**Proyecto: RESGES**

**Plan del proyecto del software**

**(Formato Resumido Pressman)**



**Miembros del equipo:**

| Andrés Enrique García de Abroleya Ortiz | Alfonso Barriga Lucena |
| --- | --- |
| Eduardo Regidor Serrano | Pablo Sánchez Redondo |
| Sergio Cuenca López | Jose Antonio Almendros Corrales |
| Joel Valencia Lorente |  |

**Control de cambios**

| **Número de versión** | **Fecha** | **Autores** | **Descripción** |
| --- | --- | --- | --- |
| Versión 1 | 13/11/23 | Andrés Enrique García de Arboleya | Importación del archivo .docx |
| Versión 2 | 8/12/23 | Andrés Enrique García de Arboleya | Inicio punto 1 |
| Versión 3 | 14/12/23 | Joel Valencia Lorente y Jose Antonio Almendros Corrales | gestión de riesgos , punto 5 |
| Versión 4 | 15/12/23 | Joel Valencia Lorente y Pablo Sanchez Redondo | Trabajo del 7.2 y del punto 2 estimaciones del proyecto |
| Versión 5 | 17/12/23 | Pablo Sanchez Redondo , Eduardo Regidor Serrano y Andrés García de Arboleya | Desarrollo de la parte estetica del PDP, Avance con el punto 1 y con el punto 7 |
| Versión 6 | 18/12/23 | Andrés García de Arboleya | Desarrollo de la estimación, perfeccionar el punto 1 |
| Versión 7 | 19/12/23 | Andrés García de Arboleya , Eduardo Regidor Serrano,  Joel Valencia Lorente | Perfeccionamiento del proyecto y homogeneización estética |
| Versión 8 | 19/12/23 | Jose Antonio Almendros Corrales | Punto 4 |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

# Índice

1. Introducción 1

1.1 Propósito del plan 1

1.2 Ámbito del proyecto y objetivos 1

1.2.1 Declaración del ámbito 1

1.2.2 Funciones principales 1

1.2.3 Aspectos de rendimiento 1

1.2.4 Restricciones y técnicas de gestión 1

1.3 Modelo de proceso 1

2. Estimaciones del proyecto 2

2.1 Datos históricos 2

2.2 Técnicas de estimación 2

2.3 Estimaciones de esfuerzo, coste y duración 2

3. Estrategia de gestión del riesgo 3

3.1 Introducción: Estudio de los riesgos 3

3.2 Priorización de riesgos del proyecto 3

3.3 Plan de gestión del riesgo Reducción, supervisión y gestión del riesgo 3

3.4 Planificación temporal del Control de Riesgos 3

3.5 Resumen 3

4. Planificación temporal (se adjunta el fuente del MS Project) 4

4.1 Estructura de descomposición del trabajo/Planificación temporal y D. de Gantt 4

4.2 Red de tareas 4

4.3 Tabla de uso de recursos 4

5. Recursos del proyecto 5

5.1 Personal 5

5.2 Hardware y software 5

5.3 Lista de recursos 5

6. Organización del personal (Gestión del Equipo) 6

6.1 Estructura de equipo (si procede) 6

6.2 Informes de gestión 6

7. Mecanismos de seguimiento y control 7

7.1 Garantía de calidad y control (Plan de Calidad) [7](#_heading=h.2p2csry)

7.2 Gestión y control de cambios (Plan GCS) 7

7.2.1 Introducción: Propósito, Alcance, Definiciones, Referencias 7

7.2.2 Tipos de artefactos a gestionar (los ECSs) 7

7.2.3 Criterios y protocolos para Nombrar los ECSs 7

7.2.4 Responsable de los procedimientos de GCS y de la creación de Líneas Base. 7

7.2.5 Políticas para el Control de Cambios y la Gestión de Versiones 7

7.2.6 Registros para mantener el rastro de los cambios 7

**1. Introducción**

En este apartado expondremos el plan de proyecto de ResGes , hablando de su propósito , del ámbito de nuestro software y objetivos y el modelo de proceso que vamos a seguir para desarrollarlo.

* 1. **Propósito del plan**

El plan del proyecto es que a partir de una buena gestión entre los que serán controlar los posibles riesgos y su gestión, garantías de calidad, proceso de gestión de cambios y costes y a su vez tener una guía que nos sirva para poder crear nuestro software ResGes cuya finalidad es ayudar a los trabajadores de la propia residencia en sus labores .

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

El ámbito del proyecto consistirá en función del contexto, los objetivos, la forma de integrarse y sus limitaciones, que establecerán el desarrollo de sus funciones donde se tendrán en cuenta su rendimiento, sus entradas y salidas.

* + 1. **Declaración del ámbito**

El ámbito del proyecto ResGes abarca el diseño, desarrollo e implementación de un software integral para la gestión de residencias de ancianos. Se centrará en la administración eficiente de información crítica, incluyendo datos de clientes, cuidados, medicación y otros aspectos relevantes para facilitar las operaciones diarias en las residencias.

* + 1. **Funciones principales**

Las funciones principales de “ResGes” permitirán gestionar adecuadamente una residencia e incluyen :

-Gestión de actividades: ( Añadir actividad,Eliminar actividad,Consulta de actividades,Modificación de actividad);

-Gestión de medicamentos: (Alta de medicamento,Actualización de medicamentos,Baja de medicamentos,Visualización de medicamentos);

-Gestión de los profesionales del cuidado: (Visualización de horas de atención,Asignar horas de atención,Eliminar horas de atención,Visualizado de necesidades);

-Gestión de residentes: (Dar de alta,Dar de baja,Modificar datos,Mostrar datos);

-Gestión de estancia: (Alta de estancia,Baja de estancia,Cambio de estancia,Cambio de duración de estancia);

-Gestión de las patologías de los residentes: (Recopilación de Datos de Patologías,Ingreso de Datos del cliente,Análisis de coincidencias,Registro de resultados);

-Gestión de pagos:(Añadir deuda,Eliminar deuda,Aumentar deuda,Disminuir deuda);

-Gestión de patologías:(Añadir patología,Eliminar patología);

El software se diseñará para mejorar la eficiencia del personal y garantizar un mejor cuidado y bienestar de los residentes.

* + 1. **Aspectos de rendimiento**

“ResGes” se enfocará en aspectos de rendimiento como la accesibilidad, la eficiencia en el procesamiento de datos, la seguridad de la información y la facilidad de uso. Se buscará un rendimiento óptimo para garantizar una experiencia sin contratiempos tanto para el personal como para los usuarios finales.

* + 1. **Restricciones y técnicas de gestión**

Las restricciones identificadas incluyen: el lenguaje con el que vamos a realizar esta aplicación (Java, actualmente la estamos estudiando) presentará unas restricciones propias del lenguaje, el presupuesto asignado y el tiempo limitado para la implementación. Para gestionar estas restricciones en primer lugar será desarrollar ResGes con Java como lenguaje principal, se aplicarán técnicas de gestión ágil, permitiendo entregas incrementales y ajustes iterativos en función de los comentarios y necesidades emergentes. Al ser un grupo de 7 personas y necesitar hacer el proyecto en poco tiempo, decidimos utilizar una metodología incremental,en concreto el proceso unificado de desarrollo, con eso, con nuestra poca experiencia y con la independencia de la mayoría de nuestros módulos, creemos que el tipo de equipo más acorde es el descentralizado controlado, debido a que facilita la comunicación horizontal y vertical, porque resulta también acorde al tamaño del equipo y porque, reitero, debido a la poca experiencia del equipo, puede sernos útil para retroalimentarnos.

* 1. **Modelo de proceso**

El modelo de proceso seleccionado para el desarrollo de ResGes es el modelo incremental, específicamente el proceso unificado de desarrollo. Esto permitirá flexibilidad, adaptabilidad a cambios y una entrega rápida de funcionalidades clave.

1. **Estimaciones del proyecto**
   1. **Datos históricos**

Al ser un equipo de trabajo nuevo y siendo este el primer proyecto carecemos de dichos datos, pero podemos asumir que nuestros datos de productividad se asemejan en cierta manera a la media realizada con la investigación sobre datos históricos de proyectos similares en el campo de desarrollo de software llevados por equipos con una experiencia similar a la nuestra.

* 1. **Técnicas de estimación**

Por la descripción del anterior punto se puede deducir que la técnica de estimación que usaremos será la estimación por descomposición, en las cuales estimamos el coste en base a la descomposición del proceso y del producto.

Debido a nuestra poca experiencia utilizaremos una técnica de descomposición basada en el problema, basada en la descomposición del producto en funciones y estimar el tamaño del software.

Dispondremos de técnicas que harán uso de la LDC(líneas de código),el tamaño del software lo consideraremos en LDC.

* 1. **Estimaciones de esfuerzo, coste y duración**

En este apartado procederemos a realizar la estimación del proyecto siguiendo la técnica de basada en el problema como anteriormente en el anterior punto mencionamos.

En cada función se estima un tamaño en LDC respecto a un valor óptimo (Vo), un valor medio(Vm), y un valor pesimista (Vp) y una vez se tiene los valores podemos sacar el valor esperado a partir de esta fórmula:

**VE** = (Vo + 4Vm + Vp)/ 6

**Módulo de Gestión de Gestión de actividades**

| **Función** | **Vo** | **Vm** | **Vp** | **Ve** |
| --- | --- | --- | --- | --- |
| **AñadirActividad** | 20 | 25 | 40 | 26,67 |
| **EliminarActividad** | 14 | 22 | 35 | 22,84 |
| **Consulta\_actividades** | 40 | 50 | 73 | 52,17 |
| **Modificación\_actividad** | 40 | 48 | 60 | 48,67 |

**Módulo de Gestión de medicamentos**

| **Función** | **Vo** | **Vm** | **Vp** | **Ve** |
| --- | --- | --- | --- | --- |
| **AltaMedicamento** | 30 | 40 | 50 | 40 |
| **ActualizaciónMédica** | 30 | 36 | 47 | 36,84 |
| **BajaMedicamento** | 20 | 30 | 40 | 30 |
| **VisualizacionMedica** | 15 | 20 | 25 | 20 |

**Módulo de Gestión de los profesionales del cuidado**

| **Función** | **Vo** | **Vm** | **Vp** | **Ve** |
| --- | --- | --- | --- | --- |
| **VisulizacionHora** | 15 | 20 | 25 | 20 |
| **AsignarHora** | 15 | 17 | 20 | 17,17 |
| **EliminarHora** | 20 | 30 | 40 | 30 |
| **Necesidades** | 40 | 48 | 60 | 48,67 |

**Módulo de Gestión de residentes**

| **Función** | **Vo** | **Vm** | **Vp** | **Ve** |
| --- | --- | --- | --- | --- |
| **AltaResidentes** | 30 | 40 | 50 | 40 |
| **BajaResidentes** | 20 | 30 | 40 | 30 |
| **ModifyResidentes** | 40 | 50 | 73 | 52,17 |
| **Residentes** | 30 | 40 | 50 | 40 |

**Módulo de Gestión de estancia**

| **Función** | **Vo** | **Vm** | **Vp** | **Ve** |
| --- | --- | --- | --- | --- |
| **AltaEstancia** | 30 | 40 | 50 | 40 |
| **BajaEstancia** | 20 | 30 | 40 | 30 |
| **ModifyEstancia** | 40 | 50 | 73 | 52,17 |
| **ModifyDuracionEst** | 40 | 50 | 60 | 50 |

**Módulo de Gestión de las patologías de los residentes**

| **Función** | **Vo** | **Vm** | **Vp** | **Ve** |
| --- | --- | --- | --- | --- |
| **DatosPatologia** | 40 | 50 | 60 | 50 |
| **DatosCliente** | 30 | 40 | 50 | 40 |
| **AnalisisCoincidencias** | 40 | 50 | 73 | 52,17 |
| **Resultados** | 13 | 20 | 30 | 20,50 |

**Módulo de Gestión de pagos**

| **Función** | **Vo** | **Vm** | **Vp** | **Ve** |
| --- | --- | --- | --- | --- |
| **AñadirDeuda** | 20 | 25 | 40 | 26,67 |
| **EliminarDeuda** | 10 | 17 | 25 | 17,17 |
| **AumentarDeuda** | 40 | 50 | 73 | 52,17 |
| **DisminuirDeuda** | 13 | 20 | 30 | 20,50 |

**Módulo de Gestión de patologías**

| **Función** | **Vo** | **Vm** | **Vp** | **Ve** |
| --- | --- | --- | --- | --- |
| **AñadirPatología** | 20 | 25 | 40 | 26,67 |
| **EliminarPatología** | 10 | 17 | 25 | 17,17 |

| **Función** | **Vo** | **Vm** | **Vp** | **Ve** |
| --- | --- | --- | --- | --- |
| **Interfaz** | 500 | 600 | 700 | 600 |
| **GestionDatos** | 900 | 1000 | 1100 | 1000 |

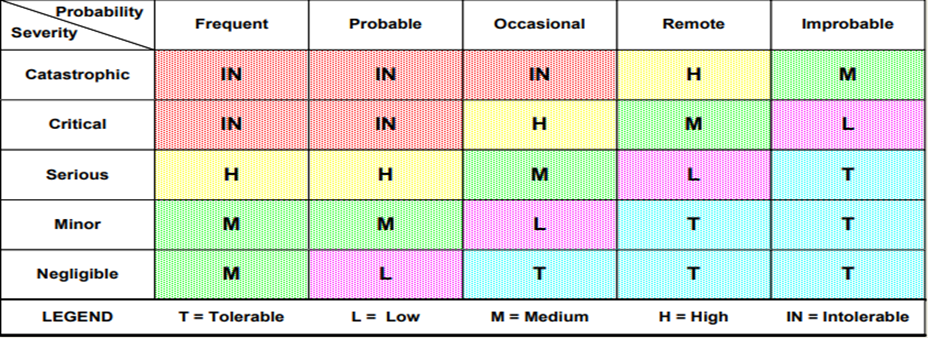
| **Total** |  |  |  | 2650,39 |
| --- | --- | --- | --- | --- |

El proyecto al tener que ser realizado en un plazo de 3 meses (lo que dura un cuatrimestre) la productividad media del equipo será de 130 LDC/pm

Esfuerzo = 2650,39 /130 =20,388 pm

De esta forma calculamos el esfuerzo que nuestro equipo necesitará para poder llevar a cabo este proyecto “RESGES” en el periodo de tiempo acordado.

1. **Estrategia de gestión del riesgo**
   1. **Introducción y Estudio de los riesgos**



Para el estudio de los riesgos utilizaremos una tabla de riesgos basada en el mecanismo SQAS-SEI

| **Nombre del Riesgo** | **Descripción del Riesgo** | **Probabilidad** | **Efecto (severidad o Consecuencias)** |
| --- | --- | --- | --- |
| **Estimación inadecuada del tiempo** | Al ser el primer proyecto que realizamos es sencillo que las estimaciones de tiempos para cada parte estén lejos de la realidad | Frequent | Critical |
| **Errores de planificación** | Se podría encontrar errores a la hora de la planificación por la ya mencionada falta de experiencia aunque al ser un problema mucho más fácilmente solucionable no supondría un impacto tan grande como el anterior | Probable | Serious |
| **Complicaciones con nuevas herramientas** | Es factible que a la hora de manejar las nuevas herramientas necesarias para este nuevo proyecto encontremos problemas al no estar familiarizados | Probable | Minor |
| **Incumplimiento de plazos de entrega** | Si los errores en las estimaciones de tiempo llegasen a ser muy severos podría llevar al punto en el que no podamos cumplir con los plazos de entrega | Ocasional | Catastrophic |
| **Re evaluación de requisitos** | Debido a la falta de experiencia puede que los requisitos iniciales no se cumplan pudiendo llegar a necesitar varios intentos hasta alcanzar un buen juicio | Frequent | Serious |
| **Falta de conocimientos clave en el personal** | Ya que este es un proyecto relativamente sencillo es difícil que aparezcan problemas de los que no tengamos el conocimiento necesario para solucionar y en el caso de hacerlo lo más probable es que fuésemos capaces de adquirir el conocimiento necesario rápidamente | Remote | Minor |
| **Problemas de organización** | Siempre podrían aparecer problemas a la hora de la división de trabajo y la organización de este pero al ser un equipo pequeño resulta una posibilidad remota y fácilmente solucionable | Ocasional | Minor |

* 1. **Priorización de riesgos del proyecto**

| **Nivel de Gravedad** | **Nombre del Riesgo** | **Descripción del Riesgo** |
| --- | --- | --- |
| **IN** | **Incumplimiento de plazos de entrega** | Si los errores en las estimaciones de tiempo llegasen a ser muy severos podría llevar al punto en el que no podamos cumplir con los plazos de entrega |
| **IN** | **Estimación inadecuada del tiempo** | Al ser el primer proyecto que realizamos es sencillo que las estimaciones de tiempos para cada parte estén lejos de la realidad |
| **H** | **Re evaluación de requisitos** | Debido a la falta de experiencia puede que los requisitos iniciales no se cumplan pudiendo llegar a necesitar varios intentos hasta alcanzar un buen juicio |
| **H** | **Errores de planificación** | Se podría encontrar errores a la hora de la planificación por la ya mencionada falta de experiencia aunque al ser un problema mucho más fácilmente solucionable no supondría un impacto tan grande como el anterior |
| **M** | **Complicaciones con nuevas herramientas** | Es factible que a la hora de manejar las nuevas herramientas necesarias para este nuevo proyecto encontremos problemas al no estar familiarizados |
| **L** | **Problemas de organización** | Siempre podrían aparecer problemas a la hora de la división de trabajo y la organización de este pero al ser un equipo pequeño resulta una posibilidad remota y fácilmente solucionable |
| **T** | **Falta de conocimientos clave en el personal** | Ya que este es un proyecto relativamente sencillo es difícil que aparezcan problemas de los que no tengamos el conocimiento necesario para solucionar y en el caso de hacerlo lo más probable es que fuésemos capaces de adquirir el conocimiento necesario rápidamente |

* 1. **Plan de gestión del riesgo Reducción, supervisión y gestión del riesgo**

**3.3.1.1 Incumplimiento de plazos de entrega**

Si los errores en las estimaciones de tiempo llegasen a ser muy severos podría llevar al punto en el que no podamos cumplir con los plazos de entrega

**3.3.1.2 Supervisión**

**-a) Forma de detección:** en el caso de detectar si ha sucedido posiblemente ya sea demasiado tarde, pero para ello se harán estimaciones de la fecha de finalización constantemente

**-b) Implementación de reducción:** con el fin de evitar el incumplimiento de los plazos de entrega se ha de trabajar de manera ordenada y constante con el fin de detectar cuanto antes todos los contratiempos que en un caso extremo nos podrían brindar esta posibilidad

**3.3.1.3 Plan de contingencia:**

Si llegasemos al caso en el que las estimaciones superan la fecha límite se avisará de forma inmediata a todo el equipo y se tomaría una decisión en conjunto sobre cómo aumentar nuestra carga de trabajo de tal forma que se cumplan las fechas

**3.3.2.1 Estimación inadecuada del tiempo**

Al ser el primer proyecto que realizamos es sencillo que las estimaciones de tiempos para cada parte estén lejos de la realidad

**3.3.2.2 Supervisión**

**-a) Forma de detección:** se encargará la realización de nuevas estimaciones con la aparición de cada nuevo inconveniente

**-b) Implementación de reducción:** Para evitar una estimación inadecuada del tiempo se encargará a uno de los integrantes del grupo la investigación sobre este ámbito de proyectos exitosos

**3.3.2.3 Plan de contingencia:**

Con su detección se procederá de similar manera que en el caso de una detección de incumplimiento de casos de entrega solo que valorando cuantos deberemos llevar a cabo estas medidas

**3.3.3.1 Re evaluación de requisitos**

Debido a la falta de experiencia puede que los requisitos iniciales no se cumplan pudiendo llegar a necesitar varios intentos hasta alcanzar un buen juicio

**3.3.3.2 Supervisión**

**-a) Forma de detección:** La ya mencionada línea de comunicación accesible, rápida y operativa

**-b) Implementación de reducción:** Mantener una línea de comunicación con el cliente operativa en todo momento y mantener un listado de requisitos presente en todo momento

**3.3.3.3 Plan de contingencia:**

En caso de un cambio inesperado y difícil se tratará de hacer una reevaluación de tiempo y recursos excepcional y se tomará una decisión respecto al resultado

**3.3.4.1 Errores de planificación**

Se podría encontrar errores a la hora de la planificación por la ya mencionada falta de experiencia aunque al ser un problema mucho más fácilmente solucionable no supondría un impacto tan grande como el anterior

**3.3.4.2 Supervisión**

**-a) Forma de detección:** estaremos atentos a los problemas presentes en cada uno de los puestos y se instará a los miembros del equipo a transmitir cualquier dificultad lo antes posible

**-b) Implementación de reducción:** Se asignará un tiempo más largo de lo normal para que los integrantes del equipo investiguen sobre la planificación en proyectos similares así como para el establecimiento de la nuestra propia y su revisión.

**3.3.4.3 Plan de contingencia:**

En caso de darse se determinará a cuanta parte de la planificación deberá de ser cambiada y que parte del equipo movilizada

**3.3.5.1 Complicaciones con nuevas herramientas**

Es factible que a la hora de manejar las nuevas herramientas necesarias para este nuevo proyecto encontremos problemas al no estar familiarizados

**3.3.5.2 Supervisión**

**-a) Forma de detección:** La detección se dará directamente al los miembros informar sobre si estos tienen o no el conocimiento sobre la herramienta requerida en cuestión.

**-b) Implementación de reducción:** Intentaremos prevenir estos problemas mediante el uso de herramientas que ya sean conocidas por el equipo.

**3.3.5.3 Plan de contingencia:**

En caso de necesitar obligatoriamente usar herramientas nuevas para el equipo investigaremos previamente para seleccionar aquella que sea más intuitiva y fácil de usar.

**3.3.6.1 Problemas de organización**

Siempre podrían aparecer problemas a la hora de la división de trabajo y la organización de este pero al ser un equipo pequeño resulta una posibilidad remota y fácilmente solucionable

**3.3.6.2 Supervisión**

**-a) Forma de detección:** se revisará periódicamente el avance en el trabajo realizado por los miembros del equipo así como se insistirá para que con cualquier problema presente o futuro que vean sea informado

**-b) Implementación de reducción:** Para evitar problemas con la organización se dejará por escrito y de manera clara y precisa toda la estructura de trabajo

**3.3.6.3 Plan de contingencia:**

Si fuese necesario se determinará la extensión del problema y actuará consecuentemente obteniendo una nueva versión de la organización original que lo solucione.

**3.3.7.1 Falta de conocimientos clave en el personal**

Ya que este es un proyecto relativamente sencillo es difícil que aparezcan problemas de los que no tengamos el conocimiento necesario para solucionar y en el caso de hacerlo lo más probable es que fuésemos capaces de adquirir el conocimiento necesario rápidamente

**3.3.7.2 Supervisión**

**-a) Forma de detección:** el propio miembro del grupo que tenga el conocimiento necesario para el avance del proyecto deberá avisar al resto de los miembros, aunque en caso de no hacerlo se podría detectar mediante los retrasos en los resultados

**-b) Implementación de reducción:** Antes de comenzar el proyecto intentaremos confirmar que tenemos los conocimientos básicos necesarios para la realización de este mismo

**3.3.7.3 Plan de contingencia:**

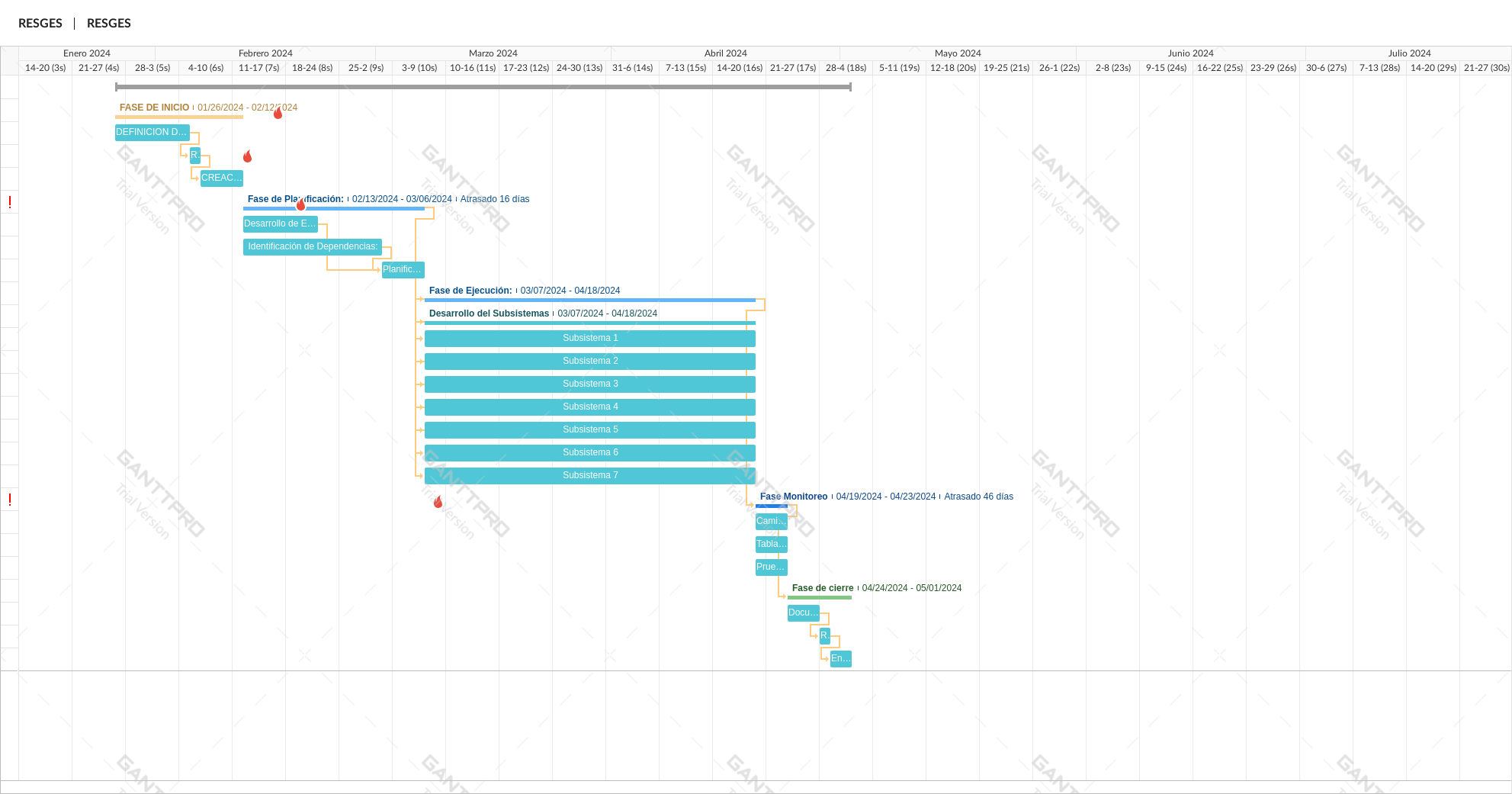
Si esto fuese detectado se cambiaría al encargado de la tarea problemática en caso de ser posible dejando como última opción la paralización de esa parte del proyecto y encargando al trabajador el estudio y aprendizaje de los conocimientos requeridos.

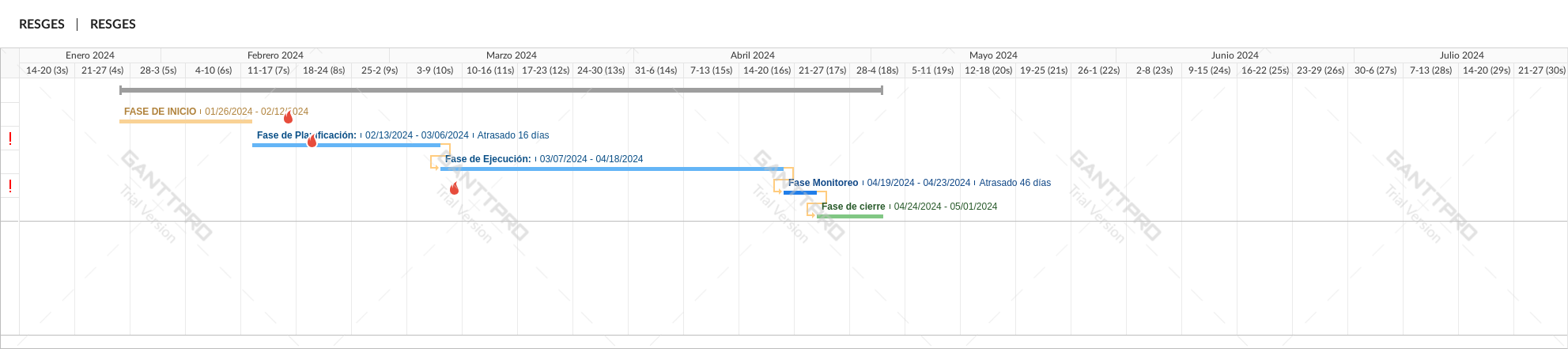
* 1. **Planificación temporal del Control de Riesgos**

Al encontrarnos frente a un proyecto de baja dificultad y sin gran carga de trabajo la mayoría de riesgos son mitigados o reducidos e incluso en el caso en que se den podrían ser solucionados de manera casi inmediata gracias a la fácil comunicación al ser un grupo pequeño de trabajadores y no contar con una carga de trabajo muy grande pudiendo hacer horas extras en caso de la aparición de un problema extraordinario. Por todo esto los riesgos se gestionaran una vez aparezcan los problemas asociados o próximo a que ocurra.

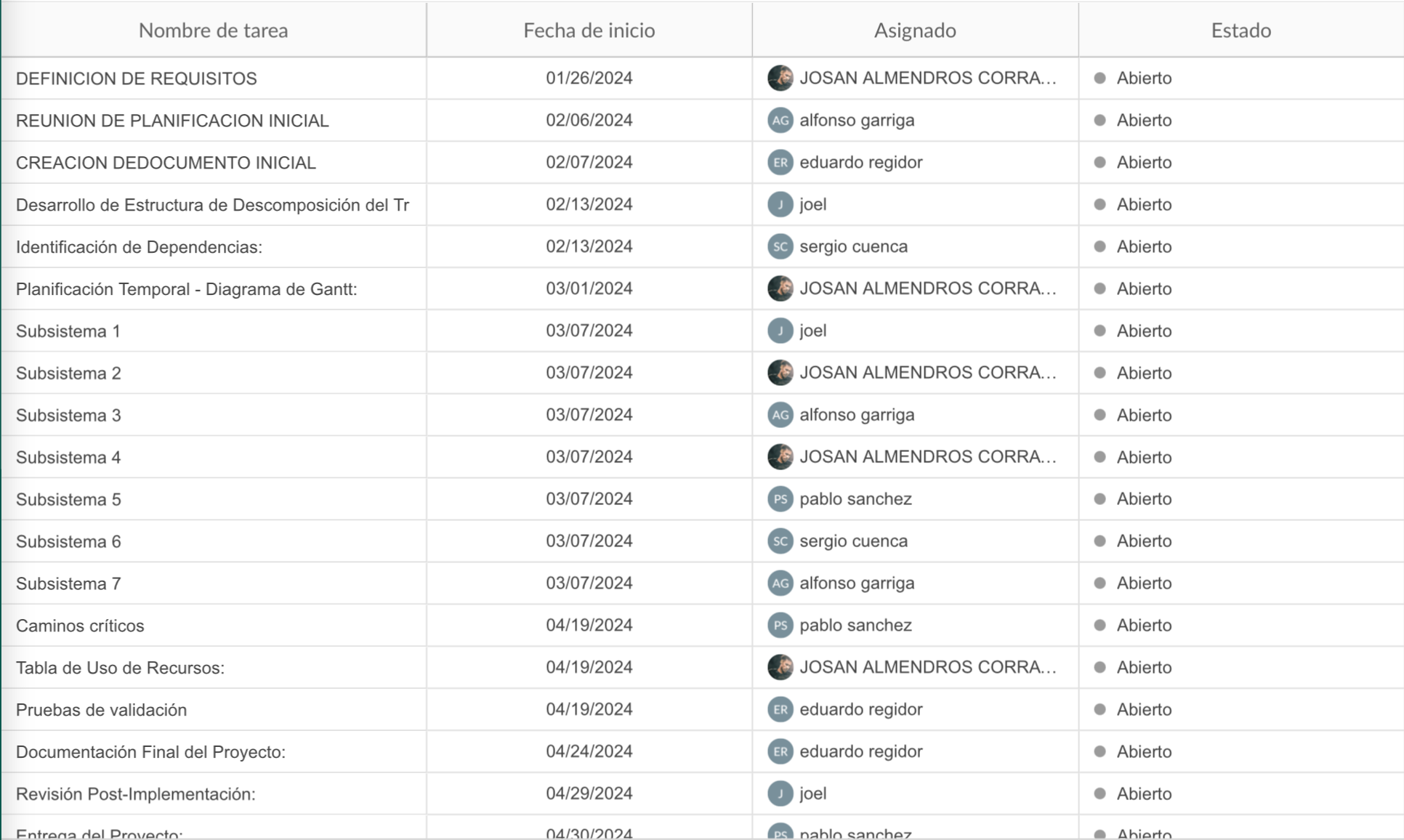
* 1. **Resumen (opcional)**

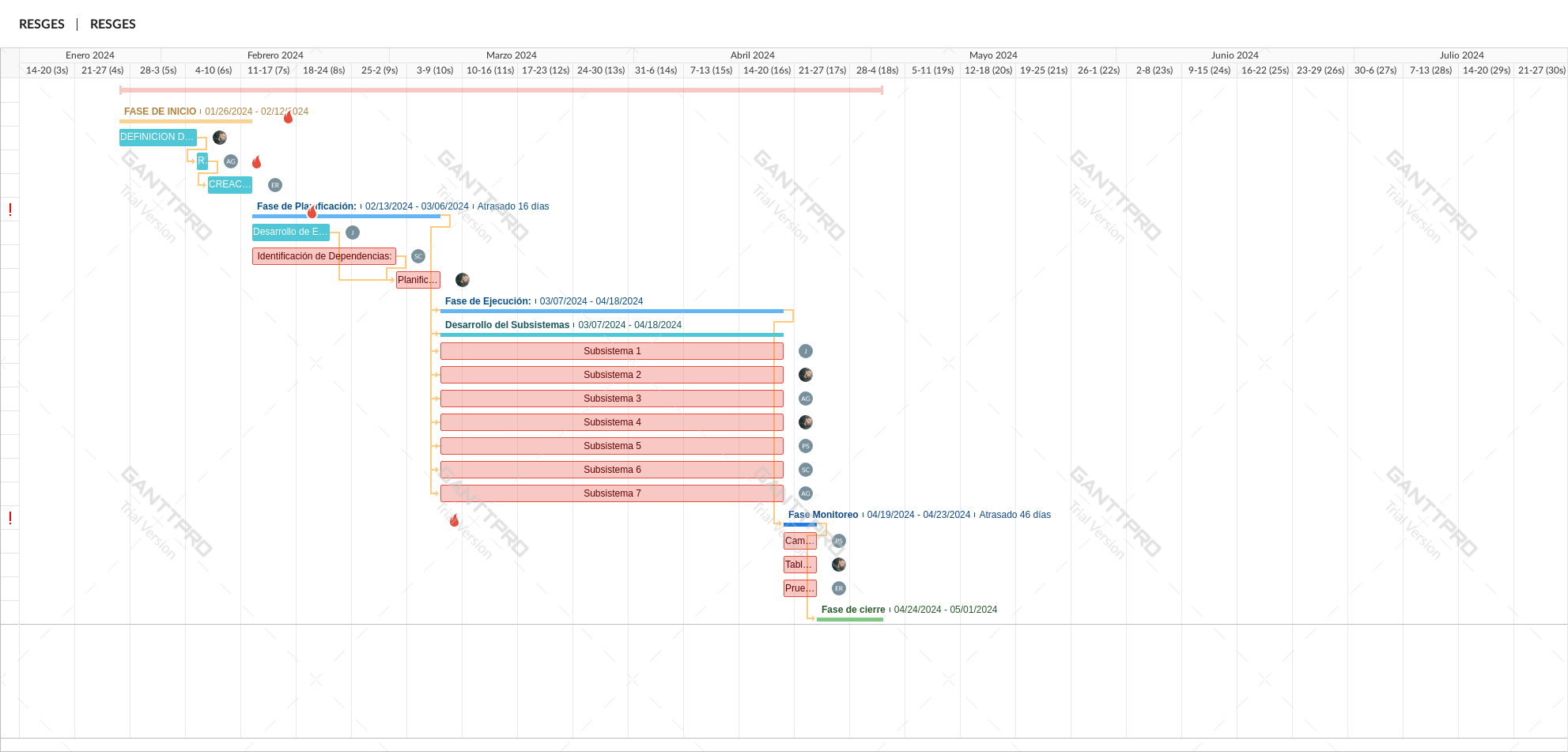
La mayor parte de las dificultades de nuestro proyecto caen en nuestra falta de experiencia pero debido a la baja extensión del mismo dichos problemas son fácilmente solucionables mediante el trabajo constante y la comunicación rápida.

1. **Planificación temporal (se entregará el fuente en MS Project)**
   1. **Estructura de descomposición del trabajo/Planificación temporal mediante Diagrama de Gantt**



* 1. **Red de tareas (generada por MS Project)**



* 1. **Caminos Críticos**

Los caminos críticos se muestran en color rojo.

* 1. **Tabla de uso de recursos**

| **Tarea** | **Recurso** | **Rol** | **Horas Asignadas** |
| --- | --- | --- | --- |
| Definición de Requisitos | Equipo de Proyecto | Analista de Requisitos | 20h |
| Reunión de Planificación Inicial | Líder de Proyecto | Project Manager | 8h |
| Creación de Documentación Inicial | Documentador | Documentador | 15h |
| Desarrollo de EDT | Líder de Proyecto | Project Manager | 30h |
| Identificación de Dependencias | Analista de Proyecto | Analista de Proyecto | 20h |
| Planificación Temporal - Diagrama de Gantt | Líder de Proyecto | Project Manager | 15h |
| Desarrollo del Subsistema 1.1 - DCU | Desarrollador | Desarrollador | 40h |
| Implementación de la Alta de Estancia | Desarrollador | Desarrollador | 25h |
| Desarrollo de la Baja de Estancia | Desarrollador | Desarrollador | 25h |
| Implementación del Cambio de Estancia | Desarrollador | Desarrollador | 30h |
| Caminos Críticos y Revisión de Hitos | Líder de Proyecto | Project Manager | 15h |
| Tabla de Uso de Recursos | Analista de Proyecto | Analista de Proyecto | 10h |
| Pruebas de Validación | Equipo de Pruebas | Tester | 40h |
| Documentación Final del Proyecto | Documentador | Documentador | 20h |
| Revisión Post-Implementación | Líder de Proyecto | Project Manager | 10h |
| Entrega del Proyecto | Líder de Proyecto | Project Manager | 15h |

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

* Líder del Proyecto:
  + Responsabilidades:
    - Coordinar y supervisar todas las actividades del proyecto.
    - Asegurar que se cumplan los plazos y objetivos.
  + Habilidades Requeridas:
    - Habilidades de liderazgo y gestión de proyectos.
    - Comunicación efectiva y toma de decisiones.
  + Integrante Asignado:
    - Jose Antonio Almendros Corrales
* Desarrolladores de Software:
  + Responsabilidades:
    - Implementar el software y realizar pruebas de unidad.
    - Colaborar con analistas y diseñadores.
  + Habilidades Requeridas:
    - Experiencia en desarrollo de software.
    - Conocimientos técnicos en lenguajes de programación relevantes.
  + Integrante Asignado:
    - Eduardo Regidor Serrano
* Analistas de Requisitos:
  + Responsabilidades:
    - Analizar y documentar los requisitos del software.
    - Colaborar con el equipo de desarrollo.
  + Habilidades Requeridas:
    - Conocimiento en técnicas de análisis de requisitos.
    - Habilidades de documentación.
  + Integrante Asignado:
    - Alfonso Barriga Lucena
* Diseñadores de Software:
  + Responsabilidades:
    - Diseñar la arquitectura del software.
    - Colaborar con desarrolladores en la implementación.
  + Habilidades Requeridas:
    - Experiencia en diseño de software.
    - Conocimientos en herramientas de modelado.
  + Integrante Asignado:
    - Andrés Enrique García de Abroleya Ortiz

#### 5.2 Hardware y Software:

* Hardware:
  + Requerimientos de Hardware:
    - Computadoras y estaciones de trabajo para el desarrollo y pruebas.
    - Servidores para la implementación del software.
* Software:
  + Herramientas de Desarrollo:
    - IDEs, compiladores, entornos de desarrollo.
  + Software de Gestión:
    - Plataformas de gestión de proyectos.
  + Diseño y Modelado:
    - Herramientas para diseño y modelado de software.

#### 5.3 Lista de Recursos:

* Espacio de Trabajo:
  + Requisitos de Espacio:
    - Oficinas o espacios de trabajo para el equipo.
    - Áreas para reuniones y colaboración.
* Documentación del Proyecto:
  + Documentos Necesarios:
    - Manuales, documentación técnica y manuales de usuario.
    - Registros y seguimientos del proyecto.
* Acceso a Bases de Datos:
  + Requerimientos:
    - Servidores o plataformas de bases de datos necesarios.
    - Acceso controlado a datos sensibles.

**6. Organización del personal**

#### 6.1 Estructura de Equipo:

* Jerarquía del Equipo:
  + Roles y Responsabilidades:
    - Definición clara de funciones y responsabilidades.
    - Establecimiento de la cadena de mando.
  + Integrante Asignado:
    - Sergio Cuenca López
* Equipos por Subsistema:
  + Asignación Específica:
    - Formación de equipos dedicados a cada subsistema.
    - Colaboración interdisciplinaria cuando sea necesario.
  + Integrante Asignado:
    - Jose Antonio Almendros Corrales

Hemos optado por una estructura de equipo jerárquica para el proyecto "ResGes" por varias razones que se alinean con las características del equipo, el modelo de proceso y los objetivos del proyecto:

1. Claridad de Roles y Responsabilidades:

- La estructura jerárquica proporciona una definición clara de roles y responsabilidades para cada miembro del equipo.

- Cada integrante tiene un propósito específico y áreas de responsabilidad asignadas, lo que reduce la ambigüedad y mejora la eficiencia.

2. Cadena de Mando Definida:

- La jerarquía establece una cadena de mando clara, facilitando la toma de decisiones y la comunicación eficiente.

- El líder del proyecto tiene la autoridad final, lo que agiliza el proceso de toma de decisiones y evita posibles conflictos de dirección.

3. Adaptabilidad al Modelo de Proceso:

- Dado que estamos siguiendo un modelo de proceso iterativo e incremental, la estructura jerárquica se adapta bien a la necesidad de gestionar cambios y ajustes continuos.

- La jerarquía permite una rápida reasignación de tareas y recursos según las demandas cambiantes del proyecto.

4. Experiencia y Especialización:

- La estructura jerárquica nos permite asignar a los miembros del equipo a áreas específicas donde puedan aprovechar sus habilidades y experiencia al máximo.

- Cada subsistema cuenta con un líder asignado que supervisa el desarrollo y asegura una comprensión profunda de los detalles técnicos.

5. Facilita la Comunicación Vertical y Horizontal:

- La jerarquía facilita la comunicación tanto vertical como horizontal. La comunicación vertical fluye de manera eficiente a través de la cadena de mando, mientras que la colaboración horizontal se mantiene a través de equipos dedicados a subsistemas específicos.

6. Gestión de Recursos Eficiente:

- La asignación de equipos a subsistemas permite una gestión de recursos más eficiente y un enfoque dedicado a aspectos particulares del proyecto.

- Se minimizan las posibilidades de duplicación de esfuerzos y se maximiza la especialización en cada área funcional.

En resumen, la estructura jerárquica proporciona una base sólida para la gestión eficiente del equipo, se adapta bien al modelo de proceso iterativo y fomenta la especialización y la comunicación efectiva. Estas ventajas son fundamentales para alcanzar los objetivos del proyecto "ResGes" de manera exitosa.

#### 6.2 Informes de Gestión:

* Informes de Progreso:
  + Frecuencia:
    - Reportes regulares sobre el estado del proyecto.
    - Actualizaciones semanales o quincenales.
  + Integrante Asignado:
    - Joel Valencia
* Comunicación Interna:
  + Mecanismos de Comunicación:
    - Establecimiento de canales de comunicación efectivos.
    - Reuniones periódicas para discutir avances y desafíos.
  + Integrante Asignado:
    - Joel Valencia Llorente
* Resolución de Problemas:
  + Procedimientos:
    - Establecimiento de procedimientos para abordar problemas y desviaciones.
    - Mecanismos claros para la toma de decisiones y resolución de conflictos.
  + Integrante Asignado:
    - Alfonso Barriga Lucena

**7. Mecanismos de seguimiento y control**

7.1 **Garantía de calidad y control (Plan de Calidad)**

**7.1.1 Introducción: Qué Gestión de calidad haces: quien, roles y tareas**

Se establecerá un equipo de Gestión de Calidad (SQA) para supervisar la calidad del producto. Este equipo trabajará en colaboración con los desarrolladores, asegurando que el software cumpla con las especificaciones de diseño a lo largo de su realización. Las tareas incluirán la ejecución de pruebas, revisiones e inspecciones para garantizar que cada componente cumple con los requisitos asignados.

**7.1.2 Qué Tecnologías de IS: Métodos y herramientas**

Durante el desarrollo de este software, se implementarán tecnologías de Ingeniería del Software específicas, adaptadas a las necesidades de gestión de residencias de ancianos. Esto puede abarcar desde metodologías especializadas hasta herramientas diseñadas para la planificación de actividades y gestión de la salud de los residentes.

**7.1.3 Qué Revisiones Técnicas Formales**

Como parte integral de la garantía de calidad, se llevarán a cabo Revisiones Técnicas Formales (RTFs) del diseño del sistema. Estas revisiones, realizadas en reuniones planificadas por los desarrolladores, permitirán identificar posibles problemas y generar un informe detallado que incluirá una lista de problemas de revisión. Esta práctica garantizará que el software se ajuste a los estándares de calidad establecidos por la organización.

**7.1.4 Qué Estrategia de pruebas**

La estrategia de pruebas se centrará en validar la funcionalidad del software. Se llevarán a cabo pruebas dinámicas e inspecciones estáticas como parte del proceso de validación y verificación (V & V) para garantizar que el software satisfaga las necesidades del usuario.

**7.1.5 Qué Control de la documentación y de cambios**

Se implementará un sistema de control estricto sobre la documentación del proyecto y los cambios realizados. Esto garantizará la consistencia y trazabilidad de la información a medida que evoluciona el proyecto, permitiendo una gestión eficiente de la documentación actualizada.

**7.1.6 Qué Mecanismos de medición y de Informes de errores, desajustes, desviaciones, métricas**

Se establecerán mecanismos de medición adaptados a la gestión de residencias de ancianos, incluyendo la evaluación de la eficacia en la gestión de la salud de los residentes y la eficiencia en las operaciones diarias. Los informes detallados incluirán análisis de errores, desajustes y desviaciones, proporcionando información valiosa para la mejora continua del software en este contexto específico.

7.2 **Gestión y control de cambios (Plan GCS)**

**7.2.1 Introducción: Propósito, Alcance, Definiciones, Referencias**

El Plan GCS tiene como propósito adaptar el sistema a cambios emergentes, ya sea por las necesidades del cliente o la evolución del proyecto. Se llevará a cabo una supervisión rigurosa del funcionamiento del software a lo largo de su vida útil, garantizando su eficiencia y alineación con los objetivos del proyecto.

**7.2.2 Tipos de artefactos a gestionar (los ECSs)**

Los Elementos de Configuración de Software (ECSs) incluyen módulos específicos del software, bases de datos, y documentos generados durante el desarrollo del sistema.

**7.2.3 Criterios y protocolos para Nombrar los ECSs**

Para facilitar la identificación y gestión, se establecerán criterios y protocolos específicos para nombrar consistentemente los ECSs, asegurando una clara correlación entre los elementos y sus funciones dentro del software.

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

Se designará un responsable para supervisar los procedimientos de GCS y liderar la creación de Líneas Base, garantizando la coherencia y estabilidad del sistema en un entorno tan crítico.

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

**- Estándares:**

En el grupo se seguirán estándares específicos para mantener uniformidad en la estructura y formato de documentos relacionados con el software.

**- Plantillas que usas en documentos**

Se utilizarán plantillas adaptadas a las particularidades del proyecto, asegurando coherencia en la documentación generada.

**- Pasos a dar en un cambio**

Los pasos para implementar cambios seguirán un enfoque estructurado, abordando la identificación del problema, análisis, reuniones técnicas, modificación, evaluación y actualización de versiones, garantizando la solidez del sistema.

**- Acuerdos en productos**

Se establecerán acuerdos claros sobre la evolución de los productos en el contexto del proyecto, asegurando consistencia y calidad.

**- Comité de Cambios: Quién hace qué**

Todos los miembros del grupo formarán parte del Comité de Cambios, desempeñando funciones desde la identificación de problemas hasta la evaluación y actualización de versiones, asegurando una gestión efectiva de cambios en el software.

**- Herramientas de gestión de versiones:**

**– Repositorios**

Se empleará el repositorio *GitHub* para gestionar versiones del software, facilitando la colaboración entre los miembros del grupo.

**7.2.6 Registros para mantener el rastro de los cambios**

Se establecerá un registro detallado de cambios en documentos y el proyecto. Se mantendrá un registro específico para cambios documentales, utilizando herramientas de seguimiento integradas. Además, se llevará un registro centralizado de cambios en el proyecto, con detalles como fecha, descripción y responsable. Este registro se actualizará regularmente para reflejar la evolución del software.