

Proyecto: NeN

Plan de proyecto del software

(Formato Pressman)

Miembros del equipo:

Rodrigo Arranz López	Laura María de Castro Saturio
Mónica Patricia Chachalo Iles	Cristina Valentina Espinosa Victoria
Giorgio García Missana	José Javier Martínez Pagés
Manuel Sánchez Pérez	Abel Serrano Juste

Control de cambios

Número de versión	Fecha	Autores	Descripción
VERSION 1	18/01/2013	José Javier Martínez Pagés	Redacción de la introducción (con excepción de “Aspectos de rendimiento” y “Restricciones y técnicas de gestión”
VERSION 1	20/01/2013	Rodrigo Arranz López	Redacción del apartado “6. Organización de personal”
VERSION 1	29/01/2013	Laura María de Castro Saturio	Redacción del apartado “7 Mecanismos de seguimiento y control”
VERSION 2	23/02/2013	José Javier Martínez Pagés	Redacción de los apartados “Aspectos de rendimiento” y “Restricciones y técnicas de gestión” de la introducción
VERSION 2	24/02/2013	Abel Serrano Juste	Terminado apartado 5 “Recursos del proyecto”
VERSION 2	25/02/2013	Mónica Chachalo Iles	Redacción del apartado 2 “Estimaciones del proyecto”
VERSION 2	25/02/2013	Valentina Espinosa Victoria	Apartados 4.2 y 4.3 hechos

VERSIÓN 2	26/02/13	Mónica Chachalo	Corrección del punto de Estimaciones del proyecto, del apartado: Datos históricos y añadida información del al apartado de estimaciones de coste y esfuerzo
VERSION 2	26/02/13	Valentina Espinosa Victoria	Terminada la planificación temporal
VERSION 2	28/02/13	Manuel Sánchez	Terminada la gestión de riesgos

Índice

Índice

1. Introducción

- 1.1 Propósito del plan
- 1.2 Ámbito del proyecto y objetivos
 - 1.2.1 Declaración del ámbito
 - 1.2.2 Funciones principales
 - 1.2.3 Aspectos de rendimiento
 - 1.2.4 Restricciones y técnicas de gestión
- 1.3 Modelo de proceso

2. Estimaciones del proyecto

- 2.1 Datos históricos
- 2.2 Técnicas de estimación
- 2.3 Estimaciones de esfuerzo, coste y duración

3. Estrategia de gestión del riesgo

- 3.1 Análisis del riesgo
- 3.2 Estudio de los riesgos
- 3.3 Plan de gestión del riesgo

4. Planificación temporal

- 4.1 Estructura de descomposición del trabajo o Planificación temporal
- 4.2 Gráfico Gantt
- 4.3 Red de tareas
- 4.4 Tabla de uso de recursos

5. Recursos del proyecto

- 5.1 Personal
- 5.2 Hardware y software
 - 5.2.1 Hardware
 - 5.2.2 Software
- 5.3 Lista de recursos
 - 5.3.1 Fungibles
 - 5.3.2 No fungibles

6. Organización del personal

- 6.1 Estructura de equipo
- 6.2 Informes de gestión

7. Mecanismos de seguimiento y control

- 7.1 Garantía de calidad y control
- 7.2 Gestión y control de cambios

8. Apéndices

1. Introducción

1.1 Propósito del plan

El proyecto NeN ha sido creado para facilitar y mejorar la experiencia de la gestión del departamento de cirugía hospitalaria, permitiendo una gestión más particular que la que ofrecen otros programas de gestión hospitalaria más generales, que carecen de la particular gestión de pacientes, personal, espacios e instrumental que se da en los bloques quirúrgicos; se pretende con esto reducir el tiempo de las actividades de gestión al ofrecer un modelo más compacto y ahorrar el tiempo que se dedica a adaptar las aplicaciones genéricas a la gestión particular del departamento quirúrgico.

1.2 Ámbito del proyecto y objetivos

1.2.1 Declaración del ámbito

Al comienzo del proyecto decidimos fijarnos en un proyecto de gestión hospitalaria, ya que nos atraía el tema, pero nos costó tiempo (más del que nos hubiera gustado) y esfuerzo determinar el ámbito de nuestro proyecto.

Decidimos que nuestro proyecto se iba a centrar en un programa de gestión del departamento de cirugía de un hospital, restringiéndonos a la división de esta sección en bloques quirúrgicos, y gestionando: materiales (inventario y pedidos, siempre internos al propio hospital, nunca gestionamos pedidos a un proveedor externo desde el propio departamento de cirugía [decisión tomada tras encontrar diferencias entre la información recopilada en lo que concierne a este aspecto]), pacientes (lista de espera, preoperatorio, postoperatorio), personal (horarios de trabajo, asignaciones a los bloques quirúrgicos según especialidad, turnos de emergencia) y espacios (espacio físicos de los bloques quirúrgicos, camas).

1.2.2 Funciones principales

Las funciones de la aplicación serán, teniendo en cuenta el ámbito:

Respectivo a instrumental: manejo del inventario del instrumental quirúrgico, pedidos internos, control de la esterilización, desinfección, limpieza, empaquetado, estado de degradación, nivel de contacto con los tejidos y nivel de riesgo infeccioso, así como una proporcionar una descripción de cada procedimiento de las características mencionadas anteriormente.

Respectivo a pacientes: alta de pacientes (datos personales, fecha de ingreso, descripción médica), baja de pacientes (fecha de baja, cantidad a pagar) , tipo de operación realizada/a realizar, fecha de operación, observaciones, médicos asignados, diagnóstico, historial.

Respectivo a personal: horarios, operaciones, trabajo del mes/día por médico, listado de personal activo diurno/nocturno, turnos de guardia, documentación, datos de personal.

Respectivo a espacios: servicio de camas, espacio físico (pasillos, área de recepción y

pre-anestesia, quirófanos, área postoperatoria), mantener un registro de los recursos estructurales del bloque quirúrgico y del equipamiento disponible, gestión de quirófanos reservados para emergencias.

1.2.3 Aspectos de rendimiento

Basándonos en la memoria del 2011 del hospital La Paz: http://www.madrid.org/cs/Satellite?cid=1142399368017&language=es&pagename=HospitalLaPaz%2FPagina%2FHPAZ_contenidoFinal

Hemos decidido que el número nominal de usuarios de nuestra aplicación no debería superar los 1000 ni bajar de los 800, soportando 600 usuarios simultáneos.

Usando los datos de actividad de los quirófanos en los respectivos intervalos horarios del día hemos determinado que el sistema no necesita rendir al 100% durante la noche, pudiendo reducirse su actividad total al 60% al ver reducido el uso de determinados módulos (gestión de personal, servicio de pedidos de materiales), cabe señalar que la gestión de pacientes y la posibilidad de buscar información sobre los materiales debe estar operativa por la noche debido a las urgencias.

Durante el día, el tiempo de respuesta de todos los módulos visibles en una situación de caso peor (el máximo número de usuarios simultáneos conectados al mismo tiempo modificando campos a la vez) debería ser menor o igual a 6 segundos, viéndose reducido este tiempo de respuesta a un tiempo perteneciente al intervalo [0,1 , 1] segundos cuando hay actividad normal.

Asimismo, si un usuario permanece conectado más de quince minutos sin detectarse actividad por su parte, se cerrará su sesión y deberá volver a iniciarla.

1.2.4 Restricciones y técnicas de gestión

Hemos encontrado las siguientes restricciones iniciales:

- Adaptación a la ley de Protección de Datos.
- Compatibilidad con los protocolos de comunicación utilizados en el hospital.
- Compatibilidad de nuestro sistema con el servidor de datos utilizado en el hospital HP ProLiant DL380 G7 Entry.
- Compatibilidad de nuestro sistema con el servidor de copias de seguridad utilizado en el hospital HP StorageWorks 1/8 G2 Tape Autoloader Ultrium 920.

Como es natural, esperamos que las restricciones aumenten y cambien a lo largo del desarrollo del proyecto.

En cuanto a la técnica de gestión utilizada, nos decidimos por una aplicación particular del modelo Descentralizado Democrático; dividimos el grupo en subgrupos y los miembros de un mismo subgrupo se coordinan para llevar a cabo sus tareas correspondientes, a pesar de esta división, los subgrupos tienen visibilidad de las tareas en proceso de realización que se llevan a cabo en los demás subgrupos, y no han sido extrañas las ocasiones en las que un miembro de un subgrupo ha ayudado a otro subgrupo en sus tareas. En el proyecto no hemos tenido nunca un jefe ni ninguna otra figura que se le parezca, efectuando la toma de decisiones bien por votación tras

reuniones celebradas semanalmente o bien mediante la utilización de determinadas páginas de web que permiten votaciones.

1.3 Modelo de proceso

Debido a la naturaleza del proyecto, hemos decidido seguir un modelo híbrido entre dos submodelos distintos del modelo en Cascada, estos son: el modelo en Cascada con subproyectos y el modelo Lineal Secuencial de Pressman.

El modelo en Cascada con subproyectos es un submodelo del cascada con la particularidad de que tras los requisitos y el diseño de la arquitectura éste se divide en subproyectos, integrando al final los componentes para formar el sistema completo; decidimos usar este modelo debido a la inmensidad del problema a tratar (la gestión del departamento de cirugía) convenía establecer rápidamente una división en grupos de trabajo (en nuestro caso: grupo de gestión de Pacientes, grupo de gestión de Personal, grupo de gestión de Materiales y grupo de gestión de Espacios) así como la interdependencia entre ellos (ver diagrama en el SRS). A pesar de tener que hacer la división en módulos después del diseño y de establecer los requisitos, nosotros decidimos hacerlo antes de acabar por completo con el análisis de requisitos; esto nos facilitó en gran medida el avance de la aplicación al poder centrarse cada grupo en sus complejos módulos. La principal dificultad encontrada al aplicar este modelo ha sido (como en cualquier otro modelo en cascada) la dificultad de dar marcha atrás y conseguir establecer una interconexión realista y eficiente entre los distintos módulos de trabajo, particularmente en esto último ya que hasta que la establecimos estuvimos trabajando dando palos de ciego.

El modelo lineal secuencial de Pressman es un submodelo del cascada especializado en la ingeniería de Sistemas, esto es, fijando el desarrollo del proyecto en el análisis y en el diseño, y dejando aparte la codificación y las pruebas. Decidimos aplicar este modelo en cada módulo de trabajo, ya que estos módulos serían más tarde parte de un sistema más grande de trabajo. Los principales problemas encontrados fueron, al igual que en el modelo de cascada con subproyectos, la dificultad de dar marcha atrás, que nos ocurrió ocasionalmente sobre todo al encontrar información contradictoria al acudir a distintas fuentes para establecer los requisitos.

2. Estimaciones del proyecto

2.1 Datos históricos

Como dato histórico se ha seleccionado un proyecto que cuyo objetivo es ofrecer un mejor servicio al cliente.

Este proyecto tendrá una duración aproximada de 6 a 7 meses, Technology and Development Solutions S.C. realizo el Kick Off oficial del proyecto de implementación MoProSoft 2010, el estándar NYCE (Normalización y Certificación Electrónica, A.C.)

El estándar MoProSoft ofrece una colección de las mejores prácticas del sector para procesos relacionados con el desarrollo de software, que ayudan a mejorar aspectos fundamentales de la empresa y que abarcan la planeación estratégica, la mejora de las capacidades competitivas, la administración de los recursos y servicios necesarios para la marcha de la empresa, la administración de los proyectos tanto internos como externos, así como la metodología de desarrollo y mantenimiento de software. Todo este conjunto de buenas prácticas tiene el propósito de generar una empresa competitiva, capaz de cumplir sus compromisos en tiempo y forma, con la Calidad requerida por los clientes.

2.2 Técnicas de estimación

Nuestro software estará conectado a varios periféricos como: ratón, pantalla e impresoras.

Utilizaremos LDC como variable de estimación.

El ámbito del software:

El usuario (jefe de departamento) controlará e interactuará con el sistema a través de una interfaz de usuario que exhibirá características de un buen diseño de interfaz hombre-máquina. Todos los datos del personal, datos de pacientes y datos de materiales y espacios y cualquier información adicional se mantendrán en una base de datos. Los módulos de análisis del diseño se desarrollarán de forma que produzcan la salida requerida en una forma que pueda ser mostrada en una gran variedad de dispositivos gráficos. El software estará diseñado para poder controlar e interactuar con dispositivos periféricos tales como un ratón, una impresora y una pantalla.

Se han identificado las siguientes funciones principales:

- Interfaz de usuario y facilidades de control (IUFC)
- Análisis de datos del personal(ADP)
- Análisis de datos de pacientes(ADPa)

- Análisis de datos de materiales y espacios(ADME)
- Gestión de la base de datos (GBD)
- Facilidades de visualización de gráficos de computadora (FVGC)
- Control de periféricos (CP)
- Módulos de análisis del diseño (MAD)

Función	Optimista	Más probable	Pesimista	Esperada	\$/línea	Línea/mes	Costo	Esfuerzo
Control de la Interfaz de usuario	800	2400	2650	2340	14	315	32760	2
Análisis de datos del personal	4100	5200	7400	5380	20	220	107600	3
Análisis de datos de pacientes	4600	6900	8600	6800	20	220	136000	1
Análisis de datos de materiales y espacios	4100	5200	7400	5380	20	220	107600	3
Gestión de la base de datos	2950	3400	3600	3600	18	240	60300	1
Visualización de gráficos	4050	4900	6200	4950	22	200	108900	1
Control de periféricos	2000	2100	2450	2140	28	140	59920	2
Análisis del diseño	6600	8500	9800	8400	18	300	151200	1
	Total			38740 LDC estimadas			764280 Costo Estimado del proyecto	

* Estimación del esfuerzo requerido: cada una de las funciones se realizarán por los mismos miembros del grupo, es decir, que realizaremos varias actividades a la vez. Por tanto aunque el total del esfuerzo estimado es de 14 personas, nosotros únicamente contamos con 7 personas por tanto cada miembro del grupo tendrá que trabajar el doble para poder completar dicha estimación y esfuerzo.

Se tiene en cuenta que el esfuerzo y el coste registrado en la tabla incluye todas las actividades (análisis, diseño, codificación y prueba) y no solo la codificación.

2.3 Estimaciones de esfuerzo, coste y duración

Se ha delimitado la tabla de estimaciones en (personas/horas a la semana) que acapara cada tarea de ingeniería del software de cada función

Tareas Funciones	Análisis de requisitos	Diseño	Codificación	Prueba	Total
IUFC	1.0	2.0	0.5	3.5	1.0
ADP	2.0	10.0	4.5	9.5	26
ADPa	2.5	12.0	6.0	11.0	31.5
ADME	2.0	10.0	4.5	9.5	26
GB	2.0	6.0	3.0	4.0	15
FVGC	1.5	11.0	4.0	10.5	27
CP	1.5	6.0	3.5	5.0	16
MA	4.0	14.0	5.0	7.0	30
Total	16.5	71.0	31.0	60	181.5
Tarifa(€)	52,00	48,00	42,50	45,00	
Costo(€)	858	3.408	13.175	2.700	20.141

El tiempo estimado de coste en horas es de 181.5h, repartidas a su vez, a cada uno de los integrantes del grupo.

Además se refleja en la tabla, el coste en horas por persona dedicados en cada fase

* Las tarifas son aproximadas, (han sido obtenidas de las transparencias), cada etapa tiene un coste diferente y además depende de la complejidad de la fase.

Además al coste, tenemos que tener en cuenta los gastos asociados:

Coste estimado de la mano de obra: 50€

Coste estimado del servidor donde se alberga la base de datos: 300€

Coste estimado de los equipos y medios para la utilización del software: 250€

Coste estimado de licencias útiles en nuestro sistema: 300€

3. Estrategia de gestión del riesgo

3.1 Análisis del riesgo

Existen tres tipos básicos de riesgos potenciales dentro de un proyecto de desarrollo de software: Los riesgos relacionados con el equipo de desarrollo, los riesgos relacionados con las tecnologías de desarrollo, y los riesgos relacionados con los objetivos de desarrollo.

En el caso del proyecto NeN, hemos identificado varios riesgos potenciales que entran dentro de dicha clasificación:

En primer lugar se encontraba la posibilidad de que los objetivos de nuestro proyecto cambiaran en mitad del desarrollo. En concreto, cabía la posibilidad de que aunque en un primer momento se definió como un proyecto de especificación de requisitos y diseño, fuera necesaria realizar la implementación del mismo.

Por otro lado, existía un alto riesgo de que el proyecto corriera peligro gracias a problemas entre los integrantes del equipo. Ésto es un hecho muy importante, ya que desde un primer momento definimos la organización del equipo de manera “democrática”, sin ningún líder definido. En el momento en el que hubiera algún problema con algún integrante del equipo, el desarrollo del proyecto podría ralentizarse o incluso quedarse paralizado.

Por último, muy relacionado con el primero, se encuentra el hecho de que ninguno de los integrantes del equipo tiene experiencia en el desarrollo de software de gestión. De manera que en el caso de que tuviera que llevarse a cabo la implementación, el equipo tendría que estudiar que tecnologías utilizar, y aprender a utilizarlas en poco tiempo.

3.2 Estudio de los riesgos

Desde un principio el proyecto NeN se definió como un proyecto de desarrollo de software “parcial”, es decir, un proyecto de desarrollo en el que se realizaría todo el trabajo de documentación y diseño necesarios, pero que no sería implementado. Aunque ésto se dejó muy claro, tuvimos en cuenta la posibilidad de que dicho criterio podría cambiar en un futuro, muy posiblemente a mitad del desarrollo, por lo que tendríamos que tener en cuenta ciertos aspectos o criterios de implementación a la hora de definir los requisitos de la aplicación, su diseño, etc.

Además, el hecho de que el equipo no tuviera experiencia previa en el desarrollo de software de gestión implicaba elegir o planificar que tecnología o plataforma sería la más adecuada en el caso de que tuviera que llevarse a cabo la implementación. Por otro lado, esa elección implicaba un proceso de aprendizaje por parte de todos los

integrantes del proyecto en un tiempo muy limitado. Seguramente, el equipo tendría que ir aprendiendo “sobre la marcha”.

Por último se encuentra el problema o problemas con integrantes del grupo de desarrollo.

Como se mencionó en el punto anterior, el equipo de desarrolladores del proyecto NeN (En adelante “el equipo”) está dividido en diferentes grupos de trabajo que abarcan cada uno de los aspectos de la gestión del Departamento de Cirugía. Además, las decisiones del proyecto son tomadas de manera democrática, el equipo no cuenta con ningún líder definido que permita mediar en las disputas.

El problema de éste modelo de trabajo es que, dado que el equipo es muy reducido, los grupos de trabajo están formados por a lo sumo dos o tres desarrolladores. En el momento en el que alguno de los integrantes de un grupo falle, el desarrollo de éste se verá ralentizado o en la mayoría de los casos paralizado.

3.3 Plan de gestión del riesgo

Para solventar el problema del cambio de objetivos y posibilidad de implementación, hemos tenido en cuenta las posibilidades de implementación de cada uno de los aspectos de especificación que hemos realizado. Es decir, aunque no estuviera estrictamente dentro del ámbito del proyecto, durante el proceso de especificación de requisitos se tuvo en consideración la manera de implementar cada uno de los aspectos que íbamos tratando.

Además, dado que el equipo todavía está en fase de formación, pensamos que en el hipotético caso de que tuviera que llevarse a cabo una implementación utilizaríamos la plataforma Java, ya que es sobre la que estamos trabajando habitualmente. Barajamos otras posibilidades (.Net Framework y C#, Qt y C++, etc) pero creemos que dado el poco tiempo con el que contamos, es más sencillo que el equipo al completo se documente sobre el uso de las diferentes APIs de Java, comparado con que se tenga que aprender una plataforma completamente nueva.

También descartamos el uso de aplicaciones web, ya que aun cuando éstas tienen innumerables ventajas respecto a las aplicaciones de escritorio (Fácil acceso, mantenimiento, e instalación), el equipo tiene aún menos experiencia en éste ámbito.

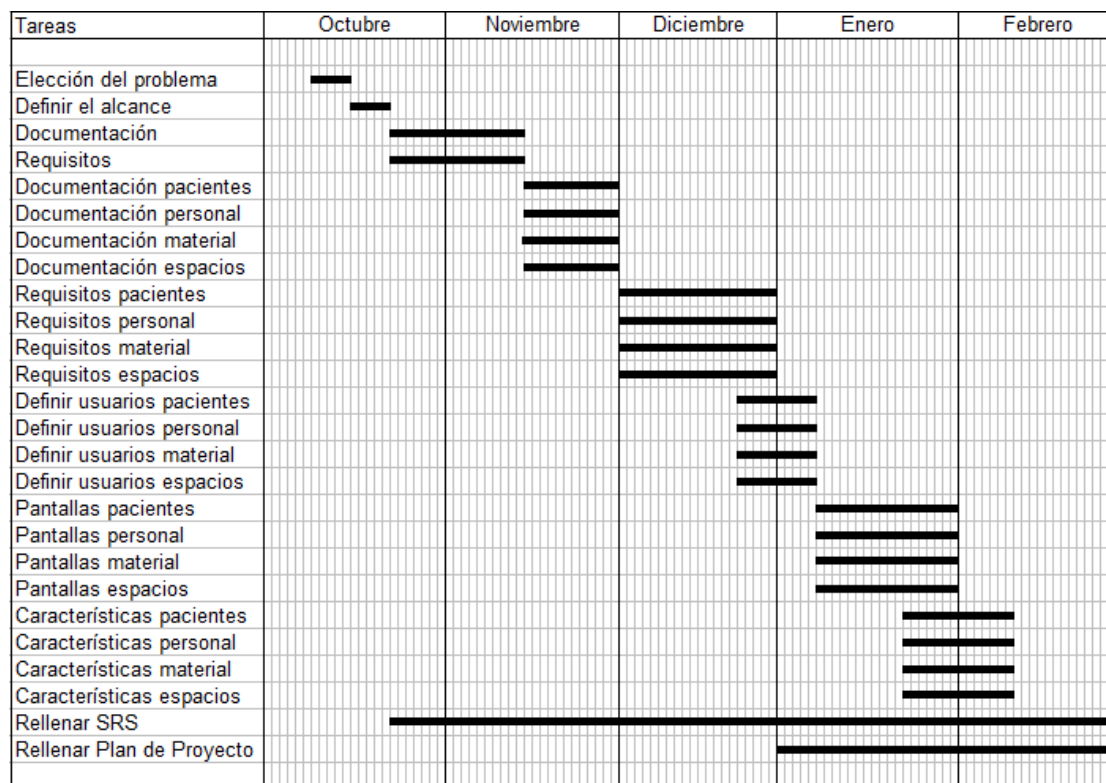
Para gestionar los problemas internos del equipo, decidimos reestructurar sobre la marcha los diferentes grupos de trabajo para compensar la falta de alguno de sus integrantes. Aunque ésto al final no ha sido necesario, ya que el grupo en concreto en el que ha ocurrido el problema por ahora se mantiene por si mismo, no descartamos la posibilidad de “equilibrar” los equipos en un futuro.

4. Planificación temporal

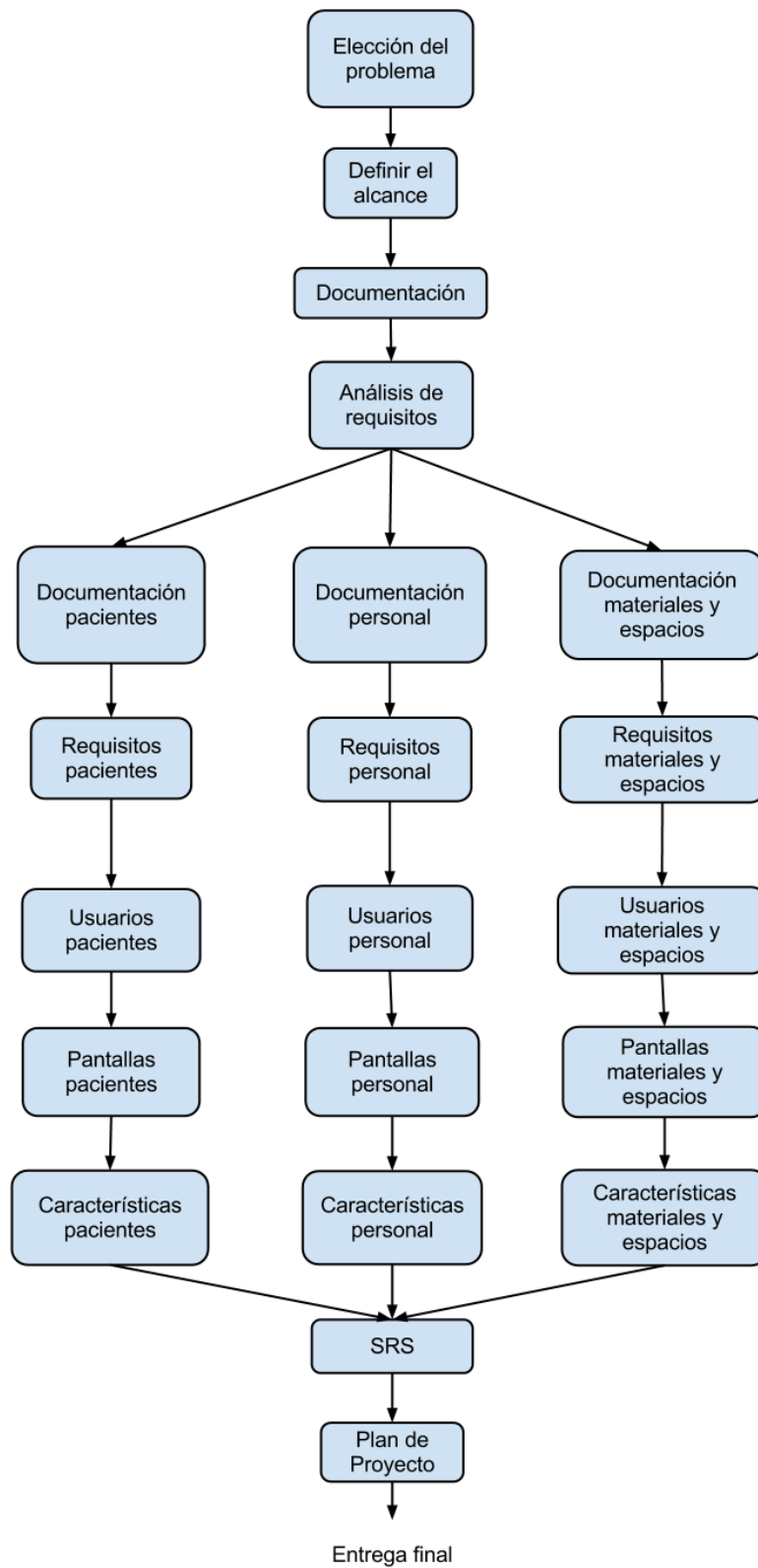
4.1 Estructura de descomposición del trabajo o Planificación temporal

AE	Comunicación			Modelado		Despliegue	
Acción	Documentación	Requisitos	Usuarios	Pantallas	Características	SRS	Plan de Proyecto
Módulo pacientes	Manuel 15.11.12 30.11.12 Doc. gestión pacientes	Manuel 01.12.12 28.12.12 Req. gestión pacientes	Manuel 24.12.12 07.01.13 Usuarios gest. pacientes	Manuel 08.01.13 31.01.13 Pantallas gest. pacientes	Manuel 23.01.13 11.02.13 Caract. gestión pacientes	Todos 23.10.12 28.02.13 Redacción del SRS	Todos 01.01.13 28.02.13 Redacción del Plan de Proyecto
Módulo personal	Laura, Mónica, Rodrigo 15.11.12 30.11.12 Doc. gestión personal	Laura, Mónica, Rodrigo 01.12.12 28.12.12 Req. gestión personal	Laura, Mónica, Rodrigo 24.12.12 07.01.13 Usuarios gest. Personal	Laura, Mónica, Rodrigo 08.01.13 31.01.13 Pantallas gest. personal	Laura, Mónica, Rodrigo 23.01.13 11.02.13 Caract. gestión personal		
Módulo materiales	Javier, Valentina 15.11.12 30.11.12 Doc. gestión materiales	Javier, Valentina 01.12.12 28.12.12 Req. gestión materiales	Javier, Valentina 24.12.12 07.01.13 Usuarios gest. materiales	Javier, Valentina 08.01.13 31.01.13 Pantallas gest. materiales	Javier, Valentina 23.01.13 11.02.13 Caract. gestión materiales		
Módulo espacios	Abel 15.11.12 30.11.12 Doc. gestión espacios	Abel 01.12.12 28.12.12 Req. gestión espacios	Abel 24.12.12 07.01.13 Usuarios gest. espacios	Abel 08.01.13 31.01.13 Pantallas gest. espacios	Abel 23.01.13 11.02.13 Caract. gestión espacios		

4.2 Gráfico Gantt



4.3 Red de tareas



4.4 Tabla de uso de recursos

Recursos		Octubre	Noviembre	Diciembre	Enero	Febrero
Personal	8					
	7					
	6					
	4					
	2					
Ordenadores	8					
	7					
	6					
	4					
	2					
Gmail						
Google Groups						
Google Drive						
Google Docs						
Balsamiq Mockups						
Github						

5. Recursos del proyecto

Para la realización del proyecto, disponemos de los recursos personales de cada miembro del equipo más los recursos que nos ofrece la universidad.

5.1 Personal

En cuanto a los recursos humanos, en un principio somos 8 los miembros del equipo que vamos a desarrollar el proyecto.

El profesor también podría considerarse parte de los recursos humanos, aunque no como desarrollador del proyecto. La figura de este guarda mayor similitud con la de un “stakeholder” del proyecto. Es decir, una persona “interesada” en el desarrollo y los resultados del proyecto. A su vez, nosotros debemos tener en cuenta los criterios que pida y las evaluaciones que nos haga el profesor de cara a que el proyecto prospere y superemos la asignatura adecuadamente.

La disponibilidad en el horario de cada miembro del equipo difiere notablemente. Con algunos miembros del grupo apenas coincidimos tan sólo con las horas de clase de Ingeniería del Software.

La necesidad de coordinarnos y superar estas diferencias, tanto de tiempo como de espacio, nos ha llevado a usar una lista de correo electrónico aprovechando las cuentas de correo Google (bajo dominio ucm.es) que nos ofrece la universidad.

Mas para la toma de decisiones y coordinación general, realizaremos una reunión de todo el equipo los viernes a mediodía, con una frecuencia aproximada de una semana.

5.2 Hardware y software

5.2.1 Hardware

Cada miembro del equipo dispone de al menos un equipo informático personal (portátil y/o de sobremesa) en perfecto funcionamiento.

También tenemos a nuestra disposición los equipos de los laboratorios de nuestra facultad, así como todos los otros servicios para el uso de ordenadores de sobremesa, préstamo de ordenadores portátiles o de iPad, etc. que ofrece la universidad.

Asimismo haremos uso de impresoras multifunción bien personales, bien de las ofrecidas por la universidad o bien de un servicio de reprografía.

Para las pruebas de la aplicación, necesitaremos un servidor de base de datos y, probablemente, también un servidor de aplicaciones, que usaremos en caso de que la aplicación se use desde una aplicación web.

Por último, la infraestructura de la red necesaria para las pruebas con los servidores.

5.2.2 Software

Disponemos de varias herramientas software específicas, cada una con un cometido particular, de las que nos valdremos para desarrollar nuestro proyecto.

- Gmail

Para la gestión de la cuenta de correo electrónico concedida a cada estudiante usaremos el concebido cliente de correo electrónico de Google: Gmail. Se trata de una aplicación web y es la que propone la universidad por defecto para el manejo de dichas cuentas.

- Google Groups

Se trata de un aplicación web para la creación y gestión de listas de correo electrónico para todo tipo de fines. Para nuestro caso crearemos una lista de este tipo específica para nuestro proyecto con las direcciones de todos los miembros del grupo.

- Google Drive

El servicio de “cloud computing” de Google. Está también incluido para las cuentas de los estudiantes de la universidad y se maneja también desde una aplicación web.

- Google Docs

Suite de programas de ofimática, en particular para la redacción de textos, presentaciones y hojas de cálculo. Viene integrado totalmente con Google Drive.

- Balsamiq Mockups

Se trata de una aplicación especialmente concebida para el diseño de “pantallas” o “casos de uso” de otras aplicaciones. Es una herramienta muy sencilla de utilizar, útil y con la que se generan rápidamente diseños.

Dispone de una extensa variedad de elementos de uso común para el diseño de pantallas de aplicaciones e incluso permite añadir *plugins* para añadir más elementos si fuese necesario.

Preferimos una aplicación de este tipo frente a otras más genéricas o al uso tradicional del papel y lápiz, por razones de mejora o modificación, reutilización y rapidez en el proceso de diseño.

Se puede usar tanto la aplicación de escritorio como la aplicación web que se integra con *Google Drive*.

■ Eclipse

Por último, para el caso que tengamos que implementar, en principio usaremos como *IDE*¹ el programa de escritorio *Eclipse*, en cualquiera de sus variantes. Se trata de una herramienta muy potente, fácil de usar y especialmente diseñada para el desarrollo de aplicaciones programadas en *Java*, aunque también soporta muchos otros lenguajes como *SQL* o *javascript*.

■ Git

Para el caso de que tengamos que codificar, usaremos git como sistema de control de versiones.

Usaremos GitHub como plataforma de desarrollo colaborativo. Este servicio nos permite alojar el código del proyecto y además incluye varias funcionalidades como páginas de discusión para el código (wiki), *bug tracking*, manejo vía web, etc.

Además de estas aplicación específicas, también necesitaremos otras, de cometido más general, necesarias para uso de las anteriores como podrían ser un navegador web o el sistema operativo.

Cualquiera de estas no tiene por qué ser un modelo específico, pero sí debe ser compatible con todas las aplicaciones antes citadas.

5.3 Lista de recursos

5.3.1 Fungibles

- Papel para impresora
- Cuadernos
- Material de oficina básico: bolígrafo, lápiz, goma, reglas
- Carpetas y portafolios

5.3.2 No fungibles

- Equipo informático y periféricos asociados.
- Impresoras
- Servidor de base de datos y servidor de aplicaciones
- Infraestructura para la red: *hub* o concentradores, cables RJ45 y tarjetas de red necesarias para los equipos.
- Programas informáticos citados en el apartado anterior.

¹ *IDE*: *Integrated Development Environment* - Entorno de Desarrollo Integrado.

6. Organización del personal

6.1 Estructura de equipo

El equipo está formado por ocho personas. El trabajo se ha dividido en tres partes: gestión de pacientes, gestión de personal y gestión de material y espacios. A cada parte se le ha asignado un grupo de personas para poder trabajar por separado quedando repartidos los miembros de la siguiente manera:

Gestión de pacientes:

- Manuel Sánchez
- ~~Giorgio García~~¹

Gestión de personal:

- Laura María de Castro
- Mónica Patricia Chachalo
- Rodrigo Arranz

Gestión de material y espacios:

- Abel Serrano
- Cristina Valentina Espinosa
- José Javier Martínez

Se reparten las personas de esta manera porque decidimos que la parte de gestión de pacientes era menos laboriosa que las demás, por lo cual se le asignó a esta parte dos personas y tres a las demás partes. El objetivo de esta repartición es que cada subgrupo vaya avanzando por su cuenta sin tener que estar pendiente de los demás subgrupos. Una vez que cada subgrupo haya terminado su parte el equipo se volverá a unir para poder interconectar las partes entre sí.

¹ Giorgio García dejó de asistir y colaborar en el proyecto por lo que al final Manuel Sánchez se ha encargado prácticamente de toda la gestión de pacientes.

6.2 Informes de gestión

A continuación se pasará a describir qué es lo que ha hecho cada subgrupo de manera un poco más detallada.

El subgrupo de gestión de pacientes se ha encargado de:

- Gestión de la lista de espera que permite tener actualizado el registro de pacientes pendientes de una intervención. La actualización de los datos deberán ser registrados en tiempo real, expresando claramente las circunstancias del suceso/cambio.
- Gestión del preoperatorio que implica la selección de los pacientes dentro de la lista de espera teniendo en cuenta su prioridad y antigüedad. Es necesario recibir información en tiempo real de los pacientes seleccionados para una operación quirúrgica
- Gestión operativa de la programación quirúrgica. Los pacientes programados constituyen un primer parte quirúrgico provisional, que con el tiempo se convertirá en definitivo. Dicho parte debe estar a disposición de cuantos servicios lo necesiten, por lo que se debe asegurar su adecuada distribución.
- Registro de intervenciones. Es necesario disponer de un registro de la actividad realizada en quirófano que permita gestionar adecuadamente los recursos del bloque quirúrgico. En el registro se refleja toda la actividad programada, además de la actividad no ejecutada, así como las intervenciones urgentes.
- Coordinación del proceso. La gestión de la actividad quirúrgica debe permitir la adecuada coordinación entre los diferentes servicios implicados, respecto de los puntos tratados anteriormente.

El subgrupo de gestión de personal se ha ocupado de organizar el personal en función de las intervenciones que haya que realizar a los pacientes. Se dividió al personal en:

- Personal médico. Son los prestadores directos del servicio a los pacientes. La responsabilidad de los médicos es directa e individual con el paciente, el médico no está formado para trabajar en función institucional, por ello rara vez toma partido en el desarrollo de la organización en la que labora.
- Personal asistencial. Son los que trabajan en colaboración directa con los médicos y los pacientes, no solamente eso sino que también son un medio de comunicación con los pacientes de la institución. La responsabilidad de este personal es directa con el paciente y complementaria, el personal asistencial está formado para trabajar en función de apoyo.
- Personal administrativo. Son los que trabajan en colaboración directa con la Dirección y La Gerencia. La responsabilidad de este personal no es directa con el paciente, el personal administrativo está formado para trabajar en función de equipo de trabajo.

En función de los pacientes y las operaciones a realizar se elabora un horario con los turnos del personal para que se puedan atender dichas operaciones lo antes posible.

El subgrupo de gestión de material y espacios se encargó de todo lo relacionado con los utensilios y dispositivos necesarios para realizar las operaciones así como de las distintas estancias por las que pasa el paciente. Esto se separa en dos grandes grupos:

- Materiales. Entendemos por esto todo el instrumental quirúrgico necesario para realizar una intervención a un paciente que se divide en instrumental general, instrumental especial y misceláneos.

- Espacios. Esto engloba todas las estancias por las que pasa un paciente desde que se le ingresa hasta que se le da el alta.

Todo esto es lo que ha hecho cada subgrupo por su cuenta.

Además de este trabajo en subgrupos se han realizado reuniones semanales, a las que acudían todos los miembros del equipo. En estas reuniones se informaba de los avances de cada subgrupo así como de los nuevos objetivos de cada subgrupo. También se elegían los encargados de hacer las presentaciones así como la persona encargada de subir al campus virtual las entregas de seguimiento requeridas por el profesor en el formato y en la fecha fijada. Las reuniones estaban organizadas por puntos a tratar que se discutían dando la palabra a todos los miembros para que cada uno diera su opinión y una vez que se llegaba a un acuerdo se pasaba al punto siguiente. En todas estas reuniones se redactaba un acta que recogía los puntos tratados y las decisiones tomadas sobre los mismos.

7. Mecanismos de seguimiento y control

7.1 Garantía de calidad y control

En un proyecto es muy importante llevar a cabo un control de la calidad del producto que estamos desarrollando y del trabajo que está produciendo. En nuestro proyecto hemos tenido en todo momento esto en cuenta. Para ello probamos diversas herramientas para llevar a cabo un control del trabajo y del producto en desarrollo hasta que encontramos una que nos aseguró un buen control y seguimiento del trabajo general e individual del proyecto.

También durante estas semanas hemos llevado a cabo reuniones semanales de aproximadamente 1 hora y media cada una, a parte de una reunión final el día 29 de enero para hacer una valoración general del proyecto, cuales son los siguientes pasos, que tenemos que mejorar, etc.

En las reuniones semanales hemos intentado controlar precisamente la calidad de nuestro proyecto. No solo repartíamos trabajo y poníamos dudas en común, sino que también hablábamos de la calidad del proyecto, de los aspectos que debíamos cambiar, o cuales creíamos que nos estaban llevando por el buen o mal camino en el desarrollo de la aplicación.

7.2 Gestión y control de cambios

Para gestionar el trabajo y tener un control sobre todos los elementos generados por los integrantes del proyecto finalmente elegimos utilizar como herramienta Google Drive.

En un principio, pensamos en utilizar la plataforma Dropbox, en la cual podíamos compartir todos los avances del proyecto y saber cuándo se habían realizado cambios. Por otra parte nos era complicado identificar que integrante del proyecto había realizado cambios y dos integrantes no podían trabajar sobre un mismo elemento al mismo tiempo.

También pensamos en utilizar la plataforma GitHub. Esta si contaba con un buen control de cambios y contaba con actualizaciones simultáneas que permiten que varios integrantes pudiesen trabajar al mismo tiempo en el mismo archivo. Pero puesto que la plataforma está pensada esencialmente para la compartición de código, no nos era útil para compartir todos los elementos del proyecto.

Por este motivo, finalmente decidimos usar Google Drive. En esta plataforma podemos compartir varios tipos de archivos: docs, pdfs, imágenes, diapositivas, tablas...incluso admite el formato de pantallas de Balsamiq Mockups, programa con el cual realizamos los prototipos de pantallas para nuestro proyecto.

Aparte de esto Google Drive cuenta con:

- Actualizaciones simultáneas, las cuales nos permiten trabajar más rápido ya que varios integrantes pueden modificar el mismo archivo al mismo tiempo, guardando de manera correcta todos los cambios.
- Notificaciones de cambios: muestra para cada documento el creador del documento y quién realizó la última modificación y cuando. También permite ver que ha sido modificado, volver a la versión anterior y ver quienes están en un momento determinado viendo o modificando un documento.
- Varias versiones del proyecto: se pueden tener tantos archivos como necesites, lo que ayuda a tener varias versiones del proyecto si es necesario. También tiene control de cambios para poder volver a una versión anterior si es necesario o para no permitir más cambios en un determinado documento.

En resumen esta herramienta nos ha permitido tener un buen control sobre el trabajo, y restringir y controlar los cambios en todo momento.

8. Apéndices