

Módulo web para la gestión de proyectos software

- Cliente: Grupo Erabaki
- Suministrador: Alazne Delgado Arteche

Resumen: El módulo web a desarrollar debe ser susceptible de ser incorporado a una aplicación web que ofrece funcionalidades útiles para el equipo de desarrollo de software. El objeto del presente proyecto es la búsqueda de una solución para la gestión de procesos y proyectos, apoyándonos en prototipos parciales para valorar el coste económico de la construcción y de la tecnología. Debe de cumplir con algunas de las funcionalidades de las que están dotadas las herramientas de gestión de proyectos tales como crear planificaciones, mostrarlas y modificarlas.

El seguimiento de una metodología estricta y una documentación de calidad es imprescindible para la reusabilidad del módulo web.

- Duración estimada: 18 meses
- Coste estimado: 865.484,64 €

Índice de la memoria

1.-Introducción.....	5
2.-Objeto.....	7
3.-Antecedentes.....	9
4.-Descripción de la situación actual.....	10
5.-Normas y referencias.....	12
5.1.-Método.....	13
5.1.1.-Procesos, roles, actividades y artefactos.....	13
5.2.-Normas.....	16
5.3.-Bibliografía.....	16
6.-Definiciones y abreviaturas.....	18
7.-Requisitos iniciales.....	24
8.-Alcance.....	25
9.-Hipótesis y restricciones.....	26
10.-Estudios de alternativas y viabilidad.....	27
11.-Descripción de la solución propuesta.....	28
12.-Análisis de riesgo.....	31
12.1.-Pérdida de datos (magnitud 10).....	31
12.2.-Problemas en selección del software de construcción y riesgo tecnológico (magnitud 9).....	32
12.3.-Velocidad de carga de los diagramas GANTT (magnitud 9).....	32
12.4.-Elevadas curvas de aprendizaje en las herramientas a utilizar (magnitud 7).....	33
13.-Organización y gestión del proyecto.....	33
14.-Planificación temporal.....	34
15.-Resumen del presupuesto.....	36
16.-Anexos justificativos de la solución adoptada.....	38
16.1.-Análisis y diseño del sistema.....	38
16.2.-Estimación de tamaño y esfuerzo.....	58
16.3.-Plan de seguridad.....	58
17.-Especificaciones del sistema.....	59
18.-Anexos referentes a la gestión del proceso de elaboración de este proyecto.....	74
18.1.-Planificación temporal para el prototipo.....	75
18.2.-Resumen del presupuesto para el prototipo.....	76
18.2.1.-Entorno educativo.....	77
18.2.2.-Proyecto empresarial.....	77
18.3.-Conclusiones.....	78

1.- Introducción

El presente documento nos va a servir para poder comprender en mayor profundidad cuál ha sido el objetivo del proyecto GesProSoft, así como el desarrollo completo de dicho proyecto, desde su inicio hasta la finalización del mismo.

En la redacción de este proyecto hemos seguido las indicaciones de la norma AENOR *UNE 157 801* "Criterios generales para la elaboración de proyectos de Sistemas de Información", en cuyo grupo de trabajo ha participado el director de este proyecto fin de carrera, Juan Manuel Pikatza.

Una de las consecuencias clave de la aplicación de la norma es la consideración por separado de la elaboración del proyecto con respecto a la construcción de la solución informática definida en la elaboración. El objetivo de la misma es facilitar la evaluación de soluciones informáticas presentadas como proyectos.

Como consecuencia de la aplicación de la norma la estructura de este documento queda de la siguiente manera:

Memoria

0. **Hojas de identificación:** Portada, índice.
1. **Introducción:** Explicación breve tanto del objetivo como del contenido y estructura de la memoria del proyecto haciendo referencia a la norma que se va a aplicar.
2. **Objeto:** Descripción breve del objetivo final del proyecto y de la finalidad que justifica su ejecución.
3. **Antecedentes:** Se enumerarán las soluciones existentes permitiendo comprender tanto las alternativas estudiadas como la solución propuesta.
4. **Descripción de la situación actual:** Descripción de todos los elementos que se ven afectados por el cambio propuesto en el proyecto: software, equipamiento hardware, licencias software, etc.
5. **Normas y referencias:** Normas, métodos y referencias que se han seguido durante la elaboración del proyecto, o que se deban tener en cuenta en la fase de construcción.
6. **Definiciones y abreviaturas:** Todas las definiciones, abreviaturas, etc. que se han utilizado en este documento y su significado.
7. **Requisitos iniciales:** Breve enumeración de requisitos solicitados por el cliente.
8. **Alcance:** Enumeración y contenido de todos los elementos del proyecto que se le van a entregar al cliente al finalizar el proyecto.

9. **Hipótesis y restricciones:** Enumeración de las hipótesis de trabajo y restricciones que tienen incidencia en cualquier tipo de estimación reflejada en este documento (coste, plazo, calidad, etc.).
10. **Estudio de alternativas y viabilidad:** Enumeración de las alternativas que se han tenido en cuenta y justificación de la alternativa elegida explicando las razones por las que las otras han sido descartadas.
11. **Descripción de la solución propuesta:** Características significativas de la solución propuesta.
12. **Análisis de riesgos:** Lista de riesgos que se pueden dar a lo largo de la construcción del proyecto, evaluando su impacto y las posibles acciones de mitigación de cada uno de ellos.
13. **Organización y gestión del proyecto:** Normas, directrices o métodos de trabajo que hagan referencia a la organización y gestión del proyecto.
14. **Planificación temporal:** Entregas parciales, reparto de tareas y duración del proyecto a partir de la fecha de iniciación del mismo hasta su comercialización.
15. **Resumen del presupuesto:** Coste total de la construcción del proyecto para el cliente.
16. **Anexos:** Justificativos de la solución adoptada
 - 16.1. **Análisis y diseño del sistema:** Diagramas de secuencia y colaboración, así como los casos de uso realizados a lo largo del ciclo de vida del proyecto.
 - 16.2. **Estimación de tamaño y esfuerzo:** Estimación del esfuerzo que nos llevaría realizar la construcción de una solución real explicando brevemente la planificación temporal.
 - 16.3. **Plan de seguridad:** Tipo de seguridad tanto física como lógica que requiere el proyecto para asegurar su correcto funcionamiento.
17. **Especificaciones del sistema:** Estudio realizado para obtener los requisitos del sistema, y definición de éstos requisitos en el sistema.
18. **Anexos referentes a la gestión del proceso de elaboración de este proyecto**
 - 18.1. **Planificación temporal para el prototipo:** Entregas parciales, reparto de tareas y duración de la fase de elaboración del proyecto.
 - 18.2. **Resumen del presupuesto para el prototipo:** Coste total de la fase de elaboración del proyecto tanto en el entorno educativo como en el entorno empresarial.
 - 18.3. **Conclusiones:** Detallar lo que ha supuesto la realización del proyecto.

2.- Objeto

Las empresas que desarrollan software tratan de dar soluciones con sus aplicaciones a muchos de los sectores que les rodean, y sin embargo, muchas veces, no consiguen solucionar algunos de los problemas que surgen en su propio proceso de producción y que les dificultan y retrasan en su trabajo. Un ejemplo que nos encontramos es que cuando un cliente plantea la necesidad de construir un nuevo software, no existen herramientas que de una forma automática generen una planificación.

El cliente que necesita una solución efectiva a su problema, exige una solución de calidad en la que se detallen los aspectos técnicos y tecnológicos de la solución junto con los aspectos organizativos, temporales y económicos de la construcción de la solución

Hacer un buen análisis y diseño de la solución hace posible elaborar una planificación y presupuesto más certera.

Así pues, y viendo esta necesidad, tomamos la decisión de llegar a un diseño detallado y a la construcción de un prototipo para conseguir una buena planificación y presupuesto. Para que el proceso de desarrollo del software mejore constantemente hemos tenido que pensar en definir un proceso y crear a partir de él las tareas de un proyecto. Al igual que en otras ingenierías, la especialización es la mejor forma de conocer un dominio y ser líderes en él.

Por lo tanto, el proyecto que hemos desarrollado, llamado GesProSoft¹, tiene como principal objetivo la determinación de una solución al problema de la gestión de proyectos probando la tecnología a usar mediante un prototipo y tomando las decisiones pertinentes en función de lo que de ese desarrollo se concluya. El punto más relevante de este proyecto es conseguir la integración entre la gestión de procesos y la gestión de proyectos, ya que en la actualidad, podemos encontrar en el mercado tanto herramientas para la gestión de procesos como herramientas para la gestión de proyectos (la más conocida MS Project). Pero sin embargo, no encontramos herramientas con la integración de ambas técnicas, con lo cual, se podía plantear el desarrollo de un proyecto de gran interés.

Vamos a ver como el proyecto GesProSoft, que hemos realizado para el dominio del desarrollo del software, utiliza procesos para generar una planificación inicial de un proyecto. Para ello, hemos definido una ontología que recoge los conceptos relacionados con el dominio de la gestión de los proyectos y los procesos. Con esta ontología hemos definido los tres tipos de procesos que vamos a utilizar en este proyecto (secuencial, decisional, subproceso). Ambas tareas se han llevado a cabo

¹ GesProSoft: GESTión de PROyectos SOFTware

con la herramienta de adquisición del conocimiento llamada Protégè[12]. Podemos ver un ejemplo de proceso secuencial en la Figura 1, y un ejemplo de definición de su ontología en la Figura 2.

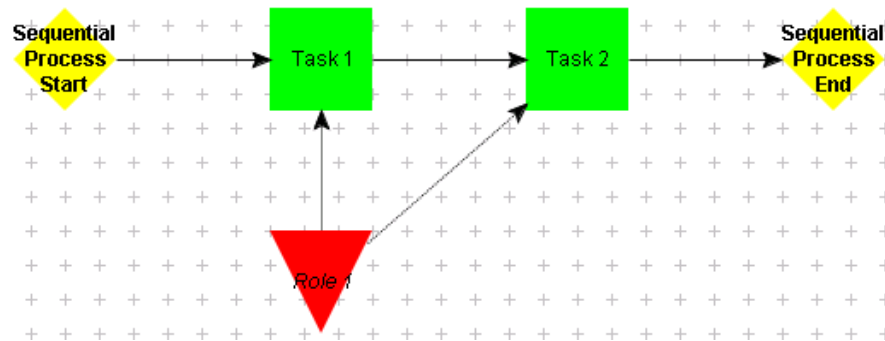


Figura 1: Ejemplo de proceso de tipo secuencial

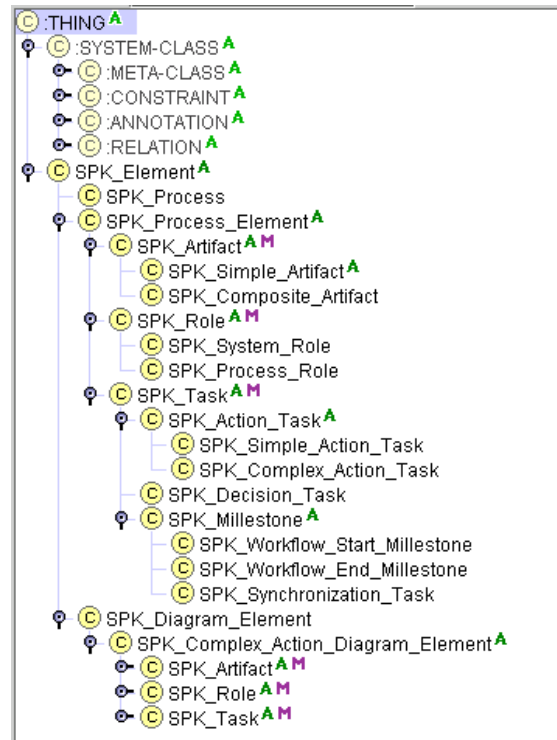


Figura 2: Ejemplo de ontología definida para un proceso

De este modo, tendremos definidos todos los datos sobre procesos de características comparables al proyecto del que queramos obtener un plan, consiguiendo gestionar los proyectos software de una forma rápida y con una mayor exactitud en la estimación de tiempos de duración de las tareas. El tiempo estimado para la creación de una nueva planificación para un proyecto es menor con Ges-ProSoft que con otras herramientas, ya que no tenemos que generar una planificación partiendo des-

de cero y añadiendo una a una las tareas que forman parte del proyecto, sino que mediante la gestión de procesos generamos una planificación inicial, y a dicha planificación le podemos realizar los cambios oportunos en los tiempos de las tareas, si es que se cree necesario, hasta generar la planificación del proyecto que nosotros deseemos.

3.- Antecedentes

Actualmente podemos encontrar en el mercado diversas herramientas que nos ayudan a realizar la gestión de los proyectos. Algunas de ellas son de tipo comercial, como el MS Project² y SuperProject³, y otras, sin embargo, están sujetas a licencias GPL como ToutDoux⁴, dotProject⁵, Mr Project⁶ y TaskJuggler⁷. Todas ellas están dotadas de muchas funcionalidades interesantes para obtener una buena gestión de los proyectos, sin embargo, podemos pensar en otras opciones de las que carezcan, y que nos faciliten la realización de la gestión.

Si analizamos detenidamente el tiempo que invertimos en cada una de las tareas que debemos desarrollar en el transcurso de la utilización de cualquiera de las herramientas existentes para la gestión de proyectos, podemos comprobar que la mayor parte del tiempo está destinado a la creación de una nueva planificación para un proyecto nuevo. Esto es debido a que para cada planificación que se va a hacer, se debe añadir una a una cada tarea que forme parte de la planificación, asignándole a cada una de ellas tanto su tiempo de duración, las dependencias que pueda tener con el resto de las tareas que formen parte del plan, el personal que va a realizar cada una de estas tareas, agregar información acerca de un recurso, etc. Aquí, de nuevo, vemos la importancia de la especialización en un dominio.

MS Project es una aplicación comercial desarrollada por la empresa norteamericana Microsoft y permite crear proyectos, tareas, definir las dependencias entre ellas, crear recursos (personas, materiales), asignar recursos a las tareas, planificar éstas en función de sus dependencias y de la disponibilidad de los recursos, etc.

SuperProject es una herramienta comercial desarrollada por Computer Associates. Tiene una funcionalidad muy parecida al MS Project, pero además ofrece características avanzadas, como planifi-

2 Microsoft Corporation: *MS Project*. <http://office.microsoft.com/es-es/FX010857953082.aspx>

3 Computer Associates: *SuperProject*. <http://www.abox.com/productos.asp?pid=173>

4 FSF: *ToutDoux*. <http://www.gnu.org/software/toutdoux/>

5 dotProject Developers: *dotProject*. <http://www.dotproject.net/>

6 CodeFactory AB: *Mr Project*. <http://mrproject.codefactory.se/>

7 Chris Schläger and Klaas Freitag: *TaskJuggler*. <http://www.taskjuggler.org/>

cación automática de las tareas, informe del camino crítico, gestión de cuentas de gastos, indicadores financieros avanzados, etc.

Mr Project y ToutDoux son dos programas con licencia GPL para la gestión de proyectos. Ambos están en una fase de desarrollo muy inicial, en el que se permite la definición de las tareas, la visualización en formato GANTT y las dependencias entre tareas. Mr Project, además, permite la definición de recursos y su asignación a tareas, aunque no comprueba los conflictos entre ellos.

TaskJuggler es una aplicación con licencia GPL, pero que está más cerca de las aplicaciones comerciales debido a su gran cantidad de características, como diagramas de descomposición del trabajo, planificaciones condicionales, administración de recursos y planificación automática, análisis de riesgos y de beneficios, etc.

Por último, dotProject tiene un enfoque diferente. Es una aplicación con licencia BSD programada en PHP que se ejecuta en un servidor web y a la que se accede mediante un navegador. Podemos definir empresas, departamentos, personal, proyectos, tareas, crear las planificaciones, controlar el seguimiento, abrir tickets de errores y mantenimiento, etc.

Todas estas herramientas, en mayor o menor medida, permiten la gestión de las tareas de un proyecto, asignando recursos a cada tarea y comprobando que no se violan las restricciones. Lo que ninguno de ellos tiene es lo que este proyecto se propone: la creación de la planificación para un nuevo proyecto sin que se deba invertir mucho tiempo en la generación del nuevo plan, pero consiguiendo que éste sea lo más preciso posible.

4.- Descripción de la situación actual

Para dar solución al problema visto en el apartado de objetivo, lo que vamos a construir es una herramienta que tenga integrado tanto la gestión de los procesos, como la gestión de los proyectos.

Vamos a construir el proyecto GesProSoft sobre un prototipo previo llamado SPK desarrollado por el grupo de trabajo Erabaki. El prototipo SPK tiene definidos cinco roles:

- SPK User: Cualquier usuario de SPK. Este usuario tiene definidos dos casos de uso, hacer login y logout del sistema.
- SPK Administrator: Responsable de la gestión de las cuentas de acceso de los Managers. Este usuario tiene definidos los casos de uso necesarios para gestionar las cuentas de usuario de los

SPK Managers

- SPK Client: Cualquiera que use el SPK para trabajar sobre un proyecto. No tiene casos de uso propios
 - ♦ SPK Manager: Responsable de gestionar sus propios proyectos y sus propios trabajadores. Tiene definidos los caso de uso necesarios para la gestión de sus proyectos y las cuentas de usuario de sus Workers.
 - ♦ SPK Worker: Responsable de ejecutar las tareas que se le asignan. Tiene los casos de uso para confirmar que sus tareas se han realizado.

Por lo tanto, GesProSoft implementa un subconjunto de funcionalidades para SPK, concretamente funcionalidades enmarcadas en el rol SPK Manager.

GestProSoft implementa las funcionalidades relacionadas con la generación de planes. Estos planes se generan a partir de procesos definidos (extraídos de conocimiento previo) una vez se fije la fecha de inicio de un proyecto.

Afortunadamente podemos disponer de algunos procesos ya definidos, como los indicados por la metodología del Rational Unified Process (RUP)[2] que se mencionará más en detalle en la sección 5.1 por haber sido utilizado como método en este proyecto.

Para la definición de los procesos vamos a emplear la herramienta Protège. Esta aplicación se utiliza para crear bases de conocimiento en función de clases e instancias. Con Protège vamos a definir cada uno de los tipos de procesos y cada una de las entidades necesarias para la creación del nuevo proyecto. De este modo, en cada proceso tendremos definidas una lista de tareas representadas mediante un diagrama, las cuales tienen como dato la duración de dicha tarea. Las dependencias están contempladas según la forma que pueda tener el diagrama. Podemos ver un ejemplo en la Figura 3, en la que si seguimos el flujo del diagrama, tenemos que después de realizarse una tarea Task 1, nos indica que pasamos a la realización de otra tarea Task 2. En este caso tenemos una dependencia directa, de manera que la realización de la tarea Task 2 comenzará justo después de la finalización de la tarea Task 1.

Es muy importante que queden reflejadas las dependencias entre tareas, ya que si se decide alargar la duración de la tarea Task 1, debemos tener en cuenta que las tareas que dependen de ella (en el ejemplo nuestro sería la tarea Task 2), deben de retrasar a su vez el inicio, y por tanto también su finalización de manera recursiva, ya que al retrasar la tarea Task 2, debemos de retrasar a su vez to-

das la tareas que sean dependientes de la tarea Task 2, etc. Los datos sobre las dependencias nos van a ser de gran utilidad no tanto para la creación de un nuevo plan, sino para el mantenimiento de la planificación una vez de que esta esté generada.

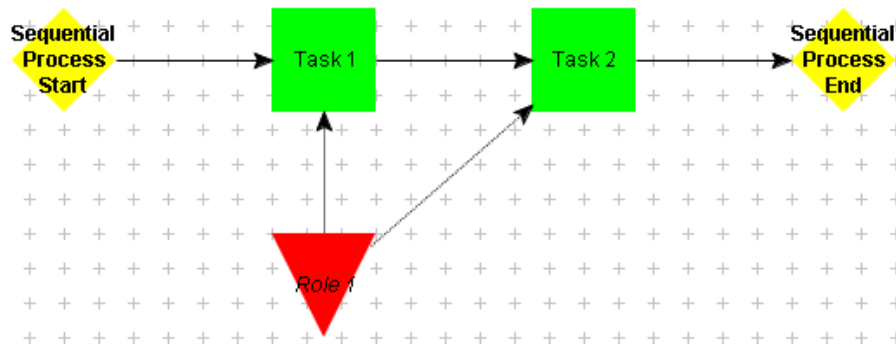


Figura 3: Proceso de tipo secuencial

Por lo tanto, podremos comprobar que el tener un diagrama base de un proceso tipo del que partir nos va a facilitar considerablemente la tarea de creación de un nuevo plan para el nuevo proyecto.

De este modo, gracias a la nueva aplicación web GesProSoft, todos los usuarios que deben trabajar con herramientas para realizar la gestión y planificación de sus proyectos software van a conseguir un ahorro de tiempo considerable a la hora de tener que hacer una planificación inicial. Este hecho trae consigo muchas consecuencias favorables, ya que a la hora de tener que realizar un presupuesto ajustado sobre un tipo de proyecto, podremos disponer con facilidad de un presupuesto inicial muy cercano al real, ya que podemos basarnos en los datos que tengamos de proyectos de magnitud similar al proyecto solicitado. Con ello, conseguimos no perder mucho tiempo en la elaboración de un presupuesto, así como obtener un valor del mismo que no nos implique pérdidas en la futura elaboración del producto final.

Ya que todos los elementos que han sido integrados en el proyecto están sujetos a licencias muy poco restrictivas y susceptibles de uso comercial (LGPL), los usuarios de esta aplicación únicamente deberían pagar el precio que se estipule por el uso de la misma.

5.- Normas y referencias

En la elaboración del proyecto GesProSoft hemos seguido distintas directrices a la hora de la planificación y la documentación de dicho proyecto.

5.1.- Método

Para conseguir desarrollar el producto con unas garantías de calidad, hemos seguido un método muy conocido como es el Rational Unified Process (RUP)[2], en el cual nos viene definido un ciclo de vida para el proyecto que vamos a realizar, tal y como podemos ver en la Figura 4.

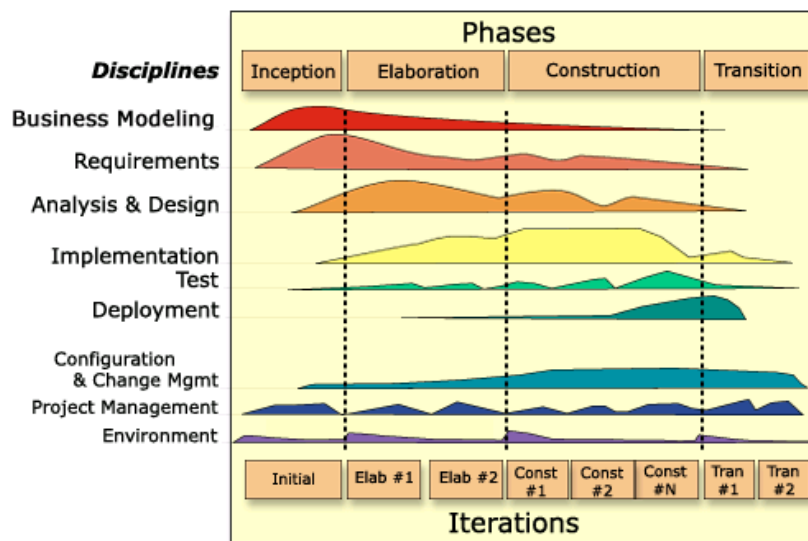


Figura 4: Fases definidas en el RUP para la elaboración de un producto

Si observamos la Figura 4, en el eje horizontal queda representado tanto el tiempo como los aspectos del ciclo de vida del proceso del RUP; y en el eje vertical vemos las disciplinas que agrupan las actividades a desarrollar.

5.1.1.- Procesos, roles, actividades y artefactos

Dentro cada una de las disciplinas que se enumeran en la Figura 4 tenemos definido un proceso. Vamos a ver en la Figura 5 un ejemplo mostrando el proceso que viene definido para la disciplina *Requirements*.

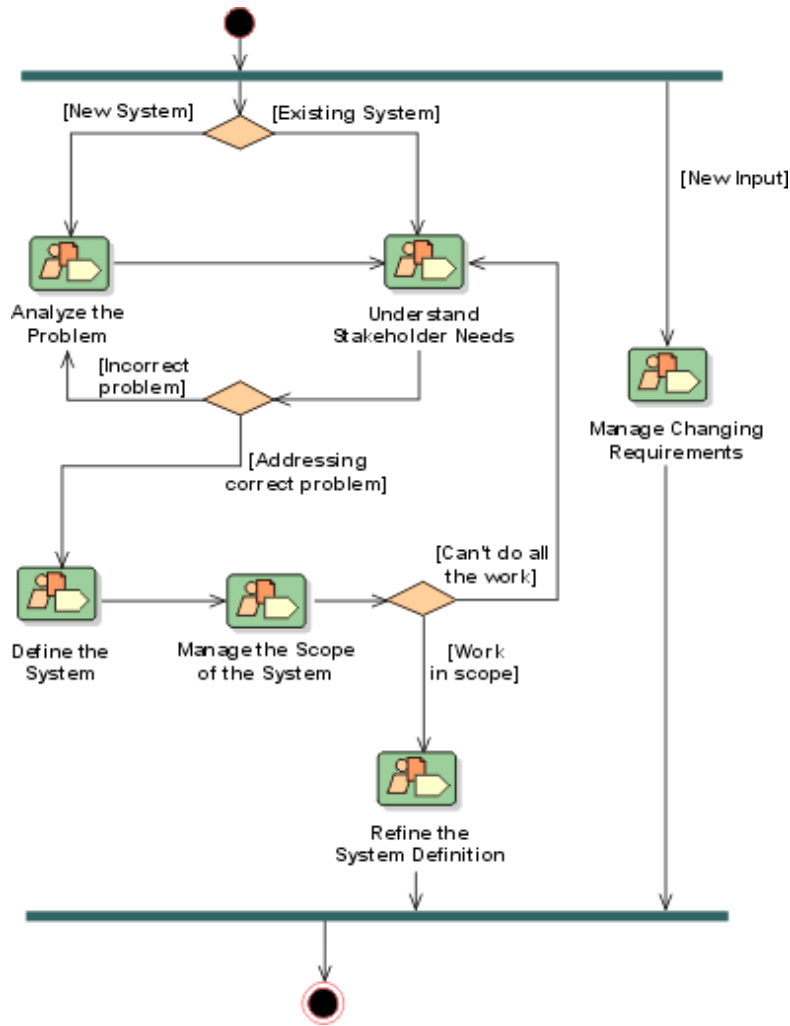


Figura 5: Nodo del proceso definido para Requirements

Podemos ver como dentro de este proceso tenemos definido un diagrama en el cual encontramos diversos nodos. Cada uno de estos nodos tiene información sobre las actividades que deben de realizarse, los artefactos a rellenar y los roles que deben de ejecutar cada uno de ellos.

De este modo, quedan desglosadas cada una de las tareas a realizar a lo largo de las disciplinas del proyecto. En la Figura 6 podemos ver todos los componentes que forman parte del nodo *Analyze the Problem*. En la parte central tenemos definido el rol *System Analyst* que va a hacerse cargo de la ejecución de cada una de las tareas definidas en este nodo. Este rol va a mantener comunicaciones con tres roles representados en color rojo *Customer*, *End User*, *Stakeholder*. Las actividades que debe realizar el *System Analyst* se muestran mediante unas flechas en la parte central, y los artefactos a rellenar son los que se encuentran alrededor del diagrama central exceptuando los roles definidos a la izquierda.

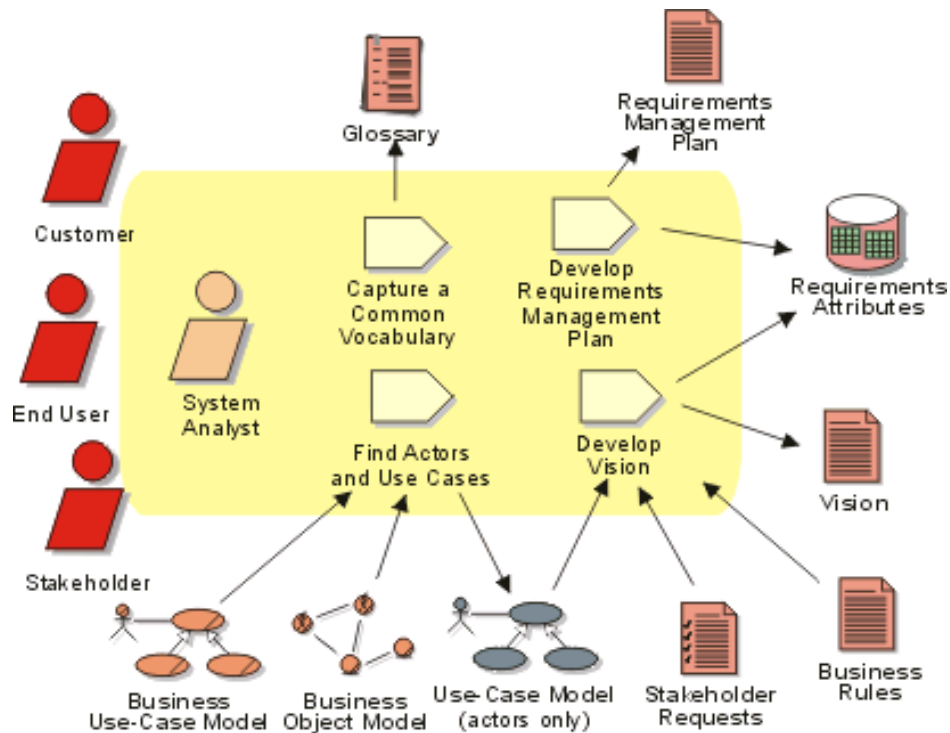


Figura 6: Detalle del trabajo a realizar

En cada una de las fases definidas por el método RUP debemos de ir rellenando una serie de artefactos mínimos según nos indica la Figura 7, consiguiendo de esta manera, dejar reflejado mediante documentos, diagramas, etc. cada uno de los pasos que se han ido dando a lo largo de todo el ciclo de vida del proyecto.

Estos artefactos han sido rellenados en su plenitud para el proyecto GesProSoft, dejando en el sitio web reflejadas cada una de las transiciones que hemos ido dando a lo largo de todo el ciclo de vida.

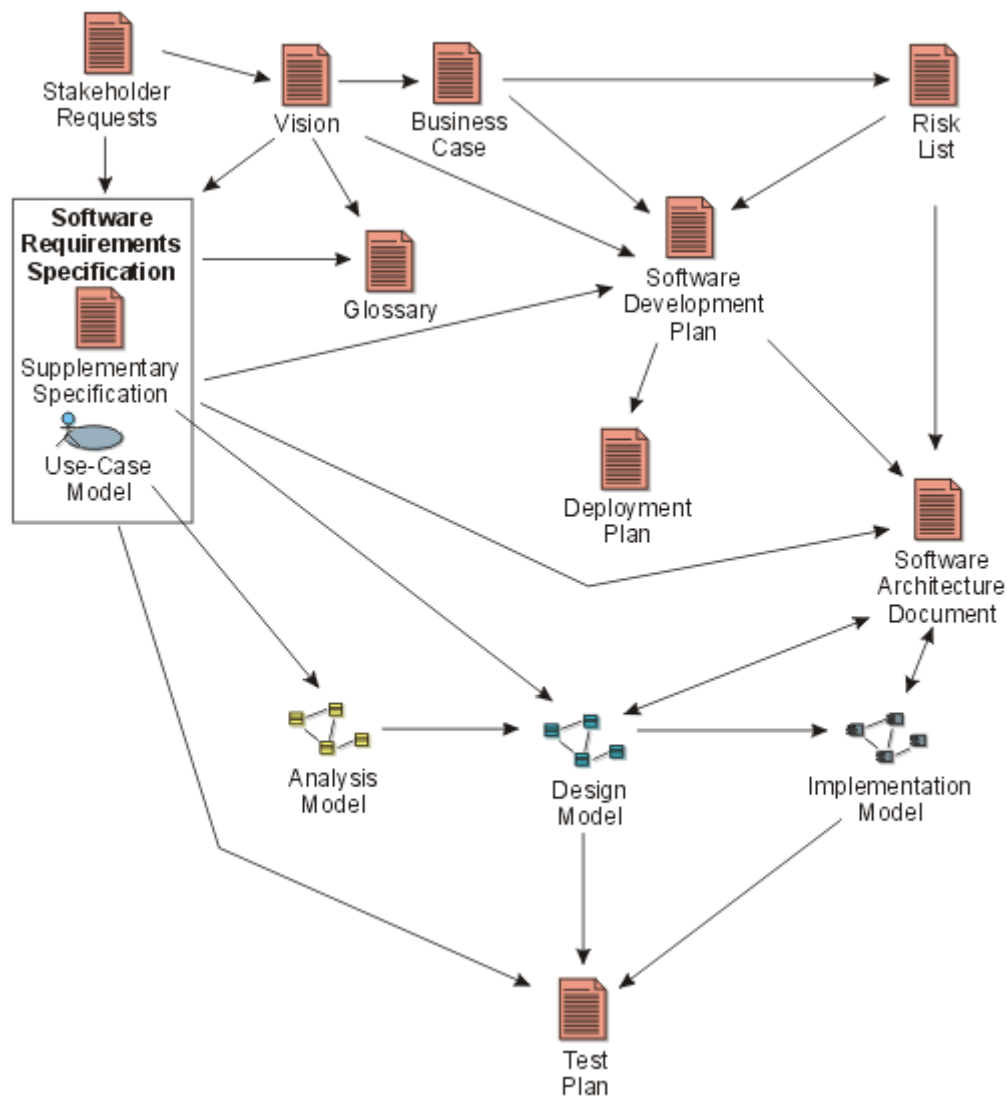


Figura 7:Conjunto de artefactos mínimos definidos en el RUP y relaciones existentes entre ellos

5.2.- Normas

A la hora de redactar la memoria del proyecto, decidimos seguir la norma de AENOR *UNE 157 801* que, tal y como hemos comentado anteriormente, es aplicable a la elaboración de proyectos informáticos. El cumplimiento de la norma fue uno de los requisitos exigido por el cliente, estudiamos el hecho de su utilización, y finalmente nos pareció muy adecuado aplicar una norma así y tener definido también la estructura de los documentos de salida del proceso de desarrollo.

5.3.- Bibliografía

1: Cohen, S., Kart, R., Peterson, S., & Withey, J.: *Models for Domains and Architectures: A Prescription for Systematic Reuse*.

- 2: Kruchten P.: *The Rational Unified Process. An Introduction.*
- 3: Microsoft Corporation: *Microsoft SQL Server TM. El sistema empresarial de administración y análisis de bases de datos relacionales.* <http://www.microsoft.com/latam/sql/64bit/productinfo/overview.asp>.
- 4: Eclipse Team: *Eclipse Web Site.* www.eclipse.org.
- 5: JBoss Group: *JBoss Web Site.* www.jboss.org.
- 6: Object Management Group, Inc. : *UML Resource Page.* <http://www.omg.org/uml/>.
- 7: Apache Software Foundation, The Apache Jakarta Project, Struts,
- 8: Ed Roman, Scott Ambler, Tyler Jewell: *Mastering Enterprise JavaBeans.*
- 9: Christian Bauer and Gavin King: *Get started with Hibernate.* http://www.javaworld.com/javaworld/jw-10-2004/jw-1018-hibernate_p.html.
- 10: JBoss Group: *Hibernate Web Site.* <http://www.hibernate.org>.
- 11: JBoss Group: *Hibernate Reference Documentation v3.* http://www.hibernate.org/hib_docs/v3/reference/en/html/.
- 12: Stanford University: *Protégé Web Site.* <http://protege.stanford.edu/>.
- 13: Grupo de Gestión de la Tecnología (GETEC): *Gestión de proyectos.* <http://www.getec.et-sit.upm.es/docencia/gproyectos/gproyectos.htm>.
- 14: M. Chantal Pérez Hernández: *Explotación de los corpórea textuales informatizados para la creación de bases de datos terminológicas basadas en el conocimiento.* <http://elies.rediris.es/elies18/531.html>.
- 15: María Alejandra Hinojosa: *Diagrama de GANTT.* <http://www.gestiopolis.com/recursos/documentos/fulldocs/ger/diaggantaleja.htm>.
- 16: Real Academia Española: *RAE Web Site.* <http://www.rae.es>.
- 17: Community: *Wikipedia.* <http://www.wikipedia.org>.
- 18: PlexityHide: *GTP.NET.* <http://www.plexityhide.com/GTP.NET.htm>.
- 19: Minute Man Systems: *Minute Man Project Management Software for Windows.* <http://www.minuteman-systems.com>.
- 20: American NETRONIC, Inc.: *VARCHART JGANTT.* <http://www.netronic-us.com/jGANTT.asp>.
- 21: dotProject Developers: *dotProject.net.* <http://www.dotproject.net/>.

22: David Gilbert and Thomas Morgner: *JFreeChart*. <http://www.jfree.org/jfreechart/index.php>.

6.- Definiciones y abreviaturas

Artefacto

Un artefacto es un producto de trabajo intermedio o final que es creado y usado durante un proyecto. Un artefacto es usado para capturar y mostrar información del proyecto y puede ser un documento, un modelo o un elemento del modelo.

Caso de desarrollo

Proceso de ingeniería del software utilizado por la organización que la lleva a cabo. Es desarrollado como una configuración del producto del Proceso Unificado, y adaptado a las necesidades del proyecto.

Caso de uso

Una descripción del comportamiento del sistema, en términos de secuencias de acciones. Un caso de uso debería producir un resultado notable a un actor. Un caso de uso contiene todos los flujos alternativos de eventos relacionados con la producción de un "resultado notable".

Ciclo de vida

Todo proyecto de ingeniería tiene unos fines ligados a la obtención de un producto, proceso o servicio que es necesario generar a través de diversas actividades. Algunas de estas actividades pueden agruparse en fases porque globalmente contribuyen a obtener un producto intermedio, necesario para continuar hacia el producto final y facilitar la gestión del proyecto. Al conjunto de las fases empleadas se le denomina "ciclo de vida".

Despliegue

Una disciplina en el proceso de ingeniería del software, cuyo propósito es asegurar una transición exitosa del sistema desarrollado a sus usuarios.

Diagrama GANTT

Modo gráfico definido para representar mediante barras las tareas que forman parte de una planificación. Mediante su utilización, se trata de realizar una programación de las tareas conforme a un

calendario, pudiendo de este modo visualizar la duración de cada actividad, con sus fechas de inicio y fin.

Disciplina

Una agrupación de tareas u artefactos relacionados semánticamente a un área de interés del proceso. Es decir, un conjunto de tareas y artefactos relacionados por el objetivo de los mismos. Hay nueve disciplinas en el RUP: Modelado Comercial, Análisis & Diseño, Implementación, Prueba, Despliegue, Configuración & Gestión de Cambios, Gestión del Proyecto, Entorno.

Elaboración

Segunda fase del proceso donde la visión del producto y su arquitectura son definidas.

Enterprise Java Beans

Una arquitectura para el desarrollo y despliegue de aplicaciones basadas en objetos distribuidos transaccionales, software de componentes del lado del servidor.

Entorno

Una disciplina en el proceso de la software-ingeniería cuyo propósito es definir y manejar el entorno en el que el sistema está desarrollándose. Incluye descripciones del proceso, dirección de la configuración, y herramientas de desarrollo.

Fase

Tiempo entre dos hitos principales del proyecto, durante el cual se reúne un conjunto bien definido de objetivos, los artefactos están completos y las decisiones son tomadas para pasar o no pasar a la siguiente fase.

Framework

Una micro-arquitectura que mantiene una plantilla extensible para aplicaciones dentro de un dominio específico.

Gestión

Una disciplina en el proceso de la ingeniería del software cuyo propósito es planificar y gestionar desarrollo del proyecto.

Gestión de proyectos

Disciplina de investigación en la que para cada uno de los proyectos se deben tener en cuenta todos los elementos de gestión (identificación de objetivos, gestión de recursos humanos, logística, identificación de riesgos, financiación, etc.).

Herramienta

Software que podemos utilizar para la realización de diversas y distintas tareas, y que nos va a facilitar considerablemente el trabajo que deberíamos realizar, disminuyendo tanto en esfuerzo como el tiempo a invertir en la realización de dichas tareas, obteniendo unos resultados de alta calidad.

Hibernate

Es una herramienta de persistencia objeto/relacional para entornos Java que permite la traducción de un dato de un modelo de orientación a objetos a un modelo relacional con un esquema basado en SQL, y viceversa.

Hito

Punto en el que una iteración formalmente finaliza; corresponde a un punto de versión.

Implementación

Una disciplina en el proceso de la software-ingeniería cuyo propósito es llevar a cabo y probar las clases.

Implicado

Un individuo que es afectado materialmente por el resultado del sistema.

Inicio

Primera fase del Proceso Unificado en que la idea inicial, requerida para la propuesta, para la generación anterior se trae al punto de ser (por lo menos internamente) consolidada para entrar en la fase de la elaboración.

J2EE

(Java 2 Enterprise Edition), plataforma de programación basada en lenguaje Java y en Enterprise Java Beans

JBoss

JBoss es un servidor de aplicaciones J2EE desarrollado por JBoss Inc.. Es un contenedor de EJBs y JSPs que, llevando integrado el servidor TomCat, es un sistema completo para la implementación de aplicaciones sobre plataforma J2EE.

JSPs

Los JSPs son páginas HTML enriquecidas con código Java que son interpretadas en el servidor y el resultado se envía de vuelta al navegador que solicitó el documento.

Modelo de caso de uso

Un modelo que describe los requerimientos funcionales del sistema en términos de casos de uso.

Modelado comercial

Una disciplina en el proceso de ingeniería del software que abarca todas las técnicas de modelado que puedas utilizar para visualmente modelar un negocio.

MVC

El Modelo-Vista-Controlador (MVC) es una arquitectura software que separa el modelo de datos de la aplicación, la interfaz de usuario y la lógica de control en tres componentes distintos. De este modo, modificaciones en el componente de visualización puede realizarse con un impacto mínimo en el componente del modelo de datos.

El MVC suele verse como un patrón de diseño software. Sin embargo, MVC está más implicada en la arquitectura de la aplicación que un patrón de diseño típico.

MVC2

Adaptación del modelo MVC a aplicaciones Web.

Ontología

Una ontología ha de verse como un entendimiento común y compartido de un dominio, que puede comunicarse entre científicos y sistemas computacionales. Ésta última característica, el hecho de que puedan compartirse y reutilizarse en aplicaciones diferentes, explica en parte el gran interés suscitado en los últimos años en la creación e integración de ontologías (Steve et al. 1998a, b).

Patrón

Estándar utilizable en el diseño de aplicaciones reconocido como eficaz y que aprovecha las posibilidades de la tecnología utilizada.

Planificación

La planificación de un proyecto debe afrontarse de manera adecuada para que al final del mismo se pueda hablar de éxito. Se deben planificar todas las tareas que se darán cita a lo largo del proyecto, intentando prever un comportamiento futuro. A cada una de las tareas, se le debe asignar un espacio de tiempo en el que ejecutarse, los recursos necesarios para poder realizarla, el equipo de trabajadores encargado de hacerla, etc.

Proceso

Conjunto de actividades o eventos que se realizan o suceden con un determinado fin.

Protège

Protège es un editor de ontologías y un framework basado en el conocimiento, de código libre y abierto. Protège está basado de Java, es extensible y proporciona una base para la construcción de aplicaciones basadas en el conocimiento.

Proyecto

Un conjunto de actividades interdependientes orientadas a un fin específico, con una duración predeterminada. Completar con éxito el proyecto significa cumplir con los objetivos dentro de las especificaciones técnicas, de costo y de plazo de terminación. Un proyecto se puede considerar como una instancia de un proceso.

Prueba

Una disciplina en el proceso de ingeniería del software cuyo propósito es la integración y la prueba del sistema.

Rational Unified Process

El Rational Unified Process (RUP) es un método iterativo de desarrollo de software creado por la Rational Software Corporation. Describe cómo desplegar software de manera efectiva utilizando técnicas bien probadas. No es un proceso rígido, sino un framework o meta modelo. Incluye un gran número de actividades y ha sido diseñado para ser adaptado, es decir, seleccionar sólo las caracterís-

ticas necesarias para cada proyecto en particular, considerando su tamaño y tipo. Ha sido comprobado que es aplicable a grandes equipos de desarrollo de software trabajando en proyectos enormes.

Requerimientos o requisitos

Una disciplina del proceso de ingeniería del software, cuyo propósito es definir lo que el sistema debería hacer. Las actividades principales son desarrollar una visión, un modelo de caso de uso y las especificaciones de los requerimientos del sistema.

Riesgo

Una preocupación continuada o próxima que tiene una probabilidad considerable de afectar adversamente al éxito de hitos principales.

RUP

Rational Unified Process

Servlets

Objetos de Java los cuales se ejecutan en el servidor como respuesta a una petición del navegador. La salida que generan, normalmente una página HTML, se envía de vuelta al navegador.

Struts

Struts es un conjunto de clases, servlets y etiquetas JSP que facilitan un diseño reutilizable del modelo MVC2. Esta definición implica que Struts es un framework, más que una biblioteca, pero Struts también contiene una extensa biblioteca de etiquetas y clases útiles que trabajan independientemente del framework.

Tarea

Trabajo que debe de realizarse en un periodo de tiempo limitado.

TomCat

TomCat, desarrollado por el grupo Apache, es un servidor de Servlets y JSPs. Permite la ejecución de Servlets e interpreta las páginas JSP para servir las al cliente. Es la implementación por referencia para estas tecnologías.

Transacción

Una unidad de proceso consistente en uno o más programas de aplicación iniciados por una única petición. Una transacción puede requerir la inicialización de una o más tareas para su ejecución.

UML

Unified Modeling Language

Unified Modeling Language (UML)

Un lenguaje para la visualización, especificación, construcción, y documentación de los artefactos de un sistema de software intensivo.

Versión

Un subconjunto del producto final que es el objeto de evaluación en un hito mayor. Una versión es un ejecutable estable del producto, junto con cualquier artefacto necesario para usar este ejecutable, como notas de la versión o instrucciones de la instalación. Una versión puede ser interna o externa. Una versión interna sólo es usada por la organización de desarrollo, como parte de un hito, o para una demostración a usuarios o clientes. Una versión externa (o entrega) se entrega a los usuarios finales. Una versión necesariamente no es un producto completo, puede ser simplemente un paso en el camino, con su utilidad sólo medida desde una perspectiva de la ingeniería. Las versiones actúan como una función forzosa que maneja el equipo de desarrollo para conseguir cierre a los intervalos regulares y evita el síndrome "90% done, 90% remaining".

Visión

La visión del cliente o usuario del producto a desarrollar, especificado al nivel de las necesidades claves del implicado y características del sistema.

7.- Requisitos iniciales

El producto que vamos a construir debe de ser de calidad. Entendiendo que el desarrollo bajo la disciplina de un proceso definido aumenta la calidad del producto final, nos hemos decidido a utilizar el Rational Unified Process (RUP) como proceso de desarrollo. Tal y como hemos visto en el apartado anterior (normas y referencias), hemos seguido cada una de las fases de inicio, elaboración, construcción y transición tal y como vienen definidas en el RUP, hemos desarrollado los artefactos correspondientes. Al desarrollar el proyecto basándonos en un método de desarrollo definido, lo

dota de un valor adicional en cuanto a la calidad del proceso y producto final. Y, aunque es de todos conocido que la realización de proyectos bajo este tipo de disciplina es más que conveniente, en muy pocos proyectos informáticos llega a aplicarse.

El cliente solicitó que dada únicamente la fecha de inicio para un proyecto, y seleccionado el tipo de proceso que más se ajustase al proyecto dado, se generara un diagrama GANTT con las tareas, y los tiempos de duración ya definidos en dicho proceso. Posteriormente, se podrán realizar ajustes en dichos tiempos siempre que se crea oportuno, consiguiendo acercarse más a la realidad de los tiempos de cada proyecto.

La implementación se va a realizar partiendo de una base en la que ya se trataban algunas funcionalidades, como la de autenticación de los usuarios, dar de altas a nuevos proyectos, etc. Este hecho hace que a la hora de continuar con la implementación para aportar nuevas funcionalidades, se decida utilizar el software que se usó para la implementación del producto previo, evitando de este modo posibles conflictos que dieran lugar a problemas generando un incorrecto funcionamiento de la aplicación web.

8.- Alcance

Se trata de elaborar el proyecto de definición de una solución para la gestión de procesos y proyectos, apoyándonos en prototipos parciales para valorar el coste de construcción y de la tecnología.

Por lo tanto, al cliente se le van a entregar, a la finalización del proyecto:

- Un manual de usuario en el que va a poder ver paso a paso la manera de utilizar, de una forma correcta, cada una de las funcionalidades que la herramienta le va a aportar, para que no tenga ningún tipo de problema en su utilización.
- Un prototipo que tenga un subconjunto de las funcionalidades que nos puede aportar la herramienta para la gestión de proyectos MS Project, pero dotando al proyecto GesProSoft, además de la gestión de proyectos, la gestión de los procesos, obteniendo un prototipo más avanzado y más interesante para los desarrolladores del software. Gracias a la gestión de procesos se va a poder tener en un breve espacio de tiempo un diagrama GANTT con la definición de tareas y tiempos de duración de las mismas. Ésto le va a ahorrar mucho tiempo al usuario en la creación de la gestión de proyectos respecto al tiempo que necesitaría usando herramientas como el MS Project, que solo tienen gestión de proyectos.

Todo lo comentado anteriormente puede ser de mucha utilidad, ya que cuando un cliente que desea que se le desarrolle una nueva aplicación le pida un presupuesto para un nuevo proyecto, no va a invertir mucho tiempo en generar una planificación, ya que, dada la fecha de inicio y el tipo de proceso que se va a seguir, se obtiene un diagrama GANTT inicial, al que posteriormente podrá ajustar algunas de las fechas, si es que lo cree necesario.

En el caso en el que el proyecto ya tenga un plan definido, porque lo ha creado con anterioridad, se le mostrará un diagrama GANTT con las tareas y los tiempos de duración de las mismas.

Los procesos, dado que están definidos, posibilitan la medición de resultados de anteriores proyectos. Dicha medición permite ajustar las duraciones de las tareas incluidas de manera suficientemente precisa. Cuantos más proyectos se hayan realizado más precisamente se podrá estimar la duración de las tareas, lo que le ayudará a poder dar presupuestos muy ajustados a la realidad.

Dado que lo que se pedía era la búsqueda de una solución, y que el proyecto va a terminar con la fase de elaboración, al cliente se le mostrará un prototipo con una funcionalidad mínima para la visualización del estado del proyecto mediante diagramas GANTT. Como el objetivo del proyecto es ofrecer una solución cuyos elementos hayan sido suficientemente probados, se ha prestado más énfasis en la funcionalidad.

Por lo tanto, aunque sí que hacemos una fase de construcción dentro de nuestro proyecto, ésta se refiere a la creación de prototipos, con lo cual, siendo estrictos con las fases del RUP, es como si nuestro proyecto se quedase en la fase de elaboración, en la que generamos prototipos y evaluamos el coste de una versión comercial. En una fase posterior, se tratará de realizar una aplicación web más completa y comercial, aportando en ella mejoras y funcionalidades de las que carecen en los prototipos elaborados.

9.- Hipótesis y restricciones

Durante toda la elaboración del proyecto GesProSoft vamos a seguir el método concreto del RUP, gracias al cual, vamos a conseguir desarrollar un software con un mínimo de calidad y bien documentado.

Como el proyecto GesProSoft es una ampliación a una aplicación web que ya estaba hecha por parte de un grupo de trabajo, y la implementación de esta fase previa se realizó utilizando Java, consideramos que podía obtenerse una solución adecuada con su utilización. Tal y como se había co-

mentado anteriormente, todo el software que ha sido utilizado para la elaboración de GesProSoft, era el software que se utilizó en el prototipo inicial creado por el grupo Erabaki, y era el aprobado por el cliente.

Por otra parte, el hecho de utilizar una herramienta como el Rational Rose para la realización de los casos de uso, análisis y diseño aplicado a nuestro proyecto, fue elegida debido a que había sido utilizada en la parte previa del desarrollo de la aplicación web, con lo cual era la mejor opción a utilizar en el desarrollo del módulo adicional, para que de ese modo se pudiera reutilizar la parte de análisis y diseño de la aplicación original. Además, nos aportaba la ventaja de disponer en ficheros texto y documentos Word o html absolutamente toda la información de un proyecto, con las ventajas de automatización posterior que ello implica.

Otra de las restricciones que fue impuesta por el cliente, fue la de que todo el software adicional que se utilizase para la elaboración del proyecto GesProSoft, debía de ser software libre con licencias no GPL, tales como LGPL, Mozilla o BSD, para no tener que estar sujeto a ningún tipo de licencia en el caso en el que la aplicación fuera comercial en un futuro.

Del mismo modo, el cliente también puso como restricción que el formato que debía de tener la memoria a entregar debía de seguir la norma AENOR *UNE 157 801* "Criterios generales para la elaboración de proyectos de Sistemas de Información".

10.- Estudios de alternativas y viabilidad

Para la fase de diseño e implementación, como hemos mencionado anteriormente, el uso de las herramientas que hemos utilizado no han sido escogidas exclusivamente por el desarrollador, sino que hemos tenido en cuenta las herramientas que fueron usadas para la generación de la aplicación previa a GesProSoft realizada por el grupo Erabaki.

Para el proyecto GesProSoft, además de las herramientas para el desarrollo de una aplicación web, debíamos de conseguir algún módulo adicional, que sirviera para la representación gráfica de diagramas GANTT. Por lo tanto, realizamos un estudio de todo el software que nos podía ser de utilidad para mostrar diagramas GANTT. En dicho estudio, apareció como mayor problema que no había mucho software para mostrar diagramas GANTT, y que además, algunos de ellos tenían licencias del tipo GPL, con lo cual, quedaban totalmente descartadas. Este era el caso de algunas como MinuteMan Project Management Software for Windows (<http://www.minuteman-systems.com>),

GTP.NET (<http://www.plexityhide.com/GTP.NET.htm>), VARCHART_JGANTT (<http://www.ne-tronic-us.com/jGANTT.asp>).

Finalmente, teníamos dos alternativas distintas para poder hacer la representación de los diagramas GANTT. Una de ellas era el dotproject con licencia BSD (<http://www.dotproject.net/>), y la otra el jfreechart con licencia LGPL (<http://sourceforge.net/projects/jfreechart>).

El jfreechart tenía la implementación de su código realizada en Java y además estaba bastante documentado. Tenía una demo en la cual se mostraban tanto las tareas planificadas como el desarrollo real de las mismas. Esto era una funcionalidad muy interesante para nuestro proyecto, ya que nosotros teníamos guardados en la base de datos ambos tipos de datos sobre las tareas. Tenía como problema que no nos mostraba mediante ningún tipo de flecha las dependencias entre las tareas, solo mostraba las tareas con las fechas en las que se desarrollan y la duración de las mismas.

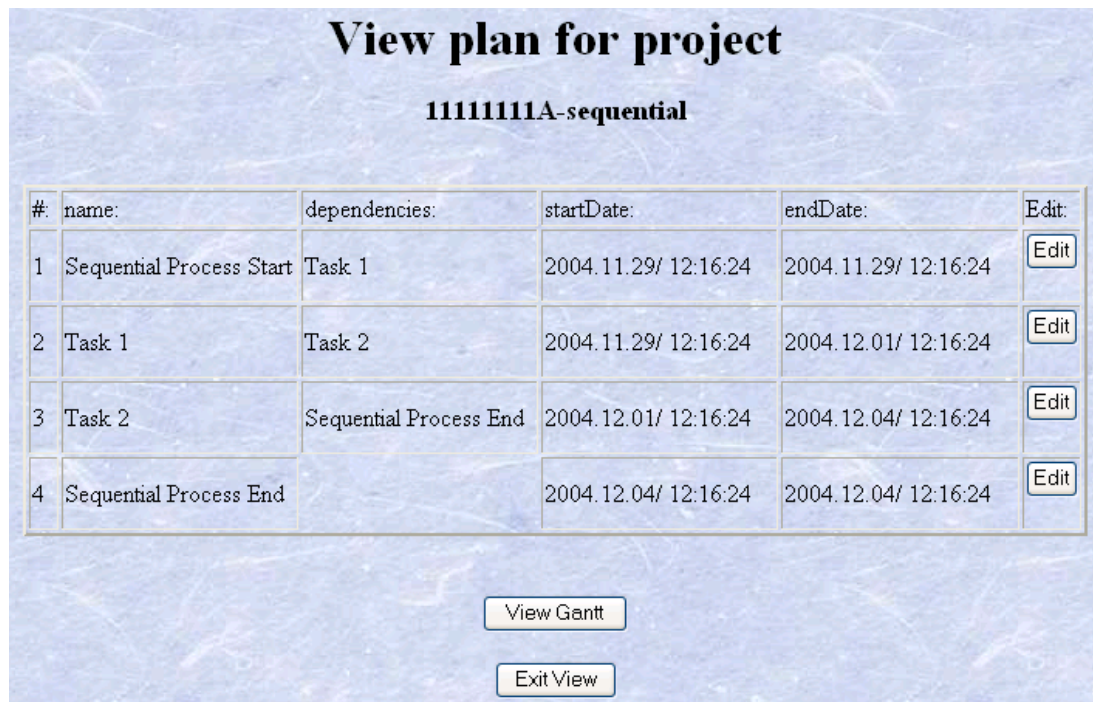
Por otro lado evaluamos el utilizar el dotproject, que aunque nos mostraba un diagrama GANTT únicamente con las tareas planificadas, sí que nos mostraba las dependencias entre las tareas. Para utilizar el dotproject había un problema, ya que era una aplicación independiente implementada en PHP, por lo que la integración con nuestra aplicación preexistente era inviable.

Por lo tanto, decidimos finalmente utilizar el jfreechart, y mitigar el problema de mostrar las dependencias entre las tareas mediante otra alternativa. En este caso, decidimos mostrar en una tabla las tareas y las dependencias existentes entre ellas, y posteriormente dar la opción de mostrar mediante un diagrama GANTT la planificación, aunque no aparezcan en él las dependencias de ningún modo gráfico.

11.- Descripción de la solución propuesta

Para el desarrollo del código, dado que partíamos de una aplicación base, a la que teníamos que dotar de una funcionalidad extra, el código estaba en lenguaje Java, y para su desarrollo se había utilizado como herramienta Eclipse, usando el plug-in MyEclipse. Como contenedor de aplicaciones J2EE se empleó JBoss. El sistema de gestión de bases de datos era Microsoft SQL Server y la herramienta de adquisición del conocimiento, Protège. Por lo tanto, y dado a que íbamos a ampliar la funcionalidad de la aplicación dada, decidimos seguir con el mismo tipo de herramientas.

Para la representación de los diagramas GANTT decidimos utilizar, tras un estudio realizado como vimos anteriormente, el jfreechart, y debíamos de solventar el problema de que no se mues-



#	name:	dependencies:	startDate:	endDate:	Edit:
1	Sequential Process Start	Task 1	2004.11.29/ 12:16:24	2004.11.29/ 12:16:24	<input type="button" value="Edit"/>
2	Task 1	Task 2	2004.11.29/ 12:16:24	2004.12.01/ 12:16:24	<input type="button" value="Edit"/>
3	Task 2	Sequential Process End	2004.12.01/ 12:16:24	2004.12.04/ 12:16:24	<input type="button" value="Edit"/>
4	Sequential Process End		2004.12.04/ 12:16:24	2004.12.04/ 12:16:24	<input type="button" value="Edit"/>

Figura 8: Visualización mediante una tabla del plan basado en un proceso secuencial

tran las dependencias dentro del diagrama de alguna manera, con lo que elegimos, sacar por un lado mediante una tabla todos los datos de las tareas, incluidas las dependencias entre las mismas, y tras mostrar todo en dicha tabla tal y como podemos ver en la Figura 8, y además, dar la posibilidad de ver las tareas de una manera más gráfica en un diagrama GANTT como muestra la Figura 9, aunque en este no aparecerán las dependencias entre las tareas.

Tal y como podemos ver en la Figura 9, en el diagrama GANTT se muestran tanto los tiempos de las tareas planificadas en color rojo como los tiempos reales de ejecución de las tareas en azul. Este tipo de visualización de las dos clases de tareas la realizan también MS Project, SuperProject, ToutDoux, Mr Project y TaskJuggler. La herramienta dotProject no nos muestra las fechas de la ejecución real de las tareas, aunque si que nos muestra dentro de la barra de cada tarea el porcentaje que ha sido realizado de cada una de ellas. Sin embargo, en todas ellas se muestran dentro del diagrama GANTT las dependencias de manera gráfica mediante la utilización de flechas, mientras que en el prototipo realizado para dentro de la fase de elaboración no consigue integrar las dependencias dentro del diagrama GANTT.

Vamos a mencionar algunos de los puntos a mejorar en una versión comercial para el producto GesProSoft, que quedan fuera del alcance de este proyecto. Por un lado, podemos fijarnos detenidamente en la parte más gráfica, como son las interfaces con las que van a interactuar los usuarios.

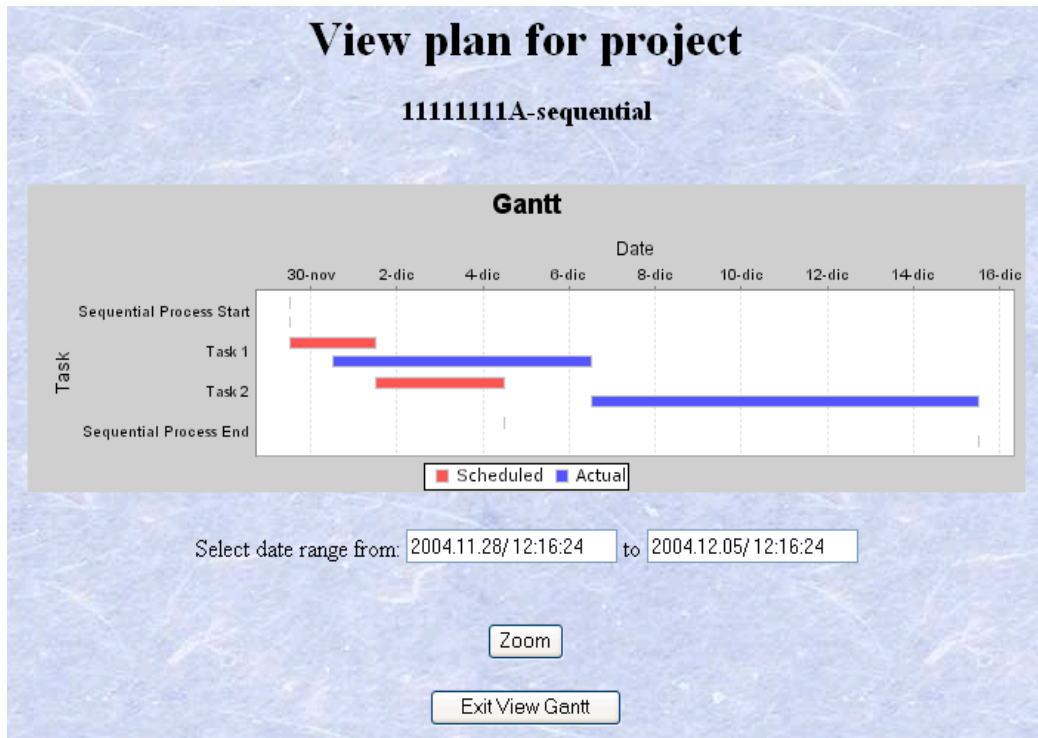


Figura 9: Visualización mediante jfreechart de los datos mostrados en la

Una mejora muy interesante es integrar dentro de la imagen que se genera para mostrar la planificación de un proyecto las dependencias entre las tareas, poniendo a modo de flecha o similar, las relaciones que existen entre ellas. Esto es un requisito vital, ya que sin él, no podremos competir con las técnicas gráficas que utilizan las herramientas de gestión de proyectos.

Por otro lado, también se puede ampliar el tipo de dependencias que existen entre las tareas, ya que en el prototipo actual de GesProSoft únicamente tenemos en cuenta dependencias fin-inicio, de manera que si una tarea Task 1 tiene como dependiente otra tarea Task 2, su dependencia implica que en el instante en el que se finaliza la tarea Task 1, da paso al comienzo de la tarea Task 2. Así pues, se podrían añadir dependencias del tipo que una vez finalizada la tarea Task 1, y tras un retardo de 2 días, se dé el comienzo de la tarea Task 2. Otra opción que se nos puede ocurrir es la de sincronizar las tareas, de manera que no de comienzo la tarea Task 3 hasta que se haya finalizado la ejecución más larga entre las tareas Task 1 y la tarea Task 2.

Otro tipo de funcionalidades que tienen las herramientas de gestión de proyectos y que se deben añadir son calendarios o agendas en las que diariamente vayan apareciendo el trabajo que debe de desarrollar cada uno de los empleados, basándose en la planificación del proyecto. Esto puede facilitar la labor de los desarrolladores, ya que pueden comprobar diariamente si se están ajustando o no al plan previsto, pudiendo poner remedio rápidamente en el caso en el que la planificación no se

esté realizando según lo esperado, buscando alguna solución al problema que esté produciendo el retraso.

Como se puede comprobar, al prototipo desarrollado en la fase de elaboración se le pueden aplicar diversas mejoras, consiguiendo, finalmente, una aplicación web de gran calidad y de completa funcionalidad para realizar una gestión de proyectos.

12.- Análisis de riesgo

A la hora de hacer una lista de riesgos, se deben de tener en cuenta todas las circunstancias desfavorables que pueden darse a lo largo del proyecto, circunstancias que pondrían en peligro una finalización exitosa del mismo. Además, dependiendo del tipo de riesgo que se de, las repercusiones serán muy diferentes en cada caso, por lo que vamos a indicar, cual es la magnitud del daño que dicho riesgo puede realizar. La magnitud la ponemos en una escala del 1 al 10, indicando con un 1 el riesgo que si se produce es el menos dañino, y con un 10 el más dañino.

12.1.- Pérdida de datos (magnitud 10)

Descripción: Debemos tener en cuenta que para el desarrollo del proyecto es indispensable un correcto funcionamiento del equipo, ya que sino podríamos perder toda la información que en el se almacena, y si se nos estropea en la finalización de nuestro proyecto, deberíamos comenzar desde cero otra vez para el desarrollo del mismo.

Impactos: Pérdida de mucho tiempo, ya que todos los datos que hayamos perdido y no podemos recuperar ya sea pérdida de información de los procesos, proyectos o bien sobre la propia aplicación web, nos han costado cierto tiempo para su realización, por lo tanto todo ese tiempo invertido no nos ha valido para nada.

Indicadores: No podemos acceder correctamente a los datos de nuestro proyecto.

Estrategia de mitigación: Crearemos copias de seguridad, mensualmente o siempre que se realice una evolución trascendente en el proyecto, en un CD de todas las versiones que obtengamos tanto de la documentación desarrollada, como del desarrollo del proyecto en sí, para así poder mitigar la pérdida de datos.

Plan de contingencia: Volver a reponer el trabajo anteriormente realizado partiendo de la última copia de seguridad.

12.2.- Problemas en selección del software de construcción y riesgo tecnológico (magnitud 9)

Descripción: Debemos tener en cuenta que para el desarrollo del proyecto es indispensable que el software que se nos recomienda en la fase de elaboración y que debemos utilizar durante la fase de construcción funcione correctamente, ya que de lo contrario, vamos a tener problemas a la hora de realizar nuestro proyecto.

Impactos: Pérdida de tiempo si el problema viene a la hora de instalar el software, además del mal funcionamiento en el caso en el que no nos proporcione las funcionalidades que se supone que nos debiera ofrecer.

Indicadores: No nos funciona bien el software instalado.

Estrategia de mitigación: Tener acceso al software que necesitemos en el caso de que precisemos de él, desde distintos orígenes.

Plan de contingencia: Desinstalar el software instalado en nuestro equipo, y volver a instalar el software de nuevo, asegurándonos de que el fallo anterior queda corregido.

12.3.- Velocidad de carga de los diagramas GANTT (magnitud 9)

Descripción: La visualización de las tareas y las relaciones entre estas en el proyecto GesProSoft, puede que sea más compleja de lo que nos podamos esperar, y que a la hora de ser mostradas en la aplicación, requieran mucho tiempo en sobre todo si los equipos no cumplen con los requisitos necesarios.

Impactos: No poder obtener de una manera intuitiva y totalmente legible los datos de las duraciones y relaciones entre tareas, con lo que tendríamos una forma menos gráfica pero en la que se mostrarían todos los datos que nos sean precisos para el desarrollo de una buena gestión.

Indicadores: Mucho tiempo de espera para mostrar un diagrama GANTT en el cliente.

Estrategia de mitigación: Indicar en el documento de arquitectura del software que será entregado al cliente en el anexo, cuales son los requisitos mínimos recomendados para que la aplicación funcione correctamente.

Plan de contingencia: En el caso en el que los equipos no cumplan con los requisitos mínimos, se le realizará un presupuesto con las mejoras a añadir teniendo en cuenta las características de

los equipos de su empresa en la actualidad.

12.4.- Elevadas curvas de aprendizaje en las herramientas a utilizar (magnitud 7)

Descripción: Puede que durante el periodo de construcción de la aplicación, invirtamos demasiado tiempo en la curva de aprendizaje de las tecnologías que han sido seleccionadas en la fase de elaboración, de manera que se pierda mucho tiempo y haya retrasos para el resto de las actividades que se deben realizar.

Impactos: Pérdida de mucho tiempo, pudiendo llegar al caso extremo en el que no se pueda desarrollar el proyecto en los plazos acordados.

Indicadores: Un retraso en cada una de las fases y su consecuente retraso para alcanzar los hitos de las mismas, sería un indicador de que el riesgo está presente.

Estrategia de mitigación: Establecer un período de aprendizaje dentro del desarrollo del proyecto, que sería breve pero intenso, para conocer las herramientas a utilizar.

Plan de contingencia: En caso de que se detecte que este riesgo está influyendo en el desarrollo del proyecto, se intentará reducir el estudio detallado de las mismas, o en el caso en el que esto no sea posible, se aumentará el tiempo de dedicación al estudio de las herramientas dentro del proyecto.

13.- Organización y gestión del proyecto

En este apartado se va a tratar la organización y la gestión del producto final que se obtendrá a partir del prototipo GesProSoft.

Debido a que para la realización del prototipo hemos seguido el método Rational Unified Process (RUP), lo más indicado para poder continuar de una manera coherente y homogénea con la documentación generada, es que para la construcción del producto final, se siga utilizando dicho método, de manera que se pueda reutilizar toda la documentación que ha sido generada.

Del mismo modo que resulta más homogéneo seguir con el método RUP, también lo sería el realizar la documentación de la memoria del producto final siguiendo la norma de AENOR *UNE 157801*.

Una vez que se comience con la fase de construcción del producto final, los encargados de realizarlo deben de tener que seguir en contacto con el cliente que desea desarrollar la aplicación web, para asegurarse de que la realización del producto final cumple todos y cada uno de los requisitos iniciales. Para ello, en el grupo de trabajo encargado de la fase de construcción, deberá existir una persona cuya labor sea estar en contacto con el cliente, y asegurarse de transmitir al grupo de desarrolladores, todos los datos de interés para que el proyecto finalice con éxito.

La aprobación de cada uno de los documentos y cambios que se apliquen en el proyecto, debe de ser realizada por el encargado del grupo de desarrolladores, que puede coincidir que sea la misma persona encargada de hablar con el cliente, aunque no tiene porque ser la misma obligatoriamente.

Dentro del grupo de trabajo, y para que resulte mucho más fácil y fluida la comunicación entre sus integrantes, se va a tener un sitio web en el que todos y cada uno de ellos va a ir poniendo cada una de las tareas que está realizando. De este modo, en el caso en el que existan módulos diferentes pero dependientes entre sí, y dichos módulos estén siendo implementados por personas distintas, pueden ver en todo momento cualquier documentación que les pueda ser de utilidad. En el sitio web, habrá instalado un sistema de versiones, para que así no se pierda ningún tipo de información que nos pueda aportar la evolución del proyecto en sí.

En el caso en el que el alcance del proyecto se quiera modificar en la fase de construcción, dependiendo de la transcendencia de los cambios, podría suponer desde un simple cambio que no trastoque el desarrollo del proyecto en cuyo caso se estudiaría la posibilidad de incorporarlos, hasta tener que cambiar la arquitectura del sistema, en cuyo caso sería necesaria una labor de reingeniería, que podría derivar en que se abandone el proyecto sin haber llegado a la finalización del mismo.

14.- Planificación temporal

Vamos a realizar una planificación para el desarrollo del producto final, partiendo desde la parte de elaboración ya realizada, con sus prototipos correspondientes. Tal y como define el RUP, las fases que realmente van a quedar por realizar son tanto la de construcción como la de transición, sin embargo, y dado a que los desarrolladores de la fase de construcción no han participado en la elaboración de los prototipos, van a tener que invertir un tiempo menor en las fases de inicio, y elaboración, para poder entrar posteriormente a participar en la fase de construcción, ya que no se puede prescindir de estas dos fases iniciales, porque sería imposible realizar un producto de calidad, sin te-

ner claros tanto los requisitos como el manejo de las herramientas que se van a utilizar en la fase de construcción.

Además, dado que en la construcción de producto final se van a añadir algunos requisitos que hagan de la aplicación web una herramienta más competitiva que el prototipo GesProSoft, tampoco va a haber un escaso tiempo invertido en la fase de inicio, ya que se va a tener que hacer toma de requisitos, lo que nos va a llevar a realizar un completo análisis y diseño del producto final, aunque vamos a poder reutilizar el análisis y diseño realizados para el prototipo GesProSoft.

Por lo tanto, vamos a hacer una estimación de tiempos en la Tabla 1 , teniendo en cuenta todas y cada una de las fases del RUP. Para que los datos sean de un carácter relevante, debemos de suponer cuantas personas van a participar en cada una de las fases, dato necesario para los cálculos.

Vamos a suponer para ello, que solo va a haber una persona encargada en realizar reuniones con el cliente, y por lo tanto, va a ser solo una la persona la que participe en la fase de inicio. Por otro lado, una vez que se hayan realizado todas las tomas de requisitos, y se haya rellenado toda la documentación correspondiente a la fase de inicio, en las fase de elaboración y construcción, vamos a partir de la hipótesis de que van a participar en ellas cuatro personas, para realizar un reparto de tareas tales que dos de ellas se encarguen de ampliar las funcionalidades necesarias para el proyecto, otra será la encargada de realizar todas las mejoras posibles en la parte de la interfaz gráfica y el consultor que seguirá manteniendo las reuniones con el cliente. Por otro lado, en la fase de transición, vamos a contar con dos trabajadores, uno puede ser uno de los que estuvo en la parte del desarrollo, y otro, el que ha mantenido las comunicaciones con el cliente. Contamos por lo tanto con un total de trabajadores de 4, de los cuales tres son programadores, y uno de ellos es ingeniero informático. Partiendo de esta hipótesis, tenemos una planificación tal que:

<i>Iteración</i>	<i>Personas</i>	<i>Duración</i>	<i>Formación</i>	<i>Herramientas</i>	<i>Fases</i>
1	1	~ 3 meses	RUP, UML	MS FrontPage, Rational Rose, Word	Inicio
2	4	~ 2 mes	Struts, RUP, UML, J2EE	Struts, Rational Rose, Eclipse, SQL Server 2000	Elaboración
3	4	~ 12 meses	Struts, Enterprise Java Beans, JSP, Servlets	Struts, Eclipse, JBoss	Construcción
4	2	~ 1 meses		MS FrontPage, Word, OpenOffice	Transición

Tabla 1: Planificación realizada para el producto final

Por lo tanto, y si se cumple la planificación, podríamos contar con un producto final de carácter comercial en un periodo de tiempo de un año y medio.

15.- Resumen del presupuesto

Teniendo en cuenta cada una de las hipótesis que se han realizado en el apartado anterior, vamos a realizar un presupuesto, teniendo en cuenta no solo el coste del personal que va a formar parte de la empresa encargada de desarrollar el producto final, sino también el coste de todas las licencias de software necesarias, tanto en nuestra empresa como en caso de que el cliente no las tuviese ya adquiridas.

En la siguiente tabla enumeran las diferentes licencias que nuestra empresa necesita disponer para la construcción del producto. Por supuesto, no son todas repercutibles al cliente, así que hemos hecho una estimación para ver cuánto cargar al cliente⁸:

⁸ Hemos tomado como baremo que las licencias de todos los productos nos duran dos años, y que vamos a realizar diez proyectos al año. Esto nos indica que debemos cobrarle al cliente la vigésima parte del total de la tabla.

<i>Software</i>	<i>Proveedor</i>	<i>Equipos</i>	<i>Precio/equipo⁹</i>	<i>Precio Total⁹</i>
Windows XP Professional	Microsoft	4	299,00 €	1.196,00 €
Office XP Professional	Microsoft	4	579,00 €	2.316,00 €
Microsoft SQL Server 2000	Microsoft	1	20.000,00 €	20.000,00 €
Eclipse	Eclipse Foundation	3	0,00 €	0,00 €
MyEclipse	Genuitec LLC	3	29,95 €	89,85 €
JBoss	JBoss Inc.	3	0,00 €	0,00 €
Rational Unified Process	Rational	1	871,00 €	871,00 €
Rational Rose	Rational	1	5.220,00 €	5.220,00 €
OpenOffice	OpenOffice.org	4	0,00 €	0,00 €

Total	29.692,85 €
Total repercutible al cliente	1.484,64 € ¹⁰

Table 2: Costes asociados a nuestra empresa.

En la siguiente tabla vienen desglosados los costes referentes al personal, directamente imputables al cliente.

<i>Tiempo de trabajo</i>	<i>€/hora</i>	<i>Nº Personas</i>	<i>Total/persona</i>	<i>Total/ personas</i>
18 meses	120 €/hora	1	345.600,00 €	345.600,00 €
18 meses	60 €/hora	3	172.800,00 €	518.400,00 €

Total	865.484,64 €
--------------	--------------

⁹ Duración de la licencia: dos años.

¹⁰ Suma total / 20 (vigésima parte del total)

16.- Anexos justificativos de la solución adoptada

16.1.- Análisis y diseño del sistema

A la hora de realizar el análisis y diseño para el prototipo GesProSoft que hemos desarrollado, hemos utilizado como herramienta el Rational Rose, una herramienta de gran calidad y de un gran reconocimiento en el mercado. Con dicha herramienta hemos desarrollado tanto los casos de uso, como los diagramas de secuencia, colaboración y realización.

En este apartado, vamos a poder ver los diagramas que hemos generado para cada uno de los casos de uso definidos en el sistema.

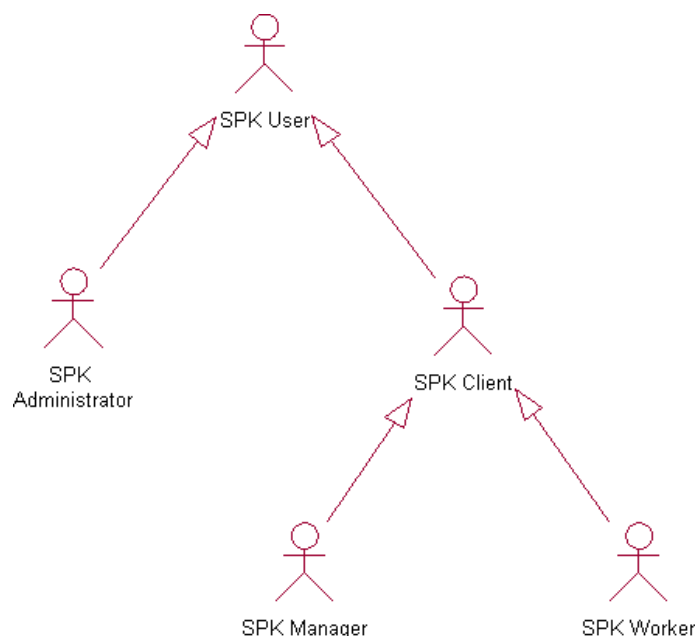


Figura 8: Actores definidos para la aplicación final

En primer lugar, vamos a ver los actores definidos para la aplicación web. Como podemos ver en la Figura 8, tenemos definidos cinco actores diferentes. Sin embargo, en el ámbito del proyecto GesProSoft, únicamente vamos a hacer referencia al actor SPK Manager, ya que es el encargado de realizar las gestiones de los proyectos.

Los casos de uso que hemos desarrollado para la aplicación web GesProSoft, son aquellos que se encuentran en la Figura 9, exceptuando el caso de uso Manage Own Project Plan que pertenece a la aplicación preexistente del proyecto y que está incluido ya que tiene una relación directa con el caso de uso View Own Project Plan, que sí que forma parte del modelo de casos de uso del proyecto GesProSoft.

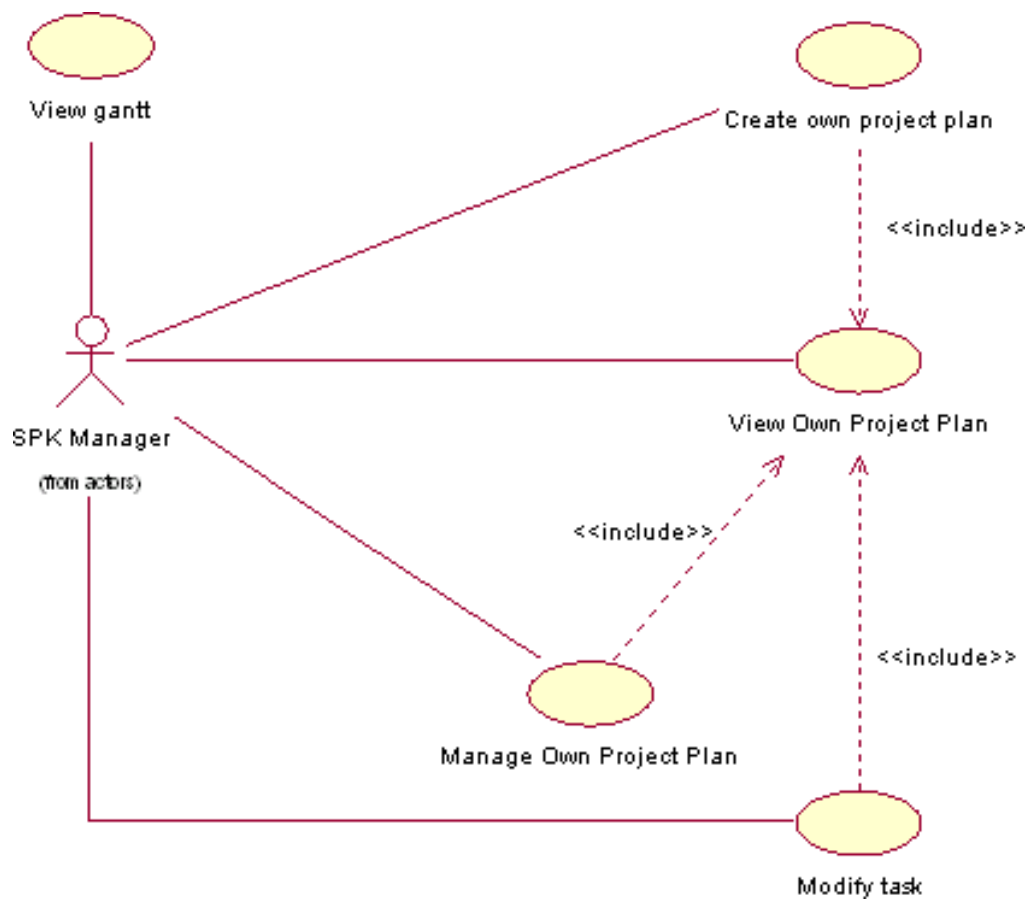


Figura 9: Modelo de casos de uso para el proyecto GesProSoft

Vamos a ver cada uno de los casos de uso que han sido definidos para el proyecto GesProSoft, para que de este modo se llegue a una mejor comprensión de los mismo:

Caso de uso View Own Project Plan :



Figura 10: Caso de uso View Own Project Plan

Este caso de uso se desarrolló para el solventar la situación en la que el usuario quiere visualizar una planificación para un proyecto específico. Este caso de uso únicamente visualiza un plan en el caso en el que este exista y esté definido dentro de la base de datos. Para el caso en el que la planificación de dicho proyecto no esté creada, deberíamos fijarnos en el caso de uso Create own project plan.

Caso de uso Create own project plan:

Este caso de uso se produce cuando el usuario quiere ver una planificación para un proyecto específico, y sin embargo la planificación aún no ha sido creada. Lo que vamos a realizar es una nueva planificación.



Figura 11: Caso de uso Create own project plan

Posteriormente, una vez que la planificación está creada, vamos a mostrar al usuario los datos sobre las tareas que la componen mediante el caso de uso mencionado anteriormente View Own Project Plan.

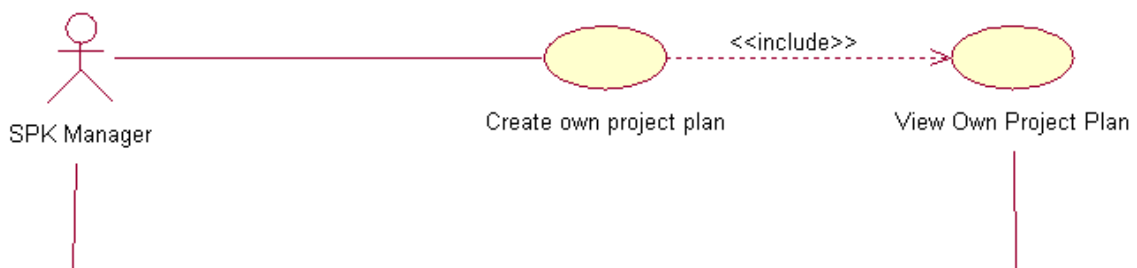


Figura 12: Caso de uso Create own project plan

Caso de uso Modify task:

Gracias a este caso de uso, el usuario es capaz de poder modificar cualquiera de las tareas que él seleccione. Los campos de las tareas que puede modificar van a ser tanto la fecha de inicio de la tarea, como la fecha de su finalización. Al realizar estos cambios, en la ejecución tendremos en cuenta las dependencias existentes entre las tareas.

Una vez que las tareas han sido modificadas y los cambios han sido guardado en la base de datos, vamos a mostrar de nuevo al usuario la planificación del proyecto con los cambios correspondientes en él.

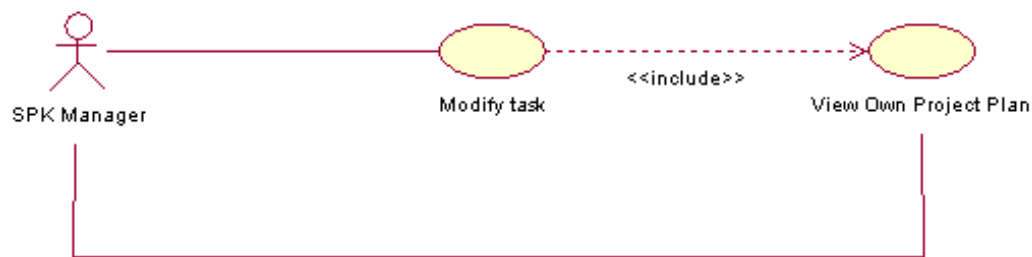


Figura 14: Caso de uso Modify task

Caso de uso View gantt:

Este caso de uso no muestra de una manera más gráfica que el caso de uso View Own Project Plan la planificación para un proyecto específico. Además, incluye no solo la planificación para dicho proyecto, sino que también nos muestra cómo va la ejecución de las tareas en realidad, lo cual nos es de mucha utilidad para ver si se están cumpliendo los plazos o no. Los datos se muestran mediante un diagrama gantt, de manera que en un golpe rápido de vista, vamos a poder tener una visión de la planificación y del estado del proyecto.



Figura 15: Caso de uso View gantt

Hemos visto todos los casos de uso definidos para el desarrollo de la aplicación web GesProSoft. A continuación vamos a ver los diagramas que hemos desarrollado para cada uno de los casos de uso que hemos visto anteriormente. Vamos a tener tres tipos de diagramas para cada caso de uso, diagrama de análisis, diagrama de colaboración y diagrama de secuencia a nivel de diseño.

En los dos primeros, vamos a poder ver a grandes rasgos, las peticiones y respuestas que se realizan desde y hacia el usuario respectivamente. Sin embargo, en el diagrama de secuencia a nivel de diseño, vamos a poder ver con mayor detalle y paso a paso, cada una de las peticiones y acciones que se deben realizar, hasta implementar cada uno de los casos de uso definidos. Dicho diagrama nos aporta toda la información necesaria acerca del código que se ejecuta para realizar este caso de uso, aunque a un nivel superior al nivel de implementación.

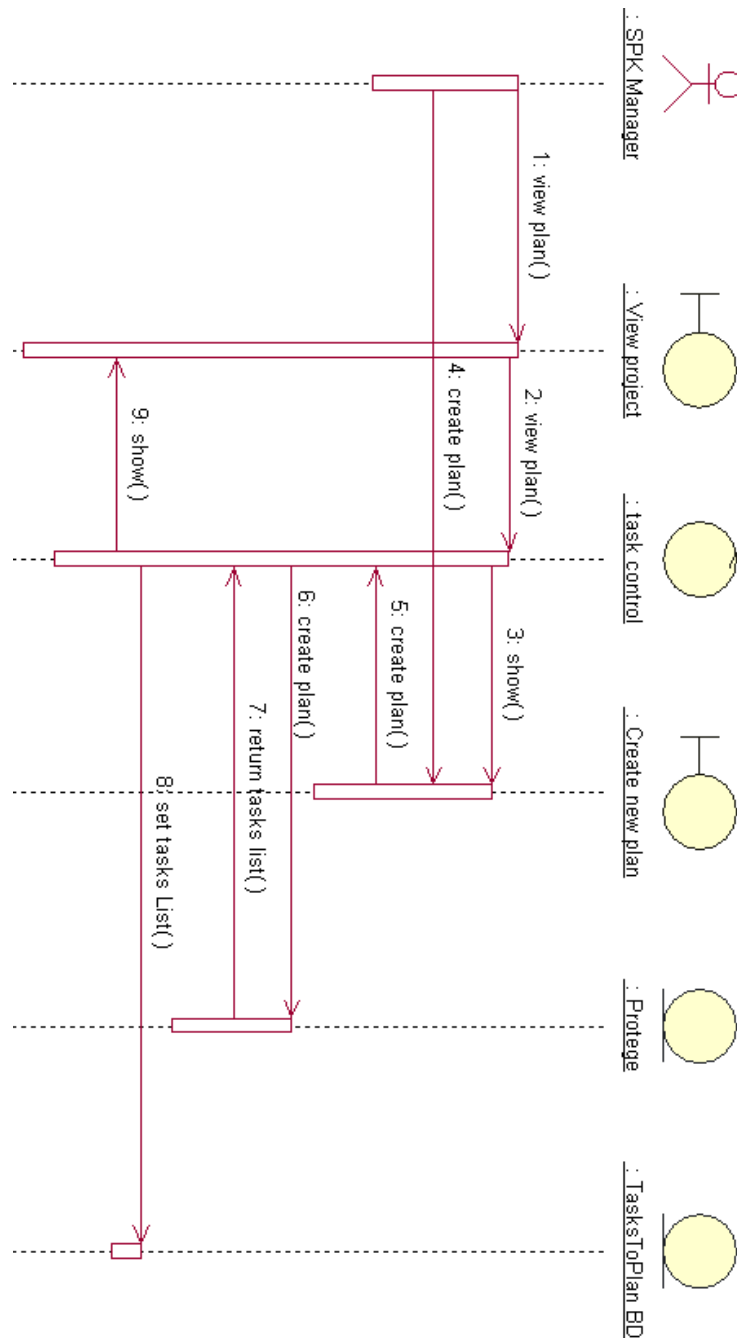


Figura 16: Diagrama de análisis para caso de uso create own project plan

En la Figura 16 podemos ver el diagrama de análisis para el caso de uso Create own project plan.

En la Figura 17 podemos ver el diagrama de colaboración para el caso de uso Create own project plan.

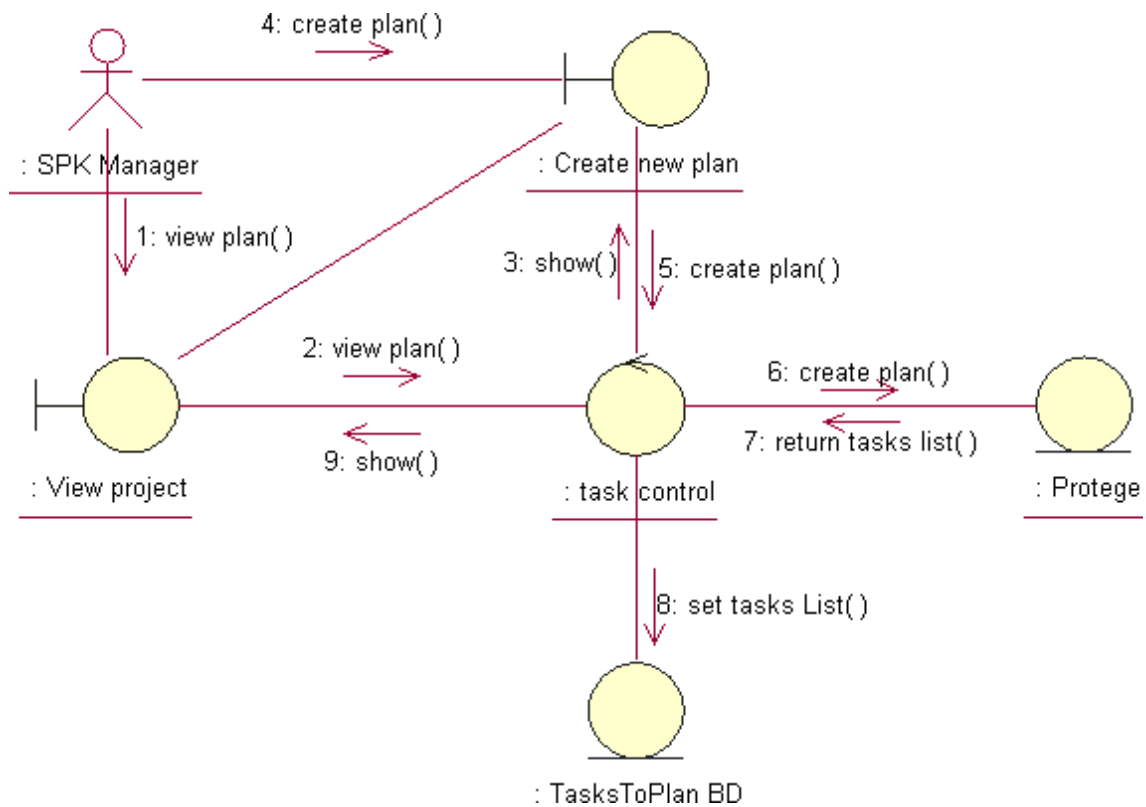


Figura 17: Diagrama de colaboración para el caso de uso create own project plan

La Figura 18 nos muestra el diagrama de secuencia a nivel de diseño para el caso de uso create own project plan, en el que podemos ver paso a paso cada una de las acciones a realizar hasta conseguir la implementación de dicho caso de uso.

Figura 18: Diagrama de secuencia a nivel de diseño para el create own plan

A continuación en la Figura 19 podemos ver para el caso de uso View Own Project Plan el diagrama de análisis:

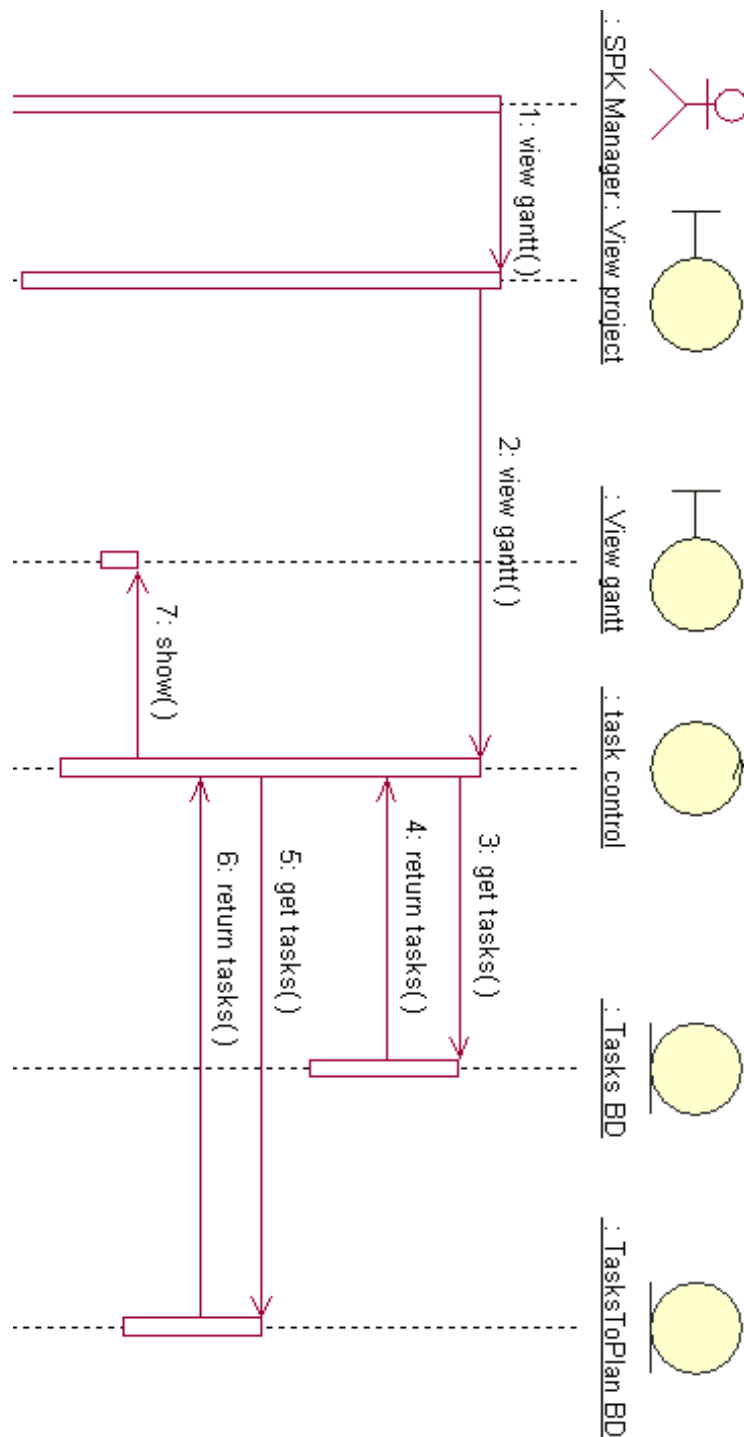


Figura 19: Diagrama de análisis para el caso de uso View Own Project Plan

En la Figura 20 tenemos el diagrama de colaboración para el caso de uso View Own Project Plan:

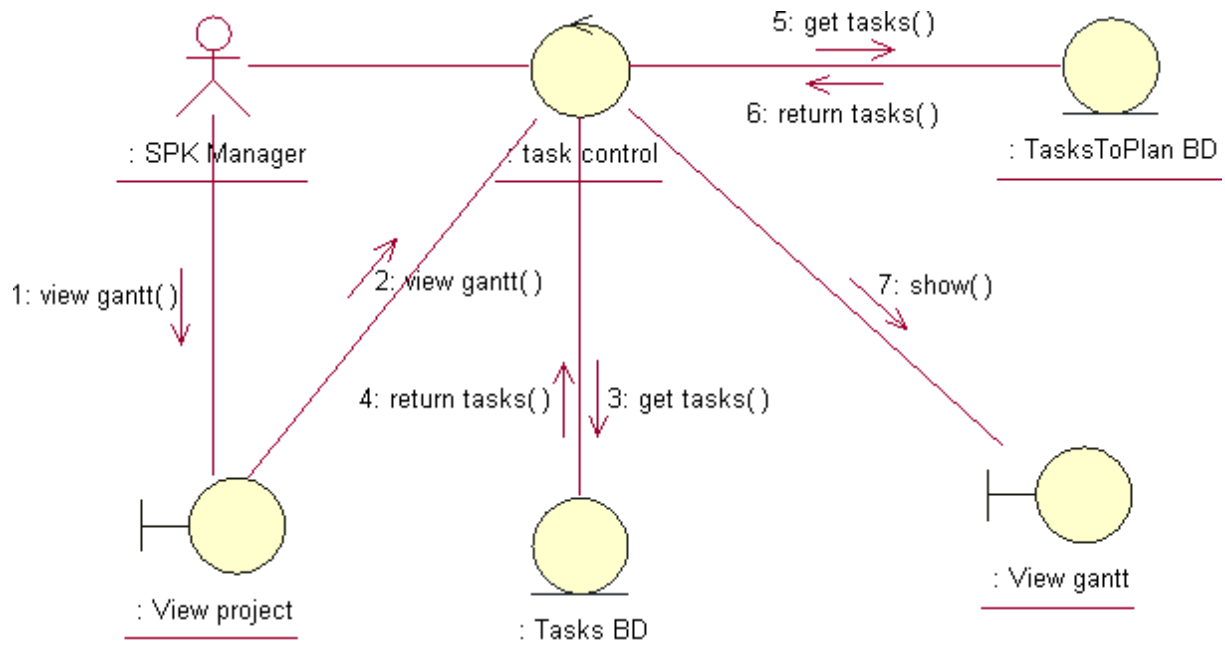


Figura 20: Diagrama de colaboración para el caso de uso View Own Project Plan

En la Figura 21 podemos ver el diagrama de secuencia a nivel de diseño para el caso de uso View Own Project Plan, en el que mostramos todos los pasos a dar para la implementación del mismo.

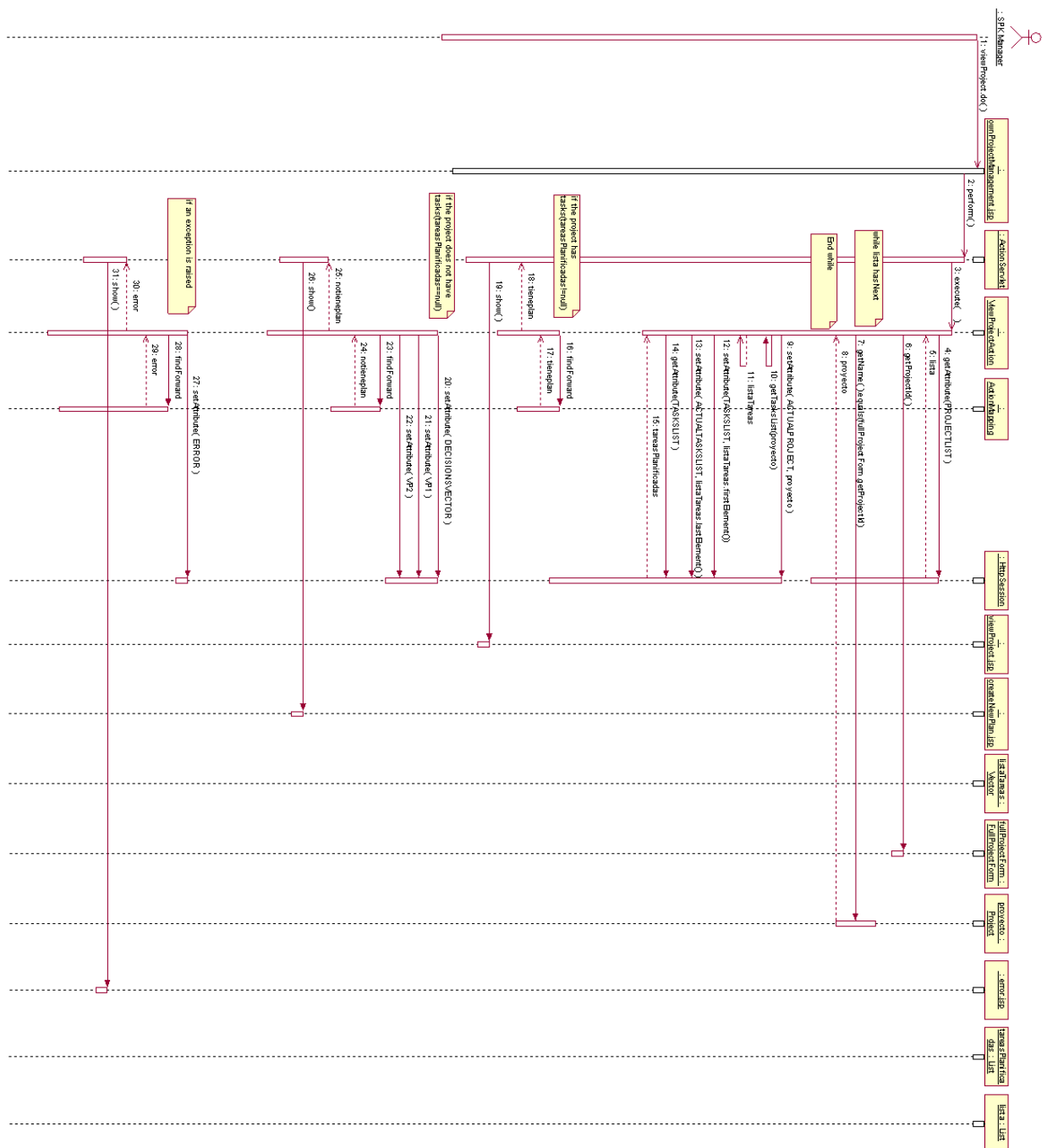


Figura 21: Diagrama de secuencia a nivel de diseño del caso de uso View Own Project Plan

En la Figura 22 podemos ver el diagrama de análisis para el caso de uso Modify task:

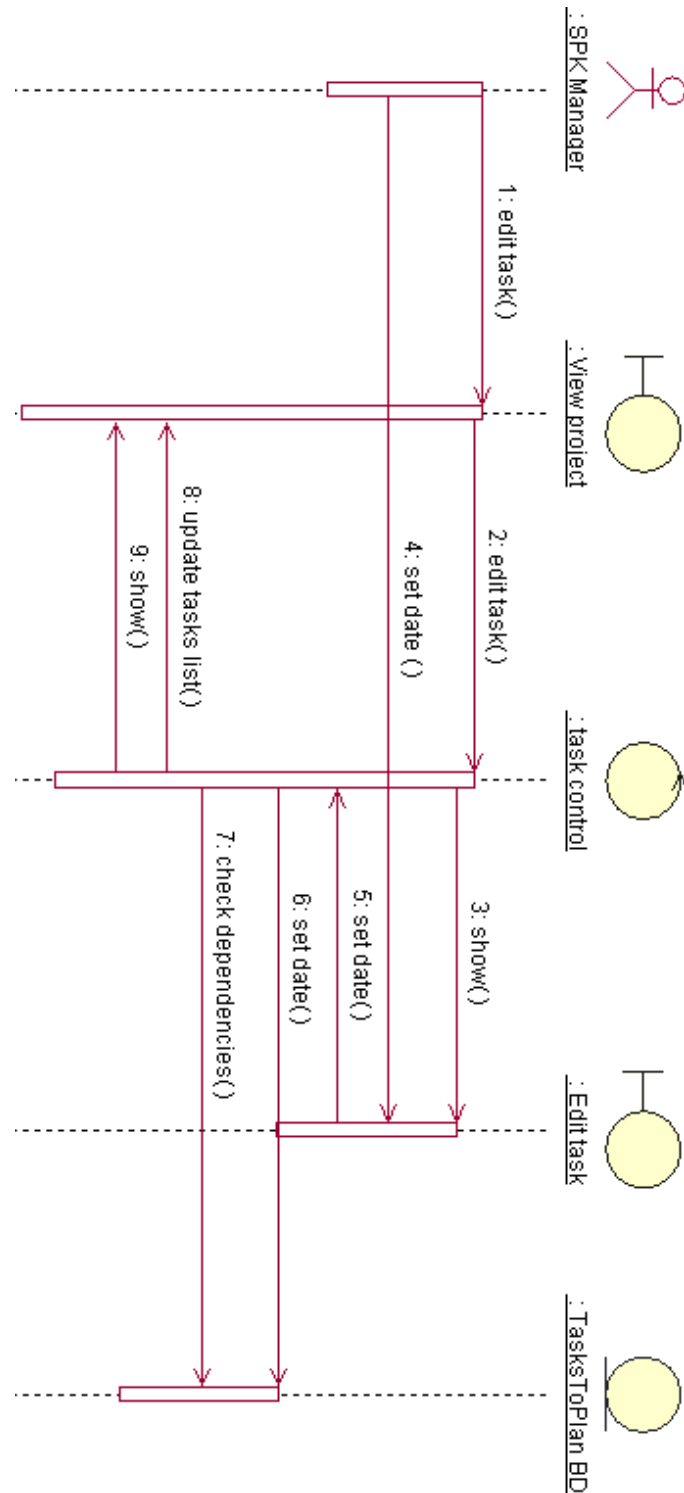


Figura 22: Diagrama de análisis para el caso de uso Modify task

También tenemos el diagrama de colaboración para el caso de uso Modify task en la Figura 23 :

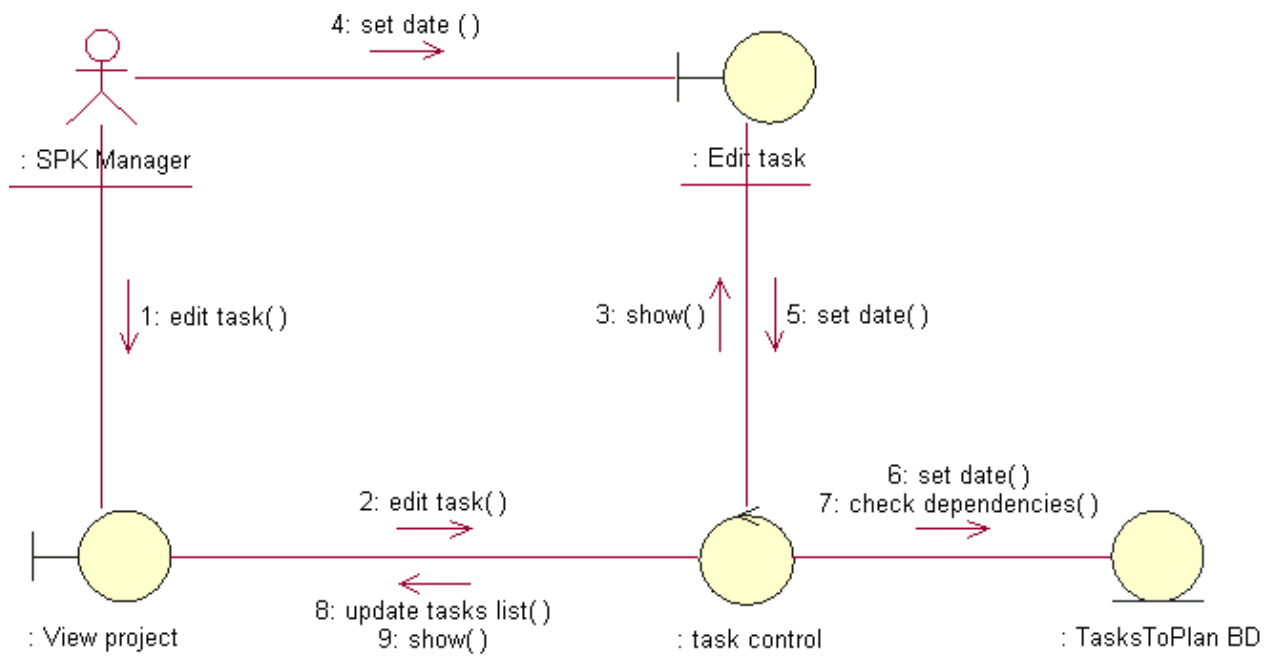


Figura 23: Diagrama de colaboración para el caso de uso Modify task

En la Figura 24 mostramos el diagrama de secuencia a nivel de diseño para el caso de uso Modify task, en el que vemos cada uno de los pasos a dar para la implementación de este caso de uso, teniendo en cuenta cada una de las dependencias posibles que se puedan dar en cada caso.

53

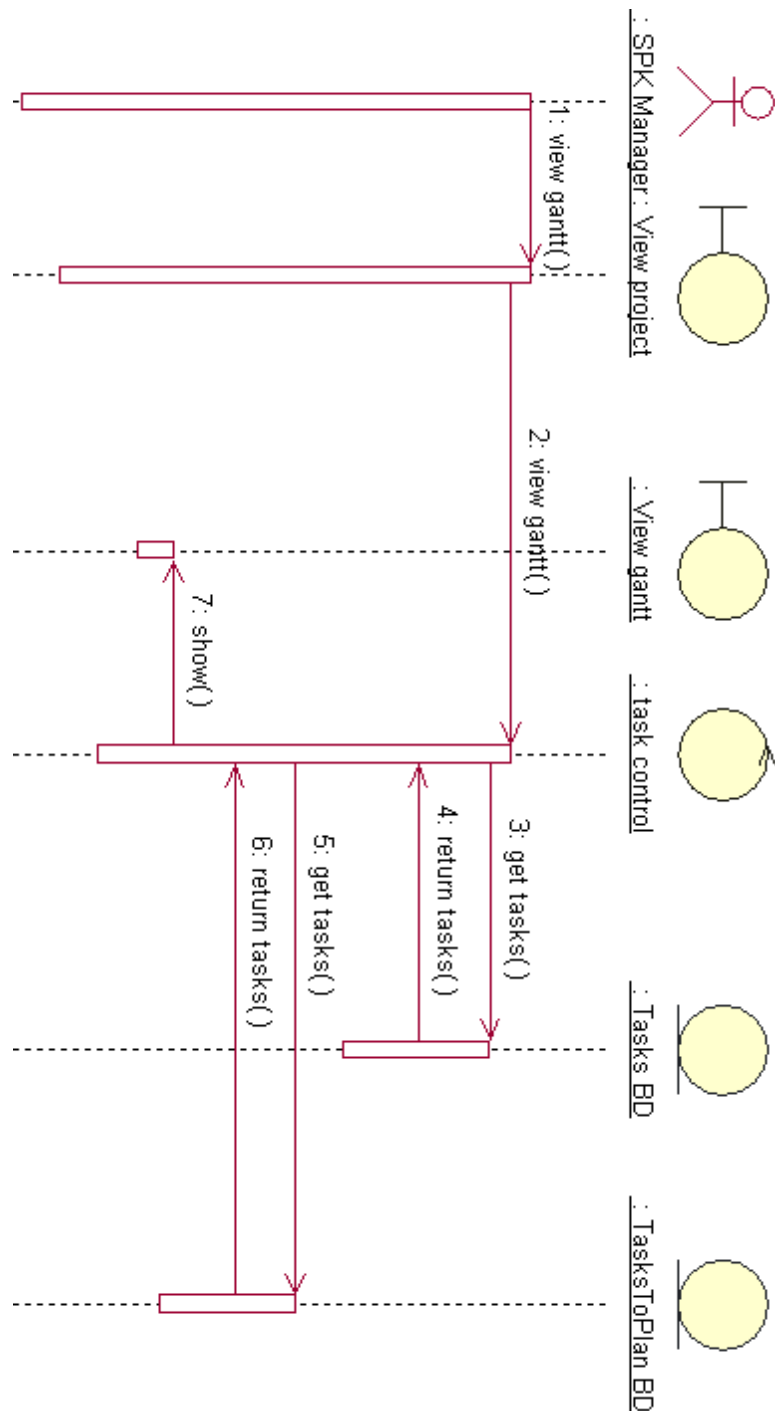


Figura 25: Diagrama de análisis para el caso de uso View gantt

Por último, tenemos el diagrama de análisis para el caso de uso View gantt en la Figura 25 y en la Figura 26 podemos ver el diagrama de colaboración para el caso de uso View gantt:

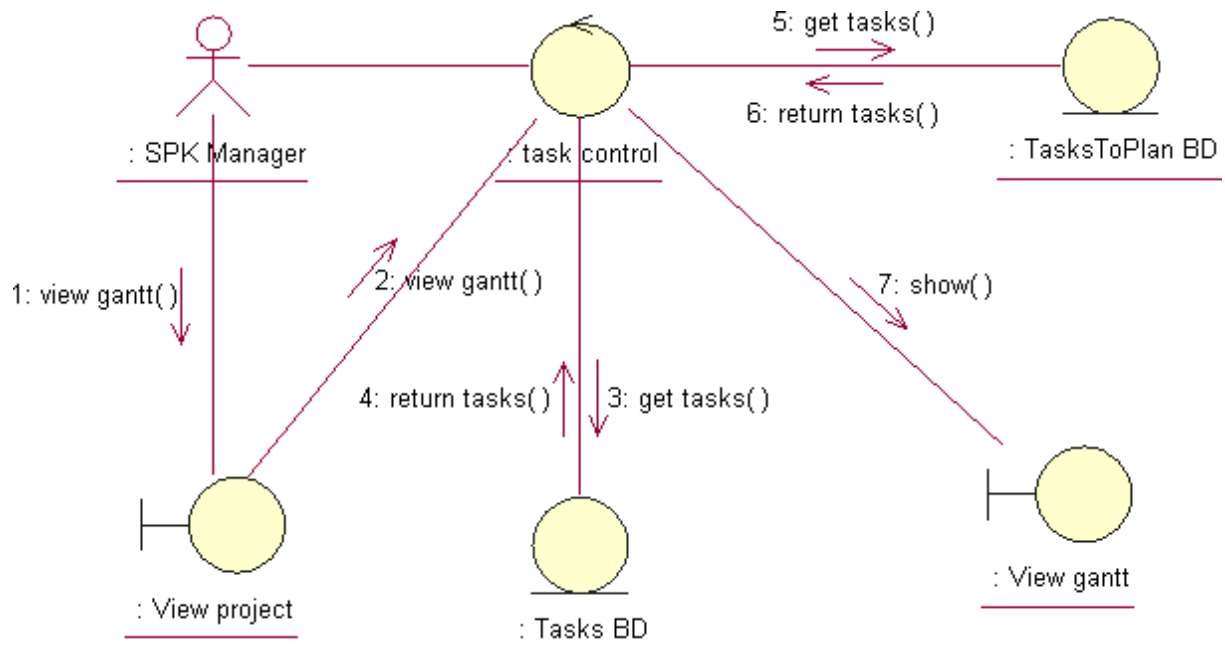


Figura 26: Diagrama de colaboración para el caso de uso View gantt

La Figura 27 nos muestra el diagrama de secuencia a nivel de diseño para el caso de uso View gantt, el cual podemos comprobar que gracias a la utilización del jfreechart no es muy complejo, lo cual no hubiera sucedido así en el caso en el que se hubiera tenido que realizar manualmente toda la visualización de las tareas para cada una de las planificaciones.

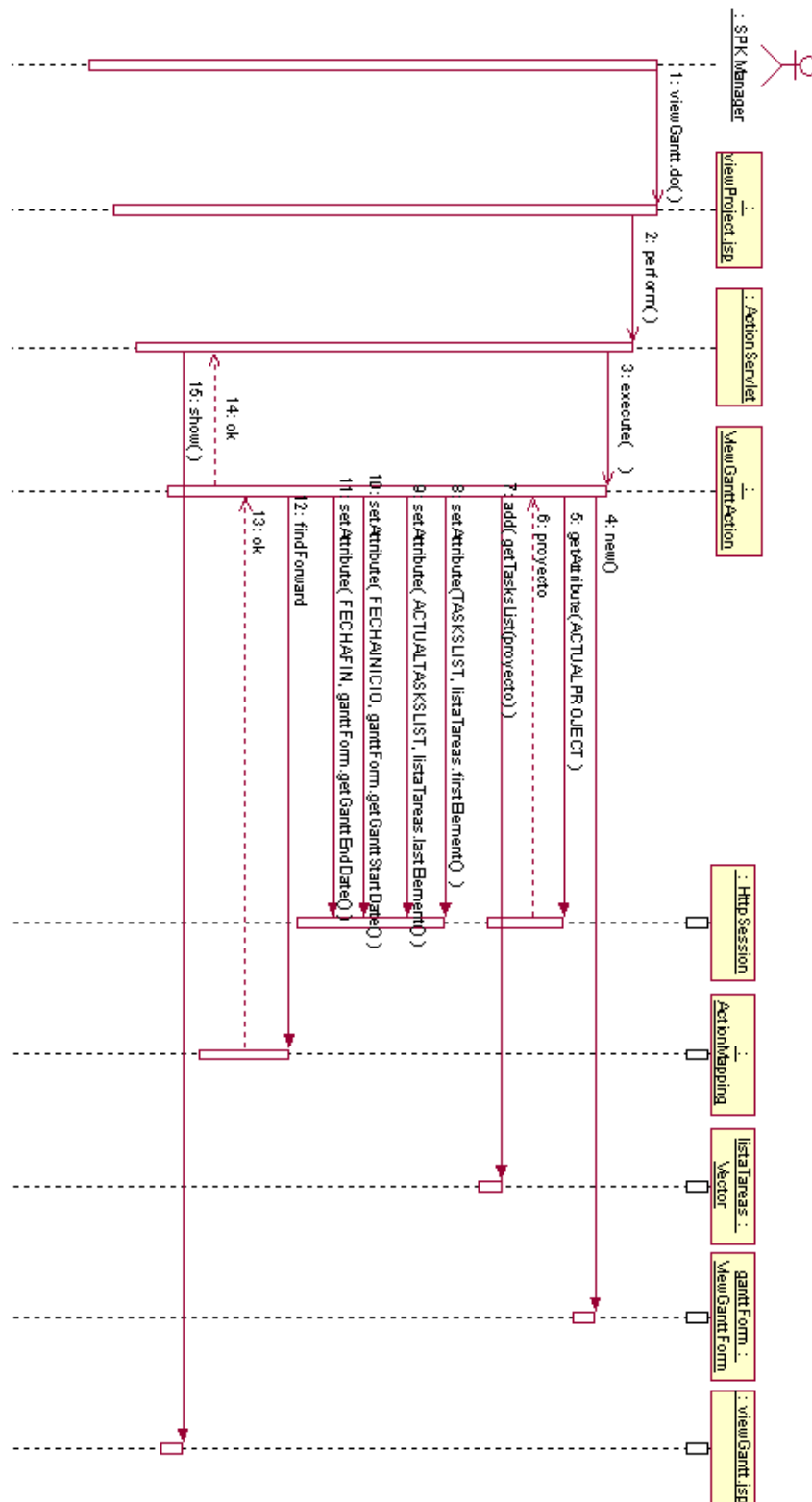


Figura 27: Diagrama de secuencia a nivel de diseño del caso de uso View gantt

16.2.- Estimación de tamaño y esfuerzo

En este apartado vamos a hablar sobre los datos correspondientes al producto final. Tal y como hemos podido ver en el apartado de planificación temporal para el producto final, teniendo en cuenta la hipótesis de que contamos en la empresa de desarrollo del producto con 4 trabajadores, de los cuales uno es ingeniero informático y los otros tres son programadores, tardaríamos un año y medio en poder obtener un producto final que se pueda comercializar, con todas las características necesarias para tener un gran éxito.

El número de horas que invertirá cada trabajador, suponiendo que trabaja 8 horas al día, será aproximadamente de 2.880 horas. Estas horas son pagadas a 120 €/hora al ingeniero informático, dando un coste a lo largo de año y medio de proyecto de 345.600 € . Los coste de pagar a 60 €/hora a los tres programadores a lo largo de año y medio de proyecto son de 518.400 €. Por lo tanto tendremos un gasto total para la realización de la herramienta en sueldo a trabajadores de 864.000 €.

A este coste, debemos de sumarle las diferentes licencias que nuestra empresa necesita disponer para la elaboración del producto. Por supuesto, no son todas repercutibles al cliente, así que hemos hecho una estimación para ver cuánto cargar al cliente¹¹, cuya estimación es 1484,64€.

Por lo tanto, teniendo en cuenta tanto los gastos de licencias como los sueldos a los trabajadores, el cliente va a tener que afrontar un gasto de 865.484,64 €.

16.3.- Plan de seguridad

Dado que la aplicación GesProSoft es una aplicación web, debemos tener en cuenta que pueden acceder muchos usuarios, y dado que la aplicación muestra los datos de todos los proyectos que forman parte de la empresa, vamos a tener que asegurarnos de que nadie que no esté debidamente autenticado pueda acceder a los datos. Para solucionar este posible problema, vamos a proteger el servidor de manera que no deje que usuarios no reconocidos puedan ejecutar la aplicación web.

Dado que en las aplicaciones pueden usarse datos de carácter personal, vamos a asegurar a todos los usuarios de la aplicación el cumplimiento de la Ley Orgánica de Protección de Datos (LOPD) .

Como los datos recogidos en el servidor son de vital importancia, se van a realizar copias de seguridad diariamente. Para que no surjan problemas, se escogerá un horario en el que nos podamos

¹¹ Hemos tomado como baremo que las licencias de todos los productos nos duran dos años, y que vamos a realizar diez proyectos al año. Esto nos indica que debemos cobrarle al cliente la vigésima parte del coste total de licencias.

asegurar de que ningún usuario de la aplicación va a trabajar con ella. Dichas copias serán guardadas y almacenadas durante un periodo de un mes. Pasado este tiempo, se realizará una copia de la última versión recogida, y ésta será guardada de manera permanente en algún dispositivo de almacenamiento.

17.- Especificaciones del sistema

En este apartado vamos a comentar todos y cada uno de los pasos que hemos seguido a lo largo del proyecto, y que han llevado a la realización del prototipo GesProSoft.

A lo largo del ciclo de vida del proyecto GesProSoft, debido a que hemos seguido el proceso del RUP para servirnos de guía a la hora de desarrollar el proyecto, además de garantizarnos la creación de un software de calidad, hemos generado diversos documentos que vamos a ver en los anexos del proyecto para darnos una visión general del ciclo de vida del proyecto. Únicamente vamos a añadir las últimas versiones de los documentos que hemos ido rellenando a lo largo de todo el proyecto según nos lo iba marcando el método del RUP.

Vamos a ver un breve resumen de la aplicación web GesProSoft. Vamos a ver cada una de las versiones que hemos ido realizando, hasta finalizar la fase de elaboración mostrando un prototipo para el proyecto GesProSoft.

Para la primera versión supusimos que la planificación del proyecto a mostrar ya estaba creada, y por lo tanto todos los datos necesarios estaban almacenados en la base de datos. De esta manera el único objetivo de la aplicación era mostrar esta planificación mediante una tabla tal y como podemos observar en la Figura 28. Partimos desde este punto, ya que el caso de uso de crear un nuevo plan era mucho más complejo, con lo cual, empezábamos por una tarea más sencilla que permitía ganar habilidad en el uso de las herramientas para abordar mejor formado el caso de uso de crear plan.

En dicha tabla, permitimos editar las tareas, y modificar las fechas de las mismas en una nueva página como vemos en la Figura 29, teniendo en cuenta las dependencias que podía haber entre ellas. Como podemos ver, permitimos modificar por separado tanto la fecha de inicio como la fecha de finalización de una tarea. Los cambios quedarán registrados en la base de datos una vez que hayamos pulsado el botón de *Save Changes*. En el caso contrario, si no queremos finalmente hacer ninguna modificación, debemos de pulsar el botón de *Exit*.

View plan for project					
1111111A-sequential					
#:	name:	dependencies:	startDate:	endDate:	Edit:
1	Sequential Process Start	Task 1	2004.11.29/ 12:16:24	2004.11.29/ 12:16:24	<input type="button" value="Edit"/>
2	Task 1	Task 2	2004.11.29/ 12:16:24	2004.12.01/ 12:16:24	<input type="button" value="Edit"/>
3	Task 2	Sequential Process End	2004.12.01/ 12:16:24	2004.12.04/ 12:16:24	<input type="button" value="Edit"/>
4	Sequential Process End		2004.12.04/ 12:16:24	2004.12.04/ 12:16:24	<input type="button" value="Edit"/>

Figura 28: Tabla que nos muestra la planificación para un proceso secuencial

Information about Sequential Process Start task	
owning to project 1111111A-sequential	
Task start date:	<input type="text" value="2004.11.29/12:16:24"/>
Task end date:	<input type="text" value="2004.11.29/12:16:24"/>

Figura 29: Página en la que se nos permite modificar las fechas para las tareas

En una segunda versión, se partía de la idea de que dado un proyecto, no se había generado ninguna planificación inicial para el mismo, con lo cual, debía de crearse una planificación basándose en un proceso definido.

Utilizamos Protège para editar tres procesos distintos con los que probar la funcionalidad de crear planes¹²: el proceso secuencial, el proceso decisional y el proceso que incluye subprocessos. El secuencial, nos muestra un diagrama en el que las tareas se procesan secuencialmente, es decir, la “Task 2” comienza tras la finalización de la “Task 1”, creándose entre ellas una dependencia directa, tal y como podemos ver en la Figura 30; el decisional, nos muestra un diagrama en el que en una de las tareas el usuario debe tomar la decisión de continuar por la primera rama “Way 1” o por la segunda “Way 2” del diagrama, tal y como podemos ver en la Figura 32; y el subprocesso, que nos sirve para poder agregar dentro de un diagrama de un proceso cualquier proceso que ya tengamos definido tal como nos muestra la Figura 31.

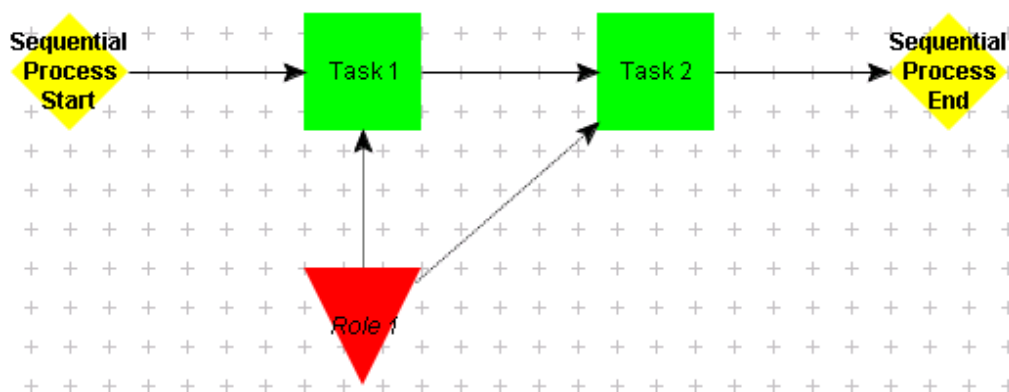


Figura 30: Tipo de proceso secuencial definido en el Protège

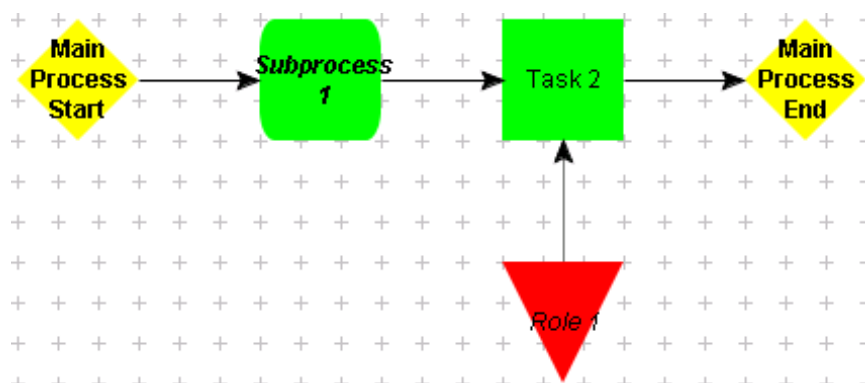


Figura 31: Tipo de proceso subprocesso definido en el Protège

12 Representación de cada uno de los elementos que forman los diagramas:

- Rombo amarillo: Hitos de inicio-fin.
- Cuadrado verde: Tarea simple
- Triángulo rojo: Rol encargado de realizar las tareas a las que apunta.
- Cuadrado verde con bordes redondeados: Subproceso dentro del cual se encontrará definido otro proceso.
- Rombo azul: Tarea decisional, en la que debe decidir por que camino seguir.

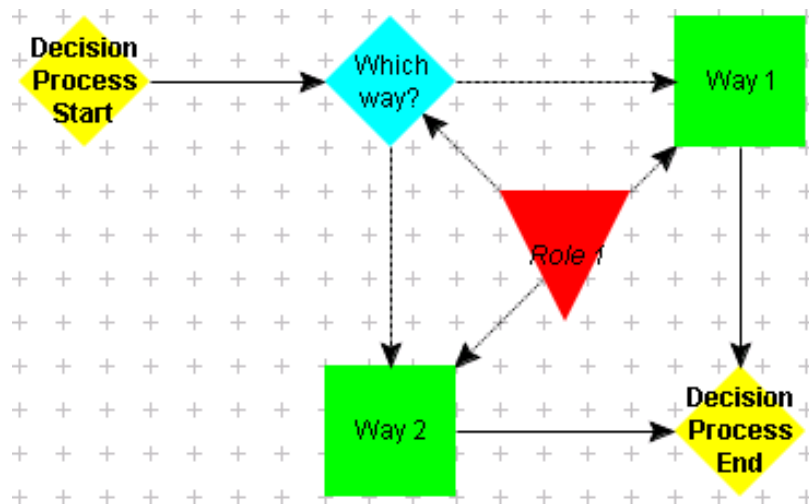


Figura 32: Tipo de proceso decisional definido en el Protégè

Una vez comprobado que la aplicación funciona bien para estos tres procesos, decidimos convenientemente probar la funcionalidad con un proceso de mayor tamaño. Para ello, en el RUP, encontramos definido una planificación del propio RUP, la cual contaba con todas las fases, las tareas a realizar en cada una de ellas, con lo cual, creamos un proceso, que resolviera únicamente la fase de inicio de la planificación del RUP, ya que estaba fase ya cuenta con 43 tareas definidas.

Vamos a ver paso a paso, como se encuentra definido el nuevo proceso llamado proyecto, creado para dar solución a la planificación de la fase de inicio del RUP, la cual podemos ver definida en un formato de texto en la Figura 33:

Para representar este plan utilizamos Protégè, que es una herramienta para la gestión de los procesos que contiene un sistema de adquisición del conocimiento.

Creamos diez instancias distintas, en las cuales vamos a poder añadir en forma de diagrama, todas las tareas y las relaciones que existan entre las mismas. Si nos fijamos, vamos a ir añadiendo en el orden que se describe en la Figura 33, cada una de las tareas, y las relaciones entre las mismas.

1	PROYECTO
2	Fase de inicio
3	Iteración 1 de inicio
4	Gestión de proyecto
5	Concepción de nuevo proyecto
6	Evaluación del alcance y riesgos del proyecto
7	Desarrollo del Plan de Desarrollo del Software
8	Planificar el resto de la iteración inicial
9	Gestión de la iteración
10	Monitorización y control del proyecto
11	Reevaluación del alcance y riesgos del proyecto
12	Plan para la siguiente iteración
13	Refinamiento del Plan de Desarrollo del Software
14	Modelado del negocio
15	Valoración del estado del negocio
16	Identificación de los procesos del negocio
17	Refinamiento de los procesos del negocio
18	Diseño de las realizaciones de procesos de negocio
19	Refine Roles and Responsibilities
20	Explorar la automatización del proceso
21	Requisitos
22	Analizar el problema
23	Comprensión de las necesidades de los afectados
24	Definición del sistema
25	Gestionar el alcance del sistema
26	Refinar la definición del sistema
27	Gestionar requisitos cambiantes
28	Análisis y diseño
29	Hacer la síntesis de la arquitectura
30	Pruebas
31	Planificar pruebas
32	Entorno
33	Preparar el entorno para el proyecto
34	Preparar el entorno para una iteración
35	Preparar guías para una iteración
36	Soporte de entorno para una iteración
37	Gestión de configuración y cambios
38	Planificar el control de configuración y cambio del proyecto
39	Crear entornos de gestión de configuración del proyecto
40	Cambiar y entregar items de configuración Change and Deliver Configuration Items
41	Gestionar artefactos base y versiones
42	Gestionar peticiones de cambio
43	Monitorización e informes del estado de configuración

Figura 33: Plan del RUP

Las relaciones existentes entre las tareas que pertenecen a dicha planificación, se pueden ver en la Figura 34. En nuestro prototipo, hay algunas relaciones de las que se encuentran en la Figura 34 definidas, que no se pueden representar. Es el caso de las dependencias existentes entre tareas que pertenecen a distintos subproceso, o bien dependencias temporales que no sean cuando finalice la tarea Task1 debe comenzar la tarea Task2, ya que solo hemos tenido en cuenta las dependencias Fin-Inicio. Del mismo modo, no se contemplamos las dependencias que requieren de una sincronización entre las tareas . Estos casos no han sido tratados en nuestro prototipo GesProSoft.

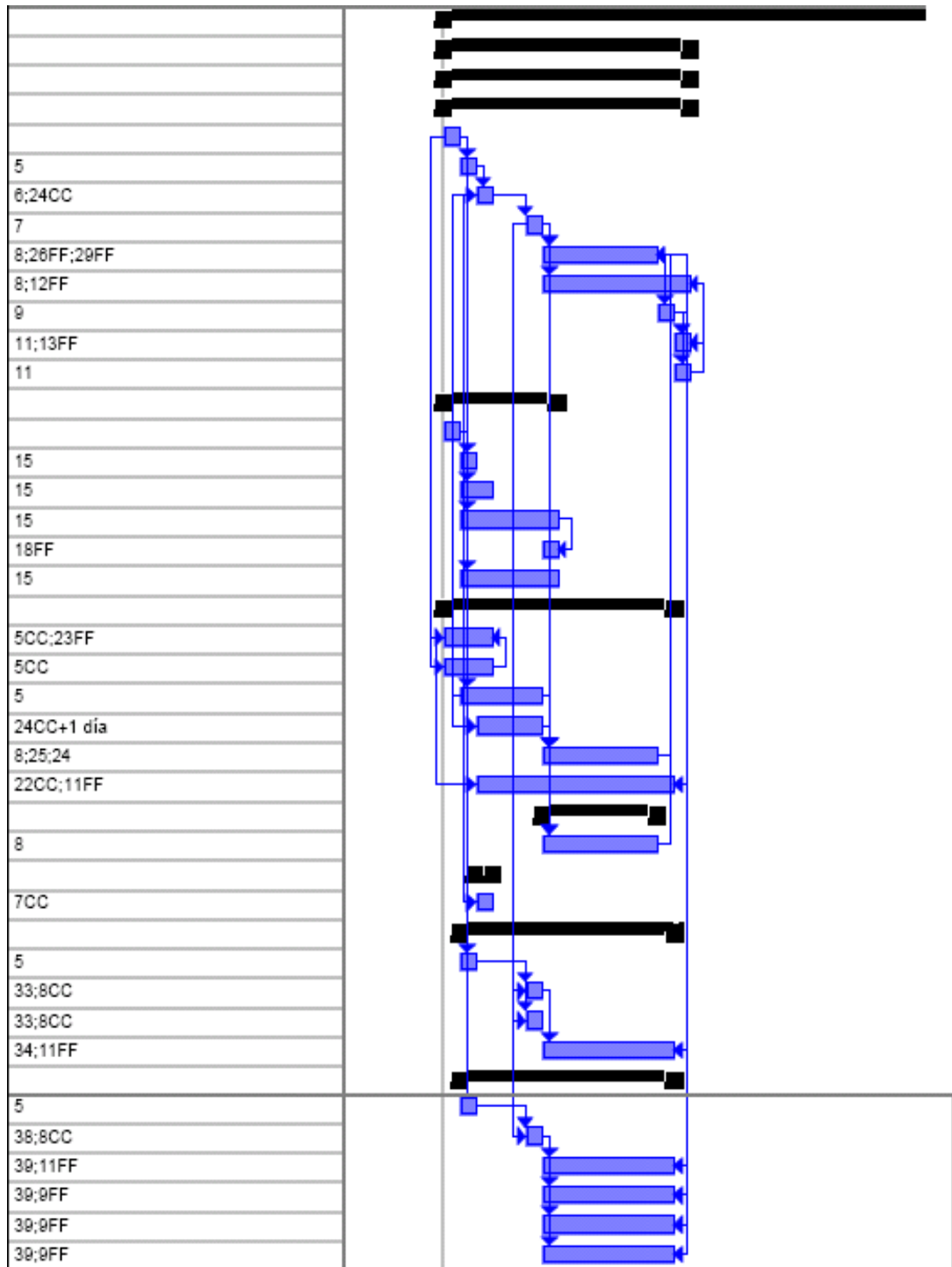


Figura 34: Dependencias entre las tareas del plan del RUP

Por lo tanto, y teniendo en cuenta todos los datos aportado, vamos a ver como queda representado en el Protégè el diagrama del proceso “proyecto” que deseamos definir. Vamos a resumir en diez

paso la definición del proceso. En un primer paso, tal y como se muestra en la Figura 35, tenemos definido el subproceso de Fase de inicio.

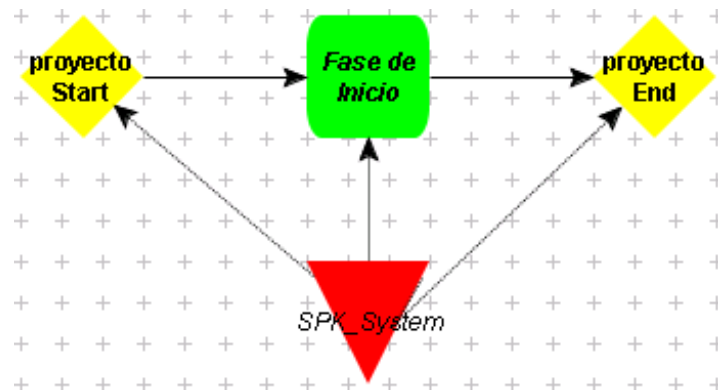


Figura 35: Paso 1 para realizar el nuevo proceso

Dentro del subproceso Fase de Inicio, vamos a definir otro subproceso llamado Iteración 1 de Inicio tal y como se nos muestra en la Figura 36.

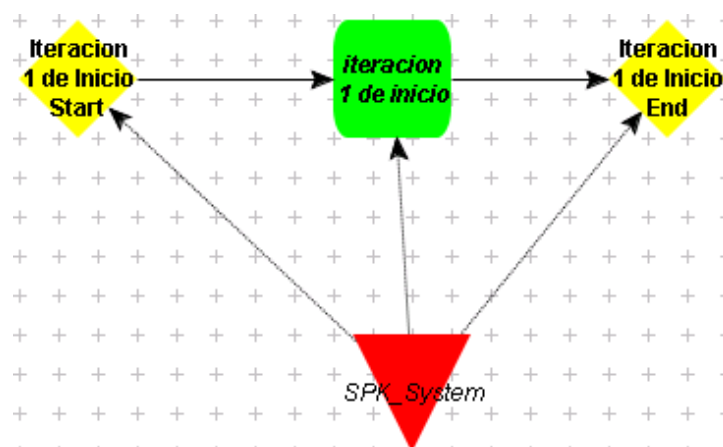


Figura 36: Paso 2 para realizar el nuevo proceso

Dentro del subproceso Iteración 1 de Inicio, vamos a tener la subIteración 1 de inicio, la cual va a tener definidas un conjunto de subprocesos. Podemos ver esta representación en la Figura 37. Lo que vamos a hacer a continuación, es entrar dentro de cada uno de estos subprocesos, e ir añadiéndoles todos los elementos que sean necesarios, ya sean tareas o más subprocesos.

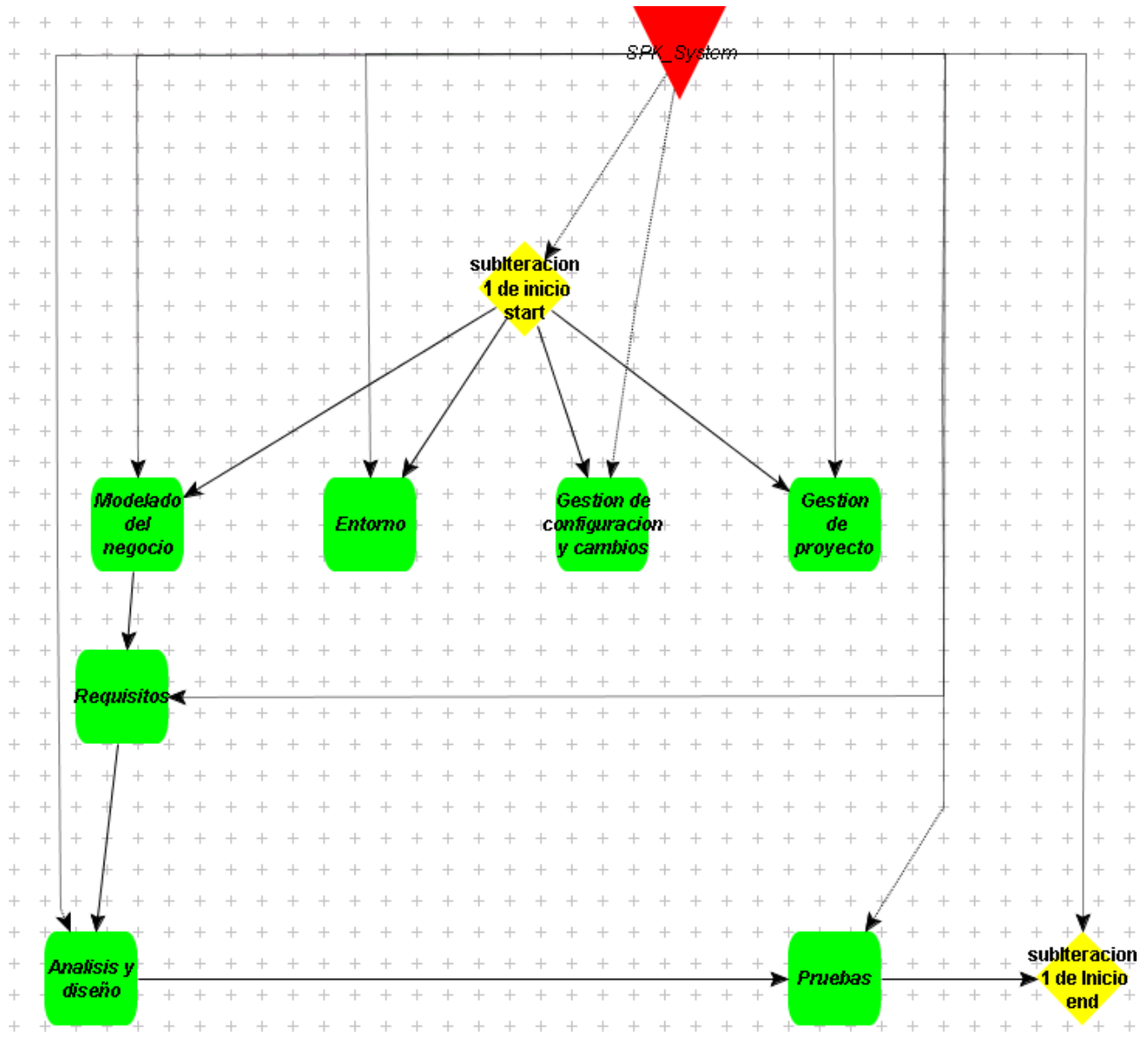


Figura 37: Paso 3 para realizar el nuevo proceso

Por lo tanto, vamos a recorrer el interior del subproceso Modelo del negocio, que se encuentra definido en la Figura 37. Podemos ver que es lo que se encuentra dentro de este subproceso en la Figura 38:

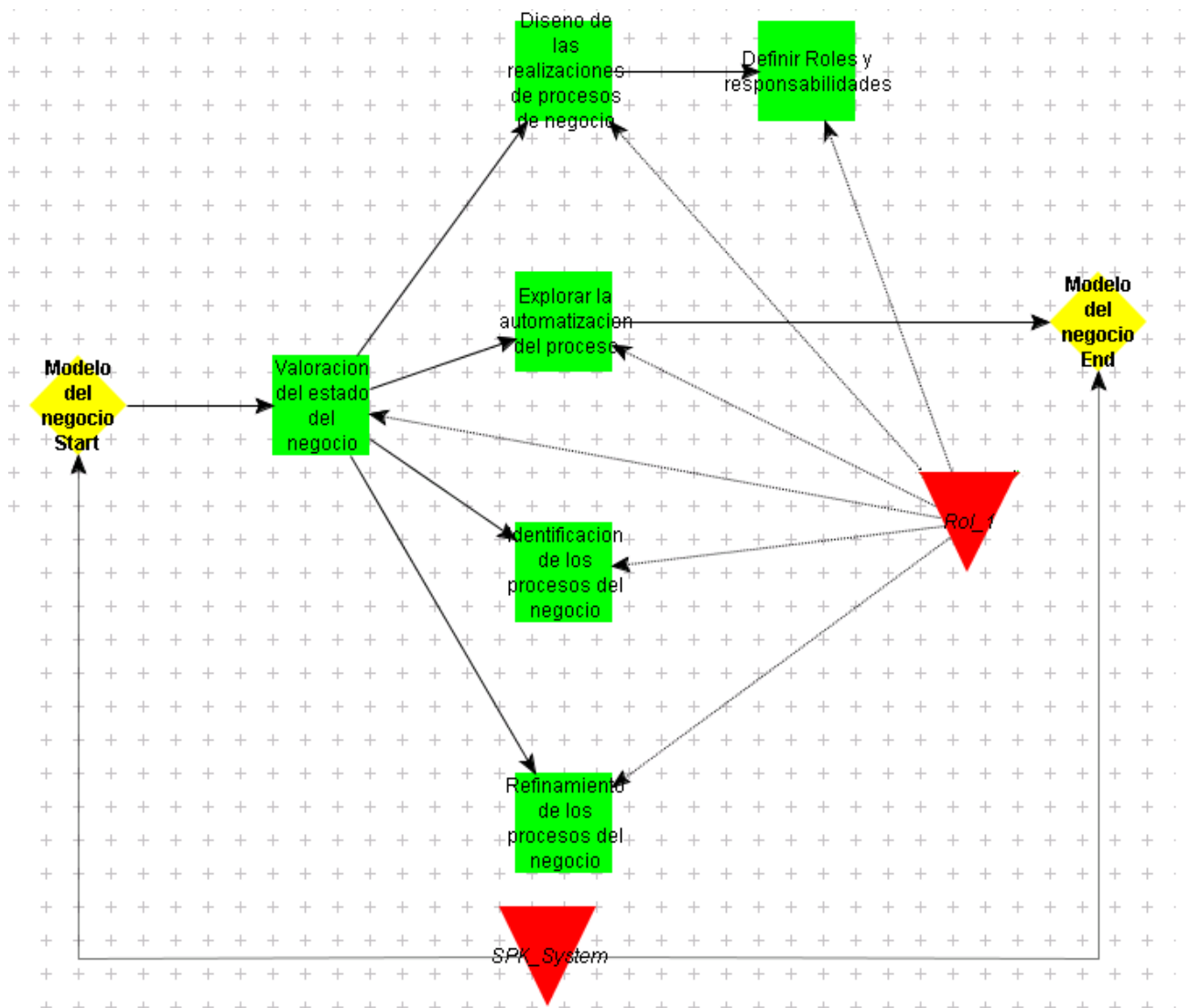


Figura 38: Paso 4 para realizar el nuevo proceso

Vamos a recorrer el interior del subproceso Entorno, que se encuentra definido en la Figura 37. Podemos ver que es lo que se encuentra dentro de este subproceso en la Figura 39:

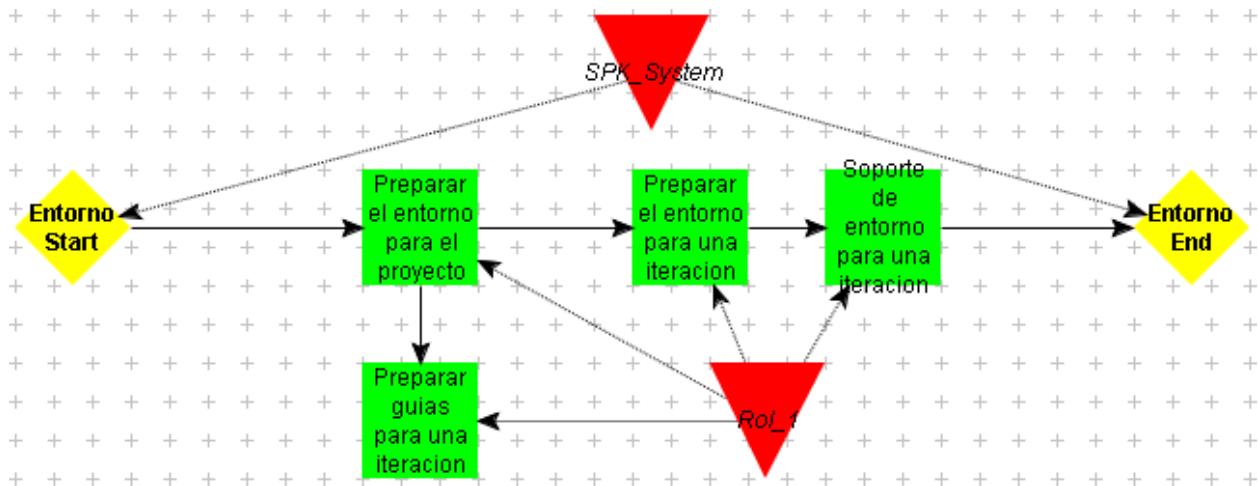


Figura 39: Paso 5 para realizar el nuevo proceso

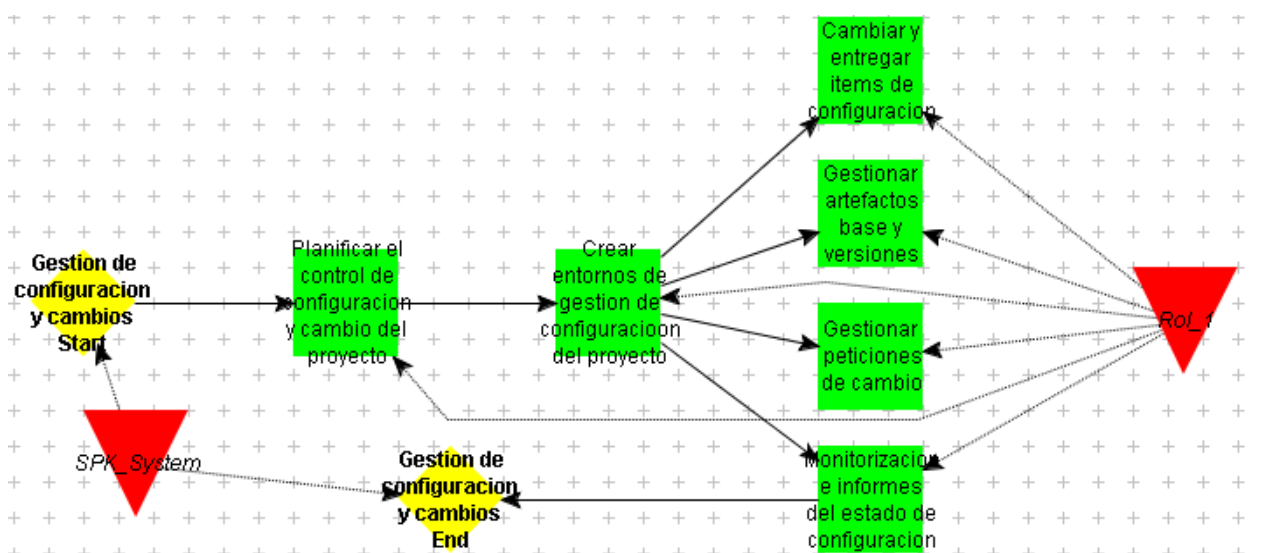


Figura 40: Paso 6 para realizar el nuevo proceso

Vamos a recorrer el interior del subproceso Gestión de configuración y cambios, que se encuentra definido en la Figura 37. Podemos ver que es lo que se encuentra dentro de este subproceso en la Figura 40:

Vamos a recorrer el interior del subproceso Gestión de proyecto, que se encuentra definido en la Figura 37. Podemos ver que es lo que se encuentra dentro de este subproceso en la Figura 41:

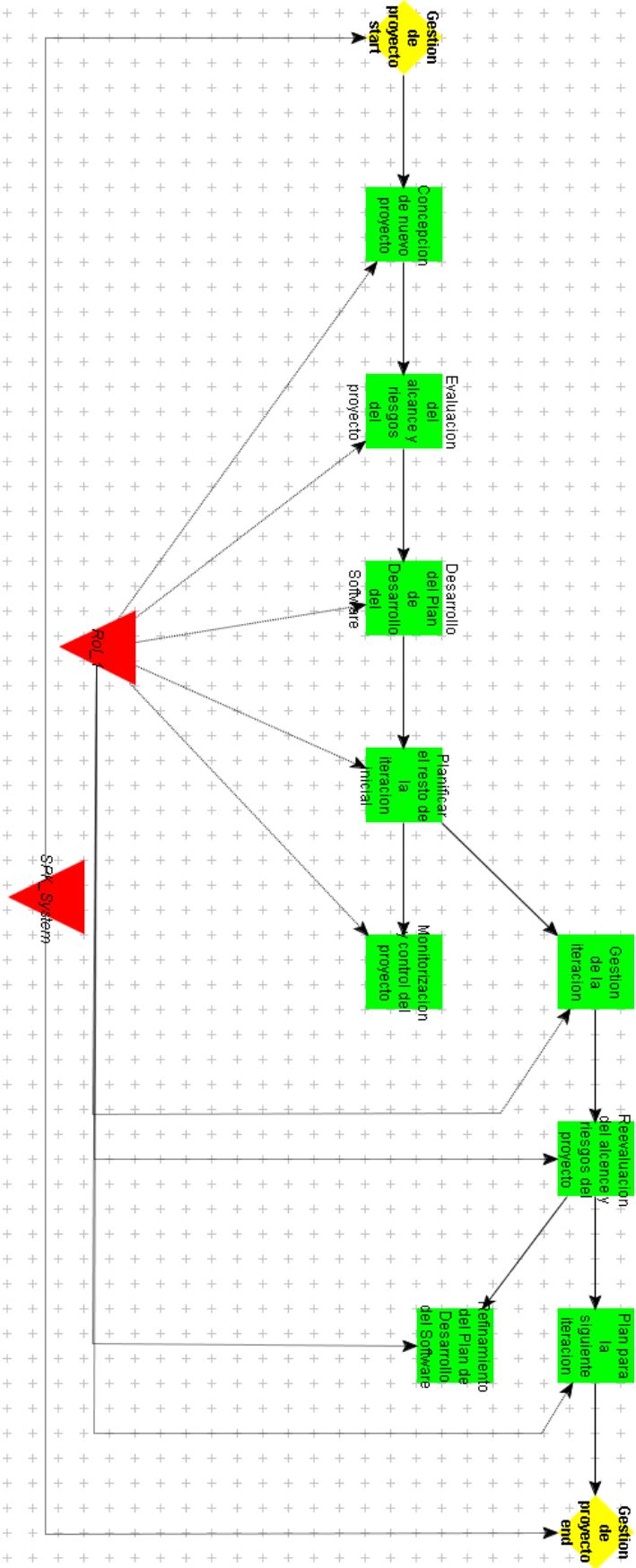


Figura 41: Paso 7 para realizar el nuevo proceso

Vamos a recorrer el interior del subproceso Requisitos, que se encuentra definido en la Figura 37. Podemos ver que es lo que se encuentra dentro de este subproceso en la Figura 42:

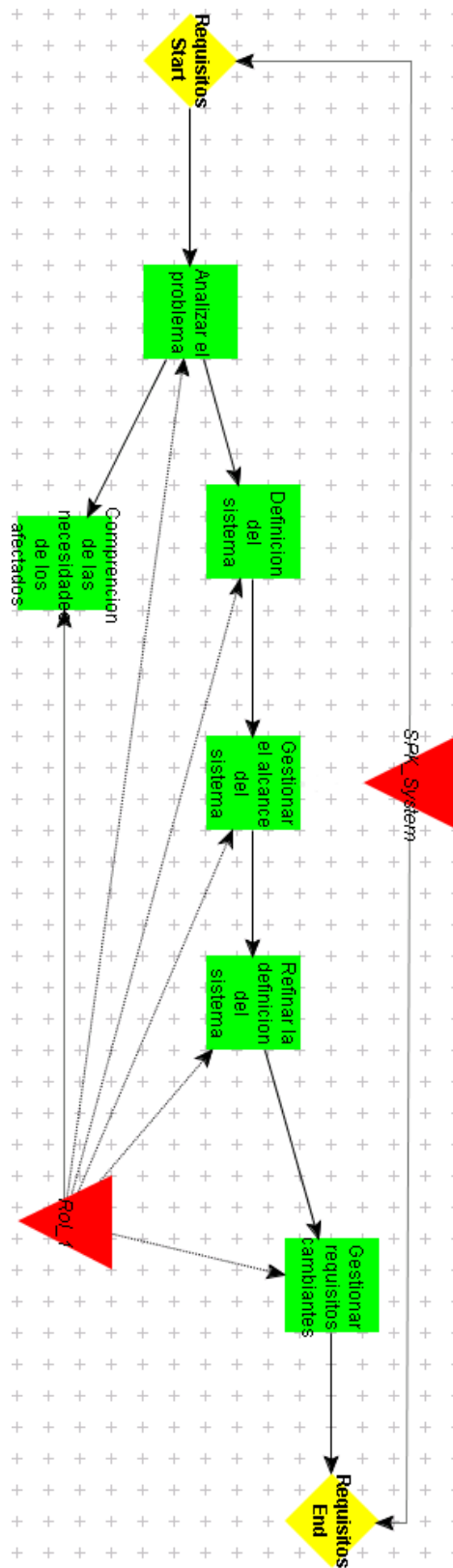


Figura 42: Paso 8 para realizar el nuevo proceso

Vamos a recorrer el interior del subproceso Análisis y diseño, que se encuentra definido en la Figura 37. Podemos ver que es lo que se encuentra dentro de este subproceso en la Figura 43:

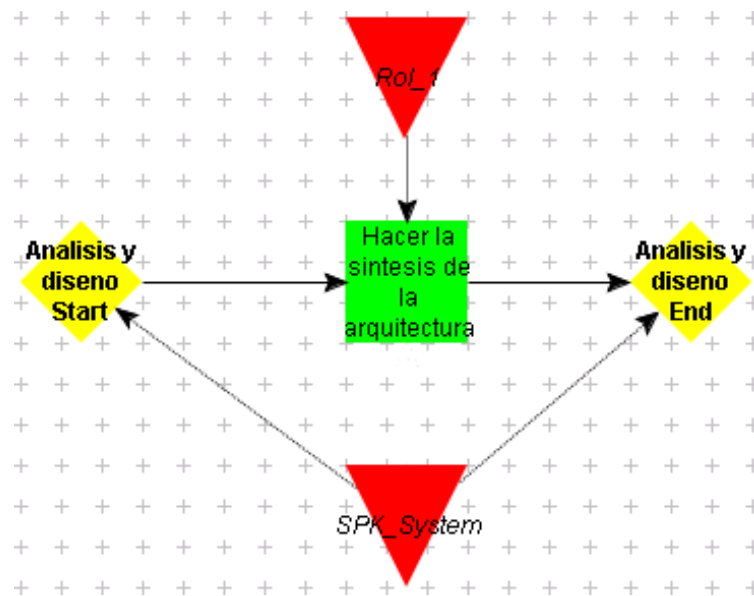


Figura 43: Paso 9 para realizar el nuevo proceso

Vamos a recorrer el interior del subproceso Pruebas, que se encuentra definido en la Figura 37. Podemos ver que es lo que se encuentra dentro de este subproceso en la Figura 44:

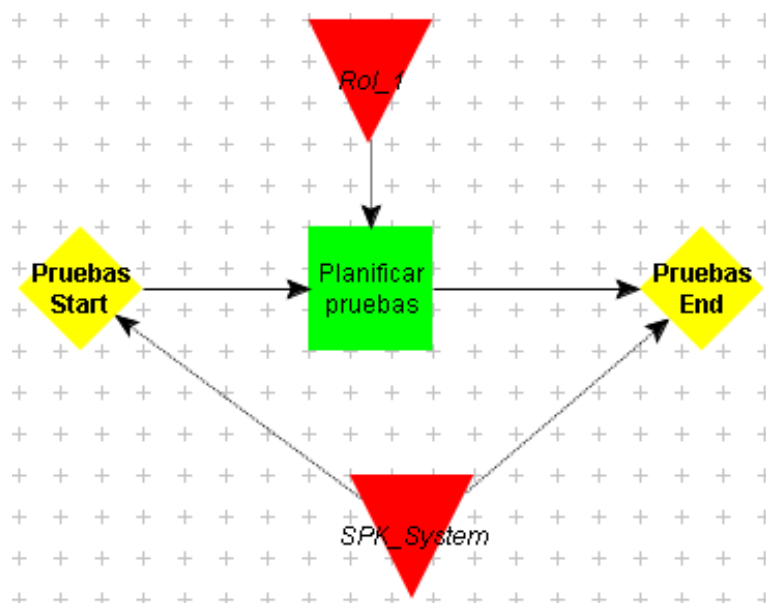


Figura 44: Paso 10 para realizar el nuevo proceso

Por lo tanto, cuando se desea realizar una planificación inicial para un nuevo proyecto, dependiendo del tipo de proceso que se seleccionase para la generación de un plan inicial, se accederá mediante el Protègè al proceso correspondiente, y se generará una planificación inicial, insertando en la base de datos tanto los datos sobre las tareas, como las dependencias definidas con el Protègè entre las mismas. Del mismo modo que en la versión anterior, al usuario se le mostrarán los datos en una tabla, desde la que podía editar las tareas, y realizar las modificaciones pertinentes en los tiempos de las mismas.

En una tercera versión, además de cambiar algunos de los datos que mostrabamos en las tablas a los usuarios, creamos una nueva funcionalidad, gracias a la cual mostrabamos tanto las tareas planificadas como el estado de la ejecución en la actualidad de las mismas en un diagrama gantt.

De este modo, podemos ver de un modo rápido y gráfico, cual era la ejecución esperada según la planificación, y cual es la situación actual de las tareas en realización.

Para distinguir ambos tipos de tareas, el diagrama se genera con dos colores tal y como podemos observar en la Figura 45; rojo para mostrar la planificación con los tiempos esperados, y azul para mostrar cual ha sido o esta siendo la ejecución real de las tareas.

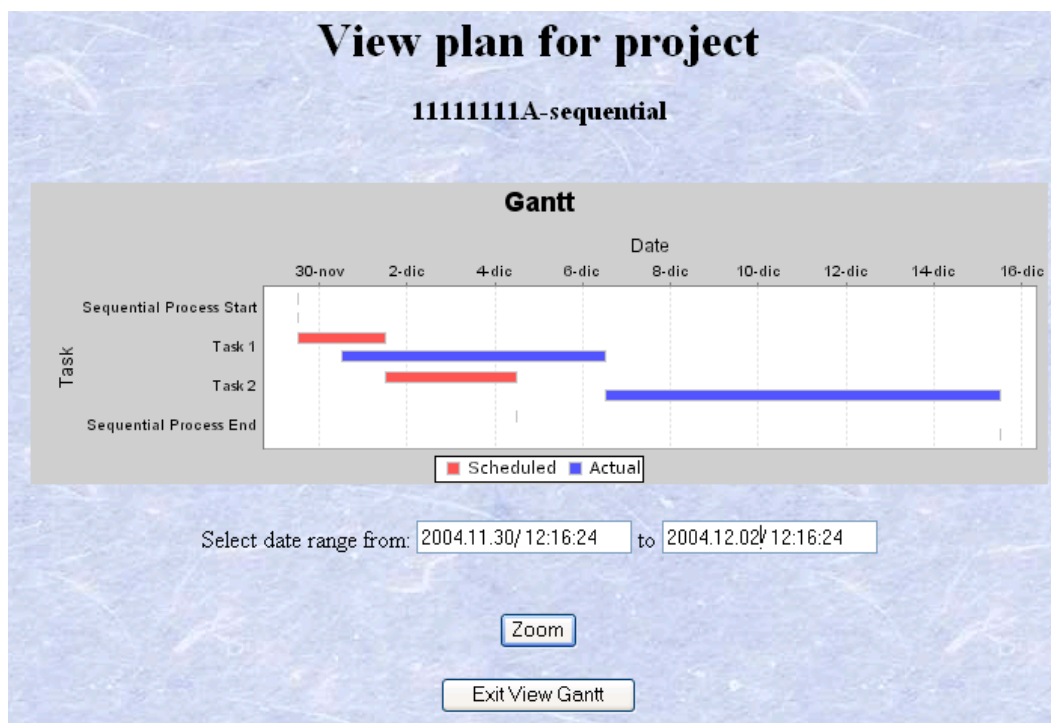


Figura 45: Diagrama gantt generado para un proceso secuencial

La última funcionalidad que añadimos en esta versión fue la de poder hacer un zoom del diagrama gantt, dándole el rango de fechas sobre las que el usuario pueda estar más interesado, llegando a mostrar un diagrama con una precisión de horas y minutos. Un ejemplo puede ser el que mostramos en la Figura 46, basándonos en el gantt mostrado anteriormente en la Figura 45:

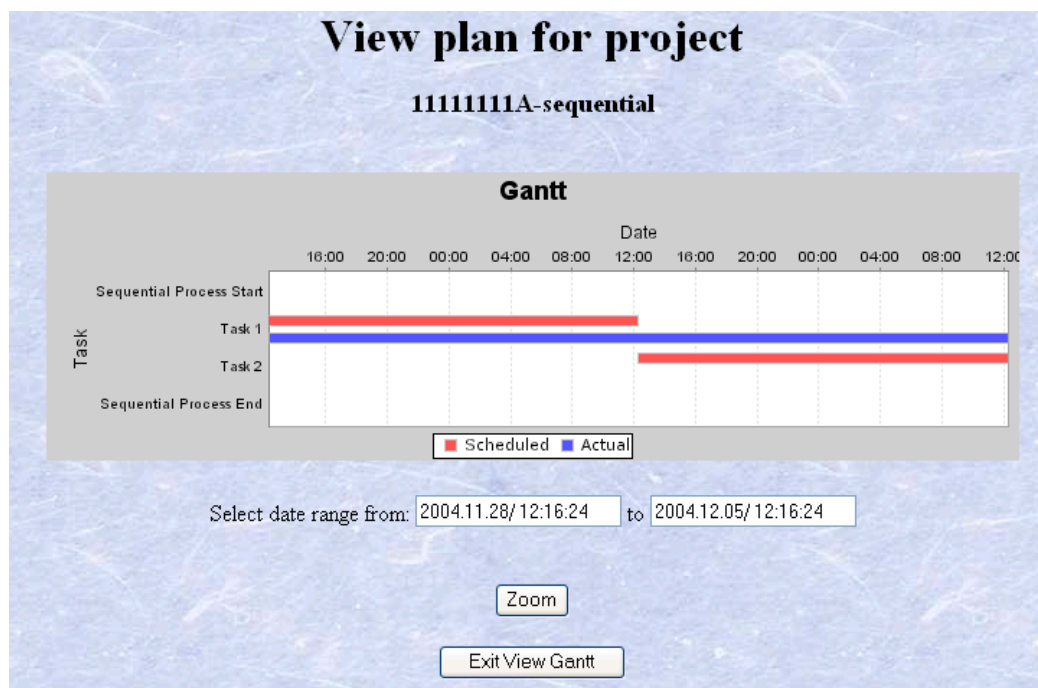


Figura 46: Zoom sobre el diagrama gantt generado para un proceso secuencial

18.- Anexos referentes a la gestión del proceso de elaboración de este proyecto

Durante todo el ciclo de vida del proyecto, y en cada una de las fases en las que se ha dividido el mismo, hemos mantenido continuas reuniones con el cliente siempre que estas eran necesarias, para indicarle la evolución del proyecto, y decidir las mejoras posibles para la aplicación web.

Como resultado, hemos ido realizando distintos prototipos para la aplicación GesProSoft. Los cambios más destacables entre los prototipos fueron el agregar la creación de un nuevo plan usando una herramienta de gestión del conocimiento para la creación de los modelos de procesos y el mostrar las tareas en un formato lo más parecido a los diagramas GANTT que se usan con herramientas de gestión de proyectos que hay en el mercado.

En el primer prototipo únicamente mostrábamos los datos de las tareas que formaban parte de la planificación de un proyecto en una tabla, pero no generaba una nueva planificación para un proyecto, sólo mostraba datos que había en la base de datos. En un segundo prototipo, generábamos un nuevo plan para un proyecto accediendo a Protègè, aunque los datos seguían mostrándose en una tabla. Este cambio marcó el hito principal del proyecto, ya que demuestra que es posible la generación automática de un plan de proyecto, a partir de un proceso definido. Esta automatización reduce drásticamente el tiempo necesario para la generación de un plan, ya que el usuario no necesita insertar las tareas del proyecto una por una. En un tercer prototipo, las tareas planificadas y las tareas en ejecución se mostraban en un diagrama GANTT, diferenciando en el mismo diagrama las tareas planificadas y reales con dos colores distintos. Esto hace que el plan de proyecto que mostramos sea mucho más legible y que los usuarios comprendan rápidamente toda la información que el diagrama GANTT le aporta.

18.1.- Planificación temporal para el prototipo

Antes de comenzar con la elaboración y construcción del proyecto GesProSoft, en la fase de inicio, en el artefacto llamado “Plan de desarrollo del software”, realizamos una planificación con las fases, incluyendo para cada una de ellas, las fechas en las que esperábamos fueran realizadas, y que tipo de herramientas se iban a utilizar para su realización.

En la Tabla 3, podemos ver cuales eran los tiempos esperados en un principio para la elaboración de todo el proyecto GesProSoft, aunque sin embargo, y tal y como podemos ver en la Tabla 4, no los hemos cumplido con exactitud. En algunos de los casos, como en las fases de inicio y transición, si comparamos los tiempos estimados con los tiempos reales, vemos que no hay una gran diferencia entre ellos, aunque no hemos cumplido los plazos con exactitud, sin embargo, es mucho más destacable la diferencia de tiempos que hay en las fases de elaboración y construcción.

En la fase de elaboración, el tiempo estimado era de 2,5 meses, sin embargo, esta fase se redujo a algo menos de un mes. Esto fue debido, a que no encontramos grandes problemas a la hora de instalar y probar el correcto funcionamiento de las herramientas mediante pruebas muy básicas. Por lo tanto, en esta fase, ahorramos una gran cantidad de tiempo.

<i>Iteración</i>	<i>Periodo</i>	<i>Duración</i>	<i>Formación</i>	<i>Herramientas</i>	<i>Fases</i>
1	Del 04/10/2004 al 16/11/2004	~ 1,4 meses	RUP, UML	MS FrontPage, Rational Rose, Word	Inicio
2	Del 16/11/2004 al 13/12/2005	~ 0,9 meses	Struts, RUP, UML, J2EE	Struts, Rational Rose, Eclipse, MyEclipse, JBoss , SQL Server 2000, Protégè	Elaboración
3	Del 13/12/2005 al 17/05/2005	~ 5,1 meses	Struts, Enterprise Java Beans, JSP, Servlets	Struts, Eclipse, MyEclipse, JBoss, SQL Server 2000, Protégè, jfree- chart	Construcción
4	Del 17/05/2005 al 10/07/2005	~ 2,7 meses		MS FrontPage, Word OpenOffice	Transición

Tabla 3: Tiempos reales que se han invertido para cada una de las fases para el proyecto GesProSoft

No sucedió así en la fase de construcción. En un primer momento tuvimos que dedicar un periodo de tiempo a la comprensión de la base de la que se partía, para poder añadir nuevas funcionalidades a la aplicación web ya existente.

Esto, y el hecho de conseguir el modo de representar gráficamente cada una de las tareas, así como el correcto funcionamiento a la hora de modificar los tiempos teniendo en cuenta las dependencias entre las tareas, y el hecho de tener que trabajar con la gestión de los procesos usando el Protégè para generar nuevas planificaciones de proyectos, hizo que fuera una labor más complicada de lo que esperábamos en un primer momento. Por lo tanto, el tiempo estimado que era únicamente de 3 meses, pasó a ser algo más de 5 meses. Sin embargo, conseguimos los objetivos que fueron marcados tanto por el director del proyecto, como por el desarrollador del mismo.

18.2.- Resumen del presupuesto para el prototipo

Teniendo en cuenta que el proyecto que vamos a desarrollar no es de gran envergadura y que en un principio no hay intenciones de comercializarlo, lo más destacable en el ámbito financiero es lo relacionado a los recursos que son necesarios para el desarrollo del proyecto. Dichos recursos serán los siguientes:

- Servidor o clusters de servidores
- PC
- Conexión a Internet
- Red rápida y estable

A pesar de ello realizaremos dos presupuestos uno desde el punto de vista de un proyecto fin de carrera teniendo en cuenta las facilidades proporcionadas por la universidad y otro suponiendo que se trata de un proyecto empresarial. En este presupuesto tendremos en cuenta el software y las horas de trabajo.

18.2.1.- Entorno educativo

<i>Software</i>	<i>Proveedor</i>	<i>Precio</i>
Windows XP Professional	Proporcionado con el portátil	0,00 €
Office XP Professional	Proporcionado con el portátil	0,00 €
Microsoft SQL Server 2000	Proporcionado por la Facultad	0,00 €
Eclipse		0,00 €
MyEclipse	Proporcionado por el grupo de investigación	0,00 €
JBoss	JBoss Inc.	0,00 €
Rational Unified Process	Proporcionado por el grupo de investigación	0,00 €
Rational Rose	Proporcionado por el grupo de investigación	0,00 €

<i>Horas de trabajo</i>	<i>€/hora</i>	<i>Total</i>
700 horas	120 €/hora	84.000,00 €

<i>Total</i>	84.000,00 €
--------------	-------------

18.2.2.- Proyecto empresarial

En la siguiente tabla enumeran las diferentes licencias que hemos necesitado para la elaboración del producto:

Software	Proveedor	Precio Total
Windows XP Professional	Microsoft	299,00 €
Office XP Professional	Microsoft	579,00 €
Microsoft SQL Server 2000	Microsoft	20.000,00 €
Eclipse	Eclipse Foundation	0,00 €
MyEclipse	Genuitec LLC	29,95 €
JBoss	JBoss Inc.	0,00 €
Rational Unified Process	Rational	871,00 €
Rational Rose	Rational	5.220,00 €
OpenOffice	OpenOffice.org	0,00 €
Total software		26.998,95 €

Table 5: Costes asociados a la fase de elaboración.

En la siguiente tabla vienen desglosados los costes referentes al personal, directamente imputables al cliente.

Tiempo de trabajo	€/hora	Total/persona
700 horas	120 €/hora	84.000,00 €
Total		110.998,95 €

18.3.- Conclusiones

Las empresas encargadas de desarrollar software dan soluciones a los sectores que les rodean, y sin embargo, a veces no buscan soluciones que les faciliten su trabajo. Uno de los casos en el que las herramientas utilizadas por los desarrolladores no cumplen con todas sus necesidades es la gestión de los proyectos software.

Las herramientas actuales requieren de mucho esfuerzo y tiempo en el proceso de creación de una nueva planificación para un proyecto. Por lo tanto, una de las necesidades de las empresas de realización del software es la de realizar de una manera rápida y eficaz nuevas planificaciones para proyectos de sus clientes, a los cuales deben de dar un presupuesto lo más ajustado posible.

Aunque muchos de los proyectos que una organización afronta comparten características similares, permitiendo la generación de plantillas de proyecto, no es denominador común que estas planti-

llas lleguen a definirse. Por este motivo, cada vez que se inicia un nuevo proyecto se hace necesario afrontar la planificación desde cero, sin aprovecharse del conocimiento adquirido en las anteriores planificaciones. Esta deficiencia provoca que las planificaciones sean más costosas en esfuerzo por parte de la organización y menos precisas en cuanto a las estimaciones realizadas en ellas.

Para evitar ésto, hemos desarrollado un prototipo de aplicación web llamado GesProSoft, con la cual, de un modo menos costoso para los desarrolladores de software, vamos a poder hacer planificaciones en un breve espacio de tiempo, y sin que esto suponga una pérdida de eficacia en la estimación de los tiempos. Esta es la funcionalidad más destacable en el prototipo GesProSoft, ya que es la mejora respecto a las herramientas que actualmente se comercializan. Para la creación de una nueva planificación utilizamos la herramienta Protège, que es una herramienta para adquisición y gestión del conocimiento. Con ella hemos definido todos los procesos con los que hemos trabajado, incluidas las estimaciones de tiempo de las tareas.

En posteriores versiones, se podría esperar que se tratase de algún modo la adquisición del conocimiento, de manera que si para un tipo específico de proyecto siempre se genera un retardo en una de sus tareas, realice una modificación en el tiempo de duración estimado para dicha tarea, para que en posteriores planificaciones la estimación sea más precisa. Esto dotaría a la aplicación de una precisión cada vez mayor a medida que va pasando el tiempo.

Todo lo comentado hasta ahora es sobre la creación de nuevas planificaciones. Sin embargo, vamos a fijarnos en la gestión de las planificaciones una vez que estas ya existen.

En el prototipo GesProSoft hemos tratado el caso de modificaciones en los tiempo de inicio y finalización para una tarea. Para ello, hemos tenido en cuenta las dependencias existentes entre las tareas, de manera que las modificaciones funcionasen correctamente. Las dependencias que hemos tenido en cuenta han sido dependencias del tipo final-inicio, es decir, que una tarea B no podía empezar hasta que una tarea A hubiera finalizado. Esto es mejorable para posteriores versiones, ya que las dependencias entre tareas pueden tener representaciones tales como inicio-inicio, final-final, inicio-final, o incluso de pueden tratar las sincronizaciones, de manera que una tarea no pueda comenzar hasta que un conjunto de ellas se hayan ejecutado.

Respecto a la representación gráfica de los diagramas, decidimos mostrarlos de manera que se puedan ver tanto las tareas en su planificación inicial, como las tareas en su ejecución real. Esto hace que podamos percibir rápidamente como va la marcha del proyecto. Sin embargo, las depen-

dencias entre las tareas se muestran en una tabla aparte, lo cual es una solución, aunque es mejorable para posteriores versiones.

Debido a que para la creación de este prototipo debíamos partir de una base ya implementada, la tecnología que hemos utilizado ha sido la misma que se usó para la primera versión: Java, Struts e Hibernate. Esto requirió un esfuerzo en aprendizaje aunque nos ayudó a cumplir los objetivos para la realización del proyecto.

Hemos realizado la fase de análisis y diseño utilizando la herramienta Rational Rose, con la cual hemos realizado todos los casos de uso, diagramas, representación de clases del proyecto, etc.

Como deseábamos hacer un producto de calidad, hemos seguido durante todo el desarrollo del proyecto el método del RUP. Este método marca continuamente el camino a seguir a lo largo de todo el proyecto, lo que nos facilita mucho realizar el proyecto, aunque también requiere mucho tiempo el realizar cada uno de los pasos que se definen en dicho método.

A modo personal, destacaría que desarrollar un proyecto fin de carrera en el que trabajar con tecnología no conocida, como eran Struts e Hibernate, y con herramientas que tampoco habíamos utilizado a lo largo de la carrera, como Rational Rose, Eclipse, Microsoft SQL Server, etc, ha hecho que se realizase un esfuerzo mayor para conseguir dar una solución al problema planteado. Sin embargo, todo ese esfuerzo me ha sido muy útil, ya que me ha permitido acercarme a tecnologías muy interesantes para futuros desarrollos. Esa fue una de las razones por las que decidí escoger dicho proyecto, y es que desde el inicio el director describió el modo de trabajo, y explicó tanto el método de trabajo que se iba a seguir (RUP), como las herramientas que íbamos a utilizar.

Para finalizar, elaborar un proyecto en el que hemos realizado un prototipo de cierta complejidad, en el que ha habido que trabajar con distintas y diversas tecnologías y herramientas hace que el aprendizaje haya sido global, tanto en disciplinas de gestión de proyectos (método de desarrollo), como en herramientas propias para el desarrollo de aplicaciones (entornos de análisis y diseño, entornos de programación, frameworks específicos, etc.).