Análisis de Requisitos de un Software de Gestión de Requisitos

Sergio Ripoll Lorca

Resumen—El objetivo de este artículo es la exposición de los resultados de realizar un análisis de requisitos de un software de gestión de requisitos. El estudio está basado en aplicar la metodología de las 3 dimensiones de la ingeniería de requisitos del software, que se basa en la selección de fuentes – stakeholders, documentación existente y el análisis comparativo de otros softwares de gestión disponibles en el mercado -, recopilación de artefactos – escenarios, objetivos y requisitos – al aplicar técnicas de elicitación, análisis documental, la elaboración de un posible prototipo software de gestión y la entrevista a un stakeholder. En la selección de fuentes se revisaron los diferentes manuales y documentos de los diferentes softwares de gestión de requisitos que se compararon durante el estudio. También se realizó una entrevista a un determinado stakeholder experimentado en la utilización de dichos softwares. Todos los resultados del análisis, comparaciones, recolección de requisitos, conclusiones se documentaron en un documento de visión, en un SRS y en unas actas con las características específicas de cada software analizado.

Palabras clave—Documento de visión, escenarios, especificación de requisitos software, gestión de requisitos, ingeniería de requisitos, objetivos, requisitos, software, stakeholders.

Abstract—The purpose of this paper is the presentation of the results of an analysis of requirements of a software requirements management. The study is based on applying the methodology of the 3 dimensions of engineering software requirements, which is based on the selection of sources - stakeholders, existing documentation and comparative analysis of other management software available on the market - collection structures - scenarios, objectives and requirements - elicitation techniques to implement, document analysis, prototype development of a possible management software and a stakeholder interview. In the selection of the different sources of manuals and documents different software requirements management were compared during the study were reviewed. An interview was also performed to a specific stakeholder experienced in the use of such software. All results of the analysis, comparisons, gathering requirements, and conclusions are documented in a vision document, in a SRS and the specific characteristics of each software analyzed.

Index Terms—Objectives, requirements, requirements engineering, requirements management, scenarios, software, software requirements specification, stakeholders, vision document.

1 Introducción

Durante el desarrollo de este proyecto se planteó un objetivo muy concreto: el análisis de requisitos de un software de gestión de requisitos. Para comprender un poco más en detalle el objetivo del TFG, podemos descomponerlo en subobjetivos. Tal y como se muestra en el apartado A1 del apéndice, vemos el árbol AND-OR (árbol n-ario, utilizado para representar conocimiento sobre grupos de tareas que se deben ejecutar para lograr algún objetivo), que desglosa el objetivo principal en subobjetivos fácilmente entendible, y los pasos a realizar para lograr cumplirlo. El propósito del estudio es especificar los requisitos que tiene que poseer un sistema software de gestión de requisitos, para que permita tener una buena recogida de requisitos y poder realizar un proyecto exitoso. Según el estudio de *Standish Group*, actualmente existen un porcentaje bastante elevado de proyectos que acaban en fracaso debido a una mala recolección de los mismos [1].

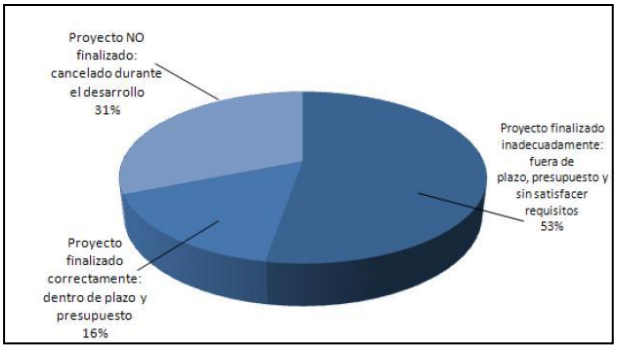


Figura 01: Gráfico sobre el estado de proyectos finales

- *E-mail de contacto: ripolls27@gmail.com*
- *Mención realizada: Ingeniería del Software.*
- *Trabajo tutorado por: Daniel Ponsa (Ciencias de la Computación)*
- *Curso 2013/14*

Para ello, es de crucial importancia tener unas pautas correctas a la hora de realizar dicha recogida.

Como podemos ver en el figura 01, más del 50% de los proyectos son terminados pero sin satisfacer al cliente. Los motivos por los que falla un proyecto derivan de requisitos incompletos, cambios en los mismos, falta de recursos, expectativas no realistas o productos finales no necesarios. Hay que tener en cuenta que el coste de corrección de defectos de requisitos es del orden de 100 a 200 veces inferior en las etapas iníciales, es decir, un fallo en los requisitos se ha de solucionar lo antes posible.

La ingeniería de requisitos hace tiempo que se ha incorporado a las empresas desarrolladoras de sistemas softwares, al comprobar que los proyectos que realizaban aumentaba su éxito de manera gradual. Como podemos ver en la figura 02, el estudio que realizó *Standish Group* se puede apreciar dicho incremento.

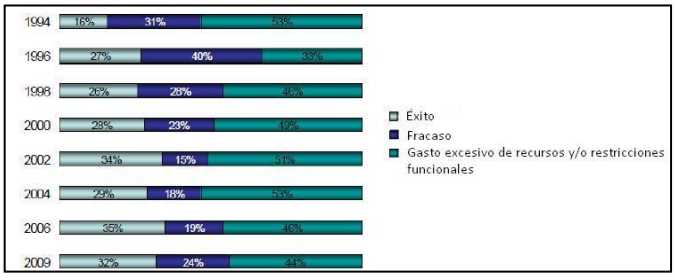


Figura 02: Gráfico sobre el estado de proyectos finales

¿Pero qué razones llevan que un proyecto termine en éxito o fracaso? Son muchos los motivos por los que un proyecto puede no llegar a buen puerto. En la figura 03 se muestran las razones más comunes.

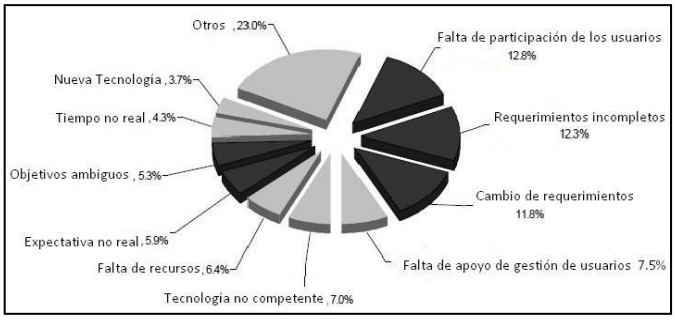


Figura 03: Razones para la insatisfacción de costos y funcionalidades

Como podemos apreciar en los sectores en gris oscuro, vemos que más del 50% están relacionadas con el hecho de no haber utilizado la ingeniería de requisitos.

Para evitar estos problemas, la clave está en determinar que acciones y/o pautas eran las correctas para poder desarrollar un sistema capaz de recoger correctamente todos los requisitos de un sistema para poder llevar a cabo con éxito el sistema a crear. Todos estos pasos se denominan ingeniería de requisitos. La idea de esta ingeniería es la especificación correcta que describa con claridad, sin ambigüedades, de forma consistente y compacta, el comportamiento del sistema.

Existe literatura y/o trabajos destacables que hablan y tratan sobre la ingeniería de requisitos [2], [3], [4], [5], [6].

Para la confección del TFG hemos seguido las 3 primeras etapas descritas en la ingeniería de requisitos. Dichas etapas son la elicitación, documentación, validación a las que seguirán la negociación y gestión. La negociación no se ejecutó debido a que solamente se pudo entrevistar a un stakeholder, y por lo tanto no hubo contradicciones en la selección de requisitos. Y la gestión no se llevó a cabo ya el objetivo del proyecto no consistía en el desarrollo del sistema.

La primera etapa es la selección de fuentes. Se tuvieron en cuenta diferentes tipos de softwares de gestión disponibles en el mercado, el análisis documental adscrita a los mismos, así como un posible prototipado que nos sirvió como apoyo visual con la entrevista al stakeholder.

Toda la selección de fuentes sienta las bases para recoger que cualidades tiene que poseer un buen software de gestión de requisitos para poder desarrollar un sistema software de manera correcta, ordenada, sin ambigüedad y lo más preciso a las necesidades del cliente. Se sondearon a dos stakeholders en la etapa de elicitación.

El primero fue Antonio López, a quien se le realizó la entrevista con el prototipo, dónde se le fue explicando los pasos y acciones que podía realizar. En este punto él dio su opinión al respecto y colaboró en las posibles mejoras y/o cambios.

El segundo fue el tutor del proyecto, Daniel Ponsa, quien revisó los documentos y diagramas del software de gestión. Una vez revisados dio su opinión y nos sugirió cambios y ampliaciones del mismo.

Todos los datos obtenidos en la etapa de la elicitación se documentaron en diferentes ficheros. Los documentos generados a partir de dicha etapa son:

- La comparativa de los softwares de gestión
- Documento de visión
- SRS
- Diagrama de clases
- Diagramas de actividades
- Casos de uso
- Escenarios
- Posible prototipo de un software de gestión

Una vez realizada la etapa de la documentación, el siguiente paso es validar dicha documentación para determinar si el proceso de obtención de los requisitos son los adecuados o no. El proceso de validación se realizó de una manera muy genérica, ya que solamente se corroboraron que los diseños estuvieran bien implementados y los documentos bien redactados. Esta validación fue realizada por Daniel Ponsa, tutor del TFG.

En los siguientes apartados podemos ver con mejor detalle el proceso del TFG. En la sección 2 se describe la metodología seguida para la recolección de requisitos, que resultados hemos obtenido y que conclusiones sacamos del estudio.

2 METODOLOGÍA

En todo proceso de investigación y desarrollo se necesitan unas pautas concretas para poder llegar a determinar con

éxito el objetivo propuesto en un proyecto.

En nuestro caso, para determinar las pautas y/o acciones que se necesitan para recoger con éxito los requisitos de un sistema de gestión de requisitos, hemos seguido 3 de las 5 etapas de la ingeniería de requisitos. Estas 3 etapas son: elicitación, documentación y validación [1].

Antes de explicar con detalle la metodología utilizada durante el desarrollo del TFG, explicaremos el proceso por el cual se consigue llegar a un objetivo exitoso o no exitoso. Todo dependiendo de cómo se utilice y en qué grado de detalle se alcance trabajando en los ejes de las 3 dimensiones de la ingeniería de requisitos. Tal y como se observa en la figura 04, el objetivo máximo a alcanzar en la ingeniería de requisitos es:

- Realizar una elicitación completa
- Tener una documentación acorde con las normas.
- Realizar una validación y negociación en dónde se lleguen a un punto de acuerdo mutuo entre todas las partes participantes.

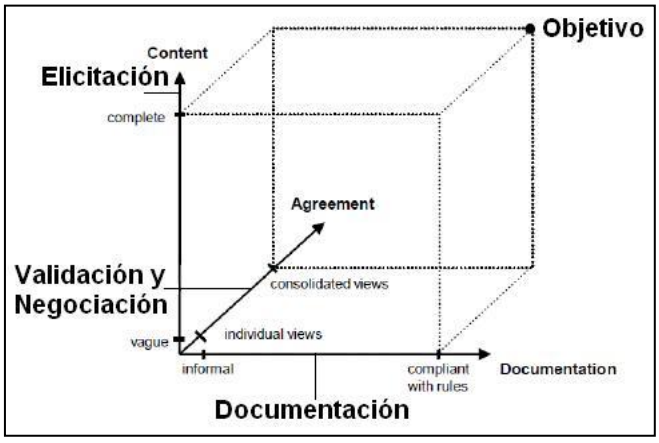


Figura 04: Las 3 dimensiones de la ingeniería de requisitos

No obstante, alcanzar este objetivo es en muchas ocasiones muy difícil, ya que, sobre todo se discrepa mucho en la validación y negociación de las partes implicadas en el proyecto.

Para tratar estos conflictos y se pueda llegar al objetivo o a un punto muy cercano, es muy importante la etapa de la elicitación, que ya es dónde se recogen todos los requisitos de todos los stakeholders involucrados en el proyecto y en donde se realiza el análisis documental del mismo.

A partir de esto, y teniendo una visión más amplia de las etapas que hay que realizar para alcanzar los objetivos, explicaremos cada una de las etapas de la ingeniería de requisitos, diciendo qué y porqué se hizo. De esta manera veremos la metodología de trabajo aplicada.

2.1 Elicitación

La elicitación de requisitos se considera como la primera etapa en el proceso de abstraer una comprensión del problema que se quiere resolver con el producto software. Se trata, esencialmente, de una actividad donde se identifican las partes interesadas y se establecen las relaciones entre el cliente, los usuarios y el equipo de desarrollado.

El objetivo de la elicitación es la definición de las tareas a realizar, los productos a obtener y las técnicas a emplear durante la actividad de elicitación de requisitos.

2.1.1 Estudio del dominio

Antes de mantener las reuniones con los clientes y usuarios e identificar los requisitos es fundamental conocer el dominio del problema y los contextos organizacional y operacional, es decir, la situación actual.

Enfrentarse a un desarrollo sin conocer las características principales ni el vocabulario propio de su dominio suele provocar que el producto final no sea el esperado por clientes ni usuarios.

Por eso es importante documentarse bien de las características y acciones que realizan los diferentes stakeholders. De esta manera se tendrá un vocabulario al mismo nivel que ellos y se podrá realizar un intercambio de información de manera más fidedigna, y se podrá controlar y acotar mejor el dominio de la aplicación a desarrollar.

En este apartado fue relativamente sencillo realizar el estudio del dominio, ya que el año pasado realicé la asignatura *Requisists del Software¸* dónde se daban las pautas necesarias para el estudio y desarrollo de una aplicación.

En la etapa de elicitación hicimos 2 pasos necesarios. La primera fue la selección de fuentes – sondear stakeholders, revisar documentación existente y compara softwares de gestión de requisitos que habían disponibles en el mercado -. Y el segundo paso la recogida y documentación de la información obtenida.

La selección de fuentes es un punto crucial, ya que nos desveló la mayoría de la información necesaria para obtener las características de la herramienta gestión de requisitos ideal.

2.1.2 Selección de fuentes

Como fuentes tuvimos tres pilares básicos, los stakeholders, la revisión de otros softwares de gestión de requisitos disponibles en el mercado actual y la lectura de la documentación sobre dichos softwares.

Primeramente se realizó la comparativa entre los softwares de gestión de requisitos disponibles en el mercado para determinar que características tenían en común y en cuales diferían. Esto nos ayudó a ver qué funciones tenían y ver si todas ellas seguían un mismo patrón estructural.

Seguidamente, la lectura de la documentación de los softwares de gestión, nos permitió ver en un rango mucho mayor las características de cada sistema, por lo que permitió ajustar y determinar con más precisión la elaboración del posible prototipo.

A partir de la revisión de los documentos y de las comparativas de los softwares del mercado, me hice una idea general de cómo podría ser el posible software de gestión. De aquí hice un posible prototipo a mano, que utilicé en la entrevista. El stakeholder entrevistado fue Antonio López. Escogimos a este stakeholder por ser el responsable de la mención de ingeniería del software y profesor de la asignatura *Laboratorio Integral de Software*,

dónde los alumnos tienen que desarrollar un sistema software desde cero, con lo que necesitan una buena herramienta que permita recoger bien los requisitos para la posterior implementación. En ella, Antonio me sugirió algunas modificaciones para que el diseño fuera mucho más competente. Esta interacción también es un punto clave ya que nos permitió ver si el software le faltaba algún detalle extra que no se hubiese visto con anterioridad en otros pasos de elicitación.

2.1.3 El prototipo

Como podemos ver en la figura 05 y 06, observamos algunas de las capturas de pantalla del posible prototipo de la herramienta de gestión de requisitos.

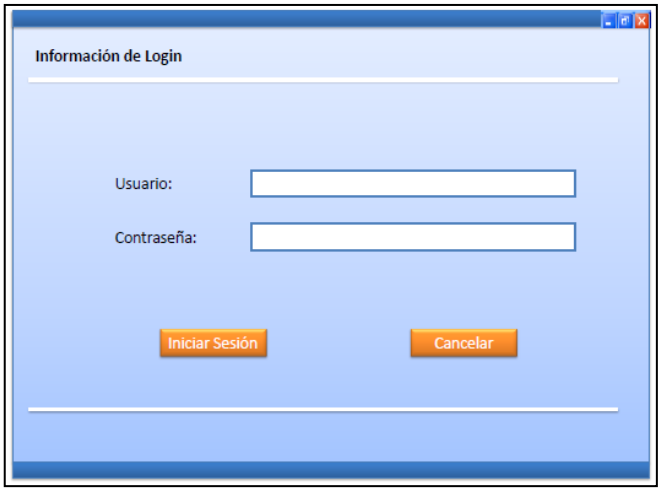


Figura 04: Pantallazo Iniciar Sesión

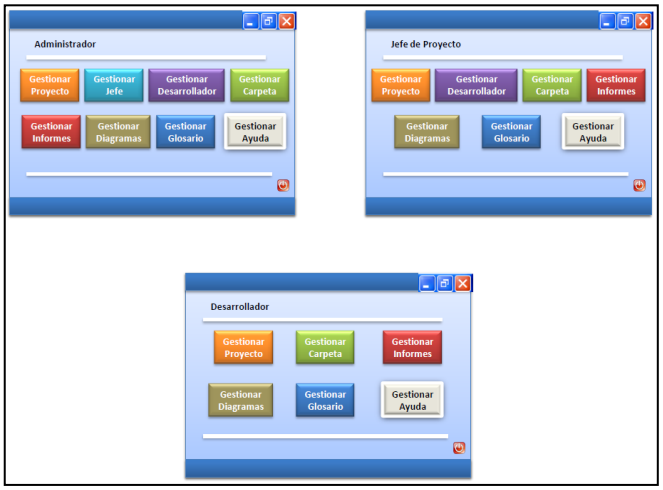


Figura 05: Pantallazo Sesión Personalizada

En la figura 04, vemos el inicio de la sesión de la aplicación, en dónde un usuario registrado en la base de datos accederá a ella mediante sus datos de sesión - identificador y contraseña.

La figura 05 muestra el acceso personalizado del usuario. Según tengas privilegios de administrador, jefe de proyectos o desarrollador, te permitirá realizar unas acciones u otras, y como muestra la figura 05.

2.2 Documentación

Una vez se realizó la etapa de elicitación, posteriormente se hizo la etapa de documentación. En la ingeniería de requisitos, toda la información que ha sido trabajada durante las diferentes fases de elicitación, tienen que ser documentadas. Esta información se recoge en unos documentos llamados documento de visión y en un SRS - Software Requirements Specification - o especificación de requisitos del software.

Un documento de visión es un documento que describe el plan general para un determinado proceso de software. Pretende ser un documento de alto nivel más breve y general que el SRS, y en él se describe lo que se espera llevar a cabo y las características que no están en el alcance, pero que se prevé agregarán al producto en posteriores etapas del desarrollo de éste. El propósito del documento es recopilar y analizar las ideas que han surgido para el futuro del producto, y asegurarse de que los interesados clave tienen una visión clara y compartida de los objetivos y alcance del proyecto. Identifica alternativas y los riesgos asociados con el proyecto. Durante el desarrollo del documento de visión, uno de los logros principales del análisis de negocio es que se deriven características para las necesidades de los interesados. Las características deben tener todos los atributos de un buen requerimiento: no redundante, claro, etc.

La especificación de requisitos de software es la actividad en la cual se genera el documento, con el mismo nombre, que contiene una descripción completa de las necesidades y funcionalidades del sistema que será desarrollado; describe el alcance del sistema y la forma en cómo hará sus funciones, definiendo los requisitos funcionales y los no funcionales. En la SRS se definen todos los requisitos de hardware y software, diagramas, modelos de sistemas y cualquier otra información que sirva de soporte y guía para fases posteriores.

En esta etapa, una vez se realizó el chequeo de los softwares, la entrevista y la lectura de la documentación existente, se realizó la documentación de todas las características deseadas.

Se redactaron informes con todas las características que tenían cada uno de los softwares analizados. El motivo era determinar que características compartían y en cuales diferían, de esta manera se podía ver y ajustar cada vez más todas las funciones para unirlas en un futuro software de gestión de requisitos.

Para poder ver con claridad como funcionaría el sistema se realizaron los diagramas de clase, diagramas de actividades, casos de uso y escenarios. Todos estos diagramas nos permitieron aclarar de manera mucho más concisa como tenía que ser y que características debía de tener el software de gestión de requisitos ideal.

2.3 Validación

Una vez documentado toda la etapa de elicitación, el siguiente paso fue realizar una validación de los artefactos y actividades elicitadas.

La validación consiste en actividades para detectar y corregir cualquier requisito innecesario e incorrecto. Es

un proceso de comprobación que sirven para evitar el riesgo de realizar una mala implementación del sistema [2], [4], [5].

En este paso se asegura que todas las características de los requisitos estén presentes en cada uno de los ellos, es decir, se identifican aquellos requisitos ambiguos, incompletos, inconsistentes, etc.

Por tanto en esta etapa se validaron los diagramas de la aplicación por parte de Daniel Ponsa, de una manera general, ya que el objetivo de la validación fue la de cerciorar que la redacción de los documentos, así como la implementación de los diagramas fueran correctos. Con sus aportes y comentarios, realicé los cambios oportunos para que los diagramas, casos de uso y documentos tuvieran la coherencia y consistencia adecuada. De esta manera nos aseguramos del buen camino hacia el correcto software de gestión de requisitos.

3 RESULTADOS

En este apartado se recogen los resultados obtenidos de aplicar la metodología descrita anteriormente.

3.1 Análisis de sistemas existentes

Al analizar los diferentes softwares de gestión del mercado hemos observado que la mayoría comparten muchas características comunes, las cuales hacen que la recogida y documentación de requisitos sea mucho más sencilla.

Los softwares utilizados fueron:

- IBM Rational DOORS 9.2
- Enterprise Architect 10
- REM
- OSRMT 1.5

Para poder ver que características comparten, lo podemos observar en las tablas 01 y 02:

Requerimientos Técnicos — Software de Gestión de Requerimientos					
Requerimientos	DOORS 9.2	E.A 10	REM	OSRMT 1.5	%
Software Libre			x	x	50%
Software Comercial	x	x			50%
S.O Windows	x	x	x	x	100%
S.O Linux	x	x		x	75%
Otro S.O	x			x	50%
Múltiples usuarios concurrentes	x	x		x	75%

Tabla 01: Tabla comparativa de los diferentes softwares analizados

Antes de nada haremos una breve explicación de las tablas. El valor de la columna % indica el porcentaje de los softwares analizados que aporta una determinada característica.

Como podemos ver, las herramientas analizadas son sistemas de gestión que poseen las mínimas características que permiten realizar una gestión y manejo de los requisitos muy completa. No obstante hay herramientas que aunque tengan las características idóneas para realizar la recogida de requisitos, su manejo no suele ser muy correcto. Esto sucede en el OSRMT 1.5. Su utilización es sencilla, tiene una interfaz intuitiva, a veces da problemas a la hora a la hora de guardar los requisitos con todos los datos y características, ya que en ocasiones no lo hace de manera correcta.

A continuación explicaremos las tablas comparativas. Los resultados con un 100% significan que todas las

Captura de Requerimientos — Software de Gestión de Requerimientos					
Requerimientos	DOORS 9.2	E.A 10	REM	OSRMT 1.5	%
Registrar Requisito	x	x	x	x	100%
Eliminar Requisito	x	x	x	x	100%
Modificar Requisito	x	x	x	x	100%
Identificar Requisito	x	x	x	x	100%
Ordenar		x	x	x	75%

Análisis de la Trazabilidad					
Requerimientos	DOORS 9.2	E.A 10	REM	OSRMT 1.5	%
Visualizar Gráfico de Dependencias	x	x		x	75%
Trazabilidad entre Requisito y Requisito	x	x	x	x	100%

Funcionalidades					
Requerimientos	DOORS 9.2	E.A 10	REM	OSRMT 1.5	
Crear Informes PDF	x	x	x	x	100%
Crear Informes HTML	x	x	x	x	100%
Crear Diagramas UML	x	x			50%
Editar Diagramas UML	x	x		x	75%
Realizar Búsquedas	x	x		x	75%
Filtrar Requisitos	x	x			50%
Gestionar Glosario	x	x	x		75%

Tabla 02: Tabla comparativa de los diferentes softwares analizados

herramientas poseen dicho requisito, con un 75% y con un 50% poseen alguna de ellas. Podemos ver que las herramientas que poseen la mayoría de los requisitos corresponden a herramientas de gestión con licencia, es decir, a sistemas privados de pago. Los open source son herramientas mucho más limitadas, no obstante se desenvuelven con soltura a la hora de realizar la recogida de los requisitos.

A pesar de que la mayoría de las herramientas son completas a la hora de realizar la recogida y selección de los requisitos por parte de los stakeholders, las soluciones conocidas no siempre están al alcance de todos. Las herramientas privadas o con licencia son herramientas con un coste muy elevado, ya que son sistemas complejos, muy trabajados por grandes compañías. Esto es una desventaja que tienen las pequeñas empresas al no poder hacer frente al precio de dichas herramientas, por lo que tiene que recurrir a las open source, ya que estas son a coste cero. Aunque sean sistemas bastante completos, no suelen llegar a tener el punto de fiabilidad y pulcritud de las privadas.

Una vez analizadas las herramientas softwares del mercado pudimos realizar un diagrama de caso de uso, diagramas de clase y actividades, un posible prototipo, la entrevista a Antonio López, redactar un documento de visión y un SRS del posible sistema a implementar. A continuación explicaremos cada una de ellas.

3.2 Principales funcionalidades identificadas

El propósito del estudio era hacer el análisis de requisitos de un sistema software de gestión de requisitos. Esto es importante ya que la mayoría de los proyectos acaban en fracaso debido a una mala recolección de los mismos.

Para ello, es de crucial importancia tener unas pautas correctas a la hora de realizar dicha recogida. Para poder tener una buena herramienta de requisitos del software ésta debe de contar con una serie de características que dé soporte a la trazabilidad de los requisitos, a su priorización, y a la gestión de los cambios que puedan ocurrir durante la gestión y desarrollo de un proyecto software.

Por otro lado también se quiere que aparte de tener dichas características, que la herramienta de gestión de requisitos sea sencilla de utilizar, intuitiva para cualquier persona que la maneje, así mismo que sea multiusuario para poder trabajar en paralelo y avanzar en el proyecto.

Interesa que la herramienta de gestión, permita crear, clasificar y mantener un catálogo de requisitos, así como tratar las relaciones entre los requisitos de manera independiente así como conjuntamente – trazabilidad -, trabajar con el glosario de términos y obtener informes de los requisitos, que nos permita tener una mejor lectura de ellos tanto para el cliente como para el desarrollador.

Como hemos dicho, el sistema tiene que ser de utilización sencilla, tiene que ser intuitiva, ya que tiene que estar prediseñada a personas con poca experiencia en el campo de gestión de requisitos, como a empresas que quieran disponer de un modo eficaz de una herramienta capaz de administrar los requisitos de un sistema.

Por lo tanto las funcionalidades más remarcables que debe de tener son:

1. Tiene que ser capaz de gestionar un sistema de carpetas donde se tienen que crear una o varias carpetas dentro de cada proyecto, de manera que se pueda tener la información ordenada y agrupada jerárquicamente

2. La herramienta tiene que tener una buena gestión de requisitos en donde tiene que dar soporte a todas las tareas relacionadas con los requisitos de un proyecto. Todos los atributos de los requisitos tienen que estar documentados, especificados y gestionados, como por ejemplo, el estado de un requisito, que stakeholders los solicita, quien les da soporte, fecha de creación tuvieron, si hay fecha de cambios o modificación del mismo, de qué tipo de requisito se trata, etc.

3. La herramienta tiene que dar soporte a la creación, modificación y eliminación de usuarios que han accedido al sistema. Es decir, tiene que tener un buen sistema de gestión de usuarios, para que cada uno de ellos pueda acceder de manera individual, pero al tiempo que puedan trabajar en paralelo en un mismo proyecto. Gracias a esto pueden trabajar de manera simultánea sin poner en peligro las tareas de los demás.

4. La aplicación, por otro lado, debe de permitir las salidas de información sobre la trazabilidad que tienen los diferentes requisitos de un proyecto. Con este fin se pretende obtener un informe mucho más legible y entendible tanto como para la persona que realiza el proyecto, como cualquier persona que se añada de nuevo a él, así como para el cliente.

El objetivo es la obtención detallada de dicho sistema donde se representa las características que necesita del cliente y, posteriormente, generar un documento donde ayude a los desarrolladores del software a crear una buena herramienta de gestión de requisitos.

3.3 Diagrama de Clases

Un diagrama de clases es un tipo de diagrama estático que describe la estructura de un sistema mostrando sus clases, orientados a objetos. El diagrama de clases lo podemos ver en la figura A2 del apéndice, donde nos muestra las conexiones del posible sistema a implementar. El proceso es el siguiente. Un usuario inicia sesión en el sistema con sus datos de sesión. Una vez logeado, podrá acceder a los proyectos que tiene asignado. En los proyectos podrá gestionar carpetas, artefactos - escenarios, objetivos y requisitos-, diagramas, el glosario y la ayuda. Las relaciones entre las clases son de composición - ◇ o agregación - ◆. Dónde la composición nos designa que un elemento posee a otro y que si el primer elemento es eliminado, el elemento que forma parte de él también es

eliminado. Por ejemplo el caso de las carpetas. Éstas estas compuestas por artefactos. Si la carpeta es eliminada, por consiguiente los artefactos también lo serán. En cambio la agregación no sucede esto. Un elemento puede sobrevivir por si solo si el primer elemento es eliminado. Por ejemplo, esto lo vemos en las fuentes y la relación con un artefacto. Si desaparece la fuente de donde se sacó el requisito, no necesariamente el artefacto desaparece del proyecto.
En cada una de las clases describimos los atributos que guarda el sistema en cada caso. Por ejemplo, retomando la figura A2 del apéndice, la clase *proyectos* guarda los siguientes atributos:

- Identificador: identificador único del proyecto, asignado de manera automática por el sistema.
- Nombre: nombre dado al proyecto.
- Autor: creador del proyecto.
- Descripción: breve descripción de las funcionalidades del proyecto.
- Fecha de Creación: fecha de inicio del proyecto.
- Versión: versión del proyecto por si ha habido modificaciones.

Así como también guardamos las funciones que puede realizar. En este caso, podemos realizar las funciones de:

- Generar Informe: genera los informes del proyecto una vez recogido y grabado los requisitos en la base de datos.
- Generar Matriz de Trazabilidad: permite mostrar la trazabilidad de los requisitos.
- Generar Árbol de Decisión: muestra el árbol de decisión de los requisitos.

3.4 Diagrama de Actividades

El diagrama de actividades es la representación gráfica de un determinado proceso, es decir, representa los flujos de trabajo paso a paso de negocio y operacionales de los componentes en un sistema. En el proyecto hemos realizado los diagramas de actividades de los principales casos de uso – añadir, modificar, eliminar, asignar, añadir relaciones entre artefactos, etc. como ver en la figura A3 del apéndice, mostramos el diagrama de actividades de *añadir artefacto.* Los pasos a seguir son los siguientes. Una vez has iniciado sesión en el sistema, seleccionas el proyecto donde trabajar. Una vez en el proyecto, seleccionas añadir artefacto y escoges que artefacto añadir – escenario, objetivos o requisitos. Un vez lo seleccionas añades los datos correspondientes a los mismo. Y lo guardas en la carpeta correspondiente. Si la carpeta aún no existe se guardará en la raíz del proyecto. Posteriormente lo podrás cambiar de ubicación.

3.5 Casos de Uso

Los casos de uso son una descripción de los pasos o las actividades que tienen que realizarse para llevar a cabo algún proceso. En el contexto de ingeniería del software, un caso de uso es una secuencia de interacciones que se desarrollan entre un sistema y sus actores en respuesta a un evento que inicia un actor principal sobre el propio sistema. Con ellos especificamos la comunicación y el comportamiento de un sistema mediante su interacción con los usuarios y/u otros sistemas [11]. Los requisitos funcionales que hemos representado mediante los casos de uso han sido – ver figura A4 del apéndice:
Hay un requisito común en los 3 roles – administrador, jefe de proyecto y desarrollador - de la herramienta. Estos son:

- Iniciar sesión: un usuario registrado en la base de datos del sistema podrá iniciar sesión con un identificador y una contraseña. Si no recuerda dichos datos podrá recuperarlos mediante el email o su identificador.

Administrador (actor)

- Gestionar proyecto: permite gestionar todos los proyectos del sistema – añadir, modificar, eliminar, asignar jefe de proyecto.
- Gestionar usuarios: podrá gestionar a todos los usuarios del sistema – añadir, modificar, eliminar.

Jefe de proyectos (actor)

- Gestionar desarrolladores: permite gestionar el equipo de proyecto.
- Gestionar proyectos asignados: donde podrá gestionar el equipo de proyecto

Jefe de proyectos y Desarrollador (actores)

- Gestionar carpetas: gestionar las carpetas del proyecto.
- Gestionar informes: gestionar los informes del proyecto – visualizar, imprimir y exportar.
- Gestionar artefactos: gestionar los diferentes tipo de artefactos – escenarios, objetivos y requisitos.
- Gestionar glosario: añadir, modificar y eliminar términos.
- Gestionar ayuda: manual de ayuda del sistema.
- Gestionar diagramas: gestionar los diferentes diagramas – casos de uso, actividades, clase, etc.

Para mayor comprensión de los casos de uso, en la figura A5 del apéndice podemos ver la descripción en detalle de un caso de uso en concreto y de su escenario.

3.6 Documento de Visión

En el documento de visión hemos definido el alcance

de alto nivel y propósito del programa, producto o proyecto. Es una declaración clara del problema, la solución propuesta, y las características de alto nivel de un producto que ayudan a establecer las expectativas y reducir los riesgos de efecto del mismo.

3.7 SRS

La especificación de requisitos de software o SRS es una descripción completa del comportamiento del sistema a desarrollar. Incluye un conjunto de casos de uso que describe todas las interacciones que tiene los usuarios con el software. En este caso los casos de uso están descritos en un documento aparte, pero que se puede añadir como apéndice del SRS. En el documento de especificación del software hemos redactado los requisitos funcionales así como los no funcionales del sistema. Como hemos comentado, los casos de uso son los requisitos funcionales. Y los requisitos no funcionales son requisitos que imponen restricciones en el diseño o la implementación.

3.8 Portafolio

Durante el desarrollo de todo el TFG, hemos ido recopilando mucha información, redactado informes y confeccionando diagramas. Las herramientas utilizadas para la redacción y confección de dichos documentos fueron los siguientes:

- Microsoft Word: utilizado para la redacción de los documentos de visión, SRS, confección del diagrama de clase, casos de uso, diagrama de actividades, informes, artículo final del TFG, entre otros.
- Microsoft Power Point: para la creación del prototipo de la herramienta de gestión de requisitos.
- Microsoft Project: diagrama de Gantt para la planificación del TFG.

Todo esto se ha ido guardando de manera clara y ordenada en el portafolio del proyecto, organizado en carpetas para un mejor acceso a la información. El portafolio que hemos utilizado ha sido *DropBox*, ya que me permitía tener un contacto más directo con el Tutor, al tener las carpetas compartidas con él, además de poder acceder a toda la información desde cualquier dispositivo - ordenador, ipad, etc.

El propósito del portafolio no es otro que el de ver el seguimiento que se ha ido realizando a lo largo de todo el proceso de desarrollo del TFG, ya que podemos ver las versiones anteriores y las actualizadas de todos los informes, actas y diagramas.

4 Conclusión

El objetivo del proyecto era realizar un análisis de requisitos de un software de gestión de requisitos. El análisis se realizó mediante la comparativa de diferentes softwares que había disponibles en el mercado, el análisis documental de dichas herramientas y la lectura de material adicional con relación a la recogida de los requisitos. Con toda esta información realizamos un posible prototipo que se presentó a Antonio López al cual le realizamos un entrevista para cotejar si el planteamiento era el correcto.

Los documentos y diagramas que generamos fueron:

- Diagrama de clase
- Diagrama de caso de uso
- Diagrama de actividades
- Documento de visión
- Documento SRS
- Prototipo
- Escenarios de los casos de uso.
- Características de los softwares analizados

No obstante en el TFG no pudimos realizar todas las etapas de la metodología de trabajo de las 3 dimensiones de la ingeniería de requisitos, ya que solamente pudimos realizar una sola entrevista, por lo cual la etapa de negociación no se aconteció. Sin embargo, si hubiéramos tenido el acceso a otros stakeholders con conocimiento sobre las herramientas de gestión hubiéramos podido extraer más información de que requisitos debe de tener las mismas. A más a más con la información de más stakeholders podríamos a ver pulido mucho más los documentos de visión y el documento SRS.

Agradecimientos

A mis padres y hermano por todo su apoyo durante estos años de carrera.

A mi tutor, Daniel Ponsa, por sus consejos y ayuda durante el desarrollo del TFG.

Bibliografía

[1] http://blog.standishgroup.com/
The Standish Group © 2013 - acceso 26/11/2013

[2] Daniel Ponsa, "Transparencias de la asignatura *Requisits del Software*", Escuela de Ingeniería, Universidad Autónoma de Barcelona.

[3] Klaus Pohl, "Requirements Engineering – Fundamentals, Principles and Techniques", Springer-Verlag Berlin Heidelberg 2010.

[4] Klaus Pohl, Chris Rupp, "Requirements Engineering Fundamentals", Rocky Nook Inc, USA, 2011.

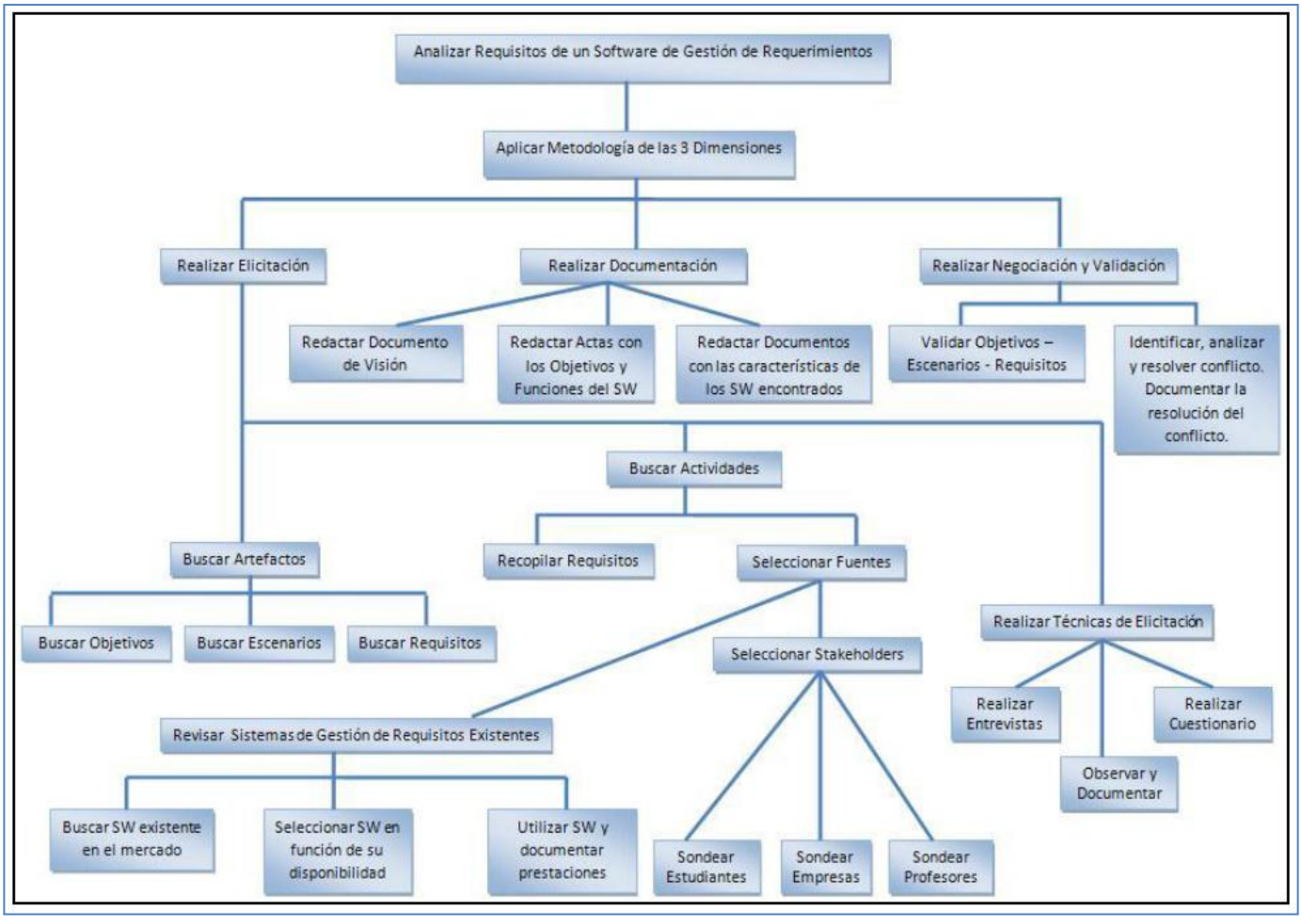
[5] M.G.Christel & K C. Kang, "Issues in Requirements Elicitation", Software Engineering Institute, Carnegie Mellon University, 1992.

[6] Umar Sajjad, "Issues and challenges of requirement elicitation in large web projects", Lap lambert academic publishing, 2010.

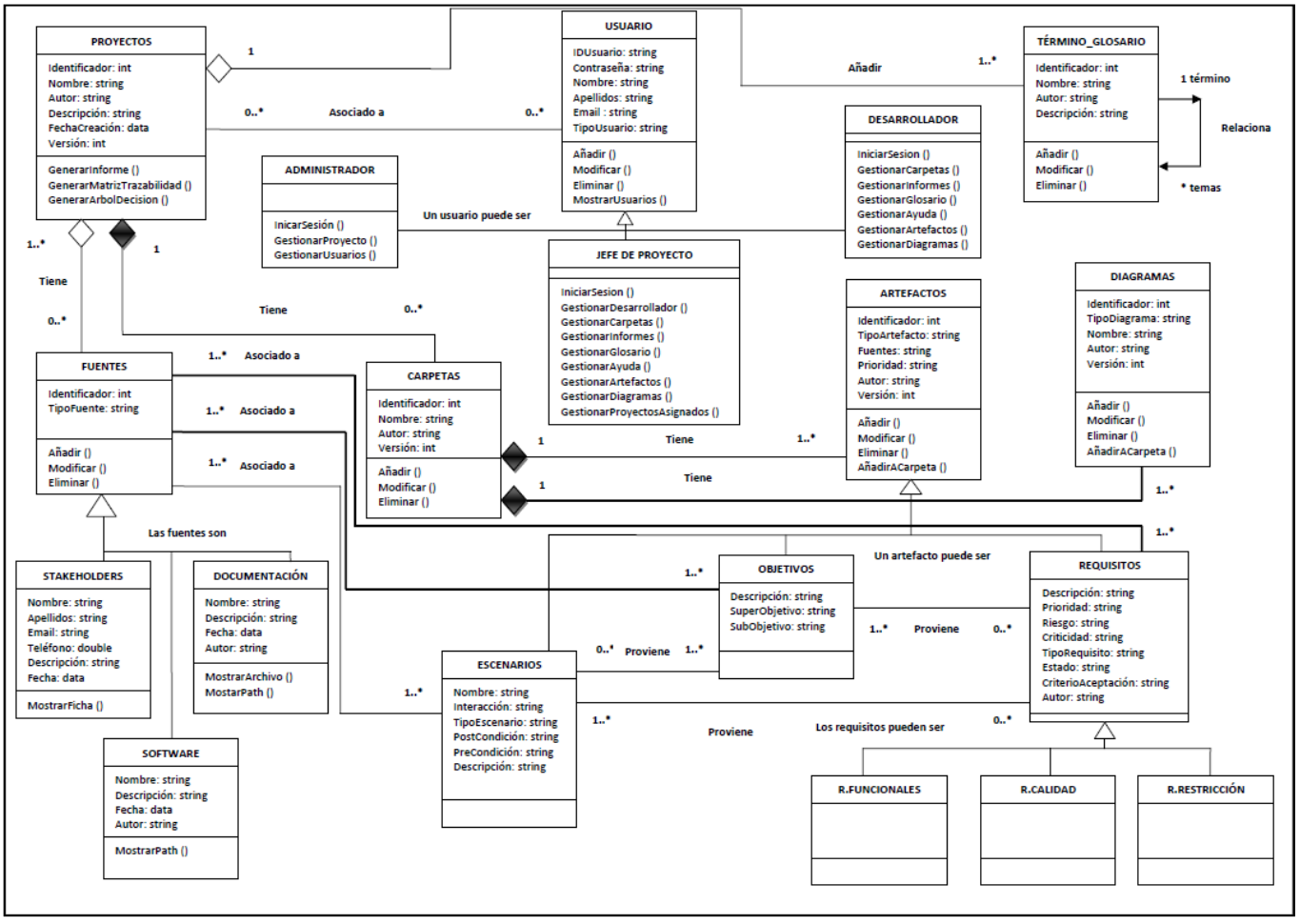
[7] www-03.ibm.com/software/products/es/ratidoor/
©International Business Machines - acceso 01/12/2013
[8] www.juntadeandalucia.es/servicios/madeja/contenido/recurso/420
© JUNTA DE ANDALUCÍA - acceso 01/12/2013
Manual de REM, Dr. Amador Durán, Universidad de Sevilla, 2000
[9] www.sparxsystems.com.ar/download/ayuda/erica_content_static.html
© 2000 – 2013 Sparx Systems Pty Ltd. - acceso 01/12/2013
[10] www.ideastub.com/**osrmt**.../**osrmt**_user_manual.pdf
© 2010-2013 ideaStub LLC - acceso 01/12/2013
[11] Thomas A. Pender, "UML Weekend Crash Course", Wiley Publishing Inc.

APÉNDICE

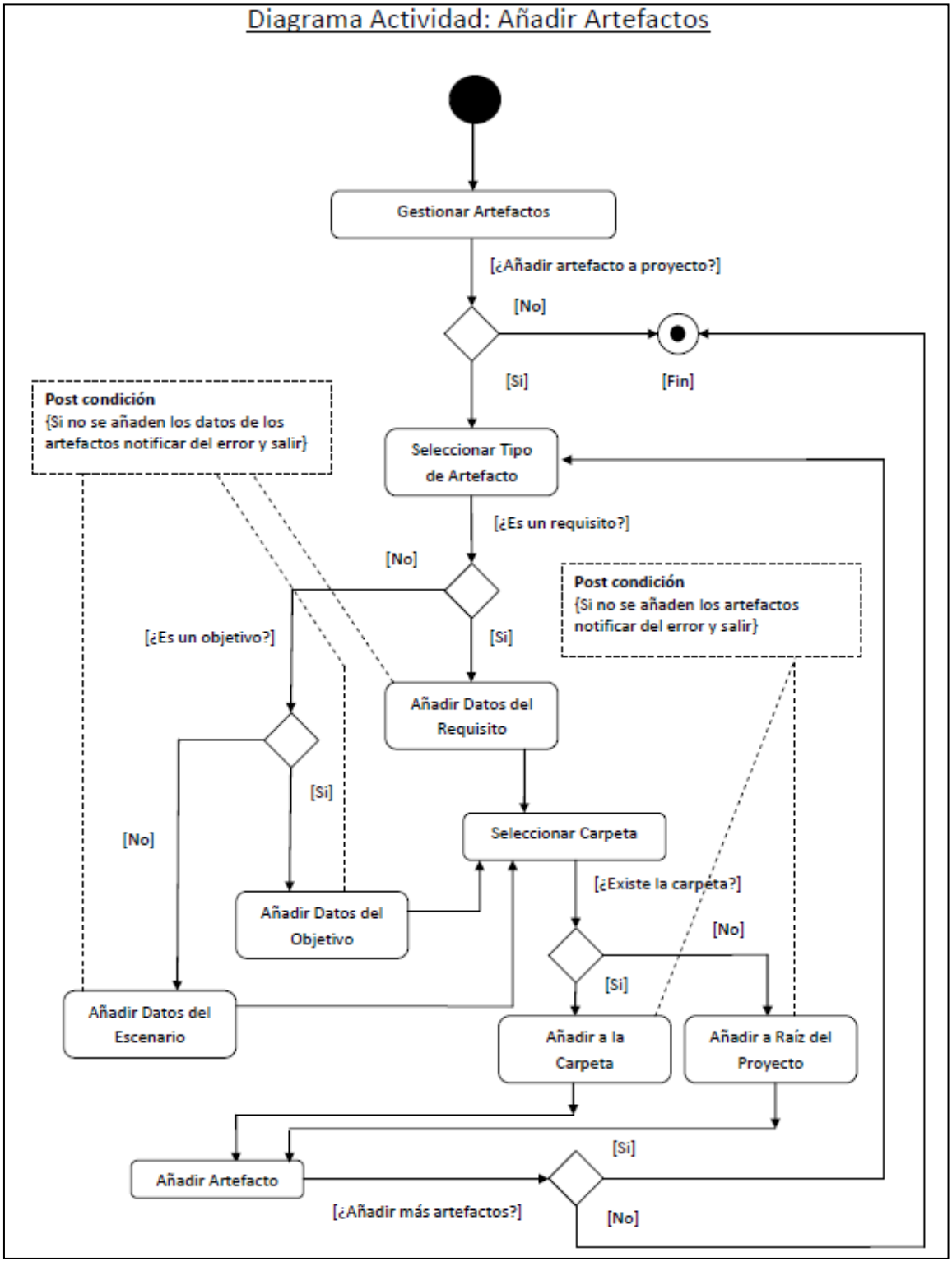
A1. Árbol AND-OR



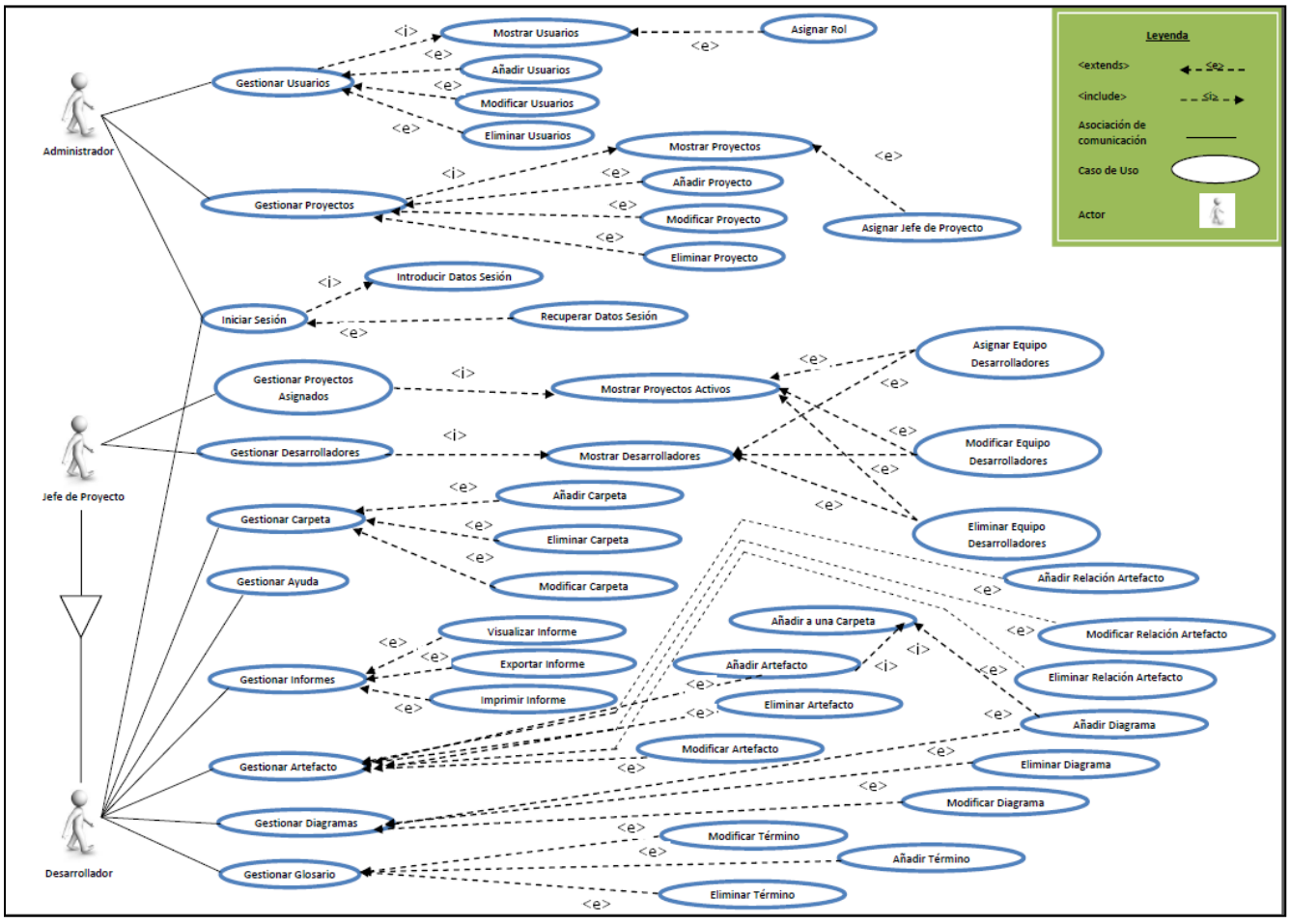
A2. DIAGRAMA DE CLASE



A3. CASOS DE ACTIVIDADES



A4. Diagrama de Caso de Uso



A5. Escenario del Caso de Uso

Escenario – Añadir Artefactos		
Sección	**Contenido**	
Identificador	TFG0010	
Nombre	Añadir Artefactos	
Autor	SRL	
Versión	2.0	
Historial de Cambios	1.0 28.01.2014 1.1 30.01.2014 2.0 06.02.2014	
Prioridad	Media	
Criticidad	Alta	
Descripción	Tanto el jefe de proyectos como el desarrollador podrán dar de alta en la base de datos del sistema un artefacto y añadirlo al proyecto asignado.	
Objetivos	Añadir al proyecto los artefactos necesarios – escenarios, objetivos y requisitos – para que el sistema sea completo.	
Actores	Jefe de Proyecto - Desarrolladores	
Pre Condición	Estar registrado en la base de datos.	
Post Condición	Acceso al proyecto asignado.	
Resultado	Artefactos agregados al proyecto.	
Pasos del Escenario	1	El usuario selecciona un determinado artefacto para agregarlo al proyecto y da aceptar.
	2	El sistema carga las características del artefacto.
	3	El usuario rellena los datos del artefacto y da aceptar.
	4	El sistema guarda los datos en la base de datos.
	5	El sistema guarda el artefacto en la carpeta correspondiente si existe.
	5a	El sistema guarda el artefacto en la raíz del proyecto si la carpeta no existe.
Calidad	Acceso al sistema en 1 segundo	