



Universidad de Valladolid

E. T. S. DE INGENIERÍA INFORMÁTICA

Ingeniería Técnica en Informática de Gestión

Gestor de cuadros de mando dinámicos para
múltiples modelos de simulación contruidos con
Dinámica de Sistemas

Alumnos: *José Antonio González Alejos*
Mario del Valle Guijarro

Tutor: *Pablo Sánchez Mayoral*

Resumen

La *dinámica de sistemas* es una metodología para el estudio de sistemas dinámicos complejos. Nace como respuesta al hecho de que muchos de los métodos de resolución de problemas, especialmente los relacionados con la Investigación Operativa no resultaban totalmente satisfactorios para la comprensión de los problemas estratégicos de sistemas complejos.

La dinámica de sistemas tiene por objetivo la creación de *modelos* o representaciones de sistemas del mundo real, y el estudio de su comportamiento en el tiempo. Persigue fundamentalmente la mejora de ese comportamiento. Su utilización facilita la comprensión de la relación existente entre el comportamiento del sistema en el tiempo y su estructura, estrategias, políticas y reglas de decisión.

El *cuadro de mando* consiste en la observación e identificación del comportamiento del sistema en el tiempo y en la creación de un modelo representativo del sistema, capaz de reproducir, mediante la simulación, los comportamientos observados en la realidad.

Este proyecto da un paso en la utilización de la dinámica de sistemas al permitir crear cuadros de mando partiendo de un modelo de dinámica de sistemas, de manera que el usuario interactúe con el cuadro de mando para llevar a cabo simulaciones con el modelo de dinámica de sistemas. Podremos seleccionar con qué variables vamos a simular, la duración de la simulación y con qué variables vamos a ilustrar los resultados.

La solución estará orientada tanto a la aplicación docente como empresarial.

Abstract

System dynamics is a methodology for the study of complex dynamic systems. Created in response to the fact that many of the methods of solving problems, especially those related to operations research were not entirely satisfactory for understanding the strategic issues of complex systems.

System dynamics is aimed at creating *models* or representations of real-world systems, and studying their behavior over time. Primarily seeks to improve that behavior. Its use facilitates understanding of the relationship between the behavior of the system over time and its structure, strategies, policies and decision rules.

The *scorecard* is the observation and identification of system behavior in time and in the creation of a representative model system, capable of playing, through simulation, the behaviors observed in reality.

This project takes a step in the use of dynamic systems to allow you to create dashboards based on a systems dynamics model, so that the user interacts with the control panel to carry out simulations with the model dynamics systems. We select which variables we will simulate the duration of the simulation variables and how we will illustrate the results.

The solution will be aimed at both teachers and business application.

Agradecimientos

A todas aquellas personas que han colaborado de alguna forma en esta andadura y en el desarrollo de este proyecto.

A mi abuelo por guiarme por el camino correcto a seguir y enseñarme los valores adecuados que se deben tener, humildad, honradez. Siempre te llevaré dentro de mi corazón.

A mis padres y mi hermano por apoyarme en todo momento a lo largo de todo este tiempo por apoyarme en los buenos y en los malos momentos, por el simple hecho de ser únicos.

A la familia Ortiz por tratarme como uno más de la familia y por cuidarme en todo momento, siempre serán como una segunda familia.

A nuestro tutor Pablo Sánchez Mayoral por guiarnos en el transcurso de este proyecto, por orientarnos, aportarnos ideas y sugerencias.

A compañeros de facultad que con el transcurso del tiempo se han convertido en grandes amigos, con una gran calidad humana y que han ayudado de manera desinteresada en todo momento.

A mi compañero y amigo Toño que conocí desde el comienzo de esta etapa por su sacrificio, trabajo, constancia y tesón.

Mario del Valle Guijarro

A toda mi familia, amigos y compañeros de facultad que me han apoyado a lo largo de la duración de este proyecto.

A mis padres por su apoyo, paciencia y por estar siempre ahí y ofrecerme la oportunidad de realizar esta aventura.

A mi hermana por su preocupación e interés que de alguna forma me ha ayudado.

A nuestro tutor Pablo Sánchez Mayoral por darnos la oportunidad de realizar este proyecto, por su comprensión y flexibilidad a la hora de encarar este proyecto.

A mi compañero Mario por su apoyo y comprensión en la etapa final de la carrera.

José Antonio González Alejos

Índice general

Capítulo 1. Introducción	1
1.1. Fundamentos teóricos	1
1.1.1. Dinámica de sistemas	1
1.1.2. Cuadro de mando.....	1
1.2. Antecedentes.....	2
1.3. Ámbito del proyecto	2
1.4. Objetivos del proyecto	3
1.5. Estructuración del proyecto	4
1.6. Estructura de la memoria	5
Capítulo 2. Fase de Inicio	7
2.1. Concepto de la Fase de Inicio	7
2.2. Plan de Desarrollo Software	8
2.2.1. Propósito del Plan de Desarrollo Software	8
2.2.2. Visión general del proyecto	8
2.2.3. Roles y Responsabilidades	9
2.2.4. Gestión de los procesos	9
2.2.5. Seguimiento y control del proceso	13
2.3. Plan de Gestión de Configuraciones	20
2.3.1. Propósito del Plan de Gestión de Configuraciones	20
2.3.2. Alcance de la Configuración del Software	21
2.3.3. Visión general.....	21
2.3.4. Programa de administración de la configuración.....	21
2.4. Plan de Iteración de la Etapa de Inicio.....	22
2.4.1. Propósito del Plan de Iteración de la Etapa de Inicio	22
2.4.2. Plan de Trabajo para la Etapa de Inicio	22
2.4.3. Recursos Necesarios.....	24
2.5. Plan de Iteración de la Etapa de Elaboración.....	25
2.5.1. Propósito del Plan de Iteración de la Etapa de Elaboración	25
2.5.2. Plan de Trabajo para la Etapa de Elaboración	25
2.5.3. Recursos Necesarios.....	28
Capítulo 3. Fase de Elaboración.....	29
3.1. Concepto de la Fase de Elaboración	29
3.2. Documento de Análisis	30
3.2.1. Funcionamiento de la forma de trabajo	30
3.2.2. Descripción de objetivos	31
3.2.3. Captura de requisitos	35
3.2.4. Requisitos de Información.....	35
3.2.5. Requisitos Funcionales	37
3.2.6. Requisitos No Funcionales	44
3.2.7. Identificación de actores.....	49
3.2.8. Diagrama de casos de uso.....	51
3.2.9. Descripción de Casos de Uso	52
3.2.10. Diagrama de clases de análisis	81
3.2.11. Diagrama Entidad-Relación de la base de datos.....	82
3.2.12. Modelo relacional del análisis	83

3.3. Documento de Diseño	83
3.3.1. Propósito del documento de diseño.....	83
3.3.2. Arquitectura del sistema.....	83
3.3.3. Tecnología empleada	85
3.3.4. Diagrama de clases.....	90
3.3.5. Descripción de las clases.....	91
3.3.6. Diagramas de secuencia	96
3.3.7. Diseño físico de la base de datos.....	126
3.3.8. Diagramas de estado	130
3.4. Plan de pruebas.....	131
3.4.1. Tipos de pruebas	132
3.5. Prototipos de interfaces de usuario	133
Alcance	133
3.5.1. Pantallas principales.....	133
3.5.2. Estándares gráficos	135
3.5.3. Jerarquía de navegación	136
3.6. Seguimiento de la Fase de Elaboración.....	139
3.7. Plan de Iteración de la Fase de Construcción	140
3.7.1. Plan de trabajo para la iteración	140
3.7.2. Recursos necesarios	142
Capítulo 4. Fase de Construcción.....	143
4.1. Concepto de la Fase de Construcción.....	143
4.2. Estructura de los directorios de la aplicación	144
4.2.1. Estructura de los directorios del servidor Apache	144
4.2.2. Estructura de la aplicación	145
4.3. Batería de pruebas	146
4.3.1. Pruebas en el cuadro de mando	146
4.3.2. Pruebas en el cuadro de decisiones	152
4.4. Seguimiento de la Fase de Construcción	155
4.5. Plan de Iteración de la Fase de Transición	156
4.5.1. Plan de trabajo para la iteración	156
4.5.2. Recursos necesarios	159
Capítulo 5. Fase de Transición	161
5.1. Concepto de la Fase de Transición	161
5.2. Manual de instalación.....	162
5.2.1. Introducción	162
5.2.2. Instalación de Apache Tomcat	162
5.2.3. Instalación de Vensim	164
5.2.4. Instalación del código de la aplicación en el servidor de aplicaciones.....	172
5.2.5. Instalación de los drivers de control de las bases de datos para Apache	172
5.2.6. Instalación de la base de datos	173
5.3. Restricciones y requisitos de Vensim	173
5.3.1. Restricciones en los modelos	173
5.3.2. Requisitos en el desarrollo con Eclipse	174
5.3.3. Requisitos generales.....	174
5.4. Manual de Usuario	174
5.4.1. Pantalla principal.....	174
5.4.2. Pantalla Principal Administrador	176
5.4.3. Pantalla Principal Usuario.....	190

Capítulo 6. Conclusiones	197
6.1. Conclusiones	197
6.2. Dificultades encontradas	197
6.3. Consecución de objetivos	198
6.4. Conocimientos adquiridos	199
6.5. Proposición de trabajos futuros.....	199
 Bibliografía y referencias web	 201
 Apéndice I. Glosario de términos	 205
 Apéndice II. Índice de figuras.....	 209
 Apéndice III. Índice de tablas	 213
 Apéndice IV. Contenido del CR-ROM	 217

Capítulo 1

Introducción

Este capítulo incluye la introducción, el origen y los objetivos del proyecto.

1.1. Fundamentos teóricos

1.1.1. Dinámica de sistemas

La *dinámica de sistemas* es una metodología para el estudio de sistemas dinámicos complejos. Nace como respuesta al hecho de que muchos de los métodos de resolución de problemas, especialmente los relacionados con la Investigación Operativa no resultaban totalmente satisfactorios para la comprensión de los problemas estratégicos de sistemas complejos.

La dinámica de sistemas tiene por objetivo la creación de modelos o representaciones de sistemas del mundo real, y el estudio de su comportamiento en el tiempo. Persigue fundamentalmente la mejora de ese comportamiento. Su utilización facilita la comprensión de la relación existente entre el comportamiento del sistema en el tiempo y su estructura, estrategias, políticas y reglas de decisión.

1.1.2. Cuadro de mando

El *cuadro de mando dinámico* consiste en una herramienta mediante la cual con la toma de diferentes decisiones en pocos pasos podremos realizar diversas simulaciones sobre modelos de dinámica de sistemas.

La metodología consiste en esencia en la observación e identificación del comportamiento problemático del sistema en el tiempo y en la creación de un modelo representativo del sistema, capaz de reproducir, mediante la simulación en el ordenador, los comportamientos observados en la realidad. El modelo se utilizará para diseñar políticas que produzcan una evolución del sistema deseable.

1.2. Antecedentes

Cuando comenzamos con el proyecto vimos que era una aplicación web en la cual se podía realizar simulaciones con solo un modelo. No se necesita tener un elevado conocimiento sobre la herramienta Vensim para realizar las simulaciones con dicha aplicación.

En este proyecto se presenta el diseño de un sistema que trata la simulación de modelos de dinámica de sistemas. Dicho entorno se presenta al usuario de forma transparente por medio de pantallas intuitivas y siguiendo una serie de sencillos pasos.

El usuario podrá interactuar a través de los distintos servicios del sistema. Dicho entorno mostrará de manera gráfica los resultados de las simulaciones realizadas mediante diferentes componentes gráficos.

El proyecto surge debido a la necesidad de dotar de nuevas funcionalidades al cuadro de mando dinámico entre las que se encuentra la opción de poder realizar simulaciones con diferentes tipos de modelo. La herramienta permite realizar simulaciones a través de una interfaz sencilla e intuitiva, la cual permite a través del multicontexto, que varios usuarios puedan realizar simulaciones a la vez sobre el mismo o diferentes cuadros de mando.

1.3. Ámbito del proyecto

Es una herramienta empleada para el desarrollo de prácticas de una asignatura impartida por el Departamento de Organización de Empresas y Comercialización e Investigación de Mercados de la E.T.S. Ingeniería Informática de la Universidad de Valladolid.

El ámbito de la solución va a ser el de la enseñanza entre otros por ello la herramienta va a ser manejada por profesores, alumnos, etc. Por tanto la solución dispondrá de diferentes roles, lo cual permitirá controlar la información a la que tendrá acceso cada tipo de usuario.

La herramienta está enfocada en la utilización de sistemas de simulación para la obtención de datos que nos sirvan como referencia en la toma de decisiones en un marco concreto y plazo de tiempo.

Otros de los ámbitos del proyecto es el ámbito profesional el cual vemos que es una solución muy útil para empresas de diversos sectores en los que están muy influenciados por diversas variables que a su vez toman muchos valores en el transcurso del tiempo.

Con esta solución pueden simular como afectaría a los diferentes departamentos de una empresa como pueden ser la producción, compras, ventas, almacén, etc. Y con ello tomar diferentes decisiones dependiendo de los resultados dados en la simulación.

1.4. Objetivos del proyecto

Los objetivos serán los que se pretenderán lograr tras la finalización de este proyecto.

Investigación del manejo de la herramienta Vensim DSS que será la encargada de realizar las simulaciones de los modelos. Adquirir los conocimientos necesarios para poder implementar los recursos que ofrece Vensim DSS a la aplicación web.

Desarrollo de una aplicación web en la cual mediante una serie de pasos podamos simular con diferentes modelos de dinámica de sistemas a través de diversas pantallas en las cuales se pueda seleccionar de manera sencilla los valores que deben tomar las variables con las que queremos jugar.

Realización de un apartado orientado a la administración de la aplicación y gestión de los diferentes componentes que componen la aplicación como pueden ser cuadros de mando, salas de operaciones y usuarios.

Creación de una pantalla de juego donde podremos diferenciar una primera zona donde seleccionamos el valor que van a tomar las variables que intervienen en el modelo y segunda zona donde podremos visualizar de manera gráfica los resultados de la simulación.

Creación de una base de datos para el almacenamiento de toda la información referente a la aplicación para la gestión y mantenimiento de los cuadros de mando, salas de operaciones donde se va a jugar y usuarios que van a interactuar con la aplicación por parte del usuario administrador.

Documentación de la herramienta que tratará todas las modificaciones y ampliaciones del cuadro de mando anterior, informando del ámbito de la herramienta de la que hemos partido y del ámbito que comprende la implementación de las nuevas funcionalidades.

Otro de los objetivos que no menos importante es poder solventar los contratiempos que había en el cuadro de mando en revisiones anteriores como por ejemplo tener que alojar los las librerías de las que depende Vensim en una carpeta exclusiva y conseguir que se puedan almacenar en ubicaciones diferentes a la que viene por defecto.

El uso de la aplicación web debe tener un nivel de seguridad, la utilización de la aplicación tiene que estar restringido solo a los usuarios registrados en el sistema y a los que se les haya proporcionado una clave facilitada por un administrador para poder jugar.

Alojar la aplicación en un servidor para que pueda ser accesible desde cualquier ubicación y no solo en el equipo local.

1.5. Estructuración del proyecto

Para el desarrollo de este proceso se seguirán las pautas establecidas en UPEDU (*Unified Process for Education*). UPEDU es un proceso de desarrollo de software se caracteriza por ser iterativo e incremental, estar centrado en la arquitectura y guiado por los casos de uso. Incluye artefactos (productos tangibles del proceso como por ejemplo, el modelo de casos de uso, el código fuente, etc.) y roles (papel que desempeña una persona en un determinado momento, una persona puede desempeñar distintos roles a lo largo del proceso).

El proceso de desarrollo se divide en ciclos, teniendo un producto final al terminar cada ciclo. A su vez, cada ciclo se divide en fases que culminan con la elaboración de un hito. Las fases y sus tareas de modo resumido son:

- **Fase de Inicio:** Se diseña un plan de fases, se identifican los principales casos de uso y se establecen los riesgos.
- **Fase de Elaboración:** Se diseña un plan de proyecto, se completan los casos de uso y se eliminan los riesgos.
- **Fase de Construcción:** Se concentra en la elaboración de un producto totalmente operativo y eficiente y se redacta el manual de usuario.
- **Fase de Transición:** Se implementa el producto en el cliente y se entrena a los usuarios. Como consecuencia de esto suelen surgir nuevos requisitos a ser analizados.

La elección de UPEDU como proceso de desarrollo, fue debida a que nos permite una forma disciplinada de asignar tareas y responsabilidades, ver el tiempo invertido en cada una de las tareas, comparación entre el tiempo estimado y el tiempo real.

MEJORES PRÁCTICAS DE DESARROLLO

- ✓ Gestión de Requisitos
- ✓ Desarrollo Iterativo
- ✓ Desarrollo Basado en componentes
- ✓ Automatización de pruebas
- ✓ Gestión de cambios e incidencias



Figura 1. Visión iterativa en UPEDU

Además, a través de la gestión de riesgos se pueden reconocer posibles problemas y fallos de forma temprana y prevenirlos o determinar un plan para corregirlos. Además UPEDU en cada una de sus fases realiza una serie de artefactos que sirven para comprender mejor tanto el análisis, como el diseño del sistema.

1.6. Estructura de la memoria

A continuación se resume brevemente cada uno de los capítulos que forman esta memoria para tener una idea general de cómo se ha organizado este trabajo:

- *Capítulo 1: Introducción.*

En este capítulo se presenta la introducción al proyecto, con una descripción del ámbito de actuación en el que se enmarca. De esta forma se pone al lector en situación dentro del contexto en el que se aplica. También se enumeran los objetivos a conseguir con la realización de este proyecto, se explica la estructuración del proyecto y por último se describe la organización de la memoria en este apartado actual.

- *Capítulo 2: Fase de Inicio.*

En este capítulo se detalla el plan de desarrollo del proyecto. Se describe la primera fase de la planificación que se ha seguido durante todo el proyecto.

- *Capítulo 3: Fase de Elaboración.*

Este capítulo se describe el análisis y el diseño. En la Fase de Elaboración se va buscar el establecimiento de la línea base de la arquitectura del sistema y proporcionar una base estable para el diseño y esfuerzo de implementación de la siguiente fase, mitigando la mayoría de los riesgos tecnológicos

- *Capítulo 4: Fase de Construcción.*

En este capítulo se comentan los aspectos más importantes surgidos durante la implementación de la aplicación. En esta fase se debe culminar el análisis y diseño de los requisitos que hayan quedado pendientes de la Fase de Elaboración y completar el desarrollo del sistema siguiendo la arquitectura del software ya definida.

- *Capítulo 5: Fase de Transición.*

En este capítulo se detallarán los manuales de instalación de la aplicación y herramientas utilizadas para el desarrollo de la misma. De esta forma un usuario podrá instalar el sistema a través de las indicaciones marcadas, así como poder hacer uso de éste sin problema.

- *Capítulo 6: Conclusiones*

En este capítulo se recopilan las conclusiones a las que se ha llegado una vez se ha terminado el trabajo. Conclusiones que pretenden plantear posibles líneas futuras de trabajo, así como ampliaciones del trabajo realizado.

- *Bibliografía y referencias web.*

En la bibliografía se relaciona toda la documentación utilizada para la realización de la memoria. Son libros de consulta y páginas Web a las que se podrá acudir para obtener o ampliar la información citada en este trabajo

- *Apéndice A: Glosario de términos.*

Definición de los diferentes términos técnicos, siglas y protocolos utilizados a lo largo de esta memoria. En este apéndice se pretende ayudar a entender cada uno de los términos del glosario con una descripción breve de su significado.

- *Apéndice B: Contenido del CD-ROM.*

En este apéndice final se muestra una relación con los diferentes ficheros, así como su organización en el disco adjunto al presente trabajo.

Capítulo 2

Fase de Inicio

En este capítulo se detalla la fase de inicio del proyecto, recogiendo los artefactos más importantes de esta fase.

2.1. Concepto de la Fase de Inicio



Figura 2. Hito en la Fase de Inicio

En la fase de inicio se define el modelo de negocio con sus respectivas clases y el ámbito del proyecto. Se identifican todos los actores del sistema y se diseñan los casos de uso del sistema a partir de los objetivos del proyecto. A su vez, se desarrolla un plan de negocio para determinar qué recursos deben ser asignados al proyecto.

Los objetivos de esta fase son:

- Establecer el ámbito del proyecto y sus límites.
- Encontrar los casos de uso y los escenarios básicos que definan la funcionalidad.
- Mostrar al menos una arquitectura candidata para los escenarios principales.
- Estimar el coste en recursos y tiempo de todo el proyecto.
- Estimar los riesgos y las fuentes de incertidumbre.

En esta fase se establecerán los siguientes artefactos:

- Un documento de visión: Una visión general de los requerimientos del proyecto, características clave y restricciones principales.
- El caso de negocio.
- Lista de riesgos y plan de contingencia.
- Plan del proyecto, mostrando fases e iteraciones.

Al terminar la fase de inicio se deben comprobar los criterios de evaluación para continuar:

- Todos los interesados en el proyecto, en este caso el desarrollador y el cliente, coinciden en la definición del ámbito del sistema y las estimaciones de agenda.
- Entendimiento de los requisitos, como evidencia de la fidelidad de los casos de uso principales.
- Las estimaciones de tiempo, coste y riesgo son creíbles.
- Comprensión total de cualquier prototipo de la arquitectura desarrollado.
- Los gastos hasta el momento se asemejan a los planeados.

2.2. Plan de Desarrollo Software

2.2.1. Propósito del Plan de Desarrollo Software

El propósito es dar la información necesaria para controlar el desarrollo del proyecto. Para el desarrollo de la aplicación se seguirán las pautas que están expuestas en UPEDU.

El proyecto va a ser desarrollado por dos personas por lo tanto todos los documentos irán dirigidos por ellas, los documentos podrán ser consultados en el momento adecuado según la etapa de trabajo, para ellos se han alojado dichos documentos en la red mediante la herramienta Dropbox para que puedan ser accesibles en todo momento y actualizados periódicamente.

Dichos documentos o artefactos permitirán planificar los recursos y esfuerzos necesarios para la realización del proyecto así como para entender qué hay que hacer en cada momento, cuándo se tiene que hacer y cómo se tiene que hacer.

2.2.2. Visión general del proyecto

La aplicación web dispondrá de una parte de administración en la cual se puede realizar la gestión de cuadros de mando, salas de operaciones y usuarios. Como esta aplicación va a ser utilizada por personal diferente habrá diferentes roles en los cuales cada uno de ellos podrá realizar unas determinadas operaciones u otras dependiendo de los permisos asignados.

Dispondrá de una sección para poder jugar partidas y simular con los diferentes modelos cargados en los cuadros de mando que hay en la sala de operaciones. Seguidamente ver los resultados de manera gráfica después de cada periodo de la simulación.

También se podrá recuperar partidas guardadas anteriormente para poder continuar con ellas más adelante gracias a que se almacenan todos los datos relacionados con la partida en una base de datos.

2.2.3. Roles y Responsabilidades

El desarrollo de esta aplicación web se desarrollará por los alumnos José Antonio González Alejos y Mario del Valle Guijarro. Dicha aplicación está concebida como Proyecto Fin de Carrera para la titulación de Ingeniería Técnica de Informática de Gestión.

Rol	Alumno
Jefe de Proyecto	José Antonio González - Mario del Valle
Analista	José Antonio González - Mario del Valle
Diseñador	José Antonio González - Mario del Valle
Desarrollador	José Antonio González - Mario del Valle
Probador	José Antonio González - Mario del Valle

Tabla 1. Roles y responsabilidades

2.2.4. Gestión de los procesos

2.2.4.1. Estimaciones del proyecto

En la realización de este proyecto se ha contado con dos alumnos de Ingeniería Técnica de Informática de Gestión. Para llevar a buen puerto este proyecto se han puesto todos los medios posibles tanto humanos como materiales para la correcta realización del mismo.

En cuanto a la estimación del tiempo y posible entrega del proyecto, en un principio se pensó en el mes de Mayo, pero la única restricción temporal era la impuesta por nosotros. El tiempo estimado sería de 1000 horas.

Para la calidad del producto se han realizado diversas reuniones con el cliente y un seguimiento, con un intervalo frecuente y distancia temporal corta para no desviarnos de los requisitos solicitados y aprobados por el cliente.

2.2.4.2. Plan del proyecto

En relación al coste final del proyecto, se ha intentado reutilizar el máximo código posible realizado en la versión anterior de esta misma aplicación, se ha buscado una filosofía de lograr el coste mínimo. Cualquier coste adicional que pueda generarse viene reflejado en la Lista de riesgos.

Seguidamente presentamos la organización del proyecto. Esta estará compuesta por 4 fases, las cuales contendrán las consiguientes iteraciones.

Planificación Estimada

Fase	Número de iteraciones	Comienzo de la etapa	Fin de la etapa
Fase de Inicio	1	22-10-2010	26-11-2010
Fase de Elaboración	2	29-11-2010	18-02-2011
Fase de Construcción	2	16-02-2011	02-05-2011
Fase de Transición	1	03-01-2011	15-06-2011

Tabla 2. Planificación Estimada

Planificación Real

Fase	Número de iteraciones	Comienzo de la etapa	Fin de la etapa
Fase de Inicio	1	22-10-2010	01-12-2010
Fase de Elaboración	2	02-12-2010	22-02-2011
Fase de Construcción	2	18-02-2011	17-05-2011
Fase de Transición	1	08-02-2011	07-07-2011

Tabla 3. Planificación Real

A continuación se muestra la planificación real del proyecto realizado con la herramienta Microsoft Project 2003 y su correspondiente Diagrama de Gantt:

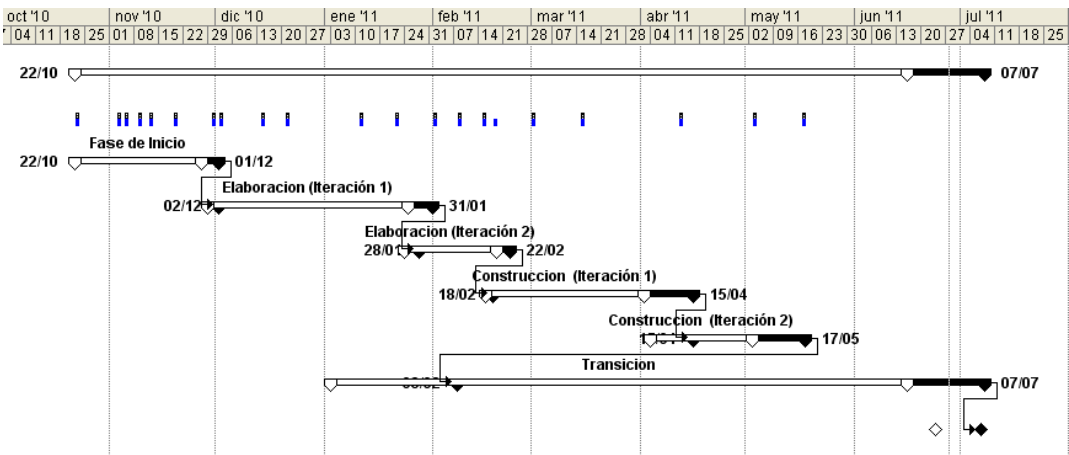


Figura 3. Diagrama de Gantt de la planificación completa

Nombre de tarea	Duración	Comienzo	Fin
<input type="checkbox"/> Proyecto Fin de Carrera	187 días	vie 22/10/10	lun 11/07/11
<input type="checkbox"/> Reuniones y Seguimiento	148 días	vie 22/10/10	mar 17/05/11
<input type="checkbox"/> Fase de Inicio	29 días	vie 22/10/10	mié 01/12/10
<input type="checkbox"/> Elaboracion (Iteración 1)	43 días	jue 02/12/10	lun 31/01/11
<input type="checkbox"/> Elaboracion (Iteración 2)	18 días	vie 28/01/11	mar 22/02/11
<input type="checkbox"/> Construccion (Iteración 1)	41 días	vie 18/02/11	vie 15/04/11
<input type="checkbox"/> Construccion (Iteración 2)	22 días	vie 15/04/11	mar 17/05/11
<input type="checkbox"/> Transicion	108 días	mar 08/02/11	jue 07/07/11
Proyecto Finalizado	1 día	lun 11/07/11	lun 11/07/11

Figura 4. Planificación Real

A continuación se muestran los diferentes hitos de cada fase, que se deben cumplir para el correcto desarrollo del plan de proyecto y transición entre fases.

Fase	Hito
Fase de Inicio	El objetivo de la fase de inicio es la puesta en marcha del proyecto. En esta fase se define el ámbito del proyecto y se inicia el análisis de negocio para el sistema hasta el punto necesario para justificar la puesta en marcha del proyecto. Otra de las actividades de esta fase es la descripción de una propuesta de arquitectura del sistema. Por último también se identifican riesgos críticos.
Fase de Elaboración	El propósito de esta fase será completar el análisis de negocio iniciado en la etapa anterior, analizar el dominio del problema y establecer una buena arquitectura que sea adecuada al proyecto. Como resultado se obtendrá un modelo del dominio y de casos de uso, así como requerimientos suplementarios que capturen los requerimientos no funcionales y cualesquiera requerimientos que no estén asociados con un caso de uso específico.
Fase de Construcción	La finalidad principal de esta fase es alcanzar la capacidad operacional del producto de forma incremental a través de las sucesivas iteraciones. En esta fase se obtiene una versión del producto llamado <i>beta</i> que puede ser entregada a usuarios. Uno de los objetivos es administrar los recursos de manera eficiente de tal forma que se optimicen los costes, los calendarios y la calidad.
Fase de Transición	En esta fase de transición se pone el producto en manos de los usuarios finales, se requerirá desarrollar nuevas versiones actualizadas del producto, completar la documentación, entrenar al usuario en el manejo del producto y en general tareas relacionadas con el ajuste, configuración, instalación y usabilidad del producto.

Tabla 4. Fases e hitos

2.2.4.3. Versiones del producto

En el transcurso del desarrollo del proyecto habrá varias versiones denominadas *beta* que presentaremos al cliente para que pueda manipularlas y como resultado sacar conclusiones de que aspectos funcionales de la aplicación que le satisfacen y cuales son susceptibles de mejorar. Aquellos aspectos que se tengan que mejorar se tendrán en cuenta en etapas futuras.

Al finalizar el proyecto se le entregará al cliente la versión final de la aplicación. Esta versión deberá cumplir todos los requerimientos solicitados por el cliente y que vienen descritos en el documento de requisitos software.

2.2.4.4. Recursos del proyecto

En la realización del proyecto se ha contado como recursos humanos a dos estudiantes de Ingeniería Técnica de Informática de Gestión, dichos alumnos desempeñarán diferentes roles a lo largo de la realización del proyecto.

Como recursos materiales se ha contado con dos ordenadores portátiles personales en los que se ha instalado todas las herramientas necesarias para poder realizar el proyecto, también se ha contado con un servidor facilitado por el cliente para poder instalar un servidor de aplicaciones web donde alojar la aplicación realizada.

Uno de los recursos más importantes a tener en cuenta es el tiempo disponible para el desarrollo del proyecto.

2.2.5. Seguimiento y control del proceso

2.2.5.1. Control de plazos

Se intentará ajustar lo máximo posible las fechas previstas con las fechas reales. Las fechas dependerán de las tareas solicitadas por el cliente, dificultad y duración de las mismas.

2.2.5.2. Gestión de requisitos

Los requisitos del sistema son especificados en la Especificación de Requisitos del Software. Los requisitos tendrán una descripción que los definen.

2.2.5.3. Control de calidad

Se realizarán diversas pruebas para asegurar la calidad del producto como por ejemplo baterías de pruebas. Los fallos detectados tendrán un posterior seguimiento.

2.2.5.4. Lista de riesgos

La gestión de riesgos en proyectos de software pretende identificar, estudiar y eliminar las fuentes de riesgo antes de que comiencen a amenazar el éxito o la finalización exitosa de un proyecto de desarrollo de software.

En la figura que se muestra a continuación ilustra el proceso de seguimiento de un riesgo en el proyecto desde su identificación.

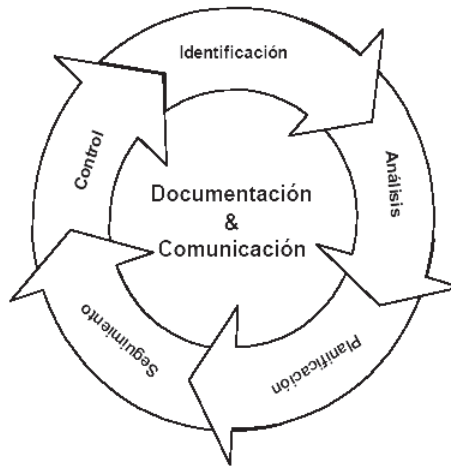


Figura 5. Modelo de administración de riesgos

Este documento es un listado de los posibles riesgos que pueden afectar en el transcurso del proyecto, dichos riesgos están ordenados de mayor a menor en cuanto al impacto en el proyecto.

A continuación describimos cada una de las características que definen a cada uno de los riesgos del proyecto.

Probabilidad

El valor que va a tomar esta característica está comprendido entre 1 y 10. Siendo 10 el riesgo más alto y 1 el riesgo más bajo. La clasificación está basada en el efecto que el riesgo ocasionaría al proyecto y la probabilidad de que este ocurra.

Descripción

En este campo se presenta una breve descripción que detalla el riesgo en cuestión.

Impacto

En este apartado se definen los distintos tipos de impactos considerados atendiendo a la gravedad con la que afectan al proyecto:

- **Crítico:** Afecta a todas las funcionalidades del proyecto y a las líneas base. Es necesario tomar una acción inmediata.
- **Alto:** Afecta a la mayoría de las funcionalidades del proyecto. Iniciar procedimientos de la planificación de riesgos.
- **Medio:** Riesgos sujetos a un plan de contingencia. Conseguir más información y realizar una revisión en la siguiente reunión.

- **Bajo:** Riesgos con un plan de contingencia de aplicación inmediata. Normalmente las decisiones se tomarán en el momento.

Indicadores

Breve descripción para detectar si ha ocurrido o si está a punto de ocurrir un riesgo.

Plan de Contingencia

Soluciones alternativas a las estrategias inicialmente previstas para reducir el efecto del riesgo lo antes posible.

A continuación mostramos la lista de riesgos conocidos.

Riesgo 1	Incertidumbre en las tareas
Probabilidad	8
Descripción	Incertidumbre a la hora de alcanzar las primeras tareas, desconocimiento de si se podrán realizar.
Impacto	Alto
Indicadores	El cliente establece unos requisitos iniciales sin saber si se podrán cumplir.
Plan de contingencia	Fijar unas tareas con certidumbre de que se pueden cumplir.

Tabla 5. Riesgo 1 - Incertidumbre en las tareas

Riesgo 2	Amplitud de los objetivos
Probabilidad	8
Descripción	La amplitud de los objetivos establecidos por el cliente puede hacer que se desborde el tiempo de desarrollo.
Impacto	Alto
Indicadores	El cliente establece unos objetivos iniciales lejanos a la realidad.
Plan de contingencia	Fijar objetivos realistas para una planificación más correcta.

Tabla 6. Riesgo 2 - Amplitud de los objetivos

Riesgo 3	Falta de conocimiento de la tecnología
Probabilidad	8
Descripción	El desarrollador necesita una formación en la tecnología que se va a utilizar en la implementación.
Impacto	Alto
Indicadores	Falta de experiencia en dicha tecnología por parte del desarrollador.
Plan de contingencia	Documentarse previamente en la tecnología descrita.

Tabla 7. Riesgo 3 - Falta de conocimiento de la tecnología

Riesgo 4	Mala planificación
Probabilidad	7
Descripción	Los desarrolladores no poseen experiencia en la planificación de proyectos.
Impacto	Alto
Indicadores	Las fechas previstas en la planificación del proyecto no se cumplen para la consecución de los diferentes hitos.
Plan de contingencia	Esforzarse en conseguir una buena planificación inicial

Tabla 8. Riesgo 4 - Mala planificación

Riesgo 5	Dependencia de la tecnología de simulación
Probabilidad	7
Descripción	Basarse solo en los métodos proporcionados por Vensim DSS para la obtención de información.
Impacto	Alto
Indicadores	El desarrollador no puede realizar determinadas operaciones si Vensim DSS no lo permite.
Plan de contingencia	Documentarse previamente de las soluciones que ofrece dicha tecnología.

Tabla 9. Riesgo 5 - Dependencia de la tecnología de simulación

Riesgo 6	Falta de concreción de algunas tareas
Probabilidad	9
Descripción	Falta de concreción por parte del cliente en la especificación de los requisitos necesarios para el desarrollo de la aplicación.
Impacto	Alto
Indicadores	Desviación entre la tarea solicitada por el cliente y la tarea desarrollada.
Plan de contingencia	Reunirse con el cliente las veces necesarias para concretar de manera correcta los requisitos.

Tabla 10. Riesgo 6 - Falta de concreción de algunas tareas

Riesgo 7	Integración de tecnologías
Probabilidad	9
Descripción	Integración entre el entorno de desarrollo Java, Vensim DSS y la BBDD, para la obtención y almacenamiento de los datos para el correcto funcionamiento de la aplicación.
Impacto	Alto
Indicadores	Intercambio de información de manera errónea.
Plan de contingencia	Documentarse de cómo pasar información de una tecnología a otra.

Tabla 11. Riesgo 7 - Integración de tecnologías

Riesgo 8	Problemas en el sistema de control de versiones
Probabilidad	9
Descripción	Problemas a la hora de poner en común el código desarrollado, sincronización de cambios, subir código al repositorio.
Impacto	Alto
Indicadores	Código resultante erróneo, falta de clases y archivos.
Plan de contingencia	Informarse de servidores de control de versiones de pago o por el contrario gratuitos que sean más fiables.

Tabla 12. Riesgo 8 - Problemas en el sistema de control de versiones

Riesgo 9	Problemas por la falta de experiencia
Probabilidad	9
Descripción	Falta de experiencia en proyectos de gran envergadura.
Impacto	Alto
Indicadores	Planificaciones erróneas.
Plan de contingencia	Asignar un proyecto de acuerdo a la experiencia de los desarrolladores.

Tabla 13. Riesgo 9 - Problemas por la falta de experiencia

Riesgo 10	Elección de tecnologías
Probabilidad	3
Descripción	Falta de conocimiento para realizar determinadas tareas con las tecnologías escogidas para el desarrollo del proyecto.
Impacto	Alto
Indicadores	No saber realizar determinadas tareas con la tecnología escogida.
Plan de contingencia	Documentarse antes de las posibilidades de las tecnologías escogidas.

Tabla 14. Riesgo 10 - Elección de tecnologías

Riesgo 11	Contratiempos con la aplicación inicial
Probabilidad	8
Descripción	Desarrollo de la aplicación inicial sin una utilización de patrones de diseño, de implementación, etc. No separación de la parte visual, de la parte de negocio.
Impacto	Alto
Indicadores	La aplicación inicial desde la que se empieza tiene todo junto.
Plan de contingencia	Revisar la aplicación y ponerse en contacto con sus desarrolladores para consultar las dudas surgidas.

Tabla 15. Riesgo 11 - Contratiempos con la aplicación inicial

Riesgo 12	Borrado involuntario de código
Probabilidad	4
Descripción	Por algún tipo de fallo humano o tecnológico se puede perder parte o la totalidad del código escrito hasta el momento.
Impacto	Alto
Indicadores	Desaparición de trozos de código ya escrito anteriormente.
Plan de contingencia	Utilización de sistema de control de versiones como Subversion, volviendo a la versión anterior donde este el código habríamos solucionado el problema.

Tabla 16. Riesgo 12 - Borrado involuntario de código

Riesgo 13	Planificación inicial incorrecta
Probabilidad	6
Descripción	La planificación inicial no se ajusta a la realidad y por tanto lejos de cumplirse.
Impacto	Medio
Indicadores	Los plazos planteados no se cumplen y como consecuencia no se obtienen los hitos.
Plan de contingencia	Volver a realizar una planificación más realista y teniendo en cuenta el retraso acumulado.

Tabla 17. Riesgo 13 - Planificación inicial incorrecta

Riesgo 14	Periodos de ausencia de los desarrolladores
Probabilidad	4
Descripción	Los desarrolladores del proyecto no pueden trabajar por diversos motivos.
Impacto	Medio
Indicadores	Imposibilidad de trabajar por no encontrarse el desarrollador presente.
Plan de contingencia	Replantear la planificación inicial para recuperar el tiempo perdido en las ausencias de los desarrolladores.

Tabla 18. Riesgo 14 - Periodos de ausencia de los desarrolladores

Riesgo 15	Periodos de ausencia del cliente
Probabilidad	6
Descripción	El cliente no puede asistir a las reuniones y seguimiento del proyecto por diversos motivos.
Impacto	Medio
Indicadores	Imposibilidad de desarrollar algunas tareas por no encontrarse el cliente presente.
Plan de contingencia	Continuar con el resto de tareas no realizadas hasta el momento, replantear la planificación inicial para recuperar el tiempo perdido en las ausencias del cliente.

Tabla 19. Riesgo 15 - Periodos de ausencia del cliente

La Incidencia real de los riesgos en la planificación ha sido la siguiente:

Riesgo	Incidencia Real
Incertidumbre en las tareas	Muy alta
Amplitud de los objetivos	Alta
Falta de conocimiento de la tecnología	Alta
Mala planificación	Media
Dependencia de la tecnología de simulación	Media
Riesgo en futuras ampliaciones	Media
Falta de concreción de algunas tareas	Alta
Integración de tecnologías	Alta
Planificación inicial incorrecta	Media
Problemas en el sistema de control de versiones	Alta
Problemas por la falta de experiencia	Alta
Elección de tecnologías	Media
Contratiempos con la aplicación inicial	Muy Alta
Borrado involuntario de código	Baja
Periodos de ausencia de los desarrolladores	Baja
Periodos de ausencia del cliente	Media

Tabla 20. Incidencia real de los riesgos

2.3. Plan de Gestión de Configuraciones

Gestión de Configuración es el proceso de identificar y definir los elementos en el sistema, controlando el cambio de estos elementos a lo largo de su ciclo de vida, registrando y reportando el estado de los elementos y las solicitudes de cambio, y verificando que los elementos estén completos y que sean los correctos.

La Gestión de Configuración del Software implica la identificación de la Configuración del software en puntos dados en el tiempo, el control sistemático de los cambios en la Configuración y el mantenimiento de la integridad y trazabilidad de la Configuración a través del ciclo de vida del software. Los productos incluidos son:

- Software distribuido al cliente.
- Documentos de requerimientos del software.
- Código.
- Elementos requeridos para crearlos (ejemplo: el compilador)

2.3.1. Propósito del Plan de Gestión de Configuraciones

Se trata de establecer y mantener la integridad de los productos de software a través del ciclo de vida del proceso de software.

2.3.2. Alcance de la Configuración del Software

Este documento es complementario al Plan de Desarrollo Software. Para más información sobre responsabilidades, los hitos y los recursos consultar el Plan de desarrollo de Software.

2.3.3. Visión general

Inicialmente, el documento explica el entorno de gestión de la configuración. Luego, continúa con el programa de gestión de la configuración (incluyendo la Identificación de Configuración, el Control de Cambios y Configuración, y el Estado de Configuración). Se proporciona información sobre las estrategias de hitos, la formación y recursos usados y el entorno de desarrollo externo.

2.3.4. Programa de administración de la configuración

Los artefactos a producir vienen en la siguiente tabla

Nombre Artefacto
Glosario
Plan de Desarrollo Software
Lista de Riesgos
Plan de Gestión de Configuración
Especificación de los requisitos del Software
Plan de Iteración de la Fase de Inicio
Plan de Iteración de la Fase de Elaboración
Modelo de Casos de Uso
Documento de Análisis
Documento de Diseño
Prototipos de Interfaces de Usuario
Plan de Iteración de la Fase de Construcción
Modelo de Implementación
Casos de Pruebas
Plan de Pruebas
Plan de Iteración de la Fase de Transición
Resultado de Pruebas

Tabla 21. Artefactos a producir

Control y configuración de cambios

Aprobación y procesamiento de solicitud de cambios: el responsable de la Gestión de la Configuración es el encargado de, antes de realizar una copia de seguridad, revisar todos los artefactos que han sido modificados desde la última copia.

Cuenta del estado de configuración

Almacenamiento de las realizaciones del proyecto: el gestor de configuraciones se asegurará de que todas las versiones de la aplicación sean almacenadas externamente de forma regular como medida de realización de copias de seguridad.

2.4. Plan de Iteración de la Etapa de Inicio

2.4.1. Propósito del Plan de Iteración de la Etapa de Inicio

En esta sección se expondrá el plan de iteración que se ha seguido en la etapa de inicio. En la finalización se tendrá el Plan de desarrollo software, las especificaciones software, el plan de gestión de la configuración así como la lista de riesgos a tener en cuenta. El siguiente paso será empezar con la etapa de análisis y diseño del proyecto.

2.4.2. Plan de Trabajo para la Etapa de Inicio

En la siguiente figura podemos visualizar el Diagrama de Gantt que ilustra el plan de iteración para la etapa de inicio. En el comienzo del proyecto se han mantenido reuniones con el cliente para un seguimiento del proyecto, también se han mantenido reuniones entre los propios desarrolladores.

Posteriormente, se procederá a la elaboración del plan de desarrollo software, las especificaciones software, el plan de gestión de la configuración así como la lista de riesgos a tener en cuenta. Posteriormente se procederá con la etapa de elaboración del proyecto.

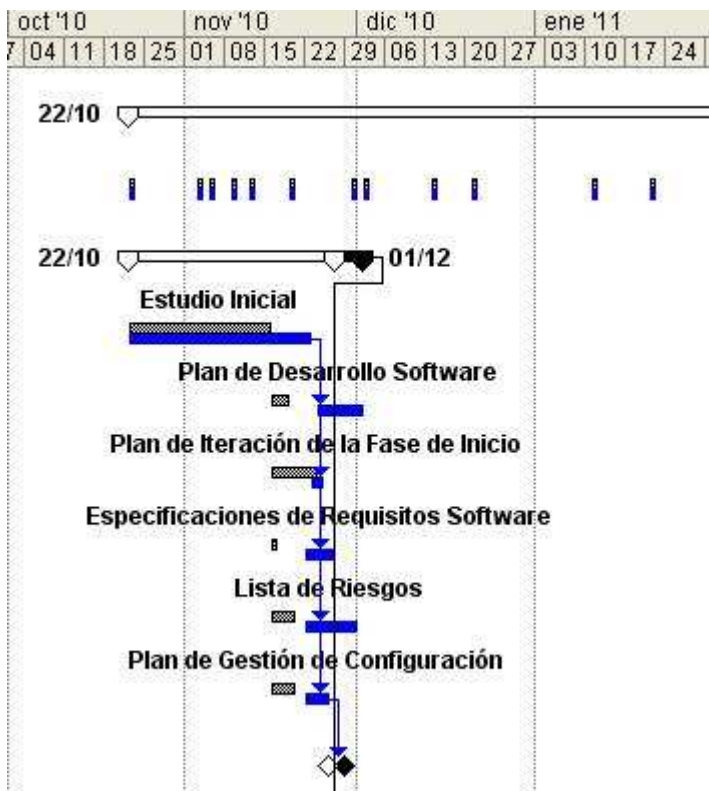


Figura 6. Diagrama de Gantt de la Fase de Inicio

Nombre de tarea	Duración	Comienzo	Fin
 Fase de Inicio	29 días	vie 22/10/10	mié 01/12/10
Estudio Inicial	22 días	vie 22/10/10	lun 22/11/10
Plan de Desarrollo Software	6 días	mié 24/11/10	mié 01/12/10
Plan de Iteración de la Fase de Inicio	2 días	mar 23/11/10	mié 24/11/10
Especificaciones de Requisitos Software	5 días	lun 22/11/10	vie 26/11/10
Lista de Riesgos	7 días	lun 22/11/10	mar 30/11/10
Plan de Gestión de Configuración	4 días	lun 22/11/10	jue 25/11/10
Plan de Iteración de la Fase de Elaboración	0 días	lun 29/11/10	lun 29/11/10

Figura 7. Tareas de la Fase de Inicio

2.4.3. Recursos Necesarios

Recursos Humanos

Los recursos humanos de esta iteración serán:

- Los desarrolladores
- El cliente.

Recursos Software

Los recursos software para esta iteración serán:

- Herramienta de planificación:
 - Microsoft Project 2003
- Herramienta de documentación:
 - Microsoft Word 2007
- Herramienta de ilustración
 - Adobe Photoshop CS5
- Herramienta de sincronización
 - Dropbox

Recursos Hardware

Los recursos hardware para esta iteración serán:

- Dos ordenadores portátiles personales
- Memoria USB para el intercambio de datos
- Conexión a internet

2.5. Plan de Iteración de la Etapa de Elaboración

2.5.1. Propósito del Plan de Iteración de la Etapa de Elaboración

En esta sección se expondrá el plan de iteración que se va a seguir en la etapa de análisis y diseño del proyecto. Se parte del análisis reflejado en el documento de análisis. En este documento cada caso de uso es explicado individualmente y va acompañado de su correspondiente diagrama de secuencia.

El siguiente hito será el modelo de arquitectura del sistema y el último hito será el documento de diseño.

2.5.2. Plan de Trabajo para la Etapa de Elaboración

En el diagrama de Gantt siguiente se muestra el plan de la iteración para la etapa de elaboración.

Se partirá de la aprobación de los requisitos por parte del cliente para empezar la elaboración de los casos de uso. Una vez fijados éstos, se procederá a la elaboración de sus correspondientes diagramas de secuencia y el diseño del diagrama de clases y máquinas de estados.

Como resultado se obtendrá el documento de análisis del sistema. Partiendo de este documento, se generará el documento de diseño de arquitectura.

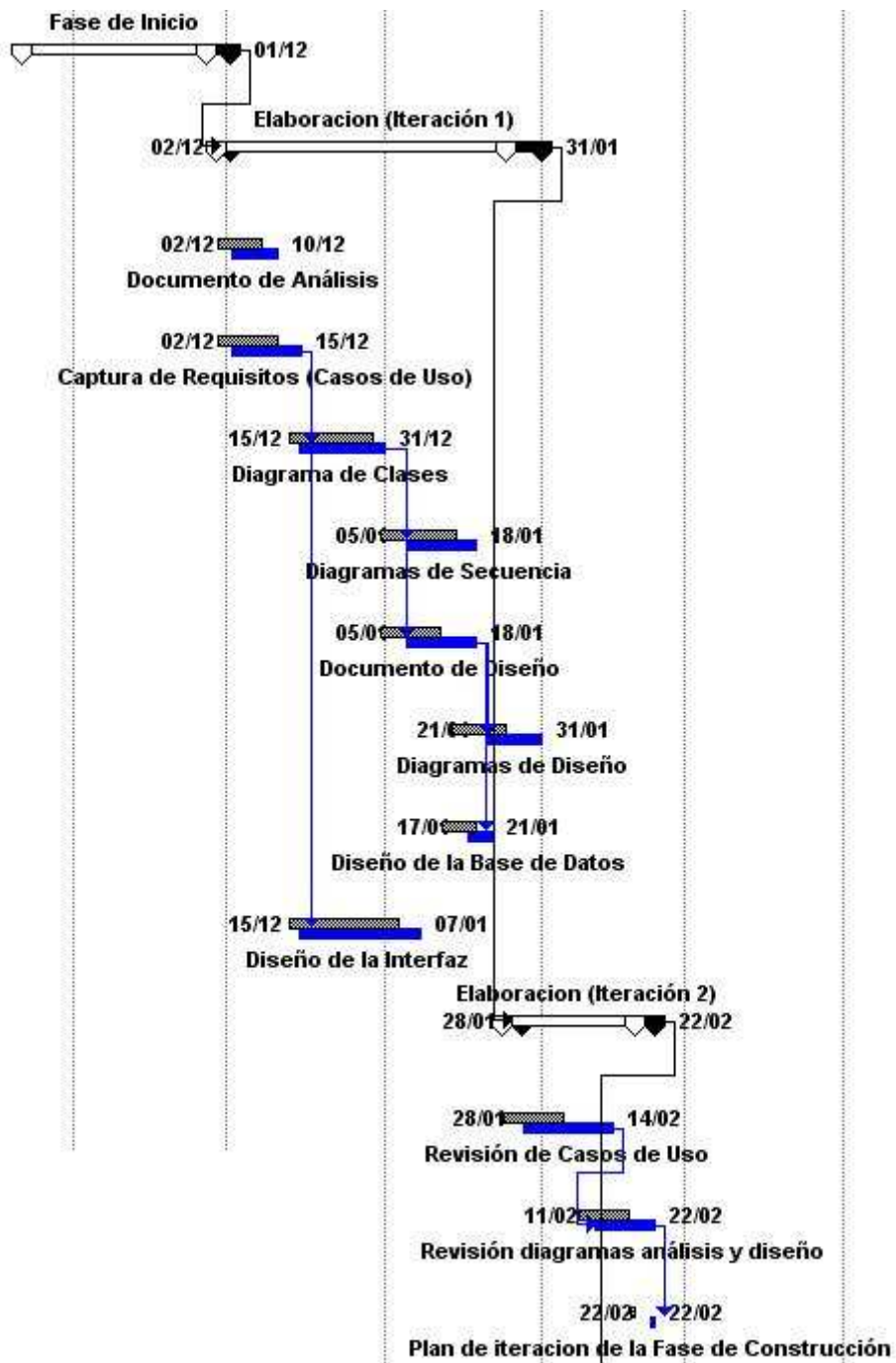


Figura 8. Diagrama de Gantt para la Fase de Elaboración

Nombre de tarea	Duración	Comienzo	Fin
 Elaboracion (Iteración 1)	43 días	jue 02/12/10	lun 31/01/11
Documento de Análisis	7 días	jue 02/12/10	vie 10/12/10
Captura de Requisitos (Casos de Uso)	10 días	jue 02/12/10	mié 15/12/10
Diagrama de Clases	13 días	mié 15/12/10	vie 31/12/10
Diagramas de Secuencia	10 días	mié 05/01/11	mar 18/01/11
Documento de Diseño	10 días	mié 05/01/11	mar 18/01/11
Diagramas de Diseño	7 días	vie 21/01/11	lun 31/01/11
Diseño de la Base de Datos	5 días	lun 17/01/11	vie 21/01/11
Diseño de la Interfaz	18 días	mié 15/12/10	vie 07/01/11
 Elaboracion (Iteración 2)	18 días	vie 28/01/11	mar 22/02/11
Revisión de Casos de Uso	12 días	vie 28/01/11	lun 14/02/11
Revisión diagramas análisis y diseño	8 días	vie 11/02/11	mar 22/02/11
Plan de iteracion de la Fase de Construcción	1 día	mar 22/02/11	mar 22/02/11

Figura 9. Tareas para la Fase de Elaboración

2.5.3. Recursos Necesarios

Recursos Humanos

Los recursos humanos de esta iteración serán:

- Los desarrolladores
- El cliente

Recursos Software

Los recursos software para esta iteración serán:

- Herramienta de modelado:
 - StarUML
 - REM
- Herramienta de planificación
 - Microsoft Project 2003
- Herramienta de documentación
 - Microsoft Word 2007
 - Pencil
- Herramienta de sincronización
 - Dropbox

Recursos Hardware

- Dos ordenadores portátiles personales
- Memoria USB para el intercambio de datos
- Conexión a internet

Capítulo 3

Fase de Elaboración

En este capítulo se detalla la fase de elaboración del proyecto, recogiendo los artefactos más importantes de esta fase.

3.1. Concepto de la Fase de Elaboración



Figura 10. Hito en la Fase de Elaboración

En la fase de elaboración se va buscar el establecimiento de la línea base de la arquitectura del sistema y proporcionar una base estable para el diseño y esfuerzo de implementación de la siguiente fase, mitigando la mayoría de los riesgos tecnológicos.

Los objetivos de esta fase son:

- Definir, validar y cimentar la arquitectura.
- Completar la visión.
- Crear un plan fiable para la fase de construcción. Este plan puede evolucionar en sucesivas iteraciones. Debe incluir los costes si procede.
- Demostrar que la arquitectura propuesta soportará la visión con un coste razonable y en un tiempo razonable.

Al terminar deben obtenerse los siguientes resultados:

- Un modelo de casos de uso: todos los actores identificados, y los casos de uso desarrollados.
- Requisitos adicionales que capturan los requisitos no funcionales y cualquier requisito no asociado con un Caso de Uso específico.
- Descripción de la arquitectura software.
- Un caso de desarrollo actualizado que especifica el proceso a seguir.
- Un prototipo arquitectónico ejecutable.

En esta fase se debe tratar de abarcar todo el proyecto con la profundidad mínima. Sólo se profundiza en los puntos críticos de la arquitectura o riesgos importantes.

En la fase de elaboración se actualizan todos los productos de la fase de inicio.

Los criterios de evaluación de esta fase son los siguientes:

- La visión del producto es estable.
- La arquitectura es estable.
- La demostración ejecutable muestra que se han tratado y resuelto los principales elementos de riesgo.
- El plan de la fase de construcción está suficientemente detallado.
- Todos los involucrados en el proyecto están de acuerdo que la visión actual que se puede alcanzar si el plan actual se ejecuta para desarrollar el sistema completo, en el contexto de la arquitectura actual.
- La diferencia entre los tiempos reales y previstos es aceptable.

Si no se superan los criterios de evaluación quizá sea necesario abandonar el proyecto o replanteárselo considerablemente.

3.2. Documento de Análisis

A continuación definimos ciertos conceptos generales e imprescindibles como objetivos principales y secundarios que se pretenden conseguir con la ejecución de este proyecto, requisitos funcionales y no funcionales obtenidos a partir de los objetivos anteriores, y actores del sistema que estarán presentes e interactuarán con éste.

3.2.1. Funcionamiento de la forma de trabajo

Después de la reunión con el cliente, se han conseguido la información necesaria para poder diseñar la forma de trabajo con la que se quiere actuar en la aplicación.

Los usuarios de la aplicación pueden ser de dos tipos

- Administrador: Este usuario tiene los permisos necesarios para poder realizar la gestión de cuadros de mando, salas de operaciones y usuarios.

- **Usuario registrado:** Cada usuario pertenecerá a una sala en concreto dependiendo de la clave que se le haya asignado, dicha sala tendrá cuadros de mando para que el usuario pueda realizar diferentes operaciones, obtener resultados de manera gráfica de las simulaciones realizadas.

Tendrá también la posibilidad de cargar y continuar partidas anteriormente guardadas, consultar datos de las simulaciones realizadas en partidas anteriormente guardadas.

3.2.2. Descripción de objetivos

Los objetivos del proyecto ya se definían al principio de esta memoria en el capítulo 1, en la sección 1.3. En este apartado vamos a realizar una descripción más detallada. Los objetivos los hemos ordenado por nivel de Importancia y dentro de ella por urgencia.

OBJ-001	Simulación con diferentes modelos
Versión	1.0
Autores	José Antonio González Alejos Mario del Valle Guijarro
Descripción	El sistema deberá poder simular con diferentes modelos.
Subobjetivos	Comprobación de que el modelo cargado sea válido o sea que tenga extensión .vpm.
Importancia	Vital (objetivo principal)
Urgencia	Vital
Estado	Validado
Estabilidad	Alta
Comentarios	Incluir la mejora de que no solo se pueda simular con un solo modelo sino que se pueda simular con cualquier modelo que carguemos.

Tabla 22. OBJ-001 – Simulación con diferentes cuadros de mando

OBJ-002	Administración y gestión de los diversos componentes
Versión	1.0
Autores	José Antonio González Alejos Mario del Valle Guijarro
Descripción	El sistema deberá poder administrar y gestionar cuadros de mando, salas de operaciones y usuarios.
Subobjetivos	El usuario podrá editar su perfil pudiendo modificar cualquiera de sus datos.
Importancia	Vital
Urgencia	Vital
Estado	Validado
Estabilidad	Alta
Comentarios	Solo habrá un usuario administrador en el sistema. Dicho usuario podrá realizar la administración de cualquier componente del sistema.

Tabla 23. OBJ-002 - Administración y gestión de los diversos componentes

OBJ-003	Creación de pantalla para jugar partidas
Versión	1.0
Autores	José Antonio González Alejos Mario del Valle Guijarro
Descripción	El sistema deberá poder ofrecer a los usuarios una pantalla para simular con los modelos y jugar partidas.
Subobjetivos	Cuadro de decisiones donde poder seleccionar el valor que van a tomar las variables palanca. Representación gráfica de los resultados de las simulaciones realizadas.
Importancia	Vital
Urgencia	Vital
Estado	Validado
Estabilidad	Alta
Comentarios	Pantalla donde se diferencie bien el cuadro de decisiones de la sección donde se representan los resultados mediante opciones gráficas como gráficas y relojes.

Tabla 24. OBJ-003 - Creación de pantalla para jugar partidas

OBJ-004	Creación de base de datos
Versión	1.0
Autores	José Antonio González Alejos Mario del Valle Guijarro
Descripción	El sistema deberá poder tener una base de datos para el almacenamiento de toda la información.
Subobjetivos	Ninguno
Importancia	Vital
Urgencia	Vital
Estado	Validado
Estabilidad	Alta
Comentarios	Estructura de algunas tablas que sea indiferente al número de variables palanca seleccionado.

Tabla 25. OBJ-004 - Creación de base de datos

OBJ-005	Aplicación cliente servidor
Versión	1.0
Autores	José Antonio González Alejos Mario del Valle Guijarro
Descripción	El sistema deberá ser accesible desde cualquier parte. Además de poder ser ejecutado de manera local, el sistema deberá ser accesible vía web desde cualquier ubicación.
Subobjetivos	Ninguno
Importancia	Vital
Urgencia	Vital
Estado	En transcurso
Estabilidad	Alta
Comentarios	Alojar la aplicación en un servidor de aplicaciones web como por ejemplo Apache Tomcat.

Tabla 26. OBJ-005 - Aplicación cliente – servidor

OBJ-006	Documentación de la herramienta
Versión	1.0
Autores	José Antonio González Alejos Mario del Valle Guijarro
Descripción	El sistema deberá contar con una documentación completa para su instalación, funcionamiento y ejecución.
Subobjetivos	Creación de un manual de usuario con las diferentes instrucciones para poder simular con modelos.
Importancia	Vital
Urgencia	Vital
Estado	Validado
Estabilidad	Alta
Comentarios	Ninguno

Tabla 27. OBJ-006 - Documentación de la herramienta

OBJ-007	Dependencia de las librerías de Vensim
Versión	1.0
Autores	José Antonio González Alejos Mario del Valle Guijarro
Descripción	El sistema deberá poder ser ejecutado desde cualquier ubicación diferente a la que hay por defecto.
Subobjetivos	Ninguno
Importancia	Importante
Urgencia	Alta
Estado	Validado
Estabilidad	Alta
Comentarios	Los modelos deberán estar en la misma carpeta donde está ubicada las librerías de Vensim.

Tabla 28. OBJ-007 - Dependencia de las librerías de Vensim

OBJ-008	Seguridad en la aplicación
Versión	1.0
Autores	José Antonio González Alejos Mario del Valle Guijarro
Descripción	El sistema deberá poder solventar cualquier fallo de seguridad, intentos de acceso de usuarios no registrados.
Subobjetivos	Ninguno
Importancia	Importante
Urgencia	Media
Estado	Validado
Estabilidad	Alta
Comentarios	Los usuarios no registrados que intenten acceder al sistema serán redirigidos a la pantalla login.

Tabla 29. OBJ-008 - Seguridad en la aplicación

OBJ-009	Aplicación multi usuario
Versión	1.0
Autores	José Antonio González Alejos Mario del Valle Guijarro
Descripción	El sistema deberá poder ser ejecutado por diferentes usuarios. Éstos podrán jugar partidas sobre el mismo cuadro de mando o diferentes.
Subobjetivos	Creación de un manual de usuario con las diferentes instrucciones para poder simular con modelos.
Importancia	Alta
Urgencia	Media
Estado	Validado
Estabilidad	Alta
Comentarios	Se permite a través del multicontexto, que varios usuarios puedan realizar simulaciones a la vez.

Tabla 30. OBJ-009 - Aplicación multi usuario

3.2.3. Captura de requisitos

Como consecuencia de las diferentes reuniones realizadas con el cliente se elabora un documento de especificación de requisitos con el propósito de describir las funcionalidades necesarias para poder validar el producto final.

3.2.4. Requisitos de Información

Los requisitos de información obtenidos de los objetivos anteriores son los que se describen en las siguientes tablas:

IRQ-001	Información sobre cuadros de mando
Versión	1.0
Autores	José Antonio González Alejos Mario del Valle Guijarro
Dependencias	FRQ-004, FRQ-005, FRQ-006
Descripción	El sistema deberá almacenar información sobre los diferentes cuadros de mando.
Datos específicos	Datos: nombre, estado, descripción, grupos, variables, unidades, modelo, rango inicial, rango final, valores.
Urgencia	Vital
Estado	Validado
Estabilidad	Alta
Comentarios	La información del cuadro de mando podrá ser administrada y gestionada por el usuario administrador.

Tabla 31. IRQ-001 - Información sobre cuadros de mando

IRQ-002	Información sobre salas de operaciones
Versión	1.0
Autores	José Antonio González Alejos Mario del Valle Guijarro
Dependencias	FRQ-007, FRQ-008, FRQ-009
Descripción	El sistema deberá almacenar información sobre las diferentes salas de operaciones.
Datos específicos	Datos: nombre, password, configuración.
Urgencia	Vital
Estado	Validado
Estabilidad	Alta
Comentarios	La información de las salas de operaciones podrá ser administrada y gestionada por el usuario administrador.

Tabla 32. IRQ-002 - Información sobre las salas de operaciones

IRQ-003	Información sobre usuarios
Versión	1.0
Autores	José Antonio González Alejos Mario del Valle Guijarro
Dependencias	FRQ-011, FRQ-012
Descripción	El sistema deberá almacenar información sobre los diferentes usuarios.
Datos específicos	Datos: cid, usuario, password, clave ver, nombre, apellido, email, tipo, valido, sala.
Urgencia	Vital
Estado	Validado
Estabilidad	Alta
Comentarios	La información de los usuarios podrá ser administrada y gestionada por el usuario administrador.

Tabla 33. IRQ-003 - Información sobre usuarios

IRQ-004	Información sobre partidas
Versión	1.0
Autores	José Antonio González Alejos Mario del Valle Guijarro
Dependencias	FRQ-002, FRQ-003
Descripción	El sistema deberá almacenar información sobre las diferentes partidas realizadas por los usuarios.
Datos específicos	Datos: nombre, usuario, cuadro de mando, carpeta, archivo, modelo, variables, valores.
Urgencia	Vital
Estado	Validado
Estabilidad	Alta
Comentarios	La información de las partidas podrá ser administrada y gestionada por el usuario administrador.

Tabla 34. IRQ-004 - Información sobre partidas

3.2.5. Requisitos Funcionales

A continuación se describirán los requisitos para este proyecto. Se tendrán dos tipos de requisitos: requisitos funcionales, que serán aquellos obtenidos a partir de los objetivos principales y secundarios del apartado anterior; y requisitos no funcionales, obtenidos de las especificaciones citadas a lo largo de la memoria y que precisen ser descritas como requisitos.

Este tipo de requisitos declaran los servicios que debe proporcionar el sistema, como debe reaccionar a una entrada particular y cómo se debe de comportar ante situaciones particulares. Describen el funcionamiento del sistema.

Los requisitos de información llevarán un nombre descriptivo y un número para identificarlos. Este número tendrá el formato *IRQ* - <*id*> donde *id* será el número del requisito de información. Además se indicarán los requisitos asociados, una descripción y cualquier información que sea relevante para una mayor claridad.

Los requisitos funcionales que debe cumplir el sistema son los siguientes:

FRQ-001	Autenticación de usuarios
Versión	1.0
Autores	José Antonio González Alejos Mario del Valle Guijarro
Dependencias	OBJ-008
Descripción	El sistema deberá poder autenticar los usuarios que intenten acceder a través de su nombre de usuario y contraseña.
Importancia	Vital
Urgencia	Inmediatamente
Estado	Validado
Estabilidad	Alta
Comentarios	Los usuarios no autenticados que intenten acceder al sistema serán redirigidos a la pantalla login.

Tabla 35. FRQ-001 - Autenticación de usuarios

FRQ-002	Gestión de simulaciones
Versión	1.0
Autores	José Antonio González Alejos Mario del Valle Guijarro
Dependencias	OBJ-001
Descripción	El sistema deberá permitir gestionar de manera independientes las simulaciones de cada usuario.
Importancia	Vital
Urgencia	Inmediatamente
Estado	Validado
Estabilidad	Alta
Comentarios	Ninguno

Tabla 36. FRQ-002 - Gestión de simulaciones

FRQ-003	Muestra de simulaciones
Versión	1.0
Autores	José Antonio González Alejos Mario del Valle Guijarro
Dependencias	OBJ-003
Descripción	El sistema deberá permitir mostrar de manera gráfica los resultados de la simulación.
Importancia	Vital
Urgencia	Inmediatamente
Estado	Validado
Estabilidad	Alta
Comentarios	Ninguno

Tabla 37. FRQ-003 - Muestra de simulaciones

FRQ-004	Crear cuadro de mando
Versión	1.0
Autores	José Antonio González Alejos Mario del Valle Guijarro
Dependencias	OBJ-002, OBJ-004
Descripción	El sistema deberá permitir la creación de un nuevo cuadro de mando.
Importancia	Vital
Urgencia	Inmediatamente
Estado	Validado
Estabilidad	Alta
Comentarios	Ninguno

Tabla 38. FRQ-004 - Crear cuadro de mando

FRQ-005	Modificar cuadro de mando
Versión	1.0
Autores	José Antonio González Alejos Mario del Valle Guijarro
Dependencias	OBJ-002, OBJ-004
Descripción	El sistema deberá permitir modificar un cuadro de mando.
Importancia	Vital
Urgencia	Inmediatamente
Estado	Validado
Estabilidad	Alta
Comentarios	Ninguno

Tabla 39. FRQ-005 - Modificar cuadro de mando

FRQ-006	Eliminar cuadro de mando
Versión	1.0
Autores	José Antonio González Alejos Mario del Valle Guijarro
Dependencias	OBJ-002, OBJ-004
Descripción	El sistema deberá permitir eliminar un cuadro de mando.
Importancia	Vital
Urgencia	Inmediatamente
Estado	Validado
Estabilidad	Alta
Comentarios	Ninguno

Tabla 40. FRQ-006 - Eliminar cuadro de mando

FRQ-007	Crear sala de operaciones
Versión	1.0
Autores	José Antonio González Alejos Mario del Valle Guijarro
Dependencias	OBJ-002, OBJ-004
Descripción	El sistema deberá permitir la creación de una nueva sala de operaciones.
Importancia	Vital
Urgencia	Inmediatamente
Estado	Validado
Estabilidad	Alta
Comentarios	Ninguno

Tabla 41. FRQ-007 - Crear sala de operaciones

FRQ-008	Modificar sala de operaciones
Versión	1.0
Autores	José Antonio González Alejos Mario del Valle Guijarro
Dependencias	OBJ-002, OBJ-004
Descripción	El sistema deberá permitir modificar una sala de operaciones.
Importancia	Vital
Urgencia	Inmediatamente
Estado	Validado
Estabilidad	Alta
Comentarios	Ninguno

Tabla 42. FRQ-008 - Modificar sala de operaciones

FRQ-009	Eliminar sala de operaciones
Versión	1.0
Autores	José Antonio González Alejos Mario del Valle Guijarro
Dependencias	OBJ-002, OBJ-004
Descripción	El sistema deberá permitir eliminar una sala de operaciones.
Importancia	Vital
Urgencia	Inmediatamente
Estado	Validado
Estabilidad	Alta
Comentarios	Ninguno

Tabla 43. FRQ-009 - Eliminar sala de operaciones

FRQ-011	Modificar usuario
Versión	1.0
Autores	José Antonio González Alejos Mario del Valle Guijarro
Dependencias	OBJ-002, OBJ-004
Descripción	El sistema deberá permitir modificar los datos del usuario excepto la contraseña.
Importancia	Vital
Urgencia	Inmediatamente
Estado	Validado
Estabilidad	Alta
Comentarios	Ninguno

Tabla 44. FRQ-011 - Modificar usuario

FRQ-012	Eliminar usuario
Versión	1.0
Autores	José Antonio González Alejos Mario del Valle Guijarro
Dependencias	OBJ-002, OBJ-004
Descripción	El sistema deberá permitir eliminar un usuario.
Importancia	Vital
Urgencia	Inmediatamente
Estado	Validado
Estabilidad	Alta
Comentarios	Ninguno

Tabla 45. FRQ-012 - Eliminar usuario

FRQ-013	Acceso a la aplicación
Versión	1.0
Autores	José Antonio González Alejos Mario del Valle Guijarro
Dependencias	OBJ-008
Descripción	El sistema deberá permitir el acceso mediante una url vía web.
Importancia	Vital
Urgencia	Inmediatamente
Estado	Validado
Estabilidad	Alta
Comentarios	Ninguno

Tabla 46. FRQ-013 - Acceso a la aplicación

FRQ-014	Conexión a Vensim
Versión	1.0
Autores	José Antonio González Alejos Mario del Valle Guijarro
Dependencias	OBJ-007
Descripción	El sistema deberá permitir conectarse a Vensim DSS para realizar simulaciones.
Importancia	Vital
Urgencia	Inmediatamente
Estado	Validado
Estabilidad	Alta
Comentarios	Ninguno

Tabla 47. FRQ-014 - Conexión a Vensim

FRQ-015	Permitir ampliaciones
Versión	1.0
Autores	José Antonio González Alejos Mario del Valle Guijarro
Dependencias	OBJ-004, OBJ-006, OBJ-007
Descripción	El sistema deberá permitir ampliaciones futuras para añadir diversas funcionalidades.
Importancia	Vital
Urgencia	Inmediatamente
Estado	Validado
Estabilidad	Alta
Comentarios	Ninguno

Tabla 48. FRQ-015 - Permitir ampliaciones

3.2.6. Requisitos No Funcionales

Este tipo de requisitos define las propiedades emergentes del sistema, tales como el tiempo de respuesta, las necesidades de almacenamiento, la fiabilidad.

Pueden especificar también la utilización de una herramienta CASE en particular, un lenguaje de programación o un método del desarrollo.

En este apartado se detalla el catálogo de requisitos no funcionales. Estos requisitos pueden ser más críticos que los requisitos funcionales, ya que son a los que normalmente debe apuntar la arquitectura y si estos no son cumplidos, el software puede no funcionar o el cliente simplemente no acepta el producto.

Estos requisitos se puede decir que no están relacionados directamente con la funcionalidad del sistema. Con esto no se pretende decir que no sean necesarios para la funcionalidad del sistema, pero sí que no pueden asociarse a ningún caso de uso en particular. Hay distintas categorías y tipos de requisitos no funcionales: hardware, plataforma software, comunicaciones, de interfaz, de seguridad, de control de concurrencia, de persistencia...etc.

Los requisitos no funcionales llevarán un nombre descriptivo y un número para identificarlos. Este número tendrá el formato *NFRQ* - <id> donde *id* será el número del requisito de información. Además se indicará una descripción y cualquier información que sea relevante para una mayor claridad.

Algunos requisitos no funcionales identificados para el desarrollo del presente proyecto son los siguientes:

NFRQ-001	Compatibilidad navegadores web
Versión	1.0
Autores	José Antonio González Alejos Mario del Valle Guijarro
Dependencias	OBJ-005
Descripción	El sistema deberá permitir ser compatible con Internet Explorer y Mozilla Firefox.
Importancia	Importante
Urgencia	Puede esperar
Estado	Validado
Estabilidad	Alta
Comentarios	Ninguno

Tabla 49. NFRQ-001 - Compatibilidad navegadores web

NFRQ-002	Requerimientos mínimos
Versión	1.0
Autores	José Antonio González Alejos Mario del Valle Guijarro
Dependencias	OBJ-005, OBJ-007
Descripción	El servidor de aplicaciones web deberá permitir ser ejecutado en una máquina, bajo sistema operativo MS Windows.
Importancia	Importante
Urgencia	Puede esperar
Estado	Validado
Estabilidad	Alta
Comentarios	Ninguno

Tabla 50. NFRQ-002 - Requerimientos mínimos

NFRQ-003	Herramientas gratuitas
Versión	1.0
Autores	José Antonio González Alejos Mario del Valle Guijarro
Dependencias	OBJ-006
Descripción	Para el desarrollo y ejecución de la aplicación se deben utilizar herramientas gratuitas que no supongan un coste para el cliente.
Importancia	Importante
Urgencia	Puede esperar
Estado	Validado
Estabilidad	Alta
Comentarios	La utilización de herramientas gratuitas reduce considerablemente los costes asociados a la realización del proyecto.

Tabla 51. NFRQ-003 - Herramientas gratuitas

NFRQ-004	Fecha de entrega
Versión	1.0
Autores	José Antonio González Alejos Mario del Valle Guijarro
Dependencias	OBJ-001, OBJ-002, OBJ-003, OBJ-005, OBJ-006
Descripción	La aplicación debe estar realizada antes del 15 de Julio de 2011 como fecha límite.
Importancia	Importante
Urgencia	Puede esperar
Estado	Validado
Estabilidad	Alta
Comentarios	La fecha límite para entrega de proyectos en Julio es hasta el día 15 de Julio de 2011.

Tabla 52. NFRQ-004 - Fecha de entrega

NFRQ-005	Licencia Vensim
Versión	1.0
Autores	José Antonio González Alejos Mario del Valle Guijarro
Dependencias	OBJ-007, OBJ-009
Descripción	La herramienta Vensim debe contar con la licencia original de la copia del producto, licencia para la universidad.
Importancia	Importante
Urgencia	Puede esperar
Estado	Validado
Estabilidad	Alta
Comentarios	Los datos de la licencia no deben ser facilitados bajo ningún concepto.

Tabla 53. NFRQ-005 - Licencia Vensim

NFRQ-006	Documentación de la aplicación
Versión	1.0
Autores	José Antonio González Alejos Mario del Valle Guijarro
Dependencias	OBJ-006
Descripción	La aplicación debe contar con la descripción de los pasos a seguir en la instalación, configuración previa para un correcto funcionamiento y una documentación sobre el manejo de la aplicación,
Importancia	Importante
Urgencia	Puede esperar
Estado	Validado
Estabilidad	Alta
Comentarios	Manual de usuario donde viene explicado de manera detallada cada una de las pantallas que conforman la aplicación.

Tabla 54. NFRQ-006 – Documentación de la aplicación

NFRQ-007	Compatible con Microsoft Windows
Versión	1.0
Autores	José Antonio González Alejos Mario del Valle Guijarro
Dependencias	OBJ-007
Descripción	La aplicación debe ser compatible con el sistema operativo Microsoft Windows.
Importancia	Importante
Urgencia	Puede esperar
Estado	Validado
Estabilidad	Alta
Comentarios	Ninguno.

Tabla 55. NFRQ-007 – Compatible con Microsoft Windows

NFRQ-008	Hojas de estilo CSS
Versión	1.0
Autores	José Antonio González Alejos Mario del Valle Guijarro
Dependencias	OBJ-002, OBJ-003
Descripción	Aplicación de hojas de estilo CSS para una visualización de las pantallas más atractivas, interfaces más amigables y una mejora de la usabilidad de la aplicación.
Importancia	Quedaría bien
Urgencia	Puede esperar
Estado	Validado
Estabilidad	Alta
Comentarios	Utilización de la aplicación externa Adobe Photoshop para la realización de las imágenes que componen el diseño y hojas de estilo CSS para la maquetación de las pantallas.

Tabla 56. NFRQ-008 – Hojas de estilo CSS

3.2.7. Identificación de actores

Todo sistema necesita interactuar con algún actor, ya sea humano o software. Un actor humano se identifica por el papel que desempeña un humano en la interacción con el sistema. Y un actor software, para este proyecto, se definirá como el papel que desempeñan el resto de servicios tecnológicos con nuestro servicio.

A continuación se pasará a describir cada uno de los actores, tanto software como humanos, identificados en nuestro sistema:

ACT-001	Usuario
Versión	1.0
Autores	José Antonio González Alejos Mario del Valle Guijarro
Descripción	Este actor representa una generalización de todos los usuarios que pueden interactuar con el sistema, representa a los usuarios que pueden identificarse en el sistema después de haberse registrado en el mismo.
Comentarios	Ninguno

Tabla 57. ACT-001 – Usuario

ACT-002	Usuario no registrado
Versión	1.0
Autores	José Antonio González Alejos Mario del Valle Guijarro
Descripción	Este actor representa a los usuarios que todavía no se han registrado en el sistema y acceden por primera vez.
Comentarios	Si este tipo de usuario intenta acceder al sistema será redirigido a la pantalla de login.

Tabla 58. ACT-002 - Usuario no registrado

ACT-003	Jugador
Versión	1.0
Autores	José Antonio González Alejos Mario del Valle Guijarro
Descripción	Este actor representa a los usuarios que están registrados en el modelo de datos, están autenticados mediante su nombre de usuario y contraseña.
Comentarios	Ninguno

Tabla 59. ACT-003 - Usuario registrado

ACT-004	Administrador
Versión	1.0
Autores	José Antonio González Alejos Mario del Valle Guijarro
Descripción	Este actor representa al usuario que tiene permisos para poder gestionar cuadros de mando, salas de operaciones y usuarios.
Comentarios	Ninguno

Tabla 60. ACT-004 - Administrador

3.2.8. Diagrama de casos de uso

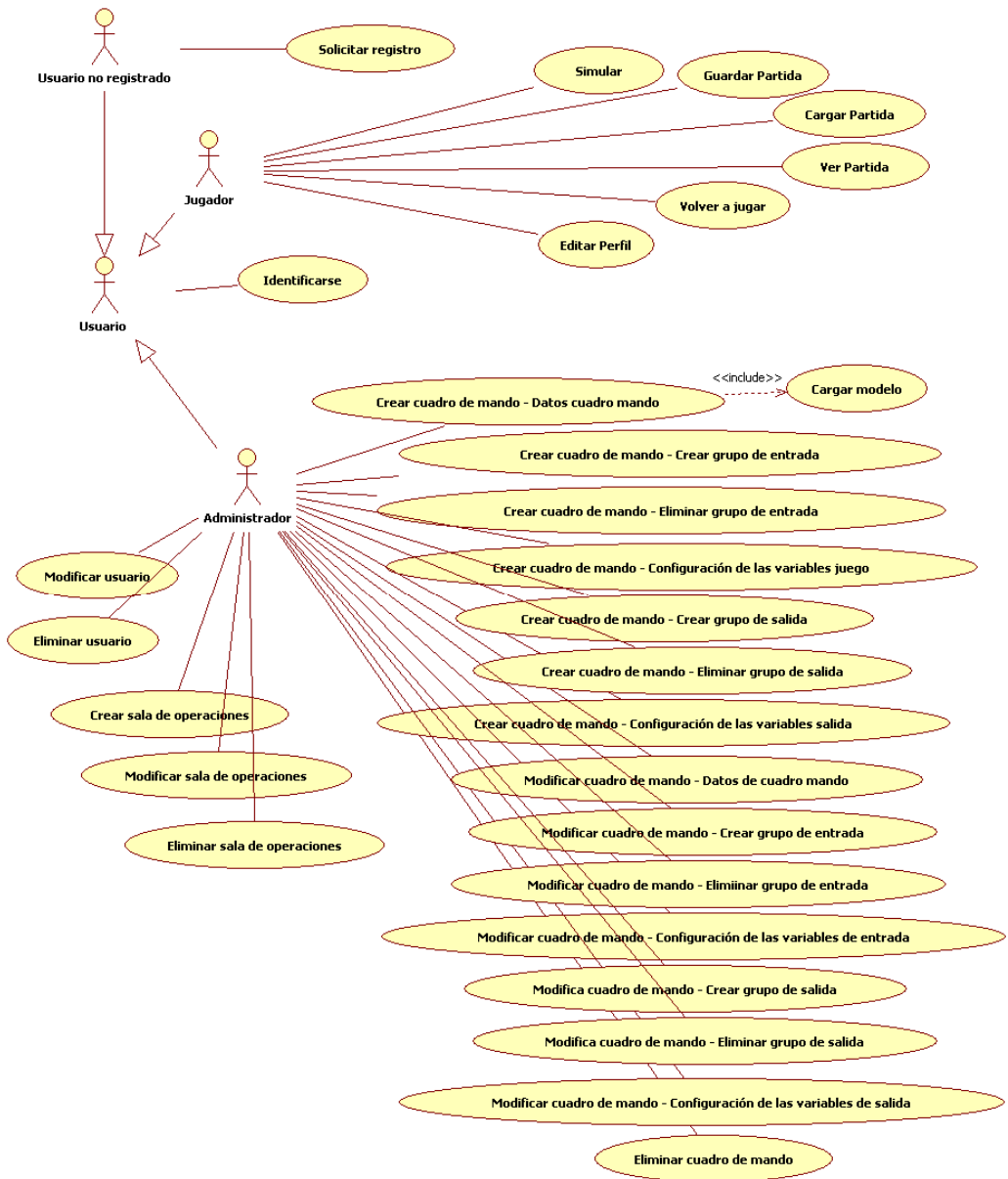


Figura 11. Diagrama de Casos de Uso

3.2.9. Descripción de Casos de Uso

A continuación se describen los casos de uso utilizados en este proyecto:

CU-001	Solicitar registro	
Versión	1.0	
Autores	José Antonio González Alejos Mario del Valle Guijarro	
Dependencias	OBJ-002, OBJ-004, OBJ-007	
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando un usuario quiera registrarse en la aplicación	
	Debe existir al menos una sala de operaciones en el sistema.	
Precondición	El usuario no existe en el modelo de datos.	
Secuencia normal	Paso	Acción
	1	El actor usuario no registrado (ACT-002) solicita registrarse en la aplicación.
	2	El sistema muestra una pantalla para que el nuevo usuario introduzca sus datos entre los cuales está la contraseña de la sala a la que va a pertenecer.
	3	El actor usuario no registrado (ACT-002) introduce los datos y la contraseña de la sala.
	4	El sistema valida los datos introducidos.
	5	El sistema muestra una pantalla indicando que el registro se ha realizado de manera satisfactoria.
Postcondición	El usuario ha quedado registrado en la aplicación.	
Excepciones	Paso	Acción
	2	Si alguno de los datos introducidos es incorrecto, la aplicación muestra un mensaje de error y vuelve al paso 2, a continuación este caso de uso continúa.
Importancia	Elevada	
Urgencia	Alta	
Estado	Completado	
Estabilidad	Alta	
Comentarios	Ninguno	

Tabla 61. CU-001 - Solicitar registro

CU-002	Identificarse	
Versión	1.0	
Autores	José Antonio González Alejos Mario del Valle Guijarro	
Dependencias	OBJ-004, OBJ-007	
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando un usuario quiera acceder al sistema.	
Precondición	No hay	
Secuencia normal	Paso	Acción
	1	El actor usuario (ACT-002) solicita identificarse.
	2	El sistema solicita los datos de identificación: nombre de usuario y contraseña.
	3	El actor usuario (ACT-002) introduce los datos en los campos indicados.
	4	El sistema inicia la sesión del usuario y muestra la pantalla con las acciones disponibles según el tipo de usuario.
Postcondición	El usuario se encuentra identificado en el sistema.	
Excepciones	Paso	Acción
	4	Si los datos introducidos no coinciden con ningún usuario registrado en el sistema, el sistema informa de error y vuelve al paso 2, a continuación este caso de uso continúa.
Importancia	Elevada	
Urgencia	Alta	
Estado	Completado	
Estabilidad	Alta	
Comentarios	Ninguno	

Tabla 62. CU-002 - Identificarse

CU-003	Crear sala de operaciones	
Versión	1.0	
Autores	José Antonio González Alejos Mario del Valle Guijarro	
Dependencias	OBJ-002, OBJ-004	
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando un administrador desea crear una sala de operaciones.	
	Debe de existir al menos un cuadro de mando libre en el sistema.	
Precondición	El usuario debe encontrarse identificado en el sistema como administrador.	
Secuencia normal	Paso	Acción
	1	El actor administrador (ACT-003) solicita introducir los datos de una nueva sala de operaciones.
	2	El sistema le muestra al usuario una lista con las salas de operaciones existentes y solicita los datos de la sala de operaciones
	3	El actor administrador (ACT-003) introduce los datos de la sala de operaciones.
	4	El sistema registra la nueva sala de operaciones en el sistema y muestra un mensaje informando del resultado de la operación.
Postcondición	Se ha creado y registrado una nueva sala de operaciones.	
Excepciones	Paso	Acción
	3	Si el usuario cancela la operación, a continuación este caso de uso queda sin efecto.
	4	Si ya existe una sala de operaciones con ese nombre, el sistema muestra un mensaje de error y vuelve al paso 2, a continuación este caso de uso continúa.
	4	Si el usuario no ha introducido ningún nombre de sala de operaciones, el sistema muestra un mensaje de error y vuelve al paso 2, a continuación este caso de uso continúa.
Importancia	Elevada	
Urgencia	Alta	
Estado	Completado	
Estabilidad	Alta	
Comentarios	Ninguno	

Tabla 63. CU-003 - Crear sala de operaciones

CU-004	Modificar sala de operaciones	
Versión	1.0	
Autores	José Antonio González Alejos Mario del Valle Guijarro	
Dependencias	OBJ-002, OBJ-004	
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando un administrador desea modificar una sala de operaciones.	
Precondición	El usuario debe encontrarse identificado en el sistema como administrador.	
Secuencia normal	Paso	Acción
	1	El actor administrador (ACT-003) solicita modificar los datos de una sala de operaciones.
	2	El sistema muestra las salas de operaciones existentes
	3	El actor administrador (ACT-003) selecciona una sala de operaciones.
	4	El sistema muestra los datos actuales de la sala de operaciones seleccionada y solicita los nuevos datos de la sala de operaciones.
	5	El actor administrador (ACT-003) introduce los nuevos datos.
	6	El sistema registra los cambios de la sala de operaciones y muestra un mensaje informando del resultado de la operación.
Postcondición	Se ha modificado la sala de operaciones.	
Excepciones	Paso	Acción
	3	Si el usuario cancela la operación, a continuación este caso de uso queda sin efecto.
	3	Si no hay salas de operaciones, a continuación este caso de uso queda sin efecto.
	4	Si el usuario no ha introducido ningún nombre de sala de operaciones, el sistema muestra un mensaje de error y vuelve al paso 2, a continuación este caso de uso continúa.
Importancia	Elevada	
Urgencia	Alta	
Estado	Completado	
Estabilidad	Alta	
Comentarios	Ninguno	

Tabla 64. CU-004 - Modificar sala de operaciones

CU-005	Eliminar sala de operaciones	
Versión	1.0	
Autores	José Antonio González Alejos Mario del Valle Guijarro	
Dependencias	OBJ-002, OBJ-004	
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando un administrador desea eliminar una sala de operaciones.	
Precondición	El usuario debe encontrarse identificado en el sistema como administrador.	
Secuencia normal	Paso	Acción
	1	El actor administrador (ACT-003) solicita eliminar una sala de operaciones.
	2	El sistema muestra las salas de operaciones existentes y solicita el nuevo nombre de la sala de operaciones.
	3	El actor administrador (ACT-003) selecciona una sala de operaciones.
	4	El sistema registra los cambios y muestra un mensaje informando del resultado de la operación.
Postcondición	Se ha eliminado la sala de operaciones y los usuarios asociados a dicha sala de operaciones.	
Excepciones	Paso	Acción
	3	Si el usuario cancela la operación, a continuación este caso de uso queda sin efecto.
	3	Si no hay salas de operaciones, a continuación este caso de uso queda sin efecto.
Importancia	Elevada	
Urgencia	Alta	
Estado	Completado	
Estabilidad	Alta	
Comentarios	Ninguno	

Tabla 65. CU-005 - Eliminar sala de operaciones

CU-006	Modificar usuario	
Versión	1.0	
Autores	José Antonio González Alejos Mario del Valle Guijarro	
Dependencias	OBJ-002, OBJ-004	
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando un administrador desea modificar un usuario.	
Precondición	El usuario debe encontrarse identificado en el sistema como administrador.	
Secuencia normal	Paso	Acción
	1	El actor administrador (ACT-003) solicita modificar los datos de un usuario.
	2	El sistema muestra los usuarios existentes.
	3	El actor administrador (ACT-003) selecciona un usuario y solicita ver sus datos actuales.
	4	El sistema muestra los datos actuales del usuario seleccionado y solicita los nuevos datos del usuario.
	5	El actor administrador (ACT-003) introduce los nuevos datos.
	6	El sistema registra los cambios del usuario y muestra un mensaje informando del resultado de la operación.
Postcondición	Se ha modificado el usuario.	
Excepciones	Paso	Acción
	5	Si el usuario cancela la operación, a continuación este caso de uso queda sin efecto.
	6	Si el usuario ha introducido los datos de manera incorrecta, el sistema muestra un mensaje de error y vuelve al paso 2, a continuación este caso de uso continúa.
Importancia	Elevada	
Urgencia	Alta	
Estado	Completado	
Estabilidad	Alta	
Comentarios	Ninguno	

Tabla 66. CU-006 - Modificar usuario

CU-007	Eliminar usuario	
Versión	1.0	
Autores	José Antonio González Alejos Mario del Valle Guijarro	
Dependencias	OBJ-002, OBJ-004	
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando un administrador desea eliminar un usuario.	
Precondición	El usuario debe encontrarse identificado en el sistema como administrador.	
Secuencia normal	Paso	Acción
	1	El actor administrador (ACT-003) solicita eliminar un usuario.
	2	El sistema muestra los usuarios existentes.
	3	El actor administrador (ACT-003) selecciona un usuario.
	4	El sistema registra los cambios y muestra un mensaje informando del resultado de la operación.
Postcondición	Se ha eliminado el usuario.	
Excepciones	Paso	Acción
	3	Si el usuario cancela la operación, a continuación este caso de uso queda sin efecto.
Importancia	Elevada	
Urgencia	Alta	
Estado	Completado	
Estabilidad	Alta	
Comentarios	Ninguno	

Tabla 67. CU-007 - Eliminar usuario

CU-008	Cargar modelo	
Versión	1.0	
Autores	José Antonio González Alejos Mario del Valle Guijarro	
Dependencias	OBJ-004	
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando un administrador desee cargar un modelo.	
Precondición	El usuario debe encontrarse identificado en el sistema como administrador.	
Secuencia normal	Paso	Acción
	1	El actor administrador (ACT-003) solicita cargar un modelo.
	2	El sistema muestra una nueva ventana emergente que es un explorador de directorios.
	3	El actor administrador (ACT-003) selecciona un modelo.
	4	El sistema muestra un mensaje informando del resultado de la operación.
Postcondición	Se ha cargado el modelo.	
Excepciones	Paso	Acción
	3	Si el usuario cancela la operación, a continuación este caso de uso queda sin efecto.
	3	Si el usuario ha seleccionado un fichero con extensión diferente a .vpm el sistema muestra un mensaje de error, a continuación este caso de uso queda sin efecto.
Importancia	Elevada	
Urgencia	Alta	
Estado	Completado	
Estabilidad	Alta	
Comentarios	Ninguno	

Tabla 68. CU-008 - Cargar modelo

CU-009	Crear cuadro de mando – Datos cuadro de mando	
Versión	1.0	
Autores	José Antonio González Alejos Mario del Valle Guijarro	
Dependencias	OBJ-002, OBJ-004	
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando un administrador desee crear un nuevo cuadro de mando.	
Precondición	El usuario debe encontrarse identificado en el sistema como administrador. El modelo debe estar cargado en el sistema.	
Secuencia normal	Paso	Acción
	1	El actor administrador (ACT-003) solicita crear un nuevo cuadro de mando.
	2	Se realiza el caso de uso Cargar Modelo.
	3	El sistema muestra una nueva ventana con campos de texto para introducir los datos del nuevo cuadro de mando.
	4	El actor administrador (ACT-003) introduce el nombre y descripción del cuadro de mando.
	5	El sistema registra los cambios.
Postcondición	Se ha guardado los datos del nuevo cuadro de mando.	
Excepciones	Paso	Acción
	3	Si el usuario deja en blanco el nombre del nuevo cuadro de mando, el sistema muestra un mensaje de error y vuelve al paso 2, a continuación este caso de uso continua.
Importancia	Elevada	
Urgencia	Alta	
Estado	Completado	
Estabilidad	Alta	
Comentarios	Ninguno	

Tabla 69. CU-009 - Crear cuadro de mando - Datos cuadro de mando

CU-010	Crear cuadro de mando – Crear grupo de entrada	
Versión	1.0	
Autores	José Antonio González Alejos Mario del Valle Guijarro	
Dependencias	OBJ-002, OBJ-004	
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando un administrador desee agrupar las variables de entrada de un nuevo cuadro de mando.	
Precondición	El usuario debe encontrarse identificado en el sistema como administrador.	
Secuencia normal	Paso	Acción
	1	El actor administrador (ACT-003) solicita agrupar las variables de entrada del nuevo cuadro de mando.
	2	El sistema muestra una ventana para agrupar las variables de entrada del nuevo cuadro de mando.
	3	El actor administrador (ACT-003) selecciona el botón llamado Crear grupo.
	4	El sistema muestra una nueva ventana emergente con un campo de texto para introducir el nombre del grupo.
	5	El actor administrador (ACT-003) introduce el nombre del grupo.
	6	El sistema registra los cambios y muestra el nuevo grupo creado.
Postcondición	Se ha creado el nuevo grupo de variables entrada.	
Excepciones	Paso	Acción
	5	Si el usuario deja en blanco el nombre del nuevo cuadro de mando, el sistema muestra un mensaje de error y vuelve al paso 4, a continuación este caso de uso continua.
	5	Si el usuario introduce un nombre de grupo ya existente en el modelo de datos, el sistema muestra un mensaje de error y vuelve al paso 2, a continuación este caso de uso continua.
	5	Si el usuario cancela la operación, a continuación este caso de uso queda sin efecto.
Importancia	Elevada	
Urgencia	Alta	
Estado	Completado	
Estabilidad	Alta	
Comentarios	Ninguno	

Tabla 70. CU-010 - Crear cuadro de mando – Crear grupo de entrada

CU-011	Crear cuadro de mando – Eliminar grupo de entrada	
Versión	1.0	
Autores	José Antonio González Alejos Mario del Valle Guijarro	
Dependencias	OBJ-002, OBJ-004	
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando un administrador desea eliminar un grupo de variables de entrada.	
Precondición	El usuario debe encontrarse identificado en el sistema como administrador.	
Secuencia normal	Paso	Acción
	1	El actor administrador (ACT-003) solicita eliminar un grupo de variables de entrada.
	2	El sistema muestra los grupos de variables de entrada existentes.
	3	El actor administrador (ACT-003) selecciona un grupo de variables de entrada.
	4	El sistema registra los cambios y muestra un mensaje informando del resultado de la operación.
Postcondición	Se ha eliminado el grupo de variables de entrada.	
Excepciones	Paso	Acción
	3	Si el usuario cancela la operación, a continuación este caso de uso queda sin efecto.
Importancia	Elevada	
Urgencia	Alta	
Estado	Completado	
Estabilidad	Alta	
Comentarios	Ninguno	

Tabla 71. CU-011 - Crear cuadro de mando - Eliminar grupo de entrada

CU-012	Crear cuadro de mando – Configuración de las variables juego	
Versión	1.0	
Autores	José Antonio González Alejos Mario del Valle Guijarro	
Dependencias	OBJ-002, OBJ-004	
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando un administrador desee configurar las variables juego de un nuevo cuadro de mando.	
Precondición	El usuario debe encontrarse identificado en el sistema como administrador.	
Secuencia normal	Paso	Acción
	1	El actor administrador (ACT-003) solicita configurar las variables juego del nuevo cuadro de mando.
	2	El sistema muestra una ventana para configurar las variables juego del nuevo cuadro de mando.
	3	El actor administrador (ACT-003) selecciona y configura cada una de las variables juego que quiere que aparezcan en el nuevo cuadro de mando.
	4	El sistema registra los cambios.
Postcondición	Se ha guardado la configuración de las variables juego.	
Excepciones	Paso	Acción
	3	Si el usuario deja en blanco el nombre de la variable juego, el sistema muestra un mensaje de error y vuelve al paso 2, a continuación este caso de uso continua.
	3	Si el usuario deja en blanco el nombre de las unidades de la variable juego, el sistema muestra un mensaje de error y vuelve al paso 2, a continuación este caso de uso continua.
	3	Si el usuario deja en blanco el rango de valores de la variable juego, el sistema muestra un mensaje de error y vuelve al paso 2, a continuación este caso de uso continua.
	3	Si el usuario deja en blanco el número de simulaciones para el juego, el sistema muestra un mensaje de error y vuelve al paso 2, a continuación este caso de uso continua.
	3	Si el usuario deja en blanco el nombre de las unidades de la simulación, el sistema muestra un mensaje de error y vuelve al paso 2, a continuación este caso de uso continua.
Importancia	Elevada	
Urgencia	Alta	
Estado	Completado	
Estabilidad	Alta	
Comentarios	Ninguno	

Tabla 72. CU-012 - Crear cuadro de mando - Configuración de las variables juego

CU-013	Crear cuadro de mando – Crear grupo de salida	
Versión	1.0	
Autores	José Antonio González Alejos Mario del Valle Guijarro	
Dependencias	OBJ-002, OBJ-004	
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando un administrador desee agrupar las variables de salida de un nuevo cuadro de mando.	
Precondición	El usuario debe encontrarse identificado en el sistema como administrador.	
Secuencia normal	Paso	Acción
	1	El actor administrador (ACT-003) solicita agrupar las variables de salida del nuevo cuadro de mando.
	2	El sistema muestra una ventana para agrupar las variables de salida del nuevo cuadro de mando.
	3	El actor administrador (ACT-003) selecciona el botón llamado Crear grupo.
	4	El sistema muestra una nueva ventana emergente con un campo de texto para introducir el nombre del grupo.
	5	El actor administrador (ACT-003) introduce el nombre del grupo.
	6	El sistema registra los cambios y muestra el nuevo grupo creado.
Postcondición	Se ha creado el nuevo grupo de variables salida.	
Excepciones	Paso	Acción
	5	Si el usuario deja en blanco el nombre del nuevo cuadro de mando, el sistema muestra un mensaje de error y vuelve al paso 4, a continuación este caso de uso continua.
	5	Si el usuario introduce un nombre de grupo ya existente en el modelo de datos, el sistema muestra un mensaje de error y vuelve al paso 2, a continuación este caso de uso continua.
	5	Si el usuario cancela la operación, a continuación este caso de uso queda sin efecto.
Importancia	Elevada	
Urgencia	Alta	
Estado	Completado	
Estabilidad	Alta	
Comentarios	Ninguno	

Tabla 73. CU-013 - Crear cuadro de mando – Crear grupo de salida

CU-014	Crear cuadro de mando – Eliminar grupo de salida	
Versión	1.0	
Autores	José Antonio González Alejos Mario del Valle Guijarro	
Dependencias	OBJ-002, OBJ-004	
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando un administrador desea eliminar un grupo de variables de salida.	
Precondición	El usuario debe encontrarse identificado en el sistema como administrador.	
Secuencia normal	Paso	Acción
	1	El actor administrador (ACT-003) solicita eliminar un grupo de variables de salida.
	2	El sistema muestra los grupos de variables de salida existentes.
	3	El actor administrador (ACT-003) selecciona un grupo de variables de salida.
	4	El sistema registra los cambios y muestra un mensaje informando del resultado de la operación.
Postcondición	Se ha eliminado el grupo de variables de salida.	
Excepciones	Paso	Acción
	3	Si el usuario cancela la operación, a continuación este caso de uso queda sin efecto.
Importancia	Elevada	
Urgencia	Alta	
Estado	Completado	
Estabilidad	Alta	
Comentarios	Ninguno	

Tabla 74. CU-014 - Crear cuadro de mando – Eliminar grupo de variables de salida

CU-015	Crear cuadro de mando – Configuración de las variables de salida	
Versión	1.0	
Autores	José Antonio González Alejos Mario del Valle Guijarro	
Dependencias	OBJ-002, OBJ-004	
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando un usuario desee configurar las variables de salida del nuevo cuadro de mando.	
Precondición	El usuario debe encontrarse identificado en el sistema como administrador.	
Secuencia normal	Paso	Acción
	1	El actor administrador (ACT-003) solicita configurar las variables de salida del nuevo cuadro de mando.
	2	El sistema muestra una ventana para configurar las variables de salida del nuevo cuadro de mando.
	3	El actor administrador (ACT-003) selecciona y configura cada una de las variables de salida que quiere que aparezcan en el nuevo cuadro de mando.
	4	El sistema registra los cambios.
Postcondición	Se ha guardado la configuración de las variables de salida.	
Excepciones	Paso	Acción
	3	Si el usuario deja en blanco el nombre de la variable de salida, el sistema muestra un mensaje de error y vuelve al paso 2, a continuación este caso de uso continua.
	3	Si el usuario deja en blanco el nombre de las unidades de la variable de salida, el sistema muestra un mensaje de error y vuelve al paso 2, a continuación este caso de uso continua.
Importancia	Elevada	
Urgencia	Alta	
Estado	Completado	
Estabilidad	Alta	
Comentarios	Ninguno	

Tabla 75. CU-015 - Crear cuadro de mando – Configuración de las variables de salida

CU-016	Modificar cuadro de mando – Datos cuadro mando	
Versión	1.0	
Autores	José Antonio González Alejos Mario del Valle Guijarro	
Dependencias	OBJ-002, OBJ-004	
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando un administrador desea modificar un cuadro de mando.	
Precondición	El usuario debe encontrarse identificado en el sistema como administrador.	
Secuencia normal	Paso	Acción
	1	El actor administrador (ACT-003) solicita modificar los datos de un cuadro de mando.
	2	El sistema muestra los cuadros de mando existentes.
	3	El actor administrador (ACT-003) selecciona un cuadro de mando y solicita ver sus datos actuales.
	4	El sistema muestra los datos actuales del cuadro de mando seleccionado y solicita los nuevos datos del cuadro de mando.
	5	El actor administrador (ACT-003) introduce los nuevos datos.
	6	El sistema registra los cambios del cuadro de mando y muestra un mensaje informando del resultado de la operación.
Postcondición	Se ha modificado el cuadro de mando.	
Excepciones	Paso	Acción
	5	Si el usuario cancela la operación, a continuación este caso de uso queda sin efecto.
	6	Si el usuario ha introducido los datos de manera incorrecta, el sistema muestra un mensaje de error y vuelve al paso 2, a continuación este caso de uso continúa.
Importancia	Elevada	
Urgencia	Alta	
Estado	Completado	
Estabilidad	Alta	
Comentarios	Ninguno	

Tabla 76. CU-016 - Modificar cuadro de mando – Datos cuadro mando

CU-017	Modificar cuadro de mando – Crear grupo de entrada	
Versión	1.0	
Autores	José Antonio González Alejos Mario del Valle Guijarro	
Dependencias	OBJ-002, OBJ-004	
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando un administrador desee modificar la agrupación de las variables de entrada.	
Precondición	El usuario debe encontrarse identificado en el sistema como administrador.	
Secuencia normal	Paso	Acción
	1	El actor administrador (ACT-003) solicita modificar la agrupación de las variables de entrada del nuevo cuadro de mando.
	2	El sistema muestra una ventana para agrupar las variables de entrada del cuadro de mando.
	3	El actor administrador (ACT-003) solicita crear un nuevo grupo o bien eliminar alguno de los existentes ya creados.
	4	El sistema muestra una nueva ventana emergente con un campo de texto para introducir el nombre del grupo en caso de la creación de un nuevo grupo. En caso de la eliminación de un grupo existente muestra una ventana de aviso para su confirmación.
	5	El actor administrador (ACT-003) introduce el nombre del grupo en caso de la creación de un nuevo grupo. Confirma la eliminación en caso de la eliminación de un grupo.
	6	El sistema registra los cambios y muestra el nuevo grupo creado.
Postcondición	Se ha creado el nuevo grupo de variables entrada.	
Excepciones	Paso	Acción
	5	Si el usuario deja en blanco el nombre del nuevo cuadro de mando, el sistema muestra un mensaje de error y vuelve al paso 4, a continuación este caso de uso continua.
	5	Si el usuario introduce un nombre de grupo ya existente en el modelo de datos, el sistema muestra un mensaje de error y vuelve al paso 2, a continuación este caso de uso continua.
	5	Si el usuario cancela la operación, a continuación este caso de uso queda sin efecto.
Importancia	Elevada	
Urgencia	Alta	
Estado	Completado	
Estabilidad	Alta	
Comentarios	Ninguno	

Tabla 77. CU-017 - Modificar cuadro de mando – Crear grupo de entrada

CU-018	Modificar cuadro de mando – Eliminar grupo de entrada	
Versión	1.0	
Autores	José Antonio González Alejos Mario del Valle Guijarro	
Dependencias	OBJ-002, OBJ-004	
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando un administrador desea eliminar un grupo de variables de entrada.	
Precondición	El usuario debe encontrarse identificado en el sistema como administrador.	
Secuencia normal	Paso	Acción
	1	El actor administrador (ACT-003) solicita eliminar un grupo de variables de entrada.
	2	El sistema muestra los grupos de variables de entrada existentes.
	3	El actor administrador (ACT-003) selecciona un grupo de variables de entrada.
	4	El sistema registra los cambios y muestra un mensaje informando del resultado de la operación.
Postcondición	Se ha eliminado el grupo de variables de entrada.	
Excepciones	Paso	Acción
	3	Si el usuario cancela la operación, a continuación este caso de uso queda sin efecto.
Importancia	Elevada	
Urgencia	Alta	
Estado	Completado	
Estabilidad	Alta	
Comentarios	Ninguno	

Tabla 78. CU-018 - Modificar cuadro de mando - Eliminar grupo de entrada

CU-019	Modificar cuadro de mando – Configuración de las variables juego	
Versión	1.0	
Autores	José Antonio González Alejos Mario del Valle Guijarro	
Dependencias	OBJ-002, OBJ-004	
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando un administrador desee configurar las variables juego de un cuadro de mando.	
Precondición	El usuario debe encontrarse identificado en el sistema como administrador.	
Secuencia normal	Paso	Acción
	1	El actor administrador (ACT-003) solicita configurar las variables juego del cuadro de mando.
	2	El sistema muestra una ventana para configurar las variables juego del cuadro de mando.
	3	El actor administrador (ACT-003) selecciona y configura cada una de las variables juego que quiere que aparezcan en el cuadro de mando.
	4	El sistema registra los cambios.
Postcondición	Se ha guardado la configuración de las variables juego.	
Excepciones	Paso	Acción
	3	Si el usuario deja en blanco el nombre de la variable juego, el sistema muestra un mensaje de error y vuelve al paso 2, a continuación este caso de uso continua.
	3	Si el usuario deja en blanco el nombre de las unidades de la variable juego, el sistema muestra un mensaje de error y vuelve al paso 2, a continuación este caso de uso continua.
	3	Si el usuario deja en blanco el rango de valores de la variable juego, el sistema muestra un mensaje de error y vuelve al paso 2, a continuación este caso de uso continua.
	3	Si el usuario deja en blanco el número de simulaciones para el juego, el sistema muestra un mensaje de error y vuelve al paso 2, a continuación este caso de uso continua.
	3	Si el usuario deja en blanco el nombre de las unidades de la simulación, el sistema muestra un mensaje de error y vuelve al paso 2, a continuación este caso de uso continua.
Importancia	Elevada	
Urgencia	Alta	
Estado	Completado	
Estabilidad	Alta	
Comentarios	Ninguno	

Tabla 79. CU-019 - Modificar cuadro de mando - Configuración de las variables juego

CU-020	Modificar cuadro de mando – Crear grupo de salida	
Versión	1.0	
Autores	José Antonio González Alejos Mario del Valle Guijarro	
Dependencias	OBJ-002, OBJ-004	
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando un administrador desee modificar la agrupación de las variables de salida de un cuadro de mando.	
Precondición	El usuario debe encontrarse identificado en el sistema como administrador.	
Secuencia normal	Paso	Acción
	1	El actor administrador (ACT-003) solicita modificar la agrupación de las variables de salida del cuadro de mando.
	2	El sistema muestra una ventana para agrupar las variables de salida del nuevo cuadro de mando.
	3	El actor administrador (ACT-003) selecciona crear un nuevo grupo o bien eliminar uno de los existentes.
	4	El sistema muestra una nueva ventana emergente con un campo de texto para introducir el nombre del grupo en caso de crear un nuevo grupo. En caso de la eliminación de un grupo existente, el sistema solicita una confirmación.
	5	El actor administrador (ACT-003) introduce el nombre del grupo en caso de la creación de un nuevo grupo. Confirma la eliminación del grupo en el otro caso.
	6	El sistema registra los cambios y muestra el nuevo grupo creado.
Postcondición	Se ha creado el nuevo grupo de variables salida.	
Excepciones	Paso	Acción
	5	Si el usuario deja en blanco el nombre del nuevo cuadro de mando, el sistema muestra un mensaje de error y vuelve al paso 4, a continuación este caso de uso continua.
	5	Si el usuario introduce un nombre de grupo ya existente en el modelo de datos, el sistema muestra un mensaje de error y vuelve al paso 2, a continuación este caso de uso continua.
	5	Si el usuario cancela la operación, a continuación este caso de uso queda sin efecto.
Importancia	Elevada	
Urgencia	Alta	
Estado	Completado	
Estabilidad	Alta	
Comentarios	Ninguno	

Tabla 80. CU-020 - Modificar cuadro de mando - Agrupación de las variables salida

CU-021	Modificar cuadro de mando – Eliminar grupo de salida	
Versión	1.0	
Autores	José Antonio González Alejos Mario del Valle Guijarro	
Dependencias	OBJ-002, OBJ-004	
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando un administrador desea eliminar un grupo de variables de salida.	
Precondición	El usuario debe encontrarse identificado en el sistema como administrador.	
Secuencia normal	Paso	Acción
	1	El actor administrador (ACT-003) solicita eliminar un grupo de variables de salida.
	2	El sistema muestra los grupos de variables de salida existentes.
	3	El actor administrador (ACT-003) selecciona un grupo de variables de salida.
	4	El sistema registra los cambios y muestra un mensaje informando del resultado de la operación.
Postcondición	Se ha eliminado el grupo de variables de salida.	
Excepciones	Paso	Acción
	3	Si el usuario cancela la operación, a continuación este caso de uso queda sin efecto.
Importancia	Elevada	
Urgencia	Alta	
Estado	Completado	
Estabilidad	Alta	
Comentarios	Ninguno	

Tabla 81. CU-021 Modificar cuadro de mando - Eliminar grupo de salida

CU-022	Modificar cuadro de mando – Configuración de las variables salida	
Versión	1.0	
Autores	José Antonio González Alejos Mario del Valle Guijarro	
Dependencias	OBJ-002, OBJ-004	
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando un usuario desee configurar las variables de salida del cuadro de mando.	
Precondición	El usuario debe encontrarse identificado en el sistema como administrador.	
Secuencia normal	Paso	Acción
	1	El actor administrador (ACT-003) solicita configurar las variables de salida del cuadro de mando.
	2	El sistema muestra una ventana para configurar las variables de salida del cuadro de mando.
	3	El actor administrador (ACT-003) selecciona y configura cada una de las variables de salida que quiere que aparezcan en el cuadro de mando.
	4	El sistema registra los cambios.
Postcondición	Se ha guardado la configuración de las variables de salida.	
Excepciones	Paso	Acción
	3	Si el usuario deja en blanco el nombre de la variable de salida, el sistema muestra un mensaje de error y vuelve al paso 2, a continuación este caso de uso continua.
	3	Si el usuario deja en blanco el nombre de las unidades de la variable de salida, el sistema muestra un mensaje de error y vuelve al paso 2, a continuación este caso de uso continua.
Importancia	Elevada	
Urgencia	Alta	
Estado	Completado	
Estabilidad	Alta	
Comentarios	Ninguno	

Tabla 82. CU-022 - Modificar cuadro de mando - Configuración de las variables salida

CU-023	Eliminar cuadro de mando	
Versión	1.0	
Autores	José Antonio González Alejos Mario del Valle Guijarro	
Dependencias	OBJ-002, OBJ-004	
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando un administrador desea eliminar un cuadro de mando.	
Precondición	El usuario debe encontrarse identificado en el sistema como administrador.	
Secuencia normal	Paso	Acción
	1	El actor administrador (ACT-003) solicita eliminar un cuadro de mando.
	2	El sistema muestra los cuadros de mando existentes.
	3	El actor administrador (ACT-003) selecciona un cuadro de mando.
	4	El sistema registra los cambios y muestra un mensaje informando del resultado de la operación.
Postcondición	Se ha eliminado el cuadro de mando.	
Excepciones	Paso	Acción
	3	Si el usuario cancela la operación, a continuación este caso de uso queda sin efecto.
Importancia	Elevada	
Urgencia	Alta	
Estado	Completado	
Estabilidad	Alta	
Comentarios	Ninguno	

Tabla 83. CU-023 - Eliminar cuadro de mando

Seguidamente vamos a ver los casos de uso del actor Usuario

CU-024	Simular	
Versión	1.0	
Autores	José Antonio González Alejos Mario del Valle Guijarro	
Dependencias	OBJ-001, OBJ-003	
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando un usuario desea realizar una simulación de un cuadro de mando.	
Precondición	El usuario debe encontrarse identificado en el sistema.	
Secuencia normal	Paso	Acción
	1	El actor usuario (ACT-001) solicita simular con un cuadro de mando.
	2	El sistema muestra los cuadros de mando existentes.
	3	El actor usuario (ACT-001) selecciona un cuadro de mando.
	4	El sistema carga las variables palanca de dicho cuadro de mando y en la parte derecha las opciones gráficas correspondientes.
	5	El actor usuario (ACT--001) selecciona el valor de cada una de las variables palanca.
	6	El sistema realiza la simulación con los valores seleccionados por el usuario y lo muestra gráficamente con diversas gráficas.
Postcondición	Se ha realizado una simulación con el cuadro de mando.	
Excepciones	Paso	Acción
	3	Si el usuario cancela la operación, a continuación este caso de uso queda sin efecto.
Importancia	Elevada	
Urgencia	Alta	
Estado	Completado	
Estabilidad	Alta	
Comentarios	Ninguno	

Tabla 84. CU-024 - Simular

CU-025	Guardar Partida	
Versión	1.0	
Autores	José Antonio González Alejos Mario del Valle Guijarro	
Dependencias	OBJ-003	
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando un usuario desea guardar una partida.	
Precondición	El usuario debe encontrarse identificado en el sistema.	
Secuencia normal	Paso	Acción
	1	El actor usuario (ACT-001) solicita guardar partida.
	2	El sistema muestra los una ventana emergente donde introducir los datos.
	3	El actor usuario (ACT-001) introduce los datos de la partida.
	4	El sistema registra los cambios y muestra un mensaje informando del resultado de la operación.
Postcondición	Se ha guardado una partida.	
Excepciones	Paso	Acción
	3	Si el usuario cancela la operación, a continuación este caso de uso queda sin efecto.
Importancia	Elevada	
Urgencia	Alta	
Estado	Completado	
Estabilidad	Alta	
Comentarios	Ninguno	

Tabla 85. CU-025 - Guardar Partida

CU-026	Cargar Partida	
Versión	1.0	
Autores	José Antonio González Alejos Mario del Valle Guijarro	
Dependencias	OBJ-003	
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando un usuario desea cargar una partida.	
Precondición	El usuario debe encontrarse identificado en el sistema.	
Secuencia normal	Paso	Acción
	1	El actor usuario (ACT-001) solicita cargar una partida.
	2	El sistema muestra las partidas almacenadas en el modelo de datos y que se pueden cargar.
	3	El actor usuario (ACT-001) selecciona una partida a cargar.
	4	El sistema carga la partida seleccionada por el usuario.
Postcondición	Se ha cargado una partida.	
Excepciones	Paso	Acción
	3	Si el usuario cancela la operación, a continuación este caso de uso queda sin efecto.
Importancia	Elevada	
Urgencia	Alta	
Estado	Completado	
Estabilidad	Alta	
Comentarios	Ninguno	

Tabla 86. CU-026 - Cargar Partida

CU-027	Ver Partida	
Versión	1.0	
Autores	José Antonio González Alejos Mario del Valle Guijarro	
Dependencias	OBJ-003	
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando un usuario desea ver una partida.	
Precondición	El usuario debe encontrarse identificado en el sistema.	
Secuencia normal	Paso	Acción
	1	El actor usuario (ACT-001) solicita ver una partida.
	2	El sistema muestra las partidas almacenadas en el modelo de datos.
	3	El actor usuario (ACT-001) selecciona la partida a visualizar.
	4	El sistema muestra la partida seleccionada.
Postcondición	Se ha visualizado una partida.	
Excepciones	Paso	Acción
	3	Si el usuario cancela la operación, a continuación este caso de uso queda sin efecto.
Importancia	Elevada	
Urgencia	Alta	
Estado	No implementado	
Estabilidad	Alta	
Comentarios	Ninguno	

Tabla 87. CU-027 - Ver Partida

CU-028	Volver a jugar	
Versión	1.0	
Autores	José Antonio González Alejos Mario del Valle Guijarro	
Dependencias	OBJ-003	
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando un usuario desea volver a jugar una partida.	
Precondición	El usuario debe encontrarse identificado en el sistema.	
Secuencia normal	Paso	Acción
	1	El actor usuario (ACT-001) solicita volver a jugar una partida.
	2	El sistema vuelve a cargar la partida con todos los datos inicializados a cero.
Postcondición	Se ha vuelto a jugar una partida.	
Excepciones	Paso	Acción
	3	Si el usuario cancela la operación, a continuación este caso de uso queda sin efecto.
Importancia	Elevada	
Urgencia	Alta	
Estado	Completado	
Estabilidad	Alta	
Comentarios	Ninguno	

Tabla 88. CU-028 - Volver a jugar

CU-029	Editar Perfil	
Versión	1.0	
Autores	José Antonio González Alejos Mario del Valle Guijarro	
Dependencias	OBJ-002, OBJ-003, OBJ-004	
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando un usuario desea editar el perfil.	
Precondición	El usuario debe encontrarse identificado en el sistema.	
Secuencia normal	Paso	Acción
	1	El actor usuario (ACT-001) solicita editar el perfil.
	2	El sistema muestra una nueva ventana emergente donde poder introducir los datos relacionados con el perfil de usuario.
	3	El actor usuario (ACT-001) modifica los datos relacionados con su perfil.
	4	El sistema registra los cambios y muestra un mensaje informando del resultado de la operación.
Postcondición	Se ha cargado una partida.	
Excepciones	Paso	Acción
	3	Si el usuario cancela la operación, a continuación este caso de uso queda sin efecto.
Importancia	Elevada	
Urgencia	Alta	
Estado	Completado	
Estabilidad	Alta	
Comentarios	Ninguno	

Tabla 89. CU-029 - Editar Perfil

3.2.10. Diagrama de clases de análisis

En la figura que viene a continuación mostramos el diagrama de clases de análisis. Inicialmente se muestra un modelo de clases de todas ellas sin atributos ni operaciones, para poder tener una visión de conjunto de todas las clases y sus asociaciones. Posteriormente se verán más en detalle.

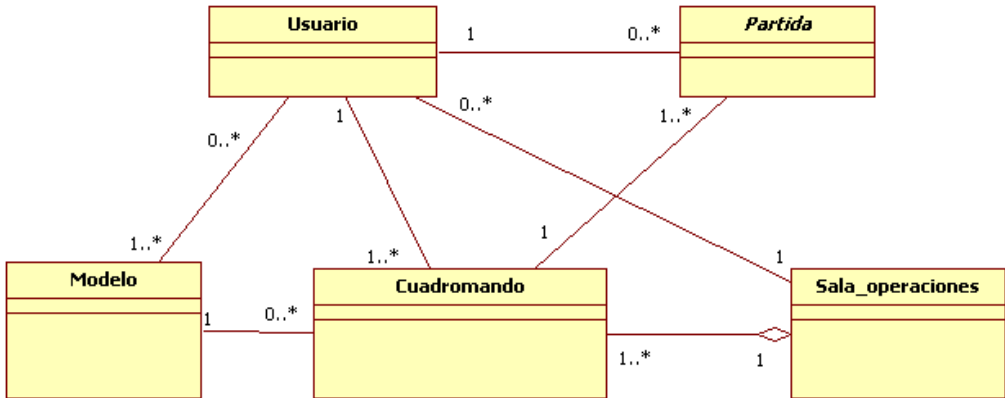


Figura 12. Diagrama de clases de análisis

3.2.11. Diagrama Entidad-Relación de la base de datos

A continuación definimos de manera detallada el modelo de datos necesario para representar y almacenar toda la información procesada por el sistema. Representa los elementos principales así como sus interrelaciones, conformando la estructura completa de datos que se necesita para la información que procesa la aplicación. Se han eliminado los atributos de las entidades para así poder mostrar con mayor claridad la información.

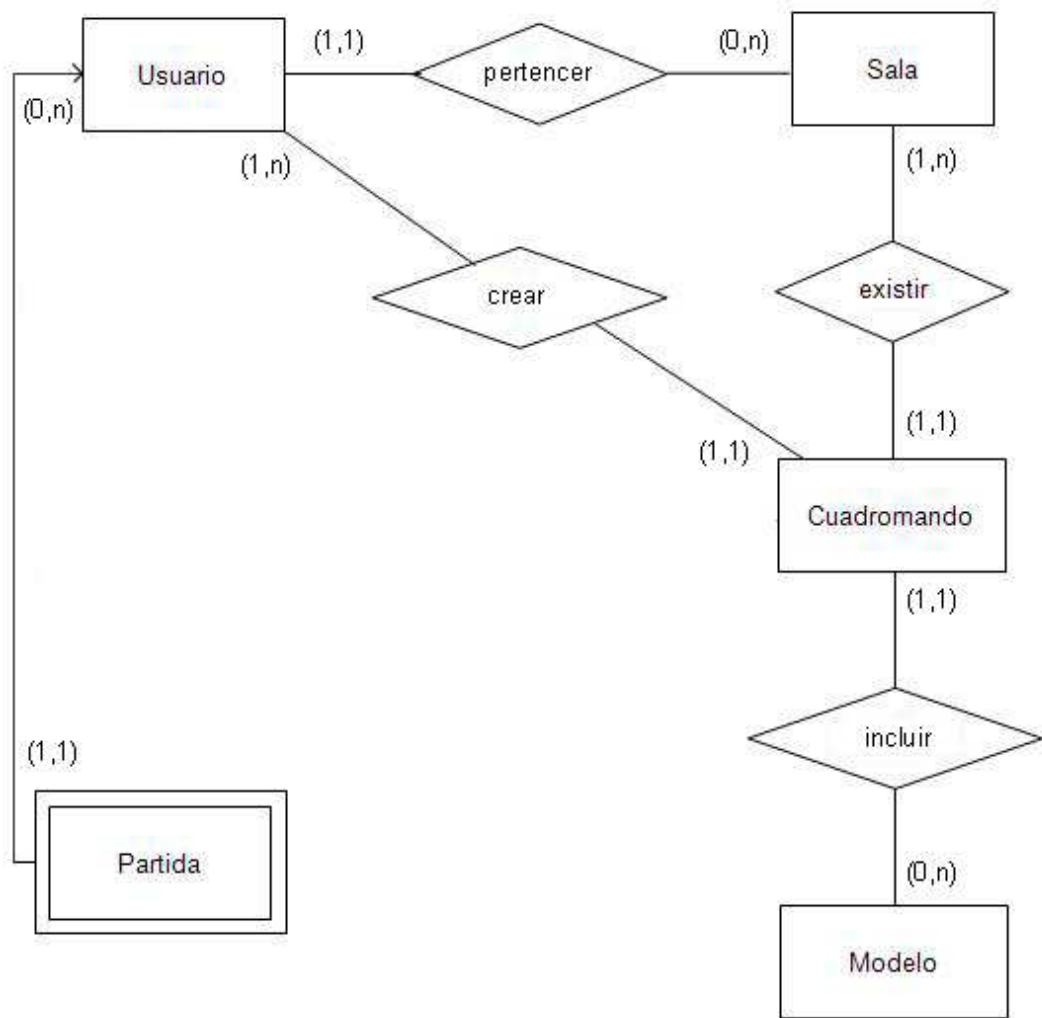


Figura 13. Diagrama Entidad Relación

3.2.12. Modelo relacional del análisis

A partir del diagrama de Entidad Relación de la figura anterior se obtiene cada instancia de entidad que representa tanto la información interna de la propia entidad, única e identificada unitariamente, así como las relaciones entre el resto de entidades.

Veamos a continuación las diferentes entidades del modelo entidad – relación y su cometido:

- **Entidad Usuario:** Almacena los datos de los usuarios presentes en el sistema.
- **Entidad Sala:** Almacena las salas de operaciones donde los usuarios van a simular.
- **Entidad Cuadromando:** Almacena los cuadros de mando que van a poder ser simulados.
- **Entidad Modelo:** Almacena la información perteneciente a los modelos.
- **Entidad Partida:** Almacena las partidas realizadas por los usuarios.

3.3. Documento de Diseño

3.3.1. Propósito del documento de diseño

El propósito de este documento es mostrar el modelado del sistema incluyendo su arquitectura, teniendo en cuenta que debe cumplir con los requisitos y las restricciones analizadas previamente.

3.3.2. Arquitectura del sistema

El sistema a desarrollar es una aplicación web dinámica. La aplicación se alojará en un servidor que soportará la lógica de negocio a la cual accederá el usuario mediante la navegación por las distintas páginas. Los resultados solicitados por el usuario serán mostrados en otras páginas generadas por el servidor. Dichos resultados serán recuperados y almacenados en una base de datos.

Por tanto es necesario un servidor web, un sistema gestor de bases de datos para almacenar los datos generados por la aplicación y permitir su posterior acceso, una conexión a dicho servidor por parte del cliente y un navegador web para poder manejarla.

Para la representación de la arquitectura se va emplear el lenguaje unificado de modelado UML, en dos vistas principales.

- Visión de casos de uso.
- Visión lógica.

A lo largo del desarrollo se ha procurado seguir el patrón de arquitectura software “Modelo Vista Controlador” (MVC) en la medida de lo posible, ya que la aplicación debía comunicarse con Vensim y acceder a su base de datos, jerarquizando las funcionalidades en diferentes niveles y separando los datos de la aplicación, la interfaz de usuario, y la lógica de control en tres componentes distintos.

La arquitectura seleccionada se compondrá de diversos nodos que se comunicarán entre sí:

- **Cliente (navegador web):** se corresponderá con un cliente web donde el usuario podrá disponer de las páginas enviadas por el servidor web, así como la posibilidad de introducir datos que puedan ser necesarios para la funcionalidad del propio sistema. Dispondrá de scripts de validación de datos introducidos.
- **Servidor Web:** será el encargado de recoger los datos que provengan del cliente y suministrárselos al servidor de aplicación. También enviará datos provenientes de la aplicación al cliente. Será además el encargado de almacenar las páginas Web.
- **SGBD:** El SGBD será el encargado de gestionar el almacenamiento de datos de nuestra aplicación así como obtención de ellos.

Una vez vista los componentes de la arquitectura propuesta, nuestro sistema además dispondrá de una arquitectura de tres capas (visión lógica):

- **Capa de presentación:** En esta capa aparecerán los elementos con los que pueda interactuar el usuario: las diversas pantallas, formularios...
- **Capa de negocio:** En la capa de negocio es donde se englobará la funcionalidad básica de nuestro sistema, así como las reglas de las funciones de la herramienta. Esta capa se comunica con la capa de presentación, para recibir las solicitudes del usuario y presentará los resultados, y con la capa de datos, para solicitar datos al gestor de base de datos para almacenarlos o recuperarlos.
- **Capa de datos o almacenamiento:** se gestionarán en esta capa los datos persistentes.

En cuanto a los niveles (visión física) será una arquitectura de 3 capas y 2 niveles, ya que la capa de presentación estará en el cliente y la capa de negocio y de almacenamiento en el servidor.

Para desacoplar las funciones del sistema (presentación, acceso y tratamiento de datos), además de la descomposición en capas, se va a implementar el patrón arquitectónico Modelo-Vista-Controlador (MVC).

3.3.2.1. Patrón Modelo Vista Controlador (MVC)

Se ha elegido este patrón para la implementación porque es capaz de independizar la representación externa de los datos de su tipo interno (su tipo en la base de datos) y puesto que ambos pueden cambiar con mayor o menor frecuencia, logra que un cambio en uno de ellos tenga una mínima repercusión en el otro.

El esquema del patrón se puede representar según la siguiente figura:

Para adaptar el patrón al desarrollo, tendremos que tener en cuenta los distintos componentes:

- **Modelo:** Contendrá las clases del sistema necesarias para el manejo de los datos de la aplicación.
- **Vista:** Contendrá aquellas páginas que se mostrarán en el navegador. Su estructura será HTML junto con otro lenguaje de servidor embebido.
- **Controlador:** Contendrá los controladores de la aplicación. Los controladores serán los encargados de recoger los eventos (en nuestra aplicación, la pulsación en los enlaces de las páginas) y comunicar tanto al modelo como a la vista, las acciones necesarias para llevar a cabo la operación.

3.3.3. Tecnología empleada

Para la implementación de la aplicación se han elegido las siguientes tecnologías, teniendo en cuenta el requisito del cliente de usar tecnologías gratuitas:

3.3.3.1. Servidor Web

Como servidor Web, el cliente ha sugerido el uso de Apache en su versión 2.2. Apache es un servidor desarrollado como proyecto libre, no está sujeto a pagos de licencias y por tanto es gratuito. Actualmente es usado por un 65% de usuarios de servidores Web. Tiene la ventaja de que se puede acoplar sin problemas módulos para trabajar con el lenguaje JSP y ha ofrecido buenos resultados en rendimiento y en seguridad.

3.3.3.2. Lenguaje de programación

Para el diseño de las páginas web se usará el lenguaje HTML apoyándose en hojas de estilo (CSS) para la maquetación. Para manejar los datos e implementar la lógica de negocio se usará JSP.

Se ha elegido JSP porque para comunicarnos con Vensim se ha empleado Java, aparte que ya nos venía elegido por los desarrolladores de la anterior fase del proyecto. Dicha elección se realizó debido a que las opciones de comunicación que ofrece Vensim son C/C++, Visual Basic, Pascal y Java. Además cumple que es un lenguaje gratuito y presenta facilidad de integración con sistemas gestores de base de datos como MySQL, que como se verá a continuación ha sido el elegido para este proyecto.

Para programar ciertos contenidos que se ejecuten en el cliente, se usará JavaScript. Es un lenguaje interpretado por el navegador web que presenta una clara orientación a eventos. Los eventos ocurren cuando hay una interacción del cliente con la aplicación, por ejemplo cuando el usuario pincha en un enlace o cuando se establece el foco para rellenar un campo de texto. El uso de JavaScript será lo más limitado posible, ya que un uso indebido puede ralentizar la ejecución al sobrecargar el equipo. En la aplicación se utilizará sobre todo en la validación de formularios y en la carga de datos dinámicamente.

A continuación hacemos una breve descripción de los lenguajes empleados para conocer más a fondo qué son cada uno y lo que nos ofrece.

Java

Java es un lenguaje de programación orientado a objetos desarrollado por James Gosling y sus compañeros de Sun Microsystems al inicio de la década de 1990. A diferencia de los lenguajes de programación convencionales, que generalmente están diseñados para ser compilados a código nativo, Java es compilado en un bytecode que es ejecutado (usando normalmente un compilador JIT), por una máquina virtual Java.

Sus principales características son:

Fácil aprendizaje: El único requerimiento para aprender Java es tener una comprensión de los conceptos básicos de la programación orientada a objetos. Así se ha creado un lenguaje simple (aunque eficaz y expresivo) pudiendo mostrarse cualquier planteamiento por parte del programador sin que las interioridades del sistema subyacente sean desveladas.

Java es más complejo que un lenguaje simple, pero más sencillo que cualquier otro entorno de programación. El único obstáculo que se puede presentar es conseguir comprender la programación orientada a objetos, aspecto que, al ser independiente del lenguaje, se presenta como insalvable.

Robusto: Java fue diseñado para crear software altamente fiable. Para ello proporciona numerosas comprobaciones en compilación y en tiempo de ejecución. Sus características de memoria liberan a los programadores de una familia entera de errores (la aritmética de punteros), ya que se han prescindido por completo los punteros, y la recolección de basura elimina la necesidad de liberación explícita de memoria.

Orientado a objetos: Java fue diseñado como un lenguaje orientado a objetos desde el principio. Los objetos agrupan en estructuras encapsuladas tanto sus datos como los métodos (o funciones) que manipulan esos datos. La tendencia del futuro, a la que Java se suma, apunta hacia la programación orientada a objetos, especialmente en entornos cada vez más complejos y basados en red.

Seguro: Dada la naturaleza distribuida de Java, donde las applets se bajan desde cualquier punto de la Red, la seguridad se impuso como una necesidad de vital importancia. A nadie le gustaría ejecutar en su ordenador programas con acceso total a su sistema, procedentes de fuentes desconocidas. Así que se implementaron barreras de seguridad en el lenguaje y en el sistema de ejecución en tiempo real.

Indiferente a la arquitectura: Java está diseñado para soportar aplicaciones que serán ejecutadas en los más variados entornos de red, desde Unix a Windows NT, pasando por Mac y estaciones de trabajo, sobre arquitecturas distintas y con sistemas operativos diversos. Para acomodar requisitos de ejecución tan variados, el compilador de Java genera bytecodes: un formato intermedio indiferente a la arquitectura diseñado para transportar el código eficientemente a múltiples plataformas hardware y software. El resto de problemas los soluciona el intérprete de Java.

Portable: La indiferencia a la arquitectura representa sólo una parte de su portabilidad. Además, Java especifica los tamaños de sus tipos de datos básicos y el comportamiento de sus operadores aritméticos, de manera que los programas son iguales en todas las plataformas.

Estas dos últimas características se conocen como la Máquina Virtual Java (JVM).

Distribuido: Java proporciona una colección de clases para su uso en aplicaciones de red, que permiten abrir sockets y establecer y aceptar conexiones con servidores o clientes remotos, facilitando así la creación de aplicaciones distribuidas.

Interpretado y compilado a la vez: Java es compilado, en la medida en que su código fuente se transforma en una especie de código máquina, los bytecodes, semejantes a las instrucciones de ensamblador.

Por otra parte, es interpretado, ya que los bytecodes se pueden ejecutar directamente sobre cualquier máquina a la cual se hayan portado el intérprete y el sistema de ejecución en tiempo real (run-time).

Multihebra: Hoy en día ya se ven muy limitadas las aplicaciones que sólo pueden ejecutar una acción a la vez. Java soporta sincronización de múltiples hilos de ejecución (multithreading) a nivel de lenguaje, especialmente útiles en la creación de aplicaciones de red distribuidas. Así, mientras un hilo se encarga de la comunicación, otro puede interactuar con el usuario mientras otro presenta una animación en pantalla y otro realiza cálculos.

Dinámico: El lenguaje Java y su sistema de ejecución en tiempo real son dinámicos en la fase de enlazado. Las clases sólo se enlazan a medida que son necesitadas. Se pueden enlazar nuevos módulos de código bajo demanda, procedente de fuentes muy variadas, incluso desde la Red.

En Java podemos crear los siguientes tipos de aplicaciones:

- Aplicaciones: Se ejecutan sin necesidad de navegador.
- Applets: se pueden descargar de Internet y se observan en el navegador.
- JavaBeans: Componentes software de Java, que se puedan incorporar gráficamente a otros componentes.
- JavaScript: Conjunto del lenguaje Java que puede codificarse directamente sobre cualquier documento HTML.

- Servlets: Módulos que permiten sustituir o utilizar el lenguaje Java en lugar de programas CGI (Common Gateway Interface) a la hora de dotar de interactividad a las páginas Web.

Servlets

Un servlet es un programa ejecutado en el servidor (a diferencia de los applets que se ejecutan en el cliente). Es un mecanismo para implantar aplicaciones en el lado del servidor. Reciben peticiones y mandan resultados en HTTP, siendo el formato más común de salida una página HTML o un archivo XML (pero puede ser cualquier tipo MIME, una imagen, un objeto serializado, etc).

El proceso de ejecución de un servlet es el siguiente:

- El proceso empieza con la petición HTTP que llega en primer lugar al servidor Web (ejemplo Apache Tomcat).
- La petición se traslada al contenedor de aplicaciones, en concreto a su motor de Servicio Servlet/JSP (con su propia JVM).
- El motor encapsula la información de la petición en un objeto del tipo `HttpServletRequest`, además encapsula en un objeto `HttpServletResponse` el flujo de respuesta.
- El motor crea por cada petición un hilo, sobre el que invoca la función `service()` del servlet. En función del método de la petición POST o GET, `service()` llamará al método correspondiente del servlet, `doPost()`, `doGet()` ... pasándoles los objetos de `HttpServletRequest` y `HttpServletResponse`. En realidad hay más métodos HTTP (trace, etc.), pero GET y POST son los más habituales.
- Cada clase del tipo servlet tiene una única instancia, sobre la que corren los diferentes hilos (peticiones).

Cada petición genera un hilo independiente. Pero solo la primera petición genera una instancia de la clase servlet. Podremos tener N peticiones al servlet, por tanto N hilos, pero sólo una instancia de la clase (un objeto).

El esquema es muy parecido al que existe en el CGI (Common Gateway Interface).

Ventajas de los servlets:

- Multiplataforma (es Java).
- Cada petición genera un hilo dentro de un proceso. Frente a la arquitectura CGI, donde cada petición genera un proceso. Existe el interfaz `SingleThreadModel` para indicar que se traten las peticiones de forma secuencial (un hilo).

JSP

Las JSP son desde un punto de vista funcional semejantes a los servlets: se convierte la página JSP en un servlet la primera vez que la página es invocada. Por ello en ocasiones nos referimos a ellas como páginas que generan 'dinámicamente' servlets. La primera vez el contenedor compila, carga en memoria y ejecuta el servlet (debido a esta acumulación de tareas lo normal es que la primera ejecución sea un poco lenta y, por ello, es realizada por el programador). En algunos contenedores se precompila el servlet. Sea cual sea el tipo de contenedor utilizado, al final se accede a un servlet.

Un servlet está realizado íntegramente en Java y su salida es un objeto que tiene el formato solicitado por el cliente (HTML, XML, etc.). Una JSP se escribe en código Java y en directivas específicas que se mezclan con código HTML

Ventajas de las JSP:

- Los servlets exigen un mayor conocimiento de Java. Con JSP tenemos una tecnología orientada a la presentación, en donde los creadores de HTML se pueden sentir más a gusto que con el “puro Java”.
- El programador puede modificar las páginas JSP y el contenedor JSP las recompila y recarga en memoria la primera vez que las invocamos. El programador no tiene que compilar (al igual que ocurre con el HTML).

3.3.3.3. Sistema Gestor de Bases de Datos

Como sistema gestor de base de datos (SGBD) se ha seleccionado MySQL en su versión 5.5. La elección de este sistema gestor viene motivada en primer lugar por ser gratuito, funciona perfectamente bajo servidores Apache y es el más usado en Internet por su velocidad

La integración de los tres componentes es tan fuerte, que resulta difícil hablar de uno de ellos sin mencionar a cualquiera de los otros. En Internet, se puede encontrar aplicaciones muy útiles para usuarios sin experiencia, que instalan y configuran en la máquina los tres elementos, aunque se aconseja la instalación de cada uno de ellos de manera independiente.

A continuación vamos a mostrar el diagrama de clases en la siguiente figura mostrando sus clases, atributos y las relaciones entre ellas. Este diagrama es utilizado durante el proceso de análisis y diseño de los sistemas, donde se crea el diseño conceptual de la información que se manejará en el sistema, y los componentes que se encargaran del funcionamiento y la relación entre uno y otro.

3.3.4. Diagrama de clases

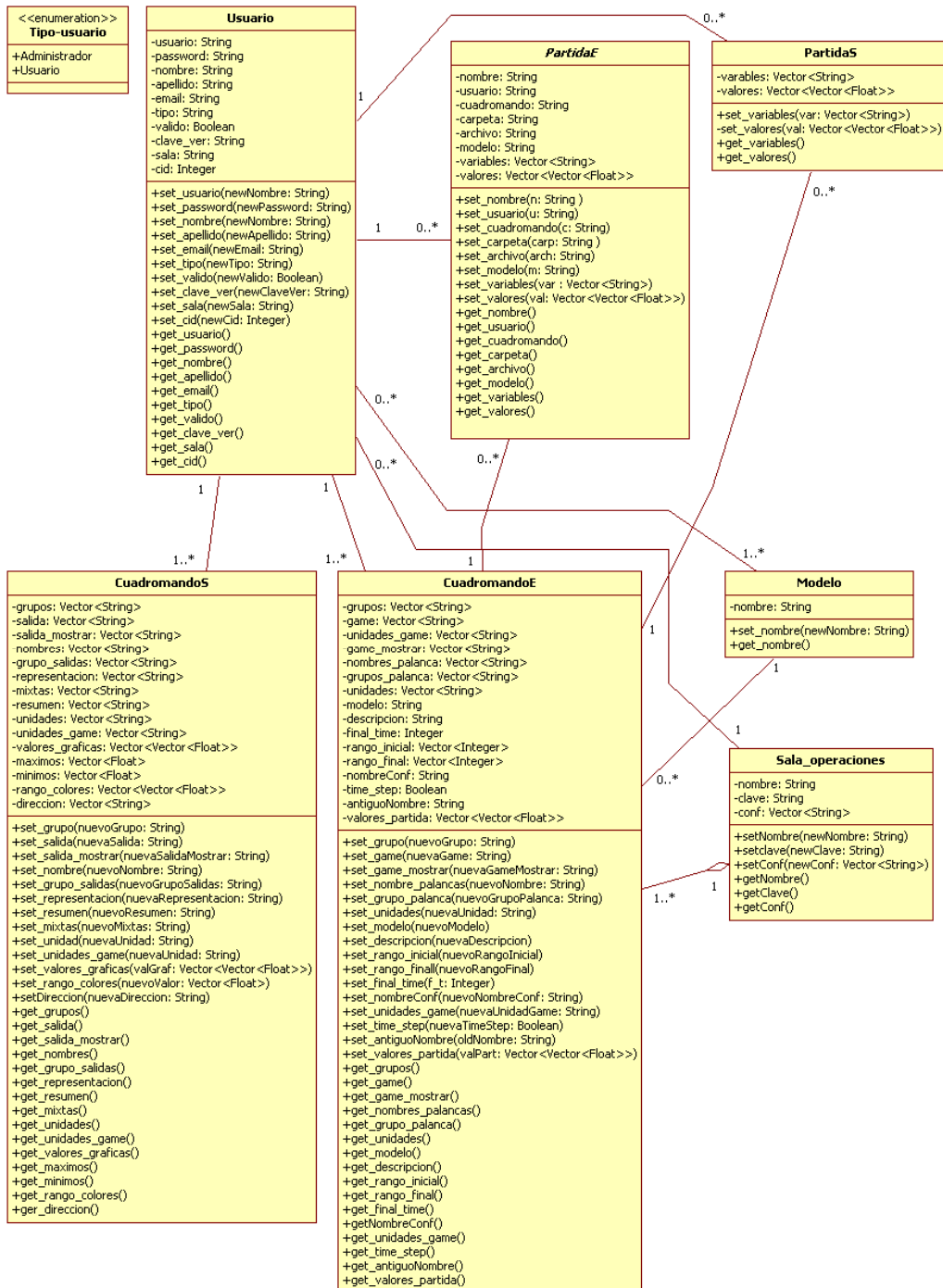


Figura 14. Diagrama de clases de diseño

3.3.5. Descripción de las clases

A continuación pasamos a detallar las clases con mayor relevancia utilizadas en este proyecto. Hemos decidido no incluir las clases que actúan como controladores debido a que son un gran número de ellos.

Clase	Usuario
Responsabilidad	Clase encargada de gestionar la información relacionada con el usuario que interacciona con la aplicación.
Atributos	<ul style="list-style-type: none"> • login • password • nombre • apellido • tipo • email • claveVer • sala • cid
Operaciones	<ul style="list-style-type: none"> • getNombre() • setNombre(String newNombre) • getApellido() • setApellido(String newApellido) • getPassword() • setPassword(String newPassword) • getCid() • setCid(int newCid) • getUsuario() • setUsuario(String newUsuario) • esValido() • setValido(boolean newValido) • getEmail() • setEmail(String nuevoEmail) • getTipo() • setTipo(String nuevoTipo) • getClaveVer() • setClaveVer(String nuevaClaveVer) • getSala() • setSala(String nuevaSala)

Tabla 90. Descripción de la clase Usuario

Clase	Sala_operaciones
Responsabilidad	Clase encargada de gestionar la información relacionada con las salas de operaciones.
Atributos	<ul style="list-style-type: none">• nombre• clave• conf
Operaciones	<ul style="list-style-type: none">• setNombre(String newNombre)• getNombre()• setClave(String newClave)• get_clave()• set_conf(Vector<String> c)• set_rutaModelo(StringNewRutaModelo)• set_bbdd(Vector<String> getConf)

Tabla 91. Descripción de la clase Sala_operaciones

Clase	CuadromandoE
Responsabilidad	Clase encargada de gestionar la información relacionada con el usuario que interacciona con la aplicación.
Atributos	<ul style="list-style-type: none">• grupos• game• unidades_game• game_mostrar• nombres_palancas• grupo_palanca• unidades• modelo• descripcion• final_time• rango_inicial• rango_final• nombreConf• time_step• antiguoNombre• valores_partida

Operaciones	<ul style="list-style-type: none"> • set_grupo(String nuevoGrupo) • set_game(String nuevaGame) • set_game_mostrar(String nuevaGameMostrar) • set_nombre_palancas(String nuevoNombre) • set_grupo_palanca(String nuevoGrupoPalanca) • set_unidades(String nuevaUnidad) • set_modelo(String nuevoModelo) • set_descripcion(String nuevaDescripcion) • set_rango_inicial(int nuevoRangoInicial) • set_rango_final(int nuevoRangoFinal) • set_final_time(int f_t) • set_nombreConf(String nuevoNombreConf) • set_unidades_game(String nuevaUnidadGame) • set_time_step(nuevoTimeStep) • set_antiguoNombre(String oldNombre) • set_valores_partida(Vector<Vector<Float>>valPart) • get_grupos() • get_game() • get_game_mostrar() • get_nombres_palancas() • get_grupo_palanca() • get_unidades() • get_modelo() • get_descripcion() • get_rango_inicial() • get_rango_final() • get_final_time() • getNombreConf() • get_unidades_game() • get_time_step() • get_antiguoNombre() • get_valores_partida()
-------------	--

Tabla 92. Descripción de la clase CuadromandoE

Clase	CuadromandoS
Responsabilidad	Clase encargada de gestionar la información relacionada con la configuración de salida del cuadro de mando.
Atributos	<ul style="list-style-type: none"> • grupos • salida • salida_mostrar • nombres • grupo_salidas • representacion • resumen • mixtas • unidades • unidades_game • valores_graficas • maximos • minimos • rango_colores
Operaciones	<ul style="list-style-type: none"> • set_grupo(String nuevoGrupo) • set_salida(String nuevaSalida) • set_salida_mostrar(String nuevaSalidaMostrar) • set_nombre(String nuevoNombre) • set_direccion(String nuevaDireccion) • set_grupo_salidas(String nuevoGrupoSalida) • set_representacion(String nuevaRepresentacion) • set_resumen(String nuevoResumen) • set_mixtas(String nuevoMixtas) • set_unidades(String nuevaUnidad) • set_unidades_game(String nuevaUnidad) • set_valores_grafica(Vector<Vector<Float>> valGraf) • set_rango_colores(Vector<Float> nuevoValor) • get_grupos() • get_salida() • get_salida_mostrar() • get_nombres() • get_direccion() • get_grupos_salidas() • get_representacion() • get_resumen() • get_mixtas() • get_unidades() • get_unidades_game() • get_valores_graficas() • get_maximos() • get_minimos() • get_rango_colores()

Tabla 93. Descripción de la clase CuadromandoS

Clase	Modelo
Responsabilidad	Clase encargada de gestionar la información relacionada con el modelo.
Atributos	<ul style="list-style-type: none"> • nombre
Operaciones	<ul style="list-style-type: none"> • set_nombre(String newNombre) • get_nombre()

Tabla 94. Descripción de la clase Modelo

Clase	PartidaE
Responsabilidad	Clase encargada de gestionar la información de entrada de la partida.
Atributos	<ul style="list-style-type: none"> • usuario • nombre • cuadromando • variables • valores • carpeta • archivo • modelo
Operaciones	<ul style="list-style-type: none"> • set_usuario(String u) • set_nombre(String n) • set_cuadromando(String c) • set_variables(Vector<String> var) • set_valores(Vector <Vector<Float>> val) • set_carpeta(String carp) • set_archivo(String arch) • set_modelo(String m) • get_usuario() • get_nombre() • get_cuadromando() • get_variables() • get_valores() • get_carpeta() • get_archivo() • get_modelo()

Tabla 95. Descripción de la clase PartidaE

Clase	PartidaS
Responsabilidad	Clase encargada de gestionar la información de salida de la partida.
Atributos	<ul style="list-style-type: none">• variables• valores
Operaciones	<ul style="list-style-type: none">• set_variables(Vector<String> var)• set_valores(Vector <Vector<Float>> val)• get_variables()• get_valores()

Tabla 96. Descripción de la clase PartidaS

3.3.6. Diagramas de secuencia

A continuación vamos a ver los diagramas de secuencia de cada uno de los casos de uso utilizados en el proyecto.

En este los diagramas de secuencia se muestran las interacciones expresadas en función del tiempo. En concreto muestra los objetos participantes y los mensajes que intercambian entre ellos a lo largo del tiempo. Sin embargo, no muestra los enlaces entre los objetos.

Los diagramas de secuencias son más apropiados para especificar restricciones de interacción en tiempo real.

Solicitar registro

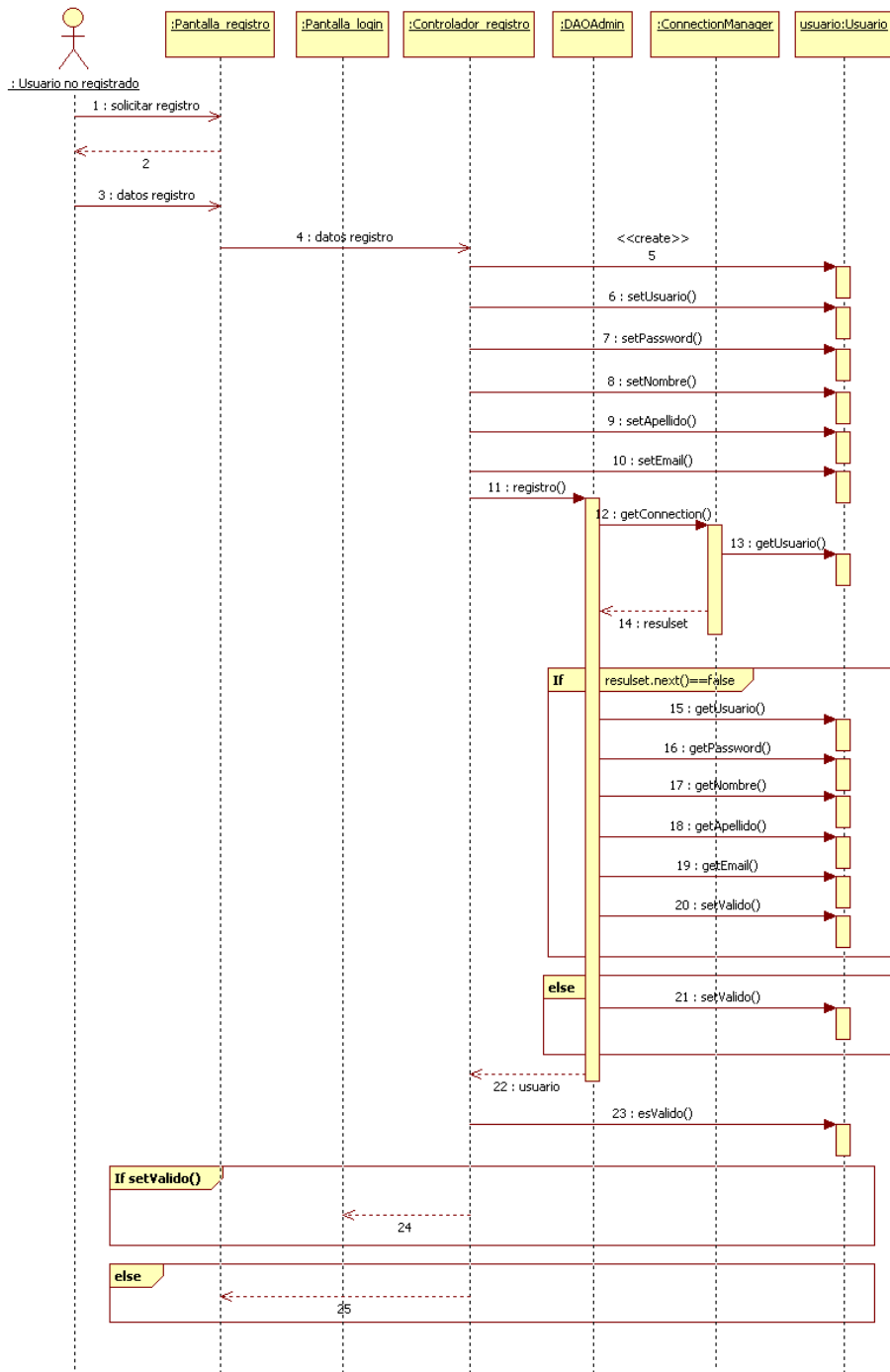


Figura 15. Diagrama de secuencia - Solicitar registro

Identificarse

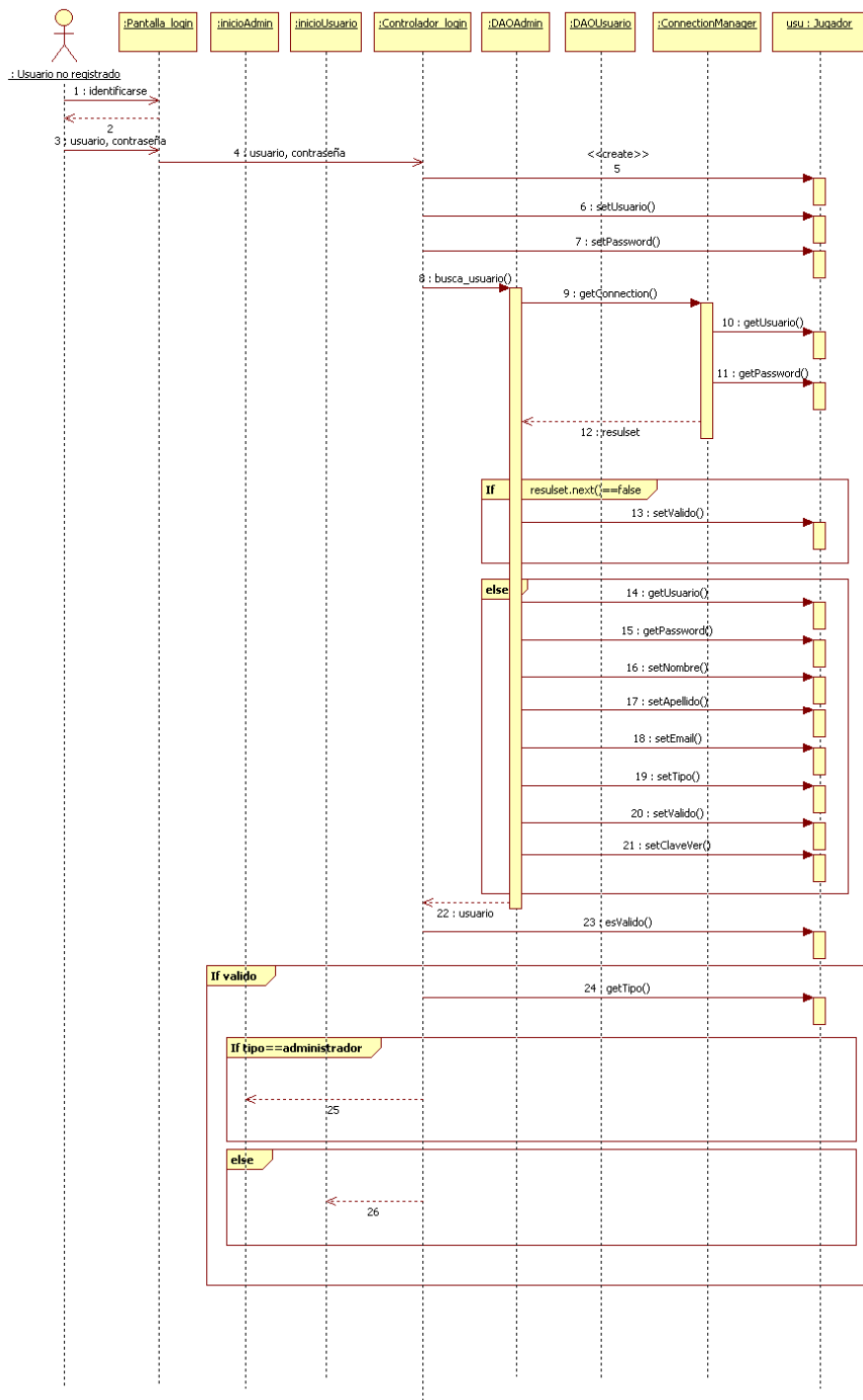


Figura 16. Diagrama de secuencia - Identificarse

Crear sala de operaciones

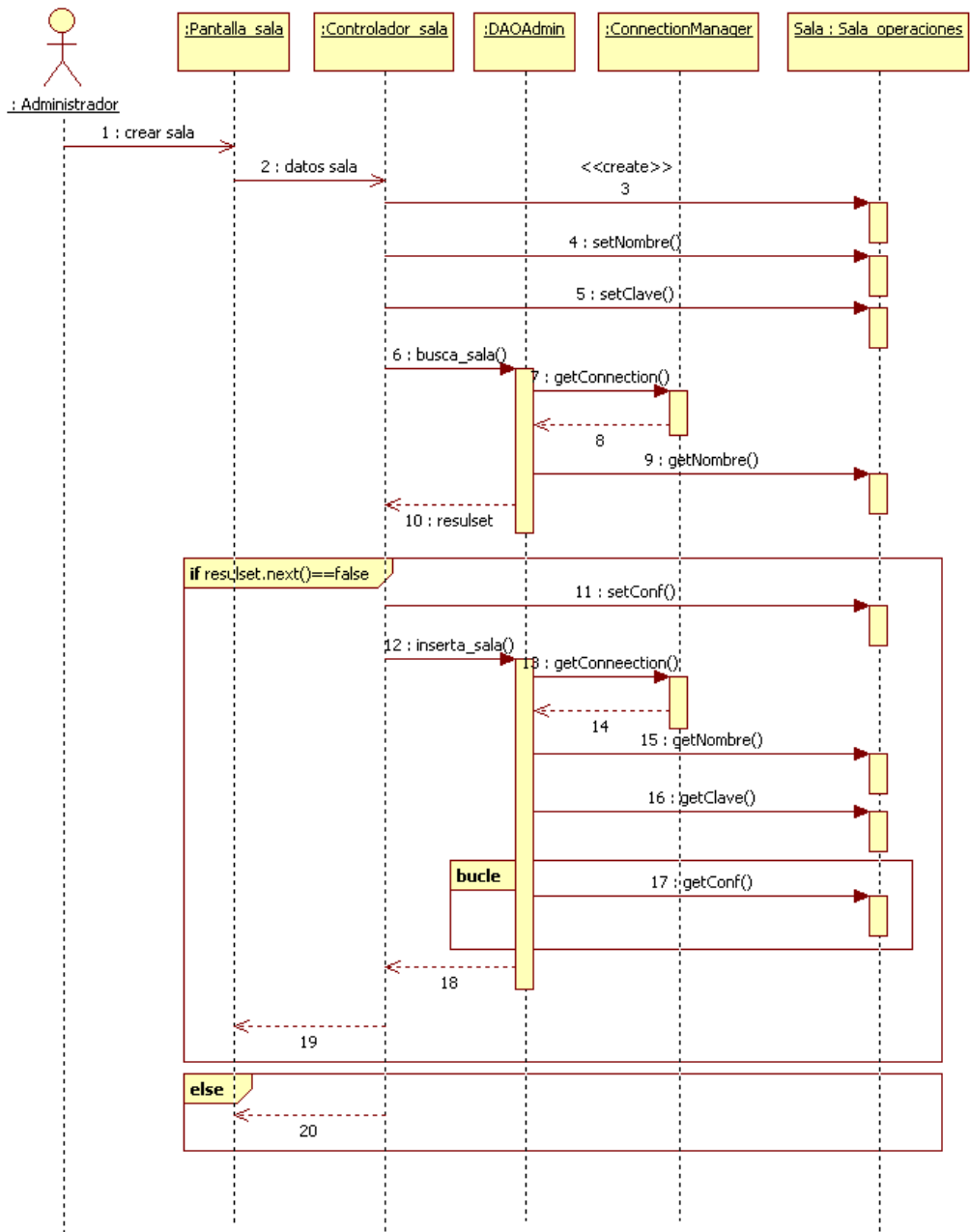


Figura 17. Diagrama de secuencia - Crear sala de operaciones

Modificar sala de operaciones

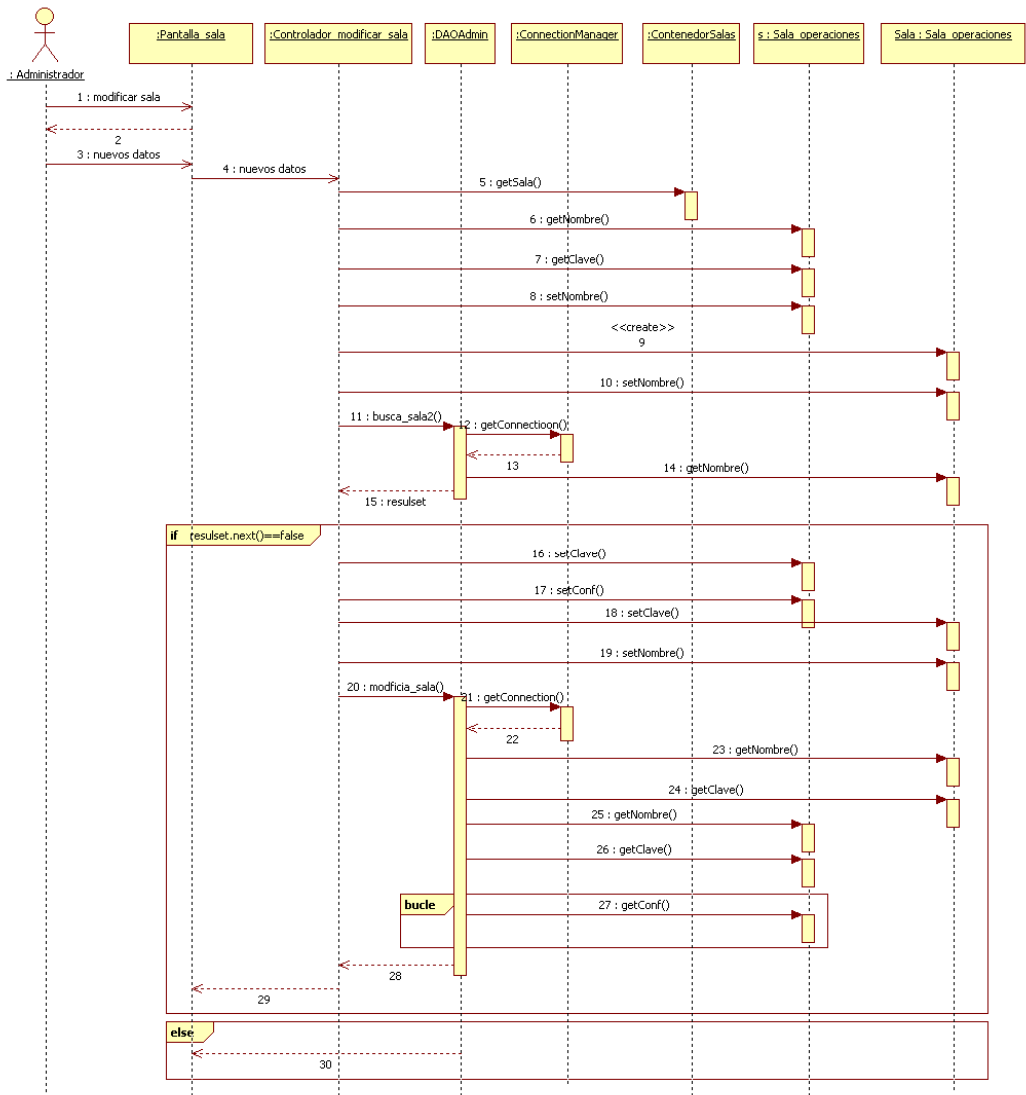


Figura 18. Diagrama de secuencia - Modificar sala de operaciones

Eliminar sala de operaciones

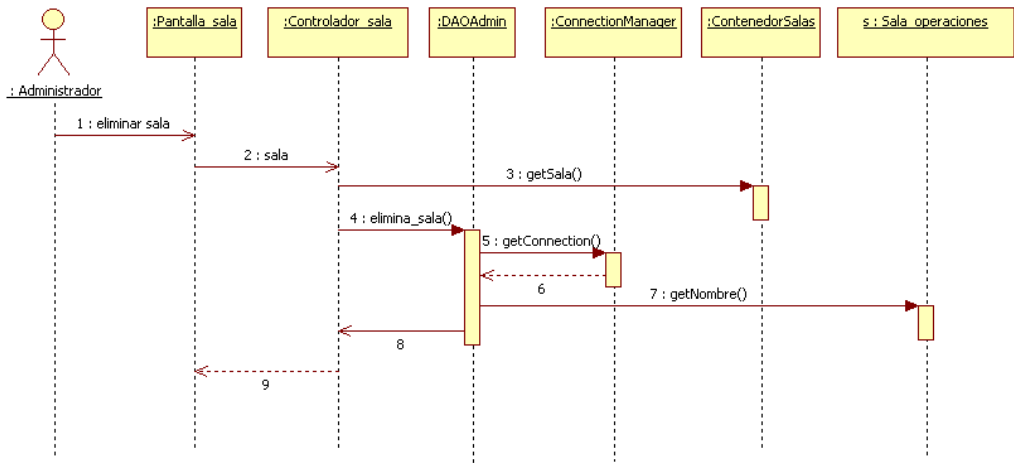


Figura 19. Diagrama de secuencia - Eliminar sala de operaciones

Modificar Usuario

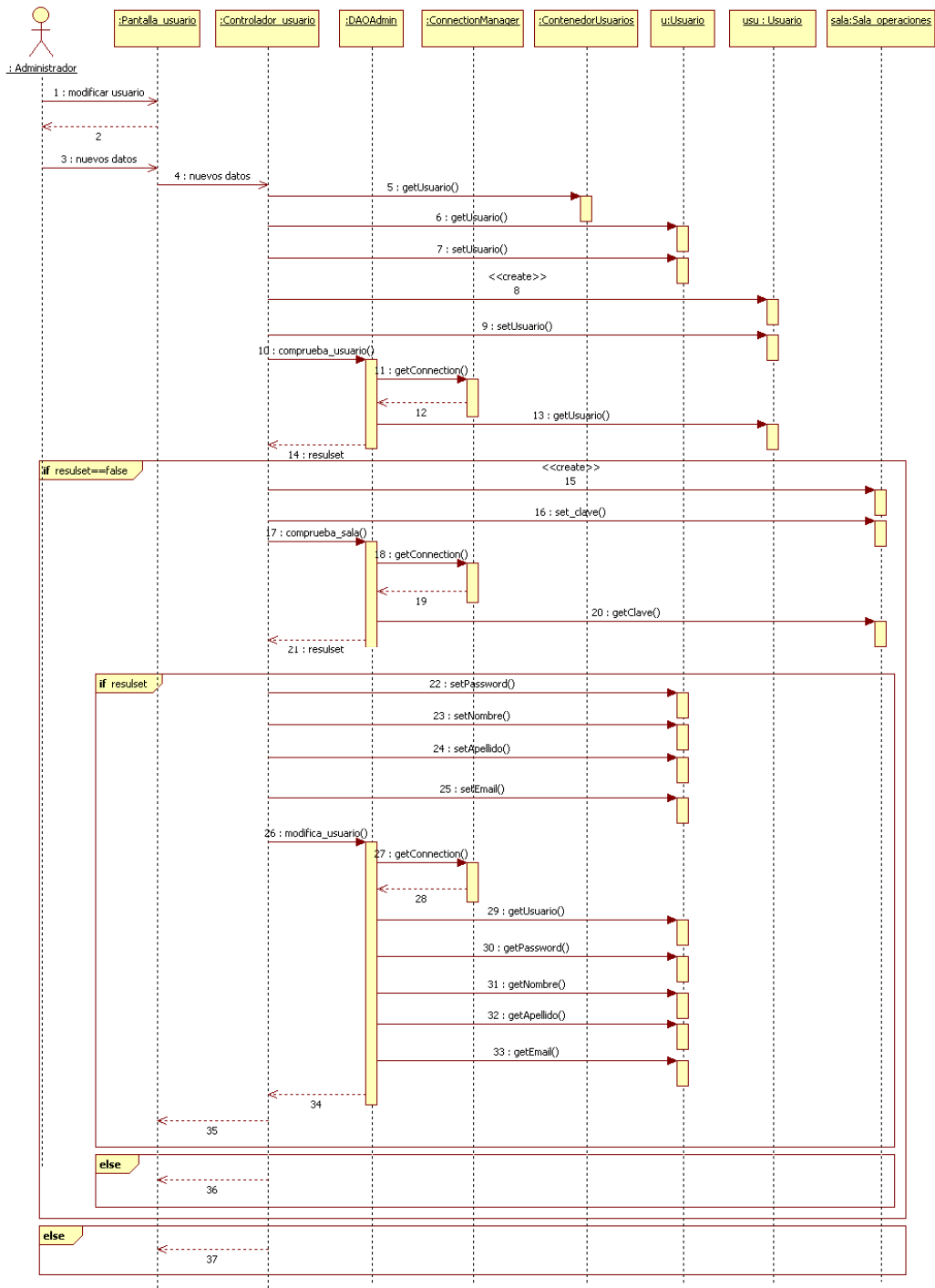


Figura 20. Diagrama de secuencia - Modificar usuario

Eliminar Usuario

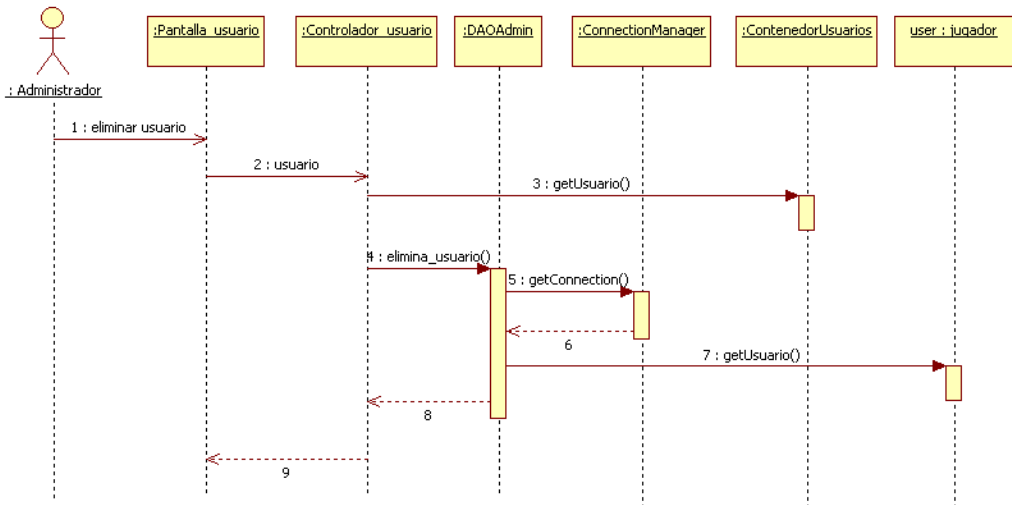


Figura 21. Diagrama de secuencia - Eliminar usuario

Cargar Modelo

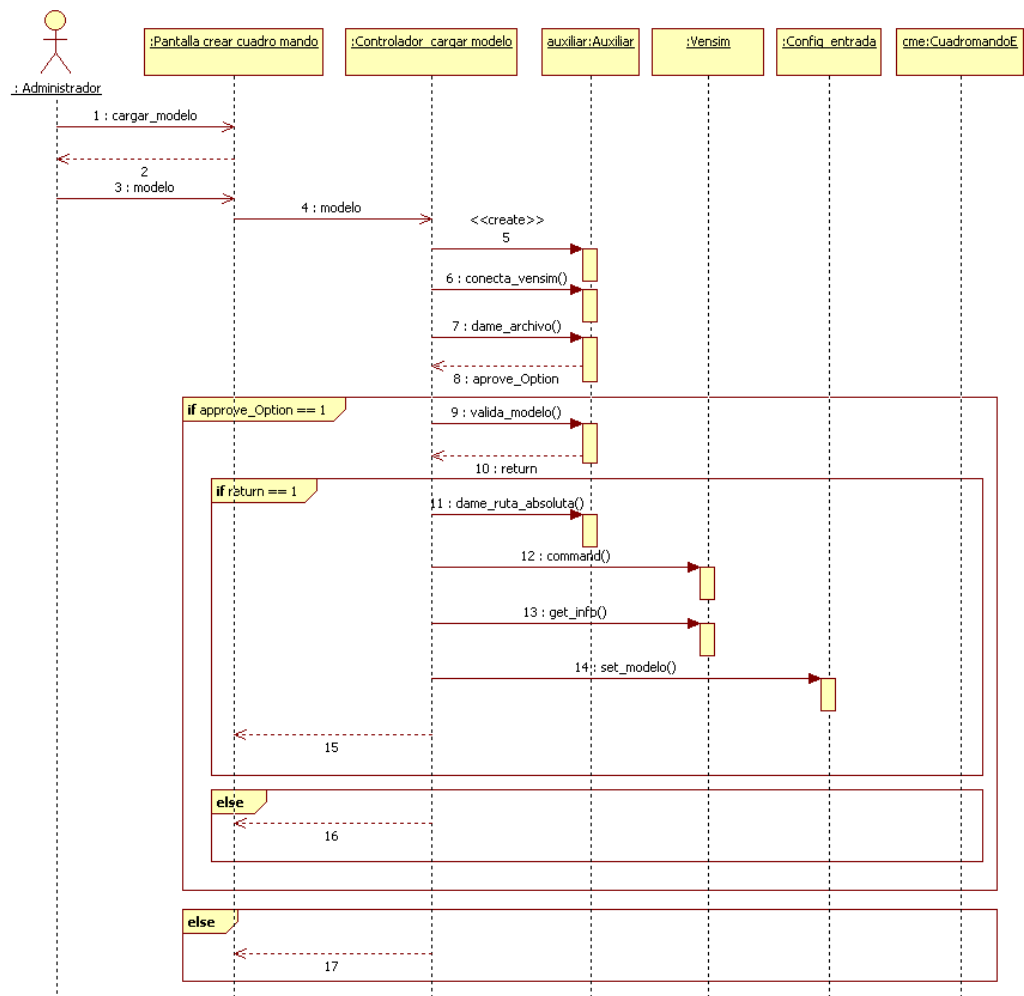
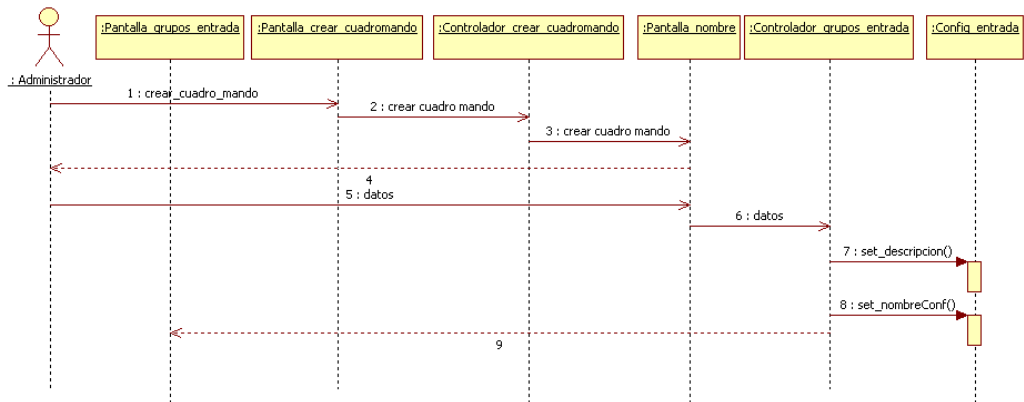


Figura 22. Diagrama de secuencia - Cargar Modelo

Crear Cuadro de Mando – Datos Cuadro de Mando**Figura 23.** Diagrama de secuencia - Crear CM - Datos Cuadro de Mando

Crear Cuadro de Mando – Crear grupo de entrada



Figura 24. Diagrama de secuencia - Crear CM - Crear grupo de entrada

Crear Cuadro de Mando – Eliminar grupo de entrada

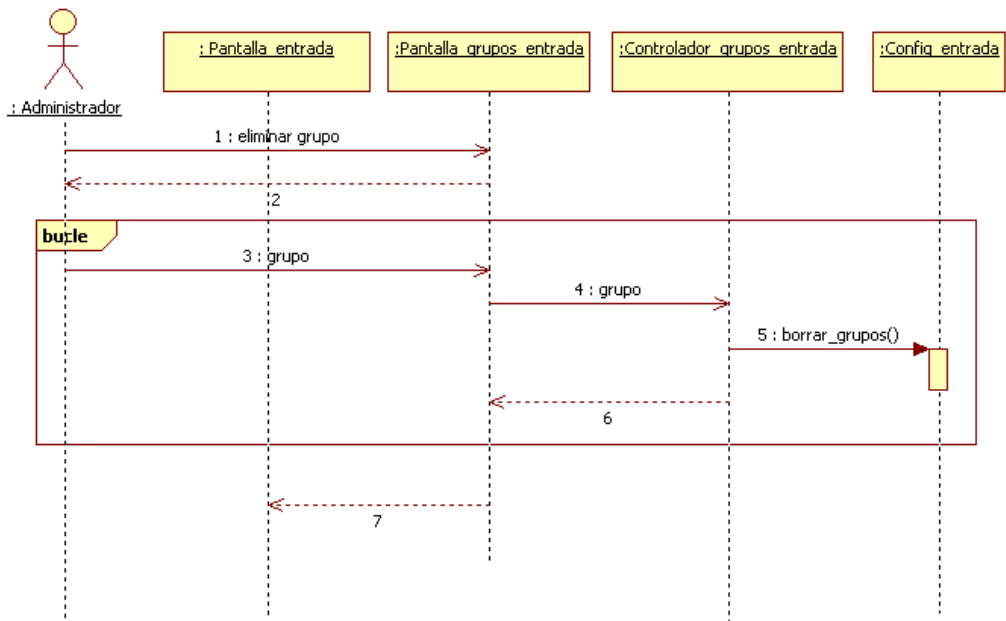


Figura 25. Diagrama de secuencia - Crear CM - Eliminar grupo de entrada

Crear Cuadro de Mando – Configuración de variables juego

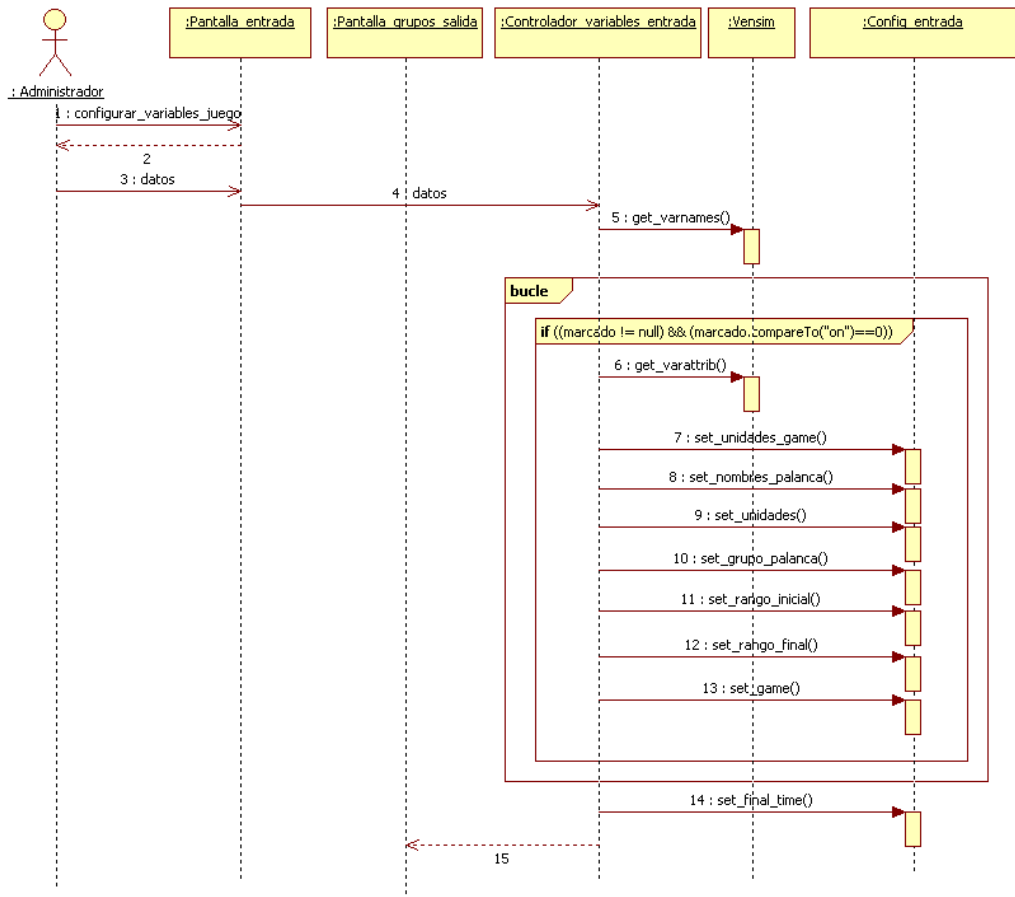
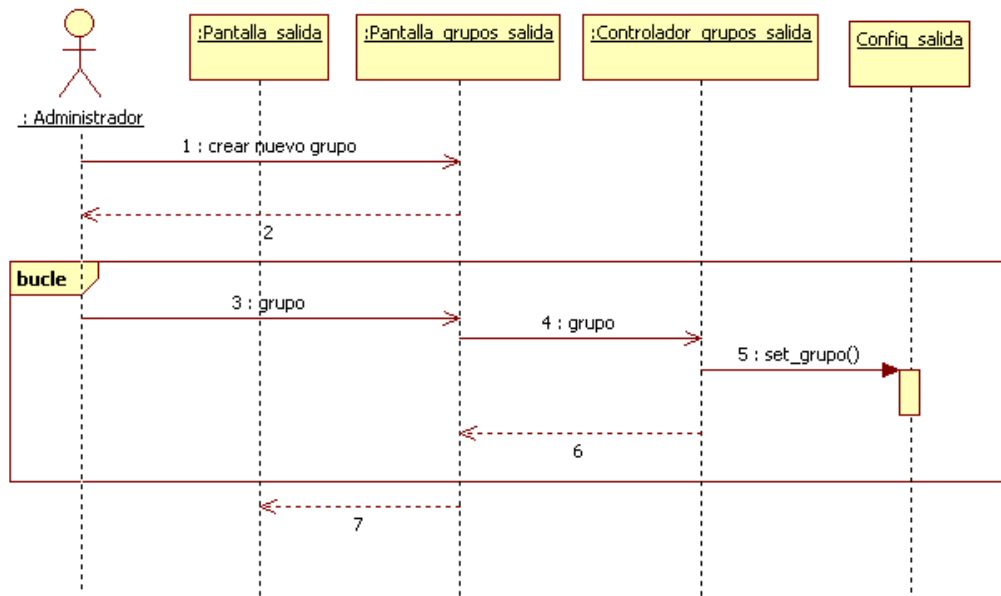
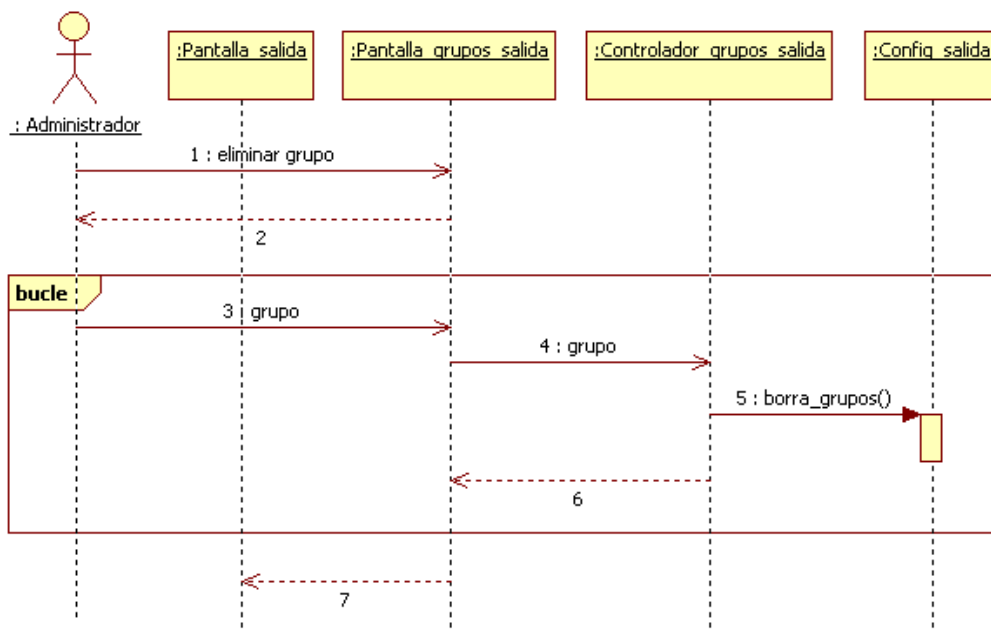


Figura 26. Diagrama de secuencia - Crear CM - Configuración de variables juego

Crear Cuadro de Mando – Crear grupo de salida**Figura 27.** Diagrama de secuencia - Crear CM - Crear grupo de salida**Crear Cuadro de Mando – Eliminar grupo de salida****Figura 28.** Diagrama de secuencia - Crear CM - Eliminar grupo de salida



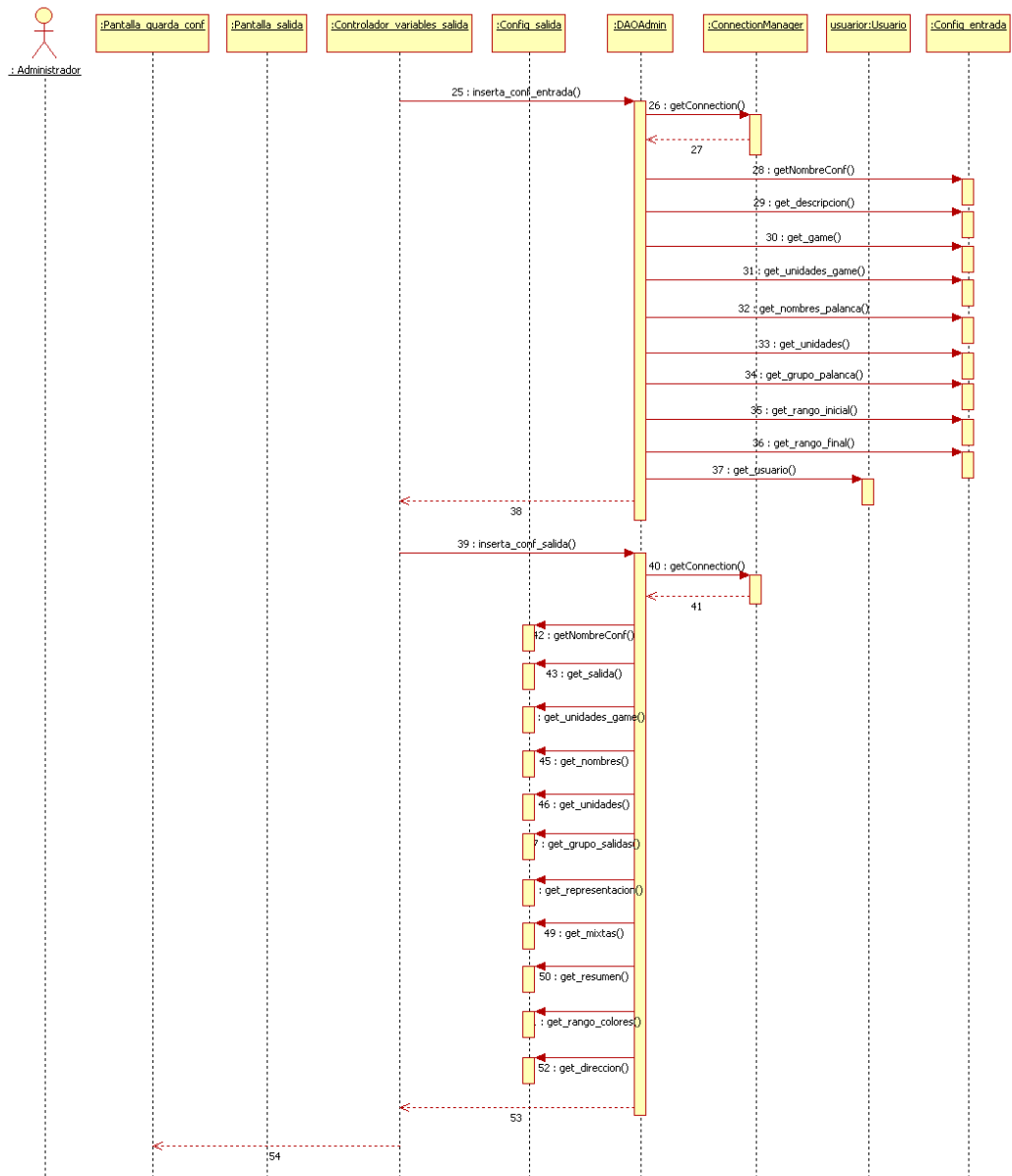


Figura 29. Diagrama de secuencia - Crear CM - Configuración de variables salida

Modificar Cuadro de Mando – Datos de Cuadro de Mando

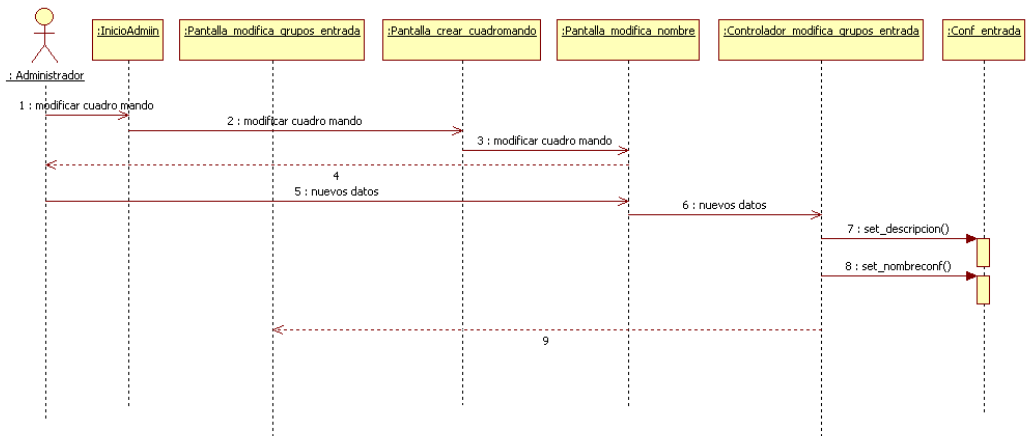


Figura 30. Diagrama de secuencia - Modifica CM - Datos de Cuadro de Mando

Modificar Cuadro de Mando – Crear grupo de entrada

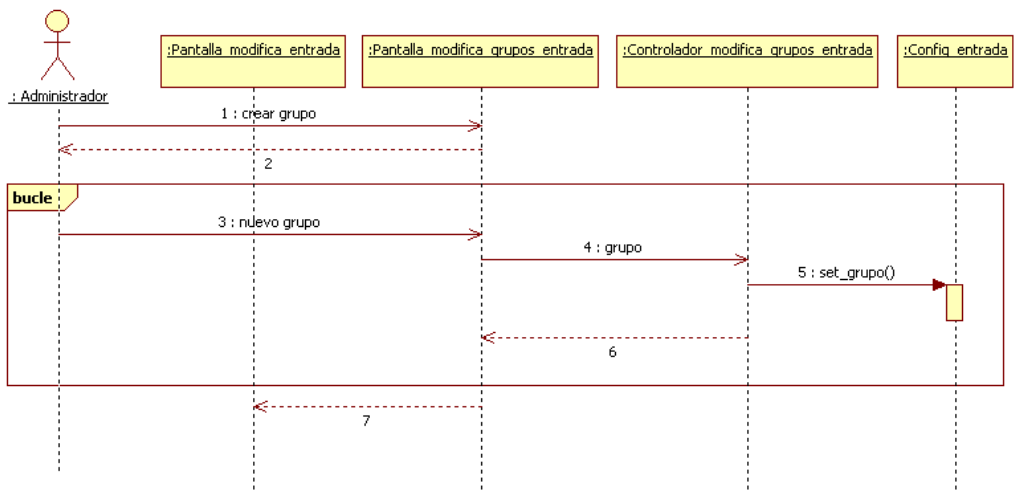


Figura 31. Diagrama de secuencia - Modificar CM - Crear grupo de entrada

Modificar Cuadro de Mando – Eliminar grupo de entrada

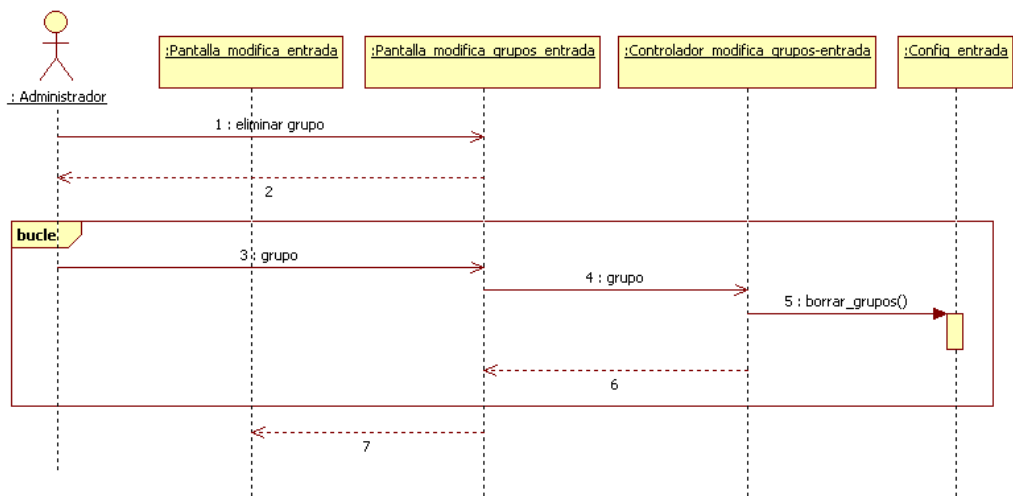


Figura 32. Diagrama de secuencia - Modificar CM - Eliminar grupo de entrada

Modificar Cuadro de Mando – Configuración de variables juego

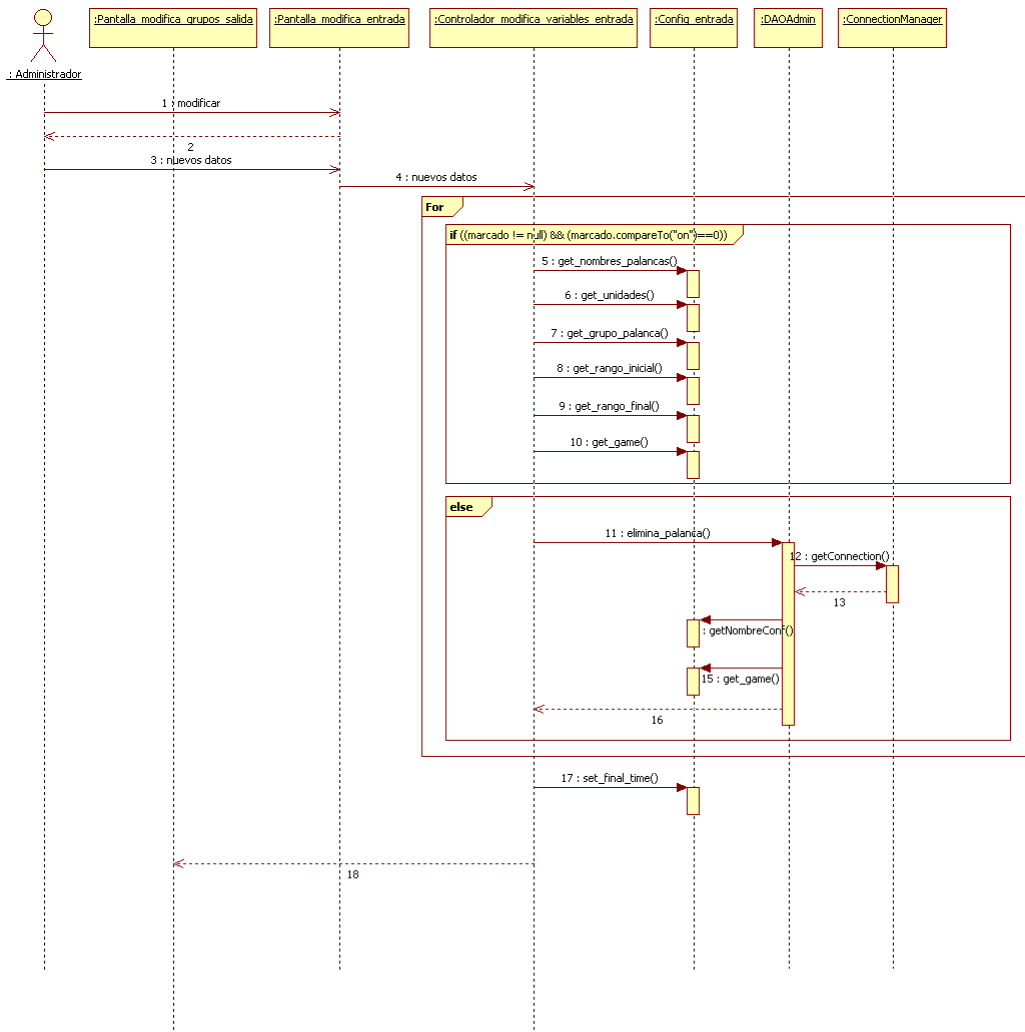


Figura 33. Diagrama de secuencia - Modificar CM - Configuración de variables juego

Modificar Cuadro de Mando – Crear grupo de salida

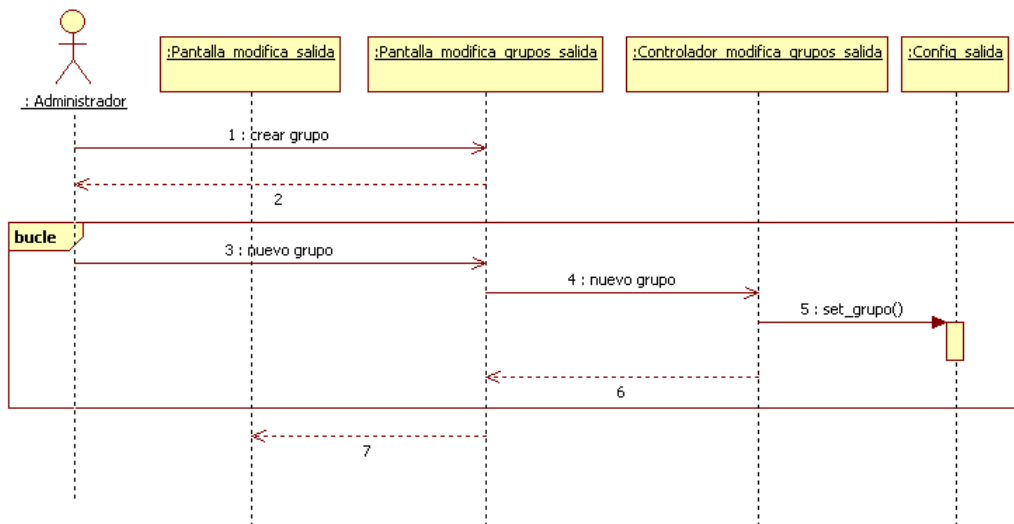


Figura 34. Diagrama de secuencia - Modificar CM - Crear grupo de salida

Modificar Cuadro de Mando – Eliminar grupo de salida

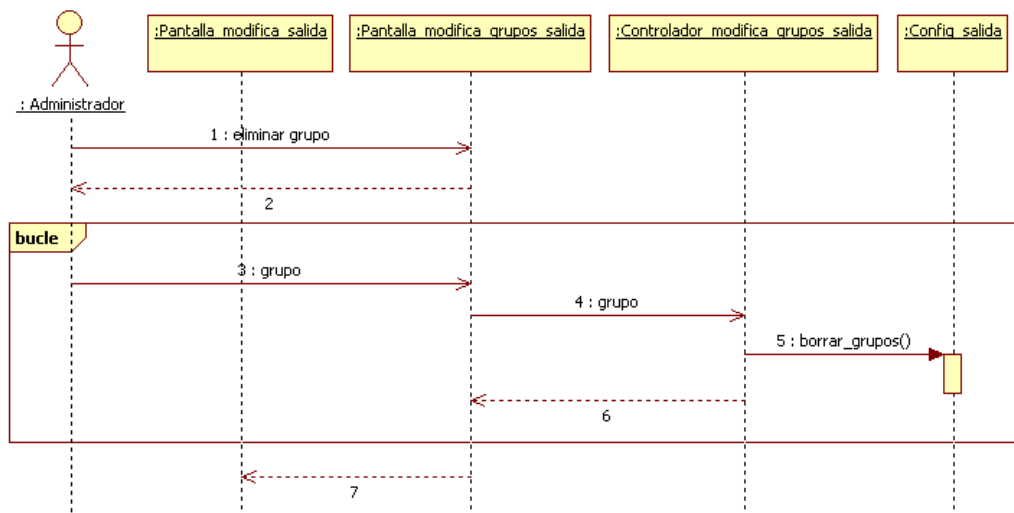
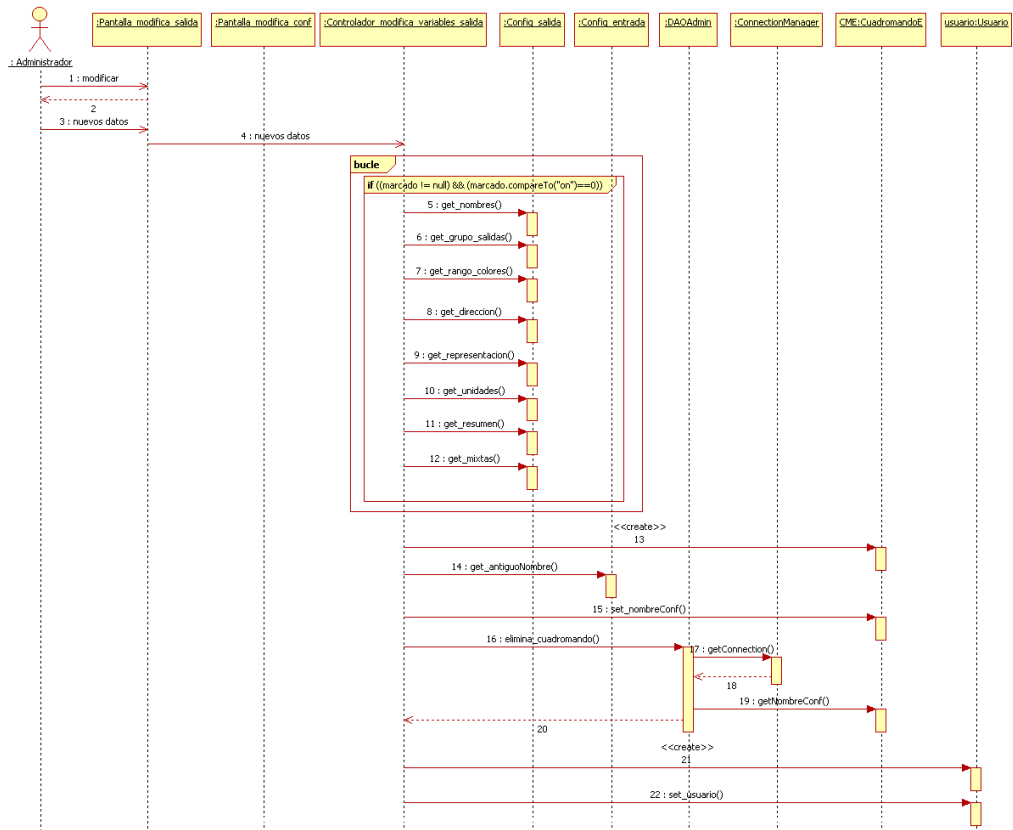


Figura 35. Diagrama de secuencia - Modificar CM - Eliminar grupo de salida

Modificar Cuadro de Mando – Configuración de variables de salida



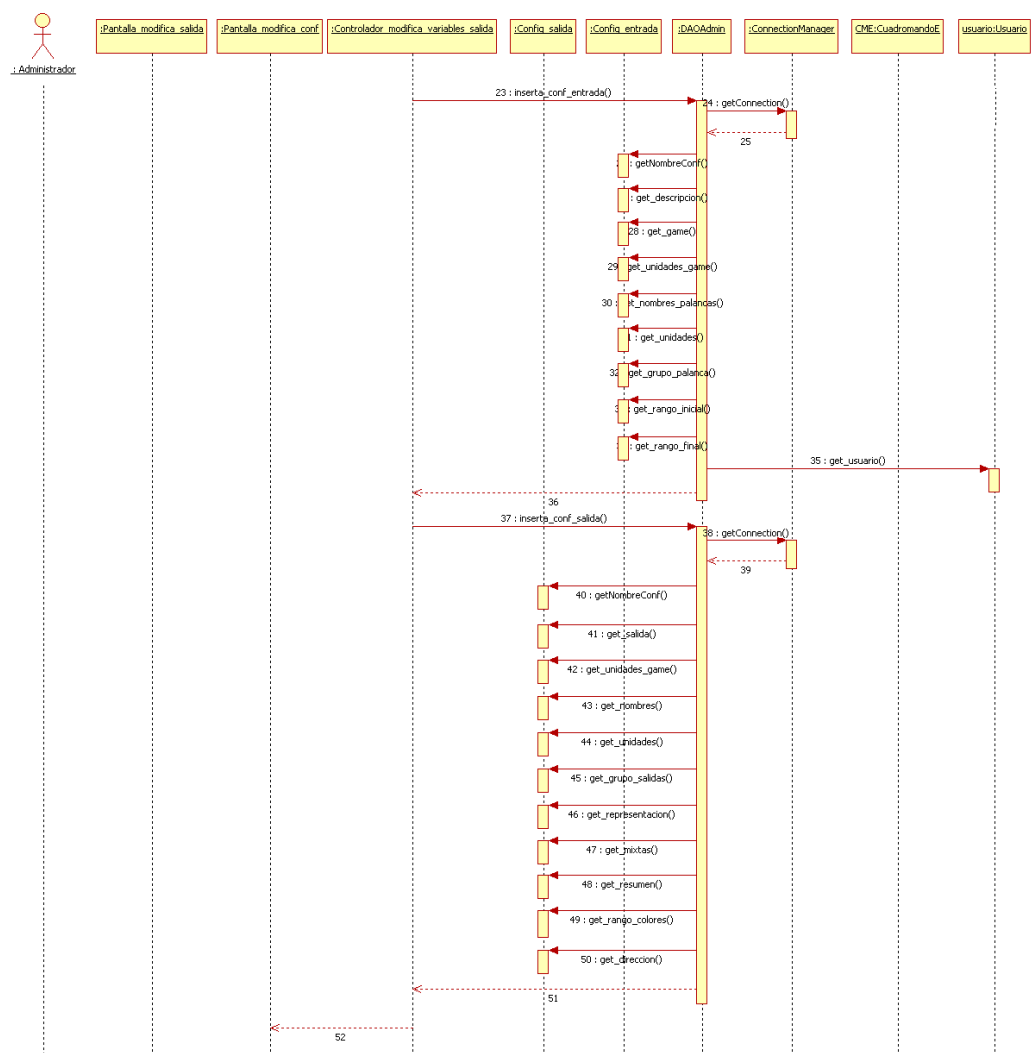


Figura 36. Diagrama de secuencia - Modificar CM - Configuración de variables salida

Eliminar Cuadro de Mando

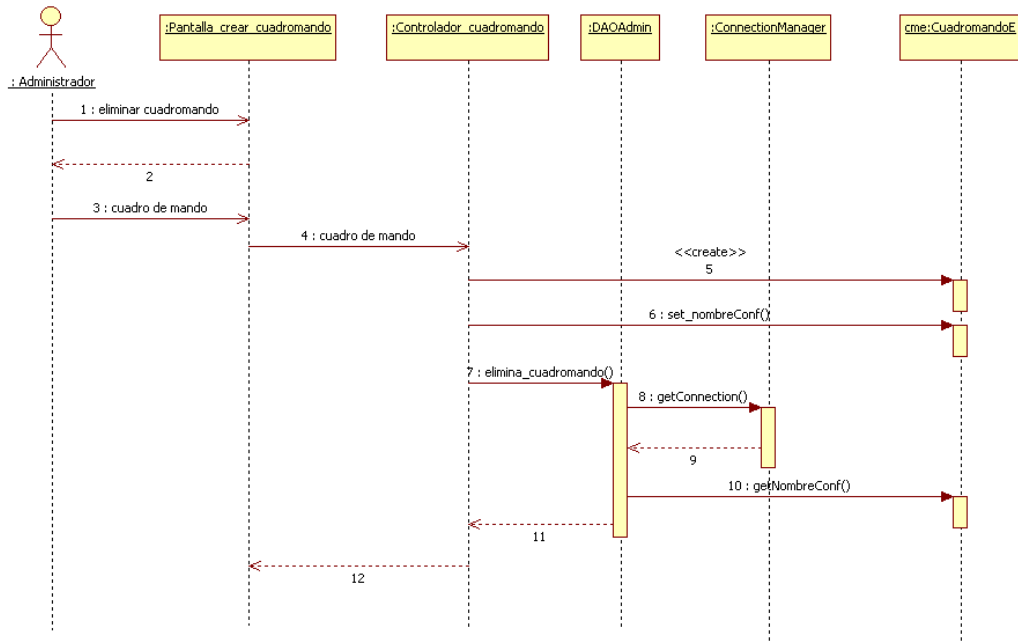


Figura 37. Diagrama de secuencia - Eliminar Cuadro de Mando

Simular

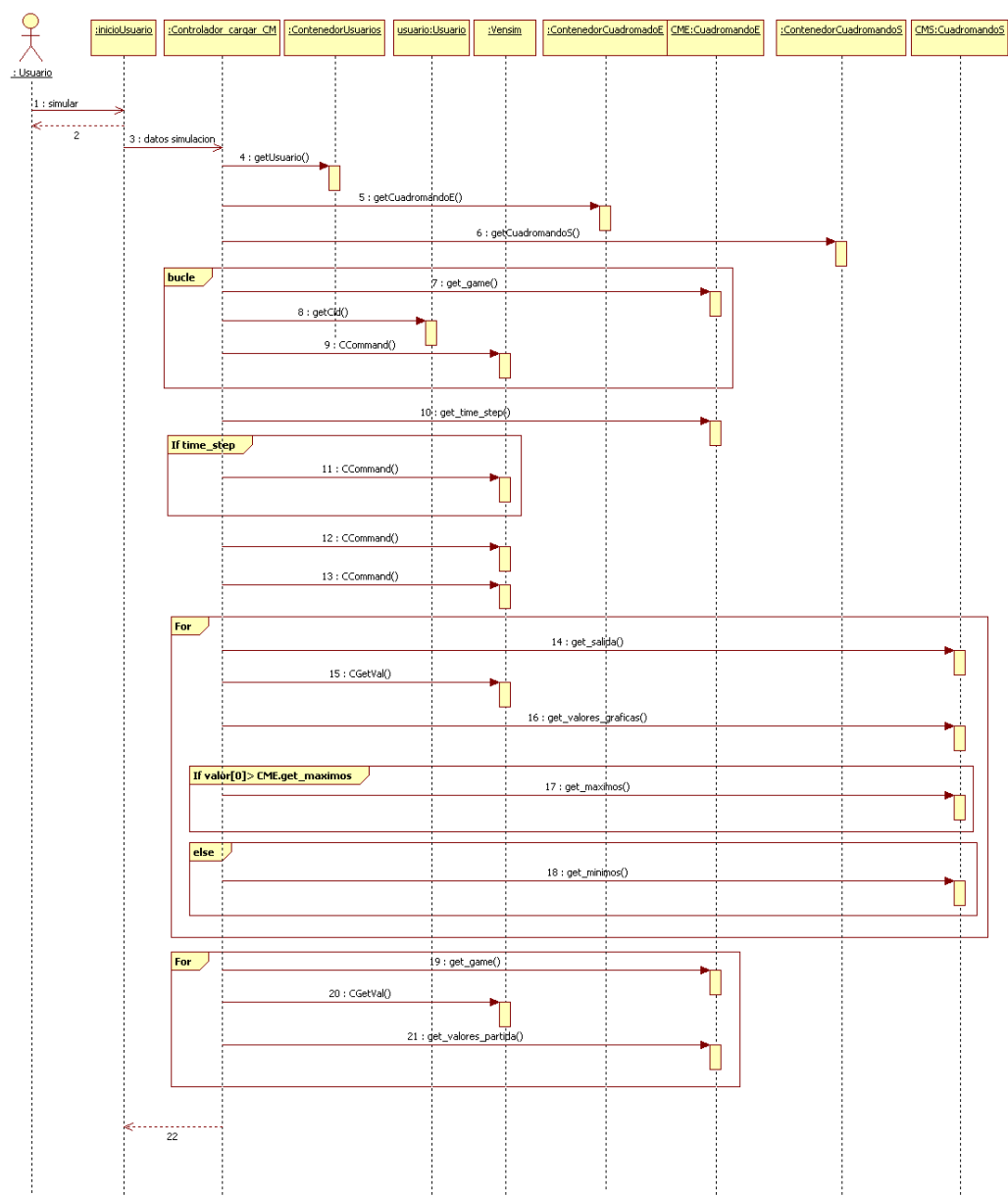


Figura 38. Diagrama de secuencia - Simular

Volver a jugar

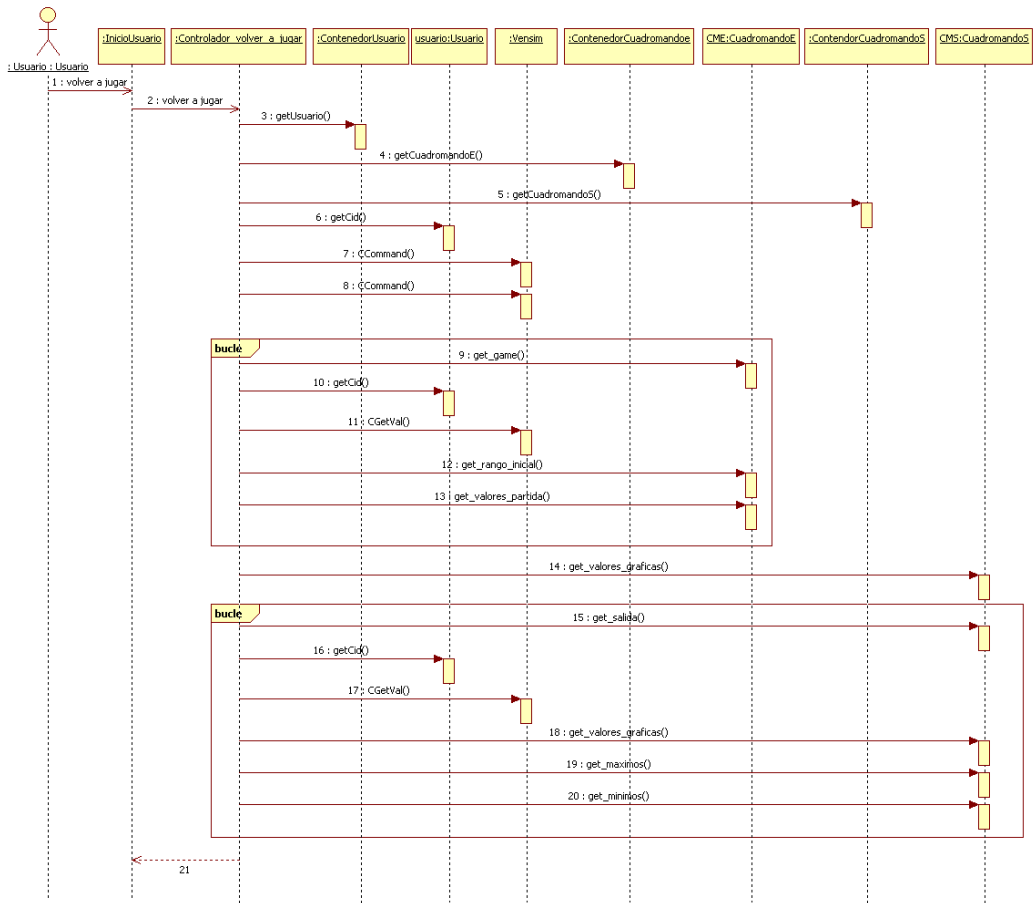
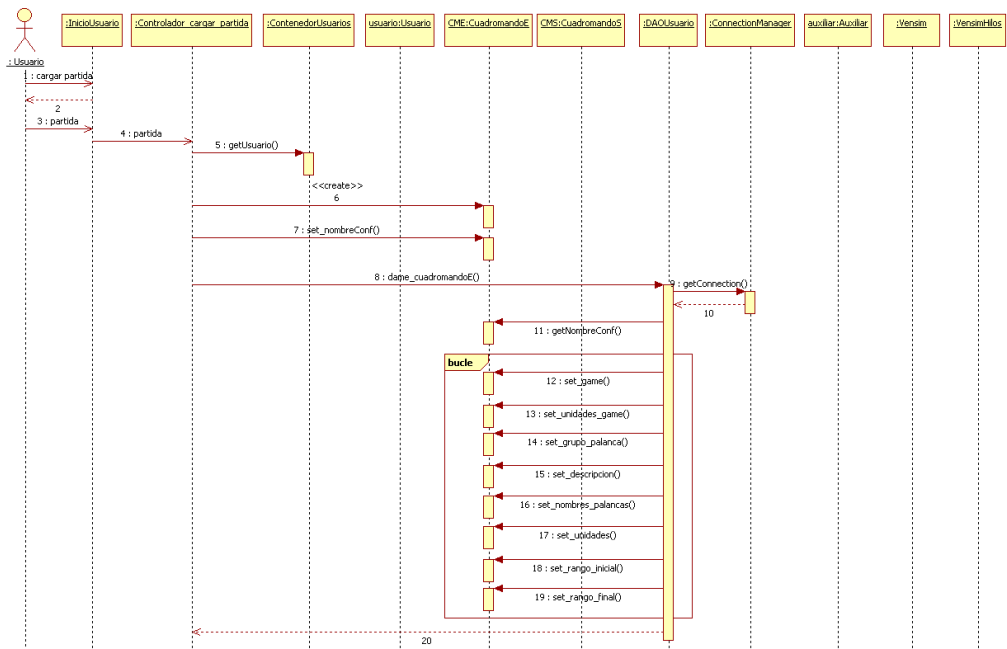
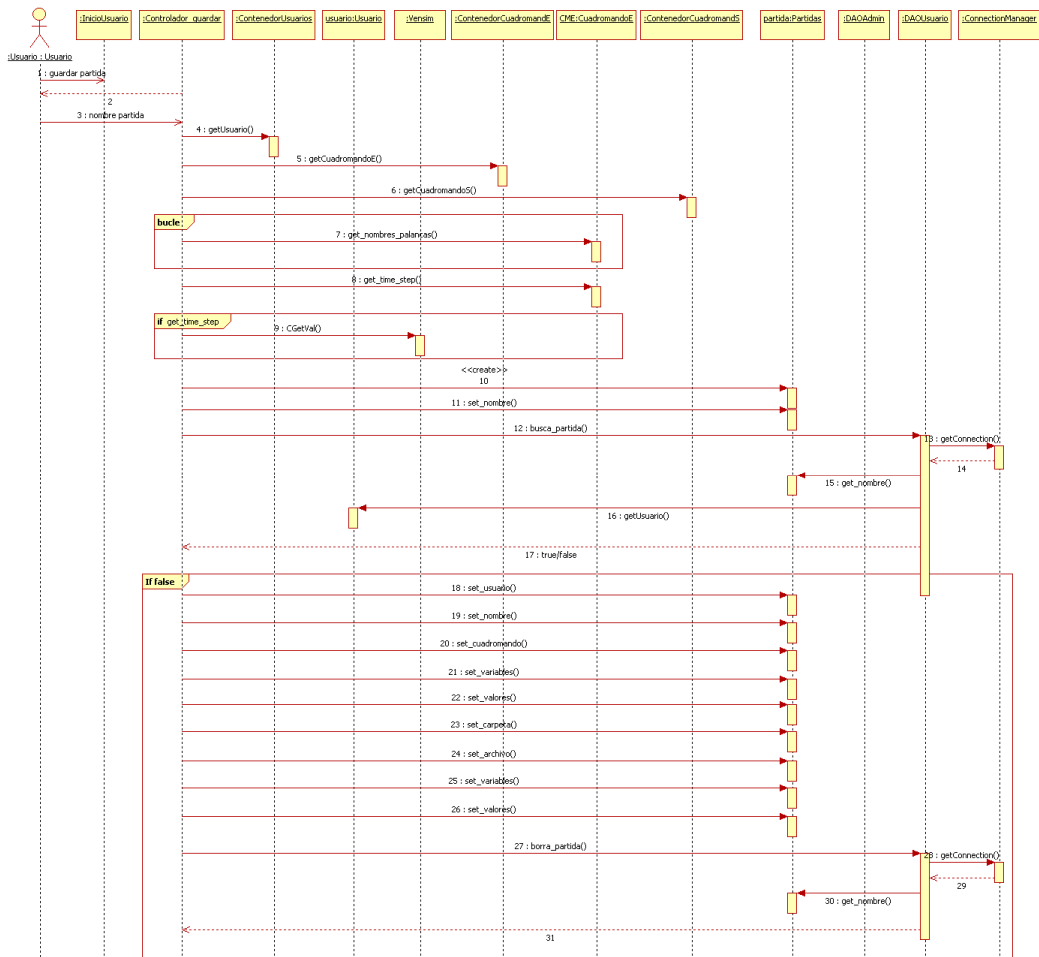


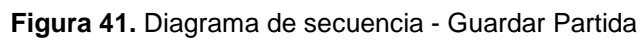
Figura 39. Diagrama de secuencia - Volver a jugar

Cargar Partida



Guardar Partida





Editar Perfil

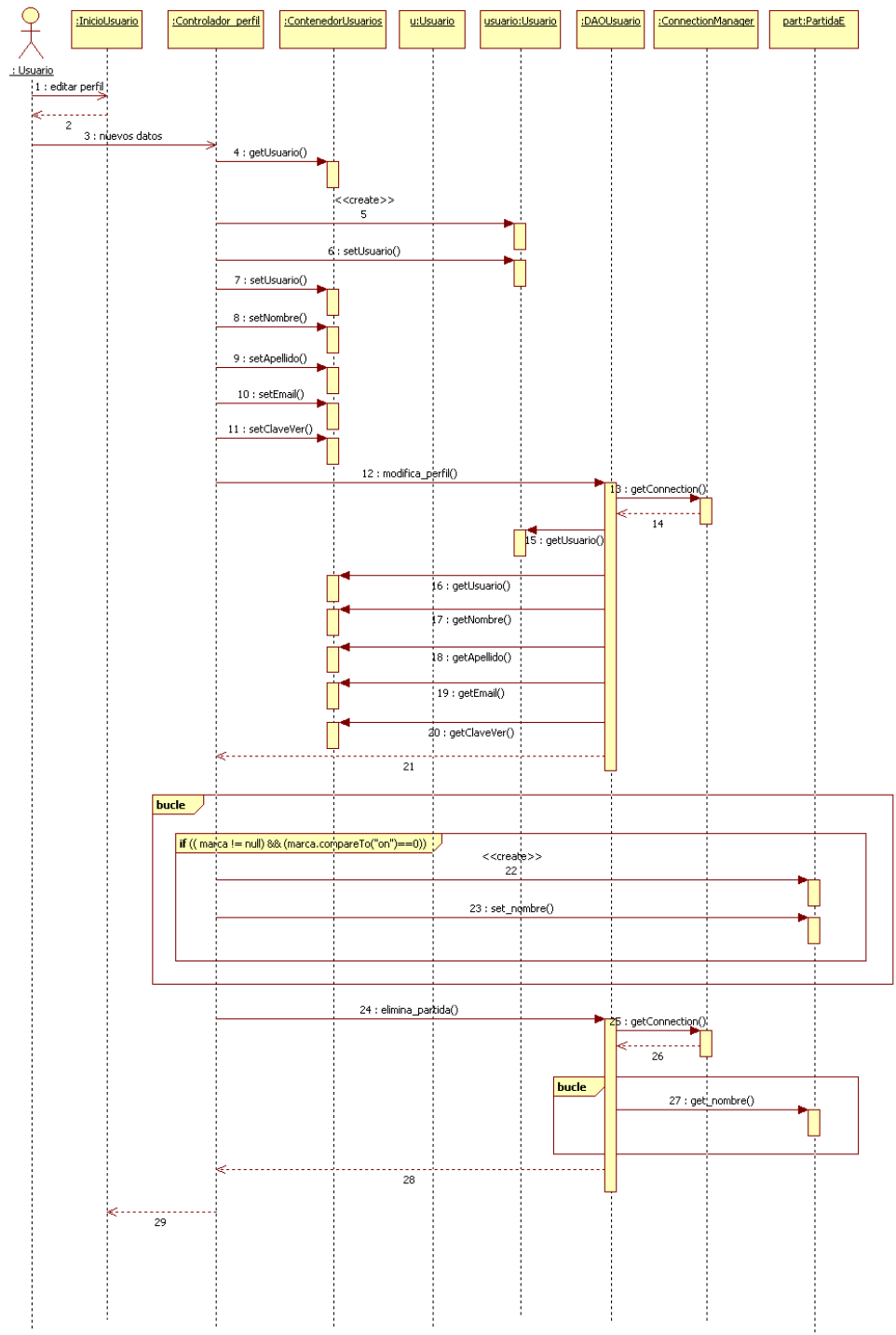


Figura 42. Diagrama de secuencia - Editar perfil

Ver Partida

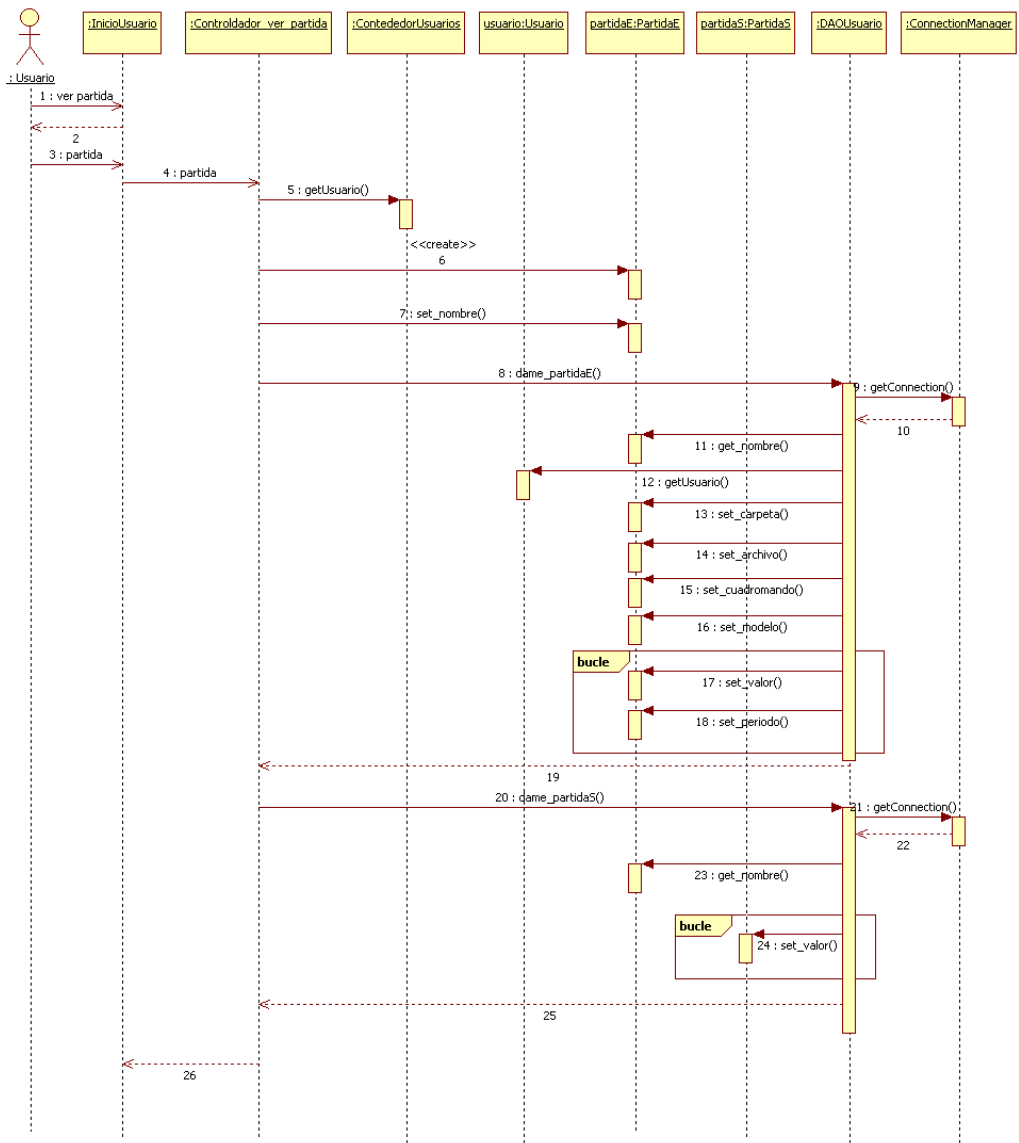


Figura 43. Diagrama de secuencia - Ver Partida

3.3.7. Diseño físico de la base de datos

La información está representada por tablas que se implementarán en la base de datos, por los atributos que formarán las tablas y por las relaciones que se dan entre las distintas tablas por medio de claves foráneas.

Del esquema entidad relación propuesto en el análisis, el diseño de la base de datos es el siguiente:

Nota: Para una mayor claridad del esquema se han tenido únicamente en cuenta las claves primarias y foráneas. El resto de atributos vienen definidos a continuación

Tabla	Confentrada
Descripción	Tabla donde alojamos la información relacionada con la configuración de entrada del cuadro de mando.
Atributos	<ul style="list-style-type: none">• id• nombre• descripcion• game• unidadGame• variable• unidad• grupo• rangolni• rangoFin
Clave Primaria	<ul style="list-style-type: none">• id• nombre

Tabla 97. Tabla Confentrada

Tabla	Confsalida
Descripción	Tabla donde alojamos la información relacionada con la configuración de entrada del cuadro de mando.
Atributos	<ul style="list-style-type: none"> • id • nombre • salida • unidadGame • variable • unidad • grupo • representacion • mixto • resumen • verde • amarillo • rojo • rango1 • rango2 • direccion
Clave Primaria	<ul style="list-style-type: none"> • id • nombre

Tabla 98. Tabla Confsalida

Tabla	Modelo
Descripción	Tabla donde alojamos la información relacionada con modelo.
Atributos	<ul style="list-style-type: none"> • nombre
Clave Primaria	<ul style="list-style-type: none"> • nombre

Tabla 99. Tabla Modelo

Tabla	Partidae
Descripción	Tabla donde alojamos la información relacionada con la partida realizada por el usuario
Atributos	<ul style="list-style-type: none"> • nombre • id • usuario • variable • valor • periodo • carpeta • archivo
Clave Primaria	<ul style="list-style-type: none"> • nombre • id • usuario

Tabla 100. Tabla Partidae

Tabla	Partidas
Descripción	Tabla donde alojamos la información relacionada con la partida realizada por el usuario.
Atributos	<ul style="list-style-type: none"> • nombre • id • usuario • variable • valor • periodo
Clave Primaria	<ul style="list-style-type: none"> • nombre • id • usuario

Tabla 101. Tabla Partidas

Tabla	Rconfmodelo
Descripción	Tabla donde alojamos la información de la relación entre la entidad configuración y la entidad modelo.
Atributos	<ul style="list-style-type: none"> • nombre • cuadromando
Clave Primaria	<ul style="list-style-type: none"> • nombre • cuadromando

Tabla 102. Tabla Rconfmodelo

Tabla	Rconfpartida
Descripción	Tabla donde alojamos la información de la relación entre la configuración y la entidad partida.
Atributos	<ul style="list-style-type: none"> • usuario • nombre • id • cuadromando
Clave Primaria	<ul style="list-style-type: none"> • usuario • nombre • id • cuadromando

Tabla 103. Tabla Rconfpartida

Tabla	Rconfsala
Descripción	Tabla donde alojamos la información de la relación entre la entidad configuración y la entidad sala.
Atributos	<ul style="list-style-type: none"> • nombre • cuadromando
Clave Primaria	<ul style="list-style-type: none"> • nombre • cuadromando

Tabla 104. Tabla Rconfsala

Tabla	Rconfusuario
Descripción	Tabla donde alojamos la información de la relación entre la entidad configuración y la entidad usuario.
Atributos	<ul style="list-style-type: none"> • usuario • cuadromando
Clave Primaria	<ul style="list-style-type: none"> • usuario • cuadromando

Tabla 105. Tabla Rconfusuario

Tabla	Sala
Descripción	Tabla donde alojamos la información relacionada con la entidad sala.
Atributos	<ul style="list-style-type: none"> • nombre • clave
Clave Primaria	<ul style="list-style-type: none"> • nombre • clave

Tabla 106. Tabla Sala

Tabla	Usuario
Descripción	Tabla donde alojamos la información relacionada con la entidad usuario.
Atributos	<ul style="list-style-type: none"> • login • password • nombre • apellido • tipo • email • claveVer
Clave Primaria	<ul style="list-style-type: none"> • login

Tabla 107. Tabla Usuario

3.3.7.1. Aspectos técnicos de la base de datos

Debido a las exigencias del cliente a la hora de la implantación de la base de datos se ha optado por MySQL. Para realizar las tareas de gestión y mantenimiento de la base de datos se ha utilizado la herramienta Toad for MySQL. El principal motivo de la elección de estas dos herramientas es que se tratan de herramientas gratuitas lo cual reduce el coste del proyecto.

3.3.7.2. Tipos de los atributos

Para los atributos de campo texto, se ha elegido el tipo *Varchar* con longitudes en caracteres que oscilan entre los 20 y los 100. Con estas longitudes se cubren las necesidades de almacenamiento según el atributo que sea. Por ejemplo para el tipo de usuario se permite una longitud de caracteres de hasta 20, mientras que para los campos como nombre y apellido se permite hasta 100.

Los campos campo que van a contener valores numéricos como id, rango y periodo se ha optado por el tipo de atributo *Int*. Para los campos cuyo valor va a ser valores numéricos en punto flotante como los rangos de valores se ha optado por el tipo de atributo *Float*.

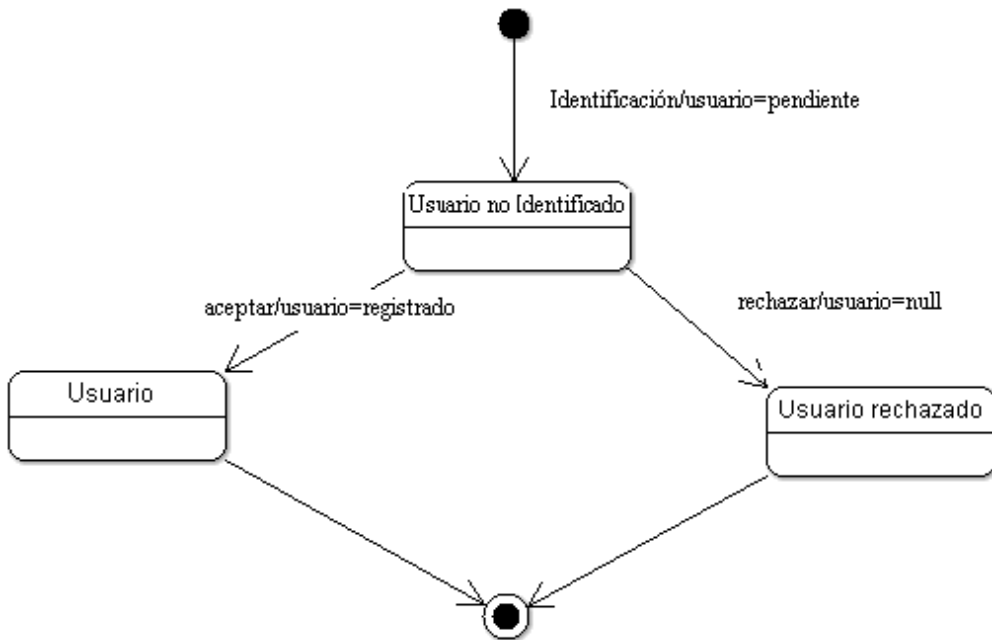
3.3.7.3. Juego de caracteres

Desde que implantamos la base de datos hemos visto que no nos ha dado problemas a la hora de almacenar la información, se ha elegido para los campos de texto la codificación *Latin* en su variante *utf8_swedish_ci*. En esta codificación se pueden añadir caracteres típicos del español como por ejemplo la letra Ñ o las tildes.

3.3.8. Diagramas de estado

Los diagramas de estado muestran el conjunto de estados por los cuales pasa un objeto durante su vida en una aplicación, junto con los cambios que permiten pasar de un estado a otro.

Vamos hacer referencia a una de las clases que puede ir tomando diferentes estados. La clase usuario tiene un comportamiento dinámico significativo. Una vez que el usuario se registra se almacenan sus datos en la base de datos. A partir de ese momento pasa a ser un usuario no identificado en el sistema. Después al iniciar sesión pasa a convertirse en un usuario.

**Figura 44.** Diagrama de estado

3.4. Plan de pruebas

En esta sección se persiguen los siguientes objetivos:

- Identificar la información existente en el proyecto y los componentes software que deben ser testados.
- Listar los requisitos recomendados para probar.
- Recomendar y describir las estrategias de prueba que deben emplearse.
- Identificar los recursos necesarios y proporcionar una estimación de los esfuerzos que pueden requerirse.
- Listar los artefactos entregables del proyecto de pruebas.

Esta sección describe cómo se evaluará el objetivo de cada test. Se identifican las técnicas que se utilizarán y el criterio para saber cuándo ha finalizado cada prueba.

3.4.1. Tipos de pruebas

3.4.1.1. Pruebas de funcionalidad

Objetivos	Asegurar la funcionalidad, incluyendo la navegación, entrada de datos, procesamiento y recuperación
Técnicas	Ejecutar cada caso de uso, flujo del caso de uso o función utilizando datos válidos e inválidos para verificar lo siguiente: <ul style="list-style-type: none">• Se obtienen los resultados esperados cuando se utilizan datos válidos.• Cuando se utilizan datos inválidos se muestran mensajes de error o advertencia adecuados.• Cada regla de negocio es aplicada correctamente.
Criterios de finalización	<ul style="list-style-type: none">• Se han ejecutado todas las pruebas planeadas.• Todos los defectos han sido registrados.
Consideraciones especiales	

Tabla 108. Pruebas de funcionalidad

3.4.1.2. Pruebas de interfaz de usuario

Objetivos	En este tipo de pruebas se realizan pruebas de utilización de la interfaz probando que cada uno de los elementos (botones, cajas de texto, etc.) funciona de forma adecuada
Técnicas	Probar cada una de las pantallas de la aplicación, probar que se permita una navegación adecuada y el correcto estado de los objetos.
Criterios de finalización	Cada una de las pantallas que conforma la aplicación funciona de forma adecuada.
Consideraciones especiales	

Tabla 109. Pruebas de interfaz de usuario

3.4.1.3. Pruebas límite o de integridad de la base de datos

Objetivos	Asegurarse de que se accede a la base de datos, se ejecutan las consultas, se almacenan y recogen los datos, todo ello de manera correcta.
Técnicas	Llamar a los métodos que acceden a la base de datos introduciendo datos tanto válidos como inválidos. A continuación inspeccionar la base de datos comprobando que los datos se han introducido correctamente. Verificar los datos devueltos para ver si son correctos.
Criterios de finalización	Todos los métodos de acceso a la base de datos funcionan según lo planeado sin corrupción de datos.
Consideraciones especiales	

Tabla 110. Pruebas límite o de integridad de la base de datos

3.5. Prototipos de interfaces de usuario

A continuación se pasa a presentar lo que el usuario visualizará al ejecutar la aplicación. Además se puede usar este documento como una presentación para futuros usuarios finales, para que puedan visualizar cómo es el producto final.

Alcance

Los roles encargados de la utilizar el documento de Prototipos de interfaces de usuario son los siguientes:

- **Analistas:** Para visualizar el impacto que tiene la interfaz en el análisis del sistema, en cada uno de los casos de uso que lo componen.
- **Diseñadores:** Para entender el impacto de las interfaces de usuario, ver la usabilidad, experiencia de usuario que se requiere.
- **Probadores:** Para planificar las actividades de prueba.
- **Usuario final:** Como una pre visualización de lo que él usará.

3.5.1. Pantallas principales

General

El *objetivo* de esta pantalla es que tenga una estructura de tal manera que el usuario pueda visualizar todos los datos necesarios sin pensar mucho. Pantallas de fácil acceso y fáciles de usar para cualquier tipo de usuario, poco recargadas, simples, intuitivas, claras y limpias.

Cuadro de decisiones

En esta pantalla se ha buscado una estructura en la cual se diferencie claramente cuál es el área del cuadro de decisiones y cual el área de representación de los valores obtenidos tras realizar las simulaciones. En cuanto a los colores se ha optado por un azul claro para el área del cuadro de decisiones y un color crema para el área de representaciones gráficas.

Los objetos que podemos encontrar en esta pantalla son los siguientes:

Objetos	Descripción	Acciones
Input	Variables palanca	Campos de texto para introducir los valores de entrada
Slider	Variables palanca	Componente desplazable para introducir los valores de entrada
Botones	Simular, simular todo, guardar partida, volver a jugar.	Botones para simular los periodos, guardar partida o volver a jugar.

Tabla 111. Objetos y acciones de la pantalla cuadro de decisiones

Administración

En este tipo de pantallas se ha elegido un estilo simple, de fácil acceso, rápido y claro. Los colores elegidos para este tipo de pantallas han sido un color azul claro de fondo, letras de color blanco y verde.

Los objetos que podemos encontrar en esta pantalla son los siguientes:

Objetos	Descripción	Acciones
Inbox	Cualquier campo que sea de tipo texto como por ejemplo nombre de usuario, apellido.	Introducir cualquier información de tipo texto.
Texto	Título, subtítulo, mensajes.	Visualiza cualquier tipo de título, subtítulo, mensaje de información.
Botones	Anterior Pantalla, Siguiente Pantalla.	Botones para navegar entre las pantallas.
Enlaces	Registro	Enlace para acceder a la pantalla de registro de usuario.
Input	Checkbox, casillas de verificación.	Verifica las variables que van a intervenir en la simulación o bien las que van a aparecer en las representaciones de salida.
Desplegables	Representación Gráfica	Desplegable que muestra las diferentes opciones de representación gráfica.

Tabla 112. Objetos y acciones de la pantalla de Administración

En la parte superior mantenemos la cabecera donde podemos encontrar en la parte izquierda el logotipo de la aplicación el cual a su vez es un enlace que nos llevara al pantalla de inicio, en la parte derecha podemos visualizar el usuario que ha iniciado sesión y la hora actual.

En las pantallas de gestión el *objetivo* buscado es que con la estructura de la pantalla sea vea de un vistazo toda la información que se necesita, poder tener acceso a la lectura o modificación de los datos como cuadros de mando, salas de operaciones o usuarios.

3.5.2. Estándares gráficos

Las pantallas están realizadas de tal manera que pueden ajustarse a diferentes resoluciones de pantalla, se ha optado por un diseño que sea fluido tanto en anchura como en altura, independientemente de la resolución elegida y de la información mostrada.

Se recomiendan resoluciones superiores a 1024x768 para poder manejar la aplicación sin necesidad de utilizar scroll horizontal ni vertical y para poder visualizar toda la información de una sola vez.

Para el diseño de las pantallas se ha utilizado hojas de estilo CSS para que su posterior mantenimiento sea más fácil y cómodo. Se ha probado en diferentes navegadores web para mantener la compatibilidad con todos ellos, a la hora de introducir Java Script hemos procurado que el código introducido sea compatible y funcional para los tres navegadores utilizados.

3.5.3. Jerarquía de navegación

A continuación vamos a ver los diagramas de navegación que ilustran como están conectadas las diversas pantallas entre sí.

3.5.3.1. Diagrama general de navegación

Este diagrama de navegación nos muestra las pantallas a las que se pueden acceder desde la pantalla de inicio.

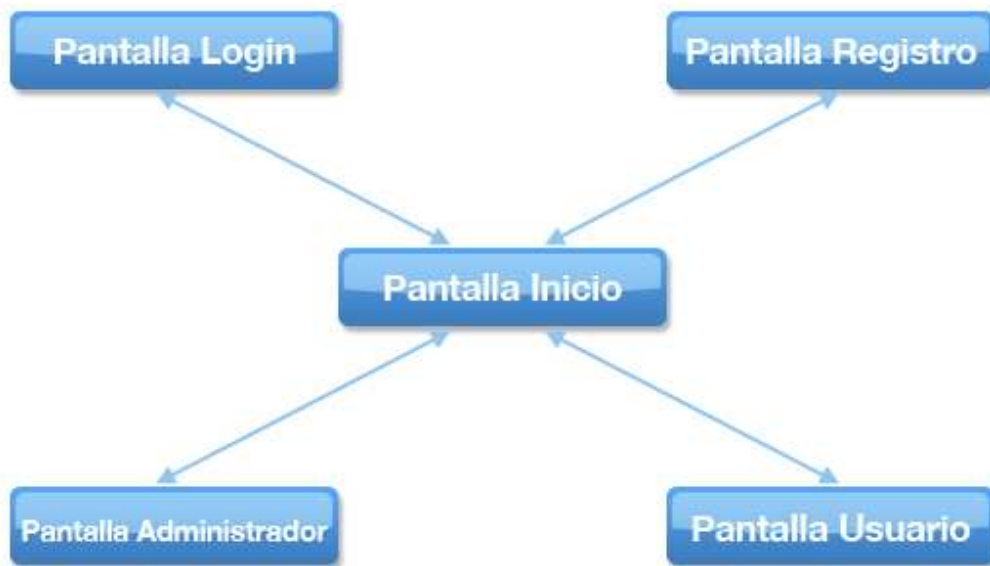


Figura 45. Diagrama general de navegación

3.5.3.2. Diagrama de navegación con las opciones de menú

El diagrama de navegación que vemos a continuación es un diagrama más detallado que el anterior donde podemos ver las diversas opciones de los menús que hay en la aplicación.

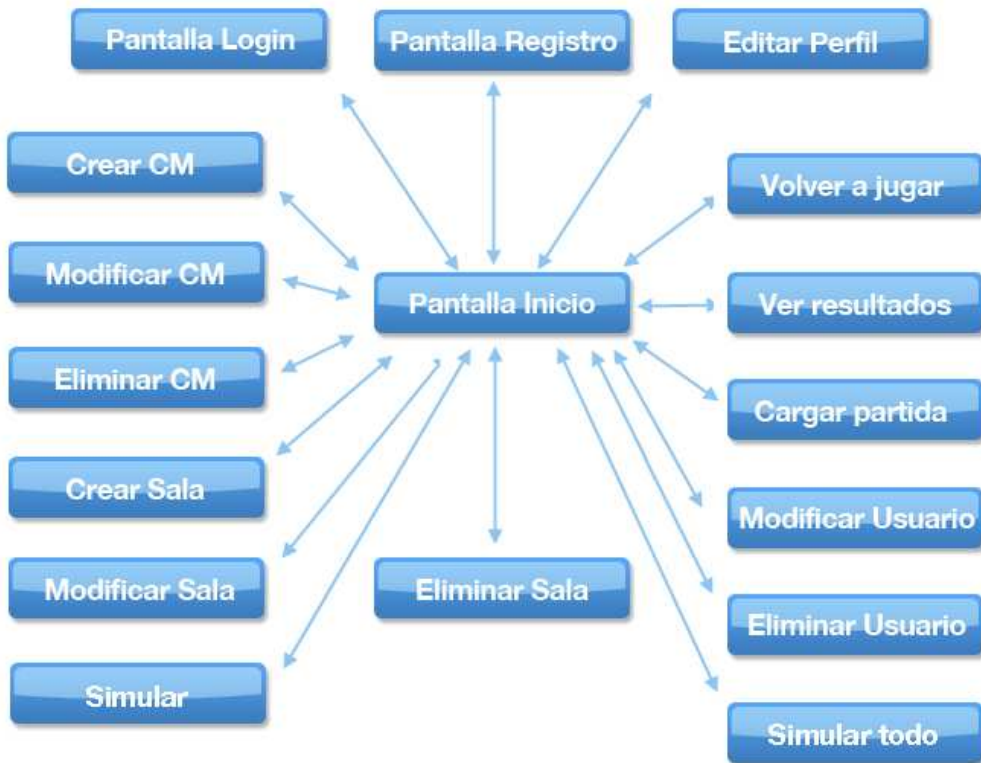


Figura 46. Diagrama de navegación con las opciones de menú

3.5.3.3. Diagrama de navegación administrador

Este diagrama de navegación muestra las pantallas pertenecientes a la gestión de cuadros de mando, como están conectadas entre sí.



Figura 47. Diagrama de navegación administrador

3.5.3.4. Diagrama de navegación usuario

Este diagrama de navegación muestra las pantallas a las que puede acceder el usuario, y como están conectadas entre sí.

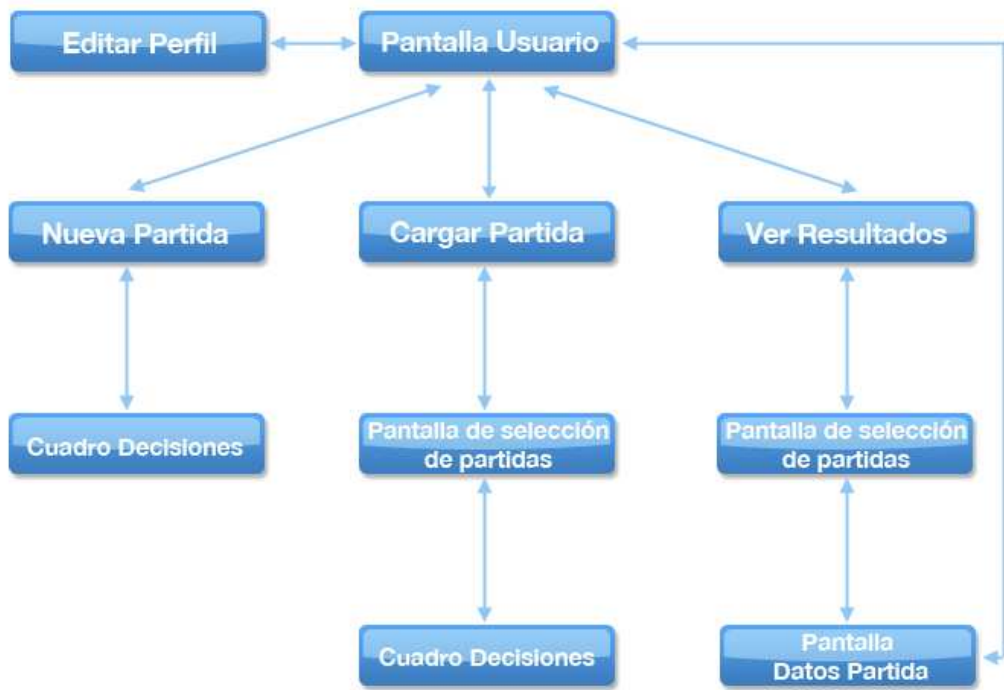


Figura 48. Diagrama de navegación usuario

3.6. Seguimiento de la Fase de Elaboración

Como vimos anteriormente en al final del Capítulo 2 de esta misma documentación. Se han reflejado los hitos finales de la fase de elaboración, han surgido modificaciones por tanto este documento ha sido modificado para adaptarlo a los cambios detectados en la fase de construcción. Como se mostró anteriormente en la planificación de esta fase se ha tenido en cuenta 2 iteraciones.

En la primera iteración se han obtenido los casos de uso y los escenarios fundamentales para comenzar con el desarrollo de la aplicación. Esta información ha sido presentada y aprobada por el cliente. En la siguiente iteración, se han completado las partes relacionadas con el diseño y la arquitectura.

A lo largo de cada iteración, se han ido especificando los casos de prueba necesarios para verificar cada escenario encontrado, pero se detallarán en el apartado correspondiente en la etapa de construcción.

3.7. Plan de Iteración de la Fase de Construcción

En esta sección se expondrá el plan de iteración que se va a seguir en la fase de construcción o implementación. Se parte de los hitos logrados en la fase de elaboración, documento de análisis y documento de diseño y el objetivo es codificar los distintos escenarios en ellos relatados para cubrir las funcionalidades del sistema. A su vez se irán probando los casos de prueba encontrados para garantizar el buen funcionamiento del desarrollo y poder solucionar los errores encontrados lo antes posible.

3.7.1. Plan de trabajo para la iteración

A continuación podemos ver el diagrama de Gantt donde se muestra el plan de la iteración para la etapa de construcción. Se han planificado 2 iteraciones. La primera para la construcción de la interfaz, pruebas e implementar los escenarios principales y la segunda para implementar los escenarios restantes así como el código.

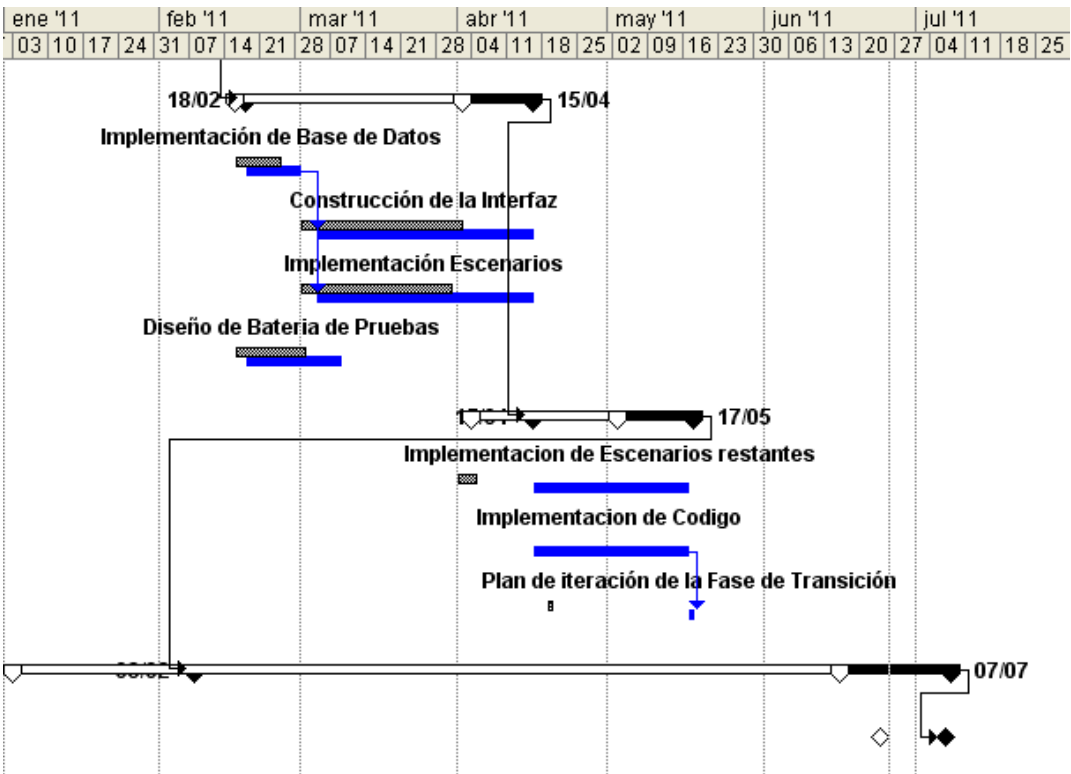


Figura 49. Diagrama de Gantt para la Fase de Construcción

Nombre de tarea	Duración	Comienzo	Fin
 Construccion (Iteración 1)	41 días	vie 18/02/11	vie 15/04/11
Implementación de Base de Datos	7 días	vie 18/02/11	lun 28/02/11
Construcción de la Interfaz	31 días	vie 04/03/11	vie 15/04/11
Implementación Escenarios	31 días	vie 04/03/11	vie 15/04/11
Diseño de Bateria de Pruebas	13 días	vie 18/02/11	mar 08/03/11
 Construccion (Iteración 2)	22 días	vie 15/04/11	mar 17/05/11
Implementacion de Escenarios restantes	21 días	vie 15/04/11	lun 16/05/11
Implementacion deCodigo	21 días	vie 15/04/11	lun 16/05/11
Plan de iteración de la Fase de Transición	1 día	mar 17/05/11	mar 17/05/11

Figura 50. Tareas para la Fase de Construcción

3.7.2. Recursos necesarios

Recursos Humanos

Los recursos humanos de esta iteración serán:

- Los desarrolladores
- El cliente

Recursos Software

Los recursos software para esta iteración serán:

- Servidor Web
 - Apache Tomcat 6
- Lenguaje de Programación
 - JSP
- Servidor de Bases de datos
 - MySQL 5
- Entorno integrado de desarrollo
 - Eclipse
- Herramientas de modelado
 - StarUML
 - REM
- Herramienta de Planificación
 - Microsoft Project 2003
- Herramientas de documentación
 - Microsoft Word 2007
- Herramienta de sincronización
 - Dropbox
- Otros Recursos Software
 - Software de Diseño Grafico: Adobe Photoshop
 - Navegador Web: Cualquier navegador compatibles con los estándares de la W3

Recursos Hardware

Los recursos hardware para esta iteración serán:

- 2 Ordenadores portátiles personales
- Servidor remoto para instalación y pruebas de la aplicación
- Conexión a Internet
- Dispositivos de almacenamiento externo USB para el intercambio de datos con el cliente

Capítulo 4

Fase de Construcción

En este capítulo se pasará a detallar la Fase de Construcción del proyecto

4.1. Concepto de la Fase de Construcción



Figura 51. Hito en la Fase de Construcción

La finalidad principal de esta fase es alcanzar la capacidad operacional del producto de forma incremental a través de las sucesivas iteraciones. Durante esta fase todas los componentes, características y requisitos, que no lo hayan sido hecho hasta ahora, han de ser implementados, integrados y testeados, obteniendo una versión del producto que se pueda poner en manos de los usuarios (una versión beta).

El énfasis en esta fase se pone controlar las operaciones realizadas, administrando los recursos eficientemente, de tal forma que se optimicen los costes, los calendarios y la calidad.

Los objetivos concretos incluyen:

- Minimizar los costes de desarrollo mediante la optimización de recursos y evitando el tener que rehacer un trabajo o incluso desecharlo.
- Conseguir una calidad adecuada tan rápido como sea práctico.
- Conseguir versiones funcionales (alfa, beta, y otras versiones de prueba) tan rápido como sea práctico.

Los productos de la fase de construcción deben ser:

- Modelos Completos (Casos de Uso, Análisis, Diseño, Despliegue e Implementación)
- Arquitectura íntegra (mantenida y mínimamente actualizada)
- Riesgos Presentados Mitigados
- Plan del Proyecto para la fase de Transición
- Manual Inicial de Usuario (con suficiente detalle)
- Prototipo Operacional – beta
- Caso del Negocio Actualizado

4.2. Estructura de los directorios de la aplicación

4.2.1. Estructura de los directorios del servidor Apache



Figura 52. Estructura de directorios de servidor Apache

Una vez instalado el Apache, en el directorio raíz de la instalación, se encontrarán los siguientes directorios:

- **bin:** Ficheros ejecutables del Apache como arranque, cierre y otros scripts.
- **conf:** Ficheros de configuración del servidor, ficheros XML y los correspondientes DTD para la configuración de Tomcat.
- **lib:** librerías de las utilidades.
- **logs:** Directorio donde se almacenan los registros de acceso y errores del servidor, logs de Catalina y de las aplicaciones.
- **manual:** directorio donde se encuentra el manual del Apache.
- **proxy:** Directorio con los ficheros de la cache del servidor.
- **webapps:** Directorio que contiene las aplicaciones web
- **work:** Almacenamiento temporal de ficheros y directorios

4.2.2. Estructura de la aplicación

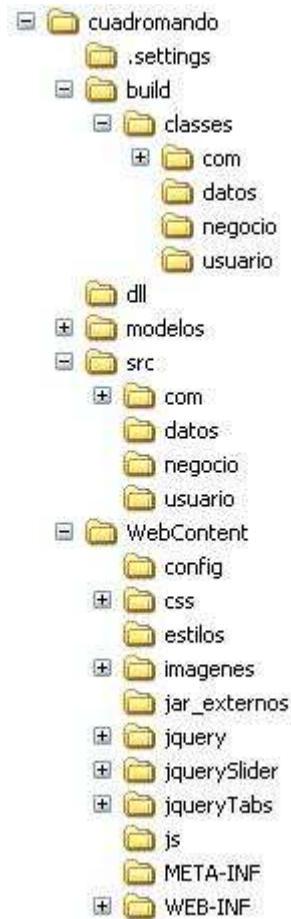


Figura 53. Estructura de directorios de la aplicación web

Directorio	Descripción
modelos	Modelos que pueden ser simulados por Vensim
src	Contiene el código fuente de las clases de la aplicación java
config	Archivos de configuración
css	Hojas de estilo de JQuery UI
estilos	Hojas de estilo
imagenes	Imágenes utilizadas en las pantallas JSP
Jar_externos	Jar externos que son utilizados por la aplicación
jquery	Archivos de JQuery
jquerySlider	Archivos de JQuery relacionados con el componente Slider
jqueryTabs	Archivos de JQuery relacionados con el componente Tabs
Js	Archivos JavaScript
META-INF	Descriptor de despliegue, Nombre y descripción de la aplicación que se despliega. Los diferentes módulos web y EJB que forman la aplicación
WEB-INF	Descriptor estándar de la aplicación Web. Incluye Filtros, Servlets, Parámetros de Contexto, Páginas de Error,...

Tabla 113. Estructura de directorios de la aplicación web

4.3. Batería de pruebas

Se han elaborado unos casos de prueba para los distintos escenarios que deben implementarse en la aplicación y deben ser probados. Casos de prueba tanto para la parte de administración de la aplicación como para la parte del cuadro de decisiones simulación de modelos.

4.3.1. Pruebas en el cuadro de mando

CP1	Acceder con un usuario no registrado
Tipo de prueba	Funcional, interfaz, BBDD.
Actor probador	Usuario no registrado
Objetivo	Comprobar el funcionamiento de las validaciones y verificar la correcta comunicación con la base de datos. Visualización de mensajes de información.
Procedimiento de prueba	En la pantalla de login intentamos acceder con un usuario que no esté registrado en la base de datos.
Salida esperada	Mensaje de información comunicándonos que el usuario que ha intentado iniciar sesión no se encuentra registrado.
Salida obtenida	Coincide con la esperada
Resultado de la prueba	La ejecución de la prueba ha sido exitosa.

Tabla 114. CP1 - Acceder con un usuario no registrado

CP2	Acceder con una contraseña incorrecta
Tipo de prueba	Funcional, interfaz, BBDD.
Actor probador	Usuario no registrado
Objetivo	Comprobar el funcionamiento de las validaciones y verificar la correcta comunicación con la base de datos. Visualización de mensajes de información.
Procedimiento de prueba	En la pantalla de login intentamos acceder con una contraseña incorrecta.
Salida esperada	Mensaje de información comunicándonos que la contraseña es incorrecta.
Salida obtenida	Coincide con la esperada
Resultado de la prueba	La ejecución de la prueba ha sido exitosa.

Tabla 115. CP2 - Acceder con una contraseña incorrecta

CP3	Registrarse con una contraseña incorrecta
Tipo de prueba	Funcional, interfaz, BBDD.
Actor probador	Usuario no registrado
Objetivo	Comprobar el funcionamiento de las validaciones y verificar la correcta comunicación con la base de datos. Visualización de mensajes de información.
Procedimiento de prueba	En la pantalla de registro de usuario intentamos registrarnos con una contraseña incorrecta.
Salida esperada	Mensaje de información comunicándonos que la contraseña es incorrecta.
Salida obtenida	Coincide con la esperada
Resultado de la prueba	La ejecución de la prueba ha sido exitosa.

Tabla 116. CP3 - Intentar registrarse con una contraseña incorrecta

CP4	Registrarse con un email incorrecto
Tipo de prueba	Funcional, interfaz, BBDD.
Actor probador	Usuario no registrado
Objetivo	Comprobar el funcionamiento de las validaciones y verificar la correcta comunicación con la base de datos. Visualización de mensajes de información.
Procedimiento de prueba	En la pantalla de registro de usuario registrarnos una dirección de correo electrónico incorrecta.
Salida esperada	Mensaje de información comunicando que la dirección introducida es incorrecta, no cumple el estándar nombre@dominio.servidor.
Salida obtenida	Coincide con la esperada
Resultado de la prueba	La ejecución de la prueba ha sido exitosa.

Tabla 117. CP4 - Intentar registrarse con una dirección de correo incorrecta

CP5	Intentar registrarse dejando algún campo del formulario vacío
Tipo de prueba	Funcional, interfaz, BBDD.
Actor probador	Usuario no registrado
Objetivo	Comprobar el funcionamiento de las validaciones y verificar la correcta comunicación con la base de datos. Visualización de mensajes de información.
Procedimiento de prueba	En la pantalla de registro de usuario intentamos registrarnos dejando alguno de los campos del formulario vacío.
Salida esperada	Mensaje de información comunicándonos que el campo con valor vacío es obligatorio y debe rellenarse.
Salida obtenida	Distinta a la esperada
Resultado de la prueba	La ejecución de la prueba no ha sido exitosa.
Solución adoptada	Realizamos una comprobación de que todos los cuadros de texto tengan valor distinto a vacío.

Tabla 118. CP5 - Intentar registrarse dejando algún campo del formulario vacío

CP6	Intentar registrarse con un nombre de usuario ya registrado en el sistema
Tipo de prueba	Funcional, interfaz, BBDD.
Actor probador	Usuario no registrado
Objetivo	Comprobar el funcionamiento de las validaciones y verificar la correcta comunicación con la base de datos. Visualización de mensajes de información.
Procedimiento de prueba	En la pantalla de registro de usuario intentamos registrarnos con un nombre de usuario ya registrado en el sistema.
Salida esperada	Mensaje de información comunicándonos que ya existe un usuario con ese nombre de usuario.
Salida obtenida	Coincide con la esperada
Resultado de la prueba	La ejecución de la prueba ha sido exitosa.

Tabla 119. CP6 - Intentar registrarse con un nombre de usuario ya registrado

CP7	Cargar un modelo que no tenga extensión vpm
Tipo de prueba	Funcional, interfaz, BBDD.
Actor probador	Administrador
Objetivo	Comprobar el funcionamiento de las validaciones y visualización de mensajes de información.
Procedimiento de prueba	En la pantalla de Gestión de Cuadros de Mando cargamos un modelo con extensión no vpm.
Salida esperada	Mensaje de información comunicándonos que el fichero a cargar no tiene extensión vpm.
Salida obtenida	Distinta a la esperada
Resultado de la prueba	La ejecución de la prueba no ha sido exitosa.
Solución adoptada	Comprobación de que el fichero seleccionado tenga extensión .vpm o .VPM

Tabla 120. CP7 - Cargar un modelo que no tenga extensión vpm

CP8	Intentar crear un cuadro de mando dejando algún campo de los formularios vacío
Tipo de prueba	Funcional, interfaz.
Actor probador	Administrador
Objetivo	Comprobar el funcionamiento de las validaciones y visualización de mensajes de información.
Procedimiento de prueba	En las pantallas de Gestión de Cuadros de Mando intentamos crear un cuadro de mando dejando a vacío alguno de los campos obligatorios.
Salida esperada	Mensaje de información comunicándonos que el campo de texto en cuestión se debe rellenar.
Salida obtenida	Coincide con la esperada
Resultado de la prueba	La ejecución de la prueba ha sido exitosa.

Tabla 121. CP8 - Intentar crear un cuadro de mando con campos vacíos

CP9	Crear un grupo de entrada/salida ya existente
Tipo de prueba	Funcional, interfaz, BBDD.
Actor probador	Administrador
Objetivo	Comprobar el funcionamiento de las validaciones y visualización de mensajes de información.
Procedimiento de prueba	En las pantallas de Gestión de Cuadros de Mando creamos un grupo de entrada/salida ya existente.
Salida esperada	Mensaje de información comunicándonos que ya existe un grupo de entrada con ese nombre.
Salida obtenida	Coincide con la esperada
Resultado de la prueba	La ejecución de la prueba ha sido exitosa.

Tabla 122. CP9 - Crear un grupo de entrada/salida ya existente

CP10	Inserción de valores no numéricos en el rango
Tipo de prueba	Funcional, interfaz.
Actor probador	Administrador
Objetivo	Comprobar el funcionamiento de las validaciones y visualización de mensajes de información.
Procedimiento de prueba	En las pantallas de Gestión de Cuadros de Mando intentamos introducir valores no numéricos en el rango de valores de la variable de entrada.
Salida esperada	Mensaje de información comunicándonos que debemos introducir un valor entero para el rango de valores de la variable de entrada.
Salida obtenida	Distinta a la esperada
Resultado de la prueba	La ejecución de la prueba ha sido exitosa.
Solución adoptada	Comprobación del texto introducido y de que éste sea numérico.

Tabla 123. CP10 - Inserción de valores no numéricos en el rango

CP11	Inserción de valores no numéricos en campos de texto que deben tener valor numérico
Tipo de prueba	Funcional, interfaz.
Actor probador	Administrador
Objetivo	Comprobar el funcionamiento de las validaciones y visualización de mensajes de información.
Procedimiento de prueba	En las pantallas de Gestión de Cuadros de Mando intentamos introducir valores no numéricos en el campo de texto de una variable numérica.
Salida esperada	Mensaje de información comunicándonos que debemos introducir un valor entero para dicha variable.
Salida obtenida	Distinta a la esperada
Resultado de la prueba	La ejecución de la prueba no ha sido exitosa.
Solución adoptada	Realizamos una comprobación de los datos introducidos para evitar la inserción de valores distintos a números.

Tabla 124. CP11 - Inserción de valores no numéricos en campos numéricos

Estos casos de uso que acabamos de describir para la creación de un cuadro de mando son válidos también para la modificación de cuadros de mando, al ser similares los casos de prueba no los incluimos en la memoria de este proyecto, los resultados de dichos casos de prueba en su ejecución son exitosos.

CP12	Creación de una sala con la misma clave que otra ya sala ya creada anteriormente
Tipo de prueba	Funcional, interfaz, BBDD.
Actor probador	Administrador
Objetivo	Comprobar el funcionamiento de las validaciones y visualización de mensajes de información.
Procedimiento de prueba	En la pantalla de Gestión de Salas de Operaciones intentamos crear una sala de operaciones con una clave que ya la tenga otra sala de operaciones.
Salida esperada	Mensaje de información comunicándonos que ya existe una sala de operaciones con esa clave.
Salida obtenida	Coincide con la esperada
Resultado de la prueba	La ejecución de la prueba ha sido exitosa.

Tabla 125. CP12 - Creación de una sala de operaciones con clave repetida

CP13	Creación de una sala sin asignarle cuadros de mando
Tipo de prueba	Funcional, interfaz.
Actor probador	Administrador
Objetivo	Comprobar el funcionamiento de las validaciones y visualización de mensajes de información.
Procedimiento de prueba	En la pantalla de Gestión de Salas de Operaciones intentamos crear una sala de operaciones sin asignarle ningún cuadro de mando.
Salida esperada	Mensaje de información comunicándonos que ya existe una sala de operaciones con esa clave.
Salida obtenida	Coincide con la esperada
Resultado de la prueba	La ejecución de la prueba ha sido exitosa.

Tabla 126. CP13 - Creación de una sala sin asignarle cuadros de mando

Estos casos de uso que acabamos de describir para la creación de una sala de operaciones son válidos también para la modificación de salas de operaciones, al ser similares los casos de prueba no los incluimos en la memoria de este proyecto, los resultados de dichos casos de prueba en su ejecución son exitosos.

CP14	Modificación de un usuario asignándole el mismo nombre de usuario que otro usuario
Tipo de prueba	Funcional, interfaz.
Actor probador	Administrador
Objetivo	Comprobar el funcionamiento de las validaciones y visualización de mensajes de información.
Procedimiento de prueba	En la pantalla de Gestión de Usuarios intentamos modificar un usuario asignándole el mismo nombre de usuario que uno ya almacenado en el sistema.
Salida esperada	Mensaje de información comunicándonos que ya existe un usuario con ese nombre.
Salida obtenida	Coincide con la esperada
Resultado de la prueba	La ejecución de la prueba ha sido exitosa.

Tabla 127. CP14 - Modificar usuario con nombre de usuario repetido

4.3.2. Pruebas en el cuadro de decisiones

CP15	Cumplimentar datos con valores que no corresponden
Tipo de prueba	Funcional, interfaz.
Actor probador	Usuario registrado
Objetivo	Comprobar el funcionamiento de las validaciones y visualización de mensajes de información.
Procedimiento de prueba	Introducir valores no válidos para las variables palanca en el formulario de la pantalla principal para el usuario general.
Salida esperada	Mensaje de error informando de la incorrección de los datos introducidos.
Salida obtenida	Distinta a la esperada
Resultado de la prueba	La ejecución de la prueba no ha sido exitosa.
Solución adoptada	Realización de una comprobación del texto introducido para evitar que se inserten caracteres no válidos como letras por ejemplo.

Tabla 128. CP15 - Cumplimentar datos con valores que no corresponden

CP16	Cumplimentar datos con valores negativos
Tipo de prueba	Funcional, interfaz.
Actor probador	Usuario registrado
Objetivo	Comprobar el funcionamiento de las validaciones y visualización de mensajes de información.
Procedimiento de prueba	Introducir valores negativos para las variables palanca en el formulario de la pantalla principal para el usuario general.
Salida esperada	No aceptar valores negativos
Salida obtenida	Distinta a la esperada
Resultado de la prueba	La ejecución de la prueba no ha sido exitosa.
Solución adoptada	Realización de una comprobación de los valores introducidos para evitar que se inserten valores negativos, en caso de introducir valores negativos el cuadro de texto toma el valor cero.

Tabla 129. CP16 - Cumplimentar datos con valores negativos

CP17	Simular un período siendo el período actual el último posible
Tipo de prueba	Funcional, interfaz.
Actor probador	Usuario registrado
Objetivo	Comprobar el funcionamiento de las validaciones y visualización de mensajes de información.
Procedimiento de prueba	Simular el último periodo
Salida esperada	Automáticamente se deshabilita el botón de simular y simular todo para no poder simular más periodos, tenemos la opción de volver a jugar si así lo deseamos.
Salida obtenida	Coincide con la esperada
Resultado de la prueba	La ejecución de la prueba ha sido exitosa.

Tabla 130. CP17 - Simular periodo actual como último posible

CP18	Simular períodos restantes siendo el período actual el último posible
Tipo de prueba	Funcional, interfaz.
Actor probador	Usuario registrado
Objetivo	Comprobar el funcionamiento de las validaciones y visualización de mensajes de información.
Procedimiento de prueba	Simular períodos restantes siendo el actual el último período.
Salida esperada	Se simulan el resto de períodos restantes, también se deshabilitan el botón de simular y simular todo para no poder simular más periodos, tenemos la opción de volver a jugar si así lo deseamos.
Salida obtenida	Coincide con la esperada
Resultado de la prueba	La ejecución de la prueba ha sido exitosa.

Tabla 131. CP18 - Simular períodos restantes siendo el actual el último

CP19	Simular períodos restantes no siendo el período actual el último posible
Tipo de prueba	Funcional, interfaz.
Actor probador	Usuario registrado
Objetivo	Comprobar el funcionamiento de las validaciones y visualización de mensajes de información.
Procedimiento de prueba	Simular períodos restantes no siendo el actual el último período.
Salida esperada	Se simulan el resto de períodos restantes, también se deshabilitan el botón de simular y simular todo para no poder simular más periodos, tenemos la opción de volver a jugar si así lo deseamos.
Salida obtenida	Coincide con la esperada
Resultado de la prueba	La ejecución de la prueba ha sido exitosa.

Tabla 132. CP19 - Simular períodos restantes no siendo el actual el último

CP20	Consultar resultados en pestañas
Tipo de prueba	Funcional, interfaz.
Actor probador	Usuario registrado
Objetivo	Comprobar el correcto funcionamiento y visualización de resultados.
Procedimiento de prueba	Visualizar en las pestañas las salidas gráficas obtenidas tras las simulaciones realizadas.
Salida esperada	Para cada pestaña se construye el gráfico y se rellena la tabla con los correspondientes valores.
Salida obtenida	Coincide con la esperada
Resultado de la prueba	La ejecución de la prueba ha sido exitosa.

Tabla 133. CP20 - Consultar resultados en pestañas

CP21	Opciones en gráficos
Tipo de prueba	Funcional, interfaz.
Actor probador	Usuario registrado
Objetivo	Comprobar la correcta visualización de los gráficos según la opción elegida.
Procedimiento de prueba	Modificar las opciones de gráficos
Salida esperada	Representa los valores obtenidos de la simulación en la opción gráfica seleccionada.
Salida obtenida	Distinta a la esperada
Resultado de la prueba	La ejecución de la prueba no ha sido exitosa.
Solución adoptada	Comprobación en la recogida de información en la base de datos y posterior tratamiento de dicha información.

Tabla 134. Opciones en gráficos

4.4. Seguimiento de la Fase de Construcción

Fijándonos en la planificación de esta fase hemos realizado dos iteraciones. En la primera iteración se han implementado la interfaz, construir las pruebas e implementar las funcionalidades de los escenarios principales. Estas funcionalidades se han implementado y probado conforme a lo establecido en la batería de pruebas, corrigiéndose los módulos que producían resultados erróneos.

En la segunda iteración se han implementado el resto de funcionalidades y código. Al final de esta iteración se presentó la aplicación al cliente, quien probó y valoró su funcionamiento y presentación, solicitando los siguientes cambios:

- Cambios en la seguridad en la barra de dirección del navegador, redireccionar a la pantalla login si el usuario que intenta acceder a una de las pantallas no está registrado.
- Cambios a la hora de mostrar los datos de los resultados de una partida.
- Cambios en las opciones gráficas, representando dos gráficos por fila.
- Cambios con el usuario, el usuario jugador puede editar su perfil.

Estos cambios en la aplicación llevaron a repasar y modificar los escenarios y casos de uso establecidos en el documento de análisis y en el documento de diseño. Como ya se explicó en el apartado correspondiente, estos cambios ya están incluidos en la documentación.

El manual de usuario y de instalación, fueron desarrollados en gran parte en esta fase. Se acabarán de completar y se explicarán en la siguiente fase, fase de transición, para evitar su repetición en ambas fases y una mejor organización.

Debido a las modificaciones que se tuvieron que llevar a cabo tanto en la aplicación, como en la documentación asociada a ella y al proyecto, en esta fase se ha ido acumulando un retraso debido sobre todo tanto a la inexperiencia a la hora de planificar (riesgo 3), como a la amplitud de objetivos que quiere abarcar el cliente (riesgo 1) y con cambios en los requisitos, bien porque no estaban bien recogidos en el documento de análisis, bien porque se hayan añadido nuevos requisitos. En ambos casos se intentó actuar conforme a lo expuesto en el plan de contingencia.

4.5. Plan de Iteración de la Fase de Transición

En esta sección se expondrá el plan de iteración que se va a seguir en la etapa de construcción o implementación. Se parte de los hitos logrados en la fase de construcción, como es cada parte de la aplicación y el objetivo es que el cliente trabaje con ellos y ajustar el producto a sus necesidades.

4.5.1. Plan de trabajo para la iteración

En el diagrama de Gantt siguiente se muestra el plan de la iteración para la etapa de transición. Se ha planificado en una iteración, que consistirá en que el cliente trabaje con la aplicación, trate de aprender su funcionamiento y muestre los aspectos que menos le gustan para mejorarlo.

En esta fase también se acabará de desarrollar el manual de usuario de la aplicación y se redactará un manual de instalación de todos los componentes. Aunque el proyecto software estaría terminado, también se ha incluido los hitos para la entrega del proyecto final de carrera, como son la unión de todos los documentos del proceso, la revisión de la memoria y su impresión, dando por finalizado el proyecto.

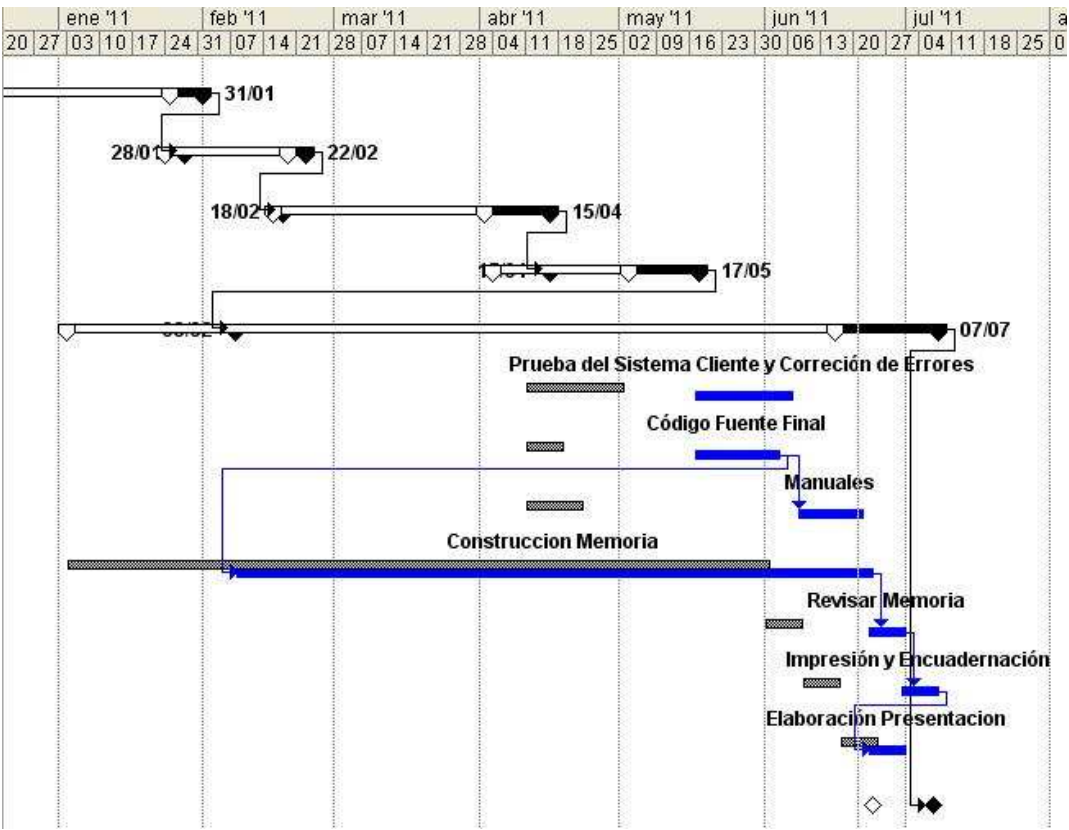


Figura 54. Diagrama de Gantt para la Fase de Transición

Nombre de tarea	Duración	Comienzo	Fin
<input checked="" type="checkbox"/> Transicion	108 días	mar 08/02/11	jue 07/07/11
Prueba del Sistema Cliente y Corrección de Errores	15 días	mar 17/05/11	lun 06/06/11
Código Fuente Final	14 días	mar 17/05/11	vie 03/06/11
Manuales	10 días	mié 08/06/11	mar 21/06/11
Construccion Memoria	92 días	mar 08/02/11	mié 15/06/11
Revisar Memoria	6 días	mié 15/06/11	mié 22/06/11
Impresión y Encuadernación	6 días	mié 22/06/11	mié 29/06/11
Elaboración Presentacion	6 días	jue 30/06/11	jue 07/07/11
Proyecto Finalizado	0 días	lun 11/07/11	lun 11/07/11

Figura 55. Tareas para la Fase de Transición

4.5.2. Recursos necesarios

Recursos Humanos

Los recursos humanos de esta iteración serán:

- Los desarrolladores
- El cliente.
- La imprenta (sólo a términos del proyecto final de carrera, no del proyecto software)

Recursos Software

Los recursos software para esta iteración serán:

- Servidor Web
 - Apache Tomcat 6.0
- Entorno de Programación
 - JavaScript
 - JQuery
 - JSP
- Sistema Gestor de Bases de datos
 - MySQL
- Entorno integrado de desarrollo
 - Eclipse
- Herramientas de modelado
 - StartUML
 - REM
- Herramienta de Planificación
 - Microsoft Project 2003
- Herramientas de documentación
 - Microsoft Word 2007
- Herramienta de sincronización
 - Dropbox
- Otros Recursos Software
 - Navegador Web: Mozilla Firefox, MS Internet Explorer, Google Chrome
 - Software de generación de PDF: Microsoft Word 2007

Recursos Hardware

Los recursos hardware para esta iteración serán:

- 2 Ordenadores portátiles personales
- Servidor remoto para instalación y pruebas de la aplicación
- Conexión a Internet
- Dispositivos de almacenamiento externo USB.

Capítulo 5

Fase de Transición

En este capítulo se describirán los pasos necesarios tanto para la instalación de la aplicación en el sistema operativo MS Windows como para el funcionamiento en un servidor de aplicaciones web como puede ser Apache Tomcat.



Figura 56. Hito en la Fase de Transición

5.1. Concepto de la Fase de Transición

La finalidad de la fase de transición es poner el producto en manos de los usuarios finales, para lo que típicamente se requerirá desarrollar nuevas versiones actualizadas del producto, completar la documentación, entrenar al usuario en el manejo del producto, y en general tareas relacionadas con el ajuste, configuración, instalación y usabilidad del producto.

A continuación se citan algunas de las cosas que puede incluir esta fase:

- Testeo de la versión Beta para validar el nuevo sistema frente a las expectativas de los usuarios.
- Funcionamiento paralelo con los sistemas legados que están siendo sustituidos por nuestro proyecto.
- Conversión de las bases de datos operacionales.
- Entrenamiento de los usuarios y técnicos de mantenimiento.
- Traspaso del producto a los equipos de marketing, distribución y venta.

Los principales objetivos de esta fase son:

- Conseguir que el usuario se valga por sí mismo.
- Un producto final que cumpla los requisitos esperados, que funcione y satisfaga suficientemente al usuario.

Los productos de la fase de transición según son:

- Prototipo Operacional.
- Documentación final como manual de usuario y manual de sistema.
- Caso del Negocio Completo.
- Línea de Base del Producto completa y corregida que incluye todos los modelos del sistema.
- Descripción de la Arquitectura completa y corregida.

Las iteraciones de esta fase irán dirigidas normalmente a conseguir una nueva versión. Las actividades a realizar durante las iteraciones dependerán de su finalidad, si es corregir algún error detectado, normalmente será suficiente con llevar a cabo los flujos de trabajo de implementación y test, sin embargo, si se deben añadir nuevas características, la iteración será similar a la de una iteración de la fase de construcción. La complejidad de esta fase depende totalmente de la naturaleza del proyecto, de su alcance y de la organización en la que deba implantarse.

5.2. Manual de instalación

5.2.1. Introducción

Para el correcto funcionamiento de la aplicación van a ser necesario disponer de los siguientes componentes:

- Servidor web Apache
- Sistema gestor de bases de datos MySQL
- Vensim 5.9c
- Cuadro de mando dinámico para juegos de empresa.

Con la realización de esta guía se busca que cualquier usuario con unos conocimientos básicos pueda instalar la aplicación para su uso.

En Internet podemos encontrar información complementaria para la instalación y configuración de servidores web y bases de datos, en los diferentes entornos y bajo las diferentes circunstancias

5.2.2. Instalación de Apache Tomcat

La instalación está basada para MS Windows XP en español. La diferencia entre la instalación en esta versión y versiones posteriores como MS Windows Vista también en español, es la sustitución de “Archivos de Programa” por “Program Files”.

Instalación de la máquina virtual de Java

Para el correcto funcionamiento del servidor de aplicaciones web es necesario tener instalado previamente una máquina virtual de Java, para ello vamos a instalar una versión de JDK, se puede descargar desde la siguiente dirección:

Dirección Web: <http://www.oracle.com/technetwork/java/javase/downloads/index.html>

La información necesaria podremos encontrar en la ruta del CD-ROM siguiente:
“\ Instalacion \ Java”.

Para instalarlo, bastará con hacer doble clic en el fichero y seguir el proceso de instalación tal y como se indica. Esto instalará el entorno de ejecución JRE 1.6.0 y el kit de desarrollo JDK 1.6.0 de Java en el sistema.

El directorio de instalación estándar es el siguiente: “C:\Archivos de programa\Java\jre6”.

Este directorio de instalación hará falta más adelante para configurar el servidor de aplicaciones Apache Tomcat

Después de haber instalado la máquina virtual, vamos a crear una variable de entorno de Windows llamada JAVA_HOME, y vamos a asignarle la ruta donde hemos instalado la máquina Java.

Para ello vamos a las propiedades de Mi PC bien pulsando la tecla de Windows + PAUSA ó botón Derecho sobre Mi PC y seleccionando la opción Propiedades del desplegable. En la ventana emergente que nos aparece hacemos clic sobre la pestaña de “Opciones avanzadas” y hacemos clic sobre el botón de “Variables de Entorno”.



Figura 57. Variable de sistema JAVA_HOME

Instalación del servidor de aplicaciones Apache Tomcat

Será necesario tener instalado un servidor de aplicaciones Apache Tomcat. Para este proyecto se ha optado por instalar la versión 6. Se puede descargar desde la siguiente dirección:

Dirección Web: <http://tomcat.apache.org/download-60.cgi>

Dentro del CD-ROM está incluido en el directorio “\ Aplicaciones \ Apache Tomcat”.

Para la instalación de Apache Tomcat basta con hacer clic sobre el fichero ejecutable llamado apache-tomcat-6.0.32 y nos aparecerá una ventana emergente siguiendo las instrucciones que nos indican realizamos la instalación.

Después de haber instalado del servidor de aplicaciones Apache Tomcat, vamos a crear una variable de entorno de Windows llamada CATALINA_HOME, y vamos a asignarle la ruta donde hemos instalado Apache Tomcat.

En nuestro caso ha sido el siguiente: “C:\Archivos de programa\Apache Software Foundation\Tomcat 6.0”.

Para ello vamos a las propiedades de Mi PC bien pulsando la tecla de Windows + PAUSA ó botón Derecho sobre Mi PC y seleccionando la opción Propiedades del desplegable. En la ventana emergente que nos aparece hacemos clic sobre la pestaña de “Opciones avanzadas” y hacemos clic sobre el botón de “Variables de Entorno.



Figura 58. Variable de sistema CATALINA_HOME

5.2.3. Instalación de Vensim

El primer paso es la instalación de la aplicación Vensim que va a ser la encargada de simular modelos de dinámica de sistemas.

Esta aplicación puede ejecutarse en MS Windows, ejecutamos el fichero llamado vendss32.exe y aparecerá una nueva ventana emergente similar a la de la figura que vemos a continuación.



Figura 59. Instalación Vensim Paso 1

Hacemos clic en el botón Next, como resultado nos aparecerá una nueva ventana emergente en la que deberemos introducir los datos relacionados con la licencia de Vensim. Dichos datos serán facilitados por los tutores del proyecto.

Vensim DSS Version 5.9e Installation Registration

Enter software registration information here.

Your name:

Company name

Registration Code

IMPORTANT: Your company name must be the same as that printed or supplied with your registration code. If you wish to transfer this license to another user please contact Ventana Systems, Inc.

Your Registration Code was either emailed to you or is printed on the jewel case or sleeve the CD was shipped in. The registration code is NOT case sensitive.

NOTE Vensim version 4 registration codes will not work.

<Back Next> Cancel

Figura 60. Instalación Vensim Paso 2

Tras introducir los datos relacionados con la licencia hacemos clic en el botón Next.

Aparecerá una nueva ventana emergente en la cual deberemos indicar el directorio donde queremos que sea instalada la aplicación Vensim como vemos en la figura que se muestra a continuación. En nuestro caso el directorio elegido es el directorio por defecto.

“C:\ Archivos de programa\ Vensim”.

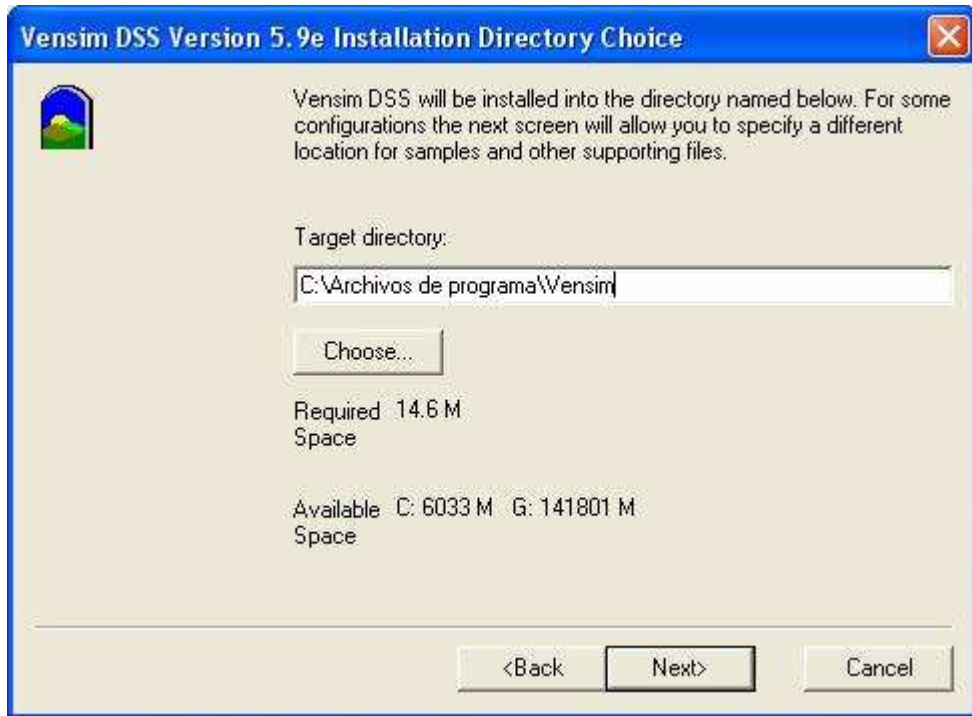


Figura 61. Instalación Vensim Paso 3

Seguidamente hacemos clic en el botón Next.

En la pantalla que nos aparece a continuación deberemos indicar la ruta deseada para almacenar los modelos y ejemplos de Vensim.

En nuestro caso la ruta seleccionada ha sido:

“C:\ Archivos de programa\ Vensim\ Modelos y ejemplos”.

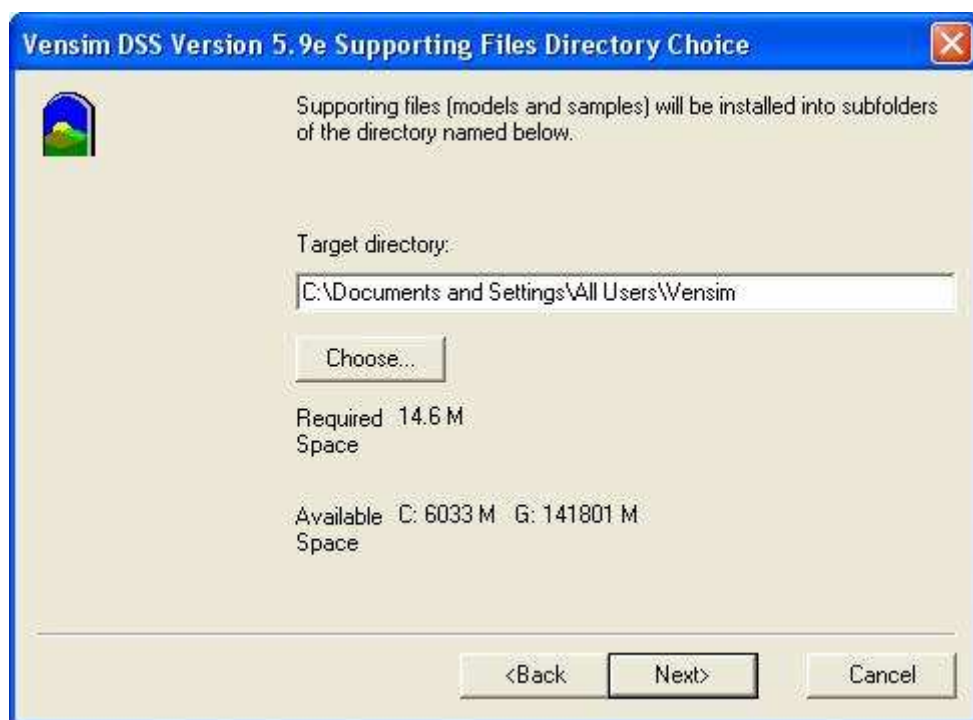


Figura 62. Instalación Vensim Paso 4

Después de introducir la ruta deseada para los modelos y ejemplos hacemos clic en el botón Next.

Nos aparecerá una nueva ventana emergente en la cual podremos configurar diferentes parámetros de la instalación de Vensim.

Podemos añadir y quitar la verificación a diversos parámetros en los cuales en algunos de ellos también tenemos la opción de seleccionar la ruta de instalación.

En este apartado se desaconseja verificar la opción llamada “(doublé précision)” ya que la instalación de Vensim con esta opción verificada trae incoherencias futuras en los resultados de las simulaciones de los modelos.

En nuestro caso hemos verificado las opciones que se pueden ver en la figura que viene a continuación.



Figura 63. Instalación Vensim Paso 5

Tras verificar las opciones deseadas hacemos clic en el botón Next.

En la siguiente ventana emergente que nos aparece tenemos la opción de seleccionar la ubicación dentro del menú de inicio de MS Windows donde nos añadirá un acceso a la aplicación Vensim.

En esta ventana también tenemos la opción de añadir un nuevo icono a nuestro escritorio para acceder de manera directa a la aplicación Vensim como vemos a continuación en la siguiente figura.



Figura 64. Instalación Vensim Paso 6

Hacemos clic en el botón Next donde seguidamente comenzará la instalación.

En la ventana emergente que nos aparece tenemos diferentes indicadores que muestran el progreso de la instalación como podemos ver en la figura siguiente.

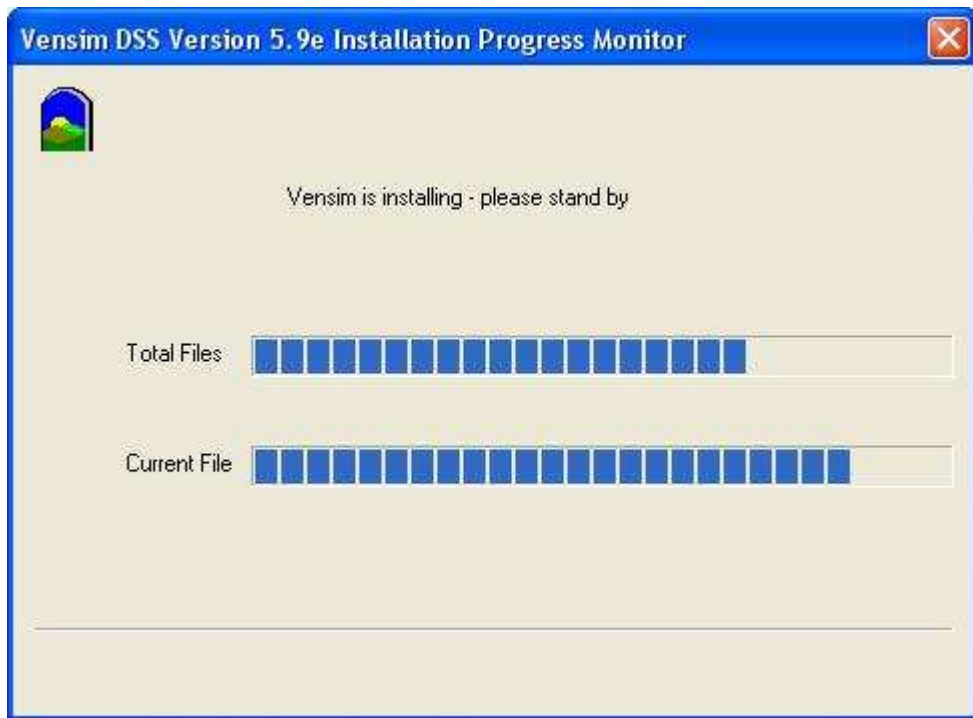


Figura 65. Instalación Vensim Paso 7

Para finalizar nos aparece la siguiente ventana emergente donde podemos visualizar el fichero de notas de la aplicación, también tenemos la opción de poder comenzar con Vensim.



Figura 66. Instalación Vensim Paso 8

5.2.4. Instalación del código de la aplicación en el servidor de aplicaciones

Este paso consistirá simplemente en copiar la carpeta cuadromando en la carpeta de aplicaciones del servidor. Para nuestro caso en concreto, la ruta donde la debemos copiar será:

C:\Archivos de Programa\Apache Software Foundation\Tomcat 6.0\webapps\cuadromando\

5.2.5. Instalación de los drivers de control de las bases de datos para Apache

Para realizar dicha instalación simplemente se copia los ficheros contenidos en la carpeta de ficheros Drivers BD-Apache en la carpeta de librerías de Apache.

C:\Archivos de programa\Apache Software Foundation\Tomcat 6.0\lib

5.2.6. Instalación de la base de datos

Será necesario tener instalado una base de datos para el almacenamiento de todos los datos relacionados con la aplicación como cuadros de mando, salas de operaciones, usuarios, etc. Para este proyecto se ha utilizado MySQL como SGBD.

Host: localhost
User: cuadromando
Password: cuadromando
Database: admin
Port: 3306

Esta aplicación puede ejecutarse en MS Windows, dicha aplicación se encuentra en la ruta del CD-ROM siguiente: “ \ Instalación \ Aplicaciones \ MySQL”.

5.3. Restricciones y requisitos de Vensim

En este apartado vamos a tratar las restricciones y requisitos en los modelos utilizados por Vensim para simular, todas las modificaciones relacionadas con Vensim que hay que realizar para un correcto funcionamiento de la aplicación, también vamos a tratar las configuraciones necesarias para una correcta simulación de los modelos.

5.3.1. Restricciones en los modelos

En cuanto a los requisitos tecnológicos, la empresa ha sido estricta en cuanto a la utilización de Vensim DSS para la creación y carga de modelos de dinámica de sistemas.

El modelo no debe contener variables tipo *constante de subíndice*. Las variables tipo *constante* deben estar agregadas como variables tipo juego (*GAME*) para que sea posible modificar su valor durante una simulación.

Todas las variables deben tener valor en las unidades.

Para que Vensim reconozca las variables deberemos modificar las variables palanca a variables tipo juego (*GAME*), de esta manera Vensim puede trabajar con ellas.

Hay que realizar un paso que es copiar el diagrama entero y crear un nuevo modelo

A la hora de simular con un modelo es necesario que esté empaquetado con extensión VPM, sino no es posible cargarlos en Vensim y hacer una simulación desde la aplicación web. Para empaquetarlos hay que acceder a Vensim y cargar el modelo que queremos empaquetar. Una vez cargado nos dirigimos a archivo/publicar. Debemos empaquetarlo junto con el archivo base.vdf.

Para empaquetarlos tenemos que acceder a Vensim y cargar el modelo que queremos empaquetar, una vez cargado nos dirigimos a File > Publish y en el nuevo cuadro de dialogo hacemos clic en el botón Publish.

5.3.2. Requisitos en el desarrollo con Eclipse

La clase Vensim que proporciona la funcionalidad para podernos comunicar con Vensim debe estar alojada en el siguiente directorio:

“src > com > vensim > Vensim.java”

Donde src y com son paquetes y vensim una carpeta

Antes de iniciar la aplicación deberemos modificar la variable de entorno “Path” de MS Windows y añadir el directorio donde se encuentra ubicada la biblioteca dll necesaria para poder comunicarnos con Vensim. Al cargar modelos, deben estar junto con la dll.

Al trabajar con una librería .dll nos encasillamos en solo poder utilizar la aplicación en MS Windows lo cual es un hándicap, no hay la posibilidad de que sea multiplataforma.

5.3.3. Requisitos generales

Uno de los requerimientos es que los modelos y las partidas deben de estar almacenados en el servidor en la misma carpeta para poder acceder a ellos.

5.4. Manual de Usuario

5.4.1. Pantalla principal

A continuación se muestra el manual de usuario. Se trata de una guía de referencia para el usuario mediante una serie de pasos pueda aprender a manejar la aplicación.

La primera pantalla que nos encontramos es la pantalla de Login de usuario, en dicha pantalla tenemos diferentes opciones. El usuario tiene la opción de conectarse si se trata de un usuario ya registrado en el sistema, tiene la opción de registrarse si aún no lo ha hecho.



La imagen muestra la interfaz de usuario para el login. En la parte superior hay una barra azul con el logo 'cuadromando' a la izquierda y el botón 'Registrar' a la derecha. El contenido principal es un recuadro azul con el título 'Login Usuario'. Debajo del título, se indica: 'Por favor introduzca su nombre de usuario y contraseña'. Hay dos campos de entrada: 'Usuario:' y 'Contraseña:'. Debajo de estos campos hay un botón que dice 'Iniciar Sesión'.

Figura 67. Login Usuario

Para conectarse a la aplicación el usuario deberá rellenar los campos de texto usuario y contraseña y a continuación hacer clic en el botón *Iniciar Sesión*, el usuario accede a la pantalla principal dependiendo de si es usuario o administrador según sus permisos, las pantallas correspondientes son diferentes para cada tipo de usuario.

Si por el contrario el usuario no está aún registrado en el sistema puede hacerlo, haciendo clic en el hipervínculo *Registrar* a continuación aparecerá una nueva ventana similar a la de la figura 22.



cuadromando Registrar

Registro Usuario

Por favor introduzca la información de registro

Usuario:

Contraseña: (Contraseña facilitada por el administrador)

Nombre:

Apellido:

Correo electrónico:

Registrar

Figura 68. Registro Usuario

En esta pantalla el usuario deberá rellenar todos los campos de texto, dichos campos son datos relacionados con el usuario, cabe destacar que en el campo de texto *Contraseña* el usuario deberá insertar la contraseña que le haya facilitado el usuario administrador del sistema.

Los datos introducidos serán validados para evitar las incompatibilidades descritas en las especificaciones del proyecto. En el caso de que los datos introducidos sean correctos el sistema nos lo indica por pantalla, el registro ha finalizado de manera satisfactoria, y nos reenviaría a la pantalla principal para que si lo deseamos podamos identificarnos.



cuadromando Registrar

Usuario jorhern registrado satisfactoriamente

Login Usuario

Por favor introduzca su nombre de usuario y contraseña

Usuario:

Contraseña:

Iniciar Sesión

Figura 69. Usuario registrado satisfactoriamente

5.4.2. Pantalla Principal Administrador

Una vez haya iniciado sesión el usuario administrador, accede al menú de administrador. En la parte superior derecha se verá el nombre del administrador conectado, mientras que en la esquina de la parte superior izquierda aparecerá el logotipo de la aplicación, el cual nos permite volver a la pantalla principal. Este enlace es común en la mayoría de las pantallas de la aplicación.

Tres serán fundamentalmente las opciones que tendrá este tipo de usuario. La primera de todas ellas es la gestión de cuadros de mando donde podrá realizar cualquier tipo de operación con los cuadros de mando. La segunda se trata de poder realizar la gestión de las salas de operaciones, de la misma manera que para los cuadros de mando, este usuario tiene permisos para realizar cualquier tipo de operación con las salas de operaciones. La tercera y última es la gestión de los usuarios registrados en nuestra aplicación.



Figura 70. Pantalla Principal Administrador

Gestión de Cuadros de Mando

Una vez elegida la opción de *Gestión de Cuadros de Mando* en el menú principal del administrador, pasaremos a una nueva pantalla que contiene un formulario donde podremos crear un nuevo cuadro de mando cargando antes el modelo que lo va a contener.

Nos va a permitir también modificar y eliminar aquellos cuadros de mando que ya han sido creados anteriormente siempre y cuando éstos no estén asignados a ninguna sala de operaciones.

Los datos introducidos durante la creación y modificación de un cuadro de mando se validarán para que cumplan las especificaciones del proyecto.



Figura 71. Gestión de Cuadros de Mando

Crear un nuevo cuadro de mando

A la hora de crear un nuevo cuadro de mando lo primero que deberemos hacer es cargar el modelo sobre el que vamos a realizar las simulaciones, una de las restricciones que nos hemos encontrado a la hora de trabajar con Vensim DSS es que el modelo a cargar debe tener extensión *vpm*, una vez cargado el modelo podremos empezar con la configuración del cuadro de mando.



Figura 72. Cargar modelo

Crear un nuevo cuadro de mando – Datos cuadro de mando

En esta pantalla nos dará la posibilidad de poder dar un nombre al cuadro de mando y una pequeña descripción.

Tras haber rellenado dichos valores hacemos clic en el Botón *Siguiente pantalla*.

The screenshot shows the 'cuadromando' web application interface. The top navigation bar is blue and contains the logo, the text 'cuadromando', and a user status 'Bienvenido josgenz (Desconectar)' with the date '15 de junio de 2011, 18:04:43'. Below the navigation bar are three menu items: 'Gestión de Cuadros de Mando', 'Gestión de Salas de Operaciones', and 'Gestión de Usuarios'. The main content area has a blue background and features a progress bar at the top with five steps: 'Datos del cuadro de mando' (highlighted), 'Grupos de variables de entrada', 'Configuración de variables juego', 'Grupos de variables de salida', and 'Configuración de variables de salida'. The title of the current screen is 'NOMBRE Y DESCRIPCIÓN DEL CUADRO DE MANDO PARA EL MODELO PRODUC.VPM'. Below the title, there are two input fields: a text box for 'Introduzca un nombre para el cuadro de mando' and a larger text area for 'Introduzca si desea una descripción de la configuración'. At the bottom of the form are two buttons: 'Pantalla anterior' and 'Siguiente pantalla'.

Figura 73. Nombre y descripción del cuadro de mando

Crear un nuevo cuadro de mando – Grupos de variables de entrada

En esta pantalla podemos crear diferentes grupos para agrupar las variables de entrada del modelo seleccionado.



Figura 74. Agrupaciones de las variables de entrada

Crear un nuevo cuadro de mando – Configuración de variables de entrada

En esta pantalla podemos configurar las variables de entrada del modelo. Seleccionamos aquellas variables que van a intervenir en la simulación. Los valores que podemos configurar son los siguientes:

- **Nombre Variable Palanca:** Nombre que va a tomar la variable en la simulación.
- **Nombre unidades:** nombre que van a tomar las unidades en las que se mide la variable.
- **Grupo de entrada:** Grupo al que va a pertenecer la variable palanca.
- **Rango de valores:** Rango en los que la variable palanca va a tomar valor.
- **Número total de simulaciones:** Número de simulaciones a realizar.
- **Nombre unidades:** Nombre que van a tomar las unidades en las que se va a simular

cuadromando

Bienvenido **jsgonz2** (Desconectar)
15 de junio de 2011, 18:05:52

Gestión de Cuadros de Mando Gestión de Salas de Operaciones Gestión de Usuarios

Diseño del cuadro de mando Grupos de variables de entrada **Configuración de variables juego** Grupos de variables de salida Configuración de variables de salida

CONFIGURACIÓN DE LAS VARIABLES JUEGO PARA EL MODELO PRODUC.VPM

Palanca	Nombre para el juego	Unidades	Nombre para el juego	Agrupamiento	Rango de valores
<input type="checkbox"/> productivity	<input type="text"/>	tasks/(month*person)	<input type="text"/>	GrupoEntrada1	Desde <input type="text"/> Hasta <input type="text"/>
<input type="checkbox"/> time step	<input type="text"/>	month	<input type="text"/>	GrupoEntrada1	Desde <input type="text"/> Hasta <input type="text"/>
<input type="checkbox"/> workers	<input type="text"/>	people	<input type="text"/>	GrupoEntrada1	Desde <input type="text"/> Hasta <input type="text"/>

Selecciona el número total de simulaciones para el juego y el nombre de las unidades: "Month"

Pantalla anterior Siguiete pantalla

Figura 75. Configuración de las variables juego

Crear un nuevo cuadro de mando – Grupos de variables salida

En esta pantalla podemos crear diferentes grupos para las variables de salida del modelo seleccionado.



Figura 76. Agrupaciones de las variables de salida

Crear un nuevo cuadro de mando – Configuración de variables de salida

En esta pantalla podemos configurar las variables de salida del modelo. Seleccionamos aquellas variables que van a aparecer en las diferentes representaciones gráficas. Los valores que podemos configurar son los siguientes:

- **Nombre Variable Palanca:** Nombre que va a tomar la variable en la simulación.
- **Nombre unidades:** nombre que van a tomar las unidades en las que se mide la variable.
- **Grupo de entrada:** Grupo al que va a pertenecer la variable palanca.
- **Tipo de representación:** Gráfica o reloj.
- **Nombre unidades:** Nombre que van a tomar las unidades en las que se va a simular

Variable	Nombre para el juego	Unidades	Nombre para el juego	Agrupamiento	Tipo representación	Grupo mixto	Grupo resumen
<input checked="" type="checkbox"/> producing		tasks/month		GrupoSalida1	Gráfica temporal	<input type="checkbox"/>	<input type="checkbox"/>

Figura 77. Configuración de las variables de salida

Después de seleccionar las variables que van a aparecer en las representaciones gráficas de salida hacemos clic en el botón *Guardar*.

Seguidamente aparece una nueva pantalla informándonos de que la creación del cuadro de mando se ha realizado de manera satisfactoria.

Figura 78. Cuadro de mando creado satisfactoriamente

A la hora de seleccionar la representación grafica en la configuración de las variables de salida sería interesante que se pueda seleccionar el escenario “Rojo – Amarillo – Verde – Amarillo – Rojo” a parte de los ya implementados, además sería interesante realizar una interfaz más intuitiva de los selectores por ejemplo que en dicha pantalla se pudiera elegir el valor de cada color mediante el componente *Slider*.

Modificar cuadro de mando – Datos cuadro de mando

En esta pantalla nos dará la posibilidad de poder modificar el nombre y descripción dados al cuadro de mando anteriormente. Tras haber rellenado dichos valores hacemos clic en el Botón *Siguiente*.

Modificar cuadro de mando – Grupos de variables de entrada

En esta pantalla podemos modificar los diferentes grupos para agrupar las variables de entrada del modelo seleccionado.

Modificar cuadro de mando – Configuración de variables de entrada

En esta pantalla podemos modificar las configuraciones realizadas en las variables de entrada del modelo.

Modificar cuadro de mando – Grupos de variables salida

En esta pantalla podemos modificar los diferentes grupos para las variables de salida del modelo seleccionado.

Modificar cuadro de mando – Configuración de variables de salida

En esta pantalla podemos modificar las configuraciones realizadas en las variables de salida del modelo. Después de realizar los cambios hacemos clic en el botón *Guardar*.

Aparece una nueva ventana avisándonos de que el cuadro de mando se ha modificado satisfactoriamente.

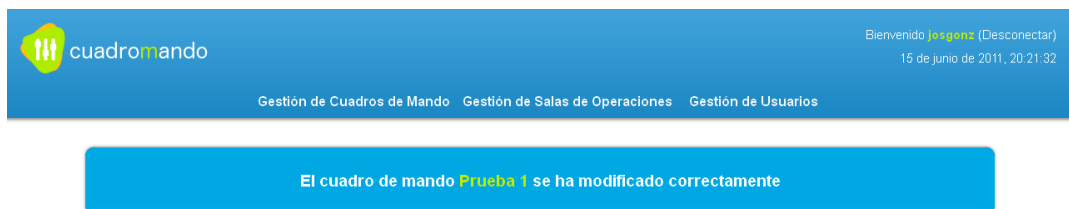


Figura 79. Cuadro de mando modificado correctamente

Eliminar cuadro de mando

Mediante esta opción se puede eliminar un cuadro de mando

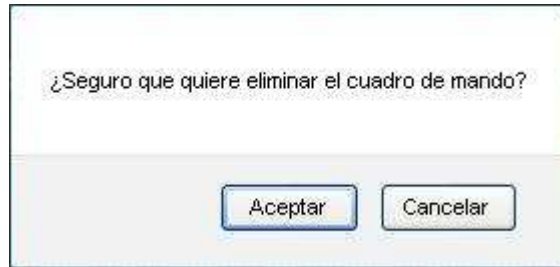


Figura 80. Eliminar cuadro de mando

Gestión de Salas de Operaciones

Una vez elegida la opción de *Gestión de Salas de Operaciones* en el menú principal del administrador, pasaremos a una nueva pantalla que contiene un formulario donde podremos crear una nueva sala de operaciones.

Los usuarios van a poder jugar partidas sobre los cuadros de mando pertenecientes a dichas salas de operaciones.

Nos va a permitir también modificar y eliminar aquellas salas de operaciones que ya han sido creadas anteriormente.

Los datos introducidos durante la creación y modificación de una sala de operaciones se validarán para que cumplan las especificaciones del proyecto.

Crear una nueva Sala de Operaciones

Aparecerá una nueva ventana emergente en la cual podemos asignar un nombre a la sala de operaciones y una clave. Mediante la clave lo que vamos a realizar es un filtro en aquellos usuarios que dispongan de la misma clave que la clave de la sala de operaciones por tanto en dicha sala de operaciones solo está los usuarios con clave del mismo valor que la clave de la sala de operaciones.

En esta pantalla también podemos ver los cuadros de mando disponibles, son aquellos que están libres o sea en ninguna sala de operaciones. Con esta selección la sala de operaciones tendrá dichos cuadros de mando para que los usuarios puedan realizar simulaciones.



Figura 81. Gestión de Salas de Operaciones

Modificar sala de operaciones

Aparecerá una nueva ventana emergente que nos muestra la información almacenada actualmente en la base de datos en la cual, podemos modificar los datos relacionados con la sala de operaciones tales como el nombre y la clave.

En esta pantalla también podemos modificar los cuadros de mando asignados a la sala de operaciones. Veremos los cuadros de mando disponibles para seleccionar.

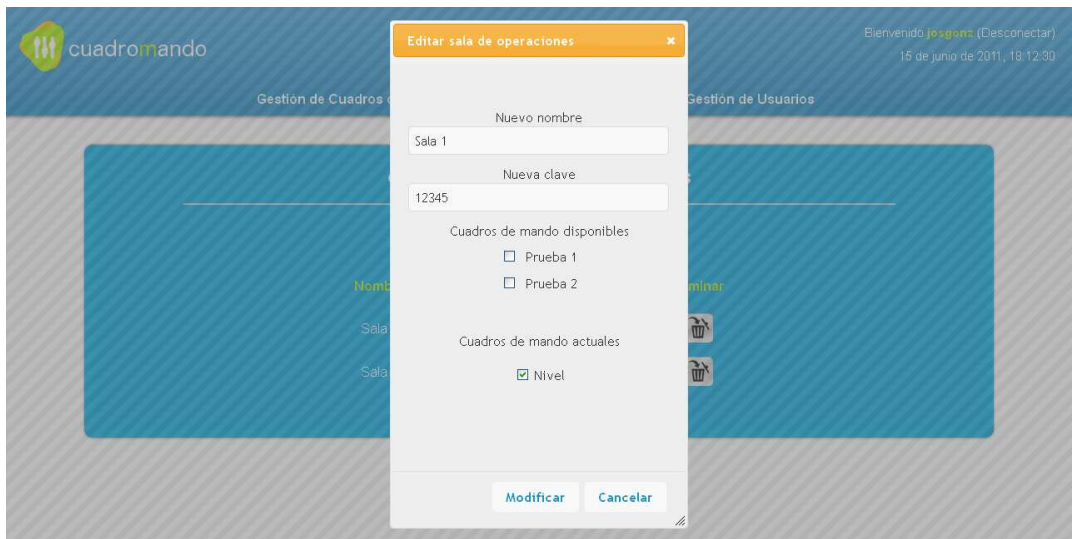


Figura 82. Modificar Sala de Operaciones

Eliminar sala de operaciones

Con esta opción podemos eliminar las salas de operaciones anteriormente creadas. Nos aparecerá un mensaje de aviso para informarnos de si estamos seguros de la decisión. En caso de confirmar la eliminación de una sala de operaciones también se eliminaran aquellos usuarios pertenecientes a dicha sala de operaciones.

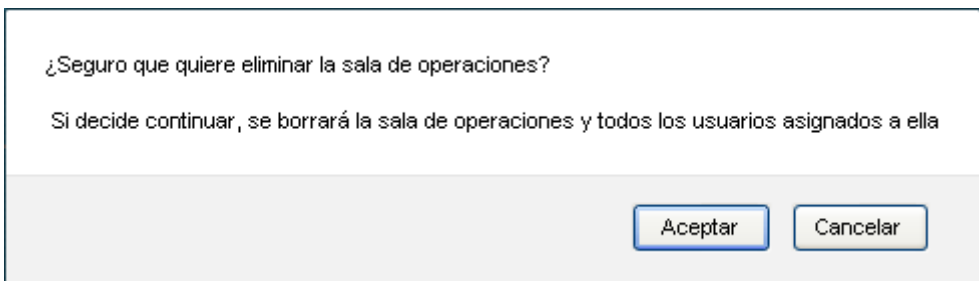



Figura 83. Eliminar Sala de Operaciones

Gestión de Usuarios

Una vez elegida la opción de *Gestión de Usuarios* en el menú principal del administrador, pasaremos a una nueva pantalla que contiene un formulario donde podremos realizar la gestión de los usuarios registrados en el sistema.

Se muestra un listado con todos los usuarios registrados, sobre ellos se pueden realizar operaciones de modificación y eliminación.

 **cuadromando**

Bienvenido **jstgona** (Desconectar)
15 de junio de 2011, 18:19:22

Gestión de Cuadros de Mando Gestión de Salas de Operaciones **Gestión de Usuarios**

GESTIÓN DE USUARIOS

Usuarios existentes

Sala de operaciones	Usuario	Password	Nombre	Apellido	Email	Modificar	Eliminar
	jsgonz	1	Jose Antonio	Gonzalez Alejos	agila23@gmail.com		
	mardelv	2	Mario	del Valle Guijarro	correodemariete@gmail.com		
Sala 1	rubmarc	12345	Rubén	Marcos Gonzalez	rubmarg@gmail.com		
Sala 1	lulvegv	12345	Luis	Vegas del Valle	luisvegas@gmail.com		
Sala 1	albnarm	12345	Albano	Narganes Merino	banete@gmail.com		
Sala 1	vichern	12345	Victor	Hernandez San Juan	vichern@gmail.com		
Sala 1	jorfem	12345	Jorge	Fernandez Velasco	jorgefernandez@gmail.com		
Sala 2	marherv	11111	Marta	Hernandez Vega	marherv@hotmail.com		

Figura 84. Gestión de Usuarios

Modificar usuario

Con esta opción podemos modificar los datos referentes a un usuario registrado en el sistema. Al elegir esta opción aparece una nueva ventana emergente que nos muestra la información almacenada en la base de datos.

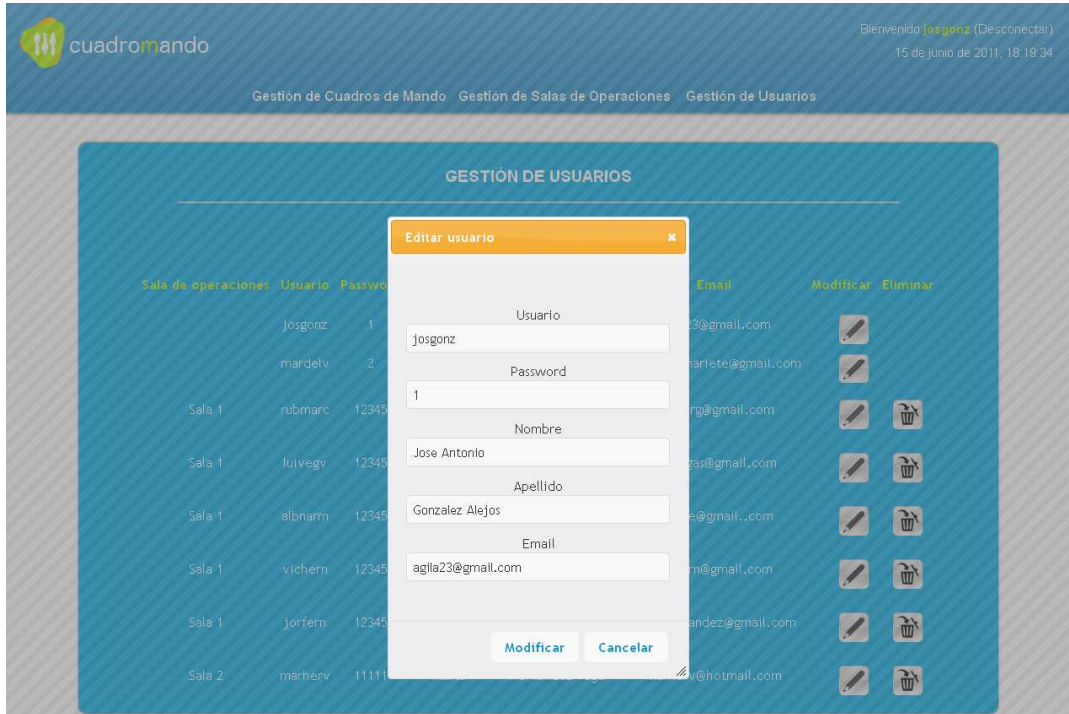


Figura 85. Modificar Usuario

Eliminar usuario

Con esta opción podemos eliminar un usuario registrado en el sistema. Al elegir esta opción aparece un mensaje de aviso para informarnos de si estamos seguros de la decisión tomada. En caso de eliminar un usuario se eliminarán todas las partidas realizadas por dicho usuario.

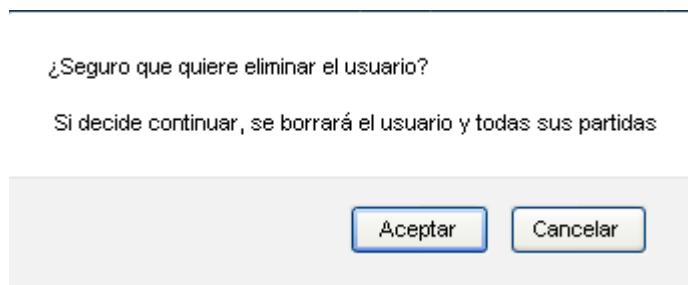


Figura 86. Eliminar Usuario

5.4.3. Pantalla Principal Usuario

Pantalla principal usuario

Una vez haya iniciado sesión el usuario, en la pantalla se diferencian 3 zonas.

En la parte superior izquierda están situados los cuadros de mando pertenecientes a la sala de operaciones en la que el usuario está jugando.

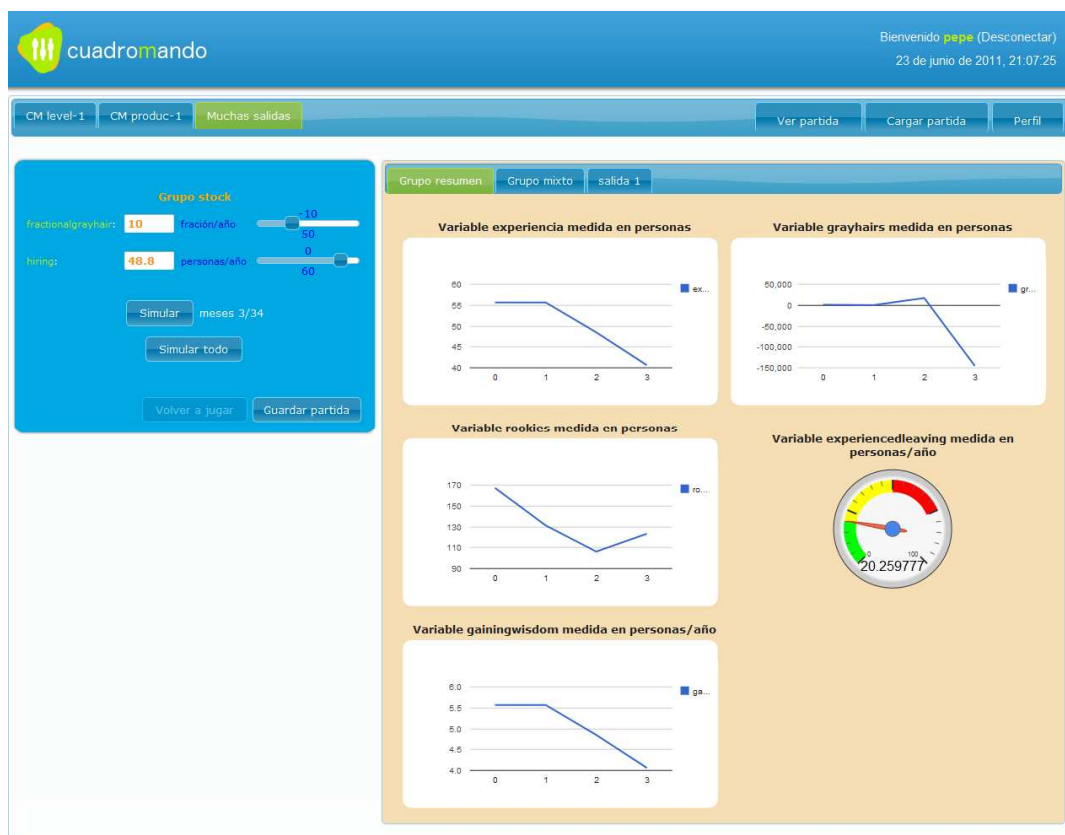


Figura 87. Pantalla principal usuario

También podemos visualizar 3 opciones en la parte superior derecha, *Ver partida*, *Cargar partida* y *Perfil*.

En la parte izquierda de la pantalla se sitúa el panel de configuración donde se visualizan las variables palanca pertenecientes al cuadro de mando con el que el usuario esta simulando, en dicho panel el usuario podrá simular con el cuadro de mando.

Dentro de este panel de configuración podemos visualizar diferentes elementos, entre ellos todas las variables palanca pertenecientes al cuadro de mando con el que se va a jugar, dichas variables palanca aparecen separadas por los grupos de entrada realizados previamente por el usuario administrador a la hora de crear el cuadro de mando.

Vemos tanto el nombre asignado a la variable palanca como el nombre asignado a las unidades de medida de dicha variable. Para dar valor a una variable palanca tenemos dos opciones, podemos introducir un valor en concreto en el cuadro de texto situado a la derecha del nombre de la variable palanca o bien deslizándonos con el componente *slider* que está situado a la derecha, a medida que nos deslizamos por el componente slider vemos que el campo de texto se actualiza automáticamente.

Si se elige introducir el valor a través del campo de texto, el “slider” se actualizará automáticamente, y viceversa.

Tras dar valor a la variable palanca, tenemos la opción de *simular* o *simular todo*.

- **Simular:** Se realiza la simulación de un periodo. Vemos los valores obtenidos por las variables palanca en ese periodo de la simulación. Esta opción es preferible si queremos modificar de valor las variables palanca en cada uno de los periodos de la simulación para influir en la evolución de los resultados.
- **Simular todo:** Se realizan las simulaciones desde el periodo actual hasta el periodo final, se realizan todas las simulaciones seguidas mostrando el resultado final de la simulación.

En la parte inferior derecha vemos un botón llamado *Guardar partida* con esta opción podemos guardar la partida hasta el periodo que llevamos simulado. Aparecerá una ventana emergente en la cual deberemos introducir el nombre que queremos darle a la partida, hacemos clic en el botón *Guardar* y ésta se almacenará en la base de datos.

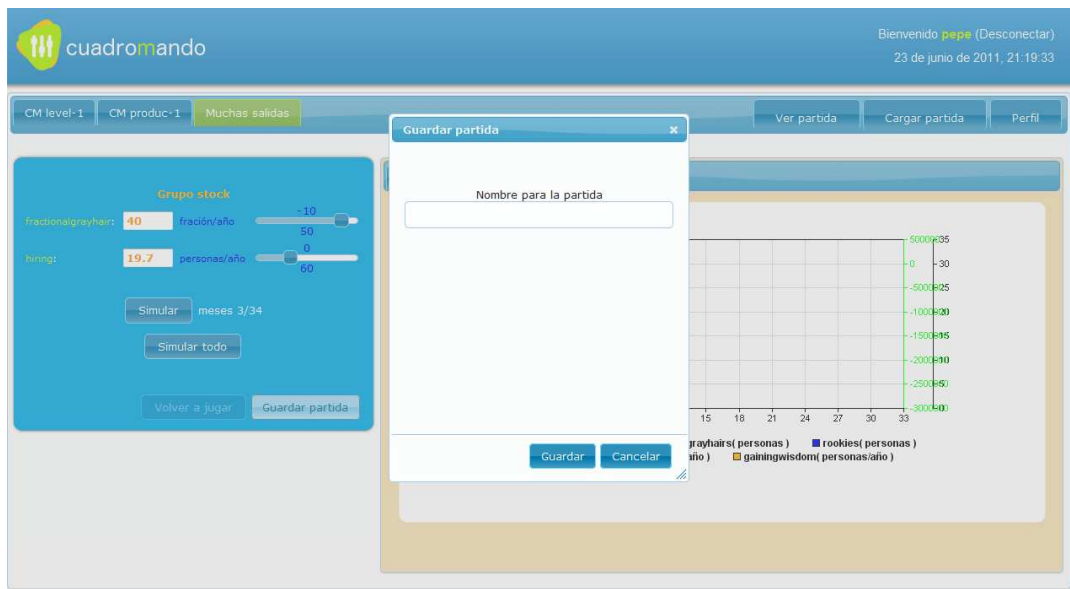


Figura 88. Guardar partida

Los resultados de la simulación se mostrarán en la parte derecha de la pantalla en distintos gráficos organizados por pestañas para cada uno de los grupos de salida. Las opciones de representación gráfica son gráfica temporal o reloj digital.



Figura 89. Representación de los resultados de la simulación

En esta parte derecha podemos visualizar también todos los grupos de salida que han sido creados previamente por el usuario administrador a la hora de crear el cuadro de mando.

Ver partida

Mediante esta opción podemos ver los resultados de una partida ya realizada anteriormente. Se visualizan los valores obtenidos tanto por las variables juego como las variables salida en cada una de las simulaciones realizadas.

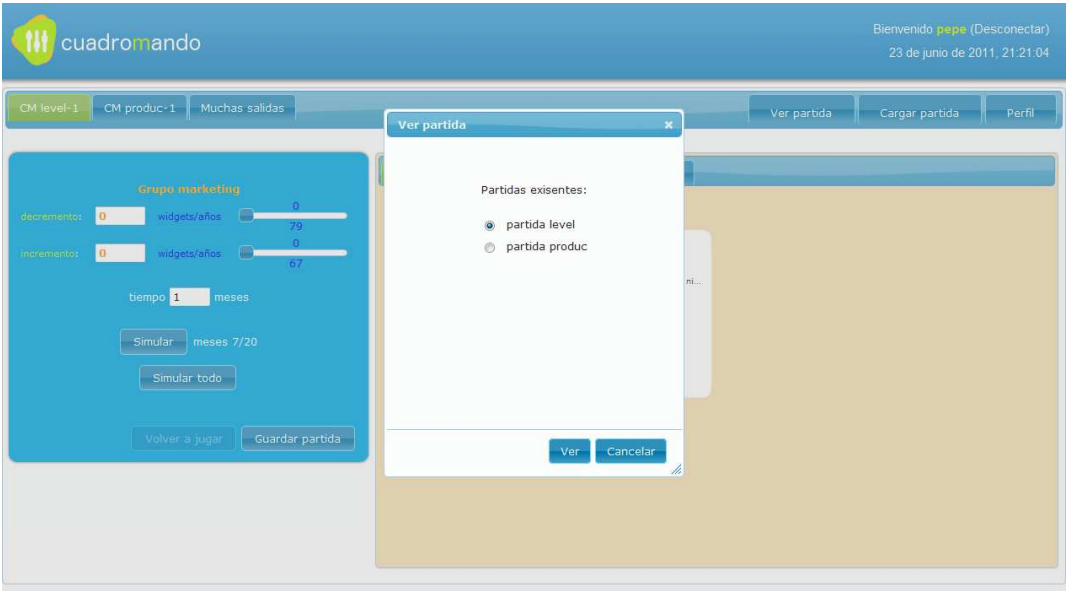


Figura 90. Ver partida



Figura 91. Ver resultados de la partida

Cargar partida

Nos permite cargar una partida guardada anteriormente y poder continuarla desde el estado como la hemos dejado.

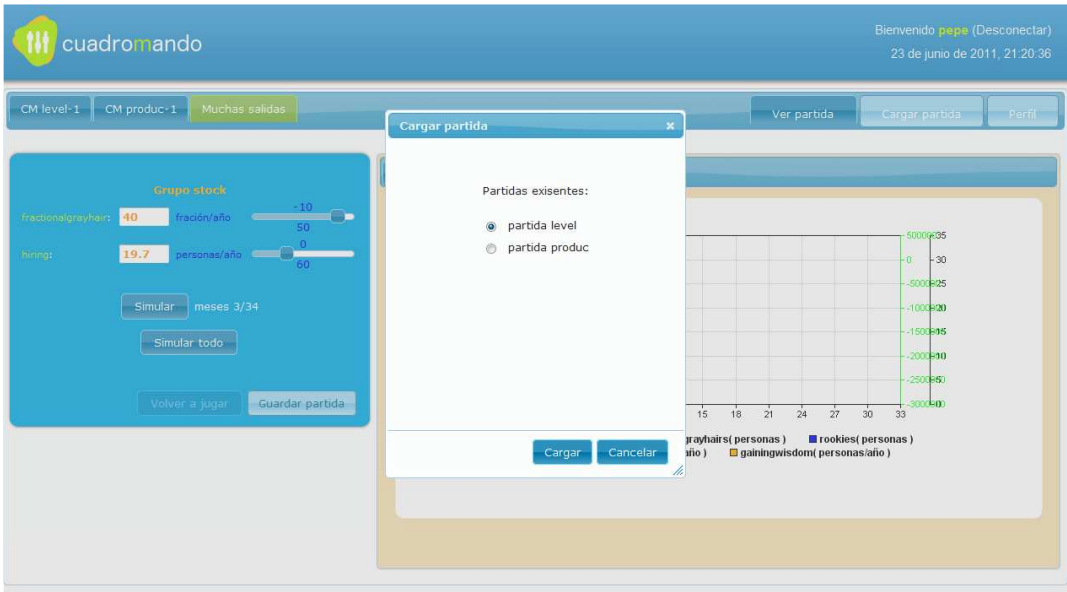


Figura 92. Cargar partida

Perfil

Esta opción permite al usuario poder modificar cualquiera de los datos relacionados con su perfil de usuario. Al hacer clic sobre esta opción aparece una nueva ventana emergente mostrando los datos del usuario que pueden ser modificados.

También existe la opción de poder borrar las partidas guardadas por el usuario, después de realizar los cambios hacemos clic en el botón *Modificar*.

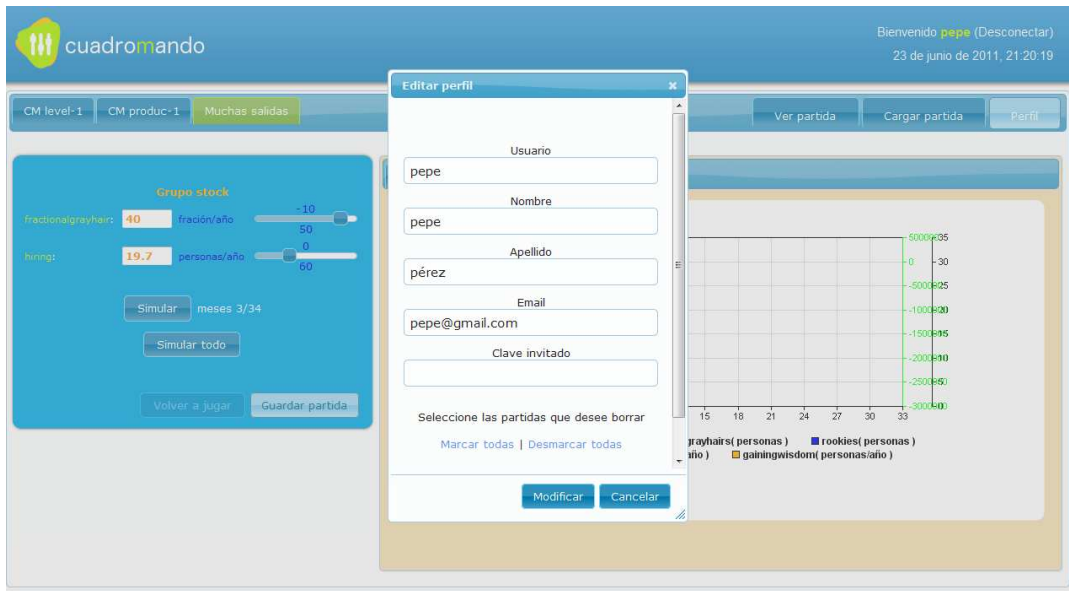


Figura 93. Editar perfil

Capítulo 6

Conclusiones

En este capítulo se presentan las conclusiones obtenidas de la realización de este proyecto, las dificultades encontradas, la consecución de objetivos marcados en un principio, los conocimientos adquiridos y propuesta de trabajos futuros.

6.1. Conclusiones

Después de la realización de este proyecto vemos que se han conseguido los objetivos fijados en un principio.

Hemos realizado una aplicación en la que se pueden realizar simulaciones con modelos diferentes los cuales a su vez van a ser la base de los cuadros de mando.

El usuario puede modificar el valor de las variables palanca o variables de entrada con el fin de obtener resultados en cada uno de los periodos de la simulación, el usuario puede ver los resultados de manera gráfica para una mejor interpretación de los mismos.

6.2. Dificultades encontradas

Uno de los objetivos fue la interacción con la aplicación Vensim. En este punto ha sido donde han surgido la mayoría de dificultades encontradas, donde hemos invertido un mayor tiempo de investigación para estudiar el funcionamiento de la aplicación Vensim, el funcionamiento de las variables y funciones que intervienen en cada uno de los modelos.

Poca documentación e información acerca de la interacción de Vensim DSS con lenguajes de programación. Hemos tenido solo documentación de la página oficial en inglés y el foro de la página oficial cuya comunicación ha sido en inglés, en la red no hay casi documentación acerca de Vensim DSS.

Problemas con la librería dll, es necesario que este la carpeta Windows / system 32. Tras investigar vimos que el problema era de MS Windows no la encontraba y de cómo estaba definida la variable de entorno *Path*. La solución es añadir a la variable *Path* la ruta donde está ubicada la librería.

A la hora de seleccionar las variables para jugar, Vensim hace una clasificación propia y no sabemos exactamente cual son las *variables palanca*. Después de investigar encontramos la solución. Las variables palanca son las variables tipo5, pero hay que convertirlas a tipo 12.

Se ha realizado una aplicación web que permite al usuario poder manejar de manera sencilla una aplicación con un grado bastante elevado de dificultad como es Vensim. Con esta herramienta hemos conseguido que el usuario disponga de múltiples opciones teniendo el control de la mayoría de objetos que intervienen en la Dinámica de Sistemas. Con esta aplicación se logra una interfaz amigable y sencilla que es una buena opción para el uso docente.

6.3. Consecución de objetivos

Se han cumplido los siguientes objetivos descritos antes en la introducción:

Uno de los objetivos principales ha sido el aprendizaje sobre el manejo de la aplicación Vensim DSS que es la encargada de realizar las simulaciones de los modelos cargados. Interactuar con dicha aplicación estudiando los datos y resultados que íbamos obteniendo.

Desarrollo de una aplicación web en la que el usuario puede manejar de manera sencilla y donde detrás hay una compleja aplicación comercial. Mediante una serie de pasos podemos realizar simulaciones con diferentes modelos de dinámica de sistemas a través de diversas pantallas.

Diseño de diferentes pantallas para la introducción de datos por parte del usuario administrador o bien para mostrar los datos de una manera sencilla y amigable al usuario en la cual le sea fácil interactuar con la aplicación a la hora de jugar una nueva partida.

Creación de una base de datos en la que fuera posible almacenar toda la información referente a la configuración de los cuadros de mando y de las partidas realizadas por los usuarios. Pensar sobre una estructura de bases de datos dinámica válida para todos los cuadros de mando ya que dependerá de la información recogida en un cuadro de mando variará dependiendo de las selecciones realizadas.

Documentación de la herramienta desarrollada, tratando todas las ampliaciones implantadas en la aplicación anterior, documentación de la instalación y posterior manejo de la aplicación.

Otro de los objetivos ha sido solventar los contratiempos en revisiones anteriores y averiguar cómo ha sido en el caso de las librerías de las que depende la aplicación Vensim se pudieran ejecutar desde ubicaciones diferentes a la que venía por defecto. Los modelos también supuestamente tenían que estar en la carpeta donde esté ubicada la librería dll.

6.4. Conocimientos adquiridos

El haber realizado este proyecto entre dos personas ha sido productivo en el sentido de que un compañero adquiere conocimientos del otro y viceversa en cada una de las fases de desarrollo del proyecto. Aprendizaje en trabajar en equipo en un proyecto de gran envergadura.

Adquirir los conocimientos necesarios para poder implementar los recursos que ofrece Vensim DSS a la aplicación web.

Aprendizaje del entorno de programación JavaScript y JSP.

Aplicación de hojas de estilo CSS

Implementación de diversos objetos JQuery.

Aprendizaje del manejo de diversas herramientas como Toad, Eclipse, MySQL,

Manejo de aplicación de control de versiones Subversion.

Manejo de servidor Apache Tomcat.

6.5. Proposición de trabajos futuros

Posibilidad de cargar modelos cuyo fichero pueda tener extensión diferente a .vpm. Y así quitar la restricción de que solo se pueda trabajar con modelos con dicha extensión.

Implementación de una pantalla en la cual se pueda visualizar de manera remota las partidas que están jugando otros usuarios. Poder así ver la toma de decisiones que están realizando otros usuarios tras ver los resultados de un periodo de simulación.

Investigar las rutas que propone Vensim por defecto cuando se realiza una simulación. Al simular con Vensim crea automáticamente un archivo donde va almacenando los datos de salida, por tanto sería interesante cambiar la ubicación donde Vensim guarda esos archivos.

A la hora de realizar una simulación se necesita crear un hilo diferente para cada una de ellas debido a que si se hacen varias simulaciones con el mismo hilo, se sobrescribe el archivo donde se guardan los datos. Sería interesante estudiar la manera que no sobrescriba los datos.

Módulo de estadísticas, más o menos detalladas, que de información sobre número de accesos, actualización de la información, filtros utilizados, actividad de los responsables, etc... para poder utilizar esa información en la continua mejora de la herramienta.

Exportación a PDF o más formatos de los resultados de una simulación, para poder ser consultados y analizados sin necesidad de estar utilizando la aplicación.

Ampliación del número de idiomas que soporta el sistema para conseguir una internacionalización más amplia y completa.

Bibliografía y referencias web

Recursos bibliográficos

- [1] Booch, Grady; Jacobson, IVar; Rumbaugh, James. (2007) *El lenguaje unificado y modelado*. Addison Wesley. ISBN: 8478290281.
Aporte al proyecto: Empleado como referencia UML a la hora de realizar la parte de análisis y diseño del proyecto.
Última fecha de consulta: 8 de Junio de 2011.

- [2] del Amo Ramos, Javier; Toledo Celada, David. (2010) *Cuadro de mando dinámico para un juego de empresa multisesión*. Proyecto Fin de Carrera. Universidad de Valladolid.
Aporte al proyecto: Empleado como referencia en la estructura de la memoria de este proyecto.
Última fecha de consulta: 27 de Junio de 2011.

- [3] Pérez Ríos, José Luis (1992) *Dirección estratégica y pensamiento sistémico*. Universidad de Valladolid, Departamento de Economía y Administración de empresas. ISBN: 8460472288.
Aporte al proyecto: Empleado como guía sobre como comenzar a abordar el proyecto, primeros pasos.
Última fecha de consulta: 15 de Febrero de 2011.

- [4] Romero, Alfonso; Lovera, Daniel; Yaringaño, Simeón; Flores, Silvana (2007) *Gestión de riesgos con CMMI, RUP e ISO en Ingeniería de Software*. Revista del Instituto de Investigaciones FIGMMG.
Aporte al proyecto: Empleado como referencia sobre cómo gestionar y tratar los riesgos en ingeniería de software.
Última fecha de consulta: 11 de Junio de 2011.

Referencias Web

- [5] Adictos al trabajo
Dirección Web: <http://www.adictosaltrabajo.com/tutoriales/tutoriales.php>
Descripción: Manuales y tutoriales sobre diferentes tecnologías de programación.
Aporte al proyecto: Gráficas en Java con JFreeChart.
Último acceso: 12 de Abril de 2011.

- [6] Apache Tomcat
Dirección Web: <http://tomcat.apache.org/>
Descripción: Documentación del servidor Web sobre el que se apoya el sistema. El sitio Web contiene todos los recursos descargables necesarios y toda la documentación de configuración y mantenimiento del servidor.
Aporte al proyecto: Recurso empleado para la configuración e instalación del servidor.
Último acceso: 25 de Octubre de 2010

- [7] Chuidiang.
Dirección Web: <http://chuwiki.chuidiang.org/index.php>
Descripción: Manuales y tutoriales sobre diferentes entornos de programación.
Aporte al proyecto: Consulta de Base de Datos desde JSP, uso de JFileChooser.
Último acceso: 20 de Febrero de 2011.

- [8] Cristalab
Dirección Web: <http://www.cristalab.com/tutoriales>
Descripción: Manuales y tutoriales sobre diferentes entornos de programación.
Aporte al proyecto: Instalar Eclipse y Tomcat para desarrollo con Java
Último acceso: 25 de Octubre de 2010.

- [9] Desarrollo Web
Dirección Web: <http://www.desarrolloweb.com/>
Descripción: Foro sobre el uso de diferentes entornos de programación.
Aporte al proyecto: Consultas sobre el entorno de desarrollo Java.
Último acceso: 5 de Mayo de 2011.

- [10] Filezilla
Dirección Web: <http://filezilla-project.org/>
Descripción: Página web oficial de Filezilla donde podemos descargar el programa.
Aporte al proyecto: Descarga del programa Filezilla, el cual es un cliente FTP.
Último acceso: 25 de Octubre de 2010

- [11] Foro de Java
Dirección Web: <http://forodejava.com/>
Descripción: Foro de consultas relacionadas con el entorno de programación Java.
Aporte al proyecto: Consulta de diferentes dudas respecto al entorno de programación JSP.
Último acceso: 12 de Abril de 2011.

- [12] Foros del Web
Dirección Web: <http://www.forosdelweb.com>
Descripción: Foro de consultas relacionadas con diversos entornos de programación.
Aporte al proyecto: Consulta de diversas dudas relacionadas con JSP, JavaScript y JQuery.
Último acceso: 10 de Junio de 2011

- [13] Galeon
Dirección Web: <http://www.galeon.com/robotk/colores.htm>
Descripción: Tabla de colores y pantones para su uso en CSS.
Aporte al proyecto: Estilos CCS para las pantallas de la aplicación
Último acceso: 4 de Mayo de 2011.

- [14] Google
Dirección Web: <http://code.google.com/intl/es-ES/apis/visualization/documentation/gallery.html>
Descripción: Gráficas de Google
Aporte al proyecto: Gráficas estadísticas.
Último acceso: 4 de Mayo de 2011.
- [15] Informática Práctica
Dirección Web: <http://informatica-practica.net/wordpress/index.php>
Descripción: Blog con diferentes tutoriales y manuales de informática.
Aporte al proyecto: Scripts para crear gráficas con Javascript, PHP, Flash, Java, ASP.
Último acceso: 12 de Abril de 2011.
- [16] Java Hispano.
Dirección Web: <http://javahispano.org>
Descripción: Foro sobre consultas relacionadas con el entorno de programación Java.
Aporte al proyecto: Consulta de diferentes dudas respecto al entorno de programación JSP.
Último acceso: 12 de Abril de 2011.
- [17] Javalangnullpointer
Dirección Web: <http://javalangnullpointer.wordpress.com>
Descripción: Página web con diversos manuales relacionados con Java.
Aporte al proyecto: Diversas técnicas para la implementación de Gráficas con Java.
Último acceso: 12 de Abril de 2011.
- [18] JQuery
Dirección Web: <http://jqueryui.com/>
Descripción: Página web oficial de JQuery UI.
Aporte al proyecto: Implementación de diversos componentes JQuery.
Último acceso: 12 de Abril de 2011.
- [19] MySQL
Dirección Web: <http://www.mysql.com/>
Descripción: Página web oficial de Mysql donde podemos descargar tanto el sistema gestor como la documentación relacionada.
Aporte al proyecto: Base de datos para alojar toda la información de la aplicación.
Último acceso: 25 de Octubre de 2010
- [20] Pencil
Dirección Web: <http://pencil.evolus.vn/en-US/Home.aspx>
Descripción: Página web oficial de Pencil donde podemos descargar el programa.
Aporte al proyecto: Descarga del programa Pencil, el cual es una herramienta para el diseño de bocetos de pantallas de la aplicación.
Último acceso: 15 de Febrero de 2011

- [21] Programación
Dirección Web: <http://www.programacion.com/>
Descripción: Foro donde se pueden consultar dudas con diferentes entornos de programación.
Aporte al proyecto: Consulta de diferentes dudas respecto al entorno de programación JSP.
Último acceso: 12 de Abril de 2011.

- [22] Project Locker
Dirección Web: <http://www.projectlocker.com/>
Descripción: Página web que ofrece alojamiento gratuito, repositorio para alojar el proyecto, control de versiones mediante Subversion.
Aporte al proyecto: Servidor de control de versiones con soporte para SVN, sistema de tickets para el seguimiento de tareas Trac.
Último acceso: 12 de Abril de 2011.

- [23] REM
Dirección Web: <http://www.lsi.us.es/descargas/descargas.php>
Descripción: Página web oficial de REM.
Aporte al proyecto: Descarga del programa REM, el cual es una herramienta para realizar el documento de requisitos y análisis del sistema.
Último acceso: 25 de Octubre de 2010

- [24] StarUML
Dirección Web: <http://staruml.sourceforge.net/en/>
Descripción: Página web oficial de StarUML donde podemos descargar tanto el programa como el manual de usuario.
Aporte al proyecto: Descarga del programa StartUML, el cual es una herramienta para el modelado de software basado en los estándares UML.
Último acceso: 25 de Octubre de 2010

- [25] Toad for MySQL
Dirección Web: <http://toadformysql.com/>
Descripción: Página web oficial de Toad for MySQL.
Aporte al proyecto: Gestión y mantenimiento de la base de datos MySQL.
Último acceso: 25 de Octubre de 2010

- [26] Vensim Help
Dirección Web: http://www.vensim.com/documentation/index.html?vensim_help.htm
Descripción: Página oficial de la aplicación Vensim
Aporte al proyecto: Consulta sobre las funciones disponibles de Vensim en el entorno de desarrollo Java.
Último acceso: 20 de Febrero de 2011.

- [27] Ventana Systems UK Forum
Dirección Web: <http://www.ventanasystems.co.uk/forum/index.php>
Descripción: Foro sobre uso y consultas de la aplicación Vensim DSS.
Aporte al proyecto: Consultas sobre el funcionamiento de la herramienta Vensim DSS.
Último acceso: 20 de Febrero de 2011.

Apéndice I

Glosario de términos

Apache Tomcat. Es un contenedor de servlets usado en la implementación de tecnologías Java Servlet y Java Server Pages (JSP). Permite la ejecución de aplicaciones web basadas en tecnologías Java, siendo uno de los más extendidos.

BBDD. Una base de datos o banco de datos es un conjunto de datos pertenecientes a un mismo contexto y almacenados sistemáticamente para su posterior uso. Una base de datos es un “almacén” que nos permite guardar grandes cantidades de información de forma organizada para que luego podamos encontrar y utilizar fácilmente.

Creador. Aplicamos este patrón para asignar la responsabilidad a una clase para crear una nueva instancia de una cierta clase.

CSS. Las hojas de estilo en cascada o CSS constituyen el estándar para la inserción de estilos (tamaños, colores, tipografías, espacios, bordes, etc.) a documentos estructurados, como por ejemplo, páginas HTML o XML. El objetivo de la definición de este estándar del W3C es permitir la separación entre las normas de presentación y el propio contenido a mostrar.

DAO. Los Objetos de Acceso a Datos son un Patrón de Diseño y considerados una buena práctica. La ventaja de usar objetos de acceso a datos es que cualquier objeto de negocio (aquel que contiene detalles específicos de operación o aplicación) no requiere conocimiento directo del destino final de la información que manipula.

Dinámica de sistemas. Es una metodología que trata de la aproximación a la modelización de la dinámica de sistemas complejos, en los que las partes demuestran propiedades interactivas entre los objetos, dando como resultado una comunicación gracias a las transacciones que se derivan de las relaciones mutuas. Es una metodología para la construcción de modelos de simulación para sistemas complejos.

Dropbox. Se trata de una herramienta de sincronización de archivos a través de un disco duro o directorio virtual. Permite disponer de un directorio de archivos de forma remota y accesible desde cualquier ordenador. Es decir, crea una carpeta en nuestro ordenador y realiza una copia a través de Internet de todos los archivos que depositemos en ella. Se ocupa de mantener la copia de nuestros archivos siempre sincronizada.

Eclipse. Es un IDE (entorno de desarrollo integrado) para Java y Java Enterprise Edition basado en Eclipse 3.6. Es principalmente una plataforma de programación, de código abierto, multiplataforma, para desarrollar aplicaciones de cliente enriquecido.

Experto. Utilizamos este patrón para asignar la responsabilidad de realizar tareas a la clase que tiene la información necesaria para cumplimentarla. Con ello conseguimos un sistema más robusto y fácil de mantener, un bajo acoplamiento y una alta cohesión.

Filezilla. Es una herramienta que permite realizar una conexión al servidor donde va a estar alojada la aplicación web.

Gráficas Google. Galería de gráficas para cubrir la necesidad de la representación de los datos almacenados. Estos gráficos se basan en la tecnología HTML5/SVG (la adopción de VML para versiones antiguas de IE). No hay necesidad de plugins para el funcionamiento de estos gráficos.

HTML. Lenguaje de Etiquetas de Hipertexto. Diseñado para estructurar textos y presentarlos en forma de hipertexto, que es el formato estándar de las páginas web. Es una sintaxis para definir los elementos que se quieren mostrar en una página, y su estructura y ubicación, así como las relaciones entre ellos y con otros elementos de la web mediante hipervínculos.

IDE (*Integrated Development Environment*). Un entorno de desarrollo integrado es un programa informático compuesto por un conjunto de herramientas de programación. Puede dedicarse en exclusiva a un sólo lenguaje de programación o bien, poder utilizarse para varios. Consiste en un editor de código, un compilador, un depurador y un constructor de interfaz gráfica (GUI). Los IDEs pueden ser aplicaciones por sí solas o pueden ser parte de aplicaciones existentes.

Java. Es un lenguaje de programación orientado a objetos y la primera plataforma informática creada por Sun Microsystems en 1995. Es la tecnología subyacente que permite el uso de programas punteros, como herramientas, juegos y aplicaciones de negocios. Tiene como principal característica ser un lenguaje independiente de la plataforma.

JavaScript. Es un lenguaje que no requiere de compilación ya que el funciona del lado del cliente, los navegadores son los encargados de interpretar estos códigos. Javascript tiene la ventaja de ser incorporado en cualquier página web, puede ser ejecutado sin la necesidad de instalar otro programa para ser visualizado.

JQuery. Es una biblioteca de componentes para el framework JQuery que proporciona un conjunto de plugins, widgets y efectos visuales basado en Javascript, obtendremos ayuda en la creación de interfaces de usuario, efectos dinámicos, aplicaciones que hacen uso de Ajax, etc.

JRE (*Java Runtime Environment*). Es un conjunto de utilidades de Java contiene la JVM.

JSP (*Java Server Pages*). Es una tecnología Java que permite crear aplicaciones web y generar contenido dinámico para web, en forma de documentos HTML, XML o de otro tipo. Las JSP's permiten la utilización de código Java mediante scripts. Además, es posible utilizar algunas acciones JSP predefinidas mediante etiquetas.

JVM. (*Java Virtual Machine*). Es la aplicación donde corren los programas hechos en Java.

MVC. Modelo Vista Controlador, es un patrón de arquitectura de software que separa los datos de la aplicación, la interfaz de usuario, y la lógica de control en tres componentes distintos. Se ve frecuentemente en aplicaciones web, donde la vista es la página HTML y el código que provee de datos dinámicos a la página. El modelo es el Sistema de Gestión de Base de Datos y la Lógica de negocio, y el controlador es el responsable de recibir los eventos de entrada desde la vista.

MySQL. MySQL es un sistema de administración de bases de datos para bases de datos relacionales. No es más que una aplicación que permite gestionar archivos llamados de bases de datos. Utilizado para almacenar todos los datos de interés de la aplicación y datos referentes de las simulaciones realizadas por los usuarios.

Properties. Se trata de un fichero de texto con propiedades del sistema. Utilizamos el fichero *properties.java* para indicar toda la información relacionada con la conexión a las bases de datos y para indicar la ruta donde están almacenados los modelos. La ventaja que nos aporta es que a la hora de modificar las rutas solo lo tenemos que hacer en un fichero.

REM. Es una herramienta de Gestión de Requisitos diseñada para soportar la fase de Ingeniería de Requisitos de un proyecto de desarrollo software de acuerdo con la metodología definida en la Ingeniería de Requisitos para Sistemas de Información.

StarUML. Es una herramienta para el modelamiento de software basado en los estándares UML (Unified Modeling Language) y MDA (Model Driven Arquitectura). Permite definir elementos propios para los diagramas, que no necesariamente tienen que pertenecer al estándar de UML.

SGBD (*Sistema Gestor de Bases de Datos*). En inglés *Data Base Management System* (DBMS); es un tipo de software muy específico, dedicado a servir de interfaz entre la base de datos y el usuario y las aplicaciones que la utilizan. Se compone de un lenguaje de definición de datos, de un lenguaje de manipulación de datos y de un lenguaje de consulta.

Singleton. Este patrón garantiza que las clases creadas en el proyecto sólo tengan una instancia y proporciona un punto de acceso global a ésta instancia.

Slider. Es un elemento de las interfaces gráficas que permite seleccionar un valor moviendo un indicador o, en algunos casos, el usuario puede hacer clic sobre algún punto del slider para cambiar hacia ese valor.

Subversión. Es sistema de control de versiones empleado en la administración de archivos utilizados en el desarrollo de software o contenido. Puede acceder al repositorio a través de redes, lo que nos permite ser usado diferentes personas que se encuentran en distintos ordenadores.

Toad for MySQL. Permite crear y ejecutar consultas rápidamente, automatizar la gestión de bases de datos de objetos, y desarrollar código SQL de manera más eficiente. Proporciona utilidades para comparar, extraer y buscar objetos, gestionar proyectos, importación y exportación de los datos, y administrar la base de datos.

TRAC. Es un sistema web libre para la gestión de proyectos y seguimiento de errores, tiene una fuerte dependencia de Subversion. Permite mediante la creación de tickets llevar un seguimiento del proyecto, los tickets tienen una descripción de la tarea que es asignada a una persona, dichos tickets tienen una duración en horas estimada.

USB (*Universal Serial Bus*). Estándar de comunicaciones serie de alta velocidad.

Vensim. Es una herramienta de modelaje que permite conceptualizar, documentar, simular, analizar y optimizar modelos de dinámica de sistemas. Vensim provee una forma simple y flexible de construir modelos de simulación, sean lazos causales o diagramas de stock y de flujo.

Apéndice II

Índice de figuras

Figura 1. Visión iterativa en UPEDU	4
Figura 2. Hito en la Fase de Inicio.....	7
Figura 3. Diagrama de Gantt de la planificación completa	10
Figura 4. Planificación Real	11
Figura 5. Modelo de administración de riesgos.....	14
Figura 6. Diagrama de Gantt de la Fase de Inicio	22
Figura 7. Tareas de la Fase de Inicio	23
Figura 8. Diagrama de Gantt para la Fase de Elaboración	26
Figura 9. Tareas para la Fase de Elaboración	27
Figura 10. Hito en la Fase de Elaboración.....	29
Figura 11. Diagrama de Casos de Uso.....	51
Figura 12. Diagrama de clases de análisis	81
Figura 13. Diagrama Entidad Relación	82
Figura 14. Diagrama de clases de diseño.....	90
Figura 15. Diagrama de secuencia - Solicitar registro.....	97
Figura 16. Diagrama de secuencia - Identificarse.....	98
Figura 17. Diagrama de secuencia - Crear sala de operaciones.....	99
Figura 18. Diagrama de secuencia - Modificar sala de operaciones.....	100
Figura 19. Diagrama de secuencia - Eliminar sala de operaciones.....	101
Figura 20. Diagrama de secuencia - Modificar usuario.....	102
Figura 21. Diagrama de secuencia - Eliminar usuario.....	103
Figura 22. Diagrama de secuencia - Cargar Modelo	104
Figura 23. Diagrama de secuencia - Crear CM - Datos Cuadro de Mando	105
Figura 24. Diagrama de secuencia - Crear CM - Crear grupo de entrada.....	106
Figura 25. Diagrama de secuencia - Crear CM - Eliminar grupo de entrada.....	106
Figura 26. Diagrama de secuencia - Crear CM - Configuración de variables juego	107
Figura 27. Diagrama de secuencia - Crear CM - Crear grupo de salida	108
Figura 28. Diagrama de secuencia - Crear CM - Eliminar grupo de salida	108
Figura 29. Diagrama de secuencia - Crear CM - Configuración de variables salida.....	110
Figura 30. Diagrama de secuencia - Modifica CM - Datos de Cuadro de Mando.....	111
Figura 31. Diagrama de secuencia - Modificar CM - Crear grupo de entrada.....	111
Figura 32. Diagrama de secuencia - Modificar CM - Eliminar grupo de entrada.....	112
Figura 33. Diagrama de secuencia - Modificar CM - Configuración de variables juego	113
Figura 34. Diagrama de secuencia - Modificar CM - Crear grupo de salida	114
Figura 35. Diagrama de secuencia - Modificar CM - Eliminar grupo de salida	114

Figura 36. Diagrama de secuencia - Modificar CM - Configuración de variables salida	116
Figura 37. Diagrama de secuencia - Eliminar Cuadro de Mando.....	117
Figura 38. Diagrama de secuencia - Simular	118
Figura 39. Diagrama de secuencia - Volver a jugar	119
Figura 40. Diagrama de secuencia - Cargar Partida	121
Figura 41. Diagrama de secuencia - Guardar Partida.....	123
Figura 42. Diagrama de secuencia - Editar perfil.....	124
Figura 43. Diagrama de secuencia - Ver Partida.....	125
Figura 44. Diagrama de estado.....	131
Figura 45. Diagrama general de navegación	136
Figura 46. Diagrama de navegación con las opciones de menú.....	137
Figura 47. Diagrama de navegación administrador.....	138
Figura 48. Diagrama de navegación usuario	139
Figura 49. Diagrama de Gantt para la Fase de Construcción	140
Figura 50. Tareas para la Fase de Construcción.....	141
Figura 51. Hito en la Fase de Construcción	143
Figura 52. Estructura de directorios de servidor Apache	144
Figura 53. Estructura de directorios de la aplicación web.....	145
Figura 54. Diagrama de Gantt para la Fase de Transición	157
Figura 55. Tareas para la Fase de Transición.....	158
Figura 56. Hito en la Fase de Transición.....	161
Figura 57. Variable de sistema JAVA_HOME	163
Figura 58. Variable de sistema CATALINA_HOME.....	164
Figura 59. Instalación Vensim Paso 1	165
Figura 60. Instalación Vensim Paso 2	166
Figura 61. Instalación Vensim Paso 3	167
Figura 62. Instalación Vensim Paso 4	168
Figura 63. Instalación Vensim Paso 5	169
Figura 64. Instalación Vensim Paso 6	170
Figura 65. Instalación Vensim Paso 7	171
Figura 66. Instalación Vensim Paso 8.....	172
Figura 67. Login Usuario	174
Figura 68. Registro Usuario	175
Figura 69. Usuario registrado satisfactoriamente.....	175
Figura 70. Pantalla Principal Administrador.....	176
Figura 71. Gestión de Cuadros de Mando.....	177
Figura 72. Cargar modelo	178
Figura 73. Nombre y descripción del cuadro de mando.....	179
Figura 74. Agrupaciones de las variables de entrada	180
Figura 75. Configuración de las variables juego	181
Figura 76. Agrupaciones de las variables de salida.....	182
Figura 77. Configuración de las variables de salida.....	183
Figura 78. Cuadro de mando creado satisfactoriamente	183
Figura 79. Cuadro de mando modificado correctamente	184
Figura 80. Eliminar cuadro de mando	185
Figura 81. Gestión de Salas de Operaciones	186
Figura 82. Modificar Sala de Operaciones	187
Figura 83. Eliminar Sala de Operaciones.....	187
Figura 84. Gestión de Usuarios	188

Figura 85. Modificar Usuario	189
Figura 86. Eliminar Usuario	189
Figura 87. Pantalla principal usuario	190
Figura 88. Guardar partida.....	191
Figura 89. Representación de los resultados de la simulación.....	192
Figura 90. Ver partida.....	193
Figura 91. Ver resultados de la partida	193
Figura 92. Cargar partida.....	194
Figura 93. Editar perfil	195
Figura 94. Árbol de ficheros del CR-ROM	217

Apéndice III

Índice de tablas

Tabla 1. Roles y responsabilidades.....	9
Tabla 2. Planificación Estimada	10
Tabla 3. Planificación Real.....	10
Tabla 4. Fases e hitos.....	12
Tabla 5. Riesgo 1 - Incertidumbre en las tareas	16
Tabla 6. Riesgo 2 - Amplitud de los objetivos.....	16
Tabla 7. Riesgo 3 - Falta de conocimiento de la tecnología	16
Tabla 8. Riesgo 4 - Mala planificación	16
Tabla 9. Riesgo 5 - Dependencia de la tecnología de simulación.....	17
Tabla 10. Riesgo 6 - Falta de concreción de algunas tareas.....	17
Tabla 11. Riesgo 7 - Integración de tecnologías	17
Tabla 12. Riesgo 8 - Problemas en el sistema de control de versiones.....	17
Tabla 13. Riesgo 9 - Problemas por la falta de experiencia.....	18
Tabla 14. Riesgo 10 - Elección de tecnologías	18
Tabla 15. Riesgo 11 - Contratiempos con la aplicación inicial.....	18
Tabla 16. Riesgo 12 - Borrado involuntario de código.....	18
Tabla 17. Riesgo 13 - Planificación inicial incorrecta	19
Tabla 18. Riesgo 14 - Periodos de ausencia de los desarrolladores.....	19
Tabla 19. Riesgo 15 - Periodos de ausencia del cliente	19
Tabla 20. Incidencia real de los riesgos	20
Tabla 21. Artefactos a producir	21
Tabla 22. OBJ-001 – Simulación con diferentes cuadros de mando	31
Tabla 23. OBJ-002 - Administración y gestión de los diversos componentes.....	32
Tabla 24. OBJ-003 - Creación de pantalla para jugar partidas	32
Tabla 25. OBJ-004 - Creación de base de datos	33
Tabla 26. OBJ-005 - Aplicación cliente – servidor	33
Tabla 27. OBJ-006 - Documentación de la herramienta.....	34
Tabla 28. OBJ-007 - Dependencia de las librerías de Vensim.....	34
Tabla 29. OBJ-008 - Seguridad en la aplicación	35
Tabla 30. OBJ-009 - Aplicación multi usuario	35
Tabla 31. IRQ-001 - Información sobre cuadros de mando.....	36
Tabla 32. IRQ-002 - Información sobre las salas de operaciones.....	36
Tabla 33. IRQ-003 - Información sobre usuarios	37
Tabla 34. IRQ-004 - Información sobre partidas.....	37
Tabla 35. FRQ-001 - Autenticación de usuarios.....	38
Tabla 36. FRQ-002 - Gestión de simulaciones	38

Tabla 37. FRQ-003 - Muestra de simulaciones	39
Tabla 38. FRQ-004 - Crear cuadro de mando	39
Tabla 39. FRQ-005 - Modificar cuadro de mando	40
Tabla 40. FRQ-006 - Eliminar cuadro de mando	40
Tabla 41. FRQ-007 - Crear sala de operaciones.....	41
Tabla 42. FRQ-008 - Modificar sala de operaciones.....	41
Tabla 43. FRQ-009 - Eliminar sala de operaciones.....	42
Tabla 44. FRQ-011 - Modificar usuario.....	42
Tabla 45. FRQ-012 - Eliminar usuario.....	43
Tabla 46. FRQ-013 - Acceso a la aplicación.....	43
Tabla 47. FRQ-014 - Conexión a Vensim.....	44
Tabla 48. FRQ-015 - Permitir ampliaciones	44
Tabla 49. NFRQ-001 - Compatibilidad navegadores web	45
Tabla 50. NFRQ-002 - Requerimientos mínimos	46
Tabla 51. NFRQ-003 - Herramientas gratuitas	46
Tabla 52. NFRQ-004 - Fecha de entrega.....	47
Tabla 53. NFRQ-005 - Licencia Vensim.....	47
Tabla 54. NFRQ-006 – Documentación de la aplicación	48
Tabla 55. NFRQ-007 – Compatible con Microsoft Windows.....	48
Tabla 56. NFRQ-008 – Hojas de estilo CSS	49
Tabla 57. ACT-001 – Usuario	49
Tabla 58. ACT-002 - Usuario no registrado.....	50
Tabla 59. ACT-003 - Usuario registrado.....	50
Tabla 60. ACT-004 - Administrador.....	50
Tabla 61. CU-001 - Solicitar registro	52
Tabla 62. CU-002 - Identificarse.....	53
Tabla 63. CU-003 - Crear sala de operaciones.....	54
Tabla 64. CU-004 - Modificar sala de operaciones.....	55
Tabla 65. CU-005 - Eliminar sala de operaciones.....	56
Tabla 66. CU-006 - Modificar usuario	57
Tabla 67. CU-007 - Eliminar usuario	58
Tabla 68. CU-008 - Cargar modelo.....	59
Tabla 69. CU-009 - Crear cuadro de mando - Datos cuadro de mando.....	60
Tabla 70. CU-010 - Crear cuadro de mando – Crear grupo de entrada	61
Tabla 71. CU-011 - Crear cuadro de mando - Eliminar grupo de entrada.....	62
Tabla 72. CU-012 - Crear cuadro de mando - Configuración de las variables juego	63
Tabla 73. CU-013 - Crear cuadro de mando – Crear grupo de salida	64
Tabla 74. CU-014 - Crear cuadro de mando – Eliminar grupo de variables de salida	65
Tabla 75. CU-015 - Crear cuadro de mando – Configuración de las variables de salida	66
Tabla 76. CU-016 - Modificar cuadro de mando – Datos cuadro mando.....	67
Tabla 77. CU-017 - Modificar cuadro de mando – Crear grupo de entrada.....	68
Tabla 78. CU-018 - Modificar cuadro de mando - Eliminar grupo de entrada.....	69
Tabla 79. CU-019 - Modificar cuadro de mando - Configuración de las variables juego.....	70
Tabla 80. CU-020 - Modificar cuadro de mando - Agrupación de las variables salida.....	71
Tabla 81. CU-021 Modificar cuadro de mando - Eliminar grupo de salida	72
Tabla 82. CU-022 - Modificar cuadro de mando - Configuración de las variables salida	73
Tabla 83. CU-023 - Eliminar cuadro de mando	74
Tabla 84. CU-024 - Simular	75
Tabla 85. CU-025 - Guardar Partida	76

Tabla 86. CU-026 - Cargar Partida	77
Tabla 87. CU-027 - Ver Partida	78
Tabla 88. CU-028 - Volver a jugar	79
Tabla 89. CU-029 - Editar Perfil	80
Tabla 90. Descripción de la clase Usuario	91
Tabla 91. Descripción de la clase Sala_operaciones	92
Tabla 92. Descripción de la clase CuadromandoE	93
Tabla 93. Descripción de la clase CuadromandoS	94
Tabla 94. Descripción de la clase Modelo	95
Tabla 95. Descripción de la clase PartidaE	95
Tabla 96. Descripción de la clase PartidaS	96
Tabla 97. Tabla Confentrada	126
Tabla 98. Tabla Confsalida	127
Tabla 99. Tabla Modelo	127
Tabla 100. Tabla PartidaE	127
Tabla 101. Tabla Partidas	128
Tabla 102. Tabla Rconfmodelo	128
Tabla 103. Tabla Rconfpartida	128
Tabla 104. Tabla Rconfsala	129
Tabla 105. Tabla Rconfusuario	129
Tabla 106. Tabla Sala	129
Tabla 107. Tabla Usuario	129
Tabla 108. Pruebas de funcionalidad	132
Tabla 109. Pruebas de interfaz de usuario	132
Tabla 110. Pruebas límite o de integridad de la base de datos	133
Tabla 111. Objetos y acciones de la pantalla cuadro de decisiones	134
Tabla 112. Objetos y acciones de la pantalla de Administración	134
Tabla 113. Estructura de directorios de la aplicación web	146
Tabla 114. CP1 - Acceder con un usuario no registrado	146
Tabla 115. CP2 - Acceder con una contraseña incorrecta	147
Tabla 116. CP3 - Intentar registrarse con una contraseña incorrecta	147
Tabla 117. CP4 - Intentar registrarse con una dirección de correo incorrecta	147
Tabla 118. CP5 - Intentar registrarse dejando algún campo del formulario vacío	148
Tabla 119. CP6 - Intentar registrarse con un nombre de usuario ya registrado	148
Tabla 120. CP7 - Cargar un modelo que no tenga extensión vpm	149
Tabla 121. CP8 - Intentar crear un cuadro de mando con campos vacíos	149
Tabla 122. CP9 - Crear un grupo de entrada/salida ya existente	149
Tabla 123. CP10 - Inserción de valores no numéricos en el rango	150
Tabla 124. CP11 - Inserción de valores no numéricos en campos numéricos	150
Tabla 125. CP12 - Creación de una sala de operaciones con clave repetida	151
Tabla 126. CP13 - Creación de una sala sin asignarle cuadros de mando	151
Tabla 127. CP14 - Modificar usuario con nombre de usuario repetido	152
Tabla 128. CP15 - Cumplimentar datos con valores que no corresponden	152
Tabla 129. CP16 - Cumplimentar datos con valores negativos	153
Tabla 130. CP17 - Simular periodo actual como último posible	153
Tabla 131. CP18 - Simular periodos restantes siendo el actual el último	154
Tabla 132. CP19 - Simular periodos restantes no siendo el actual el último	154
Tabla 133. CP20 - Consultar resultados en pestañas	155
Tabla 134. Opciones en gráficos	155

Apéndice IV

Contenido del CR-ROM

Los contenidos presentes en el CD-ROM que acompañan a este proyecto son los que se muestra en la siguiente figura.



Figura 94. Árbol de ficheros del CR-ROM

A continuación se describe brevemente los contenidos del CD-ROM más importantes a tener en cuenta

- **readme.txt:** Fichero de texto con una versión reducida de los contenidos de este apéndice.
- **Aplicaciones:** Directorio con las aplicaciones de terceros utilizadas en la realización de este proyecto.
 - **Apache Tomcat 6:** Servidor de aplicaciones web.

- Eclipse: IDE de desarrollo.
- MySQL: Sistema Gestor de Bases de Datos.
- Toad for MySQL: Aplicación para la gestión de bases de datos en MySQL.
- **Documentacion:** Directorio con los ficheros de la documentación.
 - **Memoria:** Versión electrónica en formato PDF de esta memoria.
 - **Planificacion:** Fichero con la planificación del proyecto para Microsoft Project 2003.
 - **Diagramas:** Fichero con los diagramas en UML de la memoria de este proyecto.
- **Estructura de BD:** Diferentes scripts para creación de las tablas pertenecientes a la BD utilizada para almacenar la información necesaria.
- **Instalacion:** Directorio con los ficheros necesarios para la instalación y correcto funcionamiento de la aplicación.
 - **Drivers BD – Apache:** Fichero para la conexión del servidor de aplicaciones web Apache Tomcat con la base de datos MySQL.
 - **Java:** Máquina virtual de Java, librerías necesarias para el desarrollo de aplicaciones en el entorno Java.
 - **Jar externos:** Librerías externas necesarias para el correcto funcionamiento de algunas funciones de la aplicación como por ejemplo la representación de gráficas mediante ChartDirector.