**Proyecto: iDoctor**

**Plan del proyecto del software**

**(Pressman)**

**Miembros del equipo:**

|  |  |
| --- | --- |
| Samuel Solo de Zaldívar Barbero |  |
| Jesús Martín |  |
| Adrián Agudo García-Heras |  |
| Javier Pino Hernández |  |
| Agustín Jofré Millet |  |
| María Dolores Quilarte |  |
| Huaibo Yang |  |

**Control de cambios**

|  |  |  |  |
| --- | --- | --- | --- |
| **Número de versión** | **Fecha** | **Autores** | **Descripción** |
| Versión 1 | 1/12/16 | iDoctor team | Inicio del documento. |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

# 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 Introducción: Estudio de los riesgos

3.2 Priorización de riesgos del proyecto

3.3 Plan de gestión del riesgo Reducción, supervisión y gestión del riesgo

3.4 Planificación temporal del Control de Riesgos

3.5 Resumen

4. Planificación temporal

4.1 Estructura de descomposición del trabajo/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

6. Organización del personal (Gestión del Equipo)

6.1 Estructura de equipo (si procede)

6.2 Informes de gestión

7. Mecanismos de seguimiento y control

7.1 Garantía de calidad y control (Plan de Calidad)

7.2 Gestión y control de cambios (Plan GCS)

7.2.1 Introducción: Propósito, Alcance, Definiciones, Referencias

7.2.2 Tipos de artefactos a gestionar (los ECSs)

7.2.3 Criterios y protocolos para Nombrar los ECSs

7.2.4 Responsable de los procedimientos de GCS y de la creación de Líneas Base.

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

7.2.6 Registros para mantener el rastro de los cambios

8. Apéndices

1. **Introducción**
   1. **Propósito del plan**

El objetivo general del proyecto iDoctor consiste en realizar una aplicación basada en consultas médicas.  Más concretamente, el proyecto va destinado a cualquier persona (potencial paciente o profesional sanitario). En primer lugar, cualquier persona podrá consultar información sobre primeros auxilios, su historial clínico y realizar cualquier pregunta a algún médico. En segundo lugar, un sanitario podrá responder a las preguntas realizadas por los demás usuarios del sistema, pero además podrá editar y agregar patologías que crea conveniente. En ambos casos, tanto el médico como el paciente se tendrán que dar de alta para realizar estos procesos, aunque con diferentes tipos de cuentas, ya que para entrar como personal sanitarios se le pedirá su número de colegiado a la hora de registrarse.

* 1. **Ámbito del proyecto y objetivos**
     1. **Declaración del ámbito**

El ámbito de este software serían 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.

* + 1. **Funciones principales**

Las funciones de la aplicación serán:

-Historial clínico: Cada paciente tiene un listado de todas 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.

* + 1. **Aspectos de rendimiento**
    2. **Restricciones y técnicas de gestión**
  1. **Modelo de proceso**

Proceso unificado de desarrollo.

1. **Estimaciones del proyecto**
   1. **Datos históricos**
   2. **Técnicas de estimación**
   3. **Estimaciones de esfuerzo, coste y duración**
2. **Estrategia de gestión del riesgo**
   1. **Introducción: Estudio de los riesgos**
   2. **Priorización de riesgos del proyecto**
   3. **Plan de gestión del riesgo Reducción, supervisión y gestión del riesgo**

PARA CADA RIESGO K importante según su gravedad:

3.k. Riesgo k-ésimo

3.k.1. Reducción: cómo evitar que suceda

3.k.2. Supervisión: controlar riesgo y supervisar los pasos de reducción

3.k.3. Plan de Contingencia: Gestión si el Riesgo sucede

* 1. **Planificación temporal del Control de Riesgos**

(cuándo hacer las tareas de control de riesgos)

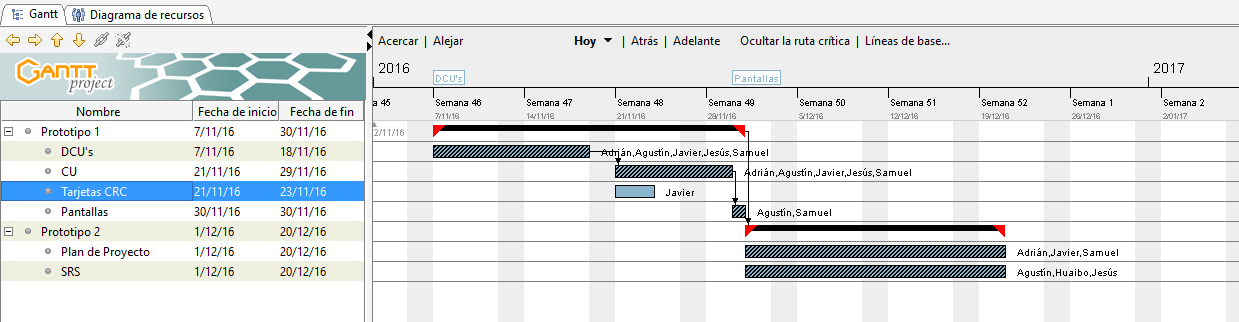
* 1. **Resumen**

1. **Planificación temporal**
   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.
  1. **Gráfico Gantt**



* 1. **Red de tareas**
  2. **Tabla de uso de recursos**

1. **Recursos del proyecto**
   1. **Personal**

El grupo iDoctor Team consta de 7 miembros, los cuáles serán repartidos de la siguiente manera:

* 2 revisores de CUs, y su función se limita a asegurar que los distintos casos de uso son coherentes y prácticos.
* 2 diseñadores gráficos, que se encargan de hacer una interfaz amigable y fácil de entender. A su vez se encargarán de la estructura de la programación.
* 3 programadores.
  1. **Hardware y software**

5.2.1 **Hardware**

El hardware empleado para desarrollar la aplicación será sencillamente ordenadores de sobremesa-

5.2.2 **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.
* Balsamiq: Desarrollo gráfico de la aplicación.
  1. **Lista de recursos**

1. **Organización del personal (Gestión del Equipo)**
   1. **Estructura de equipo (si procede)**
   2. **Informes de gestión**
2. **Mecanismos de seguimiento y control**

La finalidad básica del seguimiento es la observación de la persona u objeto que se controla, creando así sistemas de control en el que es fundamental especificar qué cambio se ha realizado en dicha persona u objeto.

* 1. **Garantía de calidad y control (Plan de Calidad)**
  2. **Gestión y control de cambios (Plan GCS)**
     1. **Introducción: Propósito, Alcance, Definiciones, Referencias**
     2. **Tipos de artefactos a gestionar (los ECSs)**
     3. **Criterios y protocolos para Nombrar los ECSs**
     4. **Responsable de los procedimientos de GCS y de la creación de Líneas Base.**
     5. **Políticas para el Control de Cambios y la Gestión de Versiones**

- Estándares (Plantillas que usas en documentos; Pasos a dar en un cambio; Acuerdos en productos) - Comité de Cambios: Quién hace qué - Herramientas de gestión de versiones – Repositorios - Gestión de Proyectos (si la usas)

* + 1. **Registros para mantener el rastro de los cambios**

(Qué ha pasado, pasos para usar herramientas del punto 7.2.5 )

1. **Apéndices**