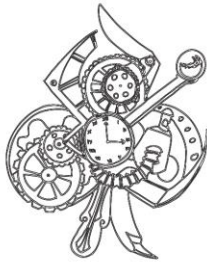


SISTEMA DE ADMINISTRACIÓN DE CASINOS

- Plan de Proyecto -



Pablo Esteban Deltell

Miguel Sánchez-Brunete Álvarez

Jonathan Carrero Aranda

Ingeniería del Software

Facultad de Informática

Universidad Complutense de Madrid



ÍNDICE

1.	INTRODUCCIÓN.....	3
1.1.	Propósito del plan.....	3
1.2.	Ámbito del proyecto y objetivos	3
1.2.1.	Declaración del ámbito	3
1.2.2.	Funciones principales	3
1.2.3.	Aspectos de rendimiento.....	3
1.2.4.	Restricciones y técnicas de gestión.....	4
1.3.	Modelo de proceso.....	4
2.	ESTIMACIONES DEL PROYECTO.....	5
2.1.	Datos históricos.....	5
2.2.	Técnicas de estimación.....	5
2.3.	Estimación de esfuerzo, coste y duración	5
3.	ESTRATEGIA DE GESTIÓN DEL RIESGO.....	8
3.1.	Análisis del riesgo.....	8
3.2.	Estudio de los riesgos.....	9
3.3.	Plan de gestión del riesgo.....	9
3.3.1.	Incumplimiento de la planificación temporal	9
3.3.2.	La falta de comprensión de los conceptos de la asignatura.....	11
4.	PLANIFICACIÓN TEMPORAL.....	12
4.1.	Estructura de descomposición/Planificación temporal	12
4.2.	Gráfico de Gantt	12
4.3.	Red de tareas	12
4.4.	Tabla de uso de recursos.....	12
5.	RECURSOS DEL PROYECTO	13
5.1.	Personal.....	13
5.2.	Hardware y Software.....	13
5.3.	Lista de recursos.....	13
6.	ORGANIZACIÓN DEL PERSONAL	15
6.1.	Estructura de equipo	15
6.2.	Informes de gestión	15
7.	MECANISMOS DE SEGUIMIENTO Y CONTROL	16

7.1.	Garantía de calidad y control.....	16
7.2.	Gestión y control de cambios.....	16
8.	Apéndice	17

1. INTRODUCCIÓN

1.1. Propósito del plan

Este documento recoge el plan de proyecto del software de la aplicación SAC (Sistema de Administración de Casinos).

En él están descritos las estimaciones del proyecto, la gestión del riesgo, la planificación temporal, los recursos del proyecto, la organización del personal y los mecanismos de seguimiento y control.

1.2. Ámbito del proyecto y objetivos

El software ha sido pensado para ser implementado en el contexto de la asignatura de Ingeniería del Software, y que, de llevarse a cabo, gestionaría el funcionamiento de un Casino (partidas en distintas mesas, crupieres a cargo de dichas partidas, apuestas de los jugadores, etc.).

El objetivo principal del proyecto será la aplicación de técnicas de Ingeniería del Software para la asignatura que lleva el mismo nombre.

1.2.1. Declaración del ámbito

El ámbito de este software serían los diferentes tipos de apuestas que pueden llegar a realizarse en un mismo casino, donde los usuarios serían el cliente que accede a las instalaciones.

La aplicación abarca la gestión de un determinado casino, así como el personal que trabaja en él, sus horarios, sus funciones en cada juego, sus jefes y los clientes a los que atienden en un instante concreto.

1.2.2. Funciones principales

Las principales funciones de este programa son la gestión de la información de las distintas partidas que se están ejecutando en un determinado momento, la información sobre la misma referente a las apuestas, la gestión de los jugadores activos y la gestión de los horarios en los que transcurren repartiéndose entre las distintas mesas de un mismo juego.

1.2.3. Aspectos de rendimiento

La aplicación debería proporcionar un rendimiento cercano al 100% debido a que está gestionando información sensible y que requiere un alto grado de confidencialidad. Información como horarios de trabajo, responsables o dinero que se maneja en las apuestas.

1.2.4. Restricciones y técnicas de gestión

El documento de especificación de requisitos y el plan de proyecto software deben ser entregados el 27 de enero de 2015 y el proyecto final deberá ser presentado el 30 de mayo.

El número de componentes del equipo debe ser entre dos y cuatro personas y el lenguaje de programación debe ser Java o C++, además de que las técnicas aplicadas a nivel técnico y de gestión deben ser las de Ingeniería del Software.

1.3. Modelo de proceso

Debido a las características del proyecto y a la reducción de riesgos el modelo de proceso utilizado para este proyecto es RUP (Proceso unificado de desarrollo) que se caracteriza por estar dirigido por casos de uso, naturaleza iterativa e incremental y centrado en la arquitectura.

2. ESTIMACIONES DEL PROYECTO

2.1. Datos históricos

Debido a que éste es el primer proyecto software en el que los componentes del equipo hemos aplicado Ingeniería del Software, el proyecto carece de datos históricos en los que basarnos para realizar las estimaciones pertinentes.

2.2. Técnicas de estimación

Existen multitud de técnicas de estimación, como por ejemplo: retrasar la estimación lo máximo posible, la estimación por analogía (proyectos similares ya terminados), la ley de Parkinson, el precio para ganar, las técnicas de descomposición y los modelos empíricos.

Las técnicas de descomposición son dos: descomposición basada en el problema, que descompone el producto en funciones y estima el tamaño del software en base a las líneas de código (LDC) o en base a su funcionalidad (PF), y basada en el proceso, que identifica un conjunto pequeño de actividades de trabajo o tareas de trabajo y estima el esfuerzo requerido para llevar a cabo cada tarea. Esta última, es la técnica más común para estimar un proyecto.

En nuestro proyecto, dada la escasez de recursos de estimación de los que contamos debido a la inexperiencia del equipo en Ingeniería del Software y a la falta de datos históricos con los que poder estimar de manera fiable, se llevará a cabo únicamente la técnica de descomposición basada en el proceso, que por tanto estará enteramente basada en la planificación temporal del proyecto, realizada previamente y que se encuentra en el **punto 4.1**.

2.3. Estimación de esfuerzo, coste y duración

El proyecto se iniciará el 16 de febrero y terminará el 30 de mayo.

Se pretende que una vez comenzado el desarrollo del proyecto, se dediquen tres días semanales y, a su vez, unas 2 horas por día. Lo que hace un total de 33 días o 66 horas. Realmente podría dedicarse más tiempo al proyecto y finalizarlo en un menor plazo, pero esta opción no es viable. En primer lugar por cómo está compuesta la asignatura y sus plazos de entrega, y en segundo lugar porque el equipo de trabajo no dispone de un tiempo total para volcarse en este proyecto, pues se entiende que cursamos más asignaturas. Aún con esto, consideramos que el tiempo es suficiente para cumplir los objetivos.

Respecto al coste, el proyecto no conlleva ningún coste monetario para su realización, por lo que el coste del proyecto será únicamente el asociado al esfuerzo.

En cuanto al esfuerzo, el proyecto se desarrollará de forma incremental y modular, por lo que realizaremos el cálculo del esfuerzo analizando cada módulo por separado.

Aquí se nos presenta una ventaja muy significativa y es que el tiempo de dedicado para cada módulo es el mismo o muy parecido. Por lo que vamos a dividir nuestro tiempo total en cuatro partes, cada una de ellas hace referencia a cada módulo. Además, las personas que formamos el equipo de trabajo dedicarán su tiempo a realizar un módulo. Una vez terminado y evaluado dicho módulo, se pasará al siguiente y así hasta finalizar los cuatro.

Módulos

- I. Módulo Gerencia: al ser el módulo con más responsabilidad, es el último que se llevará a cabo. Primero han de hacerse todos los que estén por debajo y después, el módulo de gerencia controlará esos tres módulos que están en una jerarquía inferior.

El esfuerzo requerido para este módulo es de 15 horas. En esas 15 horas, las tres personas que formamos el proyecto, nos dedicaremos en la totalidad del tiempo a completar este módulo. Por lo tanto, el esfuerzo por cada persona es de 15 horas x 3 personas = 45 horas empleadas en este módulo.

- II. Módulo Jefe de juego: este módulo engloba la gestión de las mesas por parte de los jefes de juego. Es el tercer módulo en ser completado, pues depende de que se hayan finalizado los juegos para poder realizar la gestión de los mismos.

El esfuerzo requerido para este módulo es de 15 horas. Al igual que en el anterior, el esfuerzo que dedica cada persona es de 15 horas x 3 personas = 45 horas para finalizar con el módulo.

- III. Módulo Crupier: aquí es donde entran los crupieres. Se controlan los horarios de apuestas y, por lo tanto, los horarios en los que los jugadores pueden hacer juego. Es el segundo módulo que se llevará a cabo, pues en primer lugar es necesario gestionar las apuestas para después implementar esas apuestas a cada juego.

El esfuerzo requerido para completar este módulo es de 15 horas para cada uno de los integrantes del equipo. 15 horas x 3 personas = 45 horas totales.

- IV. Módulo Jugador: lleva el control de las apuestas por parte de los jugadores, es decir, es el módulo encargado de interaccionar entre los usuarios que se registran y el casino.

El esfuerzo requerido para finalizar este módulo es de 15 horas en la que cada persona dedicará 15 horas y que, al igual que los módulos anteriores, tendrá una dedicación de 45 horas de esfuerzo.

Como conclusión, los módulos se finalizan de forma correlativa y observamos el total esfuerzo que se invierte en la realización del proyecto:

$$4 \text{ módulos} \times 15 \text{ horas} \times 3 \text{ personas} = 180 \text{ horas de esfuerzo total}$$

Además, las 6 horas restantes se dedicarían a la revisión total del proyecto para perfilar los últimos detalles y ver que todo ha salido tal y como se hubo estimado.

3. ESTRATEGIA DE GESTIÓN DEL RIESGO

Para nuestro proyecto software aplicaremos una estrategia proactiva de gestión del riesgo. Entre las características de esta estrategia están la identificación de riesgos potenciales antes que el comienzo de los trabajos técnicos y la creación de un plan de gestión del riesgo que comentaremos en el **punto 3.3**.

3.1. Análisis del riesgo

En esta sección identificaremos los posibles riesgos junto a la probabilidad y consecuencias que tienen cada uno mediante la tabla SQAS-SEI.

- I. Abandono de uno de los miembros del equipo
Probabilidad: Remota
Consecuencias: Crítica
- II. Incumplimiento de la planificación temporal
Probabilidad: Frecuente
Consecuencias: Seria
- III. Falta de comprensión de los conceptos de la asignatura
Probabilidad: Ocasional
Consecuencias: Crítica
- IV. El programa es poco mantenible
Probabilidad: Remota
Consecuencias: Catastrófica
- V. Dificultades en la implementación
Probabilidad: Ocasional
Consecuencias: Seria
- VI. Rendimiento de los miembros del equipo menor al esperado
Probabilidad: Remota
Consecuencias: Seria
- VII. Lentitud en la toma de decisiones

Probabilidad: Ocasional

Consecuencias: Seria

VIII. Aparición de propiedades erróneas

Probabilidad: Remota

Consecuencias: Catastrófica

IX. Pérdida de datos de diseño e implementación del software

Probabilidad: Improbable

Consecuencias: Catastrófica

3.2. Estudio de los riesgos

En primer lugar calculamos el nivel de riesgo de cada uno de los riesgos mencionados en el apartado anterior según la SQAS-SEI.

	Frecuente	Probable	Ocasional	Remota	Improbable
Catastrófica				Riesgo IV Riesgo VIII	Riesgo IX
Crítica				Riesgo III	Riesgo I
Seria	Riesgo II		Riesgo V Riesgo VII	Riesgo VI	
Menor					
Insignificante					

3.3. Plan de gestión del riesgo

3.3.1. Incumplimiento de la planificación temporal

Evitar el riesgo es inviable puesto que es la primera vez que realizamos una planificación temporal de un proyecto software y no podemos elegir una alternativa de menor riesgo. Por otro lado, no estamos en condiciones de asumir el riesgo puesto que tendría unas consecuencias críticas de cara al proyecto. Así, por descarte, nos hemos decantado por controlar el riesgo.

» Reducción:

A continuación planteamos ciertas técnicas de reducción o planes de contingencia que hemos considerado para controlar este riesgo.

Por un lado se reutilizará software para así ahorrarse el tiempo que llevaría a cabo implementar ese código. Aunque, se ha de tener en cuenta que, debido la inexperiencia del equipo, esta técnica es poco viable. Esto se debe a que no se cuenta con proyectos anteriores. No obstante, no hemos podido incluir como técnica de reducción la realización de una planificación temporal exhaustiva debido a la falta de datos históricos y la inexperiencia del equipo tal y como comentábamos anteriormente.

Otra estrategia que usaremos para reducir este riesgo será el desarrollo del proyecto de un modo incremental e iterativo. Esto reducirá el riesgo de incumplir la planificación temporal y permitirá mitigar desde el inicio los riesgos del proyecto.

Desde la primera iteración el equipo tiene que gestionar los problemas que pueden aparecer en una entrega del proyecto. Al hacer patentes estos riesgos, es posible iniciar su disminución de manera anticipada.

» Supervisión:

El control y supervisión de este riesgo se hará mediante reuniones periódicas entre los miembros del grupo, en las que se controlará que todos han realizado de manera satisfactoria las tareas que les han encomendado.

Además, se hará un seguimiento exhaustivo mediante el cumplimiento de hitos y el control de fechas clave a partir de las cuales deben estar realizadas ciertas actividades estructurales. Por último también se analizarán y se tendrán en cuenta las opiniones subjetivas de los componentes del equipo a lo largo del proyecto.

» Gestión:

A continuación explicamos las soluciones que hemos propuesto para reducir el riesgo en el hipotético caso de que éste se convierta en realidad.

En primer lugar, se incrementaría el esfuerzo de los componentes del equipo para no empeorar más aun la falta de seguimiento de la planificación temporal.

También hemos considerado priorizar el desarrollo de ciertas partes del producto para asegurar que las funcionalidades principales de la aplicación puedan ejecutarse de manera correcta.

3.3.2. La falta de comprensión de los conceptos de la asignatura

En este caso, nos centraremos en intentar evitar el riesgo, ya que el control de éste es inviable al sostenerse el proyecto completo sobre estos conocimientos.

» Reducción:

Para reducir la exposición a este riesgo hemos considerado ciertas técnicas que expondremos a continuación.

La asistencia a clases por parte de todos los miembros del equipo será fundamental para la comprensión de la materia y, por tanto, para minimizar la posibilidad de que este riesgo se vuelva real.

Además, para reforzar los conocimientos en la asignatura se obtendrá una bibliografía complementaria a los apuntes de clase y se acordarán tutorías para poder repasar los fundamentos del proyecto.

» Supervisión:

El control y supervisión de este riesgo se hará mediante reuniones entre los componentes del equipo para así evaluar los conocimientos y supervisar que están fijados de manera adecuada.

» Gestión:

La única posible solución si este riesgo se volviese realidad sería un aumento del esfuerzo. Este esfuerzo estaría enfocado a repasar los conceptos de la asignatura y asegurar su comprensión para su correcta aplicación en el proyecto.

4. PLANIFICACIÓN TEMPORAL

4.1. Estructura de descomposición/Planificación temporal

Para acceder a la estructuración de descomposición, planificación temporal, Gantt de seguimiento, escalas de tiempo, hoja de tareas, uso de recursos y diagrama de red, acceder al documento [*Microsoft Project*](#).

4.2. Gráfico de Gantt

Para acceder a la estructuración de descomposición, planificación temporal, Gantt de seguimiento, escalas de tiempo, hoja de tareas, uso de recursos y diagrama de red, acceder al documento [*Microsoft Project*](#).

4.3. Red de tareas

Para acceder a la estructuración de descomposición, planificación temporal, Gantt de seguimiento, escalas de tiempo, hoja de tareas, uso de recursos y diagrama de red, acceder al documento [*Microsoft Project*](#).

4.4. Tabla de uso de recursos

Para acceder a la estructuración de descomposición, planificación temporal, Gantt de seguimiento, escalas de tiempo, hoja de tareas, uso de recursos y diagrama de red, acceder al documento [*Microsoft Project*](#).

5. RECURSOS DEL PROYECTO

5.1. Personal

Debido a la escasez de personal, todos los componentes del equipo llevarán a cabo los diferentes tipos de tareas del proyecto sin especializarse en ninguna en concreto. Además, como se menciona en el **punto 6.1**, al ser una estructura de equipo descentralizado democrático cada persona del equipo no tiene una posición concreta en la organización, si no que estas posiciones van surgiendo de manera natural a medida que se desarrolla el proyecto.

El personal que desarrollará el software es el siguiente:

- ✓ Pablo Esteban Deltell
- ✓ Miguel Sánchez-Brunete Álvarez
- ✓ Jonathan Carrero Aranda

Dicho personal es necesario para desarrollar el proyecto durante todas las etapas del mismo.

5.2. Hardware y Software

El hardware disponible es el proporcionado por la Universidad Complutense de Madrid y los equipos propios de los componentes del equipo. Este hardware ha sido y va a ser utilizado a lo largo de todo el proyecto.

En cuanto al software disponible contamos con *Eclipse*, *Project 2013*, *Sublime Text 2* y *MySQL*.

Eclipse será utilizado a partir del 27 de enero de 2015 hasta el final del proyecto para el desarrollo del código. *Microsoft Project 2013* ha sido utilizado desde diciembre de 2014 hasta la primera entrega, el 27 de enero de 2015, para la creación y edición de la planificación temporal.

Para la edición del diseño de la aplicación web usaremos distintos programas entre ellos destacar Sublime Text 2 (editor de texto con herramientas o ayudas para la programación), que será utilizado sobre todo para el HTML5 y el CSS.

La gestión de la base de datos se llevara a cabo con MySQL (a través de PhpMyAdmin). Ahí se almacenara las cuentas de los clientes y más información.

5.3. Lista de recursos

A día de hoy disponemos de todo lo necesario para poder continuar y finalizar el proyecto SAC. Nuestra aplicación web requiere de los siguientes recursos:

- Ordenadores portátiles donde desarrollar el proyecto.
- Eclipse Luna.
- Sublime Text 2.
- MySQL con PhpMyAdmin.
- Servidor para el alojamiento de la web.
- Algún disco duro externo que nos sirva como salvador en caso de que quisiéramos volver a una etapa anterior (por algún tipo de fallo, compatibilidad, modificación durante la realización de los módulos, etc).

6. ORGANIZACIÓN DEL PERSONAL

6.1. Estructura del trabajo de equipo

La estructura de equipo elegida para el proyecto es la de descentralizado democrático. Esta estructura es la elección más lógica debido a que la comunicación es horizontal entre los miembros del equipo y al escaso personal del que disponer y que imposibilita la formación de equipos especializados.

El modelo de esta estructura no contempla la existencia de un jefe fijo pero dependiendo de las tareas se asignará una persona del equipo que domine mejor el ámbito. En cualquier caso en este proyecto en particular consideramos que, debido a la poca experiencia del equipo, no podemos fijar de antemano dichos cargos. Así que permitiremos que cada uno de los miembros tienda a controlar de manera natural aquel ámbito donde se encuentre más cómodo. Sin embargo, asignaremos parcialmente a cada uno de los miembros del equipo un entorno para evitar un descontrol durante el desarrollo del proyecto.

6.2. Informes de gestión

Veamos en qué consisten las actividades de gestión que desarrollan cada uno de los integrantes del equipo de trabajo.

En primer lugar, tanto Pablo Esteban como Miguel Sánchez-Brunete llevan a cabo la mayor parte del diseño, codificación y pruebas de cada uno de los módulos. Concretamente, Miguel se encargará más específicamente de la parte de diseño y codificación, pues tiene buenos resultados en programación y una visión muy buena en cuanto a cómo codificar un determinado diseño. Por otro lado, Pablo pondrá su esfuerzo en las pruebas llevadas a cabo para la detección de posibles errores que pudieran surgir, por eso, tanto él como Miguel, han de estar continuamente comunicados por si hubiera que subsanar fallos tanto en el diseño como en la codificación.

Y en segundo lugar, Jonathan Carrero centrará sus esfuerzos en el análisis y la evaluación completa de los módulos, asumiendo así un papel de cierta responsabilidad a la hora de estimar si la evaluación de un módulo se está llevando a cabo conforme a la calidad que hubiéramos exigido desde el principio.

7. MECANISMOS DE SEGUIMIENTO Y CONTROL

7.1. Garantía de calidad y control

Aunque no se realizarán revisiones técnicas formales (debido a la falta de tiempo por el equipo), se realizarán pruebas de verificación y corrección de errores sobre el código y la planificación temporal.

7.2. Gestión y control de cambios

La gestión y control de cambios sobre las diferentes versiones del proyecto, se llevará a cabo a través del propio documento de Project, pues puede gestionarse a través del denominado Gantt de seguimiento (Tarea > Diagrama de Gantt > Gantt de seguimiento).

Para lo referente a la documentación del proyecto, la gestión se llevará a cabo a través de <https://www.dropbox.com>, donde el equipo de trabajo podrá crear y editar los documentos relativos a SAC.

8. Apéndice

Información referente a la documentación sobre el plan de proyecto y sobre los conceptos que se han aplicado al llevar a cabo el desarrollo del mismo:

- **Ingeniería del Software, Teoría y Práctica.** *Por Shari Lawrence Pfleeger.*
- **Ingeniería del Software, Un enfoque práctico.** *Por Roger S. Pressman.*