse en la aplicación.
2. Entrar al sistema.
3. Salir del sistema.
4. Modificar perfil.
5. Acceder a "Mi Página".

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

Conclusiones parciales del Capítulo.

1. El análisis de las diferentes etapas de la IR propuestas por varios autores, permitió definir cuál sería la que se utilizaría en el presente trabajo como son las enunciadas por Dorfman y Thayer, teniendo en cuenta las características y condiciones del proyecto y de su equipo de desarrollo.
2. A través del estudio y comprensión de cada técnica de captura de requisitos se decide utilizar: lluvia de ideas, entrevistas y revisión de documento, teniendo en cuenta que son las más adecuadas y completas para la investigación.
3. Se evidenció la necesidad de un nuevo levantamiento de requisitos al módulo Administración y Gestión de Usuarios, lo que contribuirá a mejorar la calidad y evolución del módulo.

CAPÍTULO 2: PROPUESTA DE SOLUCIÓN

CAPÍTULO 2: PROPUESTA DE SOLUCIÓN

Introducción

En el presente capítulo se define la propuesta de solución para el desarrollo del trabajo, con el objetivo de hacerlo más sencillo y eficiente, este consiste en el cumplimiento de las actividades que se desarrollaran sobre el módulo Administración y Gestión de Usuarios enfocadas en cada una de las etapas de la Ingeniería de Requisitos. Por la importancia de esta para el desarrollo de la investigación se realiza también el flujo de trabajo de dicha propuesta para un mayor entendimiento del mismo y finalmente se describirá cada una de las etapas de la Ingeniería de Requisitos, y el cumplimiento de cada una de las técnicas identificadas.

2.1 Propuesta de Solución para la Ingeniería de Requisitos del módulo Administración y Gestión de Usuarios.

La siguiente figura describe la propuesta de solución basada en la Ingeniería de Requisitos de acuerdo a las etapas definidas en el capítulo anterior. La misma pretende reflejar de forma clara y resumida, las actividades que se realizarán sobre el módulo Administración y Gestión de Usuarios del sistema GESPRO, definidas por cada una de las etapas de la Ingeniería de Requisitos. En el transcurso del capítulo se realizará una explicación más detallada de cada una de estas actividades.

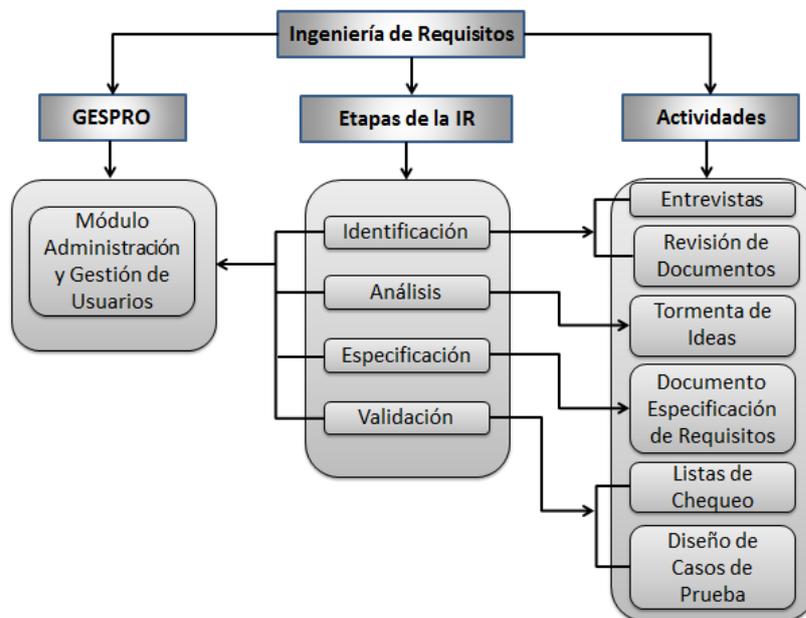


Figura 3. Propuesta de Solución.

CAPÍTULO 2: PROPUESTA DE SOLUCIÓN

Como se puede observar, la figura anterior (Figura 3), muestra la propuesta para la Ingeniería de Requisitos del módulo Administración y Gestión de Usuarios del sistema GESPRO 12.05, para la cual se propone que se haga en cuatro etapas como son: Identificación, Análisis, Especificación y Validación, para el cumplimiento de estas fases se realizan diferentes actividades como son:

1. En la fase de Identificación, la realización de entrevistas y la revisión de documentos para identificar los requisitos funcionales y no funcionales.
2. En la fase de Análisis, se utiliza la técnica de tormenta de ideas, con el objetivo de negociar y describir correctamente los requisitos identificados previamente.
3. En la etapa de Especificación, se especifican detalladamente cada uno de estos requisitos en el documento Especificación de Requisitos de Software del Expediente de Proyecto.
4. En la etapa de Validación, se aplican técnicas como las Listas de Chequeo y Diseño de Casos de Prueba para comprobar la calidad de los requisitos especificados y contribuir al proceso de Aseguramiento de la Calidad del Sistema.

CAPÍTULO 2: PROPUESTA DE SOLUCIÓN

2.2 Flujo de Actividades para la propuesta de Solución.

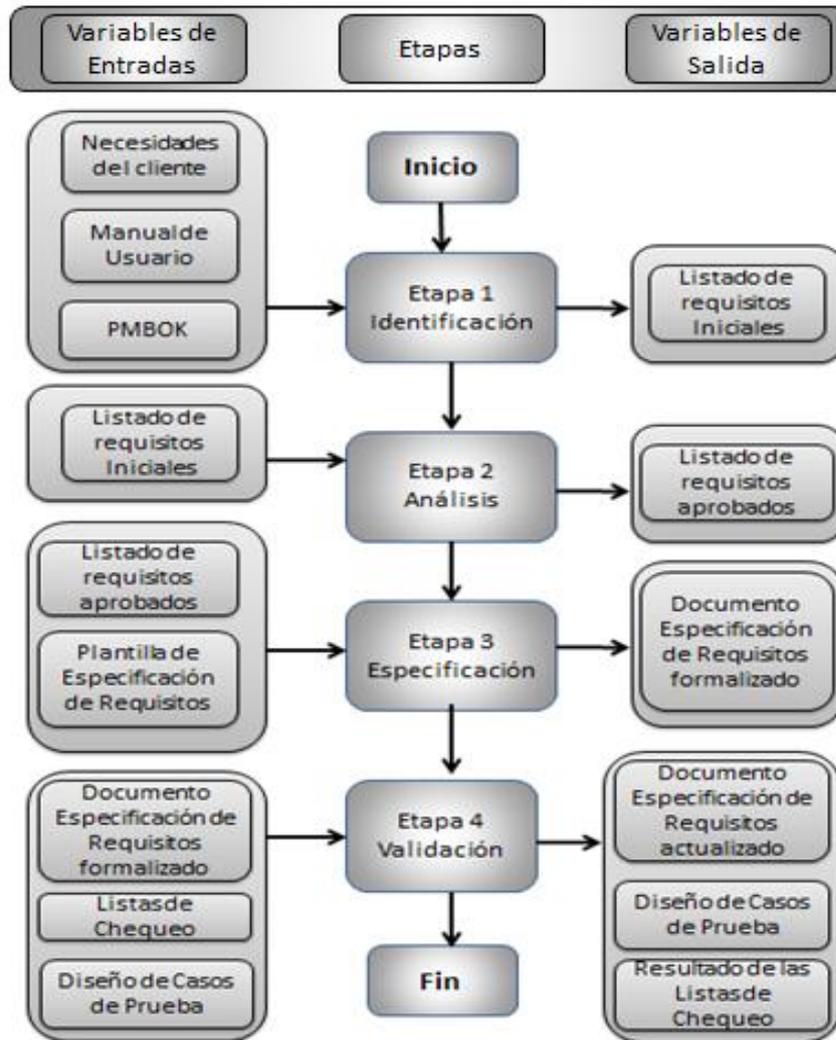


Figura 4. Vista del flujo de actividades de la propuesta de solución.

La Figura 4, muestra el flujo de actividades de la propuesta de solución para el módulo Administración y Gestión de Usuarios del sistema GESPRO 12.05, enfocados en las etapas definidas como son: Identificación, Análisis, Especificación y Validación.

Para una primera etapa se define el listado de requisitos iniciales a partir de las necesidades de los clientes, el manual de usuarios y el PMBOK, dándole salida a una segunda etapa, donde a partir del listado de requisitos iniciales y las entrevistas con los clientes se obtiene el listado de requisitos aprobados, en una tercera etapa, estos requisitos son especificados en el documento especificación de requisitos, pasando así a la cuarta y última etapa, en la que a partir del documento de especificación

CAPÍTULO 2: PROPUESTA DE SOLUCIÓN

de requisitos, las listas de chequeo y la plantilla del diseño de casos de prueba, se obtiene como salida el documento de especificación de requisitos actualizado, el diseño de los casos de prueba realizados y el resultado de las listas de chequeo.

2.3 Ingeniería de Requisitos del módulo Administración y Gestión de Usuarios del sistema GESPRO.

A continuación se expondrán cada una de las etapas de la ingeniería de requisitos ejemplificando y describiendo cada una de las actividades a cumplirse en cada una de ellas.

2.3.1 Etapa 1: Identificación de los requisitos.

Para el desarrollo de esta etapa, es importante recordar que GESPRO centra su base en el REDMINE, herramienta que ya existía y que según las necesidades existentes en la Universidad se le crearon nuevos módulos y funcionalidades, es por esta causa que se utilizó la ingeniería inversa, con el objetivo de obtener los requisitos que ya tenía esta herramienta y a partir de estos definir los nuevos de acuerdo a las necesidades nuestras (Alfredo, 2003).

En el capítulo anterior se definieron cuáles eran las técnicas de captura de requisitos, de estas técnicas se propone las entrevistas y la revisión de documentos. A continuación una explicación del uso de ellas.

Entrevistas.

Para el cumplimiento de esta técnica se realizaron entrevistas a trabajadores del Centro de Desarrollo Industrial, específicamente a los trabajadores del Laboratorio de Gestión de Proyectos (Jefe de proyecto, analistas y desarrolladores) con el objetivo de entender el funcionamiento del sistema, requerimientos y funcionalidades para hacer una correcta definición y especificación de los mismos.

Revisión de documentos.

La revisión de documentos es una técnica muy usada para la especificación de requisitos y muy fácil de realizar, para ello fueron revisadas las documentaciones existentes del sistema GESPRO, como son entre otros la ayuda de GESPRO, en la que se encuentran variados documentos de gran importancia, además del PMBOK entre otros.

CAPÍTULO 2: PROPUESTA DE SOLUCIÓN

2.3.2 Etapa 2: Análisis de los requisitos.

En esta etapa se hizo un análisis detallado de los requisitos definidos, o sea aquellos que luego de las entrevistas con los clientes, finalmente quedaron definidos, con el objetivo de eliminar ambigüedades y con vistas a excluir las inconsistencias, una técnica propuesta para lograr estos objetivos fue tormenta de ideas.

2.3.3 Etapa 3: Especificación de requisitos.

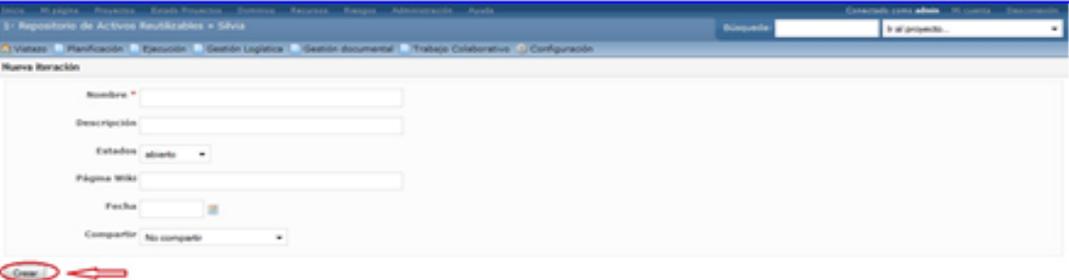
Para el cumplimiento de esta etapa se realizó la descripción detallada de los requerimientos, se utilizó el documento Especificación de Requisitos de Software, documento que propone Calisoft y que es de estricto cumplimiento. Esta plantilla quedó definida a peticiones de los contratistas del Programa de Mejoras. CMMI exige que la especificación de requisitos tenga un nivel suficiente de lenguaje técnico, lenguaje que sea apropiado, no solo para el entendimiento de los usuarios, sino también para los desarrolladores, para que estos, posteriormente puedan implementar el sistema que se quiere obtener.

La estructura del documento es sencilla, en él se recoge todos los requisitos del sistema en cuanto a requisitos funcionales la información que en él se recogen son:

- Nombre del requisito: Como indica es el nombre del requisito.
- Descripción: Breve descripción del requisito.
- Complejidad: Especificar si posee Alta, Media o Baja complejidad.
- Prototipo: En este campo se debe incluir el prototipo de interfaz gráfica o de usuario.
- Campos: Nombre del campo entrada y/o salida que se procede a describir.
- Tipo de datos: Tipo de dato que recibe o muestra, en caso de que no se reciba o muestre ningún tipo de dato se escribe no procede.
- Reglas o restricciones: Son las Reglas o restricciones que deben cumplirse o tenerse en cuenta. Valores válidos de los datos. Para ello es muy importante que el analista tenga conocimiento de las reglas del negocio y los requisitos funcionales. En caso de que no aplique se escribe no procede.

A continuación se muestra un fragmento del mismo.

CAPÍTULO 2: PROPUESTA DE SOLUCIÓN

RF1.7.1	Crear las iteraciones del proyecto	El sistema debe permitir definir nuevas iteraciones del proyecto.	Alta	Alta
Prototipo				
				
Campos		Tipos de Datos	Reglas o Restricciones	
Nombre		Campo de Texto	No procede	
Descripción		Campo de Texto	No procede	
Estado		Lista desplegable	No procede	
Página Wiki		Campo de Texto	No procede	
Fecha		Campo de selección	No procede	
Compartir		Lista desplegable	No procede	
Observaciones				
Variables de Salida		1. El sistema crea la iteración 2. El sistema muestra mensaje de campos incorrectos o inválidos.		

En el caso de los requisitos no funcionales:

- Nombre del requisito: Como indica es el nombre del requisito.
- Descripción: Descripción detallada del requisito.

Siguiendo el siguiente formato.

RnF 1. <Tipo de Requisito (Usabilidad, Eficiencia, etc) 1...n>
 [Descripción del requisito].

CAPÍTULO 2: PROPUESTA DE SOLUCIÓN

2.3.4 Etapa 4: Validación de los requisitos.

Evaluar los requisitos obtenidos ofrece la ventaja de conocer los errores para suprimirlos en procesos de desarrollo de software futuros. Las actividades de la Ingeniería de Requisitos tienen su contrapartida evaluativa en la validación de los requisitos de software, desde el modelamiento del negocio como punto de partida de las actividades de Ingeniería de Requisitos hasta la modelación del sistema.

En el capítulo anterior se definieron cuáles eran las técnicas de validación de requisitos, de estas técnicas se propone las listas de chequeo y el diseño de casos de prueba. A continuación una explicación del uso de ellas.

Listas de Chequeo.

Con el objetivo de validar que el levantamiento de requisitos se hizo de forma correcta, se propone utilizar las Listas de chequeo, del expediente del proyecto propuesta por Calisoft. en la cual la columna “Peso” define si el indicador a evaluar es crítico o no, mientras que la columna “Eval” proporciona una evaluación para el indicador, que puede ser: 1 en caso de mal y 0 en caso que elemento revisado no presente errores. La columna “Cantidad de elementos afectados” muestra la cantidad de errores encontrados sobre el mismo indicador y en “Comentarios” se incluyen los señalamientos o sugerencias que quiera incluir la persona que aplica la lista de chequeo. Para más detalle, ver documento Listas de Chequeo en el Expediente de Proyecto de GESPRO/1.ingeniería/1.1 requisitos. A continuación se muestra parte de la misma.

Estructura del documento					
Peso	Indicadores a Evaluar	Eval	(NP)	Cantidad de elementos afectados	Comentarios
crítico	1. ¿Está el documento acorde con la plantilla estándar del proyecto o del expediente de proyecto?	0			
crítico	2. ¿Contiene las secciones obligatorias definidas en el expediente? (Ver Expediente de Proyecto)	0			

Elementos definidos por la metodología

CAPÍTULO 2: PROPUESTA DE SOLUCIÓN

Peso	Indicadores a Evaluar	Eval	(NP)	Cantidad de elementos afectados	Comentarios
crítico	1. ¿Se han identificado los requisitos funcionales que son las características que el sistema debe cumplir?	0			
crítico	2. ¿Se han identificado los requisitos no funcionales del sistema, que son las cualidades que el sistema debe tener?	0			
crítico	3. ¿Están todos los requisitos redactados de forma simple y clara para aquellos que vayan a consultarlo en un futuro? Ver documento de Especificación de Requisitos.	0		1	
	4. ¿Debería especificarse algún requisito con más detalle? Ver documento de Especificación de Requisitos.	0		3	
	5. ¿Debería especificarse algún requisito con menos detalle? Ver documento de Especificación de Requisitos.	0		1	
crítico	6. ¿Todos los requisitos identificados se centran en lo que el sistema debe hacer y no como el sistema debe hacerlo? Ver documento de Especificación de Requisitos.	0			
crítico	7. ¿Han sido abordados e identificados los valores de entradas y salidas? Ver documento de Especificación de Requisitos.	1			En la plantilla no están identificados los valores de salida

CAPÍTULO 2: PROPUESTA DE SOLUCIÓN

	8. ¿Han sido incluidas las respuestas válidas y no válidas de los valores de entrada? Ver documento de Especificación de Requisitos. Nota(puede hacerse referencia a un documento Diccionario de Datos)	0			
	9. ¿Se han identificado los requerimientos de software y de hardware? Ver documento de Especificación de Requisitos.	1			
	10. ¿Han sido identificadas las restricciones de diseño e implementación? Ver documento de Especificación de Requisitos.	1			
	11. ¿Han sido identificadas las restricciones de interfaz externa? Ver documento de Especificación de Requisitos.	0			
	12. ¿Los requerimientos de soporte, usabilidad, fiabilidad y eficiencia han sido identificados? Ver documento de Especificación de Requisitos.	0			
	13. ¿Se han identificado los requerimientos de seguridad (confidencialidad, integridad, disponibilidad)? Ver documento de Especificación de Requisitos.	0			
crítico	14. ¿Se puede verificar cada requisito? (Un requisito se dice que es verificable si existe algún proceso no excesivamente costoso por el cual una persona o una máquina pueda chequear que el software satisface dicho requerimiento, ejemplo la especificación del	0			

CAPÍTULO 2: PROPUESTA DE SOLUCIÓN

	caso de uso). Ver documento de Especificación de Requisitos.				
	15. ¿Se han enumerado los requisitos incluso los que se derivan de otros requisitos? Ver documento de Especificación de Requisitos.	1		3	
	16. ¿Se puede trazar cada requisito al origen en el entorno del problema, (caso de uso del negocio)? Ver documento de Especificación de Requisitos.		NP		
	17. ¿No aparece un mismo requisito en más de un lugar del documento de especificación? Ver documento de Especificación de Requisitos.	1			
crítico	18. ¿Son los requisitos consistentes? ¿No existe contradicción entre lo especificado por un requisito y lo especificado por otro? Ver documento de Especificación de Requisitos.	0			
	19. ¿Existe correspondencia entre el modelo de caso de uso, las Especificaciones Suplementarias y las especificaciones de requerimientos? Ver documento de Especificación de Requisitos.		NP		
	20. ¿Soportan los requerimientos los objetivos del negocio, sistema de software y el proyecto?		NP		
	21. ¿Algún requerimiento no es requerido o se encuentra fuera del alcance del proyecto?	1			

CAPÍTULO 2: PROPUESTA DE SOLUCIÓN

22. ¿Son las metas y objetivos del sistema de software clara y completamente definidos?	0			
23. ¿Se han manejado todos los eventos y condiciones?	0			
24. ¿Han sido especificadas todas las operaciones?	1			
25. ¿Son las operaciones suficientes para reunir los objetivos del sistema de software?	1			
26. ¿Se han especificados todas las definiciones y reglas requeridas?	0			
27. ¿Satisface las especificaciones el nivel de detalle requerido el equipo de diseño?	0			
28. ¿Los requerimientos se encuentran limpios de polarización de implementación (no restringidos a una alternativa de diseño específica)?	0			
29. ¿Se ha identificado algún requisito de licencia o restricción de uso que será seguidos por el software?	0			

Semántica del documento

Peso	Indicadores a Evaluar	Eval	(NP)	Cantidad de elementos afectados	Comentarios
Crítico	1. ¿Ha identificado errores ortográficos?	0		2	
Crítico	2. ¿Se entiende claramente lo que se ha especificado en el documento?	0			
	3. ¿El número de página que aparece en el índice coincide con el contenido que se refleja realmente en dicha página?		NP		

CAPÍTULO 2: PROPUESTA DE SOLUCIÓN

4. ¿El total de páginas que aparecen en las reglas de confidencialidad coincide con el total de páginas que tiene el documento?		NP		
---	--	----	--	--

Diseño de casos de prueba.

En la Ingeniería de software, los casos de prueba son un conjunto de condiciones o variables bajo las cuáles el analista determinará si el requisito de una aplicación es parcial o completamente satisfactorio. Con el propósito de comprobar que todos los requisitos de una aplicación son revisados, debe haber al menos un caso de prueba para cada requisito a menos que un requisito tenga requisitos secundarios, de ser así, cada requisito secundario deberá tener por lo menos un caso de prueba (Aristegui, 2010). Esta fue una de las estrategias usada y fueron diseñadas en una plantilla definida por Calisoft que es muy sencilla, esta, está compuesta tres secciones como son:

- **Presentación:** Se especifican el nombre del proyecto, del producto, del caso de prueba y además presenta una tabla de control de versiones, en donde se registra: la fecha, versión, descripción y autor, como se muestra en la figura.

CAPÍTULO 2: PROPUESTA DE SOLUCIÓN



DISEÑO DE CASOS DE PRUEBA BASADOS EN REQUISITOS V2.0

Proyecto Paquete para la gestión de proyectos GESPRO 12.05

CP Crear las iteraciones del proyecto v0.1

Control de Versiones

Fecha	Versión	Descripción	Autor
30 de abril	0.1	Creación del CP	Silvia Torres Arrastre

Figura 5. Sección Presentación del diseño de Casos de Prueba.

- **Casos de Prueba:** Se especifican las condiciones de ejecución, que se refiere a las condiciones que se deben existir para que se realice el requisito, además se especifica el requisito a probar y se realiza el caso de prueba correspondiente al requisito especificado, a partir de una tabla que cuenta con los siguientes campos:
- ✓ Escenario: se especifica el número y nombre del mismo.
 - ✓ Descripción: descripción del escenario de prueba.
 - ✓ Variables: son los campos con que cuenta el requisito.
 - ✓ Respuesta del sistema: se describe la respuesta esperada del sistema.
 - ✓ Flujo central: pasos a desarrollar para probar la funcionalidad que se indicó.

La figura muestra un ejemplo de ello.

CAPÍTULO 2: PROPUESTA DE SOLUCIÓN

Condiciones de ejecución									
El usuario debe estar autenticado en la aplicación.									
El usuario debe ser administrador de la aplicación.									
SC CP Crear las iteraciones del proyecto									
Escenario	Descripción	Nombre	Descripción	Estado	Página Wiki	Fecha	Compartir	Respuesta del sistema	Flujo central
EC 1.1 Crear las iteraciones del proyecto de forma correcta	El usuario debe llenar los datos para crear las iteraciones del proyecto de forma correcta	V Iteración 1	NA	NA	NA	NA	NA	El sistema muestra el mensaje "Creación correcta"	1. Acceder al sistema. 2. Pulsar "Administracion" en la parte superior. 3. Pulsar "Proyecto" en la parte Izquierda. 4. Seleccionar el proyecto 5. Seleccionar la pestaña "Fases"
EC 1.2 Crear las iteraciones del proyecto de forma incorrecta	En caso que el usuario debe llene los datos de forma incorrecta	I	NA	NA	NA	NA	NA	El sistema muestra el mensaje "Nombre no puede estar en blanco"	1. Acceder al sistema. 2. Pulsar "Administracion" en la parte superior. 3. Pulsar "Proyecto" en la parte Izquierda. 4. Seleccionar el proyecto 5. Seleccionar la pestaña "Fases"

Figura 6. Sección Casos de Prueba.

➤ **Variables:** Donde se registra una tabla con los siguientes campos:

- ✓ Número: se especifica el número de las variables con que cuenta el caso de prueba.
- ✓ Nombre de campo: se especifica el nombre de campo de entrada.
- ✓ Clasificación: se clasifica según el componente de diseño utilizado
- ✓ Valor nulo: se especifica si el campo puede ser nulo o no.
- ✓ Descripción: se especifica una breve descripción de los datos que deben introducirse.

La figura muestra un ejemplo de ello.

Descripción de las variables.				
No	Nombre de campo	Clasificación	Valor Nulo	Descripción
1	Nombre	Campo de texto	NO	Nombre de la iteración
2	Descripción	Campo de texto	SI	Breve descripción de la fase
3	Estados	Lista Desplegable	SI	El usuario debe definir el estado que tendrá la iteración, como son abierto, bloqueado y cerrado
4	Fecha	Lista Desplegable	SI	Fecha de creación de la iteración
5	Compartir	Lista Desplegable	SI	El usuario selecciona con quien desea compartir la iteración

Figura 7. Sección Variables del Diseño de Casos de Prueba.

CAPÍTULO 2: PROPUESTA DE SOLUCIÓN

Conclusiones parciales del Capítulo.

1. Se realizó una propuesta de solución, la cual describe el proceso de la Ingeniería de Requisitos, enfocadas en cada una de las etapas definidas, lo que permitió entender con mayor facilidad las actividades que se realizarían para darle solución al problema.
2. Se realizó el flujo de trabajo correspondiente a esta propuesta lo que posibilitó un mayor entendimiento de la propuesta de solución.
3. Se describió cada una de las etapas de la Ingeniería de Requisitos lo que facilitó concluir con el desarrollo de la Ingeniería de Requisitos para el módulo Administración y Gestión de Usuarios.

CAPÍTULO 3: VALIDACIÓN DE RESULTADOS

CAPÍTULO 3: VALIDACIÓN DE RESULTADOS

Introducción

En este capítulo se realizará la validación de los resultados obtenidos a partir de lo realizado, para ello se identifican el total de requisitos identificados, tanto funcionales como no funcionales. Con el objetivo de conocer el alcance competitivo y comercial del sistema en cuanto al módulo Administración y Gestión de Usuarios se realiza una comparación de este con otras herramientas con el objetivo de identificar fortalezas y debilidades. Con el objetivo de visualizar los resultados obtenidos se realiza el análisis de las listas de chequeo y del diseño de casos de prueba y finalmente se analiza la evolución y el impacto de este módulo en la comunidad.

3.1 Requisitos de Software.

Es muy importante destacar que los requisitos deben estar redactados de la manera más clara y concisa posible para evitar malas interpretaciones entre las partes involucradas en el sistema. Los requisitos se agrupan en dos formas como son los requisitos Funcionales y los Requisitos no Funcionales.

3.1.1 Requisitos Funcionales.

Entre los módulos Administración y Gestión de Usuarios fueron identificadas un total de 73 Requisitos Funcionales. Como se muestra a continuación:

Requisitos Funcionales del sub Módulo Proyectos	
RF.01 Crear un nuevo proyecto	RF.11 Eliminar las iteraciones del proyecto
RF.02 Seleccionar los módulos del proyecto	RF.12 Modificar las iteraciones del proyecto
RF.03 Seleccionar las peticiones del proyecto	RF.13 Relacionar iteraciones con las fases del proyecto
RF.04 Visualización de Proyectos	RF.14 Ver resumen de las tareas a ejecutar en el proyecto
RF.05 Modificar información del proyecto	RF.15 Listar proyectos
RF.06 Gestionar miembros del proyecto	RF.16 Filtrar listado de proyectos
RF.07 Crear las fases del proyecto	RF.17 Buscar proyectos
RF.08 Eliminar las fases del proyecto	RF.18 Eliminar proyectos
RF.09 Modificar las fases del proyecto	RF.19 Copiar proyectos
RF.10 Crear las iteraciones del proyecto	RF.20 Archivar proyectos

Requisitos Funcionales del sub Módulo Usuarios	
RF.01 Crear miembros de la organización, registrarse en	RF.04 Salir de la aplicación

CAPÍTULO 3: VALIDACIÓN DE RESULTADOS

la aplicación	
RF.02 Guardar los datos especificados por el usuario	RF.05 Buscar usuario
RF.03 Entrar a la aplicación	RF.06 Acceder a los datos personales

Requisitos Funcionales del sub Módulo Grupos

RF. 01 Crear grupos de usuarios	RF.03 Modificar grupo de usuario
RF.02 Listar grupos de usuarios	RF.04 Eliminar grupo de usuarios

Requisitos Funcionales del sub Módulo Perfiles y Permisos

RF. 01 Creación de perfiles y permisos	RF.03 Ordenar listado de perfiles	RF.05 Eliminar perfil
RF. 02 Otorgar permisos a los perfiles	RF.04 Listar perfiles	

Requisitos Funcionales del sub Módulo Tipos de Peticiones

RF.01 Crear tipos de peticiones	RF.03 Ordenar listado de tipos de peticiones
RF.02 Listar tipos de peticiones	RF.04 Eliminar tipo de petición

Requisitos Funcionales del sub Módulo Estado de las Peticiones

RF.01 Definir estados de las peticiones	RF.04 Modificar estado de peticiones
RF.02 Listar estados de las peticiones	RF.05 Eliminar estado de peticiones
RF.03 Ordenar listado de estados de peticiones	

Requisitos Funcionales del sub Módulo Flujos de Trabajo

RF. 01 Configurar flujos de trabajo	RF.02 Copiar flujo de trabajo	RF.03 Mostrar resumen
-------------------------------------	-------------------------------	-----------------------

Requisitos Funcionales del sub Módulo Campos Personalizados

RF.01 Definir campos personalizados	RF.04 Eliminar campos
RF.02 Listar campos	RF.05 Agrupar por categorías
RF.03 Ordenar campos	

Requisitos Funcionales del sub Módulo Listas de Valores

RF.01 Ordenar listas de valores	RF.02 Eliminar valores de las listas
---------------------------------	--------------------------------------

Requisitos Funcionales del sub Módulo Configurar el sistema

RF.01 Configurar el sistema

Requisitos Funcionales del sub Módulo Autenticación LDAP

RF.01 Nuevo modo de autenticación	RF. Modificar modo de autenticación
-----------------------------------	-------------------------------------

CAPÍTULO 3: VALIDACIÓN DE RESULTADOS

RF.02 Probar modo de autenticación	RF. Eliminar modo de autenticación
RF.03 Listar modo de autenticación	

Requisitos Funcionales del sub Módulo Configuración de Despliegue

RF.01 Configuración de Despliegue

Requisitos Funcionales del sub Módulo Datos del Centro

RF.01 Crear nuevo departamento	RF.02 Listar Datos del centro	RF.03 Editar datos del centro
--------------------------------	-------------------------------	-------------------------------

Requisitos Funcionales del sub Módulo Sincronización pre-proyecto

RF.01 Crear nuevo origen de datos	RF.02 Editar origen de datos	RF.03 Eliminar origen de datos
-----------------------------------	------------------------------	--------------------------------

Requisitos Funcionales del sub Módulo Importar Recursos

RF.01 Importar recursos

Requisitos Funcionales del sub Módulo Extensiones

RF.01 Configurar Extensiones	RF.02 Listar Extensiones
------------------------------	--------------------------

Requisitos Funcionales del sub Módulo Información

RF.01 Mostrar información

Para más detalle ver documento de Especificación de Requisitos en el Expediente de Proyecto de GESPRO/1.ingeniería/1.1 requisitos.

3.1.2 Requisitos no Funcionales.

Fueron definidos entre los requisitos de usabilidad, seguridad, eficiencia, soporte, restricciones del diseño, de software, de hardware, de interfaz, de licencia y finalmente los Legales, de Derecho de Autor un total de 45 requisitos como se muestra a continuación:

Requisitos No funcionales de Usabilidad

RNF1	Facilidad de uso por parte de los usuarios: el sistema debe presentar una interfaz amigable que permita la fácil interacción con el mismo y llegar de manera rápida y efectiva a la información buscada. Debe, además, ser una interfaz de manejo cómodo que posibilite a los usuarios sin experiencia una rápida adaptación.
------	---

CAPÍTULO 3: VALIDACIÓN DE RESULTADOS

RNF2	Especificación de la terminología utilizada: el sistema debe adaptarse al lenguaje y términos utilizados por los usuarios en la rama abordada con vista a una mayor comprensión por parte del cliente de la herramienta de trabajo.
RNF3	Potencialidades de capacitación orientadas a interfaces intuitivas, lo que enaltece la posibilidad de que el usuario aprenda mediante el uso y explotación de la herramienta.
RNF4	Menús: el sistema debe presentar una serie de menús tanto laterales como en barra de iconos flotantes que permitan el acceso rápido a la información por parte de los usuarios, lo que aprovecha las potencialidades de estas estructuras.
Requisitos No funcionales de Seguridad	
RNF5	La seguridad está a nivel de roles, con el fin de mantener la integridad de los datos en función del acceso de cada uno de ellos, trayendo consigo además la protección de la información.
RNF6	Políticas de seguridad por usuarios y roles: el sistema debe contar con un grupo de políticas de accesibilidad a las diferentes funcionalidades del mismo en dependencia del nivel de autorización que presente un usuario determinado.
Requisitos No funcionales de Eficiencia	
RNF7	El sistema debe soportar un tiempo de respuesta menor o igual a 5 segundos.
RNF8	El sistema debe soportar una conexión simultánea de más de 3000 usuarios.
Requisitos No funcionales de Soporte	
RNF9	El sistema brinda como apoyo una Ayuda contextual en la cual se refleja detalladamente la explicación de cada una de las pantallas con sus respectivas funcionalidades.
Requisitos No funcionales de Restricciones de diseño	
RNF10	IDE de desarrollo: NetBeans 6.9.1
RNF11	Sistema Gestor de BD: PostgreSQL 9.0
RNF12	Lenguaje de programación: Ruby on Rails, XML
RNF13	Navegador Web: Mozilla Firefox 6 o superior.
Requisitos No funcionales de Hardware (Servidor de aplicación)	
RNF14	Cantidad: 3
RNF15	CPU : 4 x 2.33 GHz (Intel Xeon 5140 Core2 2.33 GHz)
RNF16	RAM: 8 Gb
RNF17	HDD: 250 Gb RAID 5

CAPÍTULO 3: VALIDACIÓN DE RESULTADOS

RNF18	LAN: 2 x NIC (1 Gbit)
RNF19	SAN: -
RNF20	Fuentes de alimentación: 2 x 800W
Requisitos No funcionales de Hardware (Servidor de almacenamiento)	
RNF21	Cantidad: 2
RNF22	CPU : 4 x 2.33 GHz (Intel Xeon 5140 Core2 2.33 GHz)
RNF23	RAM: 8 Gb
RNF24	HDD: 250 Gb RAID 5
RNF25	LAN: 2 x NIC (1 Gbit)
RNF26	SAN: 1.5 Tb
RNF27	Fuentes de Alimentación: 2 x 800W
Requisitos No funcionales de Software (Servidor de Aplicación)	
RNF28	Sistema operativo: Ubuntu server 10.04, Debian 4 GNU/Linux o superior
RNF29	Servidor de aplicación: Apache2
RNF30	Módulos básicos para el despliegue de Redmine: Ruby1.8, ri1.8, libpgsql-ruby1.8, ruby1.8-dev, librmagick-ruby1.8, libopenssl-ruby1.8, build-essential, apache2-threaded-dev, rake, libxslt-dev, libxml2-dev
RNF31	Aplicación base: Redmine 1.2
RNF32	Framework de Desarrollo: Ruby on Rails
RNF33	Paquete de PHP: php5, libapache2-mod-php5, php5-cli,php5-gd, Php5-mysql, php5-pgsql, php5-sqlite, php5-xsl.
RNF34	Componentes de PATDSI-Generador de Reportes: <ul style="list-style-type: none">• RServer• Chart Server• librería rjson
RNF35	Control de versiones: Subversion 2.6
RNF36	Gestor documental: eXcriba 2.0/Alfresco

CAPÍTULO 3: VALIDACIÓN DE RESULTADOS

Requisitos No funcionales de Software (Servidor de almacenamiento)

RNF37 Gestor de Base de Datos: PostgreSQL versión 9.0

Requisitos No funcionales de Requisitos para la documentación de usuarios en línea

RNF38 Manual de usuario: el sistema debe tener un manual de usuario, permitiendo con ello un correcto uso de sus funcionalidades y brindarle al usuario una mayor experiencia del trabajo con el mismo.

RNF39 Documentación actualizada del grupo de desarrollo: se precisa que la documentación del sistema esté actualizada en todos los aspectos, fases de trabajo y ciclos de desarrollo del mismo, permitiendo con ello un respaldo tanto ingenieril como legal del desarrollo de dicho sistema.

Requisitos No funcionales de Componentes comprados

RNF40 Para el desarrollo del sistema no fue necesario comprar ningún componente.

Requisitos No funcionales de Interfaz

RNF41 Interfaz Web: la interfaz es sencilla con colores suaves a la vista y sin cúmulo de imágenes u objetos que distraigan al cliente del objetivo.

Requisitos No funcionales de Interfaces de Hardware

RNF42 La comunicación entre el servidor de aplicaciones y la base de datos se lleva a través del protocolo de conexión TCP/IP.

RNF43 La comunicación entre el cliente y el servidor de aplicaciones se lleva a través del protocolo HTTP.

Requisitos No funcionales de Requisitos de licencia

RNF44 No hay ninguna restricción de uso para el sistema.

Requisitos No funcionales de Legales, de Derecho de Autor y otros

RNF45 El sistema debe ser sometido a un análisis legal por parte de los abogados y personal autorizado con vistas a declarar su autenticidad y evitar restricciones legales para su uso y comercialización; así mismo se debe proceder a una evaluación y certificación por parte del cliente del producto.

Para más detalle ver documento de Especificación de Requisitos en el Expediente de Proyecto de GESPRO/1.ingeniería/1.1 requisitos.

CAPÍTULO 3: VALIDACIÓN DE RESULTADOS

3.2 Comparación de GESPRO con otros sistemas.

En este epígrafe se pretende hacer una comparación de GESPRO con otros sistemas. Según el Diccionario de la “Real Academia Española”, comparar es fijar la atención de dos o más objetos para descubrir sus relaciones, esta se establece cuando las partes son similares entre sí, la misma puede realizarse respecto a diversas situaciones que implica que dos o más elementos comparten algunas de sus características, volviéndose similares o parecidos entre sí, esta tiene como objetivo fundamental descubrir semejanzas y diferencias (2010).

A continuación se muestra la comparación de GESPRO con otras herramienta como son Trac, Jira, DotProject (DP), Redmine, GForge, Team Foundation Server (TFS), Primavera, estas herramientas se seleccionaron por ser las más fuertes en cuanto a Administración y Gestión de Usuarios, por un lado se especifican las funcionalidades a resaltar o que no tiene GESPRO, y por otro lado las herramientas entre las que se posiciona GESPRO en la primera columna con el objetivo de observar lo que posee o no con respecto a las demás.

Funcionalidades	Herramientas Gestoras de Proyectos							
	GESPRO	Trac	Jira	DP	Redmine	GForce	TFS	Primavera
Autenticación LDAP.	x	x	x					x
Los usuarios no autenticados tienen acceso al sistema.	x	x				x		x
Mostrar todos los usuarios que están conectados.		x						
Notificaciones por e-mail.	x		x	x		x		
Configurar Sistema.	x		x		x	x	x	
Gestionar Apariencia del Sistema.	x		x					
Usar filtros para restringir la búsqueda de usuarios y proyectos.	x		x	x	x			
Asignar permisos a los perfiles.	x	x	x	x			x	x
Gestionar tipo de peticiones.	x							

CAPÍTULO 3: VALIDACIÓN DE RESULTADOS

Importar tareas y recursos.	x	x	x		x	x		x
Gestionar grupos de usuario.	x			x			x	x
Definir tipos de administradores.			x				x	
Estado de las Peticiones.	x							
Gestionar Estadísticas en forma de reportes.	x							
Asignación de un usuario a un grupo.	x		x	x		x	x	x
Asignar un usuario a un proyecto.	x		x				x	x
Gestionar Campos Personalizados.	x							
Activar permisos.	x	x					x	x
Mostrar listado de plugins instalados.	x	x						
Gestionar Listas de Valores.	x							
Desactivar o Bloquear usuario.	x		x					
Ayuda para proyectos.	x	x				x	x	x
Soporta varios idiomas.	x			x		x		
Entradas recientes al sistema.						x		
Gestionar Flujos de trabajo.	x							
Gestionar Utilidades del sitio						x		
Modificar el número máximo de elementos a mostrar en las listas	x							x

3.2.1 Fortalezas y debilidades identificadas del Sistema.

CAPÍTULO 3: VALIDACIÓN DE RESULTADOS

Los resultados obtenidos al aplicar la comparación que se muestra en la tabla anterior evidencian que el sistema GESPRO, posee muchas funcionalidades que los otros sistemas no tienen, entre las funcionalidades que cabe resaltar tenemos: autenticación LDAP, gestión de tipo de peticiones, estado de las peticiones, gestión de flujos de trabajo, campos personalizados, lista de valores, soporta varios idiomas y ayuda para proyectos, todas estas funcionalidades lo hacen más fuerte y robusto, posicionándolo a la altura de las grandes competencias.

Sin embargo, es válido reconocer que se le podrían agregar algunas funcionalidades, que no son imprescindibles para el trabajo, pero que se pueden tener en cuenta para una mayor gestión y facilidad de uso, tales como:

- ✓ Mostrar los usuarios que están conectados en tiempo real: esta funcionalidad permitiría la visualización de los usuarios que están conectados en el sistema en un instante de tiempo dado, para ello se realizaría un enlace que diga usuarios conectados o se mostraría en alguna parte en la interfaz.
- ✓ Mantenimiento de suscripciones: esta funcionalidad permitiría realizar un mantenimiento en cuanto a los registros de información de todo aquel usuario que hayan sido baja del sistema o información que ya no se necesite.
- ✓ Gestionar entradas recientes al sistema: esta funcionalidad permitiría mostrar al administrador del sistema todas las entradas recientes por cualquier usuario, mostrando la fecha de acceso, el nombre de usuario y el IP desde el cual accedió al sistema.
- ✓ Definir tipos de administradores: esta funcionalidad sería muy útil, ya que no existiría un solo administrador sino que este delegaría permisos a otros tipos de administradores dependiendo del área que se trate, pues en caso que algo sucediera con el mismo el sistema no presente ningún problema.

3.3 Análisis del resultado de las Listas de Chequeo.

En el capítulo anterior, se seleccionó como una de las técnicas de validación, las listas de chequeo, con el objetivo de demostrar que el levantamiento de requisitos fue satisfactorio. Esta fue aplicada en 2 iteraciones como se explican a continuación.

En la primera iteración de un total de 35 preguntas, se detectaron 5 indicadores que no proceden (NP), 8 no conformidades (NC) y 22 indicadores evaluadas correctamente. Las no conformidades encontradas fueron las siguientes:

- ✓ No todos los requisitos estaban redactados de forma simple y clara.

CAPÍTULO 3: VALIDACIÓN DE RESULTADOS

- ✓ Algunos requisitos requerían de especificarse con más detalle.
- ✓ En el caso de un requisito, debe especificarse con menos detalle ya que estaba muy amplia su descripción.
- ✓ No había un campo con la opción variables de salida.
- ✓ No se encontraban especificados los requisitos de software y de hardware.
- ✓ No se encontraban las restricciones de implementación.
- ✓ No estaban bien enumerados aquellos requisitos que se derivan de otros.
- ✓ Fueron identificadas la ausencia de dos tildes en el caso de dos requisitos.

Los indicadores de No Proceden dependen de la integración de este módulo con los demás del sistema.

En una segunda iteración, estas no conformidades, fueron corregidas satisfactoriamente arrojando como resultado final 5 indicadores de NP y 30 evaluadas correctamente. La siguiente figura muestra en forma de gráfica lo ante expuesto.

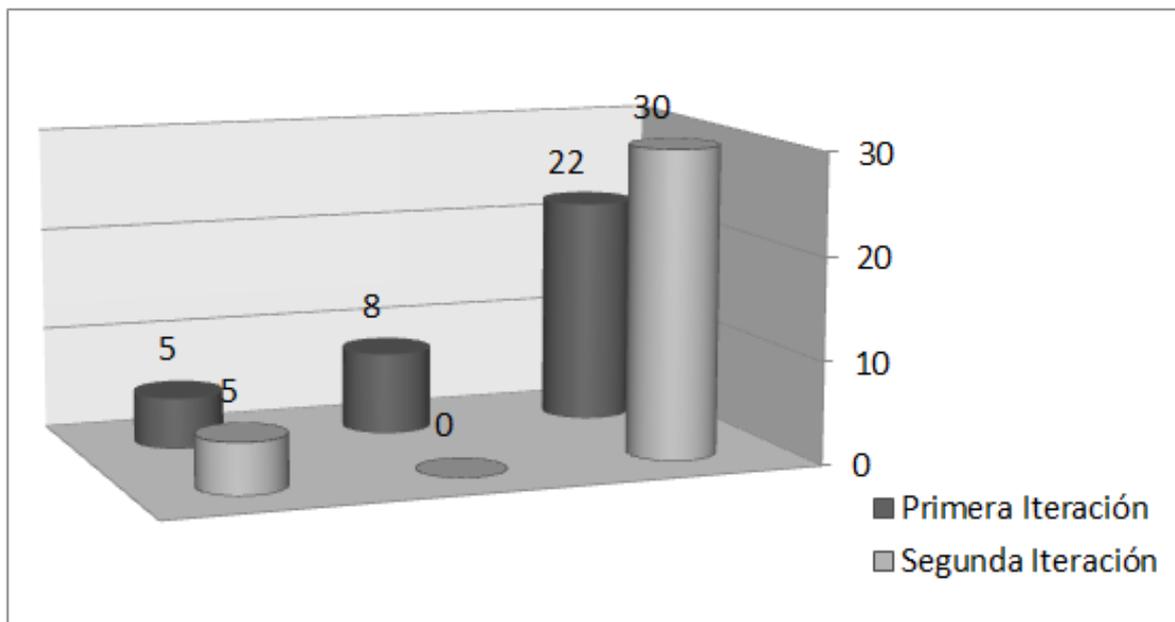


Figura 8. Iteraciones de las Listas de Chequeo.

El resto de la lista de comprobación aplicada se puede consultar en el documento Listas de Chequeo en el Expediente del Proyecto de GESPRO/1.ingeniería/1.1 requisitos.

3.4 Resultado de los Casos de Prueba.

El diseño de casos de prueba, fue otra de las técnicas de validación definidas a usar en el capítulo anterior, se debe realizar al menos un caso de prueba para cada requisito, a menos que un requisito

CAPÍTULO 3: VALIDACIÓN DE RESULTADOS

tenga requisitos secundarios, de ser así, cada requisito secundario deberá tener por lo menos un caso de prueba. Entre los módulos Administración y Gestión de Usuarios se realizaron 73 casos de prueba. A continuación se muestra una gráfica que relaciona la cantidad de casos de prueba generados a partir de los requisitos funcionales.

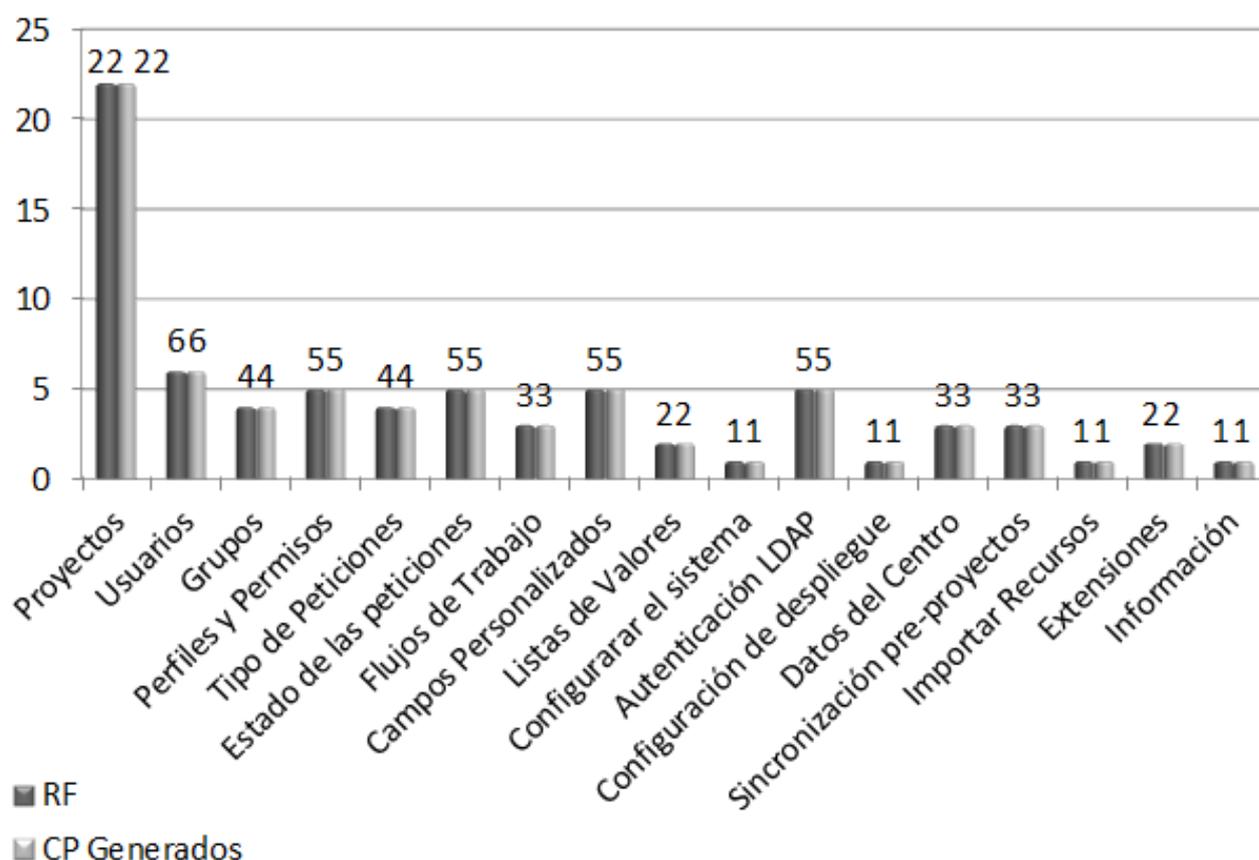


Figura 9. Gráfica de Casos de Prueba con Requisitos Funcionales

3.5 Evolución de los módulos Administración y Gestión de Usuarios del sistema GESPRO.

Como bien se expone en el trabajo los centros de desarrollo de software surgen en el 2008 y con ellos la necesidad de una herramienta que gestionara todos sus procesos, pues sin ella, era menos organizado y se hacía más difícil el trabajo. A continuación se mencionaran los principales problemas que existían y que se enmarcan en estos módulos.

- ✓ La información de los integrantes del proyecto era recogida por el líder personalmente, este encuestaba a cada miembro y se quedaba con los datos que realmente necesitaba.

CAPÍTULO 3: VALIDACIÓN DE RESULTADOS

- ✓ Inexistencia de roles en el equipo de trabajo, por lo que todos los miembros del equipo debían asumir tareas de cualquier índole.
- ✓ La comunicación entre los integrantes era personal y directa.
- ✓ El cumplimiento de las tareas asignadas eran revisadas a través de chequeos personales y semanales.
- ✓ Con el objetivo de comprobar que la calidad era adecuada se probaba con los instaladores del producto cada cierto tiempo.
- ✓ No era sencillo definir el avance del proyecto, este se definía a partir de estimaciones que se establecían a las tareas, de acuerdo a parámetros relacionados con la complejidad de las mismas.
- ✓ Los cronogramas variaban considerablemente, pues se tenía en cuenta la posible desviación de acuerdo al fondo de tiempo de los miembros del proyecto y las demás actividades que podían surgir.

Todos estos problemas se le fueron dando solución poco a poco. La primera acción en este sentido fue la decisión de usar el Redmine en mayo del 2010, debido a la gran cantidad de funcionalidades que facilitaba, se tomó como propuesta de herramienta para la gestión de los proyectos informáticos en la universidad, teniendo un entorno para cada centro, que aunque no estaba en su versión final, se encontraba funcional y en proceso de desarrollo de mejoras, que incluían integración con otras herramientas para la gestión documental como Alfresco y otras. La interacción con esta herramienta era a través de la Web y de cierto modo solucionaba gran parte de los problemas que estaban latentes.

A partir de aquí, comenzó a trabajarse en función de esas mejoras, que no son más que nuevas funcionalidades de acuerdo a la gestión de proyectos en la universidad. Actualmente el producto se llama GESPRO y se trabaja en su versión 12.05. Esta versión es lo suficientemente apropiada y completa, pues con ella se logra una realización de las actividades de manera racional, proporciona información útil para el análisis y la toma de decisiones, es organizada y sobre todo centralizada.

En el capítulo anterior, se realizó una comparación de esta herramienta con otros sistemas con el objetivo de tener una idea clara del alcance competitivo y comercial del sistema, en la que se identificaron nuevas funcionalidades, que como bien se expone, no son imprescindibles para el trabajo básico, pero se pueden tener en cuenta para una mayor gestión y facilidad de uso.

También se realizó una comparación del módulos Administración y Gestión de Usuarios con versiones anteriores, lo que evidenció la evolución de estos en cuanto a número de funcionalidades y número de módulos, arrojando como resultado que hay una diferencia de 15 funcionalidades y 6 módulos como se

CAPÍTULO 3: VALIDACIÓN DE RESULTADOS

muestra en la figura.

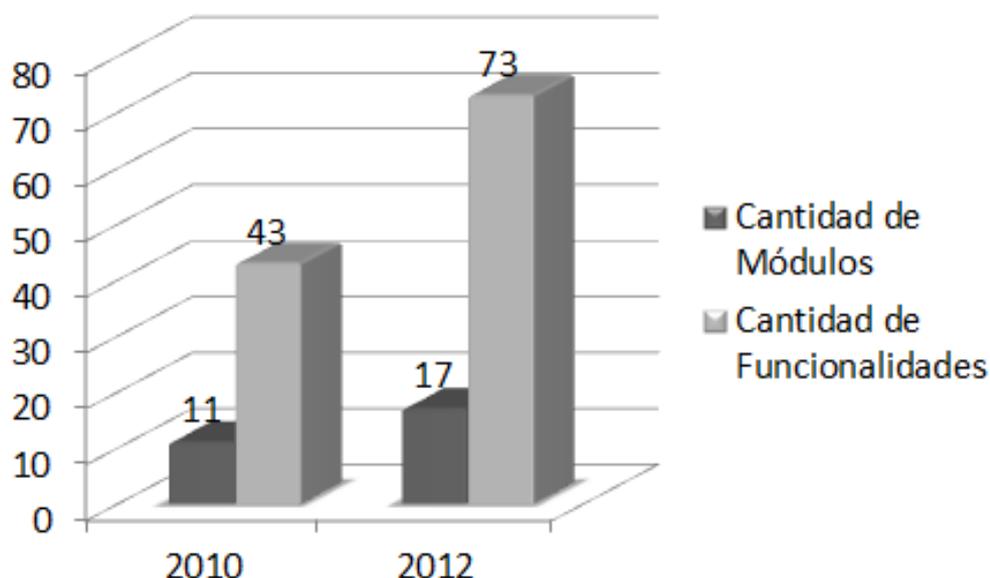


Figura 10. Evolución del módulo Administración en cuanto a funcionalidades y módulos.

A grandes rasgos se evidencia la evolución de este módulo desde su creación hasta la actualidad, con el objetivo de seguir trabajando en este sentido, a través de la comparación de GESPRO con otros sistemas se identificaron nuevas funcionalidades, que de ser implementadas contribuiría al aumento de las funcionalidades, a la mejora de los procesos que se llevan a cabo y por consiguiente a la calidad del sistema. De modo general esta herramienta se encuentra a la altura de los principales competidores a nivel mundial.

3.6 Análisis del impacto social del sistema GESPRO.

Cuando se habla de impacto se refiere a una impresión o efecto muy intenso, dejados en alguien o en algo por cualquier acción o suceso. El impacto de un proyecto es la magnitud cuantitativa y cualitativa del cambio de la población sobre el problema (Atom, 2009).

El sistema GESPRO como bien se expone en los capítulos anteriores surge por la necesidad de una herramienta que cubriera todos los problemas que existían en aquel entonces en cuanto a la producción, problemas que de cierto modo influían en el resultado y calidad de los productos software desarrollados. La implantación de este sistema causó un impacto social, pues de cierto modo, dio solución y ayudó en la realización de la mayoría de las actividades, haciéndolas más fácil para los

CAPÍTULO 3: VALIDACIÓN DE RESULTADOS

integrantes.

Para obtener información al respecto se realizaron entrevistas a trabajadores de diferentes centros, en los que reflejaba complacencia con el sistema al expresar “con GESPRO es más fácil el trabajo en el proyecto”, pues ahora hay roles definidos y de este modo las tareas están precisadas y dirigidas a personas capacitadas, el cumplimiento de las tareas es revisado por el que la asigna desde un ordenador sin la necesidad de reunirse personalmente, a menos que así lo requiera. También el sistema permite gestionar el tiempo, a través de la definición y secuencia de las actividades y estimación de la duración de estas, proporciona facilidades para la gestión del alcance del proyecto, en cuanto a la comunicación define los procesos requeridos para asegurar la generación, recopilación, diseminación, almacenamiento y disposición final de la información del proyecto, facilitando la planificación de las comunicaciones, distribución de la información, reporte del rendimiento, gestión de acuerdos y gestión de compromisos, y posee procesos para la gestión de riesgos como la identificación, análisis y respuesta a los riesgo del proyecto, a través del y monitoreo y control de los riesgos.

Estas son algunas de las características que posee GESPRO y que favorecieron de forma positiva al impacto en la comunidad, pues contribuyó al ahorro de tiempo, el análisis y la toma de decisiones de la manera más rápida, fácil y racional posible.

Con el objetivo de llegar hasta las personas que utilizan la herramienta y con el objetivo de conocer la aceptación que ha tenido la misma en estos momentos se desarrolló una encuesta dirigida a:

1. Dirección General de la Producción.
2. Personal de los Centros de Producción (Jefes de Proyecto, Jefes de Línea, Analistas, Directores de Centro).
3. Empresa Comercializadora ALBET.
4. Calisoft.

Esta se realizó con el objetivo de conocer sus opiniones con respecto a problemas o carencias existentes en el módulo Administración y Gestión de Usuarios, así como aspectos positivos y la propuesta de mejoras en caso que consideraran conveniente.

A pesar de la limitante que posee el módulo Administración de ser solo usado por usuarios con roles o permisos de administración, se pudo concluir que la aplicación de este módulo ha teniendo resultados satisfactorios, en la mayoría de los casos, dándole una prioridad máxima. De forma general el módulo Administración y Gestión de Usuarios satisface los objetivos para lo que fue creado. Para visualizar la encuesta consultar Anexo 1.

La siguiente figura muestra de modo gráfico los resultados de la encuesta.

CAPÍTULO 3: VALIDACIÓN DE RESULTADOS

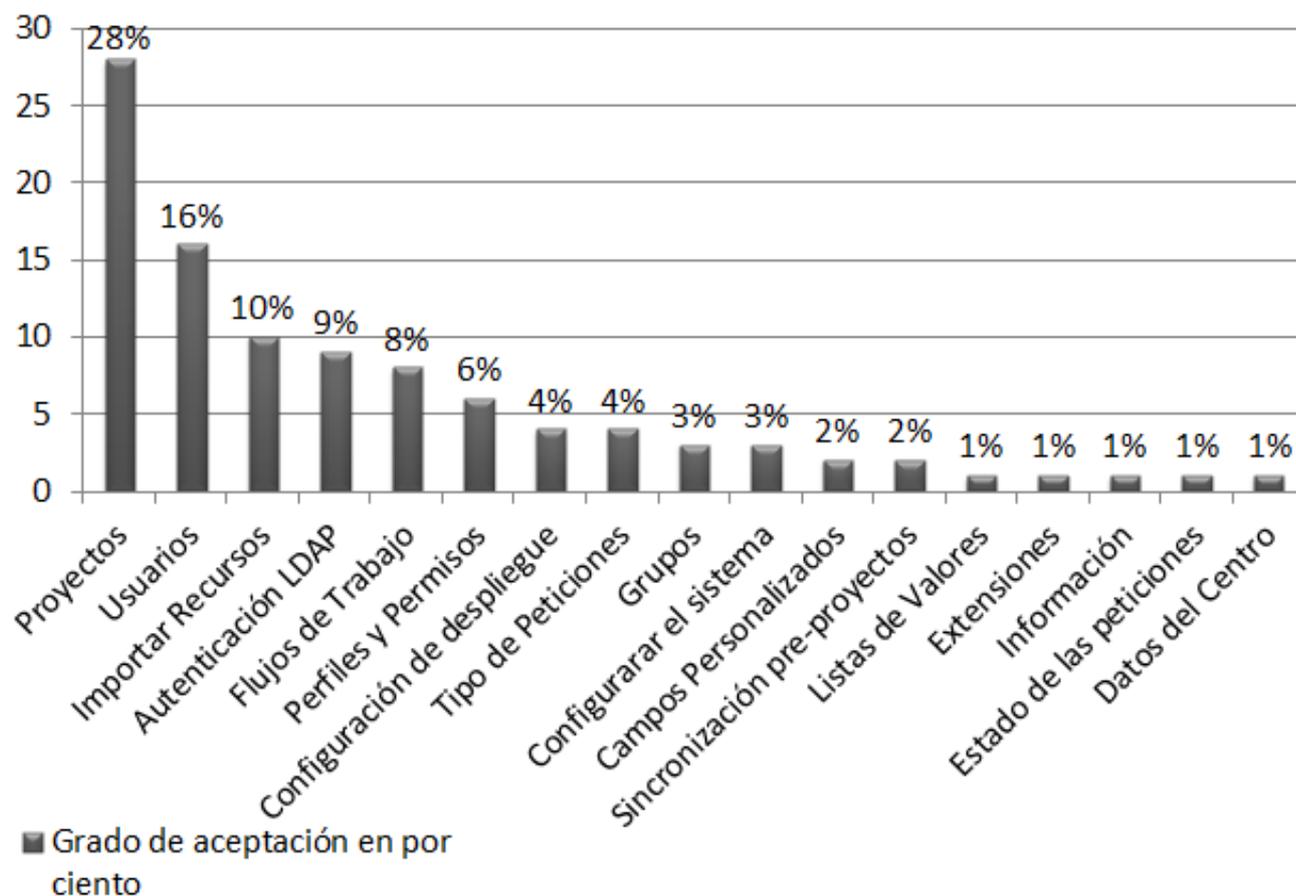


Figura 11. Grado de aceptación de sub módulos del módulo Administración en la comunidad.

CAPÍTULO 3: VALIDACIÓN DE RESULTADOS

Conclusiones parciales del Capítulo.

1. Se identificaron requisitos funcionales y no funcionales del sistema obteniéndose un total de 73 requisitos funcionales y 45 requisitos no funcionales.
2. Se analizó la aplicación de las listas de chequeo, la cual en 2 iteraciones demostró que el documento especificación de requisitos de software está correctamente redactado.
3. Se analizó el diseño de casos de prueba lo que permitió la correcta realización de estos, al obtenerse como resultado un caso de prueba por cada requisito, para un total de 73 casos de prueba.
4. La comparación de GESPRO con otras herramientas de gestión de proyectos permitió la identificación de fortalezas y debilidades, convirtiendo las debilidades en futuras funcionalidades.
5. Se analizó la evolución del módulo Administración y Gestión de Usuarios lo que demostró el progreso del mismo desde su creación hasta la actualidad.
6. Se analizó el impacto del módulo Administración y Gestión de Usuarios a partir de encuestas, lo que permitió la identificación de los submódulos más aceptados por la comunidad así como el grado de importancia atribuidos a los mismos.

CONCLUSIONES GENERALES

CONCLUSIONES GENERALES

Luego de la realización del presente trabajo se llega a las siguientes conclusiones:

1. Se identificaron requisitos funcionales y no funcionales del sistema a partir del estudio de las principales características del sistema GESPRO, generándose el documento Especificación de Requisitos de Software de los módulos Administración y Gestión de Usuarios.
2. Se realizó un análisis comparativo de GESPRO con otras herramientas de gestión en cuanto a actividades relacionadas con los módulos Administración y Gestión de Usuarios, lo que permitió la identificación de fortalezas y debilidades convirtiéndolas en futuras funcionalidades.
3. Con la aplicación de las Listas de Chequeo al documento de Especificación de Requisitos de Software de los módulos Administración y Gestión de Usuarios, se validó la calidad del mismo en correspondencia con los aspectos establecidos por Calisoft.
4. Se diseñaron los casos de prueba para los diferentes escenarios de los módulos con el fin de probar su correcto funcionamiento.

RECOMENDACIONES

Después de finalizar la presente investigación se sugieren las siguientes recomendaciones:

1. Implementar las nuevas funcionalidades de los módulos Administración y Gestión de Usuarios para poder ser liberados en la próxima versión del sistema.
2. Realizar las pruebas de liberación a partir del diseño de casos de prueba propuestos, para demostrar el correcto funcionamiento de todas las funcionalidades y se pueda liberar el producto de modo satisfactorio.

REFERENCIAS BIBLIOGRÁFICAS

REFERENCIAS BIBLIOGRÁFICAS

- 2012.** Administrar Team Foundation Server. [En línea] Junio de 2011. [Citado el: 5 de marzo de 2012.] <http://msdn.microsoft.com>.
- Alexander, Roberts. 2002.** Gestión de Proyectos. EdinBurgh : s.n., 2002.
- Alfredo, Salas Cristobal. 2003.** Ingeniería Inversa y Reingeniería Aplicadas a Proyectos de Software Desarrollados. Tijuana, Baja California : s.n., 2003.
- Altamiranda, Junior. 2006.** Manual de Usuario Para el Uso de Gforge. Merida : s.n., 2006.
- . **2000.** PLATAFORMA GFORGE. Merida, Venezuela : s.n., 2000.
- . **2006.** REQUERIMIENTOS Y SERVICIOS DE GFORGE. Venezuela : s.n., 2006.
- Arencibia. 2012.** gestion de proyectos con redmine. [En línea] enero de 2012. [Citado el: 15 de marzo de 2012.] <http://www.neotechgw.com/ofertas/gestor-de-proyectos-y-bug-tracking-redmine-101/>.
- Aristegui, José Luis. 2010.** Los casos de prueba en la prueba de software. Chile : s.n., 2010.
- Atom. 2009.** Impacto Social. [En línea] 24 de 9 de 2009. [Citado el: 10 de 5 de 2012.] <http://3a-electroproject-28.blogspot.com/2009/09/impacto-social>.
- Barba, Brunner Silvia. 2001.** Metodología DoRCU para la Ingeniería de Requerimientos. La Habana : s.n., 2001.
- Capuz, Eliseo Gomez y Salvador. 1999.** El proyecto y su dirección y Gestión. Valencia : s.n., 1999.
- Cervantes, Omar Higinio Caballero. 2007.** TECNOLOGÍAS DE INFORMACIÓN Y HERRAMIENTAS PARA LA ADMINISTRACIÓN DE PROYECTOS DE SOFTWARE. Campus Monterrey : s.n., 2007.
- CESLCAM, Centro de Excelencia de Software Libre. 2010.** Análisis de aplicación: Redmine. Castilla, La Mancha : s.n., 2010.
- Chaves, Michael Arias. 2006.** La ingeniería de requerimientos y su importancia en el desarrollo de proyectos de software. Costa Rica : Edición Digital: 26 / 07 / 2007, 2006.
- Compendio de herramientas SW Libre para la calidad del software. Promoción del desarrollo de SW libre en un entorno de calidad y confianza adaptando las metodologías, procesos, modelos de negocio y últimas tecnologías. [En línea]
- Consultoría Certia.** Team Foundation Server. Certia. [En línea] [Citado el: 16 de Febrero de 2012.] <http://www.certia.net/TFS>.
- 2008.** danielvn7. [En línea] marzo de 2008. [Citado el: 29 de enero de 2012.] danielvn7.wordpress.com.
- Edgewall, Software. 2003.** Trac Permisos. [En línea] noviembre de 2003. [Citado el: 12 de marzo de 2012.] <http://trac.edgewall.org/>.
- Enamorado, Jordan Borjas. 2009.** Entrevista a Calisoft. La Habana, UCI : s.n., 2009.

REFERENCIAS BIBLIOGRÁFICAS

- Escalona, María José. 2004.** [En línea] diciembre de 2004. [Citado el: 22 de febrero de 2012.] <http://www.lsi.us.es/docs/informes/LSI-2002-4.pdf>.
- EUDE, Escuela Europea de Dirección y Empresas. 2001.** Máster en Gestión y Dirección de Recursos Humanos. Madrid : s.n., 2001.
- Experiencia, Calidad y Compromiso. Business Consulting Group and Management. [En línea] [Citado el: 14 de 03 de 2012.] http://www.consultorabcg.com/contenido_15.php. 7.
- Gestión de Proyectos. Seguimiento y Control del Proceso. [En línea] [Citado el: 12 de 03 de 2012.] <https://sites.google.com/site/gdpumh/tema-iv-seguimiento-y-control-de-proyectos/4-5-caso-practico>.
- Hernández, Rolando Alfredo. 2005.** Gestión de proyectos para informáticos. La Habana : s.n., 2005.
- 2010.** Herramientas para la Gestión de la Calidad. [En línea] 12 de Julio de 2010. [Citado el: 5 de marzo de 2012.] <http://demo.redmine.org>.
- IEEE, Standard Glossary of. 1990.** Institute of Electrical and Electronics Engineers. New York : s.n., 1990. Standard 610.
- Lacoba, Rocío Navarro. 2000.** Manual de Administración Jira 3.13. 2000.
- López, Esteban Saavedra. 2010.** Trac: Gestión de proyectos de desarrollo de Software. Bolivia : s.n., 2010.
- Macarrón, Pedro. 2011.** El sistema integrado para el control del código fuente y la gestión del ciclo de vida de las aplicaciones. [En línea] 10 de mayo de 2011. [Citado el: 5 de marzo de 2012.] <http://certia.net>.
- Márquez, José Manuel.** ¿Qué es Ingeniería de Requisitos (IR)? [En línea] [Citado el: 14 de 03 de 2012.] <http://danielvn7.wordpress.com/2008/03/27/%C2%BFque-es-ingenieria-de-requisitos-ir/>.
- . **2007.** ¿Qué es Ingeniería de Requisitos? [En línea] 2007. [Citado el: 12 de 3 de 2012.] <http://danielvn7.wordpress.com/2008/03/27/%C2%BFque-es-ingenieria-de-requisitos-ir/>.
- Norambuena, Rodrigo Ramírez. 2009.** Gestión de proyectos de software con Trac. Santa María, Concepción : s.n., 2009.
- Passador, Victor.** Nuevos conceptos en TFS 2010. VEMN Team Blog. [En línea] [Citado el: 16 de Febrero de 2012.] <http://www.vemn.com.ar/Blog/?tag=/Team-Foundation-Server&page=3>.
- Perez, Pedro Piñero. 2010.** PAQUETE PARA LA DIRECCIÓN INTEGRADA DE PROYECTOS GESPRO. La Habana, UCI : s.n., 2010.
- Perez, Pedro Yobanis Piñero. 2010.** MANUAL DE CONFIGURACIÓN DE PAQUETE DE HERRAMIENTAS PARA LA GESTIÓN DE PROYECTOS V 1.0. La Habana, UCI : s.n., 2010.
- Pressman, Roger.** Ingeniería de Software: Un enfoque practico. McGrawHill : VI Edición.
- PRODUCCIÓN, DIRECCIÓN TÉCNICA DE LA. 2011.** ANÁLISIS COMPARATIVO DE GESPRO. La

REFERENCIAS BIBLIOGRÁFICAS

Habana : s.n., 2011.

Proyectos Orquestados: DotProject la batuta ideal. Software Libre Venezuela. [En línea] [Citado el: 18 de Febrero de 2012.] <http://softwarelibrevenezuela.blogspot.es/i2008-12/>.

Quizhpe, Aurelio Capa. 2002.Ingeniería de Requisitos. Ecuador : s.n., 2002.

Sánchez, Héctor Poveda. 2008.Aplicaciones de gestión de proyectos desarrolladas sobre dotProject. Murcia : s.n., 2008.

Scalzone, Alejandra. Team Foundation Server: Un equipo ordenado y eficiente. MSDN Flash, TechNet Flash & Newsletter de Seguridad. [En línea] [Citado el: 16 de Febrero de 2012.] <http://loseditoresdelatam.wordpress.com/2011/11/23/tfs/>.

—. **2011.** Team Foundation Server: Un equipo ordenado y eficiente. [En línea] 23 de 11 de 2011. [Citado el: 12 de 2 de 2012.] <http://loseditoresdelatam.wordpress.com/2011/11/23/tfs/>.

Sistema de manejo y seguimiento de proyectos: Trac. Tecnología, Internet, Software Libre y GNU/Linux. [En línea] [Citado el: 18 de Febrero de 2012.] <http://blog.milmazz.com.ve/archivos/2010/03/16/sistema-de-manejo-y-seguimiento-de-proyectos-trac>.

Sommerville. 2005.Ingeniería de Software. s.l. : Editorial Pearson, 2005.

Team Foundation Server – Reutilizar un workspace creado en otra máquina. Cerebro en la Sombra. [En línea] [Citado el: 16 de Febrero de 2012.] <http://blog.osusnet.com/2010/11/05/team-foundation-server-reutilizar-un-workspace-creado-en-otra-maquina/>.

Team, Abartia. 2007.PROYECTO MANUAL USUARIO DOTPROJECT. Bilbao : s.n., 2007.

Torres, G. 2008. Proyectos Orquestados: DotProject la batuta ideal. [En línea] 4 de 2008. [Citado el: 9 de 12 de 2011.] <http://softwarelibrevenezuela.blogspot.es/i2008-12/>.

Varas, Marcela. 2000.Gestión de Proyectos de Desarrollo de Software. Universidad de Concepción : s.n., 2000.

Yebel Fornaris Licea, Yoanna Fernandez. 2010.Modelamiento del Negocio de Ingeniería de Requisitos del Sistema Integral de Documentación e Información Judicial(SIDIJ). 2010.

ANEXOS

Anexo 1. Encuesta Impacto Social del sistema GESPRO 12.05.

UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS

Encuesta

Impacto Social del sistema GESPRO 12.05

Silvia Torres Arrastre

29/05/2012

Introducción

Para el desarrollo de un proyecto se hace necesario prescindir de una herramienta que permita la gestión del mismo, logrando que su administración y organización sea más fácil y viable.

Esta encuesta tiene como objetivo obtener información para analizar el impacto social del módulo X del sistema para la gestión de proyecto GESPRO.

Datos generales

Marque con una X la categoría a la que usted pertenece:

Dirección General de Producción

Centro de Producción:

Jefe de proyecto

Jefe de línea

Director del centro

Analista

Desarrollador

Administrador de la calidad

Administrador del centro

Formación:

Tutor

Especialista en gestión de proyecto
 Empresa comercializadora Albet
 Calisoft

La tabla que se muestra a continuación deberá completarse de la siguiente forma:

- 1- En la columna "Prioridad" debe establecer con un orden numérico comenzando por 1 el nivel de prioridad de cada submódulo según su criterio. En caso de que usted considere que el submódulo no deba incluirse dentro de la herramienta deje la celda en blanco.

Notas:

- Mientras menor sea el número mayor será su prioridad.
 - En caso que deje alguna celda en blanco, no debe ejecutar los pasos 2 y 3.
- 2- En la columna de "Evaluación Funcional" debe marcar con una X el criterio que describa funcionalmente el submódulo.
 - 3- En cualquier caso que sea seleccionado debe argumentar el motivo de su selección en la columna "Observaciones".

• **Módulo Administración**

Prioridad	Submódulo	Evaluación funcional	Observaciones
	Proyectos	<input type="checkbox"/> Funcional <input type="checkbox"/> Poco funcional <input type="checkbox"/> No cumple objetivo	
	Usuarios	<input type="checkbox"/> Funcional <input type="checkbox"/> Poco funcional <input type="checkbox"/> No cumple objetivo	
	Grupos	<input type="checkbox"/> Funcional <input type="checkbox"/> Poco funcional <input type="checkbox"/> No cumple objetivo	
	Perfiles y permisos	<input type="checkbox"/> Funcional <input type="checkbox"/> Poco funcional <input type="checkbox"/> No cumple objetivo	
	Tipos de peticiones	<input type="checkbox"/> Funcional <input type="checkbox"/> Poco funcional <input type="checkbox"/> No cumple objetivo	
	Estado de las peticiones	<input type="checkbox"/> Funcional <input type="checkbox"/> Poco funcional <input type="checkbox"/> No cumple objetivo	
	Flujo de trabajo	<input type="checkbox"/> Funcional <input type="checkbox"/> Poco funcional <input type="checkbox"/> No cumple objetivo	
	Campos personalizados	<input type="checkbox"/> Funcional <input type="checkbox"/> Poco funcional <input type="checkbox"/> No cumple objetivo	
	Lista de valores	<input type="checkbox"/> Funcional <input type="checkbox"/> Poco funcional <input type="checkbox"/> No cumple objetivo	
	Configuración	<input type="checkbox"/> Funcional <input type="checkbox"/> Poco funcional <input type="checkbox"/> No cumple objetivo	
	Autenticación LDAP	<input type="checkbox"/> Funcional <input type="checkbox"/> Poco funcional <input type="checkbox"/> No cumple objetivo	

	Tipos de reunión	_ Funcional _ Poco funcional cumple objetivo	_ No	
	Configuración de despliegue	_ Funcional _ Poco funcional cumple objetivo	_ No	
	Datos del Centro	_ Funcional _ Poco funcional cumple objetivo	_ No	
	Sincronización pre-proyectos	_ Funcional _ Poco funcional cumple objetivo	_ No	
	Importar Recursos	_ Funcional _ Poco funcional cumple objetivo	_ No	
	Extensiones	_ Funcional _ Poco funcional cumple objetivo	_ No	
	Información	_ Funcional _ Poco funcional cumple objetivo	_ No	