

SIGESTPROSO

Plan de desarrollo software

Grupo 1

Patricia Cadenas Quijano
Javier García Tomillo
Mario Muñoz Sanz
Mario Santos Serrano

Historial de revisiones del documento

<i>Fecha</i>	<i>Versión</i>	<i>Descripción</i>	<i>Autor/es</i>
21/11/2010	0.9.1	Versión preliminar	Grupo 01
23/11/2010	0.9.2	Versión propuesta	Grupo 01
27/11/2010	1.0	Versión entregada	Grupo 01

ÍNDICE

HISTORIAL DE REVISIONES DEL DOCUMENTO	2
ÍNDICE	3
ÍNDICE DE FIGURAS	4
ÍNDICE DE TABLAS	4
1. INTRODUCCIÓN	5
1.1. PROPÓSITO	5
1.2. ALCANCE	5
1.3. DEFINICIONES, ACRÓNIMOS Y ABREVIATURAS	5
1.4. REFERENCIAS	5
1.5. RESUMEN	5
2. VISTA GENERAL DEL PROYECTO	7
2.1. PROPÓSITO, ALCANCE Y OBJETIVOS	7
2.2. SUPOSICIONES Y RESTRICCIONES	7
2.3. ENTREGABLES DEL PROYECTO	8
2.4. EVOLUCIÓN DEL PLAN DE DESARROLLO DEL SOFTWARE	9
3. ORGANIZACIÓN DEL PROYECTO	10
3.1. ESTRUCTURA ORGANIZATIVA	10
3.2. INTERFACES EXTERNAS	10
3.3. ROLES Y RESPONSABILIDADES	11
4. GESTIÓN DEL PROYECTO	12
4.1. RESUMEN DE LA PLANIFICACIÓN	12
4.2. PLAN DE PROYECTO	12
4.2.1. PLAN DE FASES	12
4.2.2. OBJETIVOS DE LAS ITERACIONES	13
4.2.3. PLAN DE ITERACIÓN	13
4.2.4. CALENDARIO DEL PROYECTO	13
4.2.5. RECURSOS DEL PROYECTO	14
4.3. MONITORIZACIÓN Y CONTROL DEL PROYECTO	15
4.3.1. GESTIÓN DE REQUISITOS	15
4.3.2. CONTROL DE CALIDAD	15
4.3.3. GESTIÓN DE RIESGOS	15
4.2.8.1. Lista de riesgos	16
4.3.4. GESTIÓN DE CONFIGURACIONES	19
ANEXO: GLOSARIO DEL PROYECTO	22
1. DEFINICIONES	22
2. ACRÓNIMOS	23
ANEXO: REFERENCIAS	24

Índice de figuras

FIGURA 1	10
FIGURA 2	21

Índice de tablas

TABLA 1	13
TABLA 2	15

1. Introducción

1.1. Propósito

El propósito del Plan de Desarrollo del Software es proporcionar la información necesaria para controlar el proyecto, en este caso SIGESTPROSO, "Seguimiento Integrado de la GESTión Temporal de PROyectos de Software". En él se describe el enfoque de desarrollo del software.

Los usuarios del Plan de Desarrollo del Software son:

- El jefe del proyecto lo utiliza para organizar la agenda y necesidades de recursos, así como para realizar su seguimiento.
- Los miembros del equipo de desarrollo lo utilizan para entender qué deben hacer, cuándo y qué otras actividades dependen de dicho trabajo.

1.2. Alcance

El Plan de Desarrollo del Software describe el plan global que será utilizado para el desarrollo de la herramienta *SIGESTPROSO*.

Los detalles de la primera iteración se encuentran descritos en el documento Plan de iteración: Inicio. El resto de iteraciones serán definidas a posteriori, ya que una de las actividades de cada plan de la iteración en curso es la elaboración del plan para la iteración siguiente, como marca el Proceso Unificado.

El plan de desarrollo descrito en este documento se basa en el pliego de condiciones o enunciado proporcionado, además de en una versión preliminar de los requisitos obtenida a partir del mismo. De esta manera se ha obtenido una estimación aproximada.

Los artefactos de Visión y Especificación de Requisitos Software (SRS) serán desarrollados con posterioridad a este documento (durante la fase de Inicio), y en ellos se definirán de forma más específica las características de la herramienta a desarrollar. Este hecho, así como el seguimiento de cada una de las iteraciones, puede desembocar en cambios en el presente Plan de Desarrollo, produciendo así nuevas versiones actualizadas.

1.3. Definiciones, Acrónimos y Abreviaturas

Ver el anexo *Glosario del proyecto*.

1.4. Referencias

Ver el anexo *Referencias*.

1.5. Resumen

Después de esta introducción, el resto del documento está organizado en las siguientes secciones:

- Vista General del Proyecto — proporciona una descripción del propósito, alcance y objetivos del proyecto, estableciendo los artefactos que serán producidos y utilizados durante el mismo.
- Organización del Proyecto — describe la estructura organizacional del equipo de desarrollo.
- Gestión del Proceso — explica los costos y planificación estimada, define las fases e hitos del proyecto y describe cómo se realizará su seguimiento.

2. Vista general del proyecto

2.1. Propósito, alcance y objetivos

SIGESTPROSO, o “Seguimiento Integrado de la GESTión Temporal de PROyectos de Software”, debe ser una aplicación que facilite a los gestores de proyectos software realizar un seguimiento temporal del trabajo que se realiza, durante el proceso de desarrollo, por parte de los trabajadores implicados en las diferentes actividades ejecutadas.

Se trata de una herramienta web orientada a ofrecer distintos servicios específicos a según qué tipo de usuario, atendiendo al rol que desempeñe en el proyecto. También permitirá la generación de informes relativos a los proyectos en curso, así como resúmenes (más o menos detallados en función del rol) de los proyectos cerrados.

Esta funcionalidad presenta otro de los objetivos del sistema: que los gestores de proyectos utilicen los datos temporales de los proyectos ya finalizados para ayudar a planificar y presupuestar los nuevos y para analizar las prácticas de trabajo seguidas con el fin de servir de ayuda en cualquier mejora del proceso de desarrollo software.

2.2. Suposiciones y restricciones

Las suposiciones y restricciones del sistema consideradas, condicionantes del desarrollo del proyecto, son:

- *Duración del proyecto.* La fecha de inicio del proyecto es el 8/11/2010, mientras que la de finalización deberá ser, como máximo, el 1/02/2011 (fecha del examen de la asignatura Ingeniería del Software II).
- *Recursos humanos y presupuesto.* El equipo de trabajo está formado por 4 personas, cada una de las cuáles dedicará de media semanalmente 10 horas al desarrollo del proyecto, repartidas de forma flexible según la disponibilidad horaria de cada miembro en cada período de tiempo. A todos los efectos, no se considerará que exista ninguna semana de vacaciones durante el período de desarrollo. Para más información, consultar apartado 4.2.4, Calendario del proyecto.
- *Recursos hardware y software.* Se dispondrá de los recursos de tipo hardware y software definidos en el apartado 4.2.5, Recursos del proyecto.
- *Restricciones tecnológicas.*
 - Se tratará de una herramienta web.
 - La herramienta diseñada deberá correr en el servidor UVa Jair sin necesidad de instalar ningún servidor o contenedor web adicional. Se decide, por tanto, utilizar PHP 5.3 para el desarrollo.
 - El Sistema Gestor de Bases de Datos a utilizar será MySQL.
- *Adiestramiento.* Una parte del tiempo asignado al desarrollo del proyecto estará destinado a la familiarización de aquéllos miembros del equipo que no han tenido un contacto previo con alguna de las tecnologías a utilizar.
- *Estructura organizativa.* La configuración del organigrama seguido se especifica en la sección 3.1, Estructura organizativa.

- *Documentación.* La documentación se elaborará siguiendo los estándares definidos en el Proceso Unificado (UP) y la IEEE, así como el modelo de proceso UPEDU.
- *Control de riesgos.* Se realizará un seguimiento y análisis de los riesgos durante el desarrollo.
- *Portabilidad.* La herramienta desarrollada podrá ser accedida a través de diferentes navegadores web y desde diferentes sistemas operativos.

2.3. Entregables del proyecto

Es preciso destacar que, de acuerdo a la filosofía del Proceso Unificado (y de todo proceso iterativo e incremental), todos los artefactos son objeto de modificaciones a lo largo del proceso de desarrollo, con lo cual, sólo a su término podremos disponer de una versión definitiva de cada uno de ellos. Sin embargo, el resultado de cada iteración y los hitos del proyecto están enfocados a conseguir un cierto grado de completitud y estabilidad de los artefactos, como se podrá comprobar cuando se presenten los objetivos de cada iteración.

Los artefactos entregables a desarrollar a lo largo del proyecto, agrupados por flujos de trabajo, son:

- Gestión del proyecto
 - Plan de Desarrollo del Software. Se trata del presente documento.
 - Planes de iteración. Uno para cada iteración definida. Los dos primeros se incluyen en esta entrega, en los documentos Plan de Iteración: Inicio y Plan de Iteración: Elaboración.
 - Informes de seguimiento de cada iteración. También se generará uno para cada iteración definida.
 - Análisis de riesgos. Se incluye en este documento.
 - Glosario. Define los principales términos utilizados en el proyecto. Se incluye en el primer anexo del presente documento.
- Requisitos
 - Documento de Visión. Define la visión del producto desde la perspectiva del cliente, especificando las necesidades y características del producto. Constituye una base de acuerdo en cuanto a los requisitos del sistema.
 - Documento de especificación de los requisitos.
 - Modelo de casos de uso.
- Análisis y Diseño
 - Modelo de Análisis.
 - Modelo de Diseño.
 - Modelo de Datos.
 - Prototipo de interfaz. Desarrollado con la intención de permitir validar la interfaz con el usuario final del sistema.
- Gestión de configuraciones
 - Plan de gestión de configuraciones. Este artefacto se incluye en el presente documento.

- Implementación
 - Modelo de Implementación.
 - Versión ejecutable (enlace).
- Pruebas
 - Plan de pruebas. Detalla el procedimiento a seguir, así como los casos utilizados para probar el sistema.
- Operación
 - Manual de instalación.
 - Manual de usuario. Desarrollado para cada tipo de usuario final.

2.4. Evolución del Plan de Desarrollo del Software

El Plan de Desarrollo del Software se revisará periódicamente, y se refinará antes del comienzo de cada iteración.

Como ya se ha comentado, junto con este documento se entrega el plan de la primera iteración, que contendrá como una actividad más la planificación de la segunda, y así sucesivamente.

3. Organización del proyecto

3.1. Estructura organizativa

Al tratarse de un grupo reducido de personas, ser un proyecto de origen académico y para conseguir el máximo aprovechamiento de los recursos disponibles, cada miembro podrá desempeñar roles distintos aunque no concurrentes. El encargado de recoger los informes de actividad de todos los roles y hacer el seguimiento de este plan de proyecto será el gestor de proyecto.

A continuación se presentan en la *Figura 1* los diferentes roles necesarios en el equipo de desarrollo del proyecto, así como su relación con el gestor o jefe del proyecto principal.

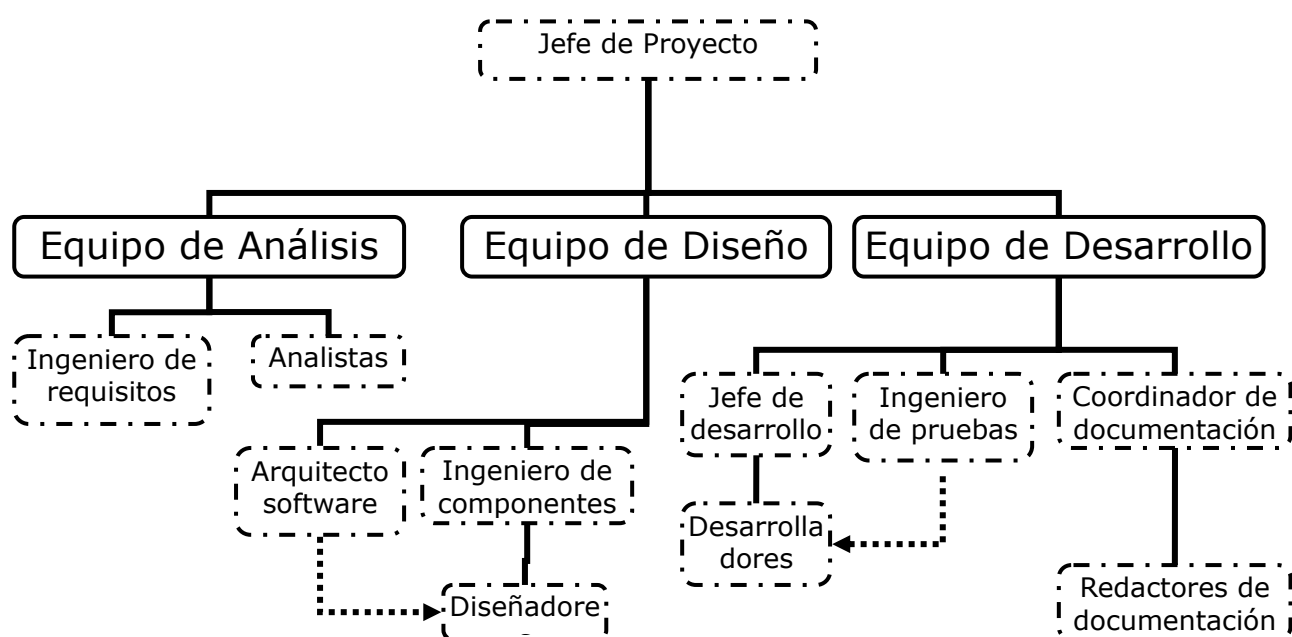


Figura 1

3.2. Interfaces externas

El equipo de desarrollo deberá interactuar con dos actores ajenos a la práctica:

- El profesor de la asignatura ejercerá el rol de cliente de la aplicación. Los ingenieros de requisitos mantendrán reuniones periódicas con el cliente para reunir los requisitos, informar del progreso del proyecto y discutir posibles cambios en los requisitos iniciales.
- Los técnicos de laboratorio de la escuela ETSI Informática proporcionaran el soporte para lo práctica y las restricciones para su posterior desarrollo.

3.3. Roles y responsabilidades

A continuación se listan los distintos roles presentes en las fases de análisis, diseño e implementación.

- **Gestor de proyecto:** estudia inicialmente el proyecto, estableciendo el plan de proyecto y el ámbito del mismo. Supervisa todas las fases del proyecto definidas en el plan inicial para controlar que se cumplan las actividades en el tiempo y el coste indicados. Coordina todos los procesos y distribuye las actividades entre los miembros del equipo de trabajo.
- **Gestor de configuraciones:** apoya a la actividad de desarrollo de software dotándola de un entorno de trabajo para construir y probar los componentes del sistema. Garantiza la constante disponibilidad de una versión estable de cada artefacto del proceso para todos los miembros del equipo. Gestiona el entorno de control de configuraciones para facilitar las actividades de revisión, cambio y seguimiento. Es también es el responsable de redactar el Plan de Gestión de Configuraciones y de gestionar e informar de las Peticiones de Cambio, si las hubiere.
- **ANÁLISIS:**
 - **Analista:** realiza entrevistas para obtener nuevos requisitos y necesidades del cliente o completar los existentes. Determina cuándo los requisitos son incompletos, ambiguos, inconsistentes o contradictorios y los resuelve. Documenta de varias formas los requisitos. Define la arquitectura del sistema a muy alto nivel. Elabora diagramas de clases y secuencia para mostrar los bloques que estarán relacionados y se comunicarán entre sí. Dichos diagramas facilitarán la comprensión de los casos de uso de los usuarios.
 - **Revisor:** controla la calidad del software desarrollado y del proceso de desarrollo.
- **DISEÑO:**
 - **Diseñador software:** Establece la arquitectura software para la herramienta. Estudia la interoperabilidad con otros sistemas y la posibilidad de reutilización de software existente. Aprovecha las últimas tecnologías disponibles. Desarrolla componentes a bajo nivel, estructuras de datos eficientes y algoritmos adecuados. También participa en detalles de la arquitectura a bajo nivel. Elabora documentación en forma de diagramas.
 - **Ingeniero de bases de datos:** diseña y construye todas las tablas de la base de datos. Establece la interfaz de acceso a datos desde la herramienta.
- **IMPLEMENTACIÓN:**
 - **Programador:** construye los componentes diseñados en el lenguaje de programación establecido. Elabora código fuente y su documentación asociada.
 - **Ingeniero de pruebas:** verifican el trabajo de los programadores mediante pruebas de los casos de uso de la herramienta usando test unitarios y de integración. Prueba formalmente algunas porciones de código y revisa otras. Se encarga de la documentación de pruebas.

4. Gestión del proyecto

4.1. Resumen de la planificación

Se desarrollará el proyecto siguiendo el modelo de Proceso Unificado, dirigido por los casos de uso e intentando distribuir el trabajo equitativamente durante el desarrollo del mismo. El plan de fases se puede consultar en el punto 4.2.1 y el planteamiento de las iteraciones se detalla en el punto 4.2.3.

La planificación y el posterior seguimiento de fechas y costes se realiza mediante la herramienta Microsoft® Project. Las valoraciones de tiempo han sido estimadas teniendo en cuenta los ejemplos vistos en clase y en UPEDU, así como nuestra experiencia a lo largo de la carrera.

En el siguiente apartado se encuentra desarrollada la planificación del proyecto.

4.2. Plan de Proyecto

4.2.1. Plan de fases

La duración del proyecto constará de 13 semanas, sin embargo 3 de ellas corresponden al período navideño y el trabajo en dichas semanas disminuirá, aunque se seguirá con él siguiendo un plan específico.

El proyecto se realizará en 4 iteraciones más la etapa de Inicio, que también se contabilizará como una iteración y por ello tendrá su propio plan de iteración.

- **Inicio:** define el alcance y los objetivos del proyecto, la funcionalidad y capacidades del producto, obtiene una arquitectura candidata, realiza estimaciones iniciales de planificación y costes, y busca detectar riesgos cuanto antes para minimizar su efecto.
- **Elaboración:** extrae todos los requisitos funcionales y no funcionales del documento de especificación del problema. Diseño y especificación de los casos de uso. Se comenzará con la parte inicial de análisis y diseño de la aplicación (definiendo la arquitectura). Desarrollo inicial del plan de pruebas.
- **Construcción 1:** se procede a la implementación de las funcionalidades más importantes. Se prosigue con la actividad de diseño del sistema y se completa la arquitectura. Se desarrolla la base de datos. Se realizarán pruebas de los componentes implementados. Asimismo se completa el diseño del plan de pruebas.
- **Construcción 2:** se finaliza la implementación cumpliendo la totalidad de los requisitos. Las modificaciones sobre el diseño serán mínimas. Se realizarán pruebas de los componentes implementados. Se comenzará la elaboración del manual del usuario final.
- **Transición:** se llevarán a cabo las pruebas de integración, y la redacción completa del manual de usuario. Se obtendrá un producto estable.

4.2.2. Objetivos de las iteraciones

En la *Tabla 1* se especifican los objetivos de cada una de las fases del proyecto. A expensas de los hitos representados, durante la fase de Inicio se desarrollarán varios prototipos desechables de uso interno, cuyo objetivo es el adiestramiento en las diferentes herramientas que utilizaremos en el desarrollo del proyecto (aprendizaje del lenguaje a utilizar, pruebas con diferentes IDE's, comunicación con el sistema gestor de base de datos...).

<i>Fase</i>	<i>Objetivo</i>	<i>Hito</i>
Inicio	Realizar la planificación.	Plan de proyecto
Elaboración	Elaboración de casos de uso y captura de requisitos.	
Construcción 1	Implementar funciones principales.	Versión Alfa
Construcción 2	Pruebas de la versión Alfa. Implementación de toda la funcionalidad.	Versión Beta
Transición	Pruebas de la versión Beta. Finalizar documentación. Entrega	Producto final

Tabla 1

4.2.3. Plan de iteración

Con el objetivo de gestionar los elementos de trabajo en el contexto de los períodos fijos de desarrollo dentro de un proyecto se establecen los planes de iteración. Con ellos se pretende crear una visión general del plan y ayudar a los miembros del equipo a organizarse mediante un detallado esquema de las actividades que se deben ir cumpliendo.

Estos planes se recogen en documentos PDF separados:

- Plan_Iteracion_Inicio.pdf (completo)
- Plan_Iteracion_Elaboracion.pdf (en desarrollo)
- Plan_Iteracion_Construccion.pdf
- Plan_Iteracion_Transicion.pdf

4.2.4. Calendario del proyecto

El calendario laboral será el mismo para todos los integrantes del grupo, trabajando fines de semana, si es necesario, para completar unas 10 horas semanales. Durante las semanas consideradas como fechas laborables se seguirá el siguiente plan de trabajo:

- Reunión semanal coincidiendo con los laboratorios de la asignatura para realizar un control del trabajo realizado, compartir dudas y repartir trabajo para la nueva semana. En dichas reuniones se discutirá la necesidad de tutorías con el profesor de la asignatura Pablo de la Fuente Redondo.
- Trabajo individual por objetivos marcados. La dedicación será personalizada para cada integrante atendiendo a su disponibilidad.

- Posibilidad de reuniones entre semana de partes o de la totalidad del grupo según las tareas encomendadas en la reunión semanal.
- Comunicación regular por medio de correos electrónicos para anuncios, cambios de tareas o avisos.

Durante las semanas de vacaciones se eliminarán las reuniones semanales, manteniendo una comunicación mucho más frecuente mediante correos electrónicos. Se distribuirán tareas al inicio de las vacaciones para cada uno de los integrantes y se realizará un seguimiento de su evolución con vistas a posibles modificaciones.

4.2.5. Recursos del proyecto

- **Recursos humanos:** En la ejecución de dicho proyecto, participarán todos los miembros del grupo 1 de prácticas de la asignatura de Ingeniería del Software II, que conformarán el equipo de desarrollo del Proyecto:
 - *Patricia Cadenas Quijano*
 - *Javier García Tomillo*
 - *Mario Muñoz Sanz*
 - *Mario Santos Serrano*
- El trabajo será supervisado por el profesor de la asignatura, Pablo de la Fuente Redondo. Además, dicho profesor orientará al equipo de desarrollo y ejercerá el rol de usuario del sistema, resolviendo todos aquellos detalles inconclusos en el guión de la práctica.
- **Recursos software:** Al tratarse de una aplicación Web, el sistema operativo sobre el que se desarrollará no será determinante, por tanto, cada miembro del equipo de desarrollo trabajará con el que tenga disponible (Microsoft Windows XP, Microsoft Windows Vista o Microsoft Windows 7).
- A continuación, se incluye una lista de las herramientas que se van a utilizar a lo largo de la iteración:
 - Editor de textos: Microsoft Word 2007 y OpenOffice.org Writer.
 - Gestor de proyectos: Microsoft Project 2007.
 - Navegadores Web: Microsoft Explorer 8, Mozilla Firefox 3 y Google Chrome.
 - Entorno de Desarrollo Integrado: Netbeans 6.9.
 - Programa de creación de diagramas UML: StarUML 5.0.2.
 - Servidor independiente de la plataforma: XAMPP que incorpora la base de datos MySQL, el servidor Web Apache y los intérpretes para lenguajes de script: PHP y Perl.
 - Repositorio Subversion: <http://code.google.com/p/sigestproso1/>
 - Cliente Subversion: TortoiseSVN 1.6.11.
 - Compartir archivos: Dropbox.
- La diferencia de uso del repositorio Subversion respecto a Dropbox consiste en que el primero se utilizará para almacenar y compartir todos aquellos documentos definitivos o semi-definitivos que necesiten mayor seguridad (control de versiones o bloqueo de archivos en uso). Sin embargo, Dropbox se utilizará como canal de intercambio de archivos durante toda la realización del proyecto con el fin de compartir ideas, avisos o ayuda.

- **Recursos hardware**

- Cada uno de los integrantes del equipo de desarrollo dispondrá de su propio ordenador para la realización del proyecto.
- Equipos del laboratorio general de la ETS de Ingeniería Informática.
- Servidor Jair de la ETS de Ingeniería Informática perteneciente a la Universidad de Valladolid.

4.3. Monitorización y Control del Proyecto

4.3.1. Gestión de Requisitos

Las características de la herramienta a desarrollar así como los requisitos del sistema serán especificados en los artefactos Visión y Especificación de Requisitos Software (SRS) respectivamente, los cuáles se desarrollarán en la fase de Inicio. Cada requisito tendrá una serie de atributos tales como descripción, prioridad y, en su caso, comentarios adicionales, que permitirán asegurar un seguimiento detallado de los mismos.

4.3.2. Control de calidad

Todos los entregables serán sometidos a un proceso de revisión por miembros del equipo de desarrollo distintos al autor. Para la revisión se utilizan las líneas guía y *checklist* de UPEDU que garantizan que todos los artefactos tengan una calidad aceptable.

4.3.3. Gestión de riesgos

Los riesgos se identifican durante la fase de Inicio, en la que se elabora una lista de riesgos asociados al proyecto y las acciones establecidas para mitigarlos o contenerlos si no es posible eliminarlos. En las siguientes fases se seguirá y monitorizará su evolución durante cada iteración, añadiendo si fuera necesario, los riesgos no contemplados con anterioridad.

Utilizaremos las siguientes definiciones y convenios para describir cada riesgo:

- **Magnitud:** Califica el nivel de riesgo del 1 al 10 basándonos en la probabilidad de ocurrencia y en la criticidad del mismo.
- **Descripción:** Breve explicación del posible problema.
- **Impacto:**
 - C - Crítico: afecta a toda la funcionalidad del proyecto y a la línea base.
 - A - Alto: afecta a las necesidades del usuario y a la funcionalidad.
 - M - Medio: se puede contener pero la mayoría de las veces es evitable.
 - B - Bajo: se puede aceptar y convivir con él pero también mitigar rápidamente.
- **Indicador:** métrica o detector del riesgo a monitorizar.
- **Plan de acción/contingencia:** medidas a tomar para evitar o convivir/ transferir el riesgo.

RIESGO – Nombre descriptivo.				
Magnitud	Descripción	Impacto	Indicador	Plan de acción/contingencia

Tabla 2

4.2.8.1. Lista de riesgos

- Inexperiencia con el modelo de proceso UPEDU
 - *Magnitud:* 6
 - *Descripción:* La inexperiencia de los miembros del equipo con el proceso UPEDU puede ser un impedimento.
 - *Impacto:* A
 - *Indicador:* Malos artefactos, baja calidad del desarrollo, pérdida de esfuerzo
 - *Plan de acción/contingencia:* Establecer una fuente de consulta (profesor de la asignatura) y unas fuentes bibliográficas suficientes y adecuadas comunes para todos los miembros del equipo.
- Personal insuficiente en algunas fases
 - *Magnitud:* 3
 - *Descripción:* Puede que debido al esfuerzo requerido en algunas fases del proyecto, el número de miembros del equipo sea insuficiente.
 - *Impacto:* C
 - *Indicador:* Trabajo atrasado de forma continuada, baja calidad.
 - *Plan de acción/contingencia:* Mejorar la organización del trabajo y eliminar objetivos secundarios en tal iteración.
- Abandono del equipo por parte de algún miembro
 - *Magnitud:* 1
 - *Descripción:* Puede producirse la baja de algún miembro del equipo debido a motivos de carácter personal, enfermedad, etc.
 - *Impacto:* C
 - *Indicador:* No existen indicadores para este tipo de riesgo porque es impredecible
 - *Plan de acción/contingencia:* Se procurará finalizar el trabajo correspondiente a cada fase antes de la fecha planificada siempre que sea posible, para que el impacto de este riesgo sea el menor posible. En caso de producirse, se revisarán de inmediato la línea base y el reparto de trabajo entre los miembros del equipo.
- Entrega del proyecto posterior a la fecha planificada
 - *Magnitud:* 7
 - *Descripción:* Pueden producirse retrasos en la entrega final del proyecto debido a motivos muy diversos.
 - *Impacto:* C
 - *Indicador:* 10% de actividades del proyecto pendientes 7 días antes de la fecha planificada.
 - *Plan de acción/contingencia:* Se monitorizará de forma continua el progreso del trabajo para compararlo con las fechas previstas. En caso necesario se aportarán horas extra y se revisará el reparto de tareas de los miembros del equipo con menor carga de trabajo.
- Entrega del proyecto posterior a la fecha límite
 - *Magnitud:* 7
 - *Descripción:* Debido a trabajo externo al proyecto, algunos miembros del equipo podrían ver comprometido su tiempo para trabajar en él.

- *Impacto: C*
- *Indicador:* Incumplimiento de las tareas asignadas por la planificación para algún miembro del grupo.
- *Plan de acción/contingencia:* comprobar continuamente lo realizado con la planificación, y modificar el reparto de tareas si fuera necesario.
- Mala captación de requisitos
 - *Magnitud: 5*
 - *Descripción:* La falta de comunicación con el cliente y el desconocimiento del entorno puede provocar una mala captación de requisitos.
 - *Impacto: A*
 - *Indicador:* entrevistas, y su objetivo, mal planteadas, pérdida de tiempo innecesaria en las entrevistas.
 - *Plan de acción/contingencia:* Estudiar la propuesta del cliente de forma exhaustiva y organizar y ordenar todas las dudas a resolver con el cliente en las primeras reuniones. Además se realizará un prototipo no operativo evaluable por el cliente a partir de los primeros requisitos captados, para comprender mejor las necesidades. Si este riesgo se produce, se realizará una nueva reunión con el cliente para resolver al detalle todas las dudas.
- Cambios en los requisitos
 - *Magnitud: 4*
 - *Descripción:* El cliente puede cambiar algún requisito o simplemente añadirlo en cualquier momento, debido a que no tenga claro las funcionalidades de la aplicación, se le ocurran nuevas ideas o descubra algún requisito erróneo.
 - *Impacto: A*
 - *Indicador:* Información directa por parte del cliente, detección de requisitos contrapuestos o incompatibles.
 - *Plan de acción/contingencia:* Se realizarán reuniones con el cliente, con alta frecuencia en las fases iniciales, y se comprometerá al cliente a aceptar los requisitos como definitivos. En caso de producirse este riesgo se estudiará el coste que supone la aceptación de dicho cambio, y se modificará la planificación de todo el proyecto en caso de ser aceptado.
- Coste del proyecto superior al planificado
 - *Magnitud: 6*
 - *Descripción:* Debido a motivos diversos el coste del proyecto (horas de trabajo) puede verse incrementado de forma inesperada.
 - *Impacto: M*
 - *Indicador:* requisitos del cliente muy exigentes, mala planificación, retraso excesivo en alguna de las fases.
 - *Plan de acción/contingencia:* Para evitar este riesgo se realizará una planificación optimizada. En caso de producirse será necesario aumentar la carga de trabajo de los miembros del equipo modificando la planificación, e intentar renegociar con el cliente las fechas de entrega.
- Incompatibilidad del producto con el software y hardware del cliente
 - *Magnitud: 3*
 - *Descripción:* El producto final podría ser incompatible con el software o hardware utilizado por el cliente (versiones antiguas de navegador web, hardware obsoleto, etc.)

- *Impacto:* M
- *Indicador:* Errores de procesamiento o presentación al ejecutar la herramienta.
- *Plan de acción/contingencia:* Utilizar durante el desarrollo tecnologías cruzadas entre navegadores y hojas CSS de nivel 3 máximo. Se utilizarán solamente tecnologías aprobadas por W3C. Además se realizará un estudio inicial del hardware y software utilizado por el cliente para evitar incompatibilidades inesperadas.
- Falta de comunicación entre miembros del equipo
 - *Magnitud:* 4
 - *Descripción:* Puede producirse una falta de comunicación entre los miembros del equipo debido a días festivos, trabajos exteriores al proyecto u otros.
 - *Impacto:* C
 - *Indicador:* más de 10 días desde la última reunión, falta de informes de progreso semanal y charlas diarias informales.
 - *Plan de acción/contingencia:* se concertará un día fijo a la semana para hacer una reunión de seguimiento. Además, si esto no fuera suficiente, se propondrían nuevas fechas para realizar reuniones de seguimiento adicionales, de forma presencial o por videoconferencia, listas de correo, chat etc.
- Cambios en el entorno de desarrollo software
 - *Magnitud:* 2
 - *Descripción:* Pueden producirse cambios en el entorno de desarrollo software (MySQL, PHP) por motivos diversos.
 - *Impacto:* B
 - *Indicador:* cambios de versiones de drivers, software o IDE en las máquinas de trabajo. Adición de nueva funcionalidad aprovechable para nuestro producto.
 - *Plan de acción/contingencia:* Se mantendrá frecuentemente actualizada la gestión de configuraciones. En caso de producirse este riesgo se reconstruirán los componentes afectados, siempre que sea posible, actualizando la planificación. Como última opción posible, se volverá a versiones anteriores del entorno o se cambiará la máquina de trabajo.
- Diseño erróneo
 - *Magnitud:* 5
 - *Descripción:* Se realizará durante la fase de elaboración un prototipo para comprobar que la arquitectura es la adecuada, en caso de no serlo se modificará el diseño. En caso de producirse este riesgo se revisará y modificará la documentación de diseño afectada si nos encontramos en la fase de elaboración. Si estamos en la fase de construcción se estudiará la solución adecuada de acuerdo con los plazos de tiempo de que se disponga.
 - *Impacto:* A
 - *Indicador:* La estructura no cumple con las expectativas, la implementación se complica más de lo esperado.
 - *Plan de acción/contingencia:* Durante la fase de Elaboración se realizará un prototipo para comprobar que la arquitectura es la adecuada. En caso de no serlo, se modificará la parte del diseño afectada. Si nos encontrásemos en la fase de Construcción se estudiaría la solución más acorde de acuerdo con los plazos de entrega de que se dispone.

- Pérdida de datos
 - *Magnitud:* 5
 - *Descripción:* Se produce una pérdida de datos de forma desconocida
 - *Impacto:* C
 - *Indicador:* Ninguno
 - *Plan de acción/contingencia:* Se usará un repositorio para el control de versiones. Además se realizarán copias de seguridad periódicas en los ordenadores personales de cada uno de los miembros del equipo. En caso de producirse una pérdida de datos, se volverá a una versión anterior a la pérdida para intentar reconstruirla.

- Errores debido a la gestión de versiones
 - *Magnitud:* 5
 - *Descripción:* Aparecen errores debido a una mala unificación de versiones entre archivos que han sido modificados por varias personas del grupo durante un mismo periodo de tiempo.
 - *Impacto:* M
 - *Indicador:* El funcionamiento de alguna parte del proyecto se ve alterado de forma inesperada.
 - *Plan de acción/contingencia:* Se bloqueará un archivo cuando un miembro del grupo lo esté modificando. En caso de producirse se regresará a una versión anterior del archivo y se revisarán las modificaciones a realizar por parte de los miembros del equipo cuyo trabajo se ha visto afectado.

- Errores producidos por software de terceros
 - *Magnitud:* 3
 - *Descripción:* Aparecen errores en la implementación del sistema debido al software empleado para el desarrollo del mismo (servidor web, sistema gestor de bases de datos, librerías utilizadas, etc.).
 - *Impacto:* B
 - *Indicador:* Origen de un error en la implementación del sistema.
 - *Plan de acción/contingencia:* se estudiarán los elementos que van a componer el sistema para detectar posibles incompatibilidades o errores. En caso necesario se valorará la posibilidad de cambiar el software utilizado por una versión distinta o un software equivalente.

4.3.4. Gestión de configuraciones

A continuación desarrollamos todos los aspectos relativos al control de versiones del proyecto, complemento necesario para que el Plan de Desarrollo Software cumpla con el estándar IEEE según UPEDU. Al principio se especifican las características del entorno de gestión de configuraciones. Después se indica el funcionamiento del programa (identificación, control de versiones...). Finalmente se añade la estructura de directorios disponible en el repositorio.

El responsable de realizar las actividades de Gestión de Configuraciones es el Gestor de Configuraciones. Debido al reducido equipo de desarrollo este rol será adquirido eventualmente por varios miembros con lo que todos participan en las actividades de Gestión de Configuraciones. A efectos oficiales, se le asigna el rol de Gestor de Configuraciones a Mario Santos Serrano.

Puesto que el equipo de desarrollo necesita total disponibilidad de los artefactos actualizados y versiones anteriores para trabajar con ellos en diferentes ubicaciones remotas, utilizaremos un servidor SVN para crear un repositorio con control de versiones.

Subversion es un sistema de control de versiones que resuelve algunas de las deficiencias presentadas por el antiguo y popular CVS: los artefactos versionados no tienen ahora un número de versión independiente, sino que todos los artefactos del repositorio tienen un único identificador de versión que identifica un estado común único del proyecto en un instante de tiempo. Es software libre (licencia Apache/BSD) y el nombre SVN le viene de svn, nombre de la herramienta en línea de comandos.

Los miembros del grupo podrán acceder al repositorio a través del cliente TortoiseSVN mediante los siguientes datos. Cada miembro del grupo dispondrá de su propio usuario y contraseña para acceder al repositorio y que a la vez identifique sus cambios en el mismo. Como ejemplo para acceder se muestra el siguiente:

- **URL:** <https://sigestproso1.googlecode.com/svn>
- **Username:** mario_mazado@hotmail.com
- **Password:** Rr3eX3Zk9bQ7

Periódicamente, el Gestor de Configuraciones realizará una copia de seguridad en su ordenador personal.

La estructura de directorios podrá sufrir variaciones dependiendo de las necesidades que vayan surgiendo, pero en al comienzo del desarrollo será la siguiente:

- **doc:** carpeta en la que se almacenará toda la documentación de la práctica. Constará, a su vez, de las siguientes subcarpetas:
 - **análisis:** contendrá los requisitos, casos de uso...
 - **diseño:** almacenará todos aquellos diagramas de diseño realizados.
 - **manuales:** recopilará el manual de instalación y el de usuario.
 - **test:** en ella se guardarán las diversas pruebas realizadas sobre la aplicación.
- **otras cosas:** carpeta reservada para todo aquello que creamos conveniente almacenar y que no se encuentre en las otras categorías.
- **src:** carpeta donde almacenaremos todos los ficheros de código. Los archivos subidos serán versiones funcionales, ausentes de errores. Para compartir archivos temporales, con errores o faltos de funcionalidad se utilizará la carpeta compartida por todos los integrantes del grupo disponible en Dropbox. Constará, a su vez, de las siguientes subcarpetas:
 - **MySQL:** contendrá todos los scripts necesarios para crear y poblar las tablas de la base de datos con las que trabajaremos..
 - **PHP:** archivos de código de la aplicación Web.

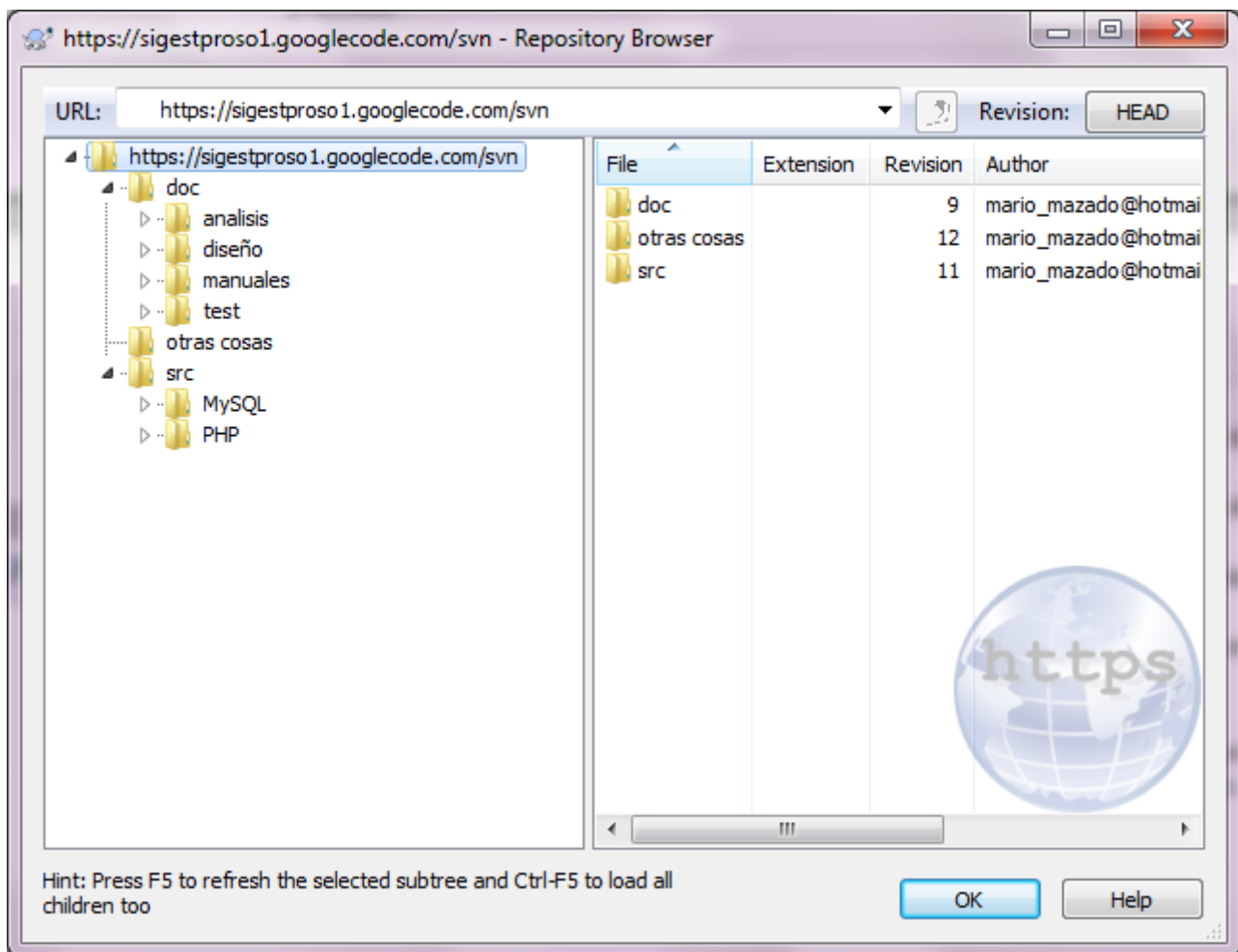


Figura 2

Anexo: Glosario del proyecto

1. Definiciones

- **Actividad:** Unidad de trabajo que puede ejecutar un individuo o varios en un rol específico. Tiene un propósito claro y se expresa en términos de actualizar artefactos.
- **Artefacto:** Es una pieza de información utilizada o producida mediante un proceso de desarrollo de software. Puede ser un modelo, una descripción o tratarse de software. No siempre es un entregable.
- **Calendario:** Tabla que indica los días en que un empleado de la empresa debe asistir al trabajo pero también aquéllos en los que éste se encuentra de vacaciones.
- **Entregable:** Artefacto que debe ser entregado al cliente.
- **Equipo de desarrollo:** Conjunto de personas implicadas en el desarrollo del proyecto.
- **Etapas del proceso:** Conjunto de actividades orientadas a la consecución de una serie de objetivos, que se denominan objetivos de la etapa.
- **Fase de desarrollo:** Cada una de las etapas que conciernen al ciclo de vida de un proyecto software, situadas entre dos hitos principales.
- **Gestor de proyecto:** Ver "Jefe de proyecto".
- **Hito:** Evento de relevancia durante el proceso. A menudo surgen de la consecución de algún objetivo.
- **Informe:** Documento detallado sobre una tarea, actividad o trabajador de la empresa del proyecto en curso.
- **Ingeniero de requisitos:** miembro del equipo de desarrollo responsable de la captación de los requisitos del cliente.
- **Iteración:** Conjunto de actividades llevadas a cabo de acuerdo a un plan (de iteración) y unos criterios de evaluación, que lleva a producir una versión, ya sea interna o externa.
- **Jefe de proyecto:** Trabajador de la empresa responsable de un proyecto de la misma. Debe coordinar el trabajo del resto de personal asignado al proyecto así como realizar la planificación.
- **Línea base:** Artefacto del proyecto revisado y aceptado por todas las partes involucradas.
- **Modelo de proceso:** Conjunto de técnicas que definen como el proceso software debe llevarse a cabo en todas sus distintas etapas, así como cuáles han de ser dichas etapas. Un modelo de proceso es una abstracción, de manera que cada proceso se convierte en una instanciación del modelo al que pertenece.
- **Peticiones de cambio:** artefacto utilizado para rastrear todas las peticiones de las partes interesadas junto con la información de estado relacionada, en todo el ciclo de vida del proyecto. Toda la historia de cambios se mantendrá con la solicitud de cambio junto con las fechas y las razones para el cambio.
- **Planificación:** Se compone del plan general de ejecución del proyecto, así como del desarrollo real de dicho plan, incluyendo información detallada sobre el tiempo que requerirá cada actividad y los recursos que se destinarán a la misma.
- **Plan de Gestión de Configuraciones:** artefacto que describe las actividades de Gestión de Configuración, los procedimientos y el cronograma para su realización, y los responsables de dichas actividades.

- **Pliego de condiciones:** Documento que incluye la descripción general del proyecto y establece las condiciones o cláusulas generales que deben cumplirse.
- **Proceso:** Es el flujo de acciones que llevan a la consecución de los objetivos del proyecto.
- **Proyecto:** Esfuerzo temporal emprendido para elaborar un producto software. Se trata de la unidad lógica de gestión dentro del entorno de la aplicación: cada proyecto requiere una planificación que se lleva a cabo siguiendo un modelo de proceso y es gestionado por un único jefe de proyecto.
- **Proyecto finalizado/cerrado:** Aquél que ha alcanzado todos los objetivos que se plantearon inicialmente. Para llegar a su finalización, un proyecto requiere haber pasado y finalizado con éxito todas las etapas del proceso que fueran planificadas en el origen.
- **Recurso:** Cada uno de los elementos que proporcionan un servicio necesario para la ejecución del proceso. Deben estar disponibles para su utilización. Pueden ser humanos, económicos o físicos.
- **Rol:** Papel de un trabajador dentro de la empresa e incluso dentro del desarrollo de un proyecto que permite acotar sus tareas con mayor facilidad.
- **Tarea:** Es la granularidad mínima en la que se dividen las acciones que integran cada etapa del proceso software.
- **Trabajador:** Empleado de la empresa. Desde el punto de vista del proyecto, se refiere a cualquier persona implicada en cualquiera de las fases del mismo.

2. Acrónimos

- **HTML:** HyperText Markup Language (Lenguaje de marcado de hipertexto).
- **IDE:** Integrated Development Environment (Entorno de Desarrollo Integrado).
- **IEEE:** Institute of Electrical and Electronics Engineers (Instituto de Ingenieros Eléctricos y Electrónicos). Asociación técnica mundial dedicada, entre otras cosas, a la estandarización.
- **PHP:** PHP Hypertext Pre-Processor. Lenguaje de programación interpretado, diseñado originalmente para la creación de páginas web dinámicas.
- **SIGESTPROSO:** Seguimiento Integrado de la GESTión Temporal de PROyectos de Software.
- **SPMP:** Software Project Management Plan (Plan de Gestión de Proyectos Software).
- **SRS:** Software Requirements Specification (Especificación de Requisitos Software).
- **UML:** Unified Modeling Language (Lenguaje Unificado de Modelado).
- **UPEDU:** Unified Process for EDUcation (Proceso Unificado para la Educación).
- **UVa:** Universidad de Valladolid.

Anexo: Referencias

- [1] Sistema para Gestión del ES-NIC. Plan de Desarrollo de Software, Versión 0.9.
- [2] Sciamachy Data Centre. NL "Software Project Management Plan". SCIA-DC-SPMP-1.1.
- [3] Booch, Grady; Jacobson, Ivar; Rumbaugh, James "El Proceso Unificado de Desarrollo Software". PEARSON Addison Wesley, 2005.
- [4] Plan de iteraciones (iteración 1 en fase de inicio) "Plan_Iteracion_Inicio".

Referencias web

- [5] Repositorio ofrecido por Google:
Disponible en: <http://www.google.com/intl/es/apis/gadgets/tools.html#Host>
- [6] UPEDU "Risk List Artefact"
Disponible en: <http://www.upedu.org/upedu/>.
- [7] De la Fuente, Pablo Lucio - Asignatura Ingeniería de Software II-
Disponible en : <http://aulas.inf.uva.es>