



Plan de Proyecto

Miembros del Equipo:

Adrian Agudo García-Heras

Agustín Jofré Millet

Huaibo Yang

Javier Pino Hernández

Jesús Martín

Samuel Solo de Zaldívar Barbero

Número de versión	Fecha	Autores	Descripción
1.0	1/12/16	Todos	Inicio del documento.
2.0	5/12/16	Samuel Solo de Zaldívar, Adrián Agudo	Introducción Planificación temporal
2.1	14/12/16	Javier Pino	Gestión de riesgos
2.2	18/12/16	Samuel Solo de Zaldívar	Recursos, Mecanismos de control
2.3	22/12/16	Samuel Solo de Zaldívar, Adrián Agudo	Gestión de equipos
2.31	26/12/16	Samuel Solo de Zaldívar, Adrián Agudo	Formato del documento
2.4	4/1/17	Javier Pino	Vídeo y página web
3.0	19/1/17	Samuel Solo de Zaldívar, Adrián Agudo	Mejoras del estilo de la presentación final

1 Introduccion	2 Estimaciones del proyecto	3 Gestión del riesgo	4 Planificación temporal	5 Recursos del proyecto
<div>1 Propósito del plan</div> <div>2 Ámbito del proyecto y objetivos</div> <div>2.1 Declaración del ámbito</div> <div>2.2 Funciones principales</div> <div>2.3 Aspectos de rendimiento</div> <div>2.4 Restricciones y técnicas de gestión</div> <div>3 Modelo de proceso</div>	<div>1 Técnicas de estimación</div> <div>2 Estimaciones de esfuerzo, coste y duración</div>	<div>1 Priorización de riesgos del proyecto</div> <div>2 Plan de gestión del riesgo</div> <div>Reducción, supervisión y gestión del riesgo</div> <div>3 Planificación temporal del Control de Riesgos</div> <div>4 Resumen</div>	<div>1 Estructura de descomposición del trabajo/Planificación temporal</div> <div>2 Gráfico Gantt</div>	<div>1 Personal</div> <div>2 Hardware y software</div> <div>2.1 Hardware</div> <div>2.2 Software</div> <div>3 Lista de recursos</div>

6 Gestión del Equipo

- 1 Estructura de equipo
- 2 Informes de gestión

7 Mecanismos de seguimiento y control

- 1 Garantía de calidad y control (Plan de Calidad)
- 2 Gestión y control de cambios (Plan GCS)
 - 2.1 Introducción: Propósito, Alcance, Definiciones, Referencias
 - 2.2 Tipos de artefactos a gestionar (los ECSs)
 - 2.3 Criterios y protocolos para Nombrar los ECSs
 - 2.4 Responsable de los procedimientos de GCS y de la creación de Líneas Base.
 - 2.5 Políticas para el Control de Cambios y la Gestión de Versiones
 - 2.6 Registros para mantener el rastro de los cambios

1 Proposito

Este documento detalla los requisitos software que debe cumplir un sistema de consultas y preguntas médicas, resaltando los aspectos claves del desarrollo que se muestran a lo largo del todo este documento.



Nuestra página web, aquí:

[iDoctorTeam](#)

2 Ámbito del proyecto y objetivos

2.1 Declaración del ámbito

El ámbito de este software serán los pacientes y los médicos que utilicen la aplicación.

La aplicación abarca un sistema de ayuda para los pacientes por parte de una serie de profesionales de la medicina, los cuales ofrecen sus conocimientos y experiencia para resolver todas las dudas posibles.

2.2 Funciones principales

Las funciones de la aplicación serán:

- **Historial clínico:** Cada paciente tiene un listado de todos sus antecedentes médicos, los cuales servirán de ayuda para que el médico que ayude al paciente tenga ciertas referencias a la hora de tratar sus casos.
- **Primeros auxilios:** Muestra un listado con diversos ejemplos (los más comunes) donde hay que dar primeros auxilios y a su vez cómo tratarlos en caso de que cierta situación lo requiera.
- **Diagnósticos:** Para la búsqueda de un diagnóstico, el paciente introduce una serie de datos a unas preguntas que le hace el sistema con el fin de ir descartando y asegurando un diagnóstico claro.

2.3 Aspectos de rendimientos

Debido a que estamos creando un software simple, las variaciones del rendimiento vendrían marcadas principalmente por las características hardware del dispositivo o la velocidad y procesamiento de acceso a registros almacenados en la base de datos.

2.4 Restricciones y técnicas de gestión

El proyecto se desarrolla entre siete componentes, los cuáles nos distribuimos el trabajo para tener las diferentes partes terminadas en el tiempo que nos habíamos propuesto. El lenguaje de programación que vamos a emplear para el desarrollo de la aplicación será Java.

3 Modelo de proceso

Proceso unificado de desarrollo, que se caracteriza por el uso casos de uso, enfoque a los distintos riesgos, naturaleza iterativa e incremental y centrado en la arquitectura. Consta de 4 fases: inicio, elaboración, construcción y transición.

1 Técnicas de estimación

En este proyecto nos decantamos por la técnica de descomposición basada en el proceso debido a que no contamos con datos históricos por la inexperiencia en este tipo de proyectos y porque se espera que esta técnica nos proporcione resultados más realistas.

2 Estimaciones de esfuerzo, coste y duración

El proyecto se inicia el día 7 de noviembre de 2016 y finaliza el 2 de junio de 2017. La jornada laboral se establece en 1 día/semana. Respecto el coste del proyecto, será de esfuerzo y no de ámbito económico.

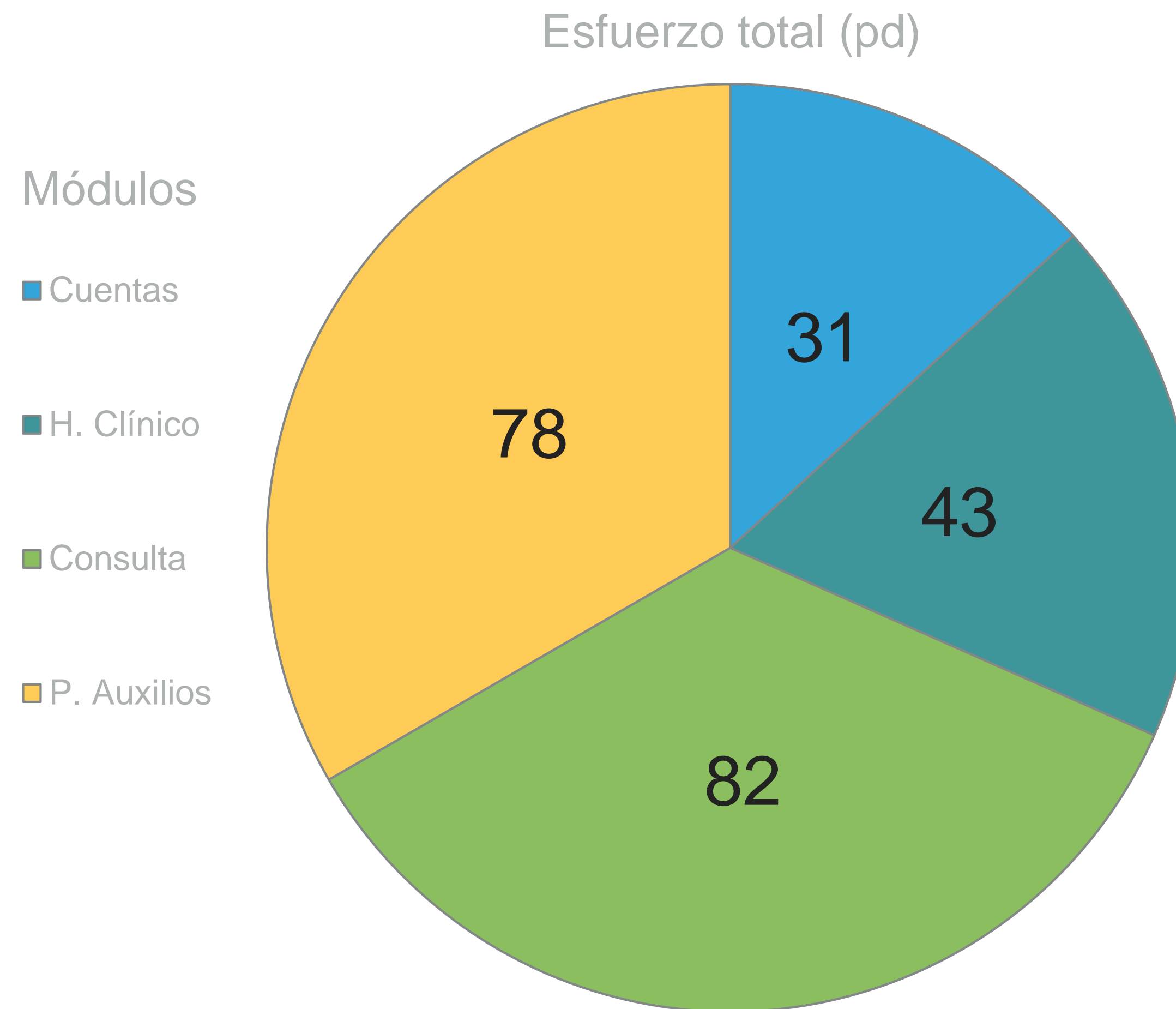
A continuación, se expone la estimación del esfuerzo realizada:

- **Especificación de requisitos** – 35 días con 3 personas involucradas y un esfuerzo total de 105 pd.
- **Plan de proyecto** – 56 días con 3 personas involucradas y un esfuerzo total de 168 pd.
- **Evaluación cliente** – 20 días con 6 personas involucradas y un esfuerzo total de 120 pd.

2 Estimaciones de esfuerzo, coste y duración

Módulo	Fase	Personas	Días	Personas * Días	Esfuerzo Total
Cuentas	Análisis	1	4	4	31
	Diseño	2	7	14	
	Codificación	2	5	10	
	Pruebas	1	3	3	
Historial Clínico	Análisis	2	8	16	43
	Diseño	1	11	11	
	Codificación	2	6	12	
	Pruebas	1	4	4	
Consulta	Análisis	2	11	22	82
	Diseño	3	10	30	
	Codificación	3	8	24	
	Pruebas	1	6	6	
Primeros Auxilios	Análisis	3	8	24	78
	Diseño	3	12	36	
	Codificación	2	8	16	
	Pruebas	2	6	12	

2 Estimaciones de esfuerzo, coste y duración



1 Priorización de riesgos del proyecto

Nº	ER	Riesgo	P	C
1	0.3	Tiempo requerido para desarrollar el software está subestimado	Alta	Serio
2	0.2	Fallos de implementación	Alta	Tolerable
3	0.03	Se proponen cambios en los requerimientos que requieren rehacer el diseño	Media	Serio
4	0.02	Falta de tiempo para puesta en Producción y pruebas	Media	Tolerable
5	0.02	Fallos técnicos e indisponibilidad del sistema	Media	Tolerable
6	0.02	El tamaño del SW está subestimado	Media	Tolerable
7	0.004	Pérdida de información	Baja	Catastrófico
8	0.003	Capacitación del personal	Baja	Serio
9	0.003	Problemas inherentes a la Base de Datos	Baja	Serio
10	0.002	Abandono de miembros del equipo	Baja	Tolerable

2 Reducción, supervisión y gestión del riesgo

2.0.1 Reducción

Ayudarnos de herramientas para la planificación temporal del proyecto tales como diagramas GANTT. Conocer bien los recursos con que contamos y su disponibilidad tanto presentes como futuras.

2.0.2 Supervisión

Controlar los plazos de entrega y fomentar la comunicación entre los integrantes del grupo a fin de evitar duplicidades o dejar temas sin abordar por no tener persona asignada.

2.0.3 Plan de Contingencia

Redimensionar la carga de trabajo de manera equitativa entre los integrantes con el fin de cumplir los plazos estimados.
Reconfigurar parte del proyecto para poder simplificar o modificar o eliminar algún aspecto del mismo.

2.1 Fallos de implementación

2.1.1 Reducción

Al comienzo de cada módulo se releen con atención los requisitos para asegurarnos de que la programación se ajuste a ellos.

Aumentar la frecuencia de las pruebas y revisiones disminuirá el posible impacto que pueda tener un fallo sobre el resto del proyecto.

2.1.2 Supervisión

Diseñar una batería de pruebas para poder medir el correcto funcionamiento del sistema.

2.1.3 Plan de Contingencia

La persona o personas encargadas se disponen a solucionarlo de inmediato, en especial si se tratara de una parte crítica del sistema o hubiese otros módulos que dependen de la parte errónea.

2.2 Cambios en los requerimientos

2.2.1 Reducción

En la medida de lo posible se debe tener a priori una “visibilidad” del proyecto lo más cerrada posible, así como su alcance y posibles vulnerabilidades que minimicen la ocurrencia de nuevas funcionalidades una vez avanzado ya el proyecto

2.2.2 Supervisión

En cada etapa repasar si el desarrollo realizado se ciñe a los requisitos especificados.

Establecer reuniones periódicas entre todos los integrantes que den una idea general de la marcha del proyecto.

2.2.3 Plan de Contingencia

Se analiza cuál de los integrantes del grupo puede dedicarse en función de su experiencia y cercanía a las nuevas funcionalidades que se proponen, a la implementación de los nuevos requisitos.

2.3 Falta de tiempo para producción y pruebas

2.3.1 Reducción

Hay que tener muy en cuenta la necesidad de realizar pruebas sobre de funcionamiento. Tanto a nivel funcionales, como de disponibilidad, tiempos de respuesta, pruebas de estrés a la BBDD, etc.

Incluir dichas pruebas en la estimación de tiempos del proyecto y sobredimensionada en una o dos jornadas.

2.3.2 Supervisión

Establecer pruebas no sólo al final del proyecto, sino durante el mismo.

2.3.3 Plan de Contingencia

Establecer un sistema de “Gestión de Configuración del Software” y dentro de éste “Gestión del Cambio” que permita de una manera dinámica y fluida comunicar una eventual incidencia.

2.4 Fallos técnicos e indisponibilidad del sistema

2.4.1 Reducción

Prestar atención a las indicaciones de esta asignatura y, caso de ser necesario, informarse sobre otras cosas que vayan a ser necesarias para la realización del proyecto (como bases de datos). También favorable trabajar en grupos de más de una persona, para disminuir la probabilidad de que se cometa un fallo. Aumentando la frecuencia de las revisiones podremos detectar antes los fallos para que afecte lo menos posible al resto del proyecto.

2.4.2 Supervisión

Debemos dedicar suficiente tiempo a la prueba y a la revisión, preferiblemente por distintos componentes del grupo a los que lo programaron.

2.4.3 Plan de Contingencia

Una vez que se ha detectado el fallo hay que solucionarlo inmediatamente, especialmente si hay otros módulos que dependan de él, para minimizar el impacto.

2.5 El tamaño del SW está subestimado

2.5.1 Reducción

Realizar a priori un análisis realista sobre el sistema que se quiere desarrollar huyendo de idealizaciones y tratando de evitar nuevas funcionalidades que no hayan sido previamente incluidas.

En este sentido se puede establecer que se trate de una primera versión de un proyecto al que puedan incorporarse posteriormente nuevas versiones o módulos añadidos.

2.5.2 Supervisión

Análisis del factor transcurso del proyecto / tiempo para evaluar cualquier desfase entre el análisis de la aplicación y el costo de su programación.

Se deben realizar reuniones oportunas para evaluar dicho aspecto

2.5.3 Plan de Contingencia

Si se percata un desfase en cuanto a la carga de trabajo de desarrollo de software y los recursos tanto humanos como temporales dedicados a esta carga se deberán redistribuir las tareas asignadas inicialmente y establecer un sistema de prioridades en las tareas.

2.6 Pérdida de información

2.6.1 Reducción

Este factor de riesgo aun siendo uno de los de mayor criticidad se puede controlar con unas medidas básicas de prevención, lo que baja drásticamente la probabilidad de que se produzca.

Con unas medidas básicas de copias de seguridad y buenos sistemas que garanticen el trabajo en grupo de manera concurrente no debería haber mayor problema.

2.6.2 Supervisión

Este riesgo se auto-supervisa por todos los elementos del grupo a lo largo de la construcción del proyecto.

2.6.3 Plan de Contingencia

Reestablecer la información perdida mediante las oportunas operaciones de “Restore”

2.7 Capacitación del personal

2.7.1 Reducción

Los integrantes del grupo son personas muy implicadas con el proyecto, dedicando buena parte de su tiempo al mismo.

Si bien se aprecian deficiencias formativas, se suplen con los apoyos del resto de los compañeros.

2.7.2 Supervisión

En este sentido se hace indispensable la buena comunicación en el grupo, poniendo en común las dificultades encontradas y buscando de manera conjunta las soluciones.

También se dispone de las herramientas informáticas oportunas para compartir información, que en nuestro caso son utilidades del estilo de Google Drive y Google Docs, etc.

2.7.3 Plan de Contingencia

Cualquier deficiencia por parte de algún miembro del grupo deberá ser adquirida por otro miembro, y tras ello, ponerla en común para el conocimiento de todos.

2.8 Problemas inherentes a la Base de Datos

2.8.1 Reducción

Se hace necesario realizar y evaluar un correcto análisis del Modelo de Datos, que cubra todos los aspectos del aplicativo. Se trata de establecer los correctos parámetros de tamaño de almacenamiento y tiempos de respuesta aceptables

2.8.2 Supervisión

El Modelo de Datos debe ser una tarea que debe quedar bastante “cerrada” en las primeras fases de Desarrollo del sistema.

Se debe someter a estudio por parte de todos los miembros del grupo, ya que previsiblemente, la BBDD afectará a todo el sistema.

2.8.3 Plan de Contingencia

Revisión de la Base de Datos y puesta a punto de nuevas modificaciones.

2.9 Abandono de miembros del equipo

2.9.1 Reducción

Este riesgo se autorregula por el propio interés de los integrantes del grupo en el proyecto. Si bien, no se está ni mucho menos exento de sufrir una baja en el equipo.

2.9.2 Supervisión

Comunicación entre los miembros del equipo.

2.9.3 Plan de Contingencia

Redistribución de las funciones asignadas a la persona saliente entre el resto de integrantes.

4 Planificación temporal del Control de Riesgos

Lo ideal sería adoptar una estrategia proactiva, es decir, prevenir los riesgos antes de que ocurran, pero no siempre es posible, adoptando una estrategia reactiva, de reacción al riesgo ya acaecido.

5 Resumen

La Gestión de los Riesgos se revela como una parte esencial del Plan de Proyecto y se erige como una herramienta útil y poderosa, que, sin ella, estaríamos expuestos a situaciones a veces dramáticas que podrían poner en riesgo la viabilidad de todo el proyecto.

Aun así, es de reconocer, que a veces puede ser complicada esta gestión, debido, en gran parte, al carácter puramente aleatorio de algunos riesgos. Pero en otros muchos casos una buena planificación de la gestión de los riesgos puede ayudar a que si estos se produjeran su impacto sea menor o incluso nulo.

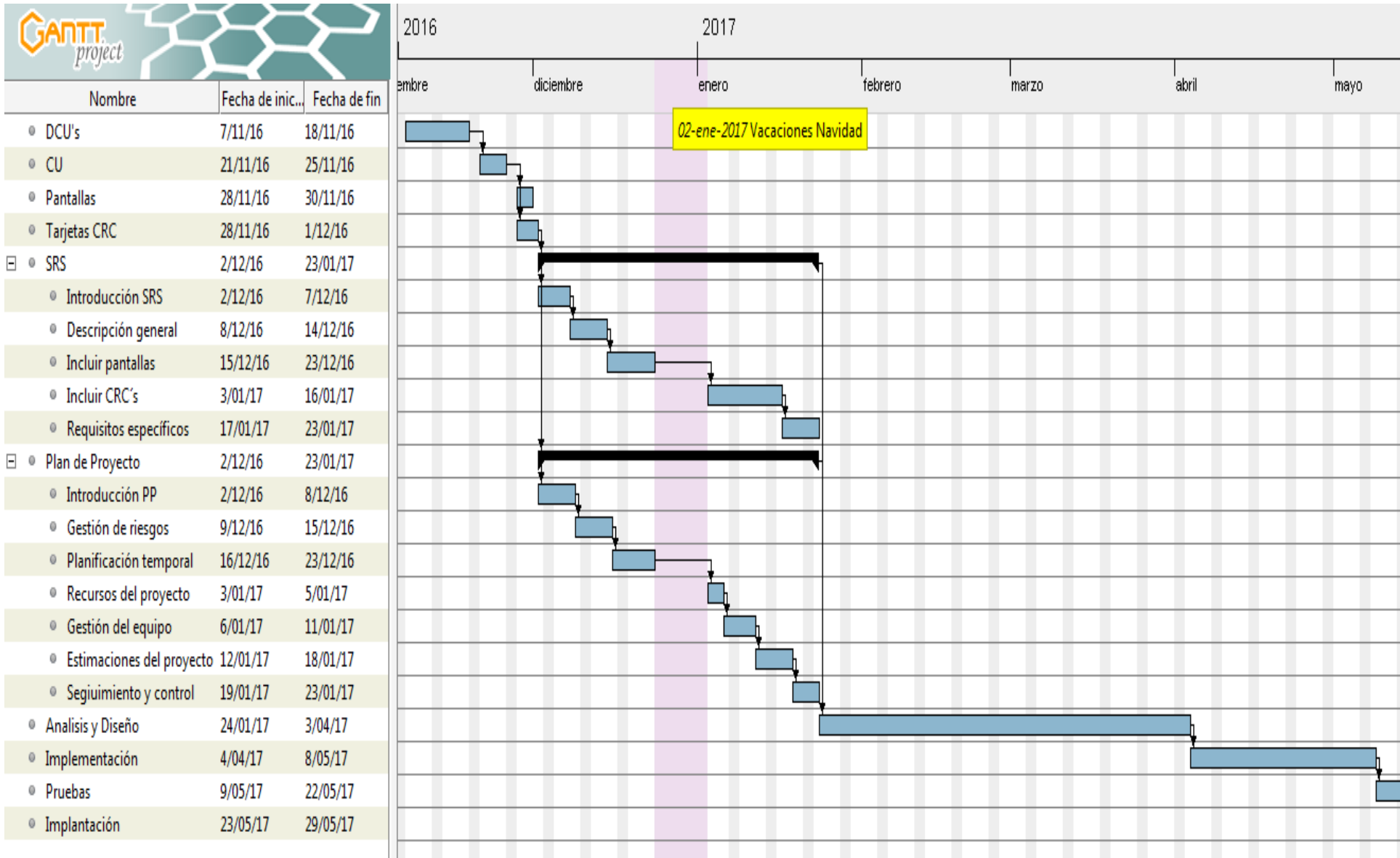
1 Estructura de descomposición del trabajo/Planificación temporal

Nuestro proyecto se va a elaborar mediante un proceso unificado de desarrollo, es por eso que vamos a dividir el proyecto en subsistemas lo que nos va a permitir trabajar en varias partes a la vez.

La organización del proyecto será:

- Análisis: recogemos la información, realizamos los requisitos y definimos los usuarios.
- Diseño: realizamos un diseño del entorno gráfico y definimos los estilos que va a tener el programa.
- Documentación: recopilamos toda la información en el SRS que nos servirá para realizar el diseño del programa.
- Construcción: realizamos la codificación del programa, las pruebas necesarias y el ensamblaje de todas las partes del proyecto.

2 Gráfico Gantt



Pulse en la imagen para acceder al fichero

1 Personal

El grupo iDoctor Team consta de 6 miembros, los cuáles serán repartidos de la siguiente manera:

- 2 revisores de CUs: Su función se limita a asegurar que los distintos casos de uso son coherentes y prácticos.
- 2 diseñadores gráficos: Sencargan de hacer una interfaz amigable y fácil de entender. A su vez se encargarán de la estructura de la programación.
- 2 programadores.

2 Hardware y software

2.1 Hardware

El hardware empleado para desarrollar la aplicación será sencillamente de ordenadores de gama media de sobremesa.

2.2 Software

EL software utilizado para el desarrollo de la aplicación será:

- Eclipse: Es una plataforma de software compuesto por un conjunto de herramientas de programación en Java y C++, compatible con SQL y elementos Web.
- GitHub: Herramienta de control de versiones.
- Adobe XD: Desarrollo gráfico de la aplicación.

3 Lista de recursos

3.1 Personal

- Samuel Solo de Zaldívar Barbero
- Jesús Martín
- Adrián Agudo García-Heras
- Javier Pino Hernández
- Agustín Jofré Millet
- Huaibo Yang
- Paciente
- Médico

3.2 Hardware

- Ordenadores de sobremesa

3.3 Software

- Entorno de desarrollo Eclipse
- GitHub
- Adobe XD
- Gantt Project
- Microsoft Word

1 Estructura de equipo

Para el desarrollo del proyecto nos hemos organizado de manera descentralizada democrática (DD) puesto que no tenemos un jefe designado en concreto, cada miembro del grupo asume dicho papel en cada una de las partes a entregar del trabajo.

La comunicación y las decisiones se toman en equipo en reuniones ocasionales así como por distintos medios de comunicación a través de Internet.

2 Informes de gestión

Con la estructura empleada no tenemos competencias o responsabilidades permanentes puesto que cada una de las partes las redistribuimos. En el caso concreto de el Plan de Proyecto, la asignación está descrita de la siguiente manera:

Samuel Solo de Zaldívar Barbero

- Introducción
- Organización del personal (Gestión del Equipo)
- Recursos del proyecto
- Mecanismos de seguimiento y control

Javier Pino Hernández

- Introducción
- Estrategia de gestión del riesgo

Adrián Agudo García-Heras

- Organización del personal (Gestión del Equipo)
- Recursos del proyecto

1 Garantía de calidad y control (Plan de Calidad)

La herramienta de versionado que utilizaremos será Git y el servidor GitHub.

Para conseguir crear software de calidad realizaremos revisiones técnicas formales luego de cada tarea finalizada.

Se designará a alguien que revise el trabajo hecho por los otros integrantes del grupo y así poder encontrar errores o simples modificaciones para que tenga más sentido la tarea.

A partir de aquí, en la siguiente reunión se tendrán que exponer las modificaciones realizadas y sus consecuencias.

2 Gestión y control de cambios (Plan GCS)

2.1 Introducción: Propósito, Alcance, Definiciones, Referencias

Gestionamos el cambio en los artefactos a lo largo del ciclo de vida del producto.

Las tareas claves son:

- Identificar y controlar el cambio
- Garantizar la correcta implementación del cambio
- Informar del cambio a todos aquellos que lo necesiten

Garantizamos que en todo momento se controla las copias, los cambios y versiones pasadas, actuales y futuras del proyecto

Las actividades de la GCS son:

- Identificar ECS
- Control versiones
- Gestión del cambio
- Auditoria de configuración
- Informes de estado

2.2 Tipos de artefactos a gestionar (los ECSs)

- La SRS o documento de Especificación de Requisitos Software
- El plan del proyecto
- El diseño
- El código
- Los casos de prueba
- Estándares y procedimientos de la Ingeniería del Software
- El editor y el compilador

2.3 Criterios y protocolos para Nombrar los ECSs

El criterio que se va a seguir para nombrar las versiones de cada elemento de la configuración es el siguiente:

vPrincipal.Secundario

Donde principal significa cambios notables y grandes dentro del elemento y Secundario los cambios pequeños.

2.4 Responsable de los procedimientos de GCS y de la creación de Líneas Base

Las actividades las haremos de manera individual, y si en algún momento alguien ve necesario realizar un cambio o cierta duda que afecte al cómputo global, se discutirá ésta hasta llegar a un acuerdo.

Si ésta es aceptada, se seguirán los pasos para realizar dicho cambio.

2.5 Políticas para el Control de Cambios y la Gestión de Versiones

Protocolo para llegar a un acuerdo y los pasos a dar en caso de realizar un cambio: Si es necesario realizar el cambio y cómo afecta éste al resto del proyecto, discutiendo las posibilidades presentadas como respuesta a dicha propuesta.

Emplearemos la herramienta Git y un servidor Git.

2.6 Registros para mantener el rastro de los cambios

La herramienta Git nos ofrece un historial de todos los cambios, así como una breve descripción, realizados en cada versión. Así mismo los datos de la persona que ha realizado dicho cambio, fecha y archivo.