
Plan de gestión del proyecto Software

MiPandilla

Versión <1.0> <Planteamiento>

**Alejandro Bermejo Vialas, Mariadalit Rosales Rodríguez, Mariano
González Salazar, Jorge Peláez González, Saad Ziyad Khalid.**

Universidad Carlos III de Madrid

<08-10-2015>

MiPandilla	Versión: <1.0>
Plan de gestión del proyecto Software	Fecha: <2015-10-08>

Tabla de contenido

1. Introducción.....	3
1.1. Resumen del proyecto	3
1.1.1. Propósito, alcance y objetivos	3
1.1.2. Supuestos y limitaciones.....	3
1.1.3. Resumen de entregables	4
1.1.4. Resumen de planificación y recursos	5
1.1.5. Definiciones	6
2. Organización del proyecto	77
2.1. Roles y responsabilidades	7
2.2. Reuniones	8
3. Plan de gestión del proceso de desarrollo	9
3.1. Plan de trabajo	9
3.1.1. Definición de paquetes de trabajo	9
3.1.2. Planificación temporal.....	9
3.1.3. Asignación de recursos	10
3.1.4. Entregas.....	11
3.2. Plan de gestión de riesgos	12
4. Plan técnico del proceso de desarrollo.....	12
4.1. Modelo de proceso	12
4.2. Métodos, herramientas y técnicas	13
Anexos	14

MiPandilla	Versión: <1.0>
Plan de gestión del proyecto Software	Fecha: <2015-10-08>

1. Introducción

A continuación se mostrará una descripción general detallada del sistema junto con los diagramas necesarios para facilitar su comprensión.

1.1 Resumen del proyecto

1.1.1 Propósito, alcance y objetivos

Hoy en día el Software Social, que reúne personas según sus gustos vía Internet, está en pleno auge. Por ello EMPRESA CLIENTE pretende crear una red social basada en eventos categorizados de tal forma que usuarios de características afines sean capaces de conocerse. Para lograr tal objetivo, contrata a la empresa TeLEDmíticos, expertos en desarrollo de software web.

El sistema deseado consiste en una plataforma tipo red social, en el que los usuarios registrados pueden crear eventos para conocer gente. Además deben poder indicar a qué eventos acudirán, o comentar en ellos. Además dichos usuarios registrados de dicha red podrán seguir a otros, sin reciprocidad alguna ni necesidad de confirmación (un usuario puede seguir a otro para ver a qué eventos acudirá, pero no necesariamente debe ser mutuo).

Por otra parte, los usuarios no registrados deben poder ver y buscar eventos, pero para apuntarse o comentar deben estar registrados.

EMPRESA CLIENTE busca obtener beneficio de esta aplicación mediante los Eventos Patrocinados, por los cuales otras compañías pueden pagar para que el sistema priorice determinados eventos en sus resultados.

Para conseguir cumplir con las expectativas de EMPRESA CLIENTE, TeLEDmíticos deberá ir entregando sus progresos en diversos plazos. En estas entregas se incluirá tanto documentación como código, para que la EMPRESA CLIENTE pueda rectificar algo si no está satisfecha.

Internamente, TeLEDmíticos trabajará con una metodología Ágil basada en 2 reuniones semanales cortas y concisas, y entregas pequeñas para asegurar el cumplimiento de los plazos. Por ello, cada miembro tiene claro su rol y está completamente dispuesto a trabajar en equipo.

1.1.2 Supuestos y limitaciones

El método de trabajo comienza con una exhaustiva planificación y descripción del proyecto y el método de trabajo. Se deciden con el cliente ciertas entregas parciales del resultado final, para demostrar el camino que se está siguiendo. La

MiPandilla	Versión: <1.0>
Plan de gestión del proyecto Software	Fecha: <2015-10-08>

fecha final para la entrega del proyecto está fijada para el 10 de diciembre, debiendo exponerlo el día siguiente.

A la hora de programar, TeLEDmíticos estará sujeto a ciertas limitaciones. En primer lugar, el código debe ser escrito completamente en Java, no pudiendo emplear lenguajes adicionales, aunque sí llamadas al sistema. Además no serán necesarios plug-ins ni para compilar ni ejecutar el código, únicamente deberá ser necesario java 1.7 o superior. Éste debe ser posible de compilar sin ningún tipo de entorno de programación, aunque para su propia depuración será permitido el programa Eclipse.

En segundo lugar, podrán incluirse librerías y/o porciones de código, siempre y cuando éste sea libre y gratuito. Deberá entregarse al finalizar el proyecto una documentación detallada sobre dicho código de 3ª persona, que deberá incluir la fuente de la que se obtuvo y el autor.

Debido a limitaciones en las llamadas del sistema, este programa sólo será compatible bajo plataformas Linux basadas en Debian (como pudiera ser Ubuntu), pudiendo la EMPRESA CLIENTE solicitar su portabilidad a otros sistemas.

Por último, la EMPRESA CLIENTE adquirirá el programa MiPandilla y su código fuente, pudiendo distribuirlo bajo la apropiada licencia Creative Commons, de forma gratuita y en formato ejecutable, de tal manera que el código no sea modificable por terceros. Una vez cerrado el proyecto, la EMPRESA CLIENTE tiene el plazo de una semana para pedir responsabilidad por bugs o fallos de seguridad, no pudiendo exigir modificaciones no justificadas. Pasado el plazo, TeLEDmíticos no se hace responsable de corregir o modificar en ningún caso el programa.

1.1.3 Resumen de entregables

El día **8 de octubre** se hará entrega a la EMPRESA CLIENTE de los documentos ERS y SPMP de FORMA PROVISIONAL, en los cuales se recoge una descripción detallada del sistema, incluyendo requisitos funcionales y no funcionales, así como los modelos de usuario y los diagramas de casos de uso. Además se hará entrega de las actas de las reuniones que procedan.

El día **29 de octubre** se hará entrega de las actas de las reuniones que procedan.

MiPandilla	Versión: <1.0>
Plan de gestión del proyecto Software	Fecha: <2015-10-08>

El día **12 de noviembre** se hará entrega de un diseño completo y detallado del sistema, el cual incluirá los diagramas UML. De forma adicional, se le permitirá probar al cliente una pequeña muestra del programa. Además se hará entrega de las actas de las reuniones que procedan.

El día **3 de diciembre** se hará entrega de las actas de las reuniones que procedan.

El día **10 de diciembre** se hará entrega del código fuente definitivo del sistema, debidamente comentado internamente y documentado por la herramienta javadoc, y la descripción de los recursos asociados al mismo. También se presentará un plan de pruebas software. Además se hará entrega de las actas de las reuniones que procedan.

El día **11 de diciembre** se hará entrega de los documentos ERS y SPMP definitivos, así como del resto de la documentación de acuerdo con las modificaciones exigidas por la EMPRESA CLIENTE. Además se realizará una defensa del proyecto de forma oral de estimadamente 15 minutos.

1.1.4 Resumen de planificación y recursos

El recurso principal son los recursos humanos de los que disponemos, es decir, los conocimientos de los miembros del equipo: Jorge Peláez (jefe), Mariano González (gestor de desarrollo), Saad Ziyad (gestor de calidad), Alejandro Bermejo (gestor de soporte) y Mariadalit Rosales (planificadora). El equipo se reunirá tres veces por semana hasta la primera entrega: lunes, viernes y domingos. La reunión de los lunes se basará en comentar la aprobación o desaprobación sobre el trabajo de cada miembro del equipo por parte del Jefe. La reunión de los viernes es necesaria para comentar dudas o avances sobre el proyecto. Las reuniones de los domingos se realizarán para presentar el trabajo correspondiente al Jefe y hacer algún comentario sobre el mismo si fuera necesario. Los lunes y domingos las reuniones se realizarán en remoto.

Todas las reuniones durarán un máximo de 15 minutos, exceptuando la primera de ellas (lunes 21 de septiembre de 2015) cuya duración es de 1 hora debido a la necesidad de explicar con detalle el proyecto y su planificación.

A partir de la primera entrega las reuniones se realizarán eventualmente dependiendo de la disponibilidad de los miembros aunque se mantendrá fijo el domingo como día de entrega del trabajo al Jefe, aunque no necesariamente se lleve a cabo una reunión. La división de tareas varía según el número de semanas disponibles hasta la fecha de entrega (a saber 8 de octubre, 11 de noviembre, 10 y 11 de diciembre). Todos los domingos anteriores a estas fechas cada miembro debe tener todas sus tareas finalizadas para que el jefe le dé su visto bueno antes de la entrega.

MiPandilla	Versión: <1.0>
Plan de gestión del proyecto Software	Fecha: <2015-10-08>

Por comodidad, existe un calendario en Google Calendar donde cada miembro puede ver la planificación por semana con detalle, junto con un foro en el que eventualmente la planificadora comentará fechas de entregas, reuniones o modificaciones sobre las mismas. También existe un documento en Google Docs donde se especifican las labores a realizar por cada miembro de manera detallada para la primera entrega y un grupo en Dropbox donde los miembros pueden subir sus tareas finalizadas o compartir el código del proyecto con mayor comodidad.

1.1.5 Definiciones

Caso de uso: es una secuencia de interacciones que se desarrollarán entre un sistema y sus actores en respuesta a un evento que inicia un actor principal sobre el propio sistema.

Eclipse: es un programa informático compuesto por un conjunto de herramientas de programación de código abierto multiplataforma, que en este caso utilizaremos para escribir y depurar nuestro código Java.

EMPRESA CLIENTE: Empresa contratante, decide qué quiere y cómo lo quiere.

Entregable: Porción del proyecto a entregar al cliente para que compruebe los progresos realizados y pueda rectificar o modificar sus condiciones en caso de creerlo necesario.

ERS: La especificación de requisitos de software (ERS) es una descripción completa del comportamiento del sistema que se va a desarrollar. Incluye un conjunto de casos de uso y de requisitos funcionales y no funcionales.

Gantt (Diagrama de Gantt): es un gráfico cuyo objetivo es exponer el tiempo de dedicación previsto para diferentes tareas o actividades a lo largo de un tiempo total determinado.

Java: Lenguaje de programación multiplataforma propiedad de Oracle. Este programa, a pesar de Java ser multiplataforma, estará limitado ya que contendrá llamadas al Sistema Operativo.

Javadoc: Herramienta utilizada para la generación de documentación de un código basado en ciertos comentarios del programador.

MiPandilla	Versión: <1.0>
Plan de gestión del proyecto Software	Fecha: <2015-10-08>

Minuta (de reunión): Archivo en el que se puede apreciar lo comentado en cada reunión, así como el progreso del equipo en ese momento y el reparto de tareas.

MiPandilla: Original nombre del proyecto a realizar.

Paquete de trabajo: Porción del proyecto asignado a un miembro del grupo, de pequeño tamaño y con su propia fecha de entrega interna. El número o dimensiones de un paquete de trabajo son independientes de un entregable.

Plataforma Linux basada en Debian: Hace referencia al sistema operativo que debe ejecutar la máquina en la que se quiere ejecutar MiPandilla.

Requisitos funcionales: Servicios que el sistema debe proporcionar

Requisitos no funcionales: Restricciones que afectaran al sistema

SPMP: Plan de gestión del proyecto, contiene la planificación y descripción del proyecto hasta el mínimo detalle.

SRS: Véase ERS.

TeLEDmíticos: Realiza un proyecto basado en las especificaciones y requerimientos de la EMPRESA CLIENTE.

2. Organización del proyecto

2.1 Roles y responsabilidades

El equipo consta de 5 miembros: Jorge Peláez González (jefe de equipo), Alejandro Bermejo Vialas (gestor de soporte), Saad Ziyad Khalid (gestor de calidad), Mariano González Salazar (desarrollador) y Mariadalit Rosales Rodríguez (planificadora).

El jefe de equipo es el responsable principal sobre el proyecto por lo que debe asegurarse de que cada miembro realice su función correctamente. Estará en contacto continuamente con el resto de miembros del equipo para dar el visto bueno al trabajo de éstos y resolver dudas si fuera necesario.

El gestor de soporte es el encargado de buscar y aportar información útil para usar durante el desarrollo del código, como por ejemplo, librerías. Por ello, debe estar

MiPandilla	Versión: <1.0>
Plan de gestión del proyecto Software	Fecha: <2015-10-08>

en contacto directo con el desarrollador, para proporcionar apoyo a la labor de éste y asegurar el buen desarrollo del código.

El gestor de calidad se encargará de corregir los errores que puedan aparecer en el código del proyecto, asegurando una buena calidad al mismo. Estará en contacto con todos los miembros con el fin de corregir los diferentes errores que cada uno pueda cometer y con el jefe en vista de que le dé su visto bueno final.

El desarrollador es el responsable directo sobre la evolución del código del proyecto. El resto de miembros deberán estar en contacto con él para no abandonar o desviarse de la idea principal que haya presentado éste ya que colaborarán en la programación del código.

La planificadora se encargará de organizar las fechas de las reuniones y las fechas para las cuales las diversas tareas deben estar finalizadas. Por ello, estará en contacto con el resto de miembros para recordar estas fechas, resolver dudas o posibles modificaciones sobre las mismas.

2.2 Reuniones

Se ha decidido realizar tres reuniones por semana: reuniones de revisión (lunes), reuniones técnicas (viernes) y reuniones de progreso (domingo) hasta la fecha 8 de octubre de 2015, en la que se realizará la primera entrega.

A partir de dicha fecha las reuniones se han realizado eventualmente dependiendo de la disponibilidad de los miembros. Concretamente, todas las reuniones que se han realizado son:

Reunión	Fecha	Resumen de la reunión
Reunión 1	21 de septiembre de 2015	Reparto de tareas para los documentos SPMP y ERS
Reunión 2	25 de septiembre de 2015	Discutir algunas adiciones al proyecto y el reparto de trabajo
Reunión 3	28 de septiembre de 2015	Revisión y comentarios del trabajo
Reunión 4	5 de octubre de 2015	Revisión y comentar el trabajo realizado
Reunión 5	16 de octubre de 2015	Repartir las tareas del documento DesignedSoftware

MiPandilla	Versión: <1.0>
Plan de gestión del proyecto Software	Fecha: <2015-10-08>

Reunión 6	6 de noviembre de 2015	Modificar tareas del documento Designed Software
Reunión 7	9 de noviembre de 2015	Discutir las tareas y resolver dudas
Reunión 8	5 de diciembre de 2015	Discutir el desarrollo del proyecto y la defensa del mismo

3. Plan de gestión del proceso de desarrollo

3.1 Plan de trabajo

3.1.1 Definición de paquetes de trabajo

Los paquetes de trabajo que hemos definido son:

- Paquete 1: Aprobación del proyecto: en este paquete definiremos los roles de cada miembro del equipo y las tareas que se llevarán a cabo.
- Paquete 2: Organización: en este paquete se define el reparto de tareas y las fechas de finalización y entrega de dichas tareas.
- Paquete 3: Realización de documentos: este paquete engloba todas las tareas sobre la escritura de los documentos: SPMP, ERS y Diseño del Software.
- Paquete 4: Diseño: engloba las tareas relacionadas con el desarrollo del diseño de la estructura del proyecto.
- Paquete 5: Desarrollo del código: en este paquete se fijan las distintas tareas divididas entre todos los miembros del equipo en relación con el desarrollo del código del proyecto.
- Paquete 6: Fase de pruebas de errores: este paquete engloba las distintas tareas relacionadas con el desarrollo del plan de pruebas de software y revisión del mismo.
- Paquete 7: Administración: engloba todas las tareas de supervisión de subida del trabajo correspondiente de cada miembro: minutas, documentos y códigos, junto con las reuniones del equipo y la supervisión continua del proyecto.
- Paquete 8: Presentación: engloba las tareas en relación a la defensa del proyecto.

3.1.2 Planificación temporal

Paquete 1: Aprobación del proyecto:

- ❖ Tarea1: asignación de roles y sus tareas.

Paquete 2: Organización:

- ❖ Tarea1: Reparto de tareas con respecto los documentos SPMP, ERS y Designed Software.
- ❖ Tarea2: Establecimiento de fechas de finalización de dichas tareas.

MiPandilla	Versión: <1.0>
Plan de gestión del proyecto Software	Fecha: <2015-10-08>

Paquete 3: Realización de documentos:

- ❖ Tarea1: escritura de los documentos SPMP y ERS.
- ❖ Tarea2: escritura del documento de Diseño del Software.

Paquete 4: Diseño:

- ❖ Tarea1: diseñar la estructura del proyecto.
- ❖ Tarea2: revisar dicha estructura y modificar en caso de que sea necesario.

Paquete 5: Desarrollo del código:

- ❖ Tarea1: decidir las herramientas a utilizar.
- ❖ Tarea2: desarrollo del código.
- ❖ Tarea3: comentar dicho código.

Paquete 6: Fase de pruebas de errores:

- ❖ Tarea1: desarrollo del plan de pruebas.
- ❖ Tarea2: revisión del plan de pruebas.

Paquete 7: Administración:

- ❖ Tarea1: supervisión de las tareas.
- ❖ Tarea2: control de la subida de actas y reportes.
- ❖ Tarea3: control sobre las reuniones.

Paquete 8: Presentación:

- ❖ Tarea1: preparación de presentación.
- ❖ Tarea2: defensa del proyecto.

El gráfico correspondiente a este apartado se puede observar en el anexo “Planificación Temporal.png”.

3.1.3 Asignación de recursos

- Hardware: cada miembro dispondrá de un ordenador individual en el que pueda realizar sus tareas.
- Software: Todos los ordenadores tendrán acceso a Internet y las aplicaciones necesarias para el desarrollo del código, como el programa Eclipse o las librerías necesarias.
- Recursos humanos: debido a los conocimientos previos sobre el desarrollo de programas en java, se asigna a cada miembro del equipo una porción del código fuente, es decir, éste se dividirá en pequeños paquetes de trabajo semanales en el que las tareas serán repartidas equitativamente entre los 5 miembros del equipo. A su vez, debido a los conocimientos que presenta el desarrollador sobre diseño gráfico y programación en java, se asigna a éste el diseño del sistema junto con los diagramas necesarios. Por último, todos los miembros presentan conocimientos de ofimática y altas cualidades expresivas, por lo que el

MiPandilla	Versión: <1.0>
Plan de gestión del proyecto Software	Fecha: <2015-10-08>

desarrollador deberá hacer frente al reporte del documento ERS, todo el equipo excepto el desarrollador llevarán a cabo el reporte del documento SPMP y el gestor de calidad rellenará el documento adquirido para las minutas de las reuniones.

La asignación de estos recursos permanecerá inmóvil desde el comienzo del proyecto (viernes 18 de septiembre de 2015) hasta la finalización del desarrollo del mismo (lunes 7 de diciembre).

3.1.4 Entregas

Identificador	Tipo	Título	Fecha
D1	Documento	SPMP	08/10/2015
D2	Documento	ERS	08/10/2015
D3	Documento	Minutas de las reuniones	08/10/2015
D4	Documento	Minutas de las reuniones	29/10/2015
D5	Documento	Minutas de las reuniones	12/11/2015
D6	Documento	Diseño del sistema	12/11/2015
D7	Documento	Minutas de las reuniones	03/12/2015
C1	Código	Código fuente final comentado	10/12/2015
D8	Documento	Plan de pruebas de software	10/12/2015
D9	Documento	Minutas de las reuniones	10/12/2015
D10	Documento	ERS final	11/12/2015
D11	Documento	Diseño y documentación del código final	11/12/2015
P1	Presentación	Presentación final del proyecto	11/12/2015

MiPandilla	Versión: <1.0>
Plan de gestión del proyecto Software	Fecha: <2015-10-08>

3.2 Plan de gestión de riesgos

	Descripción	Probabilidad	Cuando puede ocurrir	Efecto	Cómo afrontarlo
Personal	El desarrollador se ponga enfermo	baja	todo el cuatrimestre	serio	Cada miembro tendrá una planificación de tareas específicas en un periodo de 2 semanas
	El equipo esté desmotivado	media	después de una entrega de notas	serio	Se realizará una reunión motivadora posterior a la entrega de notas en la misma semana
Seguridad	Algún miembro revela información confidencial	baja	todo el cuatrimestre	catastrófico	El trabajo que releva dicha información será directamente expulsado de la empresa
	Alguien ajeno a la empresa roba la contraseña de algún ordenador y modifica el proyecto	baja	todo el cuatrimestre	catastrófico	Los miembros del equipo deberán modificar su contraseña cada mes
Hardware	No tener suficientes ordenadores	bajo	todo el cuatrimestre	serio	Agrupar a los desarrolladores en grupos de dos y que realicen distintas tareas según el día
	Fallo del sistema	bajo	todo el cuatrimestre	catastrófico	Cada trabajador realizará copias de seguridad del trabajo a diario y lo subirá a la nube. Los trabajadores se reagruparán según el número de ordenadores disponibles hasta que los rotos se hayan sustituidos por unos nuevos
Software	No hay acceso a las herramientas necesarias	baja	todo el cuatrimestre	serio	El gestor de soporte proporcionará las herramientas necesarias para el
	Los trabajadores no saben usar las herramientas	media	todo el cuatrimestre	serio	El gestor de soporte proporcionará directrices o tutoriales para solventarlo
	Fallo del sistema	bajo	todo el cuatrimestre	catastrófico	Cada trabajador realizará copias de seguridad del trabajo a diario y lo subirá a la nube
Organización	Alguno de los miembros olvide una reunión o entrega	media	todo el cuatrimestre	serio	El planificador deberá recordar al resto de miembros las reuniones/entregas el día anterior y el mismo día
	Dos o más miembros realicen la misma labor	baja	todo el cuatrimestre	serio	El jefe decidirá y seleccionará la mejor de las dos. Se dará un plazo de 1 día a aquel que se equivocara de tarea para que realice la que le corresponde

4. Plan técnico del proceso de desarrollo

4.1 Modelo de proceso

El modelo de ciclo de vida que se seguirá en el proceso de desarrollo del proyecto será en cascada, dado que el cliente nos define los requisitos mínimos de la aplicación desde un principio y se puede afrontar la complejidad del proyecto de una forma ordenada.

Primero definimos bien los requisitos que especifica el cliente y luego se diseña cómo será el sistema, para que cuando se lleve a cabo la implementación del mismo, se puedan solventar los posibles problemas que aparezcan de una forma rápida y eficaz. Una vez diseñado, se implementará el sistema y se verificará que los requisitos que nos especificó el cliente están bien realizados y se cumplen todos ellos.

MiPandilla	Versión: <1.0>
Plan de gestión del proyecto Software	Fecha: <2015-10-08>

4.2 Métodos, herramientas y técnicas

Para llevar a cabo la implementación de la aplicación y las funcionalidades de la misma, se descompondrá en partes para facilitar la labor del desarrollador/es. De esta manera, podrá identificar los métodos que serán necesarios a la hora de implementar las diferentes funcionalidades. Para ello se realizará un diagrama de casos de uso utilizando herramientas como Papyrus UML.

La aplicación será desarrollada en lenguaje Java utilizando Eclipse como entorno de desarrollo o editores de código y compilando/ejecutando mediante línea de comandos. Para ello, necesitaremos tener instalado Java (JRE 1.8.0_60-b27).

Por último, la aplicación se testeará para comprobar que todos los requisitos que propuso la empresa contratante han sido llevados a cabo y funcionan correctamente.

El sumario de todas la herramientas para el desarrollo del proyecto es el siguiente:

- Linux (SO)
- Word (Procesador de texto de los documentos)
- Umodel / OmniGraffle (Diseño de los diagramas)
- Eclipse (Entorno de desarrollo)
- Eclipse (Pruebas JUnit)
- Java (Lenguaje de programación)

MiPandilla	Versión: <1.0>
Plan de gestión del proyecto Software	Fecha: <2015-10-08>

Anexos

A.1 Archivos adicionales:

Para aportar mayor claridad, junto al documento SPMP se adjuntarán dos pdf, uno que mostrará la tabla del plan de riesgos llamado “Plan_de_riesgos.pdf”, y otro que mostrará el diagrama de Gantt llamado “Planificacion_Temporal.png”. Además se adjuntará un archivo llamado “diagramaUML.jpg” que contiene el diagrama de casos de uso del sistema.