

SEPROSO.
Seguimiento de Proyectos
Software.
SPMP.

Francisco Javier Delgado del Hoyo
Yuri Torres de la Sierra
Rubén Martínez García
Abel Lozoya de Diego

Diciembre, 2008

Signature

Rubén Martínez García

Yuri Torres de la Sierra

Abel Lozoya de Diego

Francisco Javier Delgado del Hoyo

Revisiones del documento

Historial de revisiones del documento

VERSIÓN	FECHA	DESCRIPCIÓN	AUTOR
0.1	7/11/08	Recopilación de Información Inicial.	Grupo III
0.2	17/11/08	Primera versión del plan de proyecto.	Grupo III
1.0	28/11/08	Completar plan de proyecto.	Grupo III

Prólogo

El propósito de este documento es especificar el plan de desarrollo del proyecto software SEPROSO (Seguimiento de Proyectos Software). Este documento presenta el plan de proyecto inicial que especifica los objetivos, los planes de gestión de proyecto, el presupuesto y la organización de los miembros del equipo. Las actualizaciones de este documento se producirán a medida que el proyecto avance, sirviendo esta especificación inicial como línea base para la comprobación del desarrollo del proyecto.

Este Plan de Desarrollo del Software es una versión preliminar preparada para ser incluida en la propuesta elaborada como respuesta al proyecto de prácticas de la asignatura de Ingeniería del Software II de la Escuela Técnica Superior de Ingeniería Informática de Valladolid. Este documento provee una visión global del enfoque de desarrollo propuesto.

La metodología básica sobre la que se sustenta el proyecto será Proceso Unificado, RUP (Rational Unified Process). El enfoque de desarrollo propuesto constituye una configuración de UPEDU, seleccionando los roles de los participantes, las actividades a realizar y los artefactos (entregables) que serán generados. Este documento es a su vez uno de los propios artefactos del Proceso Unificado.

Indice

Signature	i
Revisiones del documento	ii
Prólogo	iii
1 Introducción.	1
1.1 Propósito.	1
1.2 Alcance.	1
1.3 Acrónimos y definiciones.	1
1.4 Referencias.	1
2 Visión General del Proyecto	3
2.1 Propósito, ámbito y objetivos.	3
2.2 Suposiciones y restricciones.	4
2.3 Entregables del proyecto.	5
2.4 Evolución del Plan.	7
3 Organización del Proyecto.	9
3.1 Estructura de la Organización.	9
3.2 Interfaces Externas.	9
3.3 Roles y Responsabilidades.	10
4 Proceso de Gestión.	13
4.1 Resumen de la planificación y el presupuesto.	13
4.2 Plan de Proyecto.	14
4.2.1 Plan de fases.	14
4.2.2 Objetivos de las iteraciones.	14
4.2.3 Plan de iteración.	16
4.2.4 Calendario del proyecto.	16
4.2.5 Recursos del proyecto.	16
4.3 Monitorización y control.	16
4.3.1 Gestión de Requisitos.	16
4.3.2 Control de Calidad.	18

<i>INDICE</i>	v
4.3.3 Informes y mediciones.	18
4.3.4 Gestión de Riesgos.	18
4.3.5 Gestión de Configuraciones.	18
A Anexos	20

Indice de figuras

4.1 Duración de la planificación de las iteraciones.	15
--	----

Indice de tablas

2.1	Fechas previstas de generación de entregables.	8
3.1	Roles asignados a miembros del equipo	11
3.2	Responsabilidades asignadas a cada rol	12
4.1	Resumen de las etapas del plan de desarrollo.	13
4.2	Hitos y objetivos de las iteraciones.	16
4.3	Recursos del proyecto.	17

Capítulo 1

Introducción.

1.1 Propósito.

El **propósito** del Plan de Desarrollo de Software es proporcionar la información necesaria para controlar el proyecto. 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, y para realizar su seguimiento.
- Los miembros del equipo de desarrollo lo utilizab para el conocimiento de la labor de cada uno, cuándo se debe hacer y qué otras actividades dependen de dicho trabajo.

1.2 Alcance.

El plan de desarrollo software describe el plan global que será utilizado por el grupo de desarrollo de la herramienta SEPROSO. Los detalles de cada una de las iteraciones se encuentran descritos en el plan de iteraciones. El plan de desarrollo descrito en en este documento se basa en los requisitos propuestos para le herramienta SEPROSO, definidos en el documento SRS.

1.3 Acrónimos y definiciones.

Véase el **Glosario**.

1.4 Referencias.

IEEE 1058 Software Project Management Plans. ANSI/IEEE Std 1058-1998.

IEEE 828 IEEE Standard for Software Configuration Management Plans,
ANSI/IEEE Std. 828-199.

IEEE 1074 IEEE Standard for Developing Software Life Cycle Processes,
ANSI/IEEE Std. 1074-1991.

IEEE 830 Estándar de documentación IEEE/ANSI 830-1993..

1. Definición de Roles y Responsabilidades de proceso unificado: http://www-128.ibm.com/developerworks/rational/library/sep06/collaris_dekker_warmer/index.html.
2. Asignatura Ingeniería de Software II, profesor: Pablo Lucio de la Fuente. Web: <http://aulas.inf.uva.es>

Capítulo 2

Visión General del Proyecto

2.1 Propósito, ámbito y objetivos.

El **propósito** del proyecto es analizar los requisitos, diseñar e implementar una herramienta web para el Seguimiento de Proyectos Software, SEPROSO, de acuerdo a los requisitos especificados en el enunciado de la práctica de Ingeniería del Software. La herramienta debe facilitar a un gestor de proyectos software la realización de un seguimiento temporal de las actividades que se realizan en sus proyectos, y ayudar a los trabajadores del proyecto a informar de sus progresos.

SEPROSO es una aplicación web, orientada a ofrecer un servicio específico a distintos grupos de usuarios en función del rol que dichos usuarios llevarán a cabo dentro del proyecto. La herramienta permitirá la generación de informes para el gestor del proyecto, y la consulta de proyectos ya finalizados como sistema de ayuda para proyectos nuevos. Esta herramienta permitirá gestionar cualquier tipo de Proyecto Software.

Actualmente esta labor de seguimiento es muy necesario puesto que el software existente se centra en la planificación inicial. El proyecto no se relaciona con proyectos anteriores, sino que su desarrollo parte de cero.

La idea del desarrollo de la herramienta es que los gestores de proyectos utilicen los datos de los proyectos finalizados (cerrados) para ayudar a planificar y presupuestar los nuevos y para analizar las prácticas de trabajo seguidas con el fin de ayudar en un proceso de mejora del software.

En el **ámbito** del proyecto se incluyen todas las actividades relacionadas directamente para conseguir el propósito. Están fuera del ámbito del proyecto actividades como el desarrollo del hardware o el mantenimiento de la red soporte de la herramienta.

Los **objetivos** del proyecto son los siguientes:

- Finalizar el proyecto antes de la fecha indicada.
- Ajustar al máximo el desarrollo posterior a este plan inicial.
- Entregar todos los artefactos de la sección 1.3.
- Cumplir todos los requisitos establecidos en el enunciado, recogidos en el documento SRS, en la herramienta entregada.
- Adquirir conocimientos de planificación y desarrollo de software mediante métodos de ingeniería.

2.2 Suposiciones y restricciones.

El proyecto se desarrolla basándonos en las siguientes **suposiciones**:

- El equipo de trabajo está formado por 4 personas que pueden emplear múltiples roles.
- Todos los miembros del equipo tendrán el mismo nivel de implicación y trabajo.
- La plataforma cliente para la herramienta será un navegador web, y el servidor soportará PHP 5.0
- El sistema gestor de bases de datos en línea estará operativo cuando se utilice la herramienta.
- La documentación se elabora siguiendo los estándares de la IEEE.

Las **principales restricciones** de la herramienta son las siguientes:

- Presupuesto: el número total de horas de trabajo de todos los trabajadores no superará xxx.
- Tiempo: el proyecto deberá estar finalizado y entregado antes del examen de la asignatura ISO II. Una vez entregado deberá presentarse al profesor de la asignatura.
- Equipo: ocupará el mismo lugar de trabajo en los días lectivos de la UVA.
- Arquitectura: será abierta, de manera que se puede añadir posteriormente funcionalidad adicional.

- Software: utilizará tecnologías abiertas de programación web y software de dominio público.
- Documentación: deberá incluirse manual de usuario y de instalación.
- Plataforma: deberá poder ejecutarse sobre el servidor jair.lab.fi.uva.es
- Control: se realizará un análisis y seguimiento de riesgos.
- Requisitos: la funcionalidad de la aplicación debe cubrir todos los requisitos del enunciado de la práctica.

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 al término del proceso podríamos tener una versión definitiva y completa 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. Esto será indicado más adelante cuando se presenten los objetivos de cada iteración.

Los principales entregables que se gestionarán durante el proceso de desarrollo son los siguientes:

Plan de Desarrollo Software (SPMP) Incluye la planificación detallada del proyecto, así como el análisis de riesgos, visión general del proyecto, asignación de roles, etc.

Documento de Especificación de Requisitos (SRD) Se recogen los requisitos funcionales y no funcionales del sistema, clasificados en función de su naturaleza.

Documento de Análisis del Proyecto Ofrece una definición arquitectónica del sistema, así como la estructuración del mismo en subsistemas. Incluirá los diagramas necesarios de casos de uso y clases de análisis.

Prototipo de Interfaz Se desarrollará un prototipo con el fin de definir y validar con el cliente la interfaz del sistema.

Documento de Diseño Detallado Incluye el diseño arquitectónico, de modelo de datos y de componentes. Incluirá todos los diagramas necesarios en dichas descripciones.

Documento de Plan de Pruebas Define detalladamente el procedimiento a seguir para probar el sistema.

Solución Software El sistema finalmente implementado y listo para su instalación.

Manual de Usuario Describe el modo de empleo de la aplicación web desde el punto de vista del usuario final.

Manual de Administrador Describe el modo de empleo de la aplicación web desde el punto de vista del administrador.

Manual de Instalación Detalla cómo debe desplegarse el sistema para su correcta utilización.

Documento de Implementación Describe someramente las tecnologías empleadas en la implementación del sistema.

La lista completa de entregables del proyecto, que serán entregados junto con el código fuente del proyecto, son las siguientes:

- Documentación.
- Plan de Proyecto.
 - Plan de desarrollo software.
 - Planes de iteración.
 - Ordenes de trabajo.
 - Registros de revisión.
- Especificación de Requisitos del Sistema.
 - Especificación anexa.
 - Modelo de Casos de Uso.
 - Especificación de Casos de Uso.
 - Glosario.
 - Prototipo de la interfaz.
- Documentación de Análisis.
 - Modelos y métodos.
 - Diagramas UML.
- Documentación de Diseño Software.

- Estructuras de Datos.
 - Acceso a la Base de Datos.
 - Diagramas UML.
- Documentación de Desarrollo.
 - Modelo de implementación.
 - Componentes.
 - Herramientas de Desarrollo.
- Documentación de Pruebas.
 - Plan de pruebas.
 - Resultados de las pruebas.
 - Test de componentes.
 - Documento de evaluación de pruebas.
- Gestión de configuraciones.
 - Plan de Gestión de Configuraciones.
 - Espacio de trabajo.
 - Informes de cambios.
- Manuales.
 - Manual de usuario
 - Manual de desarrollo
- Documento final de estudio.
- Solución Software.

En la figura 2.1 se muestran las fechas de generación de cada entregable del proyecto. Cada uno coincide con la consecución de un hito, que marca el final de una tarea.

2.4 Evolución del Plan.

La naturaleza reducida del grupo de trabajo facilita sobremanera el seguimiento del desarrollo de las actividades del plan. El plan se considera un documento dinámico que muestra las estrategias y contenidos del proyecto, por lo que se actualizará y revisará (por defecto) al final de cada iteración y cuando sea necesario de forma imprevista.

Tabla 2.1: Fechas previstas de generación de entregables.

ITEM ENTREGABLE	FECHA
Plan de desarrollo Software (este documento)	Diciembre, 08, 2008
Especificación de Requisito	Diciembre, 08, 2008
Documento de análisis	Diciembre, 19, 2008
Documento de diseño software	Enero, 08, 2009
Documentación de desarrollo	Enero, 20, 2009
Plan de validación del software	Enero, 20, 2009
Actualización SPMP	Enero, 29, 2009
Plan de pruebas de unidades	Diciembre, 30, 2008
Plan de pruebas de integración	Diciembre, 30, 2008
Documentación de pruebas	Enero, 19, 2009
Documento final de estudio	Enero, 30, 2009

Toda actualización o cambio será notificado a los miembros del grupo mediante correo electrónico. Cuando se produzca una modificación, ésta será registrada en el historial de cambios, actualizándose el número de versión del documento y añadiendo el autor de dicha modificación.

Cuando este plan inicial sea completado se creará una línea base del proyecto, sobre la que se desarrollarán los posteriores cambios. No se creará otra línea base si el ámbito del proyecto no sufre cambios significativos.

El plan inicial de proyecto se considerará obtenido tras la aprobación definitiva del profesor de la asignatura en la fecha acordada en (

2.1), y después se deberán seguir las fechas y planes establecidos en él.

Capítulo 3

Organización del Proyecto.

3.1 Estructura de la Organización.

Debido al pequeño equipo de personas encargado del proyecto y para conseguir el máximo aprovechamiento de los recursos disponibles cada miembro puede desempeñar varios roles distintos, aunque no a la vez. El gestor de proyecto es el encargado de recoger los informes de actividad de los otros roles y hacer el seguimiento de este plan de proyecto.

En la siguiente diagrama se muestran los roles (sin miembros del equipo asignados) necesarios para que el equipo desarrolle el proyecto, así como las relaciones con el gestor de proyecto principal.

Cuando haya necesidad de más de un miembro del mismo rol, uno de ellos tomará las funciones de líder para hacer de interfaz con el gestor de proyecto cuando sea necesario.

3.2 Interfaces Externas.

El equipo de desarrollo de SEPROSO interactuará activamente con dos entidades ajenas a la práctica:

- El **profesor de prácticas** que ejercerá el rol de cliente de la aplicación, por tanto los ingenieros de requisitos deberán entrevistarse con él. Se mantendrán reuniones periódicas para informar del progreso del proyecto y discutir posibles cambios y refinamientos de los requisitos iniciales.
- El segundo actor serán los **técnicos del laboratorio** de la escuela ETSI Informática que nos proporcionarán el soporte y restricciones para su posterior desarrollo. Los técnicos del laboratorio también

tendrán un papel importante cuando surjan problemas en el despliegue de la práctica.

3.3 Roles y Responsabilidades.

Las tres etapas principales donde clasificamos los roles son la de Análisis, Diseño e Implementación. En nuestro equipo, y sin asociar con nombres, este es el listado de roles y sus responsabilidades asociadas:

- **Gestor de Proyecto:** estudia inicialmente el proyecto, estableciendo el plan de proyecto y el ámbito del mismo. Supervisa todas las fases del proyecto establecidas en el plan inicial para controlar que se cumplan las actividades en el tiempo establecido y con el coste indicado. 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 el responsable de redactar el Plan de Gestión de Configuraciones, de gestionar e informar de las Peticiones de Cambio.
- **ANÁLISIS:**
 - **Analista:** realiza entrevistas para obtener nuevos requisitos y necesidades del cliente o completar los existentes. Determina cuando 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

Tabla 3.1: Roles asignados a miembros del equipo

ROL	Rubén.	Fran.	Yuri.	Abel.
Gestor de Proyecto	X	X	X	X
Analista		X		X
Revisor	X		X	
Ingeniero de pruebas			X	X
Ingeniero de bases de datos	X	X		
Diseñador Software			X	X
Programador	X	X		
Gestor de configuraciones	X			X

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.

En la tabla 3.1 se establecen los miembros del equipo a los que se pueden asignar cada uno de los roles definidos y en 3.2 se ilustran la distribución de responsabilidades entre los diferentes roles .

Tabla 3.2: Responsabilidades asignadas a cada rol

ROL	Requisitos.	Análisis.	Diseño.	Implementación.	Integración.	Pruebas.	Gestor Cambios.	Gestor config.	Revisiones.	Cualquiera.
Gestor de Proyecto								X	X	X
Analista	X	X								
Revisor		X					X		X	
Ingeniero de pruebas					X	X				
Ingeniero de bases de datos			X							
Diseñador Software			X	X						
Programador				X						
Gestor de configuraciones							X	X		

Capítulo 4

Proceso de Gestión.

4.1 Resumen de la planificación y el presupuesto.

El presupuesto en este proyecto no existe pues se trata de una actividad académica sin ánimo de lucro.

Se desarrollará el proyecto siguiendo el modelo de Proceso Unificado dirigido por los casos de uso e intentado distribuir el trabajo equitativamente durante el desarrollo de la herramienta. El planteamiento de las iteraciones se detalla en el punto 6.1. El plan de proyecto se puede consultar en el punto 5.1.

A un alto nivel, el proyecto tiene la planificación mostrada en la tabla 4.1.

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 en a lo largo de la carrera.

Tabla 4.1: Resumen de las etapas del plan de desarrollo.

ETAPA	INICIO	DURACIÓN
Planificación	03-11-2008	1 mes
Análisis	11-11-2008	28 días
Diseño	18-12-2008	15 días
Implementación	26-12-2008	25 días
Pruebas	26-12-2008	1 mes
Implantación	25-01-2008	Varios días

4.2 Plan de Proyecto.

4.2.1 Plan de fases.

La duración del proyecto constará de 14 semanas, sin embargo 3 de ellas corresponden al período navideño y el trabajo en dichas semanas disminuirá, aunque se seguirá con el trabajo en un número menor de días laborables.

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

1. **Inicio:** define el ámbito del proyecto y los recursos iniciales, planifica las actividades del proyecto teniendo en cuenta los requisitos, y elaborando el primer esbozo del modelo de casos de uso. El resultado de esta fase es el plan de desarrollo software.
2. **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 del plan de pruebas.
3. **Construcción 1:** se procede a la implementación de las funcionalidades más importantes y proseguir con la actividad de diseño del sistema, se desarrolla la base de datos y se completa la arquitectura de sistema. Se realizarán pruebas de los componentes implementados. Completar el diseño del plan de pruebas, con nuevos casos de prueba necesarios.
4. **Construcción 2:** se completará 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.
5. **Transición:** se llevarán a cabo las pruebas de integración, y la redacción completa del manual de usuario. Obtendremos un producto estable.

En la imagen anexa 4.1 se muestra un diagrama reducido donde se delimita la duración de estas iteraciones.

4.2.2 Objetivos de las iteraciones.

La tabla que se indica a continuación 4.2 muestra los objetivos y los hitos que han de realizarse en cada iteración.

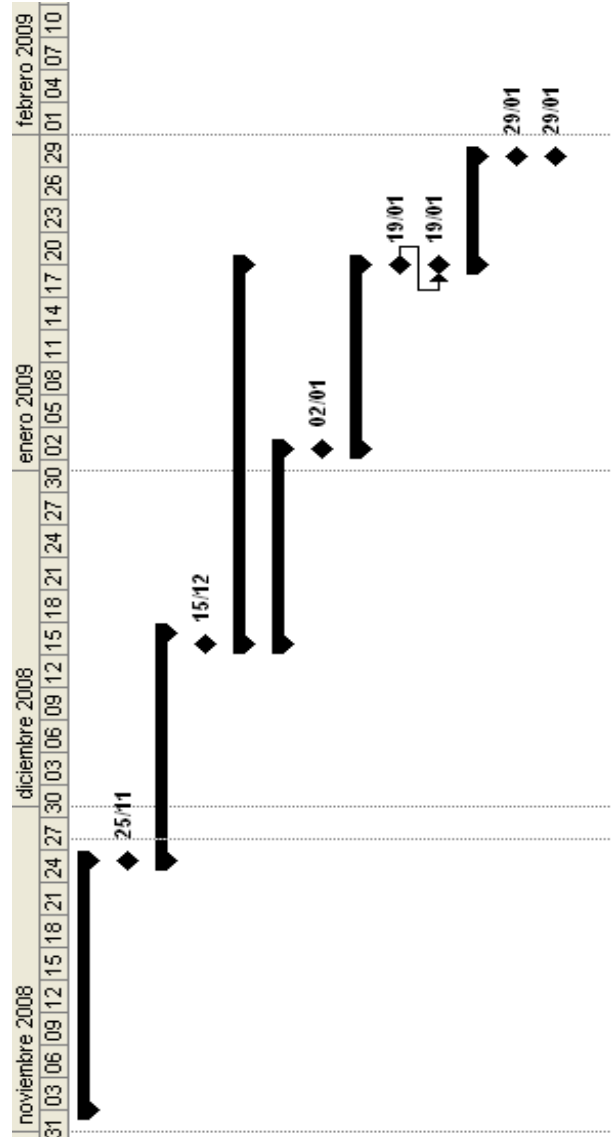


Figura 4.1: Duración de la planificación de las iteraciones.

Tabla 4.2: Hitos y objetivos de las iteraciones.

ITERACION	OBJETIVO	HITOS
Inicio	Realizar la planificación	Plan de Proyecto
Elaboración	Análisis de Casos de Uso y Requisitos.	Documento de arquitectura SW.
Construcción 1	Implementar funciones principales.	Versión Alfa
Construcción 2	Pruebas de la versión alfa. Implementación de toda la funcionalidad.	Versión Beta
Transición	Pruebas beta. Finalizar documentación. Entrega.	Producto final

4.2.3 Plan de iteración.

El plan de iteración para cada una de las etapas de desarrollo, se recogen en los documentos de iteración.

4.2.4 Calendario del proyecto.

El calendario laboral será el mismo para todo el personal del proyecto. Se tendrán en cuenta los días festivos y las vacaciones de fin de año. El calendario laboral será: lunes a viernes de 16:00 a 18:00 y sábados de 10:00 a 12:00. El resumen de las características es el siguiente:

- 2 horas diarias.
- 12 horas semanales.
- 20 días al mes.

4.2.5 Recursos del proyecto.

La definición de recursos se resume en la siguiente tabla 4.3. Tendremos en cuenta que cada rol necesario no se corresponde con una persona diferente en la realidad, una misma persona puede asumir varios roles simultáneamente.

4.3 Monitorización y control.

4.3.1 Gestión de Requisitos.

Los requisitos se encuentran recogidos y clasificados en los documentos SRS y SSS. A cada requisito se le asocian atributos como importancia, estado, etc, que permiten realizar un seguimiento de cada uno.

Tabla 4.3: Recursos del proyecto.

ITERACION	RECURSOS	NÚMERO
Inicio	Gestor de Proyecto.	3
	Analista.	1
	Revisor.	1
Elaboración	Gestor Proyecto	1
	Analista.	2
	Revisor.	1
	Gestor de configuraciones.	1
Construcción	Gestor Proyecto	1
	Analista.	1
	Ingeniero de bases de datos.	2
	Diseñador Software.	2
	Programador.	2
	Ingeniero de pruebas	2
	Gestor de configuraciones.	1
	Revisor.	1
Transición	Gestor Proyecto	1
	Gestor de configuraciones	1
	Revisor.	1

Los cambios en algún requisito serán gestionados mediante una Solicitud de Cambio, la cual debe distribuirse al equipo y evaluarse para conseguir la integridad del sistema y el buen funcionamiento del proceso de gestión de configuraciones.

4.3.2 Control de Calidad.

Todos los artefactos pasarán por un proceso de revisión por parte de todos los miembros del equipo. Los defectos detectados durante la revisión se formalizarán mediante una Solicitud de Cambio tendrán un seguimiento posterior para asegurar la solución de esas deficiencias. Para la revisión se utilizan las líneas guía y "checklist" de UPEDU que garantizan que todos los artefactos tienen una aceptable calidad.

4.3.3 Informes y mediciones.

Cada semana todos los miembros del equipo deben realizar un informe de actividades realizadas que entregan al jefe de proyecto. Se empleará esfuerzo y tiempo planificado para llevar a cabo el seguimiento del proyecto.

El jefe de proyecto compara la situación actual con la planificada para medir el progreso del proyecto y actualizar el plan de desarrollo si fuese necesario.

4.3.4 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, realizando al menos una comprobación al final de ella.

La lista inicial, así como todas las revisiones y cambios están documentados en la **Lista de Riesgos**.

4.3.5 Gestión de Configuraciones.

Elegiremos la herramienta más apropiada para el control del repositorio de versiones de todos los artefactos del proyecto. Estos incluyen: código fuente, documentación y datos. Todos ellos se incluyen en líneas base del proyecto (registros del estado de artefactos, una versión concreta). La documentación del código fuente también se incluye como documentación específica de diseño. También se incluirá la gestión de las Solicitudes de Cambio y de las modificaciones que ocasionen, notificando e informando de

ellas a todos los miembros del equipo.

Al final de cada iteración se creará una línea base de referencia para la siguiente iteración que incluya también todos los artefactos a entregar al cliente y que sólo podría ser alterada por alguna Solicitud de Cambio formalmente aprobada.

Apéndice A

Anexos