

AgroManagement

Plan del proyecto

Álvaro González Sánchez

Federico Ibáñez Moruno

Kurosh Javier Dabbagh Escalante

Marc Mila Peramos

Simón Augusto Da Silva Álvarez

Youness El Guennouni

Raúl Caballero Girol

Control de cambios

[illegible]

Índice de contenido

Introducción.....	4
Propósito del plan.....	4
Ámbito del proyecto y objetivos.....	4
Declaración del ámbito.....	4
Funciones principales.....	4
Aspectos de rendimiento.....	5
Restricciones y técnicas de gestión.....	5
Modelo de proceso.....	5
Estimaciones del proyecto.....	6
Datos históricos.....	6
Técnicas de estimación.....	6
Estimaciones de esfuerzo, coste y duración.....	6
Estrategia de gestión del riesgo.....	7
Análisis del riesgo.....	8
Estudio de los riesgos.....	9
Plan de gestión del riesgo.....	10
Planificación temporal.....	12
Estructura de descomposición del trabajo / Planificación temporal.....	12
Gráfico Gantt.....	13
Red de tareas.....	13
Tabla de uso de recursos.....	13
Recursos del proyecto.....	13
Personal.....	13
Hardware y software.....	14
Lista de recursos.....	14
Organización del personal.....	14
Estructura de equipo.....	14
Informes de gestión.....	14
Mecanismos de seguimiento y control.....	15
Garantía de calidad y control.....	15
Gestión y control de cambios.....	16

Introducción

Propósito del plan

El presente documento tiene el propósito de detallar de forma clara y concisa un plan de proyecto, con el fin de conseguir una planificación definida y evitar problemas como entregas fuera de la fecha estipulada.

Ademas, servirá como guía tanto al jefe de proyecto como a los desarrolladores durante todas las fases de desarrollo del producto.

El presente documento sigue el guión establecido por el modelo Pressman donde se aborda la planificación completa del proyecto que será desarrollado por los todos los miembros del grupo.

Ámbito del proyecto y objetivos

AgroManagement es una herramienta pensada para el pequeño agricultor. Le proporcionará las herramientas necesarias para la gestión de una explotación agraria que, tradicionalmente, se viene realizando en papel.

Gracias a esto, se logrará una mayor productividad y un control exhaustivo de los recursos de la explotación, logrando con esto un considerable ahorro de costes.

Declaración del ámbito

Gracias a las nuevas tecnologías se ha abierto un gran abanico de posibilidades en cuanto a la gestión de pequeñas y medianas empresas. AgroManagement logrará beneficios en empresas pequeñas o agricultores autónomos en la gestión de los recursos con los que cuentan.

La empresa será una pequeña explotación con varias fincas. La empresa decide en cada momento qué trabajos se realizan en cada finca y con qué recursos cuenta, asignando los recursos necesarios a cada trabajo programado.

La gestión de los recursos y los trabajos realizados en cada finca se realizan actualmente de manera manual basados en la observación (recursos disponibles en el almacén) o mediante un registro en papel con las posibles consecuencias que eso puede tener (extravío, destrucción, ilegibilidad).

Además, gracias a AgroManagement, se dará la posibilidad a los trabajadores de consultar los trabajos programados para el día y los recursos de los que disponen. Podrán comparar de una cosecha a otra los recursos utilizados y si han sido beneficiosos o perjudiciales a la hora de la recolección

Funciones principales

AgroManagement se divide en 4 módulos principales:

- Gestión de usuarios

La función del módulo de gestión de usuarios será la creación, modificación, consulta y eliminación de los usuarios con acceso al sistema.

- Gestión de fincas

La función del módulo de gestión de fincas será la creación, modificación, consulta y eliminación de las fincas asociadas a la explotación.

- Gestión de recursos

La función del módulo de gestión de recursos será la creación, modificación, consulta y eliminación de los recursos asociados a la explotación.

- Gestión de trabajos

La función del módulo de gestión de trabajos será la creación, modificación, consulta y eliminación de los trabajos asociados a la explotación así como el control de los recursos asociados a dicho trabajo.

Aspectos de rendimiento

El rendimiento de la aplicación debe ser bueno para evitar crear sensación de lentitud al usuario. No obstante, al ser una aplicación en la que sólo actúa un usuario a la vez no se necesitará un rendimiento especialmente optimizado. Las acciones deberán tardar menos de 3 segundos en ejecutarse.

Restricciones y técnicas de gestión

La única restricción prevista será la de trabajar con sistema operativo compatible con Java 7. Además, con respecto a las técnicas de gestión, conviene especificar lo siguiente:

Una primera entrega, formada por la SRS y del Plan del Proyecto, será realizada el 23/01/2015. La segunda y última entrega, se formalizará el 21/05/2015, y consistirá en el producto final, incluyendo el diseño mismo de la aplicación y una memoria. Todas las entregas serán realizadas en un CD/DVD, y en el lugar donde el cliente considere oportuno.

Finalmente, el (fecha por determinar) se procederá a exponer al cliente el producto realizado. En cuanto a las técnicas de gestión, se crearán planes de seguridad e informes de estado, que se explicarán más detalladamente en el apartado “Informes de gestión” de este mismo documento.

Modelo de proceso

Para el presente proyecto se utilizará el Modelo Unificado de Desarrollo de software. Dicho modelo, al tratarse de un modelo iterativo, permitirá el avance del proyecto mediante subproductos funcionales.

Gracias a esto y debido a la poca experiencia del grupo en ingeniería de software, podrán ir descubriéndose fallos que posiblemente se hayan obviado en iteraciones anteriores.



En la fase de inicio se define el negocio: facilidad de realizar el proyecto, se presenta un modelo, visión, metas, deseos del usuario, plazos, costos y viabilidad.

Continuaremos con la elaboración. En esta fase se obtiene la visión refinada del proyecto a realizar, la implementación iterativa del núcleo de la aplicación, la resolución de riesgos altos, nuevos requisitos y se ajustan las estimaciones.

En la construcción se abarca la evolución hasta convertirse en producto listo incluyendo requisitos mínimos. Aquí se afinan los detalles menores como los diferentes tipos de casos o los riesgos menores.

En la fase final (transición), el programa debe estar listo para ser probado, instalado y utilizado por el cliente sin ningún problema. Una vez finalizada esta fase, se debe comenzar a pensar en futuras novedades para la misma.

Desde el punto de vista **Técnico**: el proyecto está formado por los flujos de trabajo fundamentales: captura de requerimientos, análisis, diseño, implementación y pruebas.

Estimaciones del proyecto

Datos históricos

Debido a la poca experiencia del equipo en ingeniería de software, no se posee de ningún dato histórico de otros proyectos sobre los que obtener una estimación.

Técnicas de estimación

Para la estimación del costo del software utilizaremos la técnica de estimación basada en el proceso, ayudándonos para ellos de la técnica de descomposición basada en el problema, descomponiendo el software en líneas de código (LDC). Esta medida nos dará una estimación bastante fiable ya que, al tratarse de un proyecto relativamente sencillo, las líneas de código suponen una métrica aceptable.

Estimaciones de esfuerzo, coste y duración

Se estima un esfuerzo humano para el proyecto de aproximadamente 8 horas semanales por miembro del equipo, estando éste integrado por 7 miembros.

Así pues, semanalmente se calculan 56 horas de trabajo para todo el equipo. Siendo la duración del proyecto aproximadamente 15 semanas (no se cuenta semana santa). El margen total del tiempo empleado en este proyecto será de aproximadamente 840 horas.

Con respecto al coste de este proyecto, dado que no se prevén ingresos, obviamente serán nulos.

Podemos dividir la aplicación en:

- Módulo de fincas (MF)
- Módulo de recursos (MR)
- Módulo de trabajos (MT)
- Módulo de usuarios (MU)
- Interfaz (IN)
- Persistencia de datos (PD)

Función	VOptimista	VMasProbable	VPesimista	VEsperado
MF	1000	1500	2000	1500
MR	1500	2000	3000	2166
MT	1500	2000	3000	2166
MU	1000	1500	2000	1500
IN	1500	2000	2500	200
PD	1000	1500	2000	1500

El total de LDC's es de 9032. Si la media estimada es de 300LDC/pm(8 horas semanales supondrían menos de 40LDC por hora), obtenemos lo siguiente:

$$\text{Esfuerzo} = 9032(\text{LDC})/250(\text{LDC}/\text{PM})=30\text{pm}$$

Al componerse el equipo de 7 personas, nos daría una duración de proyecto de 4,2 meses.

Suponiendo un coste por trabajador/mes de 2500€

Coste= 30pm*2500€=75000€

Estrategia de gestión del riesgo

Durante el desarrollo de AgroManagement se seguirá una estrategia proactiva de gestión de riesgos durante todo el desarrollo del proyecto. Se realizará una evaluación previa de los riesgos, se evaluarán las posibles consecuencias y se tomarán una serie de salvaguardas con el fin de minimizar en la medida de lo posible la aparición de los mismos.

Nuestro proyecto consta de cuatro módulos: fincas, usuarios, recursos y trabajos que serán la estructura principal de AgroManagement por lo cual es importante tener controlado cualquier riesgo que pueda ser generado durante su elaboración o que pueda surgir después de la entrega de nuestro software, para ello construiremos en los siguientes puntos las tablas de niveles de riesgo, además de la descripción y tratamiento de cada uno de ellos.

En este proyecto vamos a desarrollar la Taxonomía SEI de Riesgos del Software, analizaremos brevemente sus fases respecto a nuestro proyecto.



Comienza con el análisis de riesgos que constituyen posibles amenazas potenciales, en nuestro proyecto los riesgos con consecuencias muy altas son las relacionadas a los cambios que pueda realizar el cliente durante la planificación del proyecto, cambios en el calendario de entrega de progreso del proyecto y los problemas que puedan ocurrir dentro del grupo de desarrollo de software.

Continúa con la identificación de riesgos específicos que implican un conocimiento profundo del proyecto. Aquí podemos encontrar inconvenientes como los relacionados con el tipo de plataformas de SO usados por nuestro equipo, las herramientas de desarrollo o control de errores.

Evaluación y análisis

Analizamos las consecuencias y efectos de los riesgos. Estudiamos y creamos una tabla de efectos de riesgo: T=Tolerable B=Bajo M=Medio A=Alto IN=Intolerable.

Seguimiento, control, planes y comunicación:

En el presente documento realizamos la descripción y tratamiento de riesgos.

- Descripción: Identificamos el nivel de riesgo, aquí analizamos según la tabla de efectos el

potencial del riesgo, verificamos cuál es su naturaleza y quiénes son los encargados de la evaluación de ellos.

- Tratamiento de Riesgos: Planificamos las soluciones, que pueden ser realizadas por el jefe de proyecto o por los encargados de cada módulo, luego los planes de contingencia deben ser especificadas en consenso entre jefe y equipo de desarrollo, por último el seguimiento y control del riesgo deben generar un registro en informes.

Análisis del riesgo

Existen riesgos que pueden crear incertidumbre en nuestro proyecto. Para AgroManagement en el apartado anterior hemos identificado dos tipos de riesgos:

- Riesgos del proyecto: Afectan a la planificación temporal, al coste y calidad del proyecto.
- Riesgos técnicos: Amenazan la calidad y la planificación temporal del software (producto) que hay que producir. Identifican posibles problemas de incertidumbre técnica, ambigüedad en la especificación, diseño, implementación, obsolescencia técnica o tecnología puntera, interfaz, verificación y mantenimiento...

Para un correcto tratamiento de los riesgos relacionados con nuestro software antes de que afecten al proyecto, realizaremos la *Declaración de riesgos*.

Riesgo 1	
Elemento	Planificación.
Riesgo	Fecha final de entrega.
Descripción	La fecha de entrega del proyecto podría no cumplirse.
Condición	El tiempo de desarrollo podría alargarse y no poder realizar las suficientes pruebas.
Consecuencia	El software podría estar inacabado y/o con numerosos fallos.
Efecto	El software no podría ser entregado.

Riesgo 2	
Elemento	Requisitos del cliente.
Riesgo	Funcionalidad de la aplicación.
Descripción	Los requisitos recogidos son la interpretación de las necesidades del cliente. Una mala interpretación de dichas necesidades afectaría a la calidad del software.
Condición	El cliente exige que el software cumpla con los requerimientos.
Consecuencia	Un cambio en los requerimientos puede suponer un alto coste si el proyecto esta avanzado, lo que podría poner en peligro el cumplimiento de la fecha de entrega.
Efecto	El software no sería del agrado del cliente.

Riesgo 3	
Elemento	Cambio de requisitos.
Riesgo	Formulación de nuevos requisitos.
Descripción	Pueden surgir nuevos requisitos por parte del cliente que no se hayan considerado desde el principio.
Condición	Este riesgo puede surgir debido a un mayor conocimiento por parte del cliente de

	la herramienta.
Consecuencia	Un cambio en los requerimientos puede suponer un alto coste si el proyecto esta avanzado, lo que podría poner en peligro el cumplimiento de la fecha de entrega.
Efecto	El software no sería del agrado del cliente.

Riesgo 4	
Elemento	Gestión de configuraciones.
Riesgo	Cambios gestionados de manera incorrecta.
Descripción	Este riesgo surge cuando no se ha tenido en cuenta la GCS.
Condición	Los cambios pueden ser inevitables.
Consecuencia	Más tiempo del estimado para el desarrollo del proyecto.
Efecto	Son inevitables los cambios, ya sean de requisitos o de diseño, si no se aplica el proceso de GCS y una buena planificación podría desarrollarse un producto defectuoso.

Riesgo 5	
Elemento	Entorno de trabajo.
Riesgo	Problemas con el entorno de trabajo y lenguaje de programación.
Descripción	Falta de experiencia por parte de algún integrante del proyecto en el lenguaje con el que se desarrolla la aplicación.
Condición	Escasos conocimientos o insuficiente experiencia con las herramientas y lenguajes de programación que van a ser utilizados.
Consecuencia	Pérdida de tiempo en el aprendizaje de uso de las herramientas y lenguajes de programación necesaria.
Efecto	Incumplimiento de objetivos en el tiempo estimado.

Riesgo 6	
Elemento	Personal
Riesgo	Ausencia de algún miembro del equipo de trabajo.
Descripción	Algún miembro del proyecto puede sufrir alguna indisponibilidad (enfermedad, accidente, etc.) y por lo tanto no puede realizar tareas relacionadas con el proyecto.
Condición	Ausencia de algún integrante del equipo.
Consecuencia	Problemas con la planificación del proyecto.
Efecto	Retraso de sus tareas asignadas, dando lugar a un retraso en la planificación global.

Estudio de los riesgos

A continuación estudiaremos cada uno de los riesgos identificados en el apartado anterior, para saber cuál es la probabilidad, impacto y exposición de los riesgos del proyecto.

En el presente documento hemos definido las siguientes escalas:

IMPACTO	PROBABILIDAD	EXPOSICIÓN
Catastrófico	Frecuente	IN=Intolerable
Crítico	Probable	A=Alto
Serio	Ocasional	M=Medio
Menor	Remota	B=Bajo*
Insignificante	Improbable	T=Tolerable*

(*) No los tomaremos en cuenta en las siguientes tablas

Prestaremos total atención a los niveles de exposición (NE) que nos presenten $NE \geq M$, ya que estas requieren acciones correctivas con más urgencia.

Para ello utilizaremos la tabla de SQAS-SEI para calcular el nivel de riesgo.

Probabilidad Impacto	Frecuente	Probable	Ocasional	Remota	Improbable
Catastrófico	IN	IN	IN	A	M
Crítico	IN	IN	A	M	
Serio	A	A	M		
Menor	M	M			
Insignificante	M				

Nivel de los posibles Riesgos que se presenten en el proyecto:

Elemento	Probabilidad	Impacto	Exposición
Planificación	Probable	Serio	Alto
Requisitos del cliente	Ocasional	Serio	Medio
Cambio de requisitos	Ocasional	Serio	Medio
Gestión de configuraciones	Ocasional	Crítico	Alto
Herramientas de desarrollo	Ocasional	Serio	Medio
Calendario	Ocasional	Crítico	Alto

Plan de gestión del riesgo

A continuación citaremos los pasos fundamentales para realización de un plan de gestión de riesgo, además analizamos la lista ordenada de riesgos y se decide cómo se pueden tratar los riesgos, veremos cómo evitarlos, controlarlos, asumirlos o transferirlos.

1. Discutir todos los posibles resultados del proyecto, tanto los positivos como los negativos. Se realizarán reuniones periódicas convocadas por el jefe del equipo con el fin de detallar todos los posibles aspectos que podrían originar un fallo tanto en la aplicación como en el desarrollo de la misma, así como la valoración y reconocimiento del trabajo bien desarrollado.
2. Redactar el plan lo más completo posible incluyendo detalles y los recursos utilizados por los desarrolladores.

3. Evitar sobrevender el proyecto. La honestidad y un análisis directo son mucho más apropiados cuando se le pide a una Organización que tome riesgos sustanciales con sus activos. Se le explicará al cliente de manera honesta, clara y concisa los riesgos explicados con anterioridad con el fin de que entienda todos los posibles inconvenientes.
4. Pedir como mínimo a tres compañeros de trabajo que revisen el informe y que comuniquen las fallas en la lógica del mismo en el tiempo previo estimado a la entrega. Esto se realizará con el fin de que además de detectar algún posible fallo en la realización del informe, los distintos grupos conozcan además de su módulo las ideas fundamentales de los demás módulos.

Lista de riesgos

Riesgo: Fecha de entrega del proyecto.

Descripción del Riesgo: La puntualidad en la entrega del proyecto representará el prestigio y el compromiso con el cliente de nuestra empresa de desarrollo de software. Esta información está plasmada en el diagrama de Gantt donde se registra las actividades de cada miembro y los periodos de entrega. El responsable será el jefe del proyecto ya que establece las actividades para el grupo de desarrollo.

Tratamiento del Riesgo: Los recursos para realizar las actividades de control de riesgos son básicamente los informes generados por cada módulo. Se podría prevenir en la planeación del proyecto realizando un buen cálculo de los días de trabajo para el proyecto y también añadiendo días de contingencia. El plan de acción si estuviéramos en el caso que no lleguemos a la fecha de entrega, deberíamos intensificar las horas de trabajo o solicitar ayuda a compañeros que hayan realizado proyectos similares.

Riesgo: Requisitos del cliente.

Descripción del Riesgo: Es relevante ya que el desarrollo de nuestro software dependerá de lo que nos exige nuestro cliente. Una mala interpretación de los requisitos establecidos por el cliente pueden suponer un retraso en el proyecto global. El responsable será el jefe de proyecto y los desarrolladores que estuvieron en las reuniones previas con el cliente.

Tratamiento del Riesgo: Confirmación de los requisitos del software con el cliente. En caso de tener que modificar alguno de los requisitos, se contará con un control de versiones que facilitará la tarea.

Riesgo: Cambio de requisitos

Descripción del Riesgo: Tiene un nivel de exposición media porque provocaría un retraso o modificaciones del producto constantemente. Esto implicaría un cambio en el plan de trabajo del equipo de desarrollo y en el peor caso no se cumpliría con la fecha de entrega del software. El responsable de esto será el jefe de proyecto ya que aceptó los nuevos requisitos sin haber planeado correctamente el trabajo distribuido.

Tratamiento del Riesgo: Son necesarios los informes de los módulos para comprobar que no haya problemas de incompatibilidad con los nuevos requisitos. Aquí es necesario el trabajo en grupo y un seguimiento y control de lo nuevo introducido.

Riesgo: Cambios gestionados de manera incorrecta.

Descripción del Riesgo: Aquí tenemos un nivel de exposición alto. Los cambios ya sean a nivel de requisitos o de diseño surgirán de manera inevitable. Esto traerá como consecuencia complicaciones a la hora de realizar cambios o consolidarlos con el proyecto, con posibilidades de contradicciones. Si no se aplica una correcta planificación es posible desarrollar un producto con muchos defectos. En este caso el jefe de proyecto debe asumir la responsabilidad porque es el encargado de revisar los informes de los módulos periódicamente.

Tratamiento del Riesgo: El jefe de proyecto debe realizar un examen exhaustivo basado en la documentación, mejorar la comunicación entre el grupo de desarrollo, debe estudiar los

conocimientos del grupo para conocer si son competentes en sus respectivas áreas de trabajo y por último realizar un seguimiento constante.

Riesgo: Desconocimiento del entorno de trabajo

Descripción del riesgo: Problemas con el entorno de trabajo y lenguaje de programación y falta de experiencia por parte de algún integrante del proyecto en el lenguaje en el que se desarrolla la aplicación. Se origina por escasos conocimientos o insuficiente experiencia con las herramientas y lenguajes de programación que van a ser utilizados, lo que originaría pérdida de tiempo en el aprendizaje de uso de las herramientas y lenguajes de programación necesaria.

Tratamiento del riesgo: El jefe de proyecto delegará en profesionales cualificados la instalación del mismo entorno (misma versión del programa utilizado para el desarrollo de la aplicación) en todas las estaciones de trabajo, así como la exigencia a los programadores del conocimiento necesario de las herramientas de desarrollo.

Riesgo: Ausencia de algún miembro del equipo de trabajo y retraso en el calendario de planificación.

Descripción del riesgo: Algún miembro del proyecto puede sufrir contratiempos (enfermedad, accidente, etc.) y por lo tanto no podría ni realizar tareas relacionadas con el proyecto ni asistir a reuniones del equipo. Este hecho daría lugar a fallos y retrasos en la realización de su parte del proyecto y a la vez retrasaría el proyecto entero.

Tratamiento del riesgo: El jefe de proyecto tiene que tener la capacidad de reservar días considerando estos contratiempos con el fin de repartir tareas a otros miembros, esperar a la recuperación del miembro afectado o contratar a otra persona que realice su trabajo.

Planificación temporal

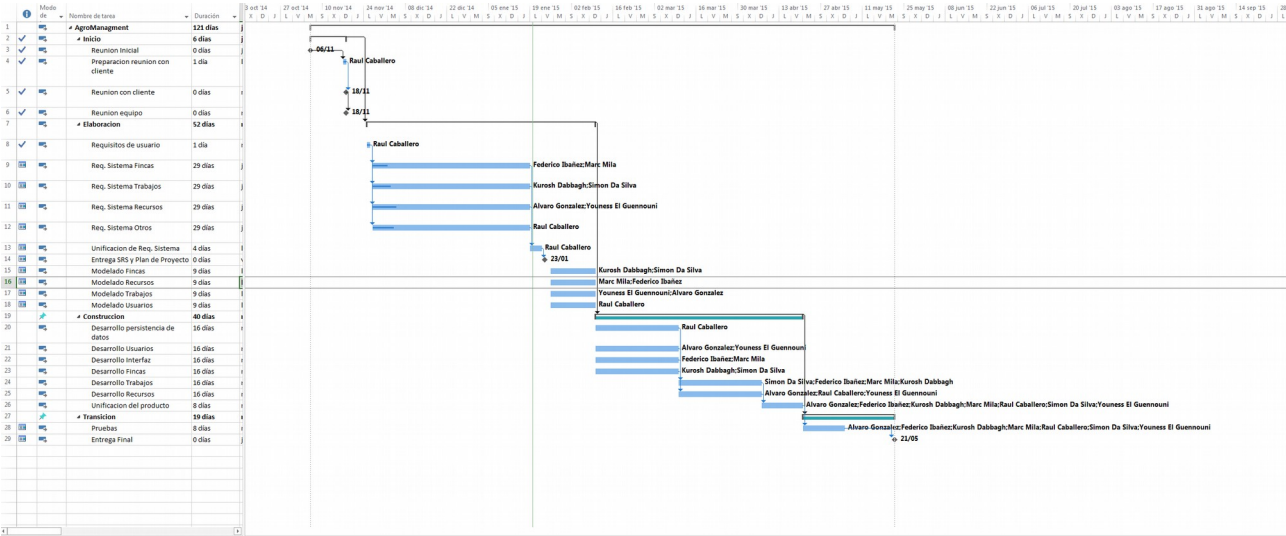
Estructura de descomposición del trabajo / Planificación temporal

A continuación se presenta la planificación temporal del proyecto realizado con Microsoft Project. Para una mejor visualización se adjunto el fichero Grupo2IS.mpp con el original.

	Modo de	Nombre de tarea	Duración	Comienzo	Fin	Predecesoras	Nombres de los recursos
1		AgroManagment	121 días	jue 06/11/14	jue 21/05/15		
2	✓	Inicio	6 días	jue 06/11/14	mar 18/11/14		
3	✓	Reunion Inicial	0 días	jue 06/11/14	jue 06/11/14		Federico Ibañez;Kurosh Dabbagh;Marc Mila;Raul Caballero;Simon Da Silva
4	✓	Preparacion reunion con cliente	1 día	lun 17/11/14	lun 17/11/14	3	Raul Caballero
5	✓	Reunion con cliente	0 días	mar 18/11/14	mar 18/11/14	4	Alvaro Gonzalez;Federico Ibañez;Kurosh Dabbagh;Marc Mila;Raul Caballero;Y
6	✓	Reunion equipo	0 días	mar 18/11/14	mar 18/11/14	5	Alvaro Gonzalez;Federico Ibañez;Kurosh Dabbagh;Marc Mila;Raul Caballero;Y
7		Elaboracion	52 días	mar 25/11/14	lun 09/02/15	2	
8	✓	Requisitos de usuario	1 día	mar 25/11/14	mar 25/11/14		Raul Caballero
9		Req. Sistema Fincas	29 días	jue 27/11/14	dom 18/01/15	8	Federico Ibañez;Marc Mila
10		Req. Sistema Trabajos	29 días	jue 27/11/14	dom 18/01/15	8	Kurosh Dabbagh;Simon Da Silva
11		Req. Sistema Recursos	29 días	jue 27/11/14	dom 18/01/15	8	Alvaro Gonzalez;Youness El Guennouni
12		Req. Sistema Otros	29 días	jue 27/11/14	dom 18/01/15	8	Raul Caballero
13		Unificacion de Req. Sistema	4 días	lun 19/01/15	jue 22/01/15	9;10;11;12	Raul Caballero
14		Entrega SRS y Plan de Proyecto	1 día	vie 23/01/15	vie 23/01/15	13	Alvaro Gonzalez;Federico Ibañez;Kurosh Dabbagh;Marc Mila;Raul Caballero;S
15		Modelado Fincas	9 días	lun 26/01/15	lun 09/02/15		Kurosh Dabbagh;Simon Da Silva
16		Modelado Recursos	9 días	lun 26/01/15	lun 09/02/15		Marc Mila;Federico Ibañez
17		Modelado Trabajos	9 días	lun 26/01/15	lun 09/02/15		Youness El Guennouni;Alvaro Gonzalez
18		Modelado Usuarios	9 días	lun 26/01/15	lun 09/02/15		Raul Caballero
19		Construccion	40 días	mar 10/02/15	lun 20/04/15	7	
20		Desarrollo persistencia de datos	16 días	mar 10/02/15	lun 09/03/15		Raul Caballero
21		Desarrollo Usuarios	16 días	mar 10/02/15	lun 09/03/15		Alvaro Gonzalez;Youness El Guennouni
22		Desarrollo Interfaz	16 días	mar 10/02/15	lun 09/03/15		Federico Ibañez;Marc Mila
23		Desarrollo Fincas	16 días	mar 10/02/15	lun 09/03/15		Kurosh Dabbagh;Simon Da Silva
24		Desarrollo Trabajos	16 días	mar 10/03/15	lun 06/04/15	22;23	Simon Da Silva;Federico Ibañez;Marc Mila;Kurosh Dabbagh
25		Desarrollo Recursos	16 días	mar 10/03/15	lun 06/04/15	20;21	Alvaro Gonzalez;Raul Caballero;Youness El Guennouni
26		Unificacion del producto	8 días	mar 07/04/15	lun 20/04/15	24;25	Alvaro Gonzalez;Federico Ibañez;Kurosh Dabbagh;Marc Mila;Raul Caballero;S
27		Transicion	19 días	mar 21/04/15	jue 21/05/15	19	
28		Pruebas	8 días	mar 21/04/15	lun 04/05/15	26	
29		Entrega Final	0 días	jue 21/05/15	jue 21/05/15	28	

Gráfico Gantt

A continuación se presenta el gráfico de Gantt del proyecto realizado con Microsoft Project. Para una mejor visualización se adjunto el fichero Grupo2IS.mpp con el original.



Red de tareas

Debido al tamaño de la imagen, se adjunto documento de Microsoft Project *ISGrupo2.mpp* con toda la información de la red de tareas.

Tabla de uso de recursos

A continuación se presenta la tabla de recursos del proyecto realizado con Microsoft Project. Para una mejor visualización se adjunto el fichero Grupo2IS.mpp con el original.

	 Nombre del recurso	 Trabajo 
1	 Raul Caballero	92 horas
2	 Marc Mila	86 horas
3	 Federico Ibañez	86 horas
4	 Alvaro Gonzalez	86 horas
5	 Simon Da Silva	86 horas
6	 Youness El Guennouni	86 horas
7	 Kurosh Dabbagh	86 horas

Recursos del proyecto

Personal

Para el desarrollo del proyecto se integrará un equipo que dispondrá de 7 estudiantes de Grado de Ingeniería de Informática.

Jefe de proyecto

Raúl Caballero Girol

Integrantes del proyecto

Álvaro González Sánchez

Federico Ibáñez Moruno

Kurosh Javier Dabbagh Escalante

Marc Mila Peramos

Simón Augusto Da Silva Álvarez

Youness El Guennouni

Hardware y software

- Portátiles particulares de los integrantes del equipo con SO Windows 7 o superior.
- Entorno de desarrollo Eclipse en su última versión.
- Microsoft Office para documentación.

Lista de recursos

Los integrantes del Grupo 2 son los recursos humanos con los que cuenta el proyecto:

- Raúl Caballero Girol
- Álvaro González Sánchez
- Federico Ibáñez Moruno
- Kurosh Javier Dabbagh Escalante
- Marc Mila Peramos
- Simón Augusto Da Silva Álvarez
- Youness El Guennouni

Organización del personal

Estructura de equipo

Se utilizará el modelo Centralizado Controlado. Debido a la poca experiencia del equipo en Ingeniería de Software, se ha decidido que este modelo puede ayudar a una organización sencilla y centralizada en la figura del Jefe de proyecto. Además, se han creado tres grupos de trabajo para una mejor separación de funciones:

Jefe de Proyecto	G. Trabajo 1	G. Trabajo 2	G. Trabajo 3
Raúl Caballero	Marc Mila Federico Ibáñez	Kurosh Javier Dabbagh Simón Augusto Da Silva	Youness El Guennoun Álvaro González

Informes de gestión

Se generarán varios tipos diferentes de informes:

- Un informe que recoja el acta de las reuniones del equipo que serán llevado a cabo los días lunes cada 2 semanas.
- Un informe que mostrará el estado actual de todas las solicitudes de cambio y será generado al final de cada 2 semanas.

Mecanismos de seguimiento y control

Garantía de calidad y control

Usaremos la garantía de calidad del software (SQA) para determinar el grado con el que un sistema, componente o proceso cumple los requerimientos especificados y las necesidades o expectativas del cliente o usuario.

Son una serie de revisiones, y pruebas utilizadas a lo largo del ciclo de desarrollo para asegurar que el producto final cumple con las SRS definidas por el equipo, y certifica el cumplimiento de los estándares de la Ingeniería del Software (IS), para ello se van a realizar una serie de Revisiones Técnicas Formales (RTF) para llevar un control de los siguientes puntos:

- Detectar riesgos potenciales → Al utilizar una estrategia proactiva, es decir, la detección de errores comienza antes que los trabajos técnicos, se utilizarán los mecanismos de detección y control de riesgos potenciales anteriormente especificados.
- Detectar problemas de calidad → La detección de problemas de calidad se refiere a validar los procesos usados para crear el software final orientado a proporcionar confianza el producto. Este control de la calidad es llevado a cabo por todos los constitutivos diferentes que tengan que ver con el proyecto:
 1. El grupo de desarrolladores del software.
 2. Un grupo de SQA (Software Quality Assurance) que se ocupa de la planificación, supervisión y mantenimiento de los registros, análisis e informes del control de calidad.
 3. Jefe de proyecto.
 4. Clientes, no se incluirá en el proceso de inspección técnica del producto pero deben dar el visto bueno a los avances en el aspecto de calidad durante las reuniones periódicas con los representantes del proyecto antes de poder continuar con su realización.
- Reducir los tiempos de desarrollo y prueba → Para ello es necesario reducir el número de ciclos de pruebas necesarios. Para una correcta reducción de los tiempos de desarrollo y prueba el código debe ser limpio, estructurado, documentado y comentado, lo que facilitaría la realización de futuros cambios por los posibles errores detectados en los dichos ciclos de prueba del producto o por cualquier otro motivo ya sea por petición del cliente o por decisiones técnicas.
- Control de los distintos ciclos de desarrollo → Para lograr el correcto control de los ciclos del desarrollo es necesario especificar unos estándares explícitamente documentados de cada ciclo del desarrollo del software así como la exigencia del cumplimiento de los plazos estipulados para cada uno de estos ciclos del desarrollo y por lo tanto evitar retrasos en la realización del proyecto.

Los RTFs promueven la seguridad y la continuidad, ya que todos los participantes del proyecto estarán familiarizados con partes del proyecto que no entran en su ámbito de realización. Se realizarán cada vez que esté por evaluar un elemento de configuración software y tendrán en cuenta tanto funcionalidad, como calidad del código escrito, dentro del plazo estipulado para cada subtarea.

Se deben centrar en una parte específica y pequeña del software total, debe desarrollarse una lista de comprobación por cada fracción del proyecto que vaya a ser revisada. Se enunciarán áreas de problemas, pero no se intentará resolver cualquier problema que se ponga de manifiesto. El objetivo de una RTF no es la resolución de problemas, se debe dejar eso al productor o productores de la parte del trabajo en la que haya que realizar la modificación. Es necesaria la correcta documentación de estas RTFs para que pueda ser comprobada por los demás revisores, así como cualquier integrante del trabajo que desee cerciorarse de la correcta corrección del problema.

Al final de la exposición de la parte a revisar se fijará una fecha de entrega para poder realizar los cambios pertinentes por el productor o productores antes de la continuación con una nueva tarea.

Gestión y control de cambios

La gestión y control del cambio apoya directamente al desarrollo y mantenimiento del software mediante la conservación de la integridad del producto identificando el cambio, controlándolo y garantizando que el cambio quede bien implantado.

Un cambio puede venir por una modificación en los requerimientos de los clientes, o el equipo de desarrollo desea modificar el enfoque técnico, en cuyo caso se deberá contar con la aprobación del cliente para llevar a cabo dicho cambio, cada cambio será registrado formalmente.

Cuando se quiera llevar a cabo un cambio se realizará una petición de cambio para corregir errores o deficiencias, adaptar un nuevo entorno será sometido al análisis de todos los implicados en el software, dicha petición será expuesta en una RTF.

Si se aprueba dicha petición de cambio se categorizará y se dará prioridad según la repercusión que tengan en el proyecto. Se tendrá una copia de la documentación actualizada de cada una de las reuniones de las RTFs donde se mantendrá una tabla de control de cambios que indicará el nombre del proyecto, la versión, la fecha de la RTF, así como el nombre de la parte a revisar, el productor o productores y una breve descripción de la naturaleza del cambio, también se incluirá la fecha máxima para la realización de los cambios para solucionar el problema.

Una vez realizado el cambio será vuelto a presentar el proyecto para que se verifiquen que todos los cambios han sido realizados tal y como se había acordado, así como que no se hayan producido ningún nuevo error a partir de dicho cambio.

Una vez revisado y aprobado el cambio realizado en el producto se convierte en una línea base, que servirá como base para un desarrollo posterior y que puede ser cambiado únicamente mediante procedimientos formales de control de cambio.

Actualmente, se está a la espera de la entrega de repositorios de Subversion por parte de la Facultad de Informática de la Universidad Complutense para el control de versiones. Una vez que se facilite se estudiará la viabilidad de la solución. En caso de que no cumpla las expectativas del equipo de desarrollo (acceso remoto desde fuera de la universidad, espacio suficiente, etc...) se estudiarán otras opción como, por ejemplo, GitHub.