



UNIVERSIDAD DE JAÉN
Escuela Politécnica Superior (Jaén)

Proyecto Fin de Carrera

SISTEMA DE APOYO AL CONSENSO EN TDG EN CONTEXTOS HETEROGÉNEOS

Alumno: Rosa M^a Rodríguez Domínguez

Tutores: Luis Martínez López
Francisco Mata Mata

Dpto: Informática

Junio, 2008

D. Luis Martínez López y **D. Francisco Mata Mata** pertenecientes al Departamento de Informática de la Universidad de Jaén.

INFORMAN

Que la memoria titulada “Sistema de Apoyo al Consenso en TDG en Contextos Heterogéneos” ha sido realizada por **D. Rosa M^a Rodríguez Domínguez** con DNI 26239113-T bajo nuestra dirección y se presenta como memoria del Proyecto Fin de Carrera para optar al grado de Ingeniería en Informática.

Jaén, 23 de Junio de 2008

Vº Bº

Fdo: **D. Luis Martínez López**

Fdo: **D. Francisco Mata Mata**

ÍNDICE

CAPÍTULO 1 INTRODUCCIÓN.....	1
1.1 Introducción al proyecto.....	3
1.2 Propósito.....	5
1.3 Objetivos.....	5
1.4 Resumen.....	6
CAPÍTULO 2 ANTECEDENTES	7
2.1 La Toma de Decisiones (TD).....	9
2.1.1 Descripción.....	9
2.1.2 Etapas de la Toma de Decisiones.....	9
2.1.3 Criterios de clasificación.....	13
2.1.4 Esquema de Resolución de problemas de TDG.....	15
2.2 Toma de Decisiones en Grupo en Contextos Heterogéneos.....	20
2.2.1 Los problemas de toma de decisiones en grupo.....	20
2.2.2 Modelado de preferencias.....	21
2.2.2.1 Dominio numérico.....	22
2.2.2.2 Dominio intervalar.....	22
2.2.2.3 Dominio lingüístico.....	23
2.2.3 Problemas de toma de decisiones en grupo definidos en contextos heterogéneos.	26
2.3 Modelo de Sistema de Apoyo al Consenso para problemas TDG en contextos heterogéneos.	26
2.3.1 Unificación de la información.....	28
2.3.1.1 Transformación de valores numéricos definidos en $[0,1]$ en $F(S_T)$	29
2.3.1.2 Transformación de los términos lingüísticos de S en $F(S_T)$	30
2.3.1.3 Transformación de valores intervalares en $F(S_T)$	31
2.3.1.4 Resultados del proceso de unificación.....	31
2.3.2 Cálculo del grado de consenso.....	32
2.3.2.1 Interpretación del grado de consenso.....	33
2.3.3 Control del nivel de consenso.....	34
2.3.4 Generación de recomendaciones.....	35
2.3.4.1 Cálculo e interpretación de la Medida de Proximidad.....	36

2.3.4.1.1	Cálculo de la Medida de Proximidad.....	36
2.3.4.1.2	Interpretación de la Medida de Proximidad.	37
2.3.4.2	Generador guiado de recomendaciones.....	38
2.3.4.2.1	Reglas de identificación.	38
2.3.4.2.2	Reglas de dirección.....	40
CAPÍTULO 3	PROYECTO	43
3.1	Descripción.....	45
3.2	Especificación de requerimientos.	46
3.2.1	Requerimientos funcionales.....	47
3.2.2	Requerimientos no funcionales.....	50
3.3	Análisis del sistema.	54
3.3.1	Perfil de usuario.....	54
3.3.2	Casos de uso.....	56
3.3.3	Escenarios.....	73
3.4	Diseño del sistema.....	84
3.4.1	Diagrama de clases.....	85
3.4.1.1	Diagrama completo de clases.....	86
3.4.1.2	Diagrama por paquetes.....	89
3.4.1.2.1	Paquete BBDD.....	90
3.4.1.2.2	Paquete Modelo.....	91
3.4.1.2.3	Paquete Heterogeneous Information.....	92
3.4.1.2.4	Paquete Controlador.....	93
3.4.2	Diseño de los datos.....	94
3.4.2.1	Esquema Conceptual.....	98
3.4.2.2	Esquema Conceptual Modificado.....	101
3.4.2.3	Tablas de aplicación.....	105
3.4.3	Diseño de la interfaz.....	112
3.4.3.1	Definir estilo.....	113
3.4.3.2	Metáforas.....	114
3.4.3.3	Pantallas.....	115
3.4.3.4	Caminos de navegación.....	133
3.4.3.5	Mensajes de error.....	139

3.4.3.6 Mensajes de éxito.....	142
3.4.3.7 Mensajes de confirmación.....	143
3.5 Implementación.....	144
3.5.1 Tipo de la arquitectura de la aplicación.....	144
3.5.2 Lenguajes de programación utilizados.....	145
3.5.3 Herramienta de desarrollo.....	147
3.5.4 Instalación en el servidor y funcionamiento.....	147
3.6 Implantación y pruebas.....	147
3.6.1 Pruebas y validación.....	148
3.6.1.1 Casos de Test.....	148
3.6.1.2 Resultados obtenidos.....	167
CAPÍTULO 4 CONCLUSIONES.....	173
4.1 Conclusiones finales.....	175
BIBLIOGRAFÍA.....	177
ANEXO I MANUAL DE INSTALACIÓN DEL SERVIDOR.....	183
BIBLIOGRAFÍA ANEXO I.....	205
ANEXO II MANUAL DE USUARIO DEL ADMINISTRADOR.....	207
ANEXO III MANUAL DE USUARIO DEL EXPERTO.....	243

CAPITULO 1

INTRODUCCIÓN

1.1 Introducción al proyecto.

La **toma de decisiones** (TD) es el proceso mediante el cual se realiza una elección entre distintas alternativas, de la que mejor resuelve una situación determinada en el mundo real. Estas situaciones se pueden presentar en diferentes contextos: a nivel laboral, familiar, político, etc, es decir, en todo momento se toman decisiones. La diferencia para elegir la alternativa solución dependerá de la estructura y proceso utilizado en cada caso.

Dependiendo de la estructura del problema, los problemas TD se clasifican de distintas formas. El interés de nuestro proyecto se centra en la toma de decisiones en grupo.

La **toma de decisiones en grupo** (TDG) se define como situaciones de decisión donde dos o más expertos intentan alcanzar una solución común a un problema dado, teniendo en cuenta sus preferencias.

Un problema de toma de decisiones en grupo se caracteriza por:

- i. La existencia de un problema que hay que resolver y un conjunto finito de posibles alternativas solución a dicho problema, $X = \{x_1, x_2, \dots, x_n\}$.
- ii. La existencia de un conjunto de expertos, $E = \{e_1, e_2, \dots, e_n\}$, con diferente grado de percepción del problema y que expresan sus opiniones o preferencias sobre el conjunto de alternativas.
- iii. Un objetivo, que no es otro que encontrar una solución común al problema planteado.

En ocasiones los expertos usan el mismo dominio de información para expresar sus preferencias. Sin embargo, en otras ocasiones puede resultar más aconsejable que los expertos expresen sus opiniones mediante diferentes dominios de expresión, más acordes con su área de conocimiento o investigación. Por ejemplo, expertos en un problema de TDG que trabajan en diferentes departamentos (contabilidad, psicología, marketing,...) pueden preferir dar sus opiniones usando un dominio de información más cercano a su campo de conocimiento. Además en un problema

de toma de decisiones podemos tratar con alternativas cuya naturaleza sea cuantitativa y permita valoraciones precisas como los números reales y otras alternativas cuya naturaleza sea cualitativa, difícil de valorar mediante números, por lo que es más conveniente usar intervalos numéricos o términos lingüísticos. En esta situación podemos decir que el problema de toma de decisión está definido en un contexto heterogéneo.

Un proceso de resolución de un problema de toma de decisión en grupo, de forma general se compone de dos etapas:

- Un proceso de consenso.
- Un proceso de selección.

Normalmente los problemas de TDG han sido resueltos aplicando sólo procesos de selección, donde la solución se obtiene directamente a partir de preferencias de los expertos. Sin embargo, puede que algunos expertos consideren que sus preferencias no han sido tenidas en cuenta para obtener la solución, y por tanto pueden estar en desacuerdo con ella. Para evitar esta situación, es recomendable llevar a cabo un proceso de consenso donde los expertos discuten y cambian opiniones sobre sus preferencias para alcanzar un mínimo acuerdo antes de realizar el proceso de selección que obtenga la solución del problema.

El consenso se define como el estado de mutuo acuerdo entre los miembros de un grupo donde todas las opiniones han sido expresadas y escuchadas por todos los miembros del grupo. El proceso de consenso es un proceso dinámico e iterativo compuesto de varias rondas donde los expertos expresan y discuten sus preferencias. Tradicionalmente este proceso está guiado por un moderador, que calcula el acuerdo entre los expertos en cada ronda. Si el acuerdo no es suficiente, el moderador recomienda a los expertos que cambien sus preferencias más alejadas de la opinión del grupo, para conseguir un acercamiento en la siguiente ronda de consenso.

El proceso de consenso termina cuando se alcanza un umbral de consenso fijado inicialmente o cuando se realizan un determinado número de rondas.

El proceso de selección se realiza una vez alcanzado una posición común entre los expertos o agotado el tiempo asignado al período de consenso. En él se seleccionará la alternativa o conjunto de alternativas que dan la mejor solución al problema.

En este Proyecto Fin de Carrera vamos a diseñar e implementar un Sistema de Apoyo al Consenso que automatice el Proceso de Consenso en un problema de Toma de Decisiones en Grupo en contextos heterogéneos. Este sistema incluye un generador de recomendaciones que asume el papel del moderador y recomienda cambios en las preferencias de los expertos con el fin de alcanzar un alto grado de consenso.

En un Sistema de Apoyo al Consenso se maneja distinta información acerca de los problemas, tales como: parámetros del problema, alternativas solución, expertos asignados al problema, preferencias de cada uno en las distintas rondas, etc...Por ello vamos a utilizar una Base de Datos MySQL para gestionar toda esta información.

La implementación del Modelo de Consenso la vamos a llevar a cabo utilizando el lenguaje de programación JAVA. La aplicación estará basada en el modelo cliente-servidor, por lo que tendremos que emplear Servlet y JSP para el tratamiento de la información entre el servidor y los distintos usuarios del sistema.

1.2 Propósito.

La principal tarea del proyecto es diseñar e implementar un sistema basado en técnicas soft computing, que automatice el Proceso de Consenso en Toma de Decisiones en Grupo en Contextos Heterogéneos.

1.3 Objetivos.

- Búsqueda y revisión bibliográfica.
- Automatización de un proceso de consenso en TDG con información heterogénea.
- Implementación de dicho sistema.
- Verificación y evaluación del sistema.
- Completar la memoria.

1.4 Resumen.

Los objetivos planteados en esta memoria se desarrollan a lo largo de la misma como sigue:

Para conseguir los dos primeros objetivos presentamos en el segundo capítulo con más detalle lo que es un problema de toma de decisión, las etapas de las que consta y una clasificación según el contexto. También explicaremos los problemas de toma de decisión en grupo, los distintos modelados de preferencias y los problemas de toma de decisión en grupo en contextos heterogéneos. Además, en este capítulo describiremos cada una de las fases del *Modelo de Sistema de Apoyo al Consenso en TDG en Contextos Heterogéneos*.

En el tercer capítulo queda plasmado el tercer y cuarto objetivo, ya que, describiremos el proyecto en sí y veremos que se trata de un proyecto de desarrollo software y, como tal, deben seguirse para su desarrollo las actividades de la Ingeniería del Software, como son: la especificación de requerimientos funcionales y no funcionales, el análisis y diseño del sistema. Una vez que tengamos todo esto claro y bien documentado, se implementará el Sistema de Ayuda a la Decisión en TDG en Contextos Heterogéneos y su posterior validación.

En el cuarto capítulo, se expondrán las conclusiones obtenidas durante el desarrollo de este PFC.

Incluiremos un apartado de bibliografía que enumerará las referencias bibliográficas utilizadas en este proyecto.

Por último, la memoria concluye con tres anexos que son:

- Un manual de instalación del servidor.
- Un manual del usuario Administrador del sistema.
- Un manual del usuario Experto

CAPÍTULO 2

ANTECEDENTES

2.1 La Toma de Decisiones (TD).

En los siguientes apartados daremos una visión detallada de lo que es la Toma de Decisión, las etapas que la componen y los diversos tipos en los que se clasifica.

2.1.1 Descripción.

La Toma de Decisiones es sin duda alguna, una de las tareas más importantes de los seres humanos, ya que constantemente nos estamos enfrentando a situaciones en las que existen varias alternativas y, al menos en algunas ocasiones, tenemos que decidir cuál es mejor, o cuál llevar a cabo.

La Toma de Decisiones es un área que se aplica en distintas disciplinas, tales como, las Ciencias Sociales, la Economía, la Ingeniería, etc. Esta amplia gama de campos de aplicación tiene como consecuencia la existencia de diferentes modelos de Toma de Decisión, encontrados bajo la denominada Teoría de Decisión.

2.1.2 Etapas de la Toma de Decisiones.

La Toma De Decisión está formada por las etapas que se enumeran a continuación:

1. Identificación y diagnóstico del problema
2. Generación de soluciones alternativas
3. Evaluación de alternativas
4. Selección de la mejor alternativa
5. Implementación de la decisión
6. Evaluación de la decisión

1. Identificación del problema:

En esta fase identificamos el problema que deseamos resolver.

2. Generación de soluciones alternativas:

Esta etapa consiste en identificar y desplegar el conjunto de todas las alternativas posibles de solución al problema.

3. Evaluación de alternativas:

La tercera etapa es aquella en la que se analizan cada una de las alternativas identificadas en la fase anterior. Se evalúa de manera crítica cada una de ellas, con el fin de obtener una serie de ventajas e inconvenientes.

La evaluación de una alternativa, se puede enfocar desde un punto de vista cuantitativo o cualitativo:

a) FACTORES CUANTITATIVOS

Son factores que se pueden medir en términos numéricos, como es el tiempo, o los diversos costos fijos o de operación.

b) FACTORES CUALITATIVOS

Son difíciles de medir numéricamente. Como la calidad de las relaciones de trabajo, el riesgo del cambio tecnológico o el clima político internacional.

Para evaluar y comparar los factores se debe reconocer el problema y luego analizar qué factor se aplica ya sea cuantitativo o cualitativo o ambos, clasificar los términos de importancia, comparar su probable influencia sobre el resultado y tomar una decisión.

4. Selección de la mejor alternativa:

Cuando el experto ha considerado las posibles consecuencias de sus opciones, ya está en condiciones de tomar la decisión. Se pueden considerar tres objetivos diferentes: maximizar, satisfacer y optimizar.

- Maximizar: es tomar la mejor decisión posible
- Satisfacer: es la elección de la primera opción que sea mínimamente aceptable o adecuada, y de esta forma se satisface una meta o criterio buscado.
- Optimizar: Es el mejor equilibrio posible entre distintas metas.

5. Implementación de la decisión:

El proceso no finaliza cuando la decisión se toma; esta debe ser implementada. Bien puede ser que quienes participen en la toma de una decisión sean quienes procedan a implementarla, o como en otras ocasiones delegan dicha responsabilidad en otras personas. Debe existir la comprensión total sobre la elección de la toma de decisión en sí, las razones que la motivan y sobre todo debe existir el compromiso de su implementación exitosa. Para tal fin, las personas que participan en esta fase del proceso, deberían estar involucradas desde las primeras etapas que anteriormente se han mencionado.

Podemos estar seguros de que cuando una decisión es tomada, ésta probablemente generará ciertos problemas durante su ejecución, por lo tanto los expertos deben dedicar el tiempo suficiente al reconocimiento de los inconvenientes que se pueden presentar así como también ver la oportunidad potencial que estos pueden representar.

6. Evaluación de la decisión:

"Evaluar la decisión", forma parte de la etapa final de este proceso. Se recopila toda la información sobre los resultados obtenidos, es decir, es un proceso de retroalimentación que puede ser positivo o negativo. Si la retroalimentación es positiva, entonces nos indica que podemos continuar sin problemas y que incluso se podría aplicar

la misma decisión a otras áreas de la organización. Si por el contrario, la retroalimentación es negativa, podría ser que:

- Tal vez la implementación requiera de más tiempo, recursos, esfuerzos o pensamiento
- Nos indique que la decisión fue equivocada, para lo cual debemos volver al principio del proceso (re)definición del problema. Si esto ocurriera, sin duda tendríamos más información y probablemente más sugerencias que nos ayudarían a evitar los errores cometidos en el primer intento.

El arte de tomar decisiones está basado en cinco ingredientes básicos:

a) Información:

Ésta se recoge tanto para los aspectos que están a favor como en contra de la alternativa para resolver el problema, con el fin de definir sus limitaciones. Sin embargo si la información no puede obtenerse, la decisión entonces debe basarse en los datos disponibles, los cuales caen en la categoría de información general.

b) Conocimientos:

Si quien toma la decisión tiene conocimientos, ya sea de las circunstancias que rodean el problema o de una situación similar, entonces estos pueden utilizarse para seleccionar un curso de acción favorable. En caso de carecer de conocimientos, es necesario buscar consejo en quienes están informados.

c) Experiencia:

Cuando un individuo aporta información para solucionar un problema en forma particular, ya sea con resultados buenos o malos, esta experiencia le proporciona información para la solución del próximo problema similar. Si ha encontrado una solución aceptable, con mayor razón tenderá a repetirla cuando surja un problema parecido. (Si carecemos de

experiencia entonces tendremos que experimentar; pero sólo en el caso en que las consecuencias de un mal experimento no sean desastrosas. Por lo tanto, los problemas más importantes no pueden solucionarse con experimentos.)

d) Análisis:

No puede hablarse de un método en particular para analizar un problema. En ausencia de un método para analizar matemáticamente un problema es posible estudiarlo con otros métodos diferentes. Si estos otros métodos también fallan, entonces debe confiarse en la intuición, ya que si los otros ingredientes de la toma de decisiones no señalan un camino que tomar, entonces ésta es la única opción disponible.

e) Juicio:

El juicio es necesario para combinar la información, los conocimientos, la experiencia y el análisis, con el fin de seleccionar el curso de acción apropiado.

2.1.3 Criterios de clasificación.

Los problemas de Toma de Decisión se pueden clasificar según diferentes criterios, nosotros los vamos a clasificar según el ambiente en el que se toma la decisión.

Las situaciones, ambientes o contextos en los cuales se toman las decisiones, se pueden clasificar según el conocimiento y control que se tenga sobre las variables que intervienen o influyen en el problema, ya que la decisión final o la solución que se tome va a estar condicionada por dichas variables.

Dependiendo de esto tenemos tres tipos de problemas:

- Ambiente de certidumbre
- Ambiente de riesgo
- Ambiente de incertidumbre

Ambiente de certidumbre

Un problema de decisión está definido en un ambiente de certidumbre cuando son conocidos con exactitud todos los elementos y/o factores que intervienen en el problema. Esta situación permite asignar valores precisos de utilidad a cada una de las alternativas presentes en el problema.

Como ejemplo, supongamos que disponemos de una determinada cantidad de dinero que queremos invertir en alguno de los diferentes productos financieros del mercado que nos garantice la inversión realizada (ej., imposición a plazo fijo). Asumiendo que conocemos con exactitud la rentabilidad de cada producto, los gastos de gestión, la duración del mismo, deberemos decidir en que producto invertir para maximizar la inversión realizada. En este caso conocemos todos los factores que se han de tener en cuenta para la toma de decisión y el problema consistirá en estructurar correctamente esta información y establecer las preferencias entre las alternativas de forma que nos permita elegir aquella que maximice el beneficio esperado.

Ambiente de riesgo

Un problema de decisión está definido en un ambiente de riesgo cuando alguno de los elementos o factores que intervienen están sujetos a las leyes del azar. En estos casos estos problemas son resueltos utilizando la Teoría de la Probabilidad.

Continuando con el mismo ejemplo, si la inversión la queremos realizar en la bolsa, inmediatamente surgen dudas sobre una posible subida o bajada en la cotización de las acciones en las que se invierta el dinero. En este caso el enfoque del problema ha de ser diferente y se utilizará una distribución de probabilidad para reflejar la posible subida o bajada de la bolsa que influirá en la utilidad de cada una de las posibles alternativas en las que invertir el dinero.

Ambiente de incertidumbre

Un problema de decisión está definido en un ambiente de incertidumbre cuando la información disponible sobre las distintas alternativas puede ser incompleta, vaga o imprecisa, lo que

implica que la utilidad asignada a cada alternativa tenga que ser valorada de forma aproximada. Esta incertidumbre surge a raíz del intento de modelar la imprecisión propia del comportamiento humano o la inherente a ciertos fenómenos que por su naturaleza son inciertos.

Los métodos clásicos no son adecuados para tratar situaciones en los que la incertidumbre se debe a la aparición de información vaga e imprecisa como por ejemplo la que puede surgir al intentar valorar fenómenos relacionados con apreciaciones sensoriales y subjetivas de los expertos. Esto ha generado la necesidad de recurrir a la definición de nuevos modelos basados en la Teoría de los Conjuntos Difusos para modelar la incertidumbre.

2.1.4 Esquema de Resolución de problemas de TDG.

Tomar decisiones en grupo, como su propio nombre indica, implica la participación de varias personas que han de tomar decisiones de forma colectiva para alcanzar una solución común a un problema.

En el ámbito empresarial, las decisiones de mayor relevancia han sido tomadas normalmente por los directivos pertenecientes a los niveles intermedios y/o superiores de la estructura de la organización [2,20] (Figura 2.1). Tradicionalmente estas decisiones han sido propuestas a título individual por un único individuo que se ha encargado de evaluar las diferentes alternativas según sus propios criterios, conocimientos e intuiciones. Sin embargo, en la actualidad esta visión individualizada de la toma de decisión está cambiando hacia otra visión más moderna en la que las decisiones son tomadas colectivamente por un grupo de individuos. Estos individuos pueden poseer puntos de vista diferentes e información relevante sobre el problema que han de tenerse en cuenta para obtener la solución final al mismo. Un proceso de toma de decisión en el que participen varios individuos o expertos, cada uno de ellos aportando sus propios conocimientos, experiencia y creatividad, dará como resultado una decisión de mayor calidad que aquella aportada por un solo experto [45,48].

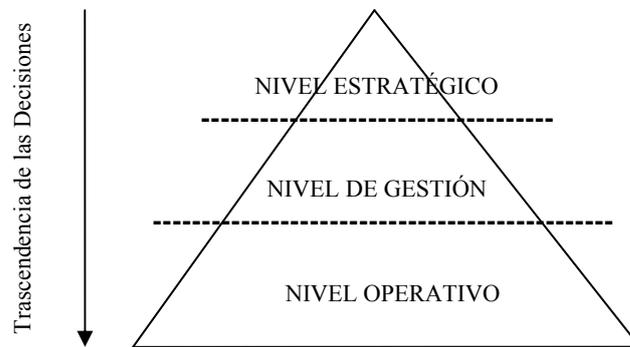


Figura 2.1: Niveles de decisión.

Un problema de TDG se caracteriza por:

1. La existencia de un problema o cuestión a resolver.
2. Un conjunto de posibles alternativas entre las que escoger.

$$X = \{x_1, x_2, \dots, x_n\} \quad (n \geq 2)$$

3. Un conjunto de individuos (jueces, expertos,...) que expresan sus juicios, opiniones o preferencias sobre el conjunto de alternativas y que tienen la intención de alcanzar una solución en común al problema planteado.

$$E = \{e_1, e_2, \dots, e_m\} \quad (m \geq 2)$$

Cuando el problema de TDG se lleva a cabo en un ambiente de incertidumbre en el que los expertos trabajan con información vaga e imprecisa, hablaremos de problemas de TDG definidos en ambientes difusos.

En la literatura podemos encontrar muchos trabajos donde se aborda la resolución de problemas de TDG en ambientes difusos aplicando procesos de selección de alternativas [14, 22, 27, 39, 42, 52, 61]. Este proceso consiste en obtener la alternativa o conjunto de alternativas que mejor resuelve el problema de decisión planteado a partir de las preferencias expresadas por el grupo de expertos.

Un inconveniente que puede aparecer en los problemas de TDG resueltos llevando a cabo exclusivamente el proceso de selección consiste en que una parte de los expertos considere que sus

opiniones no han sido tenidas en cuenta a la hora de obtener la solución y por lo tanto la rechacen. Para evitar esta situación no deseable, parece lógico plantearse la posibilidad de llevar a cabo un proceso de consenso previo al proceso de selección. En este proceso los expertos expresan y discuten sobre sus opiniones e intentan aproximarlas con el propósito de alcanzar un nivel de acuerdo alto antes de tomar una decisión (Figura 2.2).

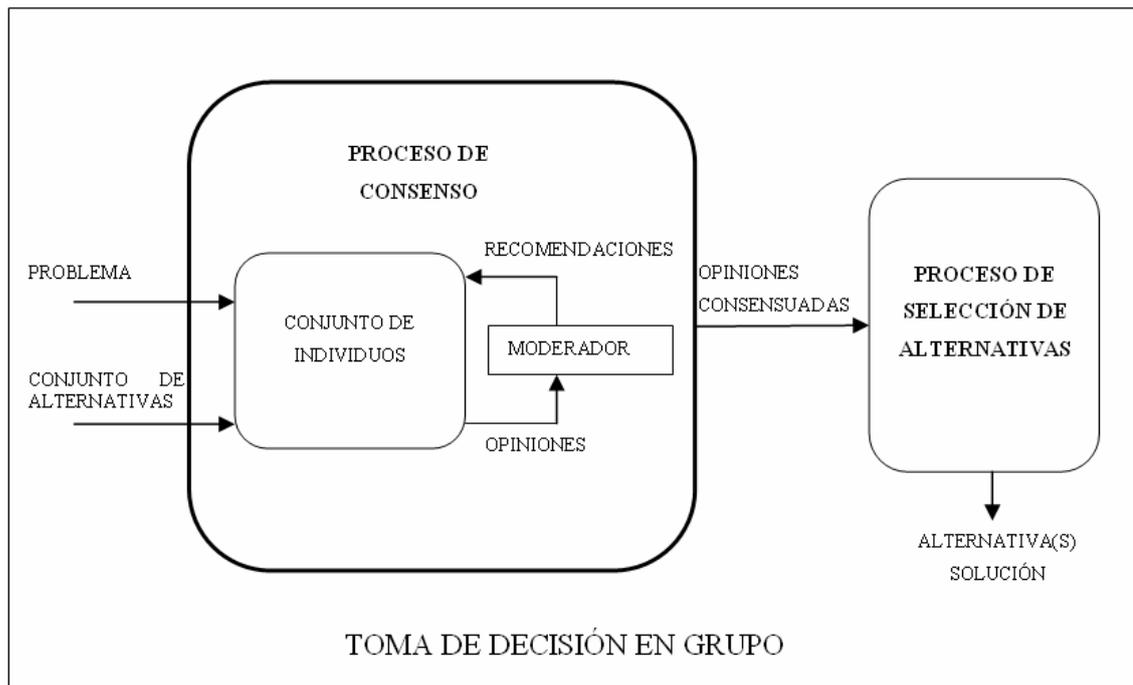


Figura 2.2: Proceso de consenso y selección en problemas de TDG.

A continuación se describen brevemente ambos procesos.

- **Proceso de consenso**

El proceso de consenso consiste en un proceso de discusión entre el grupo de expertos con el objeto de acercar sus preferencias y alcanzar un grado de consenso mínimo antes de pasar a seleccionar el conjunto de alternativas solución a un problema de TDG. Está compuesto por varias rondas de consenso donde los expertos expresan sus preferencias sobre el conjunto de alternativas. En cada una de estas rondas los expertos discuten y justifican sus preferencias. Como consecuencia de esa discusión, los expertos cambian sus preferencias siguiendo las recomendaciones de una persona que

hace la función de moderador. El propósito de estos cambios es aproximar las preferencias y de este modo aumentar el nivel de acuerdo en la siguiente ronda de consenso.

- **Proceso de selección**

El proceso de selección de alternativas consiste en la elección de la mejor o mejores alternativas que resuelvan el problema planteado.



Figura 2.3: Fases del proceso de selección en problemas de TDG

En los problemas de TDG, tradicionalmente el proceso de selección ha estado compuesto por las dos fases que se muestran en la Figura 2.3 [23, 47]

- a) *Agregación*: La fase de agregación consiste en la combinación de las preferencias individuales de los expertos con el propósito de obtener una única preferencia colectiva que refleje de forma resumida las preferencias aportadas por el conjunto de expertos. Esta combinación de preferencias se realiza utilizando operadores de agregación.

El problema de la agregación de información ha sido ampliamente estudiado en la literatura, existiendo una gran cantidad de publicaciones al respecto [4, 5, 9, 12, 18, 19, 53, 54]. Para llevar a cabo la operación de agregación es necesario definir una función de

agregación encargada de transformar todas las preferencias individuales en una única preferencia colectiva.

Las distintas formas de llevar a cabo la combinación de las preferencias han originado que muchos autores se hayan dedicado al estudio y diseño de operadores de agregación de información, entre los que cabe destacar:

1. Agregación de información cuantitativa [18, 19, 53, 54]
 2. Agregación de información cualitativa [10, 16, 24, 28, 32, 50, 51, 55, 56, 57]
- b) *Explotación*: En la fase de explotación se selecciona la/s mejor/es alternativa/s para resolver el problema de decisión a partir de la preferencia colectiva obtenida en la fase de agregación.

Para llevar a cabo esta fase es necesario definir un criterio de selección que permita establecer un orden entre el conjunto de alternativas al problema. El procedimiento que normalmente se sigue es la utilización de una función de selección que asigna un grado de selección a cada una de las alternativas. El grado de selección establece un orden de preferencia entre el conjunto de alternativas. Se utilizan funciones de selección que permiten medir la intensidad del grado de selección en cada alternativa. Aquellas alternativas con mayor intensidad son las que constituyen el conjunto de alternativas solución al problema de decisión. Las siguientes dos funciones de selección son las más utilizadas [3, 25, 43, 46]:

1. *Función de Dominancia*. Mide e indica el grado en que una alternativa es preferida o domina al resto de alternativas.
2. *Función de No-dominancia*. Mide e indica el grado en que una alternativa no es dominada por alguna del resto de alternativas.

2.2 Toma de Decisiones en Grupo en Contextos Heterogéneos.

Empezaremos esta sección introduciendo los problemas de Toma de Decisiones en Grupo (TDG) con información numérica. Después continuaremos revisando brevemente diferentes enfoques propuestos en la literatura para expresar las preferencias de los expertos, abordando el problema de la unificación de la información, para finalizar describiendo las características de los problemas de TDG heterogéneos.

2.2.1 Los problemas de toma de decisiones en grupo.

Los problemas de TDG son situaciones de decisión en las que dos o más individuos o expertos, $E = \{e_1, e_2, \dots, e_m\}$ ($m \geq 2$), expresan sus preferencias sobre un conjunto de alternativas, $X = \{x_1, x_2, \dots, x_n\}$ ($n \geq 2$), para obtener una solución (una alternativa o conjunto de alternativas). Dependiendo de la naturaleza o del conocimiento sobre las alternativas, los expertos pueden expresar sus preferencias usando diferentes enfoques. Las preferencias de los expertos se expresan usualmente por medio de relaciones de preferencia numéricas [35]. Una relación de preferencia puede definirse como una matriz $P_{e_i} \subset X \times X$,

$$P_{e_i} = \begin{pmatrix} p_i^{11} & \cdots & p_i^{1n} \\ \vdots & \ddots & \vdots \\ p_i^{n1} & \cdots & p_i^{nn} \end{pmatrix}$$

donde el valor $\mu_{p_{e_i}} = p_i^{lk}$ se interpreta como el grado de preferencia de la alternativa x_l sobre x_k expresada por el experto e_i .

Supongamos que $p^{lj} \in [0,1]$, entonces:

- $p^{lj} = 1$ indica el máximo grado de preferencia de x_l sobre x_j

- $0.5 \leq p^{ij} \leq 1$ indica una preferencia definitiva de x_i sobre x_j
- $p^{ij} = 0.5$ indica la indiferencia entre x_i y x_j

2.2.2 Modelado de preferencias.

El modelado de preferencias es una de las actividades inevitables en los problemas de TDG, independientemente del área en el que se esté trabajando. Los expertos en base a su conocimiento, experiencias y creencias han de emitir sus valoraciones sobre el conjunto de alternativas y establecer un orden de preferencia sobre la idoneidad de cada una de ellas como solución al problema.

En los problemas de decisión entendemos por dominio de expresión de preferencias el dominio de información utilizado por los expertos para expresar sus preferencias. En la literatura podemos encontrar problemas de toma de decisión en los que todos los expertos expresan sus preferencias en el mismo dominio de información, hablándose de problemas definidos en contextos homogéneos [7, 10, 15, 21, 26, 27, 38, 39, 41, 42, 44, 52], o bien problemas en los que los expertos utilizan dominios de información diferentes, conocidos como problemas definidos en contextos heterogéneos [14, 22, 29, 30, 61].

En este tipo de problemas definidos en contextos heterogéneos, los expertos pueden decidir sobre la utilización de modelos de representación de preferencias que les resulten cercanos a sus disciplinas o campos de trabajo. Por ejemplo, expertos que pertenecen a áreas técnicas se pueden sentir cómodos representando sus preferencias mediante valores numéricos. Sin embargo, expertos que pertenecen a otro tipo de disciplinas menos técnicas (Historia, Sociología,...), o cuando la información disponible es demasiado imprecisa, pueden preferir expresar sus preferencias utilizando expresiones más cercanas al lenguaje humano, tales como palabras o etiquetas lingüísticas. También puede ocurrir que, incluso sintiéndose cómodos con valores numéricos, los expertos tengan que valorar alternativas sobre las que no tienen un conocimiento lo suficientemente preciso para asignarles valores numéricos exactos, pudiendo preferir entonces el uso de valores intervalares.

Adaptar el modelado de preferencias al contexto en el que se desarrolla el problema de decisión consigue que los expertos se sientan más seguros a la hora de valorar sus preferencias, y por tanto que la solución final tenga mayor garantía de éxito [17].

En la literatura [14, 21, 30, 38, 61] encontramos que los expertos utilizan principalmente tres tipos de dominios de información para expresar sus preferencias.

- Dominio Numérico.
- Dominio Intervalar.
- Dominio Lingüístico.

En los siguientes apartados se presentan ejemplos, características y una breve justificación de las circunstancias en las que es adecuado utilizar un dominio u otro.

2.2.2.1 Dominio numérico.

El uso del dominio numérico para modelar las preferencias implica que los expertos expresen sus preferencias mediante valores numéricos precisos.

Una relación de preferencia numérica R sobre X se define como un subconjunto difuso del producto cartesiano $X \times X$, es decir, $R : X \times X \rightarrow [0,1]$. El valor $R(x_l, x_k) = p^{lk}$ indica el grado en el que una alternativa x_l se prefiere a la alternativa x_k .

$$P_{e_i} = \begin{pmatrix} 0.5 & \dots & 0.7 \\ \vdots & \ddots & \vdots \\ 0.3 & \dots & 0.5 \end{pmatrix}$$

Estas fueron el primer tipo de relaciones de preferencia usadas en la toma de decisiones [35], pero pronto aparecieron otros enfoques para tratar contextos con incertidumbre, que serán estudiados en las siguientes secciones.

2.2.2.2 Dominio intervalar.

El hecho de considerar la incertidumbre en los problemas de decisión ha originado la necesidad de definir modelados de preferencias más flexibles capaces de recoger dicha incertidumbre,

siendo el modelado intervalar uno de ellos. La valoración de alternativas por medio de intervalos $[\underline{a}, \bar{a}]$ ($\underline{a} \leq \bar{a}$) se ha mostrado como una técnica eficaz para tratar la incertidumbre en ciertos problemas de decisión [1, 38, 49].

Una relación de preferencia intervalar R sobre X se define como:

$$R : X \times X \rightarrow I([0,1])$$

donde $R(x_l, x_k) = p^{lk}$ denota el grado de preferencia intervalar de la alternativa x_l sobre x_k . En estos enfoques, las preferencias proporcionadas por los expertos son intervalos definidos sobre $I([0,1])$, donde la preferencia se expresa como $[\underline{a}, \bar{a}]^k$, con $(\underline{a} \leq \bar{a})$.

$$P_{e_i} = \begin{pmatrix} [0.5,0.5] & \cdots & [0.7,0.9] \\ \vdots & \ddots & \vdots \\ [0.1,0.3] & \cdots & [0.5,0.5] \end{pmatrix}$$

De este modo los expertos pueden valorar alternativas sobre las que no tienen un conocimiento suficiente como para asignarles valores exactos mediante un valor preciso.

2.2.2.3 Dominio lingüístico.

Los expertos pueden utilizar un modelado de preferencias lingüístico en aquellas situaciones de decisión en las que la información disponible es demasiado imprecisa o se valoran aspectos cuya naturaleza recomienda el uso de valoraciones cualitativas [11, 13, 26, 58, 59]. En estas situaciones el experto puede considerar más adecuado utilizar una palabra o término lingüístico para expresar sus preferencias, antes que un valor numérico más o menos preciso.

Una relación de preferencia lingüística difusa se define como:

$$R : X \times X \rightarrow S$$

siendo $S = \{s_0, \dots, s_g\}$ un conjunto de etiquetas.

El enfoque lingüístico difuso representa la información como valores lingüísticos por medio de variables lingüísticas [58]. Este enfoque es adecuado para cualificar fenómenos relativos a la percepción humana que normalmente valoramos usando palabras en lenguaje natural. Esto puede surgir por diferentes razones. Hay algunas situaciones donde la información puede ser no cuantificable debido a su naturaleza, y por tanto, puede ser expresada solamente mediante términos lingüísticos (por ejemplo cuando evaluamos el “confort” o el “diseño” de un coche, pueden usarse términos como “malo”, “pobre”, “tolerable”, “intermedio”, “bueno” [40]). En otros casos [59], hay cierta tolerancia para la imprecisión que puede ser explotada para conseguir extensibilidad, robustez, costes de solución bajos, y mejor correspondencia con la realidad (por ejemplo cuando evaluamos la velocidad de un coche, se usan términos lingüísticos como “rápido”, “muy rápido”, “lento”, en lugar de valores numéricos).

Una tarea importante cuando se trabaja con el enfoque lingüístico difuso, es la de elegir los descriptores lingüísticos apropiados para el conjunto de términos y su semántica. Una posibilidad de generar el conjunto lingüístico de términos consiste en suministrar directamente el conjunto de términos considerando que todos los términos están distribuidos en una escala en la que está definido un orden total. Por ejemplo, un conjunto de siete términos S , puede darse como:

$$S = \{s_0 = \text{mucho peor}, s_1 = \text{peor}, s_2 = \text{poco peor}, s_3 = \text{indiferente}, \\ s_4 = \text{poco mejor}, s_5 = \text{mejor}, s_6 = \text{mucho mejor}\}$$

En estos casos, se requiere que exista:

1. Un operador de negación, $Neg(s_i) = s_j$ tal que $j = g - i$ ($g + 1$ es la cardinalidad del conjunto de términos).
2. Un operador de maximización: $Max(s_i, s_j) = s_i$ si $s_i \geq s_j$.
3. Un operador de minimización: $Min(s_i, s_j) = s_i$ si $s_i \leq s_j$.

La semántica de los términos se da mediante números difusos, definidos en el intervalo $[0,1]$. Una forma de caracterizar un número difuso es utilizar una representación basada en parámetros de sus funciones de pertenencia [6]. Por ejemplo, podemos asignar la siguiente semántica al conjunto de siete términos mediante números difusos triangulares:

mp = mucho peor = (0, 0, 0.17)
 p = peor = (0, 0.17, 0.33)
 pp = poco peor = (0.17, 0.33, 0.5)
 i = indiferente = (0.33, 0.5, 0.67)
 pm = poco mejor = (0.5, 0.67, 0.83)
 m = mejor = (0.67, 0.83, 1)
 mm = mucho mejor = (0.83, 1, 1)

Como se muestra gráficamente en la Figura 2.4

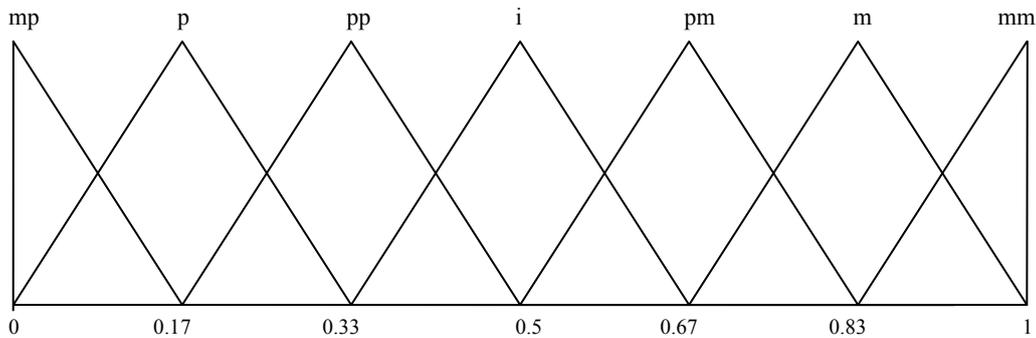


Figura 2.4: Conjunto de siete términos lingüísticos con su semántica asociada.

Por tanto una relación de preferencia lingüística $R(x_l, x_k)$ denota el grado de preferencia lingüística de la alternativa x_l sobre x_k . Usando el conjunto de términos lingüísticos mostrado en la Figura 2.4, una relación de preferencia lingüística podría ser:

$$P_{e_i} = \begin{pmatrix} i & \cdots & m \\ \vdots & \ddots & \vdots \\ p & \cdots & i \end{pmatrix}$$

2.2.3 Problemas de toma de decisiones en grupo definidos en contextos heterogéneos.

La situación ideal en un problema TDG es que todos los expertos tengan un conocimiento extenso sobre las alternativas y proporcionen sus opiniones un una escala numérica precisa. Sin embargo, en algunos casos, los expertos pueden pertenecer a distintas áreas de investigación y tener diferentes niveles de conocimiento sobre las alternativas. Debido a esto pueden preferir expresar sus preferencias por medio de diferentes dominios de información. En estos casos, podemos considerar que el problema está definido en un contexto heterogéneo.

Nos centraremos pues en este tipo de problemas, donde los expertos expresan sus preferencias usando diferentes dominios de expresión (numéricos, intervalares o lingüísticos) $D_i \in \{N | I | L\}$. Cada experto expresa sus opiniones por medio de una relación de preferencia definida sobre un único dominio de expresión, P_{e_i} :

$$P_{e_i} = \begin{pmatrix} p_i^{11} & \cdots & p_i^{1n} \\ \vdots & \ddots & \vdots \\ p_i^{n1} & \cdots & p_i^{nn} \end{pmatrix}$$

donde $p_i^{lk} \in D_i$ representa la preferencia de la alternativa x_l sobre la alternativa x_k dada por el experto e_i

2.3 Modelo de Sistema de Apoyo al Consenso para problemas TDG en contextos heterogéneos.

En esta sección presentamos un modelo de Sistema de Apoyo al Consenso (SAC) para problemas TDG con información heterogénea, tomando como base el modelo descrito en [33]. El modelo tiene dos características principales:

- Es capaz de llevar a cabo el proceso de consenso en problemas TDG en contextos heterogéneos con valoraciones numéricas, intervalares y lingüísticas.

- Incluye un generador de recomendaciones que asume el papel del moderador y recomienda cambios en las preferencias de los expertos con el fin de obtener un alto grado de consenso.

Este modelo además, ha sido construido usando:

- Una metodología para unificar la información heterogénea en un único dominio de expresión.
- Dos medidas para trabajar con el consenso: grado de consenso y medida de proximidad. La primera se usa para evaluar el acuerdo entre los expertos, mientras que la segunda se usa para medir la distancia entre la preferencia colectiva y las preferencias individuales de los expertos.
- En base a estas medidas se usa un conjunto de reglas de recomendación para guiar la dirección de los cambios en las opiniones de los expertos.

Así mismo, el modelo está formado por las siguientes fases:

1. *Unificación de la información.* En esta fase, las preferencias heterogéneas de los expertos se unifican en un único dominio.
2. *Cálculo del grado de consenso.* En esta fase se calcula el grado de consenso entre los expertos. Para hacer esto se define una función de similaridad para calcular la coincidencia entre las preferencias de los expertos.
3. *Control del nivel de consenso.* En esta fase el SAC controla el nivel de acuerdo conseguido entre los expertos. Si el acuerdo es mayor que un umbral de consenso especificado (γ) entonces el proceso de consenso acaba y se aplicará el proceso de selección para obtener la solución del problema. En otro caso, en la siguiente fase deben modificarse las preferencias de los expertos.

4. *Generación de recomendaciones.* Para ayudar a los expertos a cambiar sus preferencias, el SAC genera un conjunto de recomendaciones ayudándose de una medida de proximidad y del grado de consenso. Para ello se ha definido un generador de recomendaciones guiado que identifica las preferencias que deben ser cambiadas, y mediante un mecanismo de realimentación recomienda a los expertos como deben hacer los cambios para incrementar el acuerdo en la siguiente ronda de consenso.

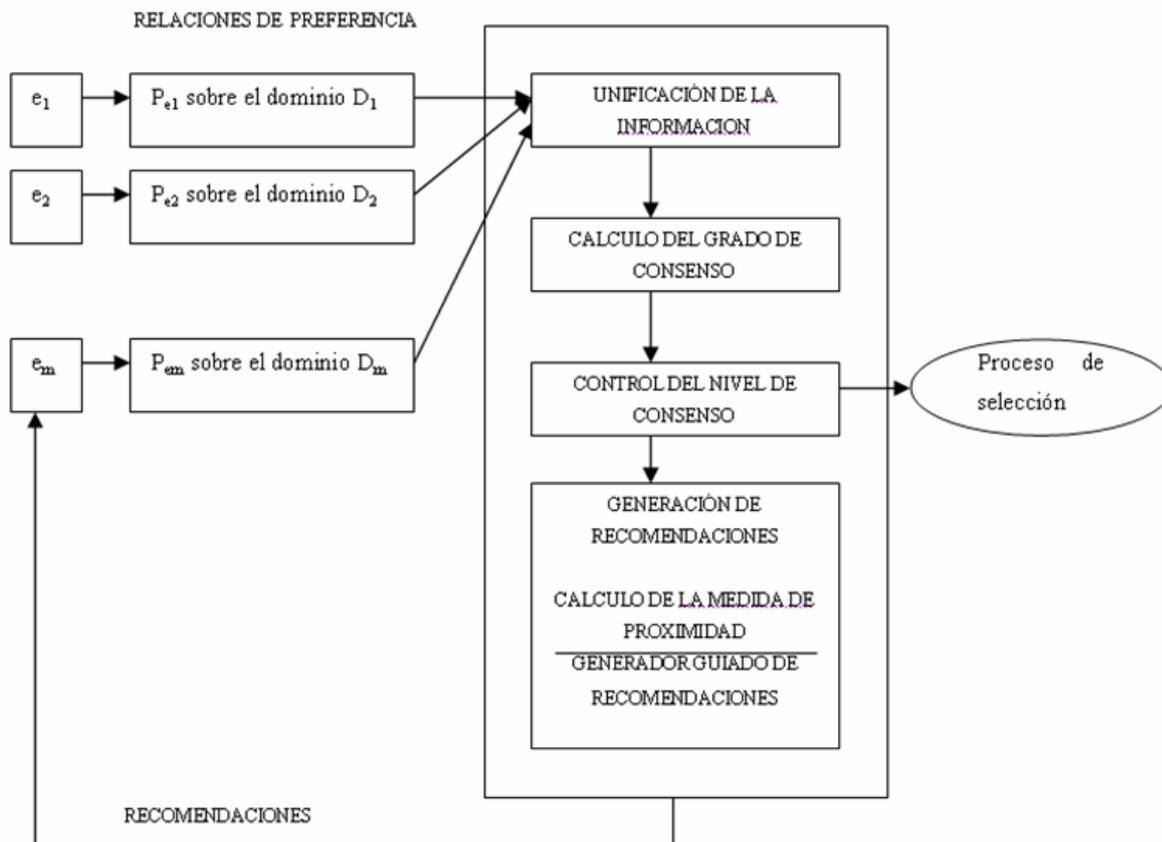


Figura 2.5: Un modelo de SAC que trabaja con información heterogénea.

2.3.1 Unificación de la información.

Considerando que estamos tratando con problemas de TDG definidos en contextos heterogéneos con información numérica, intervalar y lingüística y como no hay operadores estándares para manipular directamente información heterogénea, necesitamos unificarla en un espacio común

que llamaremos conjunto básico de términos lingüísticos (CBTL), $S_T = \{s_0, \dots, s_g\}$ (Figura 2.6). Para hacer esto, como se propone en [31], definimos una función de transformación diferente para transformar cada valor de preferencia numérica, intervalar y lingüística en un conjunto difuso definido sobre el CBTL, $F(S_T)$:

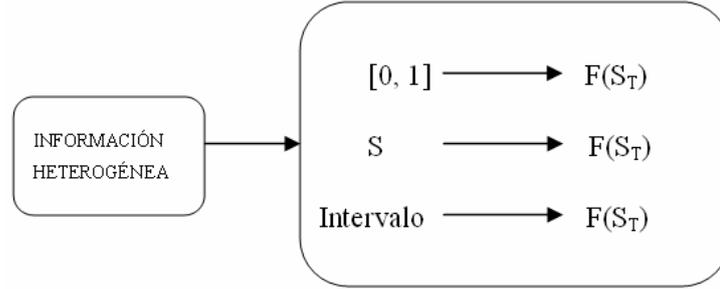


Figura 2.6: Proceso de unificación de información heterogénea.

2.3.1.1 Transformación de valores numéricos definidos en [0,1] en $F(S_T)$.

Para transformar un valor numérico en un conjunto difuso sobre S_T , usamos la siguiente función. Sea \mathcal{G} un valor numérico, $\mathcal{G} \in [0,1]$ y $S_T = \{s_0, \dots, s_g\}$ el CBTL. La función τ_{NS_T} que transforma un valor numérico \mathcal{G} en un conjunto difuso sobre S_T se define como [31]:

$$\tau_{NS_T} : [0,1] \rightarrow F(S_T)$$

$$\tau_{NS_T}(\mathcal{G}) = \{(s_0, \gamma_0), \dots, (s_g, \gamma_g), s_i \in S_T, \gamma_i \in [0,1]\}$$

$$\gamma_i = \mu(\mathcal{G}) = \begin{cases} 0, & \text{si } \mathcal{G} \notin \text{Soporte}(\mu_{s_i}(x)) \\ \frac{\mathcal{G} - a_i}{b_i - a_i}, & \text{si } a_i \leq \mathcal{G} \leq b_i \\ 1, & \text{si } b_i \leq \mathcal{G} \leq d_i \\ \frac{c_i - \mathcal{G}}{c_i - d_i}, & \text{si } d_i \leq \mathcal{G} \leq c_i \end{cases}$$

Nota 1: Consideramos las funciones de pertenencia, $\mu_{s_i}(\cdot)$ para las etiquetas lingüísticas, $s_i \in S_T$ representadas por una función paramétrica (a_i, b_i, d_i, c_i) . Un caso particular son las valoraciones lingüísticas cuyas funciones de pertenencia son triangulares, en cuyo caso $b_i = d_i$.

2.3.1.2 Transformación de los términos lingüísticos de S en $F(S_T)$.

Para transformar un valor lingüístico en un conjunto difuso sobre S_T , usamos la siguiente función. Sean $S = \{l_0, \dots, l_p\}$ y $S_T = \{s_0, \dots, s_g\}$ dos conjuntos de términos lingüísticos, tales que $g \geq p$. Entonces, la función τ_{SS_T} que transforma $l_i \in S$ en un conjunto difuso sobre S_T se define como:

$$\begin{aligned} \tau_{SS_T} : S &\rightarrow F(S_T) \\ \tau_{SS_T}(l_i) &= (s_k, \gamma_k^i) / k \in \{0, \dots, g\}, \forall l_i \in S \\ \gamma_k^i &= \max_y \min\{\mu_{l_i}(y), \mu_{s_k}(y)\} \end{aligned}$$

donde $F(S_T)$ es el conjunto de conjuntos difusos definidos en S_T , y $\mu_{l_i}(\cdot)$ y $\mu_{s_k}(\cdot)$ son las funciones de pertenencia de los conjuntos difusos asociados con los términos l_i y s_k , respectivamente.

Por tanto, el resultado de τ_{SS_T} para cualquier valor lingüístico de S es un conjunto difuso definido en S_T .

Nota 2: En el caso de que el conjunto de términos lingüísticos S de los contextos no homogéneos sea elegido como S_T , el conjunto difuso que representa un término lingüístico será todo 0 excepto el valor correspondiente al ordinal de la etiqueta lingüística, que será 1.

2.3.1.3 Transformación de valores intervalares en $F(S_T)$.

Para transformar un valor intervalar en un conjunto difuso sobre S_T , usamos la siguiente función. Sea $I = [\underline{i}, \bar{i}]$ un valor intervalar en $[0,1]$ y $S_T = \{s_0, \dots, s_g\}$ el CBTL. Entonces, la función τ_{IS_T} que transforma el valor intervalar I en un conjunto difuso sobre S_T se define como:

$$\begin{aligned} \tau_{IS_T} : I &\rightarrow F(S_T) \\ \tau_{IS_T}(I) &= (s_k, \gamma_k^i) / k \in \{0, \dots, g\} \\ \gamma_k^i &= \max_y \min\{\mu_I(y), \mu_{s_k}(y)\} \end{aligned}$$

donde la función $F(S_T)$ es el conjunto de conjuntos difusos definidos en S_T , y $\mu_I(\cdot)$ y $\mu_{s_k}(\cdot)$ son las funciones de pertenencia asociadas con el valor intervalar I y los términos s_k , respectivamente.

Nota 3: Asumimos que el valor intervalar tiene una representación inspirada en la función de pertenencia de los conjuntos difusos [36]:

$$\mu_I(\mathcal{G}) = \begin{cases} 0, & \text{si } \mathcal{G} < \underline{i} \\ 1, & \text{si } \underline{i} \leq \mathcal{G} \leq \bar{i} \\ 0, & \text{si } \bar{i} < \mathcal{G} \end{cases}$$

donde \mathcal{G} es un valor en $[0,1]$.

2.3.1.4 Resultados del proceso de unificación.

Una vez que hemos introducido en las subsecciones previas cada una de las diferentes funciones de transformación, observamos que aplicando las funciones de unificación, y asumiendo que cada conjunto difuso se mostrará por medio de sus grados de pertenencia $(\alpha_{i0}^{lk}, \dots, \alpha_{ig}^{lk})$, las preferencias de todos los expertos se expresan mediante matrices de conjuntos difusos, \tilde{P}_{e_i} , definidos en el CBTL:

$$\tilde{P}_{e_i} = \begin{pmatrix} \tilde{p}_i^{11} = (\alpha_{i0}^{11}, \dots, \alpha_{ig}^{11}) & \dots & \tilde{p}_i^{1n} = (\alpha_{i0}^{1n}, \dots, \alpha_{ig}^{1n}) \\ \vdots & \ddots & \vdots \\ \tilde{p}_i^{n1} = (\alpha_{i0}^{n1}, \dots, \alpha_{ig}^{n1}) & \dots & \tilde{p}_i^{nn} = (\alpha_{i0}^{nn}, \dots, \alpha_{ig}^{nn}) \end{pmatrix}$$

2.3.2 Cálculo del grado de consenso.

El grado de consenso evalúa el nivel de acuerdo entre los expertos. Para calcular el nivel de acuerdo es necesario calcular una matriz de consenso que representa las preferencias de todos los expertos y que se obtiene agregando las distancias entre las preferencias de los expertos.

Dado que la información con la que trabajamos son conjuntos difusos, la distancia entre dos preferencias \tilde{p}_i^{lk} y \tilde{p}_j^{lk} se calcula por medio de la función de similaridad $s(\tilde{p}_i^{lk}, \tilde{p}_j^{lk})$ medida en el intervalo unitario $[0,1]$ [33]:

$$s(\tilde{p}_i^{lk}, \tilde{p}_j^{lk}) = 1 - \left| \frac{cv_i^{lk} - cv_j^{lk}}{g} \right| \quad (1)$$

El valor cv_i^{lk} es el valor central del conjunto difuso:

$$cv_i^{lk} = \frac{\sum_{h=0}^g index(s_h^i) \cdot \alpha_{ih}^{lk}}{\sum_{h=0}^g \alpha_{ih}^{lk}} \quad (2)$$

y representa el valor central o centro de gravedad de la información contenida en el conjunto difuso $p_i^{lk} = (\alpha_{i0}^{lk}, \dots, \alpha_{ig}^{lk})$, siendo $index(s_h^i) = h$. El rango de este valor central es el intervalo cerrado $[0, g]$.

Un valor de $s(\tilde{p}_i^{lk}, \tilde{p}_j^{lk})$ más cercano a 1 indica que las preferencias \tilde{p}_i^{lk} y \tilde{p}_j^{lk} son similares, mientras que un valor de $s(\tilde{p}_i^{lk}, \tilde{p}_j^{lk})$ más cercano a 0 indica que las preferencias \tilde{p}_i^{lk} y \tilde{p}_j^{lk} son más distantes.

Una vez que hemos definido la función para evaluar la similaridad, el grado de consenso se calcula mediante los siguientes pasos:

Se calculan los valores centrales de todos los conjuntos difusos:

$$cv_i^{lk}; \forall i = 1, \dots, m; l, k = 1, \dots, n \wedge l \neq k. \quad (3)$$

Para cada pareja de expertos e_i y e_j ($i < j$), se calcula una matriz de similaridad $SM_{ij} = (sm_{ij}^{lk})$, donde

$$sm_{ij}^{lk} = s(\tilde{p}_i^{lk}, \tilde{p}_j^{lk}) \quad (4)$$

Se obtiene una *matriz de consenso* CM , agregando todas las matrices de similaridad.

$$CM = \begin{pmatrix} cm^{11} & \dots & cm^{1n} \\ \vdots & \ddots & \vdots \\ cm^{n1} & \dots & cm^{nn} \end{pmatrix}$$

Esta agregación es llevada a cabo a nivel de pares de alternativas:

$$cm^{lk} = \phi(sm_{ij}^{lk}); i, j = 1, \dots, m \wedge \forall l, k = 1, \dots, n \wedge i < j$$

En nuestro caso, usaremos la media aritmética como función de agregación ϕ , aunque podrían usarse diferentes operadores de agregación, según las propiedades particulares que se quieran implementar.

2.3.2.1 Interpretación del grado de consenso.

A partir de la matriz de consenso, podemos analizar el grado de consenso a tres niveles diferentes: pares de alternativas, alternativas y relaciones. De esta forma, tenemos la posibilidad de saber de forma precisa la situación del consenso y, por ejemplo, podemos identificar aquellos pares de alternativas en los que existe mayor desacuerdo.

Nivel 1. Consenso sobre pares de alternativas. El grado de consenso sobre un par de alternativas (x_l, x_k) , llamado cp^{lk} , mide el grado de consenso entre todos los expertos sobre ese par en concreto. En nuestro caso, esto se expresa por el elemento de la matriz de consenso CM , o sea,

$$cp^{lk} = cm^{lk}, \forall l, k = 1, \dots, n \wedge l \neq k$$

Valores de cp^{lk} cercanos a 1 denotan un mayor acuerdo. Esta medida permitirá la identificación de aquellos pares de alternativas con un bajo nivel de consenso.

Nivel 2. Consenso sobre alternativas. El grado de consenso sobre una alternativa x_l , llamado ca^l , mide el grado de consenso entre todos los expertos sobre esa alternativa. Se calcula como la media de cada fila l de la matriz de consenso CM , o sea,

$$ca^l = \frac{\sum_{k=1, l \neq k}^n cp^{lk}}{n-1} \quad (5)$$

Nivel 3. Consenso sobre relaciones o consenso global. El grado de consenso sobre relaciones, llamado cr mide el grado de consenso global entre las preferencias de los expertos. Se calcula como la media de todos los grados de consenso sobre alternativas, es decir,

$$cr = \frac{\sum_{l=1}^n ca^l}{n} \quad (6)$$

El Sistema de Apoyo al Consenso (SAC) usa este valor para comprobar el nivel de acuerdo alcanzado en cada ronda de consenso.

2.3.3 Control del nivel de consenso.

En esta fase el SAC controla el nivel de acuerdo alcanzado en la actual ronda de consenso. Antes de aplicar el modelo de consenso, debe ser fijado un umbral mínimo de consenso, $\gamma \in [0,1]$, que dependerá del problema particular con el que estemos tratando. Cuando las consecuencias de la decisión son de importancia trascendental, el mínimo nivel de consenso requerido para tomar esta

decisión lógicamente debería ser alto, por ejemplo $\gamma = 0.8$ o superior. En el otro extremo, cuando las consecuencias no son tan trascendentales (pero no dejan de ser importantes) o cuando es urgente la obtención de una solución al problema, podría requerirse un umbral de consenso cercano a 0.5.

En cualquier caso, independientemente del valor γ , cuando el consenso global cr alcanza el valor γ , el SAC se detendrá. Sin embargo, existe la posibilidad de que el consenso global no converja al umbral de consenso y el proceso quede bloqueado al entrar en un bucle infinito. Con objeto de evitar esta circunstancia, el modelo incorpora un parámetro, $nMaxRondas$, para limitar el número de rondas de consenso a realizar.

2.3.4 Generación de recomendaciones.

El proceso de obtención del consenso se define como un proceso dinámico e iterativo compuesto de varias rondas, donde los expertos expresan y discuten sus opiniones. Tradicionalmente este proceso está coordinado por un moderador humano, que calcula el acuerdo entre los expertos en cada ronda usando diferentes medidas de consenso [34, 37]. Si el acuerdo no es suficiente, el moderador recomienda que los expertos cambien las preferencias más alejadas de la opinión del grupo en un esfuerzo para acercar posturas en la siguiente ronda de consenso [8, 60].

El moderador puede ser una figura polémica, ya que los expertos pueden estar descontentos con su grado de objetividad. Además, en contextos heterogéneos el moderador puede tener problemas para comprender de forma adecuada los distintos dominios y escalas en que los expertos expresan sus opiniones. Para evitar esto, el modelo de SAC incorpora un generador guiado de recomendaciones, de manera que cuando el acuerdo no es suficiente (es decir, $cr < \gamma$) los expertos deben modificar sus preferencias para intentar un acercamiento e incrementar el consenso en la siguiente ronda de consenso. Para hacer esto, usamos medidas de proximidad para identificar las preferencias de los expertos más alejadas de la opinión colectiva. Una vez que estas preferencias han sido identificadas, el generador guiado de recomendaciones es el encargado de sugerir como cambiarlas para incrementar el consenso en la siguiente ronda de consenso. Ambos procesos se describen detalladamente a continuación:

2.3.4.1 Cálculo e interpretación de la Medida de Proximidad.

2.3.4.1.1 Cálculo de la Medida de Proximidad.

La medida de proximidad evalúa la distancia entre las preferencias de los expertos individuales y la preferencia colectiva del grupo de expertos. Para calcularla, primeramente necesitamos obtener una relación de preferencia colectiva \tilde{P}_c ,

$$\tilde{P}_c = \begin{pmatrix} \tilde{p}_c^{11} & \cdots & \tilde{p}_c^{1n} \\ \vdots & \ddots & \vdots \\ \tilde{p}_c^{n1} & \cdots & \tilde{p}_c^{nn} \end{pmatrix}$$

que representa la opinión del grupo de expertos. \tilde{P}_c se calcula agregando el conjunto de relaciones de preferencia individuales unificadas $\{\tilde{P}_{e_1}, \dots, \tilde{P}_{e_m}\}$:

$$\tilde{P}_c^{lk} = \psi(p_1^{lk}, \dots, p_m^{lk}) = (\alpha_{c0}^{lk}, \dots, \alpha_{cg}^{lk})$$

donde

$$\alpha_{cj}^{lk} = \psi(\alpha_{1j}^{lk}, \dots, \alpha_{mj}^{lk})$$

siendo ψ un “operador de agregación”, que en nuestro caso ha sido la media aritmética.

Una vez que el SAC ha obtenido la relación de preferencia colectiva, se calcula la matriz de proximidad, PM_i , para cada experto e_i ,

$$PM_i = \begin{pmatrix} pm_i^{11} & \cdots & pm_i^{1n} \\ \vdots & \ddots & \vdots \\ pm_i^{n1} & \cdots & pm_i^{nn} \end{pmatrix}$$

Para evaluar la proximidad entre las preferencias individuales de los expertos \tilde{P}_{e_i} , y las preferencias colectivas \tilde{P}_c usamos la función de similaridad definida en la expresión (1),

$$pm_i^{lk} = s(\tilde{p}_i^{lk}, \tilde{p}_c^{lk})$$

Estas matrices contienen la información necesaria para saber la posición de las preferencias de cada experto con respecto a la posición del grupo.

2.3.4.1.2 Interpretación de la Medida de Proximidad.

Las matrices de proximidad se estudian a nivel de pares de alternativas, alternativas y relaciones. De esta forma es fácil identificar los expertos más alejados en aquellas valoraciones donde no hay suficiente consenso:

Nivel 1. Proximidad sobre pares de alternativas. Dado un experto e_i , su medida de proximidad sobre un par de alternativas, (x_l, x_k) , llamado pp_i^{lk} , mide la proximidad entre su valor de preferencia y el valor de preferencia colectivo en ese par. En nuestro caso, este valor coincide con el elemento (l, k) de la matriz de proximidad PM_i , es decir,

$$pp_i^{lk} = pm_i^{lk}, \forall l, k = 1, \dots, n \wedge l \neq k$$

Nivel 2. Proximidad sobre alternativas. Dado un experto e_i , su medida de proximidad sobre una alternativa x_l , llamada pa_i^l , mide la proximidad entre sus valores de preferencia en esa alternativa y los valores colectivos. Se calcula como la media de las proximidades sobre pares de alternativas de x_l ,

$$pa_i^l = \frac{\sum_{k=1, k \neq l}^n pp_i^{lk}}{n-1} \quad (7)$$

Nivel 3. Proximidad sobre la relación. Dado un experto e_i , su medida de proximidad sobre la relación, llamada pr_i , mide la proximidad global entre sus valores de preferencia sobre todas las alternativas y el correspondiente valor colectivo. Se calcula como la media de todos los valores de proximidad sobre alternativas, es decir,

$$pr_i = \frac{\sum_{l=1}^n pa_i^l}{n} \quad (8)$$

Si pr_i está próximo a 1 entonces e_i contribuye positivamente al consenso, mientras que si pr_i está próximo a 0 e_i contribuye negativamente al consenso.

2.3.4.2 Generador guiado de recomendaciones.

El objetivo del generador orientado de recomendaciones es identificar las preferencias de los expertos más alejadas y sugerir como cambiarlas con objeto de incrementar el consenso. Para conseguir este propósito el generador guiado de recomendaciones usa dos tipos de reglas de recomendación: reglas de identificación y reglas de dirección.

2.3.4.2.1 Reglas de identificación.

Estas reglas identifican qué expertos, alternativas y pares de alternativas deben cambiarse. De esta forma, el modelo sólo se centra en las preferencias en desacuerdo y no recomendará cambiar aquellas preferencias donde el acuerdo es suficiente. El modelo usa tres reglas:

1. *Regla de identificación de expertos.* Identifica aquellos expertos que deberían cambiar algunos de sus valores de preferencias. Para ello calculamos la media de las medidas de proximidad de relación de los expertos,

$$\overline{pr} = \frac{\sum_{i=0}^m pr_i}{m}$$

Una vez calculado \overline{pr} , los expertos cuya medida de proximidad de relación sea menor que la media, deberán cambiar sus preferencias (es decir, $pr_i < \overline{pr}$)

Llamaremos a este conjunto de expertos *EXPCH*. Por tanto, la regla de identificación de expertos es la siguiente:

IR.1 $\forall e_i \in E \cap EXPCH$, entonces e_i debe cambiar sus preferencias, siendo

$$EXPCH = \{e_i \mid pr_i < \overline{pr}\}$$

2. *Regla de identificación de alternativas.* Esta regla identifica aquellas alternativas donde no hay suficiente consenso y por tanto deben ser cambiadas. Llamaremos a este conjunto de alternativas ALT y estará compuesto por todas aquellas alternativas cuyo grado de consenso ca^l es menor que el grado de consenso global alcanzado cr , es decir,

$$ALT = \{x_l \in X \mid ca^l < cr\}$$

La regla de identificación de alternativas es la siguiente:

IR.2 $\forall e_i \in EXPCH$, e_i debe cambiar algunas valoraciones asociadas con los pares que pertenecen a la alternativa x_l , tal que $x_l \in ALT$

3. *Una regla de identificación de pares de alternativas.* Esta identifica particularmente aquellos pares de alternativas (x_l, x_k) de las alternativas en desacuerdo $x_l \in ALT$ que deben cambiarse. Llamaremos a este conjunto de pares de alternativas $PALT_i$. Para hacer esto debemos calcular la media de las matrices de proximidad de los expertos, \overline{PM} ,

$$\overline{PM} = \begin{pmatrix} \overline{pp}^{11} & \cdots & \overline{pp}^{1n} \\ \vdots & \ddots & \vdots \\ \overline{pp}^{n1} & \cdots & \overline{pp}^{nn} \end{pmatrix}$$

donde $\overline{pp}^{lk} = \frac{\sum_{i=1}^m pp_i^{lk}}{m}$ y $m = n^\circ$ de expertos.

La regla de identificación de pares de alternativas es la siguiente:

IR.3 $\forall x_l \in ALT \wedge e_i \in EXPCH$, si $(x_l, x_k) \in PALT_i$ entonces e_i debe cambiar p_i^{lk} , siendo el conjunto de pares de alternativas (x_l, x_k) , cuyos valores de proximidad a nivel de pares, pp_i^{lk} , son menores que \overline{pp}^{lk} , es decir,

$$PALT_i = \{(x_l, x_k) \mid x_l \in ALT \wedge e_i \in EXPCH \wedge pp_i^{lk} < \overline{pp}^{lk}\}$$

2.3.4.2.2 Reglas de dirección.

Una vez que el modelo ha identificado los expertos y los pares de alternativas que deben cambiarse, $(x_l, x_k) \in PALT_i$, éste usa un conjunto de reglas de dirección para sugerir a los expertos como cambiar las valoraciones actuales con objeto de incrementar el acuerdo en la siguiente ronda de consenso. Antes de continuar, recordemos que las preferencias de los expertos han sido transformadas en conjuntos difusos sobre S_T ,

$$\tilde{p}_i^{lk} = (\alpha_0^{lk}, \dots, \alpha_g^{lk})$$

y que agregando todas las preferencias individuales,

$$\tilde{p}_c^{lk} = \psi(\tilde{p}_1^{lk}, \dots, \tilde{p}_m^{lk})$$

se obtienen las preferencias colectivas \tilde{p}_c^{lk} que representan la opinión del grupo de expertos.

El incremento del nivel de consenso se consigue acercando las preferencias individuales más alejadas a las preferencias colectivas. Para ello se necesita conocer la posición actual de ambas preferencias. Estas posiciones se obtienen calculando los valores centrales de los conjuntos difusos que representan las preferencias individuales y colectivas. Tal como comentamos en la subsección 2.3.2, el valor central se define como la posición o el centro de gravedad de la información contenida en el conjunto difuso $\tilde{p}_i^{lk} = (\alpha_0^{lk}, \dots, \alpha_g^{lk})$ y se calcula según la expresión (2):

$$cv_i^{lk} = \frac{\sum_{h=0}^g index(s_h^i) \cdot \alpha_{ih}^{lk}}{\sum_{h=0}^g \alpha_{ih}^{lk}}$$

donde $index(s_h^i) = h$.

Una vez conocidas las posiciones de las preferencias individuales y colectivas, ya es posible definir un conjunto de reglas que permitan identificar la dirección en la que han de cambiar las preferencias individuales para aproximarlas a las colectivas. Estas reglas comparan los valores centrales de los conjuntos difusos que representan las preferencias individuales y colectivas. El

resultado de esta comparación será recomendar incrementar o decrementar las valoraciones individuales dadas por los expertos.

Proponemos tres reglas de dirección que se aplicarán sobre los conjuntos de preferencias a modificar, $PALT_i$, generados por las reglas de identificación:

DR1. Si $(cv_i^{lk} - cv_c^{lk}) < 0$, el experto e_i debería incrementar la valoración dada al par de alternativas (x_l, x_k) .

DR2. Si $(cv_i^{lk} - cv_c^{lk}) > 0$, el experto e_i debería decrementar la valoración dada al par de alternativas (x_l, x_k) .

DR3. Si $(cv_i^{lk} - cv_c^{lk}) = 0$ el experto e_i no modificará la valoración dada al par de alternativas (x_l, x_k) .

CAPÍTULO 3

PROYECTO

3.1 Descripción.

Una vez presentado el propósito y los objetivos del proyecto y explicado el modelo teórico del Sistema de Apoyo al Consenso en Toma de Decisión en Grupo en Contextos Heterogéneos, llega el momento de pasar a detallar el desarrollo del proyecto que se va a realizar.

El proyecto aborda el desarrollo de un sistema Cliente/Servidor basado en WEB con tecnología JAVA que implementa un Sistema de Apoyo a la Toma de Decisión en Grupo en Contextos Heterogéneos.

En este capítulo se detalla el proceso de desarrollo de dicho software y las actividades de la Ingeniería del Software que deben seguirse. No existe una definición única y estandarizada para la Ingeniería del Software pero, las dos que se presentan a continuación pueden resultar perfectamente válidas para este cometido:

- *Ingeniería del Software* es la construcción de software de calidad con un presupuesto limitado y un plazo de entrega en contextos de cambio continuo.
- *Ingeniería del Software* es el establecimiento y uso de principios y métodos firmes de ingeniería para obtener software económico que sea fiable y funcione de manera eficiente en máquinas reales.

Las actividades que conforman la Ingeniería del Software son las siguientes:

- **Especificación de Requerimientos:** se obtiene el propósito del sistema, las propiedades y restricciones del mismo.
- **Análisis del Sistema:** se obtiene un modelo del sistema correcto, completo, consistente, claro y verificable.
- **Diseño del Sistema:** se definen los objetivos del proyecto y las estrategias a seguir para conseguirlos.

- **Implementación:** se traduce el modelo a código fuente.
- **Prueba:** verificar y validar el sistema.

En los puntos siguientes se profundizará en cada una de estas actividades y en como se han llevado a cabo en el ámbito de nuestro proyecto.

3.2 Especificación de requerimientos.

El primer paso en la Ingeniería del Software debe ser determinar el propósito último del proyecto, las propiedades que debe satisfacer y las restricciones a las que está sometido. Este es, sin duda, un paso de vital importancia dentro del desarrollo de un proyecto software ya que, sin conocer el propósito del proyecto y todas las limitaciones de diversa índole a las que debe hacer frente, difícilmente se podrá realizar una aplicación software que cumpla dicho propósito.

En un proyecto de ámbito comercial para una empresa real, para determinar el propósito del mismo se recurre a una serie de estudios como pueden ser entrevistas con los clientes, estudios de la situación actual del sistema o estudios de viabilidad. En nuestro caso no nos encontramos ante un proyecto comercial sino ante un proyecto académico por lo que el propósito es conocido desde el mismo momento de la concepción del mismo:

Implementar un sistema de Apoyo a la Toma de Decisión en Grupo en Contextos Heterogéneos con arquitectura Cliente/Servidor basada en WEB y con tecnología JAVA.

Una vez determinado el propósito del proyecto, el siguiente paso es especificar los requerimientos del mismo. Los requerimientos de un proyecto software son el conjunto de propiedades o restricciones definidas con total precisión, que dicho proyecto software debe satisfacer. Existen dos tipos bien diferenciados de tales requerimientos:

- I. *Requerimientos funcionales*: aquellos que se refieren específicamente al funcionamiento de la aplicación o sistema.
- II. *Requerimientos no funcionales*: aquellos no referidos al funcionamiento estricto sino a otros factores externos.

En los dos siguientes subapartados definiremos cuáles son estos requerimientos (tanto funcionales como no funcionales) para el proyecto del que se ocupa esta memoria. Sin embargo, estas definiciones sólo serán preliminares ya que, en la actividad de análisis del sistema donde se crearán los casos de uso y sus escenarios, se descubrirán nuevas necesidades que no son observables en esta primera actividad y que permitirán refinar completamente estos requerimientos.

3.2.1 Requerimientos funcionales.

Los requerimientos funcionales de un sistema software son aquellos que se encargan de describir las funcionalidades que el sistema debe proporcionar a los usuarios del mismo para cumplir sus expectativas.

Las funcionalidades que se espera de un sistema como el nuestro son las siguientes:

Para ambos usuarios (*Administrador y Experto*):

- a) Identificación y validación cuando un usuario entra en el sistema.
- b) Salir del sistema.

Para el usuario *Administrador*:

- a) Crear un problema.
- b) Modificar un problema.
- c) Consultar un problema.
- d) Eliminar un problema.
- e) Crear/modificar/consultar y eliminar un *Experto*.

- f) Asignar expertos a un problema.
- g) Calcular el nivel de consenso de una ronda de un problema.
- h) Consultar la evolución de consenso de un problema.

Para el usuario *Experto*:

- a) Introducir las preferencias de un problema.
- b) Consultar el estado de un problema.

Una vez definidas cuales son las funcionalidades que los distintos usuarios pueden reclamar a nuestro sistema, se hace necesario caracterizar de una manera más formal y concreta cómo va a responder a estas funcionalidades nuestro sistema. Además, para distinguir cada uno de los requisitos utilizaremos este identificador: RF-Número_del_Requisito:

1) RF-01: Identificación y validación cuando un usuario entra en el sistema.

Para poder entrar en nuestra aplicación, el sistema debe proporcionar al usuario un formulario en el que introducirá, su nombre de usuario y su contraseña. El sistema valida estos datos mediante una base de datos, y si son correctas el usuario accede a la aplicación.

2) RF-02: Salir del sistema.

En los menús de usuario, ya sea éste *Administrador* o *Experto*, el sistema proporciona esta opción para salir de la aplicación, mostrando un formulario que da la posibilidad de volver a entrar en el sistema.

3) RF-03: Crear un problema.

El sistema debe proporcionar al *Administrador* la posibilidad de crear un nuevo problema en el que se especifique el identificador, una breve descripción del mismo, el número de expertos que intervienen, el número de alternativas y establecer todos los parámetros necesarios. Debe permitir definir las alternativas y asignar *Expertos* almacenados en la BBDD.

4) RF-04: Modificar un problema.

El sistema debe permitir modificar un problema almacenado en la BBDD, en el que se pueda añadir o eliminar alternativas, asignar nuevos expertos, eliminar expertos asignados al problema y cambiar los parámetros definidos.

5) RF-05: Consultar un problema.

Debe permitir consultar toda la información referente a un problema, como puede ser identificador, definición, alternativas, expertos asignados y parámetros establecidos.

6) RF-06: Eliminar un problema.

Debe permitir eliminar de la BBDD un problema existente.

7) RF-07: Crear/modificar/consultar y eliminar un *Experto*.

El sistema debe proporcionar al *Administrador* la posibilidad de añadir un nuevo *Experto*, modificarlo, consultar la información referente a dicho *Experto* (problemas asignados, estado de los problemas, n° de ronda en la que se encuentra cada problema y n° de cambios que debe realizar en la siguiente ronda) y eliminar un *Experto*.

8) RF-08: Asignar *Expertos* a un problema.

Debe permitir asignar *Expertos* a un problema para que puedan introducir sus preferencias en dicho problema.

9) RF-09: Calcular el nivel de consenso de una ronda de un problema.

Debe permitir realizar una ronda en la que el sistema unifica la información, calcula el consenso y genera las recomendaciones necesarias para que en la próxima ronda los expertos modifiquen sus preferencias y el grado de consenso sea mayor.

10) RF-10: Consultar la evolución de consenso de un problema.

El sistema debe proporcionar al *Administrador* la posibilidad de consultar el grado de consenso de la ronda actual, permitir seleccionar una ronda anterior y ver el grado de consenso, n° de cambios totales e individuales (de cada uno de los *Expertos* asignados)

realizados en la ronda seleccionada. También debe permitir seleccionar un *Experto* asignado al problema y mostrar las preferencias introducidas por dicho *Experto* en la ronda indicada.

11) RF-11: Introducir las preferencias de un problema.

El sistema debe proporcionar al *Experto* la posibilidad de introducir las preferencias en un problema en el que esté asignado y visualizar las recomendaciones de la ronda anterior, en el caso que existan.

12) RF-12: Consultar el estado de un problema.

El sistema debe proporcionar al *Experto* la posibilidad de consultar la descripción de un problema, las preferencias introducidas por dicho *Experto* en la ronda actual o anterior, así como visualizar el grado de consenso y recomendaciones generadas por el sistema en cualquier ronda realizada.

3.2.2 Requerimientos no funcionales.

Los requerimientos no funcionales son aquellos que restringen los requerimientos funcionales. Son tan importantes como los propios requerimientos funcionales y pueden incluso llegar a ser críticos para la aceptación del sistema. Estos requerimientos normalmente especifican propiedades del sistema o del producto en sí (plataforma, velocidad, rendimiento...) y del diseño de la interfaz gráfica con el usuario además de todas las restricciones impuestas por la organización (políticas de empresa, estándares, legalidad vigente...).

Los requerimientos no funcionales que se deben obtener y analizar para este proyecto son los referentes a las necesidades hardware y software de los equipos informáticos. Para que éstos, proporcionen al usuario las funcionalidades requeridas de forma eficiente, así como las referentes a la interfaz gráfica entre la aplicación y el usuario.

A. Requerimientos del equipo informático

Al hablar de los requerimientos del equipo informático y debido a que el marco del desarrollo de la aplicación es una arquitectura Cliente/Servidor, debemos diferenciar los requerimientos de equipo que necesita el servidor y los que necesita el cliente.

Las necesidades de equipo informático del cliente son muy simples ya que tan solo le hace falta un computador conectado a Internet (preferiblemente de banda ancha) y tener instalado un navegador capacitado para visualizar de forma correcta la aplicación (se recomienda Internet Explorer, pero podría ser válido cualquier otro).

Los requerimientos del equipo informático del servidor, el cual se aconseja que sea un equipo dedicado, son más amplios y se dividen en dos tipos: los requerimientos de hardware y los requerimientos software.

1. Hardware

- *Velocidad*: el equipo debe ser lo suficientemente rápido como para ejecutar la aplicación en el menor tiempo posible y con la mayor fiabilidad. Cualquier microprocesador actual es capaz de cumplir con esta labor.
- *Memoria*: el equipo debe disponer de la suficiente memoria RAM libre para realizar las operaciones que se soliciten entre la aplicación y la base de datos.
- *Almacenamiento*: el equipo que haga la labor de servidor debe tener una capacidad de almacenamiento suficiente para almacenar la base de datos con la que trabaja la aplicación y permitir con holgura las transacciones entre ambas entidades.
- *Tarjeta gráfica*: las tarjetas gráficas de las que disponen los equipos informáticos actuales son de gran potencia por lo que, no es necesario establecer ningún requerimiento en este aspecto.

- *Monitor*: el monitor debe soportar una resolución de 1024x768 o superiores.
- *Conexión a Internet*: el servidor debe encargarse de que la aplicación sea accesible a través de Internet para todos sus usuarios. Por lo que, es indispensable que se encuentre conectado a Internet a través de banda ancha las 24 horas del día.

2. Software

- *Sistema Operativo*: el servidor de la aplicación trabaja sobre un sistema operativo Windows XP Profesional.
- *Navegador*: la aplicación debe poder ser visualizada desde cualquier navegador Web actual aunque se recomienda el uso de Internet Explorer en sus últimas versiones.
- *Sistema Gestor de Bases de Datos*: la aplicación trabaja con la base de datos mysql versión 5.0.51a-win32.
- El resto del software necesario será proporcionado al Administrador de la aplicación, el cual dispone de un manual para su instalación en el Anexo I.

B. Requerimientos de la interfaz

Los requerimientos de la interfaz gráfica entre la aplicación y el usuario están íntimamente ligados a la usabilidad y sus principios. La usabilidad se puede definir de varias formas:

- Usabilidad se define coloquialmente como facilidad de uso, ya sea de una página Web, una aplicación informática o cualquier otro sistema que interactúe con un usuario.

- Usabilidad se refiere a la capacidad de un software de ser comprendido, aprendido, usado y ser atractivo para el usuario, en condiciones específicas de uso.
- Usabilidad es la efectividad, eficiencia y satisfacción con la que un producto permite alcanzar objetivos específicos a usuarios específicos en un contexto de uso específico.

A partir de estas tres definiciones se pueden obtener los principios básicos de la usabilidad, los cuales se asociarán a los requerimientos no funcionales que deberá cumplir la interfaz gráfica:

- **Facilidad de aprendizaje:** se refiere a la facilidad con la que nuevos usuarios pueden tener una interacción efectiva. Depende de los siguientes factores:
 - ✓ *Predecibilidad:* una vez conocida la aplicación, se debe saber en cada momento a que estado se pasará en función de la tarea que se realice.
 - ✓ *Síntesis:* los cambios de estado tras una acción deben ser fácilmente captados.
 - ✓ *Generalización:* las tareas semejantes se resuelven de modo parecido.
 - ✓ *Familiaridad:* el aspecto de la interfaz tiene que resultar conocido y familiar para el usuario.
 - ✓ *Consistencia:* siempre se han de seguir una misma serie de pasos para realizar una tarea determinada.
- **Flexibilidad:** relativa a la variedad de posibilidades con las que el usuario y el sistema pueden intercambiar información. También abarca la posibilidad de diálogo, la multiplicidad de vías para realizar la tarea, similitud con tareas anteriores y la optimización entre el usuario y el sistema.

- **Robustez:** es el nivel de apoyo al usuario que facilita el cumplimiento de sus objetivos o, también, la capacidad del sistema para tolerar fallos. Está relacionada con los siguientes factores:
 - ✓ *Navegable:* el usuario debe poder observar el estado del sistema sin que esta observación repercuta de forma negativa en él.
 - ✓ *Recuperación de información:* la aplicación debe poder deshacer alguna operación y permitir volver a un estado anterior.
 - ✓ *Tiempo de respuesta:* es el tiempo necesario para que el sistema pueda mostrar los cambios realizados por el usuario.

3.3 Análisis del sistema.

Una vez conocido el propósito del proyecto software, las propiedades que debe cumplir y las restricciones a las que debe someterse, llega el momento de analizar el sistema y crear un modelo del mismo que sea correcto, completo, consistente, claro y verificable. Para conseguir esto se estudiarán los perfiles de usuario, se crearán y definirán casos de uso en base a los requerimientos previamente obtenidos. Por último, se describirán ciertos escenarios de acción de dichos casos de uso.

3.3.1 Perfil de usuario.

En esta fase el primer paso es determinar quienes son los usuarios potenciales de la aplicación, y a partir de ellos, obtener las características generales que nos permitan caracterizar los requisitos de usabilidad. Que posteriormente habrá que tener en cuenta en el diseño de la aplicación y de su interfaz gráfica. Nuestro sistema, cuenta con dos tipos de usuarios: *administrador* y *experto*.

Pasamos a comentar las características de cada uno de los usuarios de nuestro sistema:

Administrador:

- *Conocimientos del dominio del problema:* debe tener un buen conocimiento del dominio para poder crear problemas, asignar expertos a dichos problemas e interpretar los resultados obtenidos después de calcular el nivel de consenso en una ronda.
- *Sobre uso de equipos/programas informáticos:* no es necesario que tenga conocimientos sobre otros programas informáticos, sólo sobre nuestra aplicación y el navegador que utilice, ya que no será necesario el uso de ningún otro programa específico. En cuanto al manejo de equipos informáticos, sí deberá tener unos conocimientos básicos a nivel de usuario, como son: encendido y apagado del equipo, manejo del teclado, ratón,...
- *Entorno de trabajo:* será su lugar de trabajo habitual, por lo que no necesitará el uso de ningún software o hardware especial.
- *Nivel cultural:* se presupone un nivel cultural medio-alto, necesario para la comprensión de sus resultados asociados.
- *Habilidades sociales:* el administrador debe tener habilidades de trato con los expertos, y también será importante que sepa trabajar en grupo y tener habilidades de interrelación personal con sus compañeros de trabajo.

Experto:

- *Conocimientos del dominio del problema:* debe tener un conocimiento experto específico del dominio del problema como para saber valorar las alternativas de un problema de una forma acertada e interpretar las recomendaciones generadas adecuadamente.
- *Sobre uso de equipos/programas informáticos:* no es necesario que tenga conocimientos sobre otros programas informáticos, sólo sobre nuestra aplicación y el navegador que utilice, ya que no será necesario el uso de ningún otro programa específico. En cuanto al

manejo de equipos si deberá tener unos conocimientos básicos a nivel de usuario como son: encendido y apagado del equipo, manejo del ratón, teclado, etc.

- *Entorno de trabajo*: será su oficina habitual, por lo que no necesitará el uso de ningún software o hardware especial.
- *Nivel cultural*: se presupone un nivel cultural medio-alto, dependiendo del tipo del problema que haya que resolver.
- *Habilidades sociales*: puesto que no es una ocupación de cara al público, no tiene porque tener habilidades específicas de trato hacia el público.

3.3.2 Casos de uso.

Un caso de uso representa una clase de funcionalidad dada por el sistema como un flujo de eventos. También se puede definir como la representación de una situación o tarea de interacción de un usuario con la aplicación.

Los casos de uso son tareas con significado, coherentes y relativamente independientes, que los actores realizan en su trabajo cotidiano. En un caso de uso concreto puede participar más de un actor.

Los casos de uso describen como se realiza una tarea de manera exacta y constan de los siguientes elementos:

- Nombre único e unívoco
- Actores participantes
- Condiciones de entrada
- Flujo de eventos
- Condiciones de salida
- Requerimientos especiales

Por lo tanto, es necesario determinar cuales son los actores participantes en cada uno de los casos de uso. Un actor modela una entidad externa que se comunica con el sistema, es decir, es un tipo de usuario del sistema. Un actor, al igual que un caso de uso, debe tener un nombre único y puede tener una descripción asociada.

En nuestro sistema contamos con los dos siguientes actores:

- **Administrador:** es la persona responsable de la aplicación, se encarga de la gestión de problemas y expertos.
- **Experto:** es la persona que participa en el proceso de consenso dando su opinión en los problemas que tenga asignados. En el sistema van a existir muchos expertos.

Una vez definidos cuales van a ser los actores del sistema, es el momento de crear los distintos casos de uso. A la hora de realizar esta acción es importante que cada uno de los requerimientos funcionales ya definidos, aparezca en al menos uno de los casos de uso. Destacamos que puede haber casos de uso nuevos, en los que no aparezca ninguno de los requerimientos, ya que estamos en una fase de refinamiento del sistema donde queremos construir un modelo detallado del mismo.

Un paso previo a la creación y descripción de los distintos casos de uso es la obtención de los diversos diagramas de casos de uso de nuestro sistema. El primero es un diagrama frontera, es decir, un diagrama que describe completamente la funcionalidad de un sistema:

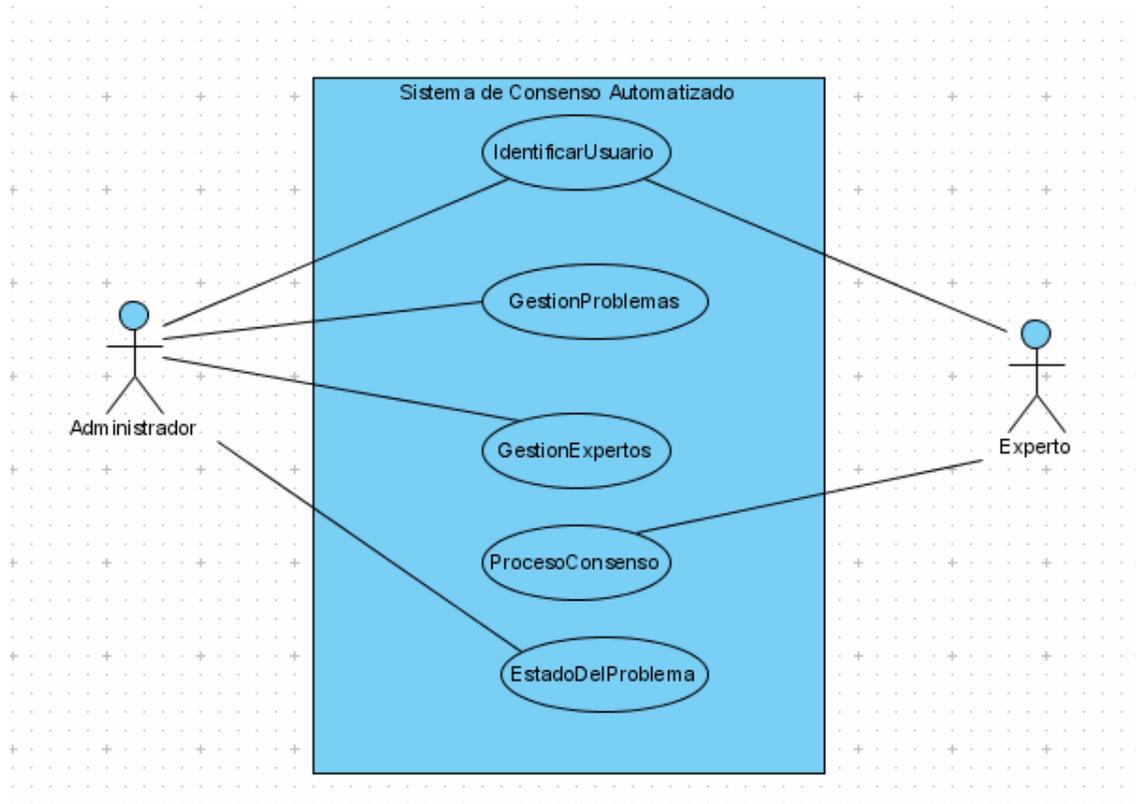


Figura 3.1: Diagrama frontera del Sistema.

Los casos de uso mostrados en un diagrama frontera pueden ser lo suficientemente exactos o, por el contrario, pueden ser concretados con un mayor detalle. A la hora de detallar un caso de uso se pueden emplear dos tipos de relaciones:

- <<extend>>: es una relación cuya dirección es hacia el caso de uso a detallar que representa comportamientos excepcionales del caso de uso.
- <<include>>: es una relación cuya dirección es contraria a la de la relación <<extend>> que representa un comportamiento común del caso de uso.

En nuestro caso nos encontramos con que los casos de uso “GestiónProblemas”, “GestionExpertos”, “ProcesoConsenso” y “EstadoDelProblema” requieren ser detallados en más profundidad. En las siguientes figuras mostramos los diagramas de casos de uso de estos cuatro casos:

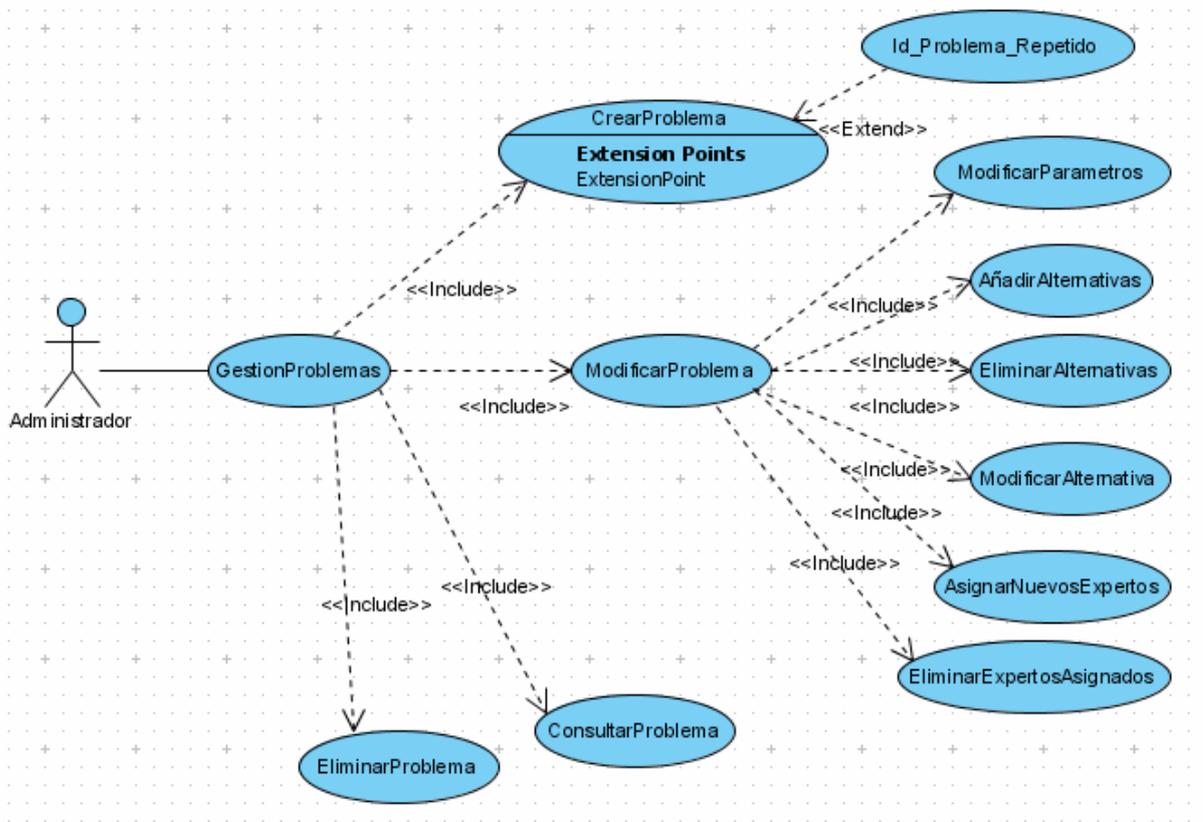


Figura 3.2: Diagrama del caso de uso "GestionProblemas"

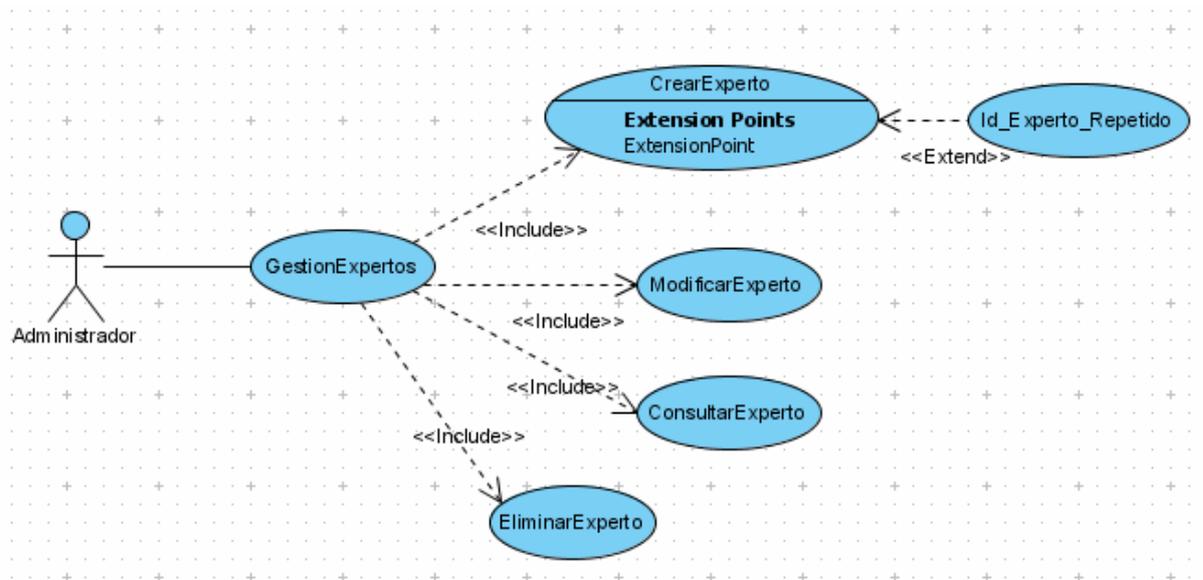


Figura 3.3: Diagrama del caso de uso "GestionExpertos"

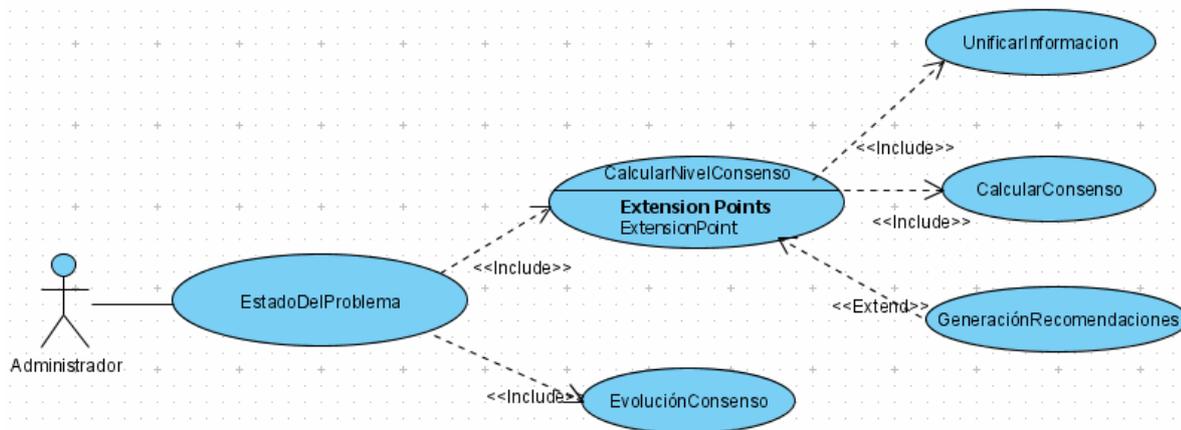


Figura 3.4: Diagrama del caso de uso "EstadoDelProblema"

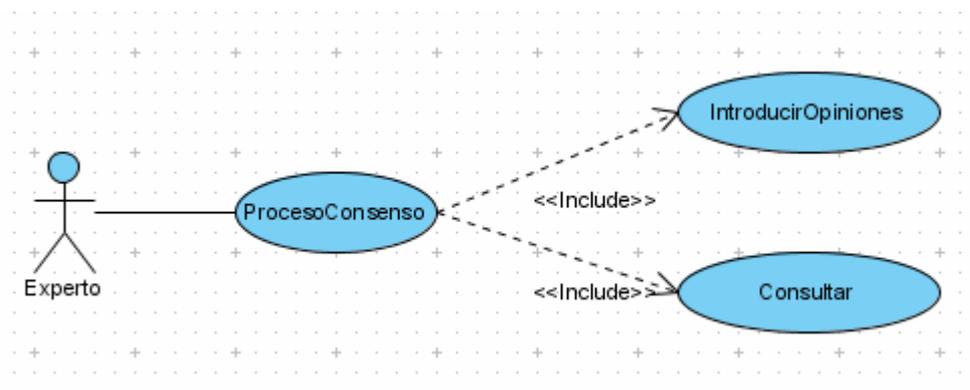


Figura 3.5: Diagrama del caso de uso "ProcesoConsenso"

A continuación, se describen detalladamente cada uno de los casos de uso mostrados en las figuras anteriores.

◆ **Caso de uso 1: Identificar Usuario**

Actores participantes: Administrador y Experto.

Condición de entrada: Que existan usuarios en el sistema que tengan nombre_usuario y contraseña.

Flujo de eventos:

1. El sistema muestra un formulario de entrada.
2. El usuario ya sea Administrador o Experto introduce el nombre de usuario y su contraseña.
3. El sistema comprueba en la BBDD que los datos del usuario son correctos (E-1).
4. Si el usuario es:
 - Administrador: el sistema mostrará el panel de control del administrador
 - Experto: el sistema mostrará los problemas que el usuario tiene asignados.

Condición de salida: El usuario (Administrador o Experto) ha sido autenticado por el sistema.

Excepciones:

E-1: El identificador introducido por el usuario no es correcto. El sistema muestra un mensaje de error y vuelve a mostrar el formulario de entrada.

◆ **Caso de uso 2: GestionProblemas**

Actores participantes: Administrador.

Condición de entrada: Que el administrador esté identificado en el sistema.

Flujo de eventos:

1. El sistema muestra un menú principal con cuatro opciones y le pide al Administrador que elija una:

- Si elige CrearProblema, se realiza S-1
- Si elige ModificarProblema, se realiza S-2
- Si elige ConsultarProblema, se realiza S-3
- Si elige EliminarProblema, se realiza S-4
- Si elige Salir, se termina el caso de uso.

Subflujo de eventos:

S-1: CrearProblema

1. El Administrador introduce el número de expertos que intervienen en el problema y el número de alternativas (E-1).
2. El Administrador introduce el nombre del problema (identificador), realiza una breve descripción del mismo y establece todos los parámetros indicados (E-2).
3. El Administrador define las alternativas.
4. El Administrador asigna Expertos al problema.
5. El sistema almacena la información en la BBDD.
6. Se inicia de nuevo el caso de uso GestiónProblemas.

S-2: ModificarProblema

1. El sistema muestra un listado de todos los problemas creados.
2. El Administrador selecciona el problema que desea modificar.
3. El Administrador elige una de las cinco posibles opciones:
 - ModificarParametros, se realiza S-2-1
 - AñadirAlternativas, se realiza S-2-2
 - EliminarAlternativas, se realiza S-2-3
 - ModificarAlternativas, se realiza S-2-4
 - AsignarNuevosExpertos, se realiza S-2-5
 - EliminarExpertosAsignados, se realiza S-2-6

S-2-1: ModificarParametros

1. El sistema muestra todos los parámetros establecidos en el problema.
2. El Administrador cambia los parámetros que desea.
3. El sistema almacena los cambios en la BBDD.
4. Se inicia de nuevo el caso de uso ModificarProblema.

S-2-2: AñadirAlternativas

1. El sistema muestra las alternativas definidas para el problema.
2. El Administrador añade nuevas alternativas.
3. El sistema guarda los cambios en la Base de Datos.
4. Se inicia de nuevo el caso de uso ModificarProblema.

S-2-3: EliminarAlternativas

1. El sistema muestra las alternativas definidas para el problema.
2. El Administrador selecciona las alternativas que desea eliminar.

3. El sistema las elimina y guarda los cambios en la Base de Datos.
4. Se inicia de nuevo el caso de uso ModificarProblema.

S-2-4: ModificarAlternativas

1. El sistema muestra las alternativas definidas para el problema.
2. El Administrador modifica la descripción de las alternativas que desea cambiar.
3. El sistema guarda los cambios en la Base de Datos.
4. Se inicia de nuevo el caso de uso ModificarProblema.

S-2-5: AsignarNuevosExpertos

1. El sistema muestra todos los expertos asignados al problema.
2. El Administrador selecciona los nuevos expertos que desea asignar.
3. El sistema asigna los nuevos expertos al problema y guarda los cambios en la Base de Datos.
4. Se inicia de nuevo el caso de uso ModificarProblema.

S-2-6: EliminarExpertosAsignados

1. El sistema muestra todos los expertos asignados al problema.
2. El Administrador selecciona los expertos que desea eliminar.
3. El sistema los elimina y guarda los cambios en la Base de Datos.
4. Se inicia de nuevo el caso de uso ModificarProblema.

S-3: ConsultarProblema

1. El sistema muestra un listado de todos los problemas creados.
2. El Administrador selecciona un problema.
3. El sistema muestra toda la información referente a dicho problema (nombre, definición, alternativas, expertos asignados, parámetros y estado actual).

4. Se inicia de nuevo el caso de uso GestionProblema.

S-4: EliminarProblema

1. El sistema muestra un listado de todos los problemas creados.
2. El Administrador selecciona los problemas que desea eliminar.
3. El sistema los elimina y almacena los cambios en la Base de Datos.
4. Se inicia de nuevo el caso de uso GestionProblema.

Condición de salida: La acción elegida se ha llevado a cabo.

Excepciones:

E-1: El número de expertos y/o alternativas introducidos no es un entero positivo. El sistema muestra un mensaje de error.

E-2: El identificador introducido para el problema ya existe, el número máximo de rondas no es un entero positivo o el umbral de consenso no es un real comprendido en el intervalo $[0, 1]$. El sistema muestra un mensaje de error.

♦ **Caso de uso 3: GestionExpertos**

Actores participantes: Administrador.

Condición de entrada: Que el administrador esté identificado en el sistema.

Flujo de eventos:

1. El sistema muestra un menú principal con cuatro opciones y le pide al Administrador que elija una:

- Si elige CrearExperto, se realiza S-1
- Si elige ModificarExperto, se realiza S-2
- Si elige ConsultarExperto, se realiza S-3
- Si elige EliminarExperto, se realiza S-4
- Si elige Salir, se termina el caso de uso.

Subflujo de eventos:

S-1: CrearExperto

1. El sistema muestra un formulario para introducir todos los datos del experto.
2. El Administrador rellena el formulario (E-1).
3. El sistema almacena la información en la BBDD.
4. Se inicia de nuevo el caso de uso GestionExpertos.

S-2: ModificarExperto

1. El sistema muestra un listado con todos los expertos creados.
2. El Administrador selecciona el experto que desea modificar.
3. El sistema muestra todos los datos del experto y ofrece la posibilidad de modificarlos.

4. El Administrador modifica los campos deseados.
5. El sistema guarda los cambios en la BBDD.
6. Se inicia de nuevo el caso de uso GestionExpertos.

S-3: ConsultarExperto

1. El sistema muestra un listado con todos los expertos creados.
2. El Administrador selecciona un experto.
3. El sistema muestra toda la información referente a dicho experto y los problemas que tiene asignados.
4. Se inicia de nuevo el caso de uso GestionExpertos.

S-4: EliminarExperto

1. El sistema muestra un listado con todos los expertos creados.
2. El Administrador selecciona un experto.
3. El sistema elimina el experto de la BBDD y guarda los cambios.
4. Se inicia de nuevo el caso de uso GestionExpertos.

Condición de salida: La acción elegida se ha llevado a cabo.

Excepciones:

E-1: El identificador introducido para el nuevo experto ya existe, por lo que el sistema muestra un mensaje de error y le pide al Administrador que vuelva a introducir el identificador.

♦ **Caso de uso 4: EstadoDelProblema**

Actores participantes: Administrador.

Condición de entrada: Que el Administrador esté identificado en el sistema y existan problemas asignados a Expertos.

Flujo de eventos:

1. El sistema muestra todos los problemas creados junto con su estado, que puede ser (finalizado o en progreso).
2. El Administrador selecciona un problema.
3. El sistema muestra dos opciones y le pide al Administrador que elija una:
 - Si elige CalcularNivelConsenso, se realiza S-1.
 - Si elige EvoluciónConsenso, se realiza S-2.
 - Si elige Salir, se termina el caso de uso.

Subflujo de eventos:

S-1: CalcularNivelConsenso

Condición de entrada: Que todos los expertos hayan introducido sus preferencias en el problema seleccionado por el Administrador.

1. El sistema realiza la unificación de la información, S-1-1
2. El sistema calcula el grado de consenso, S-1-2
3. El sistema comprueba el grado de consenso.
 - Si es igual o mayor al umbral de consenso se termina el proceso de consenso para este problema.

- Si es menor, comprueba que no se ha alcanzado el número máximo de rondas y se realiza S-1-3.
 - Si se ha alcanzado el número máximo de rondas, se termina el proceso de consenso.
4. El sistema guarda la información en la BBDD.
 5. Se inicia de nuevo el caso de uso EstadoDelProblema.

S-1-1: UnificarInformacion

1. Al tratar con problemas en TDG definidos en contextos heterogéneos con información numérica, intervalar y lingüística, el sistema necesita unificar la información en un espacio de utilidad común, que llamaremos conjunto básico de términos lingüísticos (CBTL).

S-1-2: CalcularConsenso

1. El sistema calcula una matriz de similaridad para cada pareja de expertos, agrega todas las matrices de similaridad y obtiene la matriz de consenso que representa las preferencias de todos los expertos.

S-1-3: GeneraciónRecomendaciones

1. En esta fase se identifican los expertos que deberían cambiar algunos de los valores de sus preferencias, aquellas alternativas donde no hay suficiente consenso (las llamaremos ALT) y por tanto deben ser cambiadas y los pares de alternativas (x_i, x_j) de las alternativas en desacuerdo $x_i \in ALT$ que deben cambiarse. Una vez identificados los expertos y los pares de alternativas que deben cambiarse, el sistema usa un conjunto de reglas de dirección para

sugerir a los expertos cómo cambiar las valoraciones actuales con objeto de incrementar el acuerdo en la siguiente ronda de consenso.

S-2: Evolución Consenso

1. El sistema muestra el grado de consenso de la última ronda realizada. Además, el administrador puede seleccionar cualquier ronda anterior y ver el grado de consenso y el número de cambios que cada experto debe hacer en dicha ronda, de esta forma puede observar como avanza el proceso de consenso. También puede seleccionar un experto asignado al problema y visualizar las preferencias introducidas por dicho experto en una ronda determinada.
2. Se inicia de nuevo el caso de uso EstadoDelProblema.

Condición de salida: La opción elegida se ha llevado a cabo.

◆ Caso de uso 5: ProcesoConsenso

Actores participantes: Experto.

Condición de entrada: Que el experto esté identificado en el sistema y que éste tenga problemas asignados.

Flujo de eventos:

1. El sistema muestra todos los problemas que el experto tiene asignados. Éstos pueden indicar tres posibles estados:
 - Si el sistema está esperando que el experto introduzca su opinión, muestra Introducir opinión.
 - Si está esperando que el resto de expertos introduzcan sus opiniones, muestra Esperando ronda.
 - Si ya ha terminado el proceso de consenso, muestra Finalizado.
2. El Experto selecciona un problema.
3. El sistema pide al experto que elija una de las dos opciones:
 - Si elige IntroducirOpiniones se realiza S-1.
 - Si elige Consultar, se realiza S-2.

Subflujo de eventos:

S-1: IntroducirOpiniones

1. El sistema muestra un formulario correspondiente al problema el cual muestra el número de ronda actual y la matriz de preferencias a rellenar. Si es la segunda ronda o posterior el sistema muestra las recomendaciones generadas para la ronda actual.

2. El experto rellena la matriz y si es la segunda ronda o posterior puede ver las recomendaciones (E-1).
3. El sistema almacena la información en la BBDD.
4. Se inicia de nuevo el caso de uso ProcesoConsenso.

S-2: Consultar

1. Si el experto aún no ha introducido sus preferencias sólo podrá visualizar la descripción del problema. En caso de haber introducido sus preferencias para la primera ronda, podrá visualizar tanto la descripción del problema como las preferencias introducidas en dicha ronda. Si se encuentra ante la segunda ronda o posterior, podrá visualizar también las preferencias de las anteriores rondas y las recomendaciones generadas en caso de que las hubiera.
2. Se inicia de nuevo el caso de uso ProcesoConsenso.

Condición de salida: La opción elegida se ha llevado a cabo.

Excepciones:

E-1: Si los valores introducidos por el experto no se encuentran dentro del dominio ya sea el tipo de información utilizado: numérico, intervalar o lingüístico, el sistema muestra un mensaje de error y le pide al Experto que introduzca de nuevo dicho valor.

3.3.3 Escenarios.

Un caso de uso es una representación abstracta, una abstracción, de una funcionalidad del sistema a realizar. La representación concreta de un caso de uso se realiza mediante la creación de uno o más escenarios que muestren todas las interacciones posibles entre el sistema y sus usuarios.

Los escenarios son historias ficticias que describen posibles interacciones persona-aplicación Web. Permiten a los diseñadores anticiparse a los problemas. Aunque son historias ficticias deben hacerse lo más detalladas posibles, así por ejemplo, los personajes deben tener nombres, motivaciones para usar la interfaz, deben encontrarse en entornos reales con las restricciones que ello conlleva, etc. De esta manera, se facilita a los diseñadores la discusión sobre la interfaz ya que a las personas nos cuesta más trabajo discutir sobre una situación abstracta.

Esta forma de proceder fuerza a los diseñadores a considerar el rango de usuarios que va a usar el sistema y el rango de actividades por las que lo van a usar. Los escenarios permiten hacer diferentes combinaciones de usuarios y actividades de forma que se tengan en cuenta todas las posibilidades.

Un escenario esta formado por los siguientes elementos:

- Un nombre único y unívoco
- Una descripción
- Los actores participantes
- El flujo de eventos

Como se ha indicado, para cada caso de uso puede haber varios escenarios. Para nuestro proyecto se han creado y descrito una cantidad importante de casos de uso. Por lo que, no vamos a definir todos los escenarios de cada uno de ellos sino que, vamos a definir unos pocos que puedan servir como ejemplo de las principales funcionalidades que el sistema va a desarrollar:

Nombre: *CrearProblema_p001*

Descripción: El usuario Rosa, con permiso de *Administrador* crea el problema *p001*.

Actor: *Administrador*.

Flujo de eventos:

1. El usuario Rosa entra en el sistema.
2. El sistema muestra el formulario de entrada y le pide al usuario el nombre de usuario y la contraseña
3. El usuario introduce el nombre de usuario *Rosa* y la contraseña *1234*.
4. El sistema valida el nombre de usuario y la contraseña. El usuario entra en la aplicación como Rosa con permiso de *Administrador*.
5. El sistema muestra el menú principal del administrador. Rosa puede elegir diferentes opciones dentro de cada sección:
 - Gestión de problemas
 - Gestión de expertos
 - Estado del problema
6. Rosa elige la opción *CrearProblema* que se encuentra en la sección Gestión de problemas.
7. Rosa introduce el número de expertos y alternativas que intervienen en el problema.
 - *Expertos: 3*
 - *Alternativas: 3*

8. Rosa introduce:

- *Id del problema*: p001
- *Descripción*: Una empresa va a comercializar una nueva bebida y debe tomar la decisión de elegir el nombre que le van a poner a dicha bebida. Se trata de una bebida elaborada con frutas y otros productos lácteos destinada a niños.
- *Umbral de consenso*: 0.8
- *Nºmax de rondas*:10

9. Rosa define las 3 alternativas:

- FrutaMilk
- MultiFruit
- CaoFruit.

10. Rosa asigna los expertos atorres, jdelgado y martin al problema a partir de la lista de expertos suministrada.

11. Rosa pulsa el botón *Guardar*.

12. El sistema comprueba que se han introducido todos los datos y que éstos son correctos.

13. El sistema envía los datos a la BBDD para actualizar la información.

14. El sistema informa al administrador Rosa de que la información se ha almacenado satisfactoriamente.

Nombre: *CrearExperto_atorres*

Descripción: El usuario Rosa, con permiso de *Administrador* crea el Experto *atorres*.

Actor: *Administrador*.

Flujo de eventos:

1. El usuario Rosa entra en el sistema.
2. El sistema muestra el formulario de entrada y le pide al usuario el nombre de usuario y la contraseña
3. El usuario introduce el nombre de usuario *Rosa* y la contraseña *1234*.
4. El sistema valida el nombre de usuario y la contraseña. El usuario entra en la aplicación como Rosa con permiso de *Administrador*.
5. El sistema muestra el menú principal del administrador. El administrador puede elegir diferentes opciones dentro de cada sección:
 - Gestión de problemas
 - Gestión de expertos
 - Estado del problema
6. Rosa elige la opción *CrearExperto* que se encuentra en la sección Gestión de expertos.
7. El sistema muestra un formulario con los campos que hay que rellenar: identificador, contraseña, especialidad, dirección de correo electrónico y tipo de información que va a utilizar el experto para introducir sus preferencias.

8. Rosa introduce todos los datos del Experto:

- *Nombre*: atorres.
- *Contraseña*: 4321
- *Especialidad*: Informática
- *Dirección de correo*: atorres@hotmail.com.
- *Tipo de Información*: Lingüístico

9. Rosa pulsa el botón *Guardar*.

10. El sistema comprueba que se han introducido todos los datos y que no existe en la BBDD otro Experto con el identificador *atorres*.

11. El sistema envía los datos a la BBDD para actualizar la información.

12. El sistema informa al administrador Rosa de que la información se ha almacenado satisfactoriamente.

Nombre: *EvolucionConsenso_p001*

Descripción: El usuario Rosa, con permiso de *Administrador* consulta el nivel de consenso del problema *p001*.

Actor: Administrador

Flujo de eventos:

1. El usuario Rosa entra en el sistema.
2. El sistema muestra el formulario de entrada y le pide al usuario el nombre de usuario y la contraseña
3. El usuario introduce el nombre de usuario Rosa y la contraseña *1234*.
4. El sistema valida el nombre de usuario y la contraseña. El usuario entra en la aplicación como Rosa con permiso de *Administrador*.
5. El sistema muestra el menú principal del administrador. El administrador puede elegir diferentes opciones dentro de cada sección:
 - Gestión de problemas
 - Gestión de expertos
 - Estado del problema
6. Rosa elige la opción *Estado del problema*.
7. El sistema muestra todos los problemas creados, en este caso sólo uno, *p001*.
8. Rosa selecciona el problema *p001*.
9. El sistema muestra dos posibles opciones:
 - Calcular nivel consenso
 - Evolución consenso

10. Rosa elige la opción *Evolución consenso*.
11. El sistema muestra el consenso de la última ronda, gr. *consenso ronda 1: 0.64*
12. Rosa puede seleccionar cualquier ronda anterior y ver el consenso y el número de cambios que cada experto ha realizado en dicha ronda. También puede escoger un experto asignado al problema *p001*, por ejemplo *atorres* y ver las preferencias introducidas por dicho experto.

Ronda 1:

$$P_{atorres} = \begin{pmatrix} - & m & i \\ pm & - & m \\ m & mm & - \end{pmatrix}$$

Nombre: *IntroducirOpiniones_p001*

Descripción: El usuario *atorres* desea introducir sus preferencias sobre el problema *p001*.

Actor: Experto cuyo identificador es *atorres*.

Flujo de eventos:

1. El usuario *atorres* entra en el sistema.
2. El sistema muestra el formulario de entrada y le pide al usuario el nombre de usuario y la contraseña
3. El usuario introduce el nombre de usuario *atorres* y la contraseña *4321*.
4. El sistema valida el nombre de usuario y la contraseña. El usuario entra en la aplicación como *atorres*.
5. El sistema muestra el menú principal del Experto cuyo identificador es *atorres*.
6. El sistema se comunica con la BBDD y obtiene el listado de problemas asignados al Experto *atorres*. El único problema asignado es *p001* cuyo estado es "Introducir opinion".
7. *atorres* selecciona el problema *p001*.
8. El sistema muestra dos posibles opciones:
 - Introducir Opiniones
 - Consultar
9. *atorres* elige *Introducir Opiniones*.
10. El sistema muestra una matriz de preferencias vacía.

11. *atorres* rellena la matriz de preferencias con valores lingüísticos:

Ronda actual: 1

$$P_{atorres} = \begin{pmatrix} - & m & i \\ pm & - & m \\ m & mm & - \end{pmatrix}$$

Si es la segunda ronda o posterior podrá ver las recomendaciones generadas por el sistema. Las recomendaciones indican las alternativas en las que hay que modificar las preferencias y la dirección, que puede ser incrementar (indicado en verde) o decrementar (indicado en rojo) las preferencias.

12. *atorres* pulsa el botón *Enviar*.

13. El sistema valida los datos suministrados por el experto.

14. El sistema envía los datos a la BBDD para actualizar la información.

15. El sistema informa al Experto *atorres* de que la operación ha finalizado con éxito.

Nombre: *ConsultarEleccion_p001*

Descripción: El usuario *atorres* desea consultar sus preferencias introducidas en el problema *p001*.

Actor: Experto cuyo identificador es *atorres*.

Flujo de eventos:

1. El usuario *atorres* entra en el sistema.
2. El sistema muestra el formulario de entrada y le pide al usuario el nombre de usuario y la contraseña
3. El usuario introduce el nombre de usuario *atorres* y la contraseña *4321*.
4. El sistema valida el nombre de usuario y la contraseña. El usuario entra en la aplicación como *atorres*.
5. El sistema muestra el menú principal del experto cuyo identificador es *atorres*.
6. El sistema se comunica con la BBDD y obtiene el listado de problemas asignados al experto *atorres*. El único problema que tiene asignado es *p001*.
7. *atorres* selecciona el problema *p001*.
8. El sistema muestra dos posibles opciones:
 - Introducir Opiniones.
 - Consultar.
9. *atorres* elige *Consultar*.
10. El sistema muestra toda la información referente al problema *p001* y las preferencias introducidas por el experto *atorres* en la ronda actual.

- *Id del problema*: p001.
- *Descripción*: Una empresa va a comercializar una nueva bebida y debe tomar la decisión de elegir el nombre que le van a poner a dicha bebida. Se trata de una bebida elaborada con frutas y otros productos lácteos destinada a niños.
- *Estado del problema*: Progreso.
- *Número de expertos asignados*: 3
- *Número máximo de rondas*: 10
- *Ronda actual*: 1
- *Alternativas*:
 - Frutamilk
 - MultiFruit
 - Caofruit.

Preferencias:

Ronda: 1

$$P_{\text{torres}} = \begin{pmatrix} - & m & i \\ pm & - & m \\ m & mm & - \end{pmatrix}$$

Si estamos ante la segunda ronda o posterior el sistema mostrará las recomendaciones generadas para la ronda actual, las cuales se indicaran mediante colores: rojo decrementar y verde incrementar. Además el usuario podrá seleccionar una ronda anterior y visualizar las preferencias y recomendaciones de dicha ronda.

3.4 Diseño del sistema.

Sin duda, realizar de manera adecuada cada una de las actividades que conlleva la Ingeniería del Software es indispensable para la realización de un proyecto software de calidad. Por lo tanto, no se puede decir que ninguna de estas actividades sea más importante que otra. Sin embargo, si podemos decir que la actividad de diseño es la más delicada y la más laboriosa de llevar a cabo.

- Es delicada porque si no se lleva a cabo correctamente se hace complejo codificar de manera correcta el modelo obtenido en el análisis del sistema. Lo que puede hacer en vano todo el esfuerzo realizado durante las primeras actividades de la Ingeniería del Software.
- Y es laboriosa porque las estrategias para conseguir que esta traducción entre modelo y código se lleve a cabo correctamente son muy diversas y complejas.

Se puede decir, por tanto, que el diseño del sistema es la actividad de la Ingeniería del Software en la que se identifican los objetivos finales del sistema, y se plantean las diversas estrategias para alcanzarlos en la actividad de implementación.

Sin embargo, el sistema no se suele diseñar de una sola vez sino que hay que diferenciar entre el diseño y la estructura de los datos que se van a manejar y, el diseño de la interfaz entre la aplicación y el usuario. Estas dos fases del diseño no se realizan de forma consecutiva una detrás de la otra sino que lo normal es realizarlas paralelamente y finalizarlas a la vez.

Las distintas fases de diseño que veremos a continuación son:

- **Diagrama de clases:** muestran la estructura del sistema.
- **Diseño de los datos:** estructura de los datos.
- **Diseño de la interfaz:** se define la apariencia visual de la aplicación.

3.4.1 Diagrama de clases.

Los diagramas de clases se utilizan para mostrar la estructura estática del sistema modelado. Pueden estar formados por clases, interfaces, paquetes, relaciones e incluso instancias, como objetos o enlaces.

Son una potente herramienta de diseño, ayudando a los desarrolladores a planificar y establecer la estructura del sistema y subsistema antes de escribir ningún código. Esto permite asegurar que el sistema está bien diseñado desde el principio.

Son utilizados en la fase de diseño prácticamente en la totalidad de los sistemas que utilizan UML para su modelado.

Los diagramas de clases tienen los siguientes componentes:

- **Clases:** son los componentes fundamentales de los diagramas de clase. Su notación general es un rectángulo dividido en tres secciones, mostrando la primera el nombre de la clase, la siguiente los atributos y la última las operaciones.
- **Relaciones:** es una conexión semántica entre elementos. Existen cuatro tipos principales de relaciones:
 - *Generalización:* es una relación de especialización.
 - *Asociación:* es una relación estructural. Existen dos subtipos, la agregación y la composición.
 - *Realización:* es una relación contractual, en la cual una clase especifica un contrato que otra clase garantiza que cumplirá (por ejemplo, una interface).
 - *Dependencia:* Es una relación de uso.

Una vez descritos brevemente los componentes de un diagrama de clases, vamos a proceder a analizar los propios del sistema.

3.4.1.1 Diagrama completo de clases.

Antes de mostrar el diagrama de clases, vemos apropiado explicar como va a ser el funcionamiento o la lógica de negocio del sistema, para ver más claramente el paso de información o comunicación entre las distintas capas diseñadas. A la hora de diseñar una aplicación con una interfaz gráfica de usuario es importante seguir el esquema de diseño *Modelo-Vista-Controlador* (MVC). En este esquema se definen tres roles bien diferenciados:

- **Modelo:** el modelo es la representación de la información de un problema.
- **Vista:** una vista es una posible visualización de la información contenida en un modelo. Un modelo puede tener varias vistas definidas.
- **Controlador:** el controlador se encarga de coordinar la interacción entre las vistas y los modelos. Cada vez que un modelo cambia internamente, actualiza sus vistas.

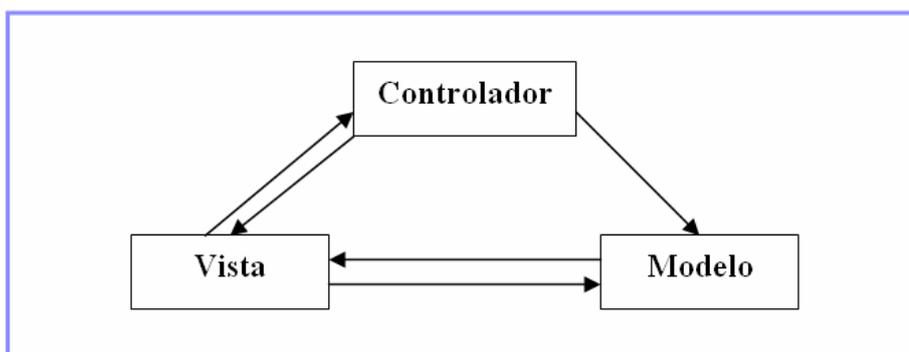


Figura 3.6: Esquema de la arquitectura MVC

El mensaje más importante que proporciona este esquema de diseño es, que hay que mantener al modelo lo más independiente posible de la vista y el controlador, de forma que se pueda cambiar uno u otros sin necesidad de realizar ninguna modificación en los demás componentes.

En nuestra aplicación tendremos una capa modelo de la información que representará todos los datos del sistema. Por lo tanto, esta capa es la encargada de obtener los datos necesarios de la BBDD. Esta información será gestionada por una serie de Servlets que harán la función de controlar el

sistema y realizar cualquier operación necesaria. Finalmente, cualquier representación gráfica de datos en la pantalla será lanzada por los Servlets utilizando para ello llamadas a páginas JSP haciendo éstas últimas la labor de la Vista de la aplicación.

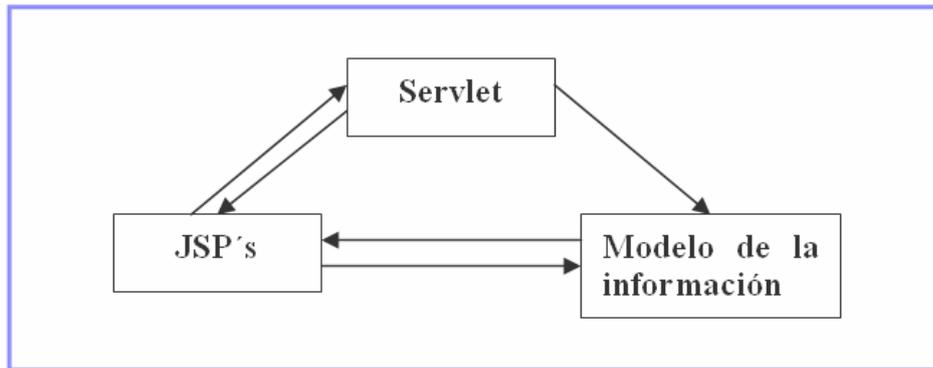


Figura 3.7: Esquema de la aplicación

El diagrama completo de clases se muestra en la Figura 3.8.

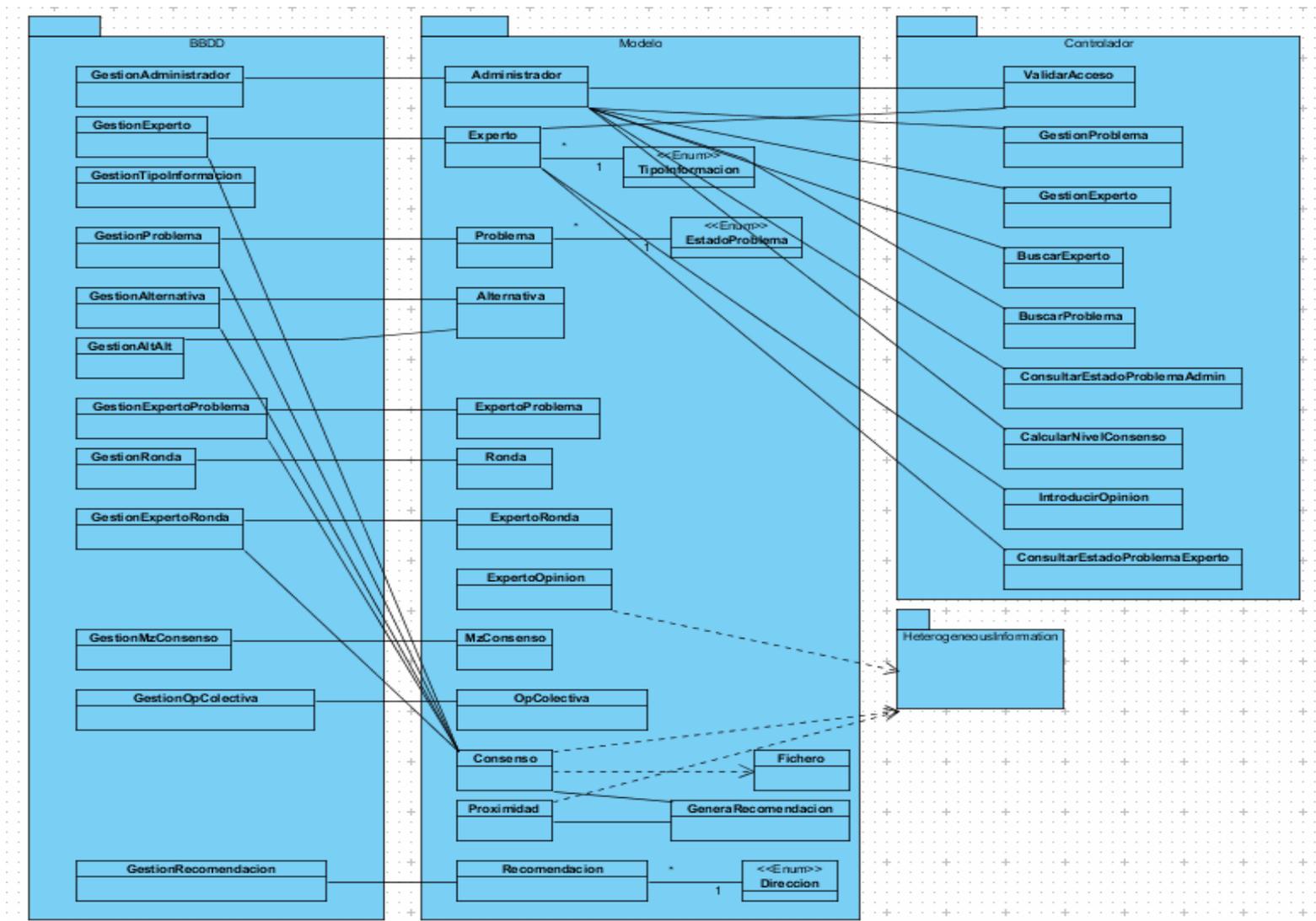


Figura 3.8: Diagrama de clases

Analizamos cada uno de los paquetes de nuestro diagrama de clases:

- **Paquete BBDD (Base de Datos):** es el paquete al que hacen referencia todas las clases pertenecientes al modelo, para obtener la conexión a la BBDD.
- **Paquete Modelo:** Implementa la lógica de la aplicación, es decir, que almacena todos los datos, el estado de la aplicación y tiene los métodos que manipulan esos datos.
- **Paquete Controlador:** la capa controladora, que en nuestro caso está formada por Servlets, llevará a cabo todo el control del sistema y permitirá al usuario realizar cualquier operación sobre éste. La capa de negocio, utilizará constantemente clases del paquete Modelo para obtener y guardar información, y será la encargada de operar con la información para mostrarla en la Vista, que en nuestro caso serán JSP's.
- **Paquete HeterogeneousInformation:** este paquete contiene todo lo necesario para el manejo de la información heterogénea.

3.4.1.2 Diagrama por paquetes.

En los siguientes apartados vamos a ver con más detalle los paquetes de nuestra aplicación.

3.4.1.2.1 Paquete BBDD.

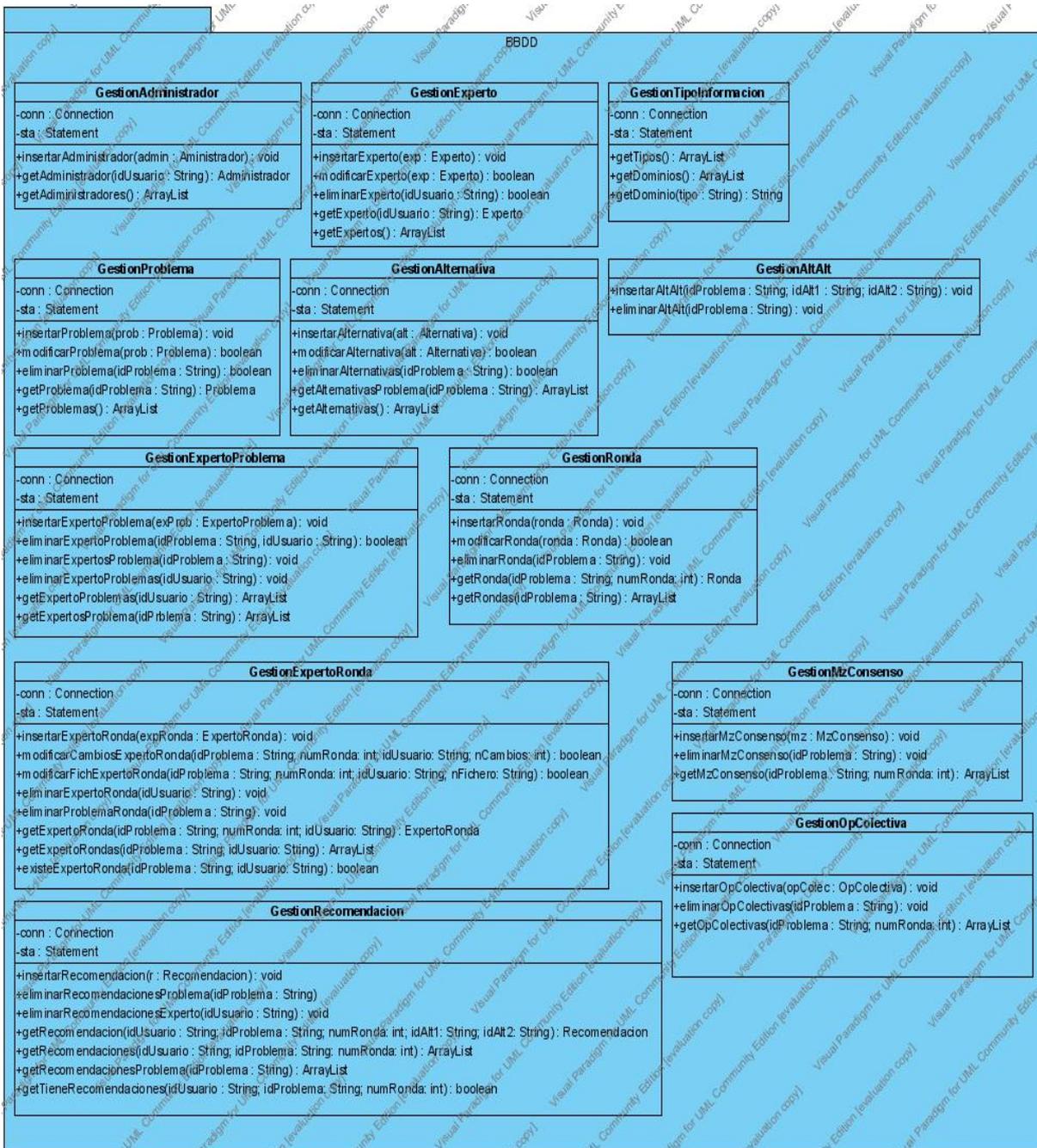


Figura 3.9: Paquete BBDD

3.4.1.2.2 Paquete Modelo.

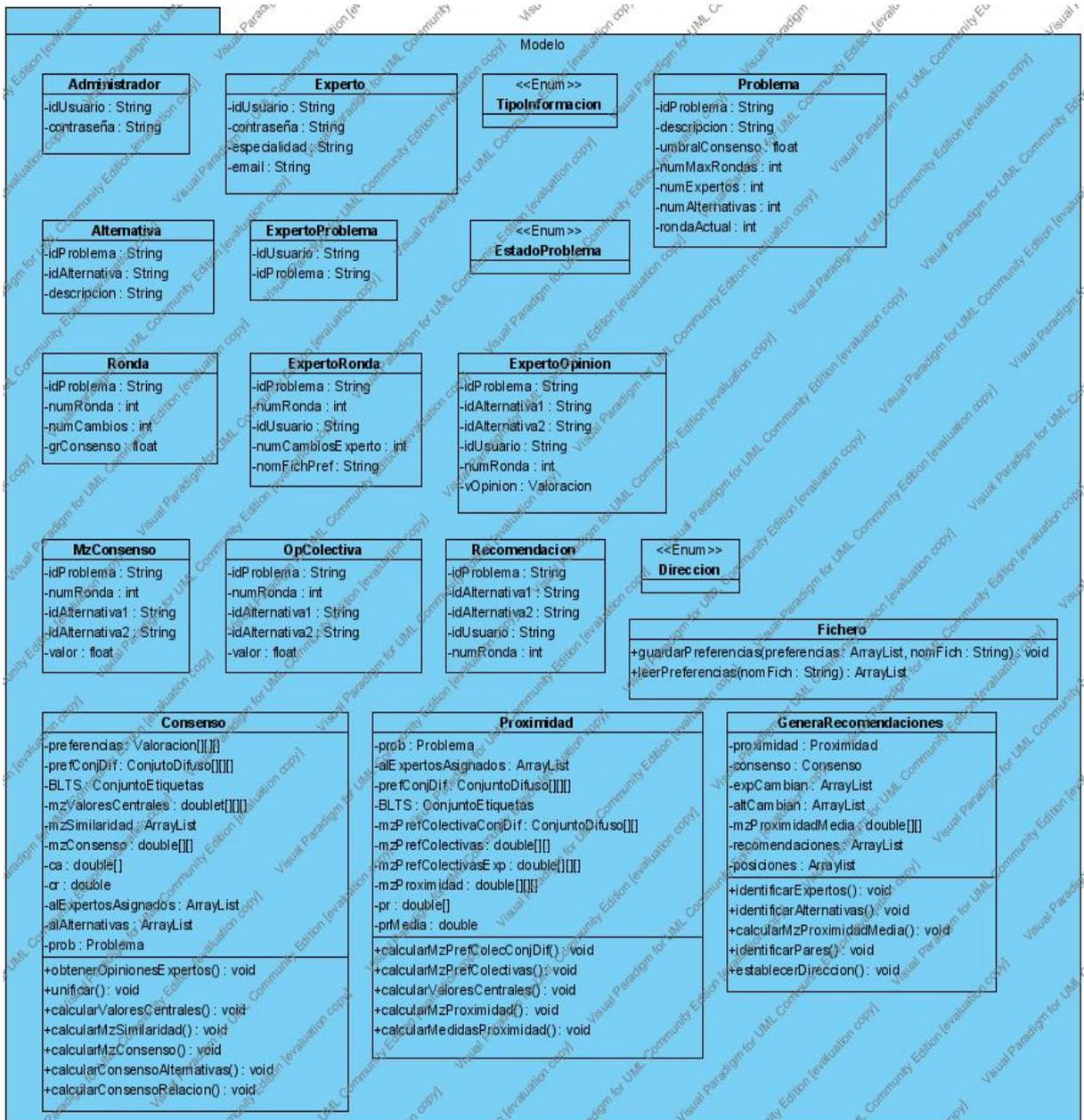


Figura 3.10: Paquete Modelo

3.4.1.2.3 Paquete HeterogeneousInformation.

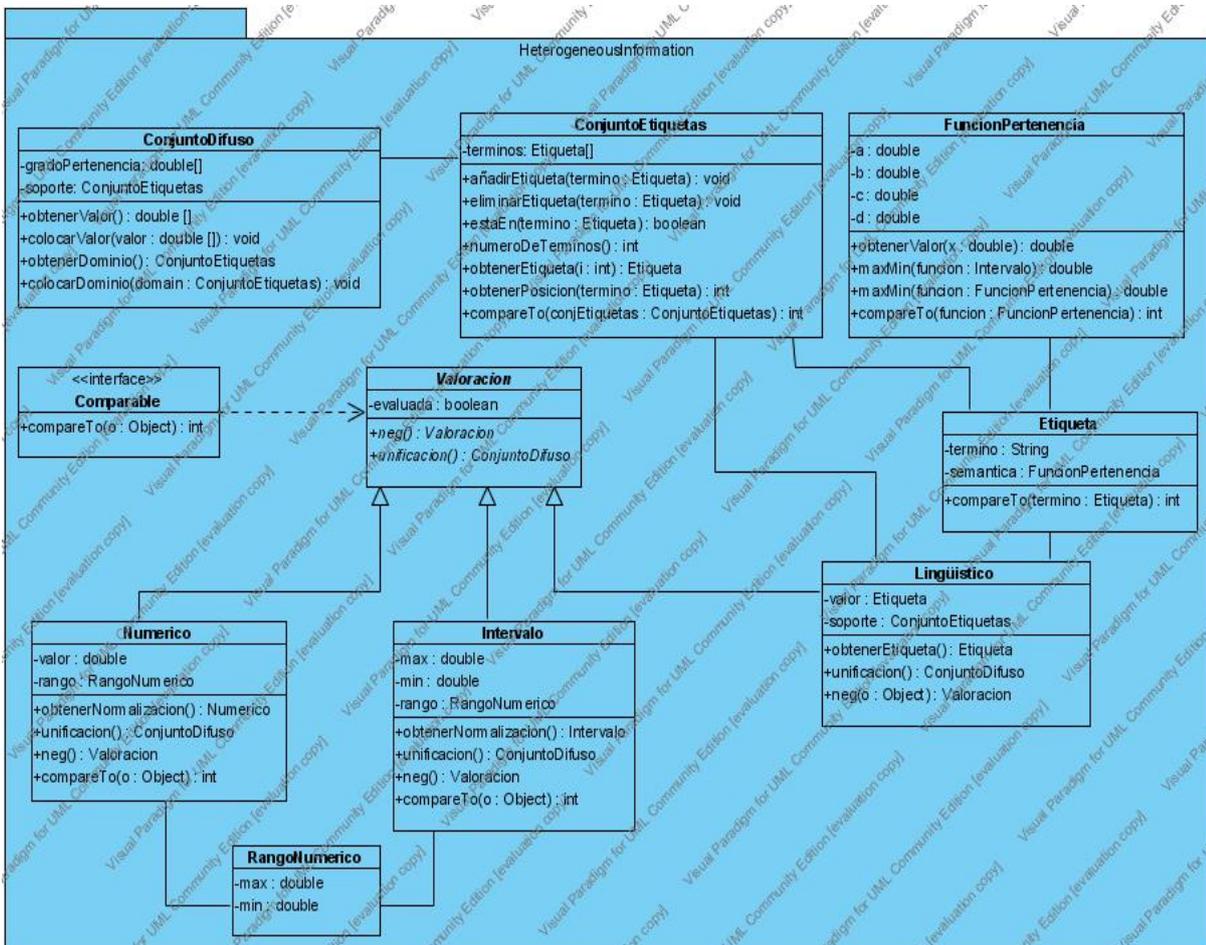


Figura 3.11: Paquete HeterogeneousInformation

3.4.1.2.4 Paquete Controlador.

Controlador
<p>ValidarAcceso</p> <pre>+processRequest(HttpServletRequest request, HttpServletResponse response): void +doGet(HttpServletRequest request, HttpServletResponse response): void +doPost(HttpServletRequest request, HttpServletResponse response): void +comprobarAdmin(usuario: String; pw: String): boolean +comprobarExp(usuario: String; pw: String): boolean</pre>
<p>GestionExperto</p> <pre>+processRequest(HttpServletRequest request, HttpServletResponse response): void +doGet(HttpServletRequest request, HttpServletResponse response): void +doPost(HttpServletRequest request, HttpServletResponse response): void +comprobarCamposVacios(HttpServletRequest request, HttpServletResponse response): boolean +comprobarIdExperto(HttpServletRequest request, HttpServletResponse response): boolean +comprobarContraseña(HttpServletRequest request, HttpServletResponse response): boolean +comprobarCorreo(HttpServletRequest request, HttpServletResponse response): boolean +notificarError(HttpServletRequest request, HttpServletResponse response): void</pre>
<p>GestionProblema</p> <pre>+processRequest(HttpServletRequest request, HttpServletResponse response): void +doGet(HttpServletRequest request, HttpServletResponse response): void +doPost(HttpServletRequest request, HttpServletResponse response): void +comprobarCamposVacios(HttpServletRequest request, HttpServletResponse response): boolean +comprobarIdProblema(HttpServletRequest request, HttpServletResponse response): boolean +comprobarUmbral(HttpServletRequest request, HttpServletResponse response): boolean +comprobarRondas(HttpServletRequest request, HttpServletResponse response): boolean +comprobarAlternativas(HttpServletRequest request, HttpServletResponse response): void +comprobarExpertos(HttpServletRequest request, HttpServletResponse response): boolean +añadirAlternativa(HttpServletRequest request, HttpServletResponse response): void +añadirExperto(HttpServletRequest request, HttpServletResponse response): void +notificarError(HttpServletRequest request, HttpServletResponse response): void</pre>
<p>BuscarExperto</p> <pre>+processRequest(HttpServletRequest request, HttpServletResponse response): void +doGet(HttpServletRequest request, HttpServletResponse response): void +doPost(HttpServletRequest request, HttpServletResponse response): void</pre>
<p>BuscarProblema</p> <pre>+processRequest(HttpServletRequest request, HttpServletResponse response): void +doGet(HttpServletRequest request, HttpServletResponse response): void +doPost(HttpServletRequest request, HttpServletResponse response): void</pre>
<p>CalcularIvdelConsenso</p> <pre>+processRequest(HttpServletRequest request, HttpServletResponse response): void +doGet(HttpServletRequest request, HttpServletResponse response): void +doPost(HttpServletRequest request, HttpServletResponse response): void +estaEnProgreso(idProblema: String): boolean +comprobarTodasPreferenciasIntroducidas(expNoPreferencias: ArrayList): boolean</pre>
<p>ConsultarEstadoProblemaAdmin</p> <pre>+processRequest(HttpServletRequest request, HttpServletResponse response): void +doGet(HttpServletRequest request, HttpServletResponse response): void +doPost(HttpServletRequest request, HttpServletResponse response): void +existenPreferencias(idProblema: String; numRonda: int; idExperto: String): boolean +problemaFinalizado(idProblema: String): boolean</pre>
<p>IntroducirOpinion</p> <pre>+processRequest(HttpServletRequest request, HttpServletResponse response): void +doGet(HttpServletRequest request, HttpServletResponse response): void +doPost(HttpServletRequest request, HttpServletResponse response): void +comprobarErrores(HttpServletRequest request, HttpServletResponse response, idProblema: String): ArrayList +existenErrores(errores: ArrayList): boolean</pre>
<p>ConsultarEstadoProblemaExperto</p> <pre>+processRequest(HttpServletRequest request, HttpServletResponse response): void +doGet(HttpServletRequest request, HttpServletResponse response): void +doPost(HttpServletRequest request, HttpServletResponse response): void +comprobarIntroducirPreferencias(idProblema: String; idExperto: String): int +existenPreferencias(idProblema: String; numRonda: int; idExperto: String): boolean +existenRecomendaciones(idProblema: String; numRonda: int; idExperto: String): boolean +problemaFinalizado(idProblema: String): boolean</pre>

Figura 3.12: Paquete Controlador

3.4.2 Diseño de los datos.

La intención de esta fase del diseño software es determinar la estructura que poseen cada uno de los elementos de información del sistema, es decir, la estructura de los datos sobre los que va a trabajar. Estos elementos son:

- *Administrador*, conocemos su nombre y contraseña.
- *Expertos*, conocemos su nombre, contraseña, especialidad y dirección de correo electrónico.
- *Alternativas*, conocemos su código y descripción.
- *Problemas*, conocemos su código, descripción, umbral de consenso, nº máximo de rondas, nº de expertos asignados, nº de alternativas definidas, ronda actual y estado actual.
- *Ronda*, conocemos su código, nº de cambios totales y grado de consenso.
- *Tipo de información*, conocemos el dominio de cada uno de los tres tipos de información (numérico, intervalar, lingüístico).
- *Matriz de consenso*, conocemos todos los valores de la matriz de consenso.
- *Matriz de opiniones colectivas*, conocemos todos los valores de la matriz de opiniones colectivas.
- *Recomendación*, conocemos el usuario, problema, nº de ronda, par de alternativas y dirección.

Una vez determinados cuales son los elementos de información del sistema, se debe definir su representación en forma de tablas de una base de datos. Para ello, se realiza en primer lugar un

diseño conceptual de la base de datos y, posteriormente, se obtienen las tablas requeridas. Para realizar este diseño conceptual se utilizará el modelo Entidad-Relación.

Modelo Entidad-Relación

El modelo Entidad-Relación (también conocido por sus iniciales: E-R) es una técnica de modelado de datos que utiliza diagramas entidad-relación. No es la única técnica de modelado pero si es la más extendida y utilizada.

Un diagrama entidad-relación está compuesto por tres tipos de elementos principales:

- **Entidades:** objetos (cosas, conceptos o personas) sobre los que se tiene información. Se representan mediante rectángulos etiquetados en su interior con un nombre. Una instancia es cualquier ejemplar concreto de una entidad.
- **Relaciones:** interdependencias entre uno o más entidades. Se representan mediante rombos etiquetados en su interior con un verbo. Si la relación es entre una entidad consigo mismo se denomina reflexiva, si es entre dos entidades se denomina binaria, ternaria si es entre tres y múltiple si es entre más.
- **Atributos:** características propias de una entidad o relación. Se representan mediante elipses etiquetados en su interior con un nombre.

En los diagramas entidad-relación también hay que tener en cuenta otros aspectos como pueden ser:

- **Entidades débiles:** son aquellas que no se pueden identificar unívocamente solo con sus atributos, es decir, necesitan estar relacionadas con otras entidades para existir. Se representan con dos rectángulos concéntricos de distinto tamaño y un nombre en el interior del más pequeño.

- **Cardinalidad de las relaciones:** existen tres tipos de cardinalidades en una relación, según el número de instancias de cada entidad que involucren:
 - *Uno a uno:* una instancia de la entidad A se relaciona solamente con una instancia de la entidad B. (1:1)
 - *Uno a muchos:* cada instancia de la entidad A se relaciona con varias de la entidad B. (1:*)
 - *Muchos a muchos:* cualquier instancia de la entidad A se relaciona con cualquier instancia de la entidad B. (*:*)

- **Claves:** cada entidad de un diagrama entidad-relación debe tener una clave que identifique unívocamente cada instancia del resto. Debe estar formada por uno o varios de sus atributos.

- **Claves foráneas:** Es aquella combinación de atributos de una entidad, que es clave en otra entidad distinta.

Una vez conocidos los elementos que forman parte de un diagrama entidad-relación podemos empezar a desarrollar el modelo entidad-relación. Los pasos a seguir son los siguientes:

1. Convertir el enunciado del problema (o, como es nuestro caso, los elementos del sistema software) en un Esquema Conceptual del mismo.
2. Convertir este Esquema Conceptual (o EC) en uno más refinado conocido como Esquema Conceptual Modificado (ECM).
3. Obtener las tablas de la base de datos a partir del Esquema Conceptual Modificado y normalizarlas.

Los conceptos básicos del *modelo E-R* pueden modelar la mayoría de las situaciones, pero algunos aspectos se modelan mejor con el *modelo E-R extendido*, el cual aporta nuevas características, como:

- **Generalización**
- **Agregación**
- **Especialización**
- **Herencia de atributos**

Vamos a definir únicamente el concepto de agregación, ya que en el Esquema Conceptual haremos uso de él.

La *agregación* ayuda a construir entidades de niveles superiores y consiste en considerar un conjunto de componentes (tipos de entidades o tipos de relaciones) como si fueran un único tipo de entidades. Se representa incluyendo en un rectángulo todos los componentes de la agregación.

Normalización en el modelo Entidad-Relación

La normalización es un proceso consistente en imponer a las tablas ciertas restricciones mediante una serie de transformaciones consecutivas. Con ello se asegura que las tablas contengan los atributos necesarios y suficientes para describir la realidad de la entidad que representan, separando aquellos que pueden contener información cuya relevancia permite la creación de otra nueva tabla.

Para asegurar la normalización *Codd* estableció tres formas normales, las cuales hacen que una base de datos (si las cumple) esté normalizada.

Estas formas normales son:

- **Primera forma Normal (FN1)**: Una tabla está en FN1 si todos los atributos no clave, dependen funcionalmente de la clave, o lo que es lo mismo, no existen grupos repetitivos para un valor de clave.
- **Segunda forma Normal (FN2)**: Una tabla está en FN2 si está en FN1 y además todos los atributos que no pertenecen a la clave dependen funcionalmente de forma completa

de ella. De esta definición se desprende que una tabla en FN1 y cuya clave esta compuesta por un único atributo está en FN2.

- **Tercera forma Normal (FN3):** Una tabla está en FN3 si está en FN2 y además no existen atributos no clave que dependan transitivamente de la clave.

3.4.2.1 Esquema Conceptual.

Necesitamos convertir nuestros elementos del sistema en entidades o relaciones. Es obvio que Administrador, Expertos, Alternativas, Problemas, Ronda, Tipo de Información, Matriz de consenso, Matriz de opiniones colectivas y Recomendación se convertirán en Entidades en nuestro Esquema Conceptual.

En cuanto a las relaciones:

- En un Problema participan un conjunto de Expertos, y un Experto puede estar asignado en más de un problema (relación de muchos a muchos).
- Un Experto utiliza un Tipo de información para introducir sus preferencias, y un Tipo de información puede ser utilizada por varios Expertos (relación de uno a muchos)
- Un Problema está formado por un conjunto de Alternativas, y una Alternativa puede estar incluida en un solo problema (relación de uno a muchos).
- Una Alternativa se relaciona con muchas Alternativas, (relación reflexiva de muchos a muchos).
- Un Experto participa en muchas Rondas, y en una Ronda participan muchos Expertos (relación de muchos a muchos).
- Un Experto puede obtener varias Recomendaciones en distintos pares de Alternativas, y en un par de Alternativas se pueden generar distintas Recomendaciones para distintos

Expertos, (agregación en la relación Alternativa y Experto, ésta a su vez se relaciona con Recomendación).

Estas entidades con sus atributos y sus relaciones se puede observar en la Figura 3.13

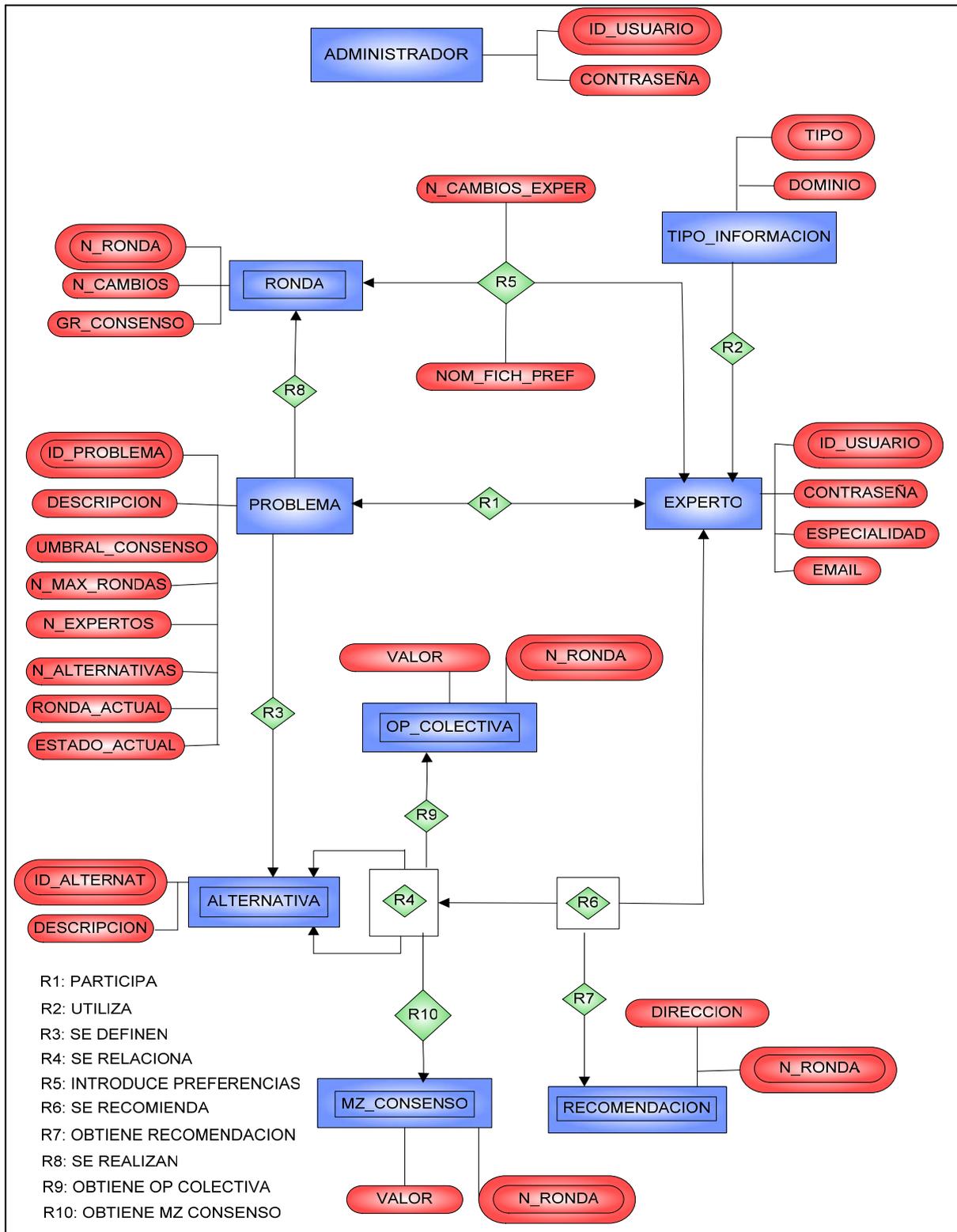


Figura 3.13: Esquema Conceptual

3.4.2.2 Esquema Conceptual Modificado.

Para obtener el Esquema Conceptual Modificado a partir del Esquema Conceptual se deben hacer los cambios que siguen a continuación:

- Eliminar todas las entidades débiles.
- Eliminar las relaciones de muchos a muchos.
- Eliminar las relaciones con atributos que haya en nuestro Esquema Conceptual.

Por lo tanto, nuestro Esquema Conceptual Modificado (ECM) quedaría como muestra la Figura 3.14, y tendríamos cuatro entidades más:

- **Prob_Exp**, que servirá al Administrador para ver los Expertos asignados a cada problema.
- **Alt_Alt**, en la que se almacenan los identificadores de cada par de alternativas que forman parte de un determinado Problema.
- **Alt_Alt_Exp_Recomendacion**, que almacena el identificador de un Problema, par de Alternativas e identificador de Experto.
- **ExpertoRonda**, que almacena el identificador de Problema, el identificador de Usuario, el n^o de ronda, el nombre del fichero que contiene las preferencias del experto y el n^o de cambios que debe realizar dicho experto en la ronda.

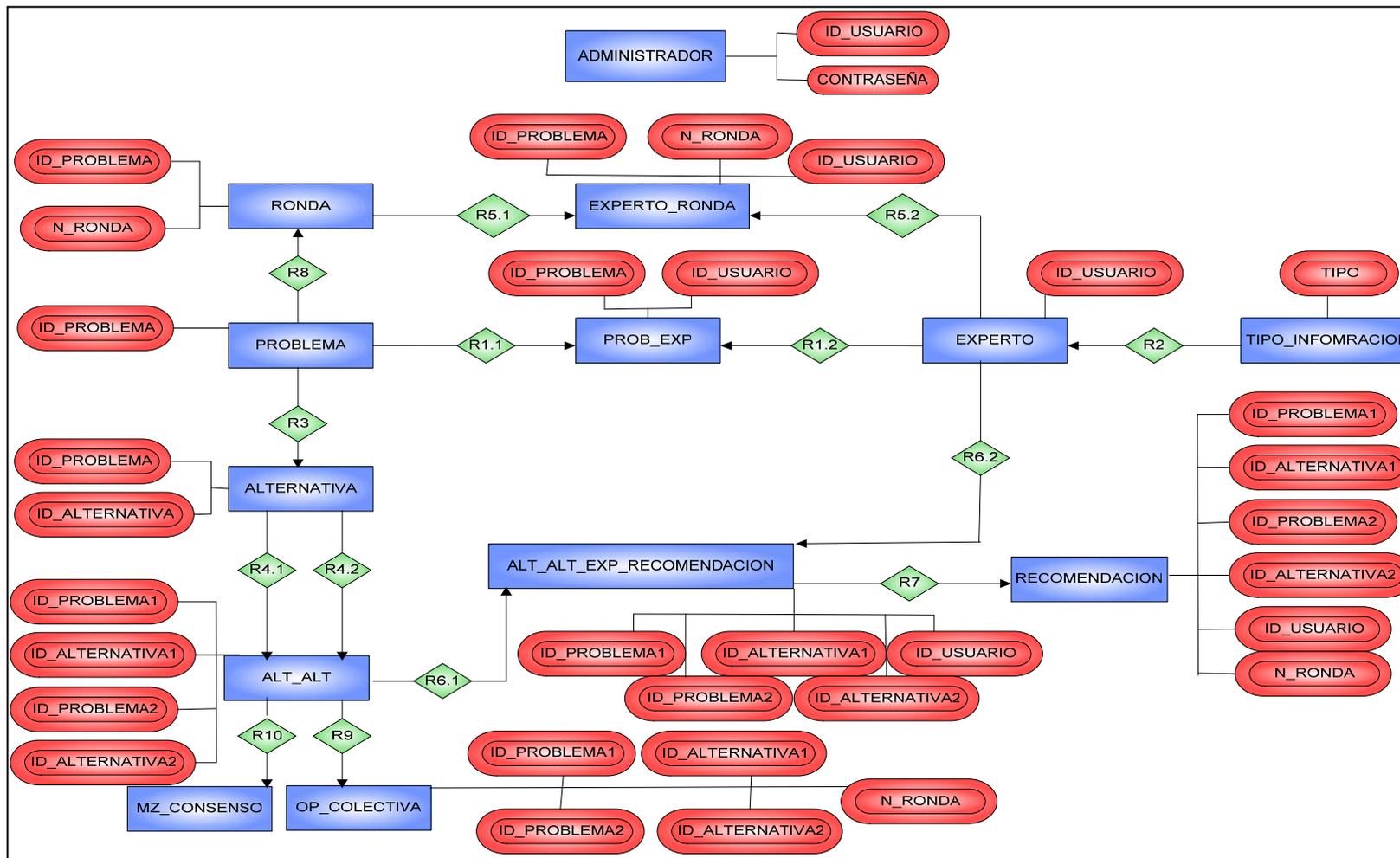


Figura 3.14: Esquema Conceptual Modificado

Las relaciones del Esquema Conceptual Modificado son las siguientes:

- **R1.1 y R1.2:** PARTICIPA
- **R2:** UTILIZA
- **R3:** SE DEFINEN
- **R4.1 y R4.2:** SE RELACIONA
- **R5.1 y R5.2:** SE INTRODUCEN PREFERENCIAS e INTRODUCE PREFERENCIAS
- **R6.1 y R6.2:** SE RECOMIENDA
- **R7:** OBTIENE RECOMENDACION
- **R8:** SE REALIZAN
- **R9:** OBTIENE OP_COLECTIVAS
- **R10:** OBTIENE MZ_CONSENSO

Se puede observar en el ECM que la entidad ALT_ALT_EXP_RECOMENDACION almacena únicamente la clave, por lo que la podemos eliminar dejando sólo la entidad RECOMENDACION.

La entidad RECOMENDACIÓN está formada por (ID_PROBLEMA1, ID_ALTERNATIVA1, ID_PROBLEMA2, ID_ALTERNATIVA2, ID_USUARIO, N°RONDA, DIRECCION).

Finalmente el Esquema Conceptual Modificado se reduce y quedaría de la siguiente forma:

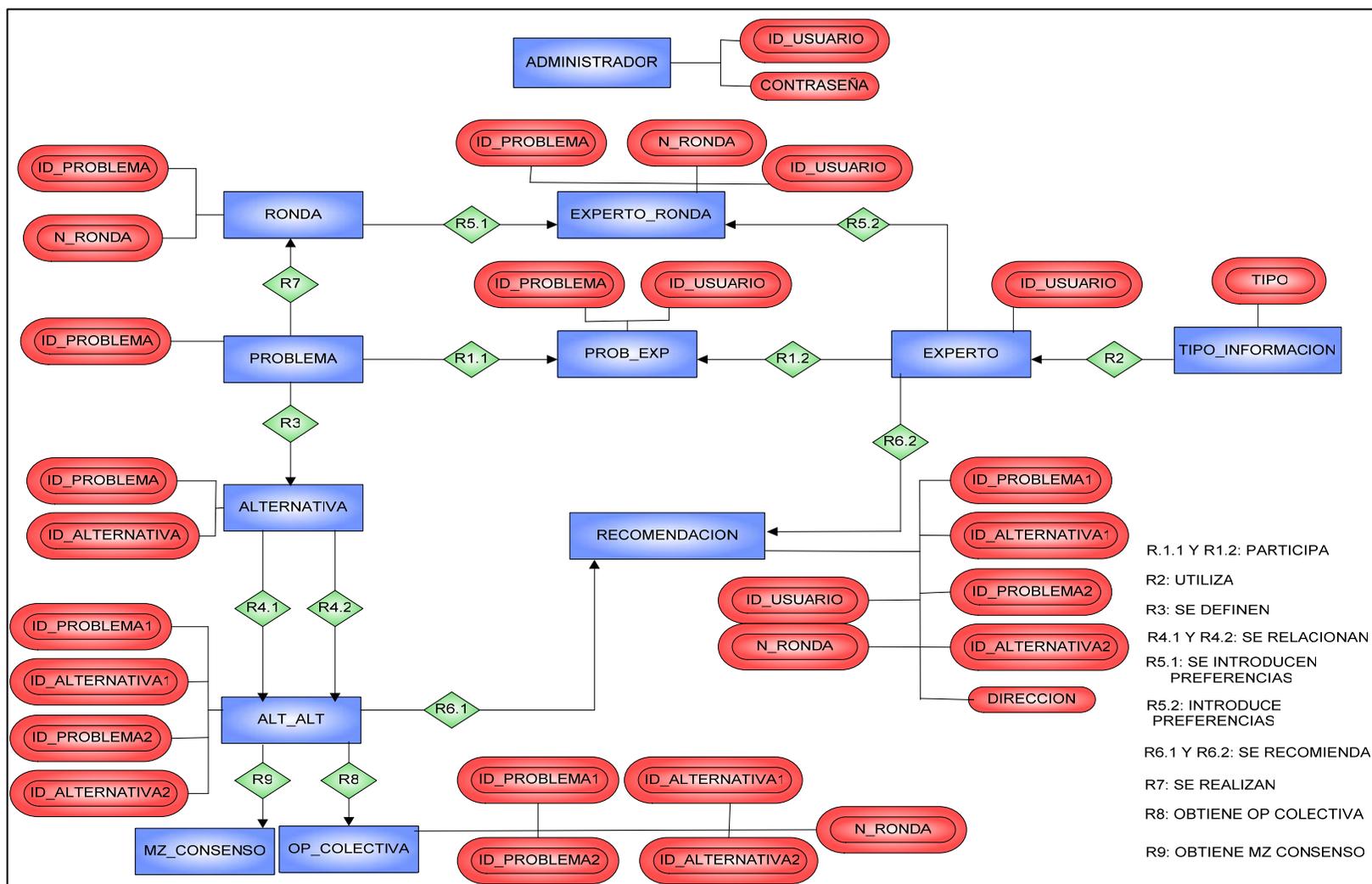


Figura 3.15: Esquema Conceptual Modificado Simplificado

3.4.2.3 Tablas de aplicación.

A partir del ECM obtenido previamente podemos determinar las tablas de la base de datos, teniendo en cuenta que:

- Cada entidad del ECM se transforma en una tabla.
- Los atributos de una entidad se convierten en los campos de las tablas respectivas.

Por lo tanto obtendremos las siguientes tablas: *ADMINISTRADOR*, *EXPERTO*, *PROBLEMA*, *PROB_EXP*, *TIPO_INFORMACIÓN*, *ALTERNATIVA*, *ALT_ALT*, *RECOMENDACIÓN*, *RONDA*, *EXPERTO_RONDA*, *OP_COLECTIVA* y *MZ_CONSENSO*. A continuación se detallan cada una de estas tablas.

ADMINISTRADOR: Tabla que contiene el conjunto de administradores del sistema, es decir, aquellos usuarios que crean los problemas, los expertos y controlan el proceso de consenso.

CAMPO	TIPO	DESCRIPCION	CLAVE	REQUERIDO
ID_USUARIO	Varchar(10)	Es el identificador del usuario administrador	*	Si
CONTRASEÑA	Varchar(15)	Es la contraseña del usuario administrador		Si

Tabla 1: Campos de la tabla ADMINISTRADOR

EXPERTO: Tabla que contiene el conjunto de expertos del sistema. Estos son los usuarios que introducen sus preferencias en los problemas que tienen asignados.

CAMPO	TIPO	DESCRIPCION	CLAVE	REQUERIDO
ID_USUARIO	Varchar(10)	Es el identificador del usuario administrador	*	Si
CONTRASEÑA	Varchar(15)	Es la contraseña del usuario administrador		Si
ESPECIALIDAD	Varchar(25)	Indica la especialidad del experto		Si
EMAIL	Varchar(25)	Indica la dirección de correo de contacto del experto		Si
TIPO	Varchar(15)	Indica el tipo de información que utiliza el usuario experto		Si

Tabla 2: Campos de la tabla EXPERTO

TIPO es clave foránea de TIPO_INFORMACION.

PROBLEMA: Tabla que contiene toda la información relacionada con los problemas creados en el sistema.

CAMPO	TIPO	DESCRIPCION	CLAVE	REQUERIDO
ID_PROBLEMA	Varchar(10)	Identificador del problema	*	Si
DESCRIPCIÓN	Varchar(100)	Descripción del problema		Si
UMBRAL_CONSENSO	Float	Valor del umbral de consenso		Si
N_MAX Rondas	Integer	Número máximo de rondas del problema		Si
N_ALTERNATIVAS	Integer	Número de alternativas que tiene el problema		Si
N_EXPERTOS	Integer	Número de expertos asignados al problema		Si

RONDA_ACTUAL	Integer	Indica la ronda actual del problema	Si
ESTADO_ACTUAL	Varchar(15)	Indica el estado del problema	Si

Tabla 3: Campos de la tabla PROBLEMA

PROB_EXP: Tabla que relaciona cada experto con los problemas que tiene asignados en el sistema.

CAMPO	TIPO	DESCRIPCION	CLAVE	REQUERIDO
ID_USUARIO	Varchar(10)	Es el identificador del usuario experto	*	Si
ID_PROBLEMA	Varchar(10)	Identificador del problema	*	Si

Tabla 4: Campos de la tabla PROB_EXP

ID_USUARIO es clave foránea de EXPERTO e ID_PROBLEMA es clave foránea de PROBLEMA.

TIPO_INFORMACION: Tabla que contiene el tipo de información que el usuario experto puede utilizar para introducir sus preferencias.

CAMPO	TIPO	DESCRIPCION	CLAVE	REQUERIDO
TIPO	Varchar(15)	Indica el tipo de información que el experto utiliza para dar sus preferencias	*	Si
DOMINIO	Varchar(45)	Es el dominio del tipo de información		Si

Tabla 5: Campos de la tabla TIPO_INFORMACION

ALTERNATIVA: Tabla que contiene las alternativas de los distintos problemas creados en el sistema.

CAMPO	TIPO	DESCRIPCION	CLAVE	REQUERIDO
ID_PROBLEMA	Varchar(10)	Identificador del problema	*	Si
ID_ALTERNATIVA	Varchar(10)	Identificador de la alternativa	*	Si
DESCRIPCION	Varchar(45)	Descripción de la alternativa		Si

Tabla 6: Campos de la tabla ALTERNATIVA

ID_PROBLEMA es clave foránea de *PROBLEMA*.

ALT_ALT: Tabla que relaciona los pares de alternativas con el problema.

CAMPO	TIPO	DESCRIPCION	CLAVE	REQUERIDO
ID_PROBLEMA1	Varchar(10)	Identificador del problema	*	Si
ID_ALTERNATIVA1	Varchar(10)	Identificador de la alternativa	*	Si
ID_PROBLEMA2	Varchar(10)	Identificador del problema	*	Si
ID_ALTERNATIVA2	Varchar(10)	Identificador de la alternativa	*	Si

Tabla 7: Campos de la tabla ALT_ALT

Tiene dos claves foráneas (ID_PROBLEMA1, ID_ALTERNATIVA1) y (ID_PROBLEMA2, ID_ALTERNATIVA2) que proceden de la relación reflexiva de la entidad *ALTERNATIVA*.

RECOMENDACIÓN: Tabla que registra las recomendaciones generadas para cada experto en cada una de las rondas de un problema en el que está asignado dicho experto.

CAMPO	TIPO	DESCRIPCION	CLAVE	REQUERIDO
ID_PROBLEMA1	Varchar(10)	Identificador del problema	*	Si
ID_ALTERNATIVA1	Varchar(10)	Identificador de la alternativa	*	Si
ID_PROBLEMA2	Varchar(10)	Identificador del problema	*	Si
ID_ALTERNATIVA2	Varchar(10)	Identificador de la alternativa	*	Si
ID_USUARIO	Varchar(10)	Identificador del experto	*	Si
N_RONDA	Integer	Identificador de la ronda	*	Si
DIRECCIÓN	Varchar(15)	Es la dirección que debe realizar el experto		Si

Tabla 8: Campos de la tabla RECOMENDACIÓN

Tiene dos claves foráneas: (ID_PROBLEMA1, ID_ALTERNATIVA1, ID_PROBLEMA2, ID_ALTERNATIVA2) de la entidad *ALT_ALT* e ID_USUARIO de la entidad *EXPERTO*.

RONDA: Tabla que almacena el grado de consenso alcanzado y el número total de cambios realizados en cada una de las rondas del proceso de consenso.

CAMPO	TIPO	DESCRIPCION	CLAVE	REQUERIDO
ID_PROBLEMA	Varchar(10)	Identificador del problema	*	Si
N_RONDA	Integer	Número de la ronda	*	Si
GRADO_CONSENSO	Float	Indica el grado de consenso alcanzado en la ronda		Si
N_CAMBIOS	Integer	Números de cambios realizados en dicha ronda		Si

Tabla 9: Campos de la tabla RONDA

Tiene como clave foránea ID_PROBLEMA de la entidad *PROBLEMA*.

EXPERTO_RONDA: Tabla que almacena el número de cambios que tiene que hacer cada experto en cada una de las rondas del proceso de consenso de los problemas en los que está asignado y el nombre del fichero donde se almacenan sus preferencias.

CAMPO	TIPO	DESCRIPCION	CLAVE	REQUERIDO
ID_USUARIO	Varchar(10)	Identificador del usuario	*	Si
ID_PROBLEMA	Varchar(10)	Identificador del problema	*	Si
N_RONDA	Integer	Indica el número de ronda	*	Si
N_CAMBIOS_EXP	Integer	Número de cambios que realiza el experto		Si
NOM_FICH_PREF	Varchar(30)	Nombre del fichero que contiene la matriz de preferencias del experto		Si

Tabla 10: Campos de la tabla EXPERTO_RONDA

Tiene dos claves foráneas: (ID_PROBLEMA y N_RONDA) de la entidad RONDA e ID_USUARIO de la entidad EXPERTO.

OP_COLECTIVA: Tabla que registra la opinión colectiva en cada par de alternativas de una rondas de un problema.

CAMPO	TIPO	DESCRIPCION	CLAVE	REQUERIDO
ID_PROBLEMA1	Varchar(10)	Identificador del problema	*	Si
ID_ALTERNATIVA1	Varchar(10)	Identificador de la alternativa	*	Si
ID_PROBLEMA2	Varchar(10)	Identificador del problema	*	Si
ID_ALTERNATIVA2	Varchar(10)	Identificador de la alternativa	*	Si
N_RONDA	Integer	Identificador de la ronda	*	Si
VALOR	Float	Es el valor		Si

Tabla 11: Campos de la tabla OP_COLECTIVA

Tiene como clave foránea: (ID_PROBLEMA1, ID_ALTERNATIVA1, ID_PROBLEMA2, ID_ALTERNATIVA2) de la entidad *ALT_ALT*.

MZ_CONSENSO: Tabla que registra el consenso alcanzado en cada par de alternativas de una ronda de un problema.

CAMPO	TIPO	DESCRIPCION	CLAVE	REQUERIDO
ID_PROBLEMA1	Varchar(10)	Identificador del problema	*	Si
ID_ALTERNATIVA1	Varchar(10)	Identificador de la alternativa	*	Si
ID_PROBLEMA2	Varchar(10)	Identificador del problema	*	Si
ID_ALTERNATIVA2	Varchar(10)	Identificador de la alternativa	*	Si
N_RONDA	Integer	Identificador de la ronda	*	Si
VALOR	Float	Es el valor		Si

Tabla 12: Campos de la tabla MZ_CONSENSO

Tiene como clave foránea: (ID_PROBLEMA1, ID_ALTERNATIVA1, ID_PROBLEMA2, ID_ALTERNATIVA2) de la entidad *ALT_ALT*.

3.4.3 Diseño de la interfaz.

En esta fase del diseño del sistema software se define cual va a ser la apariencia visual de la aplicación, es decir, se define la interfaz visual entre el usuario y la aplicación. Sin duda, realizar un buen diseño de la interfaz resulta primordial ya que, ésta debe presentarse atractiva al usuario de la aplicación pero a la vez le debe de resultar fácil de entender y trabajar sobre ella.

Esta importancia es mayor en nuestro caso ya que, la interfaz de nuestro proyecto es una interfaz Web. Para las aplicaciones con interfaces Web no existe una guía de estilo estándar para que resulten, a la vez, atractivas y familiares como existe, por ejemplo, para desarrollar interfaces para aplicaciones de escritorio de Windows XP. Cada programador, desarrollador o diseñador Web debe definir su propia guía de estilo y procurar que, en base a ella, la interfaz resultante consiga unas cotas dignas de atractivo visual, familiaridad y facilidad de uso.

La ingeniería de usabilidad define en esta fase los siguientes aspectos:

- Definir estilo.
- Metáforas.
- Pantallas.
- Caminos de navegación.
- Secuencias de diálogo.
- Mensajes de error.

Además nosotros hemos añadido otro aspecto que hemos llamado ‘Notificaciones’ para distinguirlas de los mensajes de error.

Cada uno de estos aspectos lo veremos a continuación de forma detallada.

3.4.3.1 Definir estilo.

Antes de diseñar una interfaz de usuario, se debe definir el estilo de la misma. Esto es de vital importancia cuando el diseño va a ser compartido entre varios diseñadores, ya que ayuda a mantener la coherencia interna de la interfaz.

Sin embargo, en contra de lo que pueda parecer en un principio, también es de mucha utilidad definir una guía de estilo cuando sólo hay un diseñador encargado de la interfaz. Esto se debe a varias razones:

- A veces mantener la coherencia y consistencia de una interfaz, si ésta es muy grande o muy ambiciosa, es complicado incluso si sólo hay un diseñador.
- El diseñador puede, por las más diversas razones, abandonar el diseño. Es útil para sus sustitutos contar con una guía de estilo predefinida para no tener que empezar de cero. Lo mismo puede aplicarse si no es el diseñador original el que se encarga del mantenimiento o la actualización de la interfaz

Quedando demostrada la utilidad del uso de guías de estilo pasamos a definir las reglas y normas que contendrá la guía de estilo de nuestra interfaz:

- **Fuentes:** para escribir cualquier texto en nuestra interfaz utilizaremos una tabla cuyas propiedades son las siguientes:
 - Tipo de letra: "Verdana"
 - Tamaño: 10pt;
 - Color de la letra: # 000000;
 - Color de fondo de la página: # 999999;
 - Color de fondo de la tabla: #C0C0C0;
 - Alineación del texto: center;

- **Cabeceras:** las cabeceras de cada .JSP tendrán las siguientes propiedades:
 - Tipo de letra: "Verdana"
 - Tamaño: 18pt;
 - Color de la letra: #000000;
 - Alineación del texto: center.

- **Logotipo:** arriba a la izquierda. Esta presente en todas las paginas .JSP.

3.4.3.2 Metáforas.

Una metáfora es el empleo de un objeto con un significado o dentro de un contexto diferente al habitual. Al diseñar una interfaz gráfica, la utilización de metáforas resulta muy útil ya que permiten al usuario, por comparación con otro objeto o concepto, comprender de una manera más intuitiva las diversas tareas que la interfaz permite desarrollar.

Al igual que pasa en el ámbito de la literatura, para que una metáfora cumpla con su cometido, el desarrollador de la aplicación y el usuario final de ésta deben tener una base cultural similar. Es muy posible que el uso de un icono de manera metafórica sea entendido de una manera por el usuario occidental y de otra, bien distinta por un usuario oriental. Hay que intentar por lo tanto, que las metáforas empleadas sean lo más universales posibles para que, así sean comprendidas a la perfección por la mayor parte de los usuarios potenciales.

Las aplicaciones de escritorio de Windows suelen seguir la Guía de Estilo XP y utilizan una serie de metáforas con las que el usuario está plenamente familiarizado (por ejemplo, una lupa con un signo '+' en su interior establece que la función del icono es, inequívocamente, la de realizar un aumento de zoom). En el mundo de las aplicaciones Web también existen una cantidad de metáforas de amplia difusión como puede ser, por ejemplo, el celebre carrito de la compra que emplean casi todos los comercios online.

Pero las metáforas no sólo dependen del tipo de aplicación (escritorio o Web) sino también del ámbito de la misma. Por ejemplo, el carrito de la compra es una metáfora conocida por todos pero, si

nuestra aplicación no va a vender nada al usuario no resulta conveniente utilizarla ya que puede confundir.

En nuestra aplicación no hemos creído conveniente la realización de ninguna metáfora, pues la interfaz es muy sencilla e intuitiva.

3.4.3.3 Pantallas.

Como hemos visto en apartados anteriores, para diseñar una buena interfaz, es necesario un concienzudo trabajo de análisis y diseño.

Los prototipos de papel son una forma de crear una imagen palpable de lo que será una futura aplicación o sitio web. Su creación y manipulación es rápida y elástica. Además permite a los usuarios imaginarse lo que será la futura aplicación en funcionamiento sin interferencias de tipo:

- *Técnico*: plataformas, utilización de elementos multimedia como Flash...
- *Gráfico*: colores, tipografía que en muchas ocasiones desvían el centro de la discusión hacia asuntos cuya importancia es muy relativa en el proceso inicial. Nos podemos encontrar debatiendo sobre el tono de verde de una pantalla (aspecto secundario) frente a lo principal: secciones, datos necesarios, utilización futura del producto. Como símil en la "vida real" sería como discutir acerca de la pintura de un coche, cuando ni siquiera sabemos cómo va a ser.

Es decir, se centra el debate en la esencia y concepto de la futura aplicación: la funcionalidad, el contenido y el flujo de pantallas. Permite el desarrollo de maquetas rápido y sin esfuerzo en escribir código (HTML, Visual Basic...).

Los prototipos en papel nos ayudarán a ver que los contenidos de las páginas son los apropiados sin la necesidad de redactar contenidos finales (basta con transmitir el concepto de lo que será cada contenido mediante texto simulado), también servirán como una primera aproximación a la

maquetación final de las pantallas que conformen la aplicación (sin necesidad de presentar una maquetación exacta de lo que será cada pantalla, ni colores e iconos).

Los participantes en el trabajo con prototipos en papel serán:

- *El experto*: es la persona que dirige la sesión y deberá tener conocimientos de usabilidad y diseño centrado en el usuario.
- *Usuarios*: representa a los usuarios finales de la aplicación y son quienes dirán si el funcionamiento es correcto. Además, ayudarán a encontrar y corregir la ubicación de los contenidos.

Las ventajas que nos aportan los prototipos en papel son las siguientes:

- *Fáciles de realizar*: No hay necesidad de escribir/saber código.
- *Económicos*: Permite la detección temprana de errores de concepto y problemas en cuanto a funcionalidades a un mínimo coste.
- *Independiente de tecnología y dispositivos*: No hay necesidad de herramientas específicas. Podemos realizar y corregir un prototipo de papel en cualquier parte con lápiz y papel.
- *Espontaneidad en críticas por parte de los usuarios envueltos en el proceso de diseño*: Ven claramente que no hay un diseño y permite una crítica más abierta sin centrarse en aspecto visual o detalles puramente técnicos.

Para construir las pantallas deberemos tener especial cuidado con el diseño de contenidos. El diseño de contenidos hipermedia debe mantener un equilibrio entre, lo que serían contenidos que no aprovecharan las nuevas posibilidades hipertexto y multimedia y, lo que serían contenidos caóticos o desorientativos debido a un uso excesivo y no sosegado de las posibilidades hipermedia. Debemos diseñar contenidos interrelacionados y vinculados, manteniendo cierta coherencia informativa, comunicacional y organizativa.

La escritura hipertextual debe ser diferente a la tradicional. Debemos ser concisos, precisos, creativos y estructurados a la hora de redactar.

Un principio general es que, los elementos que posee la interfaz tienen que estar colocados de tal modo que, a la hora de realizar una tarea, el usuario deba recorrer la interfaz en la misma dirección que lee un texto, es decir, de izquierda a derecha y de arriba abajo. En países orientales no se realizaría de la misma forma.

Por lo tanto, las pantallas que tendremos serán las siguientes:

- Página principal de acceso
- Menú principal del Administrador.
- Submenú del Administrador
- Introducir N° Exp y Alt
- Crear problema
- Buscar problema
- Modificar problema
- Consultar problema
- Eliminar problema
- Crear experto
- Buscar experto
- Modificar experto
- Consultar experto
- Eliminar experto
- Mostrar estado seleccionado
- Mostrar consenso
- Evolución de consenso de un problema
- Menú principal del Experto
- Introducir opinión
- Consultar el estado de un problema
- Salir

A continuación se mostraran algunas pantallas que se han construido.

Página principal de acceso: Como se puede observar, en la parte superior de la pantalla tenemos el logotipo de la página (recordamos que estos diseños en papel no serán definitivos y es muy probable que este logotipo o su aspecto cambien en sucesivas etapas). Se puede observar fácilmente que nos encontramos en la página principal de acceso, en la que hay que autenticarse como usuario *Experto* o *Administrador*.

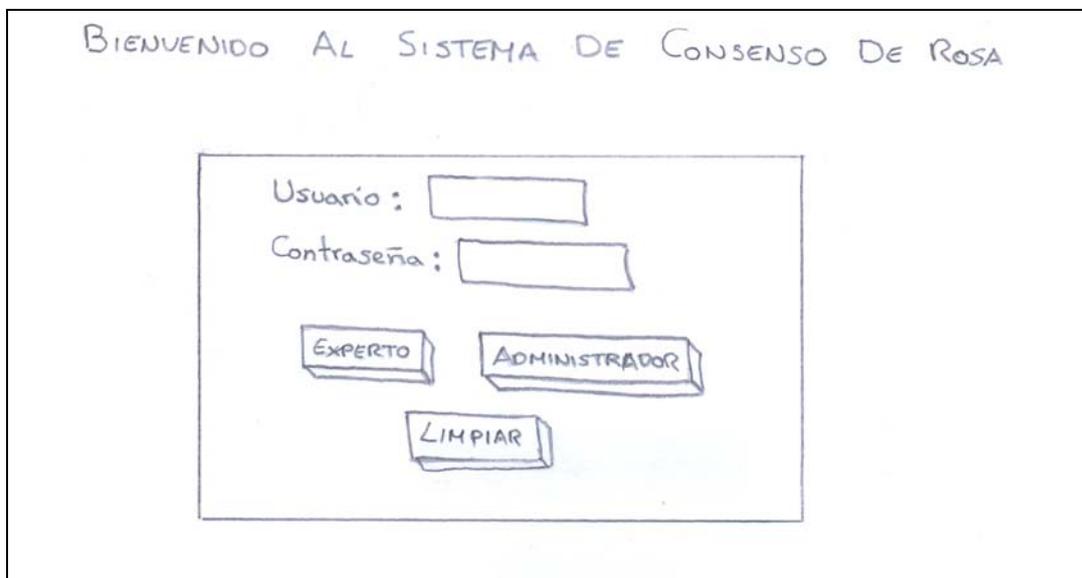


Figura 3.16: Pantalla principal de acceso

Menú principal del Administrador: En esta página se pueden apreciar las distintas opciones que el Administrador puede gestionar.

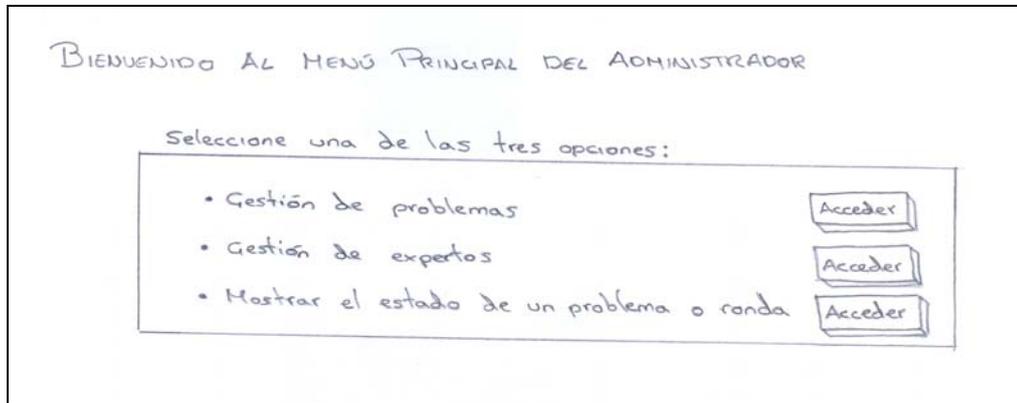


Figura 3.17: Menú principal del Administrador.

Submenú del Administrador: Dependiendo de la opción elegida Problemas, Expertos o Estado Problema, aparecerá un grupo de botones a la izquierda con las distintas acciones que el Administrador puede realizar dentro de esa opción. Además en cualquier momento puede cambiar de opción con los tres botones de la parte superior.



Figura 3.18: Submenú del Administrador

Crear problema: Una vez introducidos en la página anterior el número de alternativas que se van a definir y el número de expertos que van a participar en el problema, se mostrará un formulario con todos los campos que hay que rellenar.

BIENVENIDO /A, rosa

Problemas Expertos Estado Problema

Crear
Modificar
Consultar
Eliminar
Guardar
Salir

id del problema:

Descripción:

Parámetros:

- Umbral consenso:
- N° max rondas:

Alternativas:

-
-
- ...
-

Expertos:

- Seleccionar
- Seleccionar
atorres
fmata
jdelgado
martin

Figura 3.19: Crear Problema

Modificar problema: En esta pantalla aparecerán todos los campos del problema rellenos, para que el Administrador los pueda modificar.

BIENVENIDO/A, ROSA

Problemas Expertos Estado Problema

Crear

Modificar

Consultar

Eliminar

Guardar

Salir

id del problema

id del problema:

Descripción:

Parámetros:

- Umbral consenso:
- Nº max rondas:

Alternativas definidas:

- win xp
- win vista
- win 2000

Expertos asignados:

- fmata
- martin

Figura 3.20: Modificar problema

Consultar problema: En esta página se pueden visualizar todos los datos referentes a un problema seleccionado.

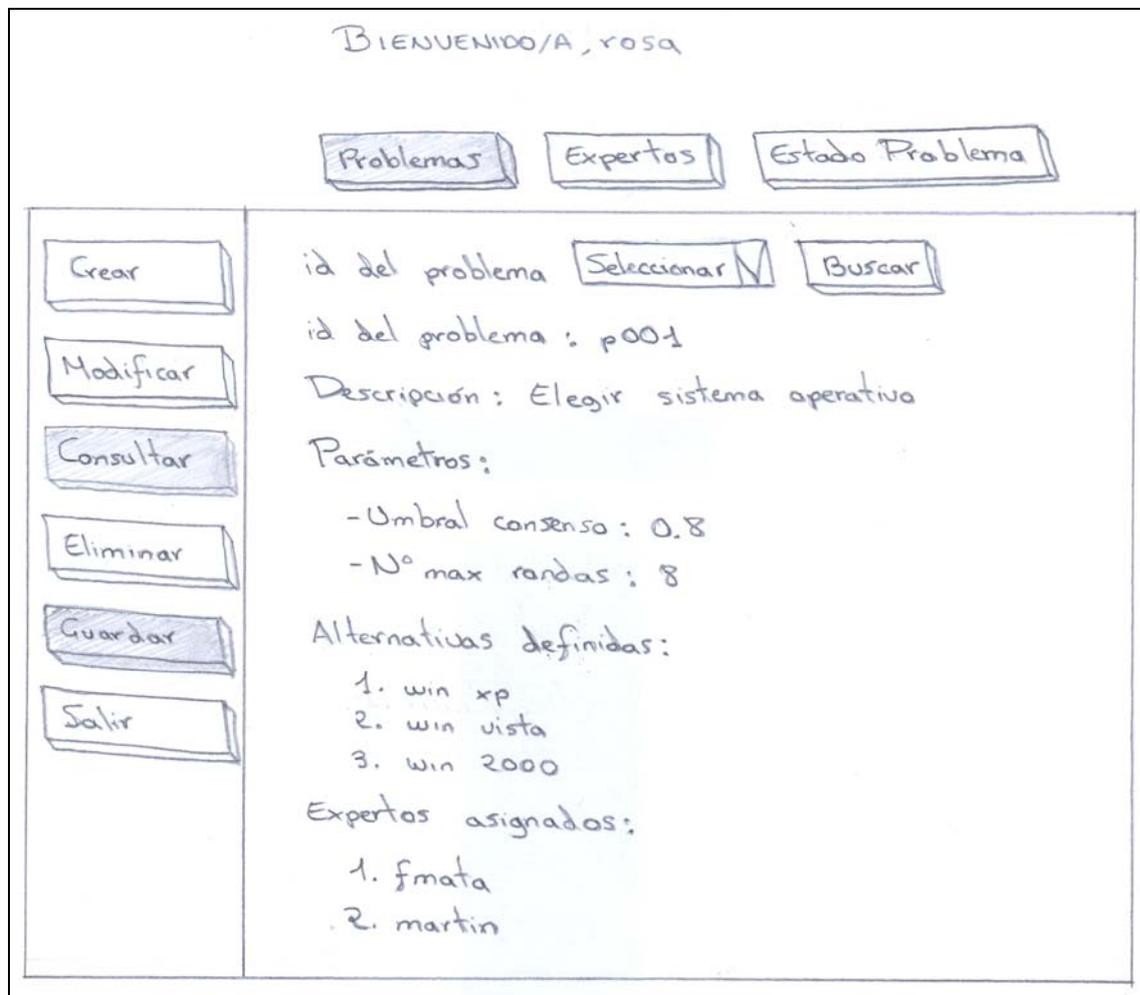


Figura 3.21: Consultar problema

Eliminar problema: En esta página nos aparece la descripción del problema y un botón *Eliminar*, el cual permitirá al Administrador eliminar dicho problema de forma definitiva.

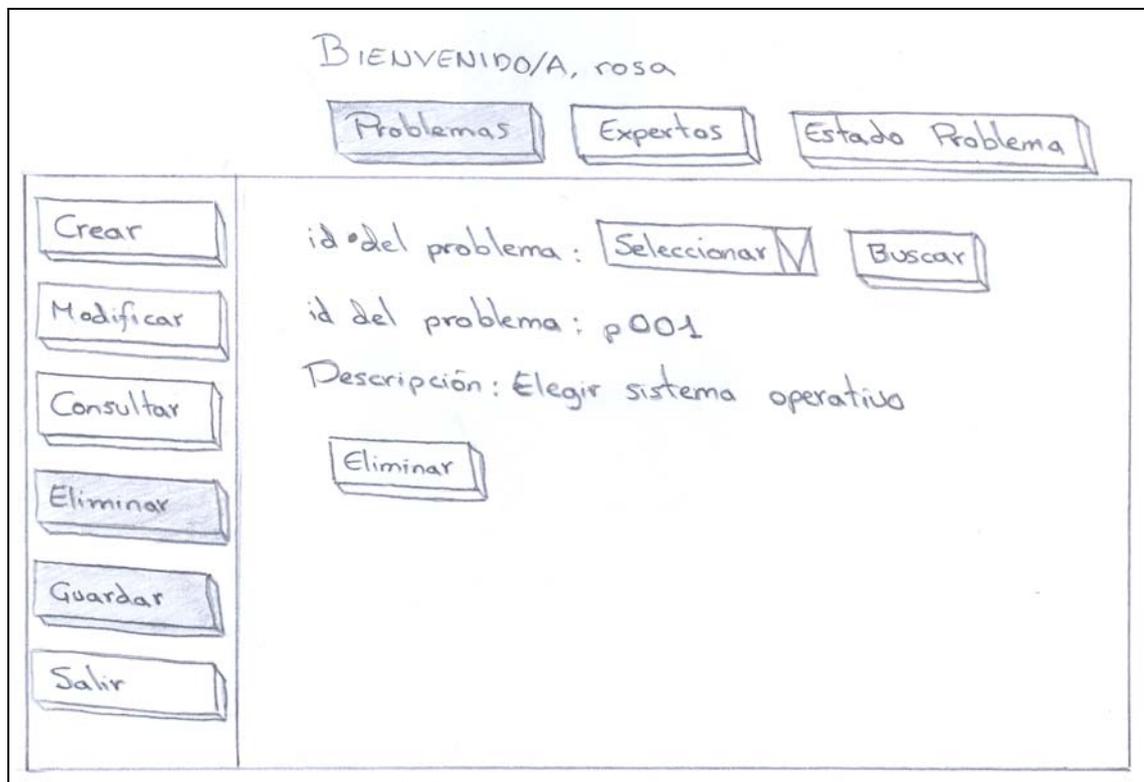


Figura 3.22: Eliminar problema

Crear experto: Esta pantalla nos muestra un formulario que hay que rellenar para poder crear un experto.

BIENVENIDO/A, rosa

Problemas **Expertos** Estado Problema

Crear
Modificar
Consultar
Eliminar
Guardar
Salir

id del experto:

Contraseña:

Confirmar Contraseña:

Especialidad:

Dirección de correo:

Tipo de información:

- Intervalar [0,1]
- Lingüístico {mp, p, pp, i, pm, m, mm}
- Numérico [0,1]

Figura 3.23: Crear experto

Modificar experto: Esta página muestra mediante campos editables toda la información de un experto previamente seleccionado, de tal forma que el Administrador pueda modificarla sin problemas.

BIENVENIDO/A, rosa

Problemas Expertos Estado Problema

Crear

Modificar

Consultar

Eliminar

Guardar

Salir

id del experto:

id del experto:

Contraseña:

Confirmar contraseña:

Especialidad:

Dirección de correo:

Tipo información:

- Interualar [0,1]
- Lingüístico {mp, p, pp, i, pm, m, mm}
- Numérico [0,1]

Figura 3.24: Modificar experto

Consultar experto: Esta página muestra toda la información referente a un experto incluidos los problemas que tiene asignados, el estado en el que se encuentran y la ronda actual de cada uno de ellos.

BIENVENIDO/A, rosa

Problemas Expertos Estado Problema

Crear

Modificar

Consultar

Eliminar

Guardar

Salir

id del experto:

id del experto: atorres

Contraseña: ana

Especialidad: informática

Dirección de correo: ana@hotmail.com

Tipo información: Lingüístico

Prob. asignado	Estado prob	Nº Rondas	Nº Cambios
p002	Progreso	1/7	0
p004	Finalizado	7/10	Total: 17

Figura 3.25: Consultar experto

Eliminar experto: Esta pantalla permite al Administrador eliminar un experto.

BIENVENIDO/A, rosa

Problemas Expertos Estado Problema

Crear

Modificar

Consultar

Eliminar

Guardar

Salir

id del experto: [Seleccionar] [Buscar]

id del experto: atorres

Especialidad: Informática

Dirección de correo: ana@hotmail.com

Tipo información: Lingüístico

Eliminar

Figura 3.26: Eliminar experto

Mostrar consenso: Una vez seleccionado un problema cuyo estado sea *Progreso* si pulsamos *Calcular nivel de consenso*, aparecerá una pantalla cuyo mensaje dependerá de si se ha alcanzado o no el grado de consenso, si se han agotado el número máximo de rondas sin conseguir el consenso o si por el contrario, no se puede calcular porque aún quedan expertos que no han introducido sus preferencias.

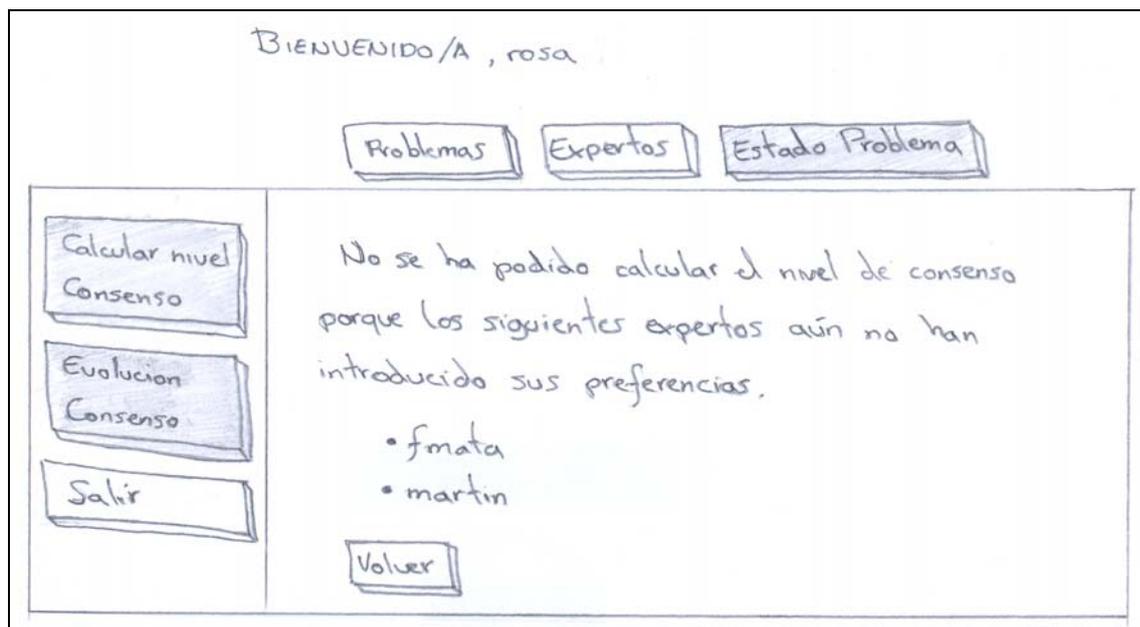


Figura 3.27: *Mostrar consenso*

Evolución de consenso de un problema: Esta página muestra toda la información referente a la evolución de consenso de un problema determinado.

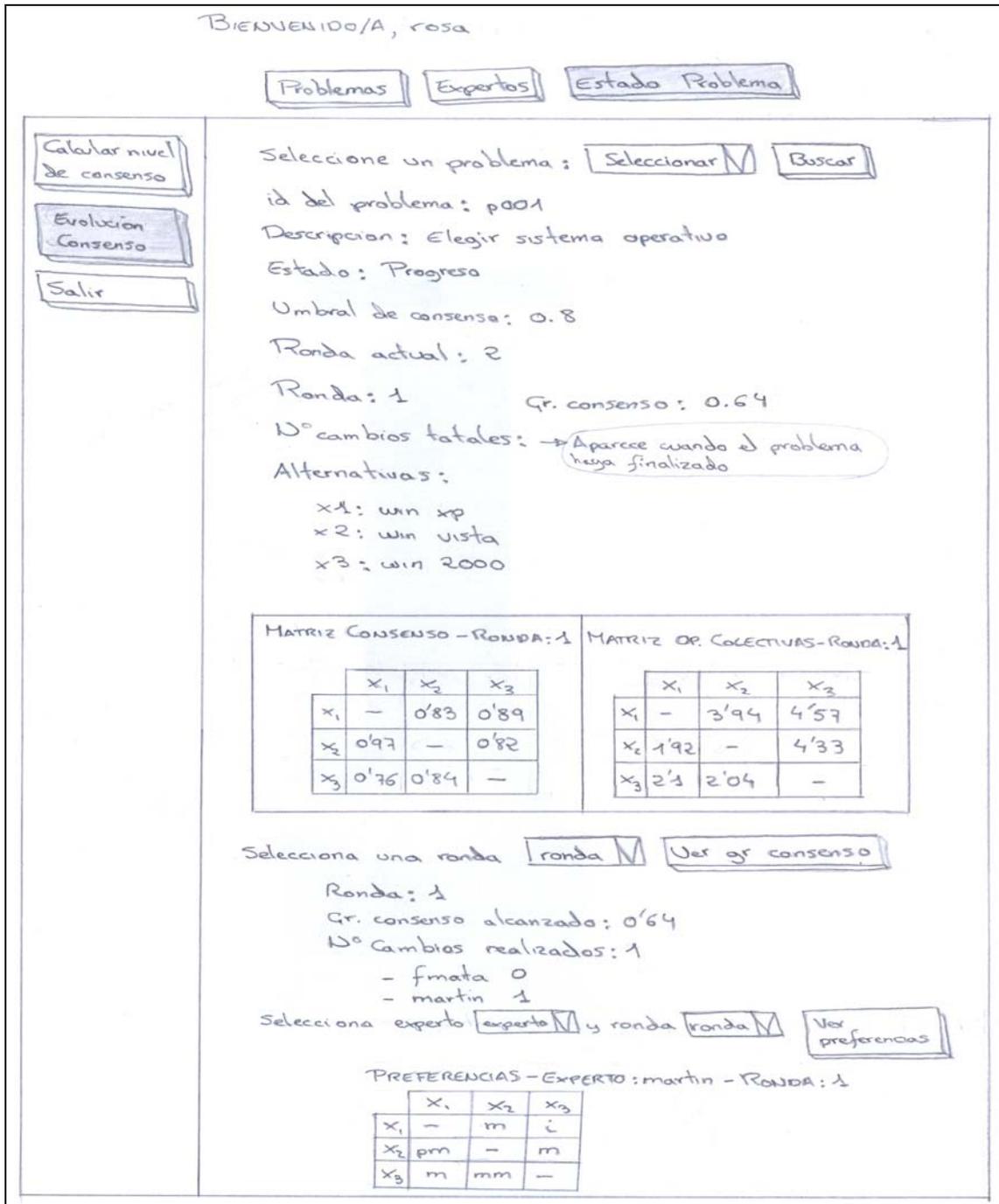


Figura 3.28: Evolución de consenso de un problema

Menú principal del Experto: Esta pantalla muestra todos los problemas que el Experto tiene asignados. Se puede observar que a la izquierda aparecen un grupo de botones con distintas opciones que el usuario Experto puede realizar. Estas opciones dependerán del estado del problema.

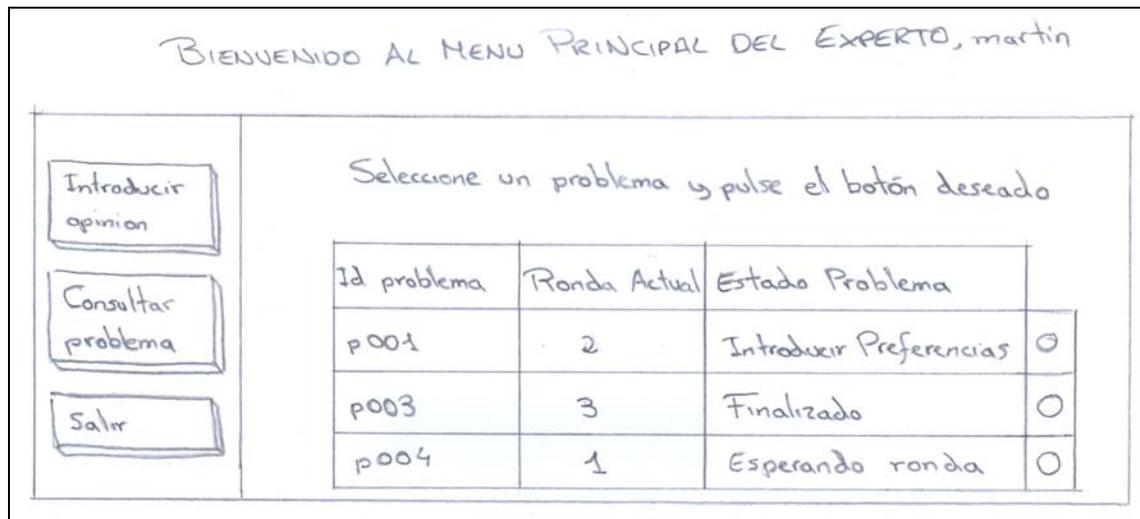


Figura 3.29: Menú principal del experto

Introducir opinión: Esta página web permite al Experto introducir sus preferencias en el problema seleccionado previamente. Además podrá visualizar las recomendaciones que se le han generado para la ronda actual en caso de existir.

BIENVENIDO/A, martin

Introducir
opinión

Consultar
problema

Salir

Id Problema : p001

Descripción : Elegir sistema operativo

Ronda actual : 2

Alternativas:

x1: win xp
x2: win vista
x3: win 2000

Recomendaciones:

	X ₁	X ₂	X ₃
X ₁	-	m	i
X ₂	pm	-	m
X ₃	m	mm	-

- Verde incrementar
- Rojo decrementar
- Negro no modificar

Tipo información: Lingüístico

Dominio: { mp, p, pp, i, pm, m, mm }

INTRODUCIR PREFERENCIAS

	X ₁	X ₂	X ₃
X ₁	-	m	i
X ₂	pm	-	m
X ₃	m		-

Enviar

Figura 3.30: Introducir opinión

Consultar el estado de un problema: En esta página el Experto podrá visualizar toda la información relacionada con el proceso de consenso de un problema determinado.

BIENVENIDO/A, martin

Introducir
opinión

Consultar
problema

Salir

Selecciona un problema

id problema: p001

Descripcion: Elegir sistema operativo

Estado del problema: Finalizado

Nº Expertos asignados: 2

Nº max rondas: 8

Ronda actual: 3

Gr. Consenso : 0'81

Nº cambios totales: 2

Alternativas:

x1: win xp
x2: win vista
x3: win 2000

PREFERENCIAS Y RECOMENDACIONES RONDA 1

	x ₁	x ₂	x ₃
x ₁	-	m	i
x ₂	pm	-	m
x ₃	m	mm	-

- Verde incrementar
- Rojo decrementar
- Negro no modificar

Ronda: 1

Gr consenso: 0'64

Seleccionar otra ronda

Figura 3.31: Consultar el estado de un problema

3.4.3.4 Caminos de navegación.

Hasta este momento tenemos un diseño visual de la interfaz estática, es decir, cada pantalla diseñada individualmente, pero no tenemos una idea de si en el conjunto de la interacción, la acción va a transcurrir de forma comprensible para el usuario. Para ello vamos a diseñar la interfaz en movimiento y comprobar que es usable.

Para estudiar los caminos de interacción se empleará una herramienta llamada *storyboard*, que consiste en mostrar, a modo de secuencia, las distintas pantallas por las que se va pasando al realizar el usuario una determinada acción sobre la aplicación. Mediante flechas se ayuda a entender que es lo que ha desencadenado el paso de una pantalla a otra.

La técnica del *Storyboarding* nos resultará altamente útil para describir los escenarios de situaciones concretas que ayuden a entender partes del sistema. Con los *storyboards* se consigue dotar al escenario descrito en lenguaje natural de la componente gráfica que facilita la comprensión y el detalle. Además, podemos analizar las pantallas de este escenario para ver si los pasos que se dan son los correctos y si la acción se entiende bien, de forma que podría ser necesario añadir más pantallas intermedias o, por el contrario, podría convenir fusionar algunas pantallas. Los *storyboard* nos facilitan esta tarea ya que nos permiten ver juntas todas las pantallas asociadas a una tarea o escenario.

Para construir los *storyboard* utilizaremos las pantallas diseñadas, y lo haremos de la manera más rápida y económica posible empleando lápiz y papel.

No se han desarrollado *storyboards* para todas las acciones de nuestro sistema por lo que los que se muestran a continuación son los que se han considerado más importantes:

- Storyboard Crear un problema (Administrador)
- Storyboard Calcular el nivel de consenso de un problema (Administrador)
- Storyboard Evolución de consenso de un problema (Administrador)
- Storyboard Introducir preferencias (Experto)
- Storyboard Consultar el estado de un problema (Experto)

Una vez diseñados deberían ser validados para comprobar que realmente la aplicación es usable.

Storyboard Crear problema

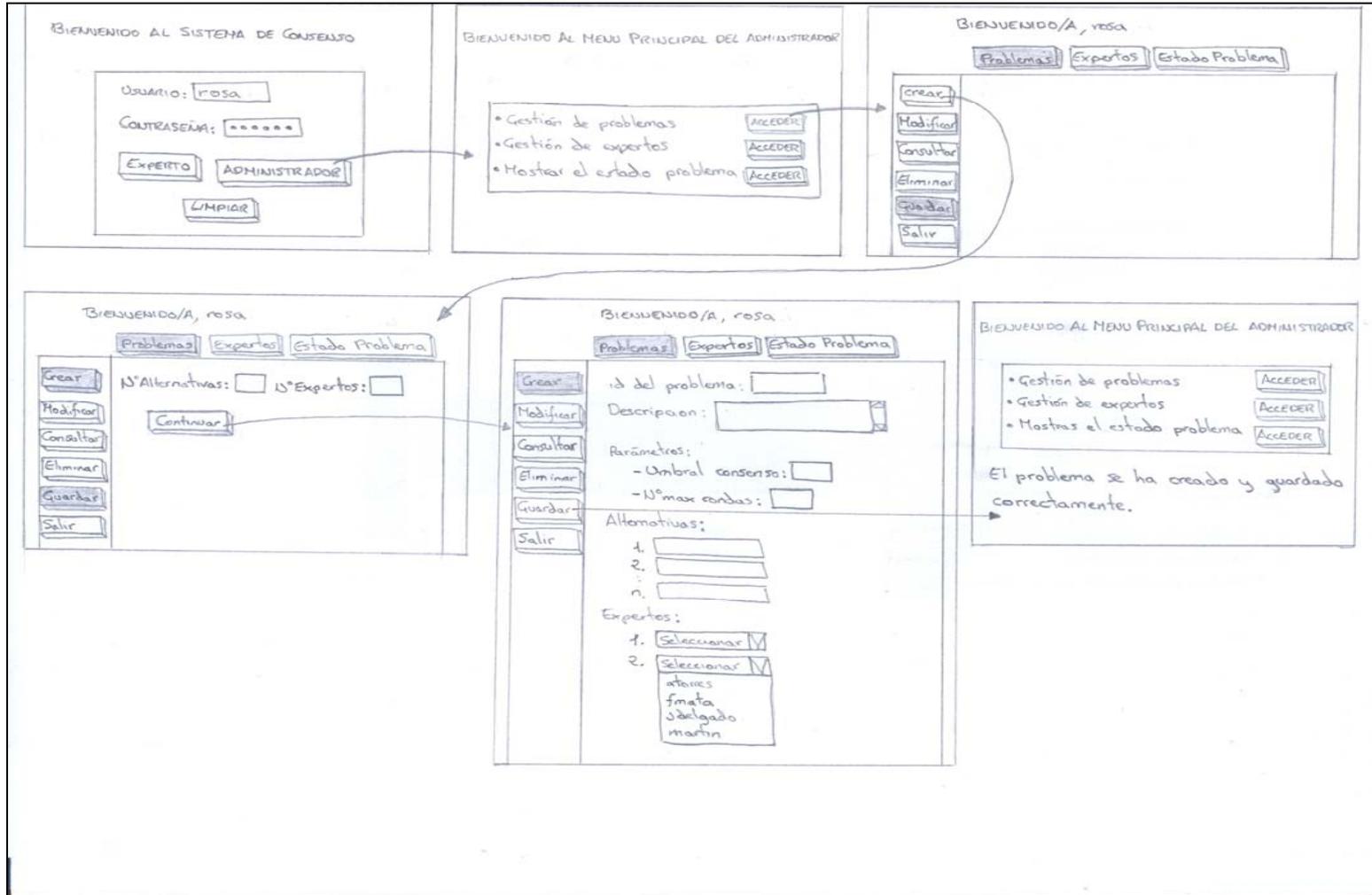


Figura 3.32: Storyboard Crear problema

Storyboard Calcular el nivel de consenso de un problema

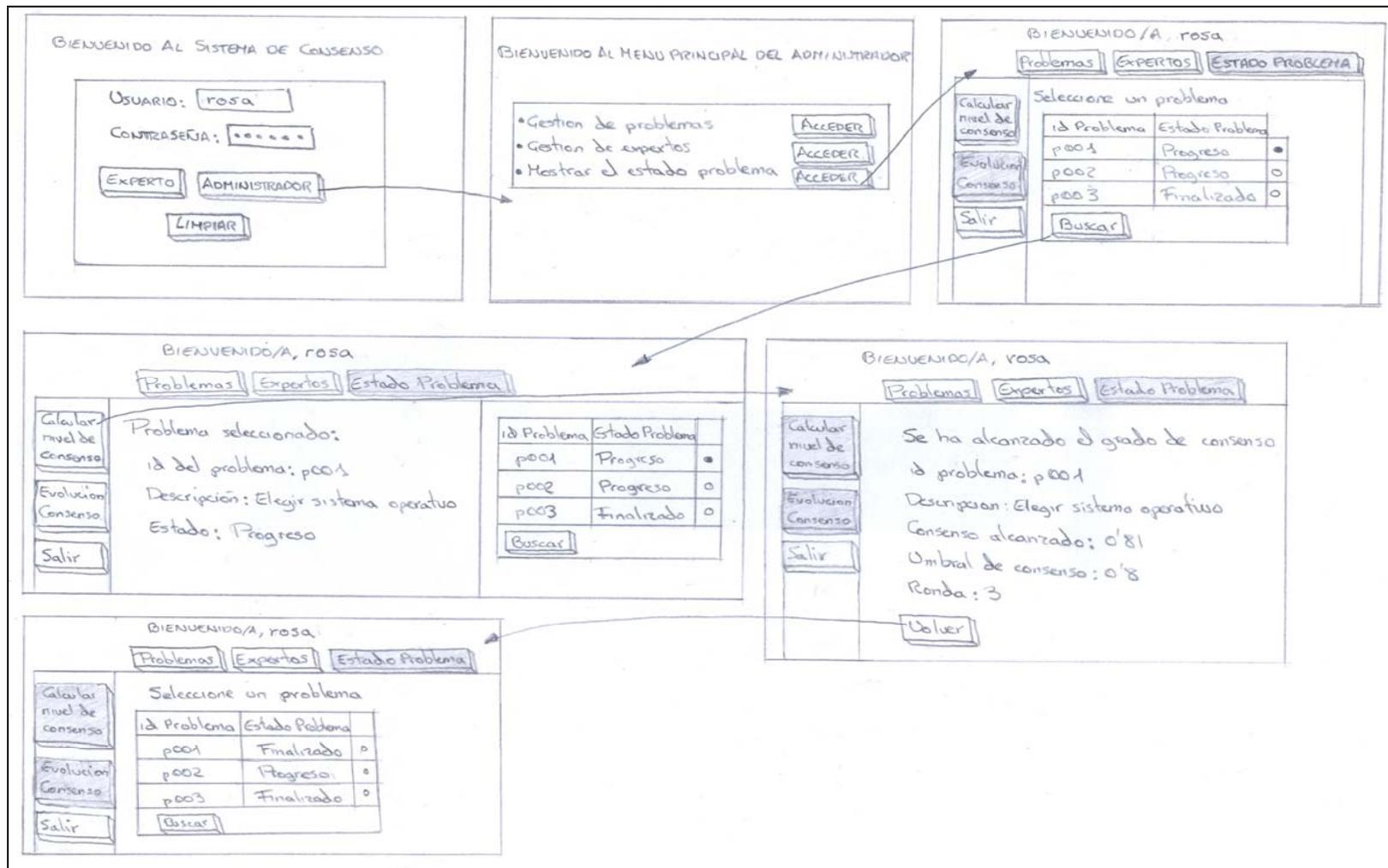


Figura 3.33: Storyboard Calcular el nivel de consenso de un problema

Storyboard Evolución de consenso de un problema

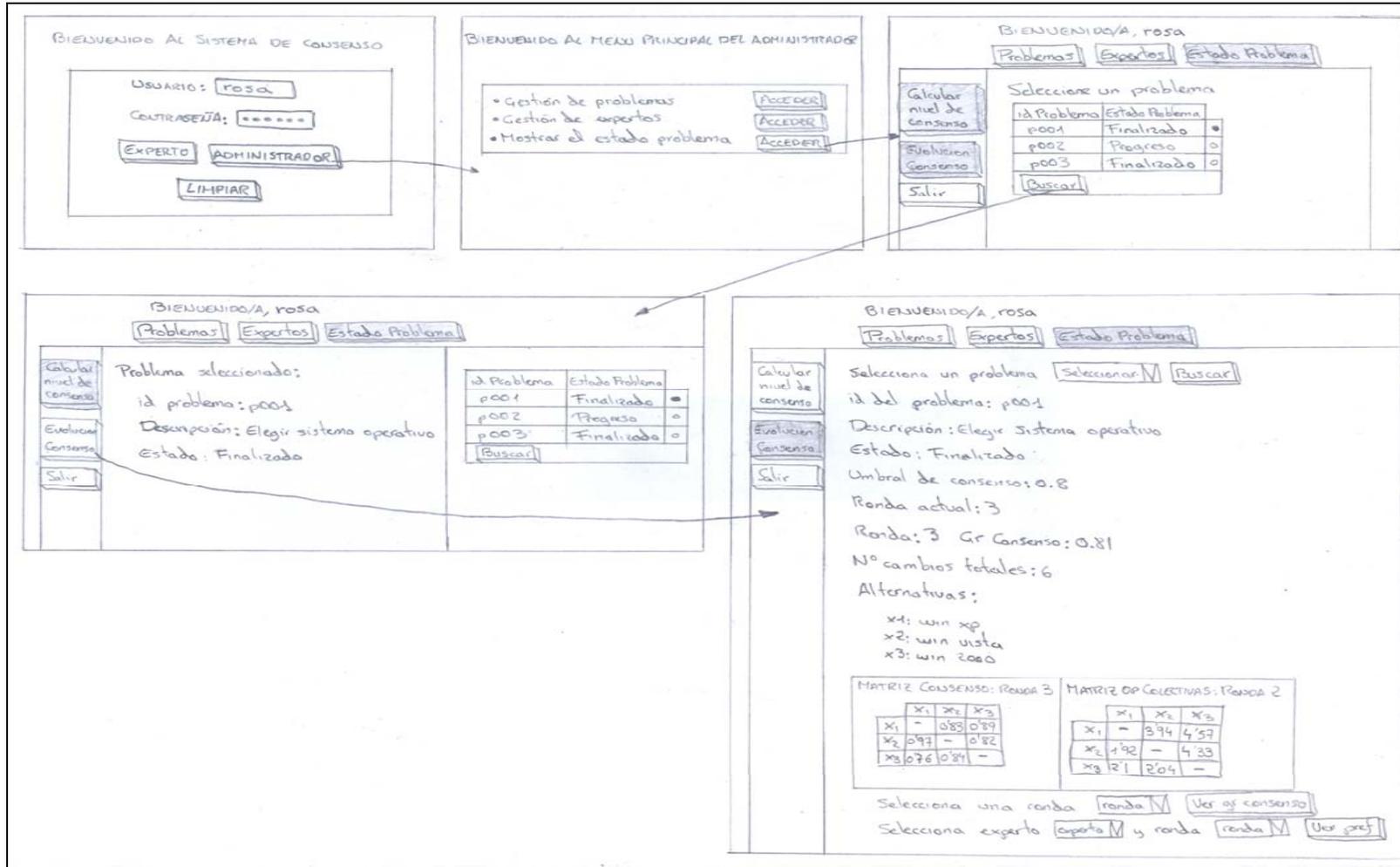


Figura 3.34: Storyboard Evolución de consenso de un problema

Storyboard Introducir preferencias

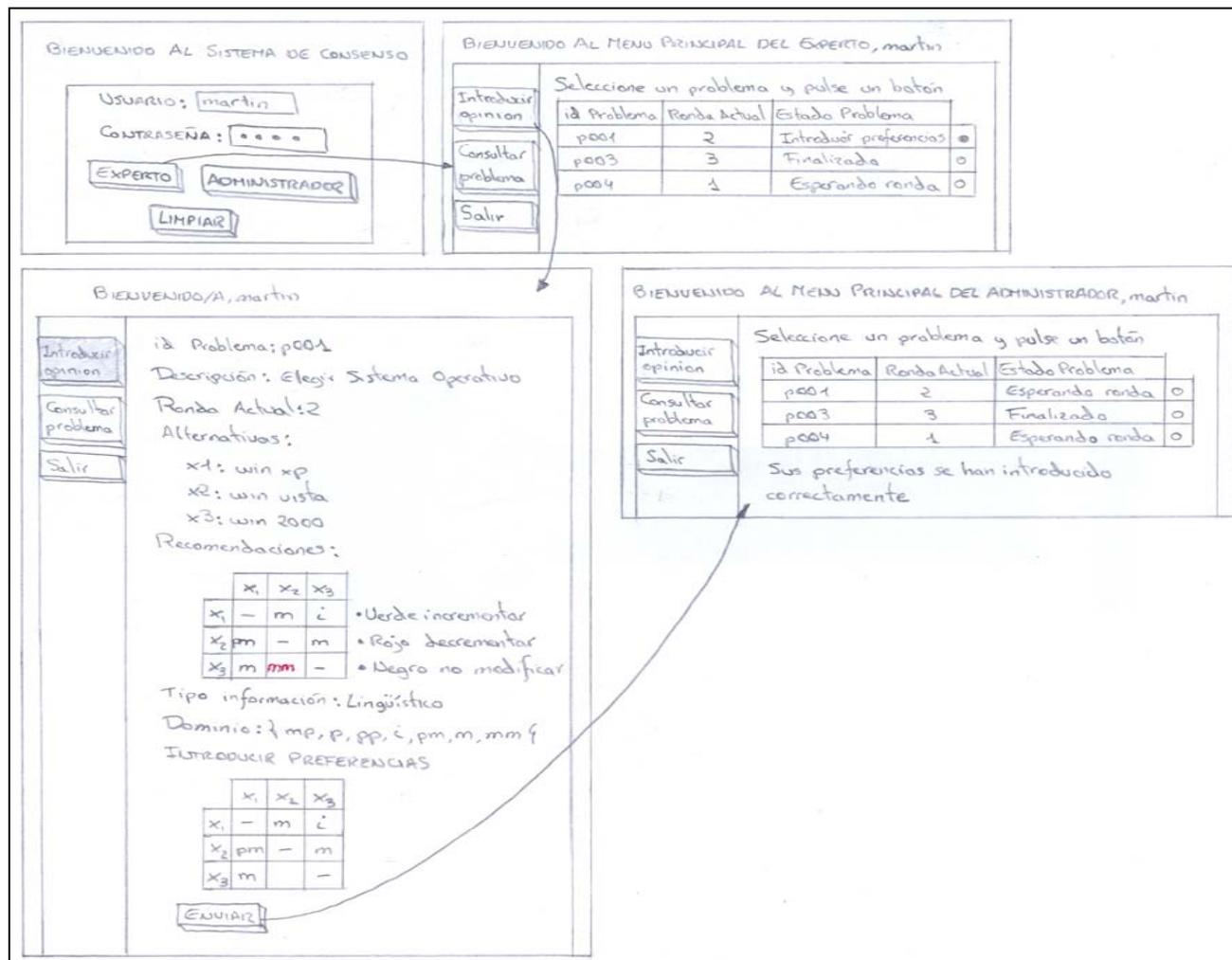


Figura 3.35: Storyboard Introducir preferencias

Storyboard Consultar el estado de un problema

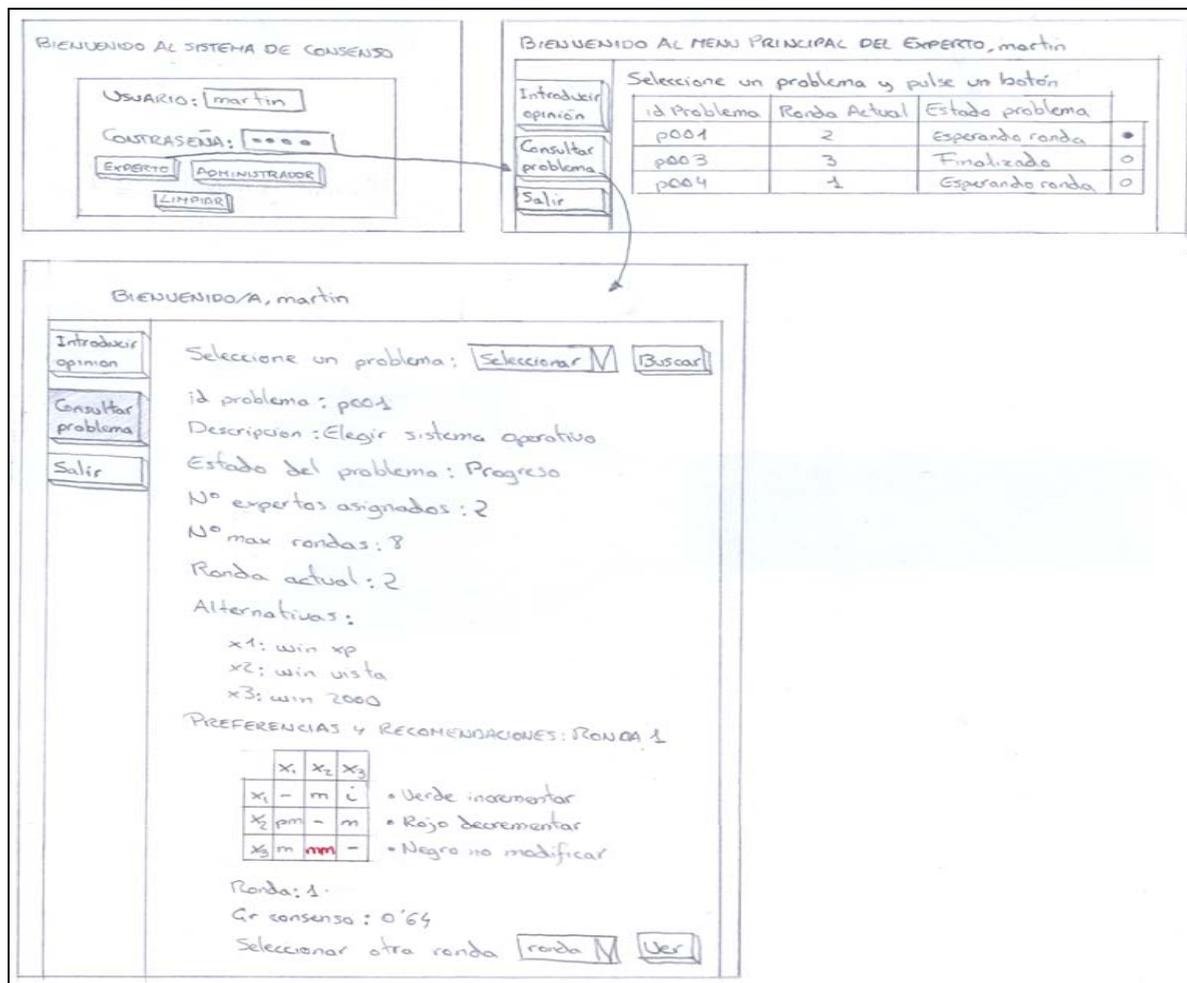


Figura 3.36: Consultar el estado de un problema

3.4.3.5 Mensajes de error.

Una vez diseñadas las pantallas, los caminos de navegación y las secuencias de diálogo, se tiene una idea muy clara de cómo se va a desarrollar físicamente la interacción entre usuario y ordenador, y por tanto de las posibles situaciones de error que pueden darse en esa interacción. Por ello ahora es el momento de diseñar los mensajes de error.

Los mensajes de error son el medio por el que el sistema comunica al usuario que se ha producido un error en la interacción.

Los errores se producen por falta de conocimiento sobre la interfaz, porque se no se ha entendido correctamente el estado del sistema o bien inadvertidamente (por ejemplo se pulsa un botón cuando se intentaba pulsar otro). Ello lleva al usuario a sentirse confuso y aumenta su ansiedad, sobre todo en usuarios noveles. En los que su falta de conocimiento y confianza en el uso de la interfaz lleva a amplificar el stress, lo que puede llevarles a una experiencia de uso de la interfaz frustrante. Esto es muy peligroso, ya que a menudo, una interfaz se recuerda más por lo que pasó en una situación de error que por lo que pasó cuando las cosas fueron bien. Eso puede llevar a una mala valoración de la calidad de la interfaz por los usuarios.

Los mensajes de error son muy importantes de cara a la usabilidad, ya que bien diseñados, permiten aumentar la confianza del usuario en el uso de la interfaz y que pueda seguir con su tarea. Por ello hay que diseñarlos con cuidado. A la hora de diseñar mensajes de error se deberían seguir las siguientes reglas:

- **Ser breves:** el usuario es una persona ocupada tratando de llevar a cabo una tarea. Si se le presenta un mensaje de error muy largo, lo normal es que no lo lea ya que no tendrá tiempo para hacerlo. Por ello hay que diseñar mensajes de error claros, pero breves.
- **Ser específico:** los mensajes demasiado generales no son adecuados ya que dicen al usuario que algo ha ido mal, pero no le indican claramente qué. Por lo tanto, el usuario no sabrá qué hacer para impedir que se produzca el error y su sensación de frustración aumentará

- **Usar un tono positivo y guía constructiva:** nunca recriminar al usuario lo que ha hecho mal, en su lugar siempre que sea posible indicarle que debe hacer para eliminar el error.
- **Usar un formato físico apropiado:** la mayoría de usuarios encuentran más fácil leer un mensaje donde se mezclan letras mayúsculas y minúsculas de la forma habitual, por tanto este formato es siempre preferible. Los mensajes escritos únicamente en mayúsculas deberían reservarse para avisos breves y graves. Si el mensaje de error debe contener un código numérico, este debería figurar al final del mensaje y entre paréntesis.

Para crear los mensajes de error nos fijaremos en los *storyboard* creados anteriormente, e iremos analizando los mensajes que pueden aparecer en cada pantalla. Los mensajes detectados son los siguientes:

Autenticación: Existe la posibilidad de que un usuario se identifique de forma incorrecta. Si esto ocurre el mensaje que se muestra es el siguiente *“Usuario y/o contraseña incorrectos, vuelva a intentarlo”*.

Creación de problemas: Cuando en el formulario de la creación de un nuevo problema no se rellenan todos los campos, aparecerá un mensaje de error indicando el campo. Los mensajes son *“El campo identificador del problema está vacío”*, *“El campo descripción está vacío”*, *“El campo umbral está vacío”*, *“El campo n° max de rondas está vacío”*, *“Hay algún campo de alternativa vacío”*.

También se comprobarán los valores de los campos rellenos. Por tanto, si el identificador del problema ya existe, se mostrará el siguiente mensaje *“El identificador del problema ya existe”*. Si el valor introducido en el campo umbral de consenso no es un número real entre 0 y 1, aparecerá el siguiente mensaje *“El umbral tiene que ser un valor entre 0 y 1”* y si el n° max de rondas no es un entero mayor que 1, aparecerá el siguiente mensaje *“El n° máximo de rondas tiene que ser mayor que 1”*.

En el caso de no seleccionar un experto en alguna de las listas desplegadas disponibles para asignar expertos, aparecerá el siguiente mensaje *“Vuelva a seleccionar los expertos”*,

Modificación de problemas: Cuando en el formulario de modificación de un problema se modifica el umbral de consenso y su valor no está entre 0 y 1, aparecerá el mensaje *“El umbral tiene que ser un valor entre 0 y 1”*. Si al modificar el valor del n^o máximo de rondas, éste no es mayor que 1, aparecerá el mensaje *“El n^o máximo de rondas tiene que ser mayor que 1”*. También se mostrará un mensaje de error si no se selecciona un nuevo experto para añadirlo *“Debe seleccionar un experto para añadirlo”*.

Creación de expertos: Si en el formulario de creación de un nuevo experto no se rellenan todos los campos, aparecerá un mensaje de error para indicar qué campo se ha quedado vacío. Los mensajes son *“El campo identificador del experto está vacío”*, *“El campo contraseña está vacío”*, *“El campo confirmar contraseña está vacío”*, *“El campo especialidad está vacío”*, *“El campo dirección de correo está vacío”*.

También se mostrarán mensajes para indicar qué valor introducido no es correcto. Así pues, si el identificador del experto ya existe, aparecerá *“El identificador del experto ya existe”*. Si el valor introducido para la contraseña no coincide con el de confirmar contraseña, aparecerá *“No coincide contraseña y confirmación de contraseña”*. Si en la dirección de correo electrónico no se ha escrito la letra @, se mostrará el siguiente mensaje *“La dirección de correo no es correcta”*.

Modificación de expertos: En el caso de modificar los campos contraseña, confirmar contraseña o dirección de correo, si los valores introducidos no son correctos, se mostrarán los mensajes comentados en *creación de expertos*.

Evolución de consenso de un problema: Cuando el Administrador desea consultar un problema y pulsa el botón *Buscar* sin seleccionar uno, se mostrará el siguiente mensaje *“No ha seleccionado un problema, vuelva a intentarlo”*. Si desea consultar información de una ronda y no selecciona ninguna, aparecerá el mensaje *“No ha seleccionado una ronda”*. Y si desea visualizar las preferencias de un experto y no selecciona un determinado experto y una ronda, se mostrará el mensaje *“Debe seleccionar experto y ronda”*.

Introducción de preferencias: Cuando el usuario Experto selecciona un problema para introducir sus preferencias, y dicho problema ya ha finalizado, se mostrará el siguiente mensaje *“El*

problema seleccionado ya ha finalizado”; si no ha finalizado, pero aún no se ha pasado a la siguiente ronda, se mostrará *“Debe esperar a que se realice la ronda actual”*,

Cuando en el formulario de introducir opinión, el experto no introduce alguna preferencia en un par de alternativas o el valor introducido no se encuentra dentro del dominio, aparecerá el siguiente mensaje *“Se ha equivocado en alguna preferencia, vuelva a intentarlo”*.

3.4.3.6 Mensajes de éxito.

Los mensajes de éxito son también muy importantes de cara a la usabilidad, ya que bien diseñados, permiten aumentar la confianza del usuario en el uso de la interfaz cuando una determinada acción ha finalizado correctamente.

En nuestra aplicación mostraremos los siguientes mensajes de éxito:

Creación de problemas: En este caso, será necesario mostrar un mensaje de éxito, ya que de lo contrario el Administrador no sabría si el problema se ha creado correctamente. El mensaje será el siguiente *“El problema se ha creado y guardado correctamente”*.

Modificación de problemas: Una vez que el Administrador ha modificado un problema, debe de aparecer el mensaje *“El problema se ha modificado y guardado correctamente”*, de esta forma al Administrador le queda claro que los cambios se han almacenado de forma correcta.

Eliminación de problemas: Cuando el Administrador elimina un problema debe de aparecer un mensaje de éxito, ya que de lo contrario no sabría si la operación se ha realizado satisfactoriamente, por tanto, el mensaje será el siguiente *“El problema se ha eliminado correctamente”*.

Creación de expertos: Será necesario mostrar el mensaje *“El experto se ha creado y guardado correctamente”* para indicar al Administrador que la operación se ha realizado con éxito.

Modificación de expertos: Si el Administrador modifica los datos de un experto, debe de aparecer el siguiente mensaje *“El experto se ha modificado y guardado correctamente”* para indicar que los cambios se ha realizado correctamente.

Eliminación de expertos: Después de eliminar un experto, aparecerá el mensaje *“El experto se ha eliminado correctamente”*, de esta forma al Administrador le queda claro que el experto se ha eliminado sin problemas.

Introducción de preferencias: Este mensaje será conveniente para mostrar al Experto que ha introducido sus preferencias adecuadamente. El mensaje será el siguiente *“Sus preferencias se han guardado correctamente”*.

3.4.3.7 Mensajes de confirmación.

Los mensajes de confirmación son también muy importantes, se utilizan para indicar al usuario si está seguro de que desea realizar la operación que ha indicado.

Los mensajes de confirmación que vamos a utilizar en nuestra aplicación son los siguientes:

Cuando el Administrador se encuentra en un formulario de creación o modificación y se cambia de opción: En este caso hay que mostrar un mensaje de confirmación ya que de lo contrario, el Administrador podría haber olvidado guardar los cambios y al realizar otra acción perdería los datos introducidos. El mensaje que aparecerá será el siguiente *“Desea cambiar de opción sin guardar?”*.

Cuando el Administrador va a eliminar un problema: En este caso es necesario mostrar un mensaje de confirmación, ya que una vez que se ha eliminado el problema no se podrá recuperar. El mensaje que se mostrará será el siguiente *“Está seguro que desea eliminar el problema?”*. Lo mismo sucederá si se desea eliminar un experto.

Cuando el Experto introduce sus preferencias y desea realizar otra tarea: Es conveniente mostrar un mensaje de confirmación ya que de lo contrario, podría haber olvidado enviar sus

preferencias y al realizar otra operación perdería todos los valores introducidos sin darse cuenta de ello. El mensaje será el siguiente “*Desea cambiar de opción sin enviar sus preferencias?*”.

3.5 Implementación.

La implementación es la actividad final de la Ingeniería del Software, aquella en la que el modelo obtenido en las actividades anteriores se debe transformar en código fuente. Para ello se debe ser cuidadoso en la elección del lenguaje de programación empleado para la codificación y de la herramienta utilizada para generarla.

En nuestro caso la elección del lenguaje de programación así como el de la herramienta utilizada para su desarrollo viene dado desde la definición del proyecto.

3.5.1 Tipo de la arquitectura de la aplicación.

En nuestro caso, vamos a desarrollar un sistema con una arquitectura cliente/servidor y una interfaz Web de comunicación con los usuarios. El funcionamiento de las arquitecturas de este tipo es sencilla: la aplicación se encuentra en un servidor central al que los usuarios acceden a través de un software cliente, en nuestro caso un navegador Web. Una vez que ha accedido a la aplicación, el usuario realiza peticiones que el servidor tiene que atender para generar una respuesta comprensible para el cliente.

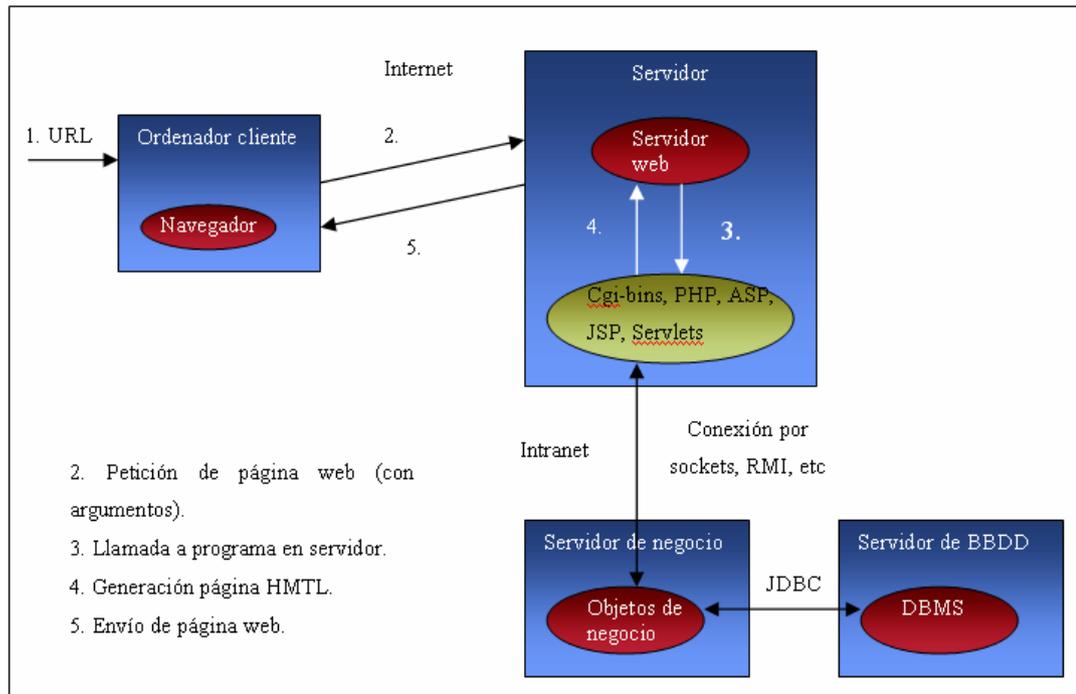


Figura 3.37: Arquitectura Cliente-Servidor

Una arquitectura cliente/servidor Web libera, por lo tanto, al usuario final de la aplicación de tener que instalarla en su máquina. Consigue que cada usuario sólo pueda acceder a la información que le corresponde. Además, este tipo de arquitectura, gracias a su diseño modular, es fácilmente escalable y ampliable tanto en nuevos clientes como en servidores añadidos.

3.5.2 Lenguajes de programación utilizados.

Resulta obvio ante la arquitectura y el funcionamiento previsto de nuestra aplicación que el uso de HTML simple y llano no es adecuado sino que, se necesita otro lenguaje capaz de generar contenido dinámico desde el servidor de manera transparente al usuario final. Para este fin hemos utilizado JSP.

Java Server Pages (JSP) es una tecnología Java que permite generar contenido dinámico para web, en forma de documentos HTML, XML o de otro tipo.

Los JSP's permiten la utilización de código Java mediante scripts. Además es posible utilizar algunas acciones JSP predefinidas mediante etiquetas. Estas etiquetas pueden ser enriquecidas mediante la utilización de Librerías de Etiquetas (TagLibs o Tag Libraries) externas e incluso personalizadas.

JSP no se puede considerar un script al 100%, ya que antes de ejecutarse el Servidor de Aplicaciones compila el contenido del documento JSP (script y etiquetas), y genera una clase Servlet. Por lo tanto, se puede decir que aunque este proceso sea transparente para el programador no deja de ser una tecnología compilada.

La principal ventaja de JSP frente a otros lenguajes es que permite integrarse con clases Java (.class) lo que permite separar en niveles las aplicaciones web, almacenando en clases java las partes que consumen más recursos (así como las que requieren más seguridad) y dejando la parte encargada de formatear el documento HTML en el archivo JSP. La idea fundamental detrás de este criterio es el de separar la lógica del negocio de la presentación de la información.

Para realizar una interfaz web adecuada se hace necesario el uso de otros dos lenguajes de programación: *CSS* y *Javascript*.

CSS, acrónimo de Cascade Style Sheets, es un lenguaje formal que ayuda a separar la estructura interna de un documento de su presentación externa. Las etiquetas de estilo CSS pueden presentarse tanto dentro de un documento HTML (encerradas dentro de las etiquetas `<style type="text/css"></style>` en la cabecera) como en un documento aparte (con extensión .css) al que el documento HTML se encarga de llamar cuando es necesario. De esta última manera no sólo, se consigue separar la estructura de la presentación sino que se consigue la centralización del estilo. Debido a que una sola hoja de estilos CSS puede ser invocada por distintas páginas de la aplicación web, lo que ayuda de manera muy importante al mantenimiento de la coherencia y consistencia del diseño de la aplicación.

Por su parte, **Javascript**, lenguaje interpretado de sintaxis similar a lenguajes como Java o C que se ejecuta del lado del cliente, ayuda a comprobar si los datos que introduce el cliente son erróneos informando de su error mediante mensajes de error o alerta.

Al igual que ocurre con CSS, el código Javascript puede ir incrustado dentro del documento HTML (entre las etiquetas `<script type="text/javascript"></script>` en el cuerpo o la cabecera) o estar almacenado en ficheros aparte (con extensión .js) y ser invocados por el documento. Para nuestra aplicación, para los estilos CSS nos hemos decantado por la segunda opción, mientras que para el código Javascript, al ser utilizado muy pocas veces lo hemos incrustado dentro del documento HTML.

Como ya se vio al comienzo de esta memoria, para la lógica de negocio del sistema se va a utilizar tecnología Servlets de JAVA.

3.5.3 Herramienta de desarrollo.

La herramienta escogida para desarrollar todo el código ha sido NetBeans (versión 6.0).

3.5.4 Instalación en el servidor y funcionamiento.

La instalación de la aplicación así como la puesta en marcha del servidor viene documentada paso a paso en el Anexo I.

Por su parte, en el Anexo II y Anexo III se encuentran disponibles los manuales de usuario para el Administrador y para el Experto, respectivamente.

3.6 Implantación y pruebas.

Esta fase de implantación junto con la del mantenimiento del software son partes muy importantes de la Ingeniería del software.

Consisten en desplegar el software realizado y tratar de mejorar u optimizar cualquier problema encontrado.

El mantenimiento de software involucra cualquier tipo de pruebas del software realizadas, las cuales son un elemento crítico para la garantía de calidad del software y representan una revisión final de las especificaciones, del diseño y de la codificación.

3.6.1 Pruebas y validación.

El objetivo de esta fase es realizar un conjunto de pruebas sobre el sistema. Con esto intentaremos conseguir llegar a un sistema sin errores garantizando, como hemos dicho, la calidad del software. Para comprobar esto realizaremos unas pruebas de sistema.

3.6.1.1 Casos de Test.

Los test diseñados son los siguientes:

Test 1: Autenticación de usuario

Requisitos testeados	RF-01
Acción	Un usuario, ya sea Administrador o Experto, introduce su nombre de usuario y contraseña y pulsa el botón <i>Administrador</i> o <i>Experto</i> correspondiente.
Checkpoint 1	El sistema debe mostrar el menú principal de dicho usuario.

Test 2: Autenticación de usuario incorrecta

Requisitos testeados	RF-01
Acción	Un usuario, ya sea Administrador o Experto, introduce su nombre de usuario y contraseña y pulsa el botón <i>Administrador</i> o <i>Experto</i> correspondiente.
Checkpoint 1	El sistema debe mostrar un mensaje de error informando de que el nombre de usuario o la contraseña o ambos son incorrectos.

Test 3: Crear un nuevo problema

Requisitos testeados	RF-03
Precondiciones	Un usuario Administrador identificado, se encuentra en el menú principal del administrador.
Acción	Accede a ' <i>Gestión de problemas</i> '.
Checkpoint 1	El sistema debe mostrar el submenú del administrador con las distintas opciones que el Administrador puede realizar en gestión de problemas.
Acción	Accede a ' <i>Crear</i> '.
Checkpoint 2	El sistema debe mostrar un formulario para introducir el nº de alternativas que se van a definir y el nº de expertos que van a participar en el problema.
Acción	Introduce el nº de alternativas y el nº de expertos y pulsa ' <i>Continuar</i> '.
Checkpoint 3	El sistema muestra el formulario para la creación del problema.
Acción	Introduce todos los datos del problema, define las alternativas, asigna los expertos y pulsa ' <i>Guardar</i> '.
Checkpoint 4	El sistema debe mostrar un mensaje de éxito de que el problema se ha creado correctamente.

Test 4: Crear un problema dejando el campo identificador en blanco

Requisitos testeados	RF-03
Precondiciones	Un usuario Administrador identificado, se encuentra en el menú principal del administrador.
Acción	Accede a ' <i>Gestión de problemas</i> '.
Checkpoint 1	El sistema debe mostrar el submenú del administrador con las distintas opciones que el Administrador puede realizar en gestión de problemas.
Acción	Accede a ' <i>Crear</i> '.
Checkpoint 2	El sistema debe mostrar un formulario para introducir el nº de alternativas que se van a definir y el nº de expertos que van a participar en el problema.
Acción	Introduce el nº de alternativas y el nº de expertos y pulsa ' <i>Continuar</i> '.
Checkpoint 3	El sistema muestra el formulario para la creación del problema.
Acción	Introduce todos los datos del problema excepto el campo identificador que lo deja en blanco y pulsa ' <i>Guardar</i> '.

Checkpoint 4	El sistema debe mostrar un mensaje de error indicando que el campo identificador está en blanco.
--------------	--

Test 5: Crear un problema dejando el campo descripción en blanco

Requisitos testeados	RF-03
Precondiciones	Un usuario Administrador identificado, se encuentra en el menú principal del administrador.
Acción	Accede a ' <i>Gestión de problemas</i> '.
Checkpoint 1	El sistema debe mostrar el submenú del administrador con las distintas opciones que el Administrador puede realizar en gestión de problemas.
Acción	Accede a ' <i>Crear</i> '.
Checkpoint 2	El sistema debe mostrar un formulario para introducir el nº de alternativas que se van a definir y el nº de expertos que van a participar en el problema.
Acción	Introduce el nº de alternativas y el nº de expertos y pulsa ' <i>Continuar</i> '.
Checkpoint 3	El sistema muestra el formulario para la creación del problema.
Acción	Introduce todos los datos del problema excepto el campo descripción que lo deja en blanco y pulsa ' <i>Guardar</i> '.
Checkpoint 4	El sistema debe mostrar un mensaje de error indicando que el campo descripción está en blanco.

Test 6: Crear un problema dejando el campo umbral de consenso en blanco

Requisitos testeados	RF-03
Precondiciones	Un usuario Administrador identificado, se encuentra en el menú principal del administrador.
Acción	Accede a ' <i>Gestión de problemas</i> '.
Checkpoint 1	El sistema debe mostrar el submenú del administrador con las distintas opciones que el Administrador puede realizar en gestión de problemas.
Acción	Accede a ' <i>Crear</i> '.
Checkpoint 2	El sistema debe mostrar un formulario para introducir el nº de alternativas que se van a definir y el nº de expertos que van a participar en el problema.
Acción	Introduce el nº de alternativas y el nº de expertos y pulsa ' <i>Continuar</i> '.

Checkpoint 3	El sistema muestra el formulario para la creación del problema.
Acción	Introduce todos los datos del problema excepto el campo umbral de consenso que lo deja en blanco y pulsa ' <i>Guardar</i> '.
Checkpoint 4	El sistema debe mostrar un mensaje de error indicando que el campo umbral de consenso está en blanco.

Test 7: Crear un problema dejando el campo n^o máximo de rondas en blanco

Requisitos testeados	RF-03
Precondiciones	Un usuario Administrador identificado, se encuentra en el menú principal del administrador.
Acción	Accede a ' <i>Gestión de problemas</i> '.
Checkpoint 1	El sistema debe mostrar el submenú del administrador con las distintas opciones que el Administrador puede realizar en gestión de problemas.
Acción	Accede a ' <i>Crear</i> '.
Checkpoint 2	El sistema debe mostrar un formulario para introducir el n ^o de alternativas que se van a definir y el n ^o de expertos que van a participar en el problema.
Acción	Introduce el n ^o de alternativas y el n ^o de expertos y pulsa ' <i>Continuar</i> '.
Checkpoint 3	El sistema muestra el formulario para la creación del problema.
Acción	Introduce todos los datos del problema excepto el campo n ^o máximo de rondas que lo deja en blanco y pulsa ' <i>Guardar</i> '.
Checkpoint 4	El sistema debe mostrar un mensaje de error indicando que el campo n ^o máximo de rondas está en blanco.

Test 8: Crear un problema dejando en blanco el campo descripción de alguna alternativa

Requisitos testeados	RF-03
Precondiciones	Un usuario Administrador identificado, se encuentra en el menú principal del administrador.
Acción	Accede a ' <i>Gestión de problemas</i> '.
Checkpoint 1	El sistema debe mostrar el submenú del administrador con las distintas opciones que el Administrador puede realizar en gestión de problemas.
Acción	Accede a ' <i>Crear</i> '.
Checkpoint 2	El sistema debe mostrar un formulario para introducir el nº de alternativas que se van a definir y el nº de expertos que van a participar en el problema.
Acción	Introduce el nº de alternativas y el nº de expertos y pulsa ' <i>Continuar</i> '.
Checkpoint 3	El sistema muestra el formulario para la creación del problema.
Acción	Introduce todos los datos del problema excepto el campo descripción de una o varias alternativas que los deja en blanco y pulsa ' <i>Guardar</i> '.
Checkpoint 4	El sistema debe mostrar un mensaje de error indicando que hay algún campo de alternativa en blanco.

Test 9: Crear un problema sin asignar uno o varios expertos

Requisitos testeados	RF-03
Precondiciones	Un usuario Administrador identificado, se encuentra en el menú principal del administrador.
Acción	Accede a ' <i>Gestión de problemas</i> '.
Checkpoint 1	El sistema debe mostrar el submenú del administrador con las distintas opciones que el Administrador puede realizar en gestión de problemas.
Acción	Accede a ' <i>Crear</i> '.
Checkpoint 2	El sistema debe mostrar un formulario para introducir el nº de alternativas que se van a definir y el nº de expertos que van a participar en el problema.
Acción	Introduce el nº de alternativas y el nº de expertos y pulsa ' <i>Continuar</i> '.
Checkpoint 3	El sistema muestra el formulario para la creación del problema.
Acción	Introduce todos los datos del problema, pero no selecciona los expertos que van a participar en el problema y pulsa ' <i>Guardar</i> '.

Checkpoint 4	El sistema debe mostrar un mensaje de error indicando que vuelva a seleccionar los expertos.
--------------	--

Test 10: Crear un problema con un identificador de problema que ya existe en la base de datos

Requisitos testeados	RF-03
Precondiciones	Un usuario Administrador identificado, se encuentra en el menú principal del administrador.
Acción	Accede a ' <i>Gestión de problemas</i> '.
Checkpoint 1	El sistema debe mostrar el submenú del administrador con las distintas opciones que el Administrador puede realizar en gestión de problemas.
Acción	Accede a ' <i>Crear.</i> '
Checkpoint 2	El sistema debe mostrar un formulario para introducir el nº de alternativas que se van a definir y el nº de expertos que van a participar en el problema.
Acción	Introduce el nº de alternativas y el nº de expertos y pulsa ' <i>Continuar</i> '.
Checkpoint 3	El sistema muestra el formulario para la creación del problema.
Acción	Introduce todos los datos del problema y pulsa ' <i>Guardar</i> '.
Checkpoint 4	El sistema debe mostrar un mensaje de error indicando que el identificador del problema ya existe en la base de datos.

Test 11: Crear un problema con un umbral de consenso que no esté en el intervalo [0,1]

Requisitos testeados	RF-03
Precondiciones	Un usuario Administrador identificado, se encuentra en el menú principal del administrador.
Acción	Accede a ' <i>Gestión de problemas</i> '.
Checkpoint 1	El sistema debe mostrar el submenú del administrador con las distintas opciones que el Administrador puede realizar en gestión de problemas.
Acción	Accede a ' <i>Crear.</i> '
Checkpoint 2	El sistema debe mostrar un formulario para introducir el nº de alternativas que se van a definir y el nº de expertos que van a participar en el problema.
Acción	Introduce el nº de alternativas y el nº de expertos y pulsa ' <i>Continuar</i> '.
Checkpoint 3	El sistema muestra el formulario para la creación del problema.

Acción	Introduce todos los datos del problema y pulsa ' <i>Guardar</i> '.
Checkpoint 4	El sistema debe mostrar un mensaje de error indicando que el umbral de consenso tiene que ser un valor entre 0 y 1.

Test 12: Crear un problema con un número máximo de rondas que no sea un número entero mayor que 1

Requisitos testeados	RF-03
Precondiciones	Un usuario Administrador identificado, se encuentra en el menú principal del administrador.
Acción	Accede a ' <i>Gestión de problemas</i> '.
Checkpoint 1	El sistema debe mostrar el submenú del administrador con las distintas opciones que el Administrador puede realizar en gestión de problemas.
Acción	Accede a ' <i>Crear</i> '.
Checkpoint 2	El sistema debe mostrar un formulario para introducir el nº de alternativas que se van a definir y el nº de expertos que van a participar en el problema.
Acción	Introduce el nº de alternativas y el nº de expertos y pulsa ' <i>Continuar</i> '.
Checkpoint 3	El sistema muestra el formulario para la creación del problema.
Acción	Introduce todos los datos del problema y pulsa ' <i>Guardar</i> '.
Checkpoint 4	El sistema debe mostrar un mensaje de error indicando que el nº máximo de rondas tiene que ser un valor entero mayor que 1.

Test 13: Crear un problema y pasar a realizar otra tarea sin guardar los datos

Requisitos testeados	RF-03
Precondiciones	Un usuario Administrador identificado, se encuentra en el menú principal del administrador.
Acción	Accede a ' <i>Gestión de problemas</i> '.
Checkpoint 1	El sistema debe mostrar el submenú del administrador con las distintas opciones que el Administrador puede realizar en gestión de problemas.
Acción	Accede a ' <i>Crear</i> '.
Checkpoint 2	El sistema debe mostrar un formulario para introducir el nº de alternativas que se van a definir y el nº de expertos que van a participar en el problema.

Acción	Introduce el nº de alternativas y el nº de expertos y pulsa ' <i>Continuar</i> '.
Checkpoint 3	El sistema muestra el formulario para la creación del problema.
Acción	Introduce todos los datos del problema y pulsa otra opción que no es ' <i>Guardar</i> '.
Checkpoint 4	El sistema debe mostrar un mensaje de confirmación preguntando si está seguro de cambiar de opción sin guardar los datos introducidos.

Test 14: Crear un nuevo experto

Requisitos testeados	RF-04
Precondiciones	Un usuario Administrador identificado, se encuentra en el menú principal del administrador.
Acción	Accede a ' <i>Gestión de expertos</i> '.
Checkpoint 1	El sistema debe mostrar el submenú del administrador con las distintas opciones que el Administrador puede realizar en gestión de expertos.
Acción	Accede a ' <i>Crear</i> '.
Checkpoint 2	El sistema muestra el formulario para la creación del experto.
Acción	Introduce todos los datos del experto y pulsa ' <i>Guardar</i> '.
Checkpoint 3	El sistema debe mostrar un mensaje de éxito de que el experto se ha creado correctamente.

Test 15: Crear un experto dejando en blanco el campo identificador

Requisitos testeados	RF-04
Precondiciones	Un usuario Administrador identificado, se encuentra en el menú principal del administrador.
Acción	Accede a ' <i>Gestión de expertos</i> '.
Checkpoint 1	El sistema debe mostrar el submenú del administrador con las distintas opciones que el Administrador puede realizar en gestión de expertos.
Acción	Accede a ' <i>Crear</i> '.
Checkpoint 2	El sistema muestra el formulario para la creación del experto.
Acción	Introduce todos los datos del experto dejando en blanco el campo identificador y pulsa ' <i>Guardar</i> '.

Checkpoint 3	El sistema debe mostrar un mensaje de error indicando que el campo identificador está en blanco.
--------------	--

Test 16: Crear un experto dejando en blanco el campo contraseña

Requisitos testeados	RF-04
Precondiciones	Un usuario Administrador identificado, se encuentra en el menú principal del administrador.
Acción	Accede a ' <i>Gestión de expertos</i> '.
Checkpoint 1	El sistema debe mostrar el submenú del administrador con las distintas opciones que el Administrador puede realizar en gestión de expertos.
Acción	Accede a ' <i>Crear</i> '.
Checkpoint 2	El sistema muestra el formulario para la creación del experto.
Acción	Introduce todos los datos del experto dejando en blanco el campo contraseña y pulsa ' <i>Guardar</i> '.
Checkpoint 3	El sistema debe mostrar un mensaje de error indicando que el campo contraseña está en blanco.

Test 17: Crear un experto dejando en blanco el campo confirmar contraseña

Requisitos testeados	RF-04
Precondiciones	Un usuario Administrador identificado, se encuentra en el menú principal del administrador.
Acción	Accede a ' <i>Gestión de expertos</i> '.
Checkpoint 1	El sistema debe mostrar el submenú del administrador con las distintas opciones que el Administrador puede realizar en gestión de expertos.
Acción	Accede a ' <i>Crear</i> '.
Checkpoint 2	El sistema muestra el formulario para la creación del experto.
Acción	Introduce todos los datos del experto dejando en blanco el campo confirmar contraseña y pulsa ' <i>Guardar</i> '.
Checkpoint 3	El sistema debe mostrar un mensaje de error indicando que el campo confirmar contraseña está en blanco.

Test 18: Crear un experto dejando en blanco el campo especialidad

Requisitos testeados	RF-04
Precondiciones	Un usuario Administrador identificado, se encuentra en el menú principal del administrador.
Acción	Accede a ' <i>Gestión de expertos</i> '.
Checkpoint 1	El sistema debe mostrar el submenú del administrador con las distintas opciones que el Administrador puede realizar en gestión de expertos.
Acción	Accede a ' <i>Crear.</i> '
Checkpoint 2	El sistema muestra el formulario para la creación del experto.
Acción	Introduce todos los datos del experto dejando en blanco el campo especialidad y pulsa ' <i>Guardar</i> '.
Checkpoint 3	El sistema debe mostrar un mensaje de error indicando que el campo especialidad está en blanco.

Test 19: Crear un experto dejando en blanco el campo dirección de correo electrónico.

Requisitos testeados	RF-04
Precondiciones	Un usuario Administrador identificado, se encuentra en el menú principal del administrador.
Acción	Accede a ' <i>Gestión de expertos</i> '.
Checkpoint 1	El sistema debe mostrar el submenú del administrador con las distintas opciones que el Administrador puede realizar en gestión de expertos.
Acción	Accede a ' <i>Crear.</i> '
Checkpoint 2	El sistema muestra el formulario para la creación del experto.
Acción	Introduce todos los datos del experto dejando en blanco el campo dirección de correo y pulsa ' <i>Guardar</i> '.
Checkpoint 3	El sistema debe mostrar un mensaje de error indicando que el campo dirección de correo está en blanco.

Test 20: Crear un experto con un identificador de experto que ya existe en la base de datos

Requisitos testeados	RF-04
Precondiciones	Un usuario Administrador identificado, se encuentra en el menú principal del administrador.
Acción	Accede a ' <i>Gestión de expertos</i> '.
Checkpoint 1	El sistema debe mostrar el submenú del administrador con las distintas opciones que el Administrador puede realizar en gestión de expertos.
Acción	Accede a ' <i>Crear.</i> '
Checkpoint 2	El sistema muestra el formulario para la creación del experto.
Acción	Introduce todos los datos del experto y pulsa ' <i>Guardar</i> '.
Checkpoint 3	El sistema debe mostrar un mensaje de error indicando que el identificador introducido ya existe en la base de datos.

Test 21: Crear un experto rellenando con distintos valores los campos contraseña y confirmación de contraseña

Requisitos testeados	RF-04
Precondiciones	Un usuario Administrador identificado, se encuentra en el menú principal del administrador.
Acción	Accede a ' <i>Gestión de expertos</i> '.
Checkpoint 1	El sistema debe mostrar el submenú del administrador con las distintas opciones que el Administrador puede realizar en gestión de expertos.
Acción	Accede a ' <i>Crear.</i> '
Checkpoint 2	El sistema muestra el formulario para la creación del experto.
Acción	Introduce todos los datos del experto y pulsa ' <i>Guardar</i> '.
Checkpoint 3	El sistema debe mostrar un mensaje de error indicando que la contraseña y confirmación de contraseña no coinciden.

Test 22: Crear un experto con una dirección de correo errónea

Requisitos testeados	RF-04
Precondiciones	Un usuario Administrador identificado, se encuentra en el menú principal del administrador.
Acción	Accede a ' <i>Gestión de expertos</i> '.
Checkpoint 1	El sistema debe mostrar el submenú del administrador con las distintas opciones que el Administrador puede realizar en gestión de expertos.
Acción	Accede a ' <i>Crear.</i> '
Checkpoint 2	El sistema muestra el formulario para la creación del experto.
Acción	Introduce todos los datos del experto y pulsa ' <i>Guardar</i> '.
Checkpoint 3	El sistema debe mostrar un mensaje de error indicando que la dirección de correo no es correcta.

Test 23: Crear un experto y cambiar de opción sin guardar los datos introducidos

Requisitos testeados	RF-04
Precondiciones	Un usuario Administrador identificado, se encuentra en el menú principal del administrador.
Acción	Accede a ' <i>Gestión de expertos</i> '.
Checkpoint 1	El sistema debe mostrar el submenú del administrador con las distintas opciones que el Administrador puede realizar en gestión de expertos.
Acción	Accede a ' <i>Crear.</i> '
Checkpoint 2	El sistema muestra el formulario para la creación del experto.
Acción	Introduce todos los datos del experto y pulsa otra opción que no es ' <i>Guardar</i> '.
Checkpoint 3	El sistema debe mostrar un mensaje de confirmación preguntando si está seguro de cambiar de opción sin guardar los datos introducidos.

Test 24: Calcular el nivel de consenso de un problema que todavía no ha sido valorado por algún o ningún experto

Requisitos testeados	RF-05
Precondiciones	Un usuario Administrador identificado, se encuentra en el menú principal del administrador.
Acción	Accede a ' <i>Estado problema</i> '.
Checkpoint 1	El sistema debe mostrar un listado con todos los problemas creados.
Acción	Selecciona un problema y pulsa ' <i>Buscar</i> '.
Checkpoint 2	El sistema muestra algunos datos referentes al problema seleccionado y muestra las opciones que el Administrador puede realizar.
Acción	Pulsa ' <i>Calcular nivel de consenso</i> '.
Checkpoint 3	El sistema debe mostrar un formulario en el que se indique que no se puede calcular el nivel de consenso porque hay expertos (indicar cuáles) que aún no han introducido sus preferencias.

Test 25: Calcular el nivel de consenso de un problema en progreso

Requisitos testeados	RF-05
Precondiciones	Un usuario Administrador identificado, se encuentra en el menú principal del administrador.
Acción	Accede a ' <i>Estado problema</i> '.
Checkpoint 1	El sistema debe mostrar un listado con todos los problemas creados.
Acción	Selecciona un problema y pulsa ' <i>Buscar</i> '.
Checkpoint 2	El sistema muestra algunos datos referentes al problema seleccionado y muestra las opciones que el Administrador puede realizar.
Acción	Pulsa ' <i>Calcular nivel de consenso</i> '.
Checkpoint 3	El sistema muestra un formulario que indica si se ha alcanzado o no el consenso, cuál es el grado de consenso obtenido y nº de ronda actual.

Test 26: Consultar la evolución de consenso de un problema

Requisitos testeados	RF-06
Precondiciones	Un usuario Administrador identificado, se encuentra en el menú principal del administrador.
Acción	Accede a <i>'Estado problema'</i> .
Checkpoint 1	El sistema debe mostrar un listado con todos los problemas creados.
Acción	Selecciona un problema y pulsa <i>'Buscar'</i> .
Checkpoint 2	El sistema muestra algunos datos referentes al problema seleccionado y muestra las opciones que el Administrador puede realizar.
Acción	Pulsa <i>'Evolución consenso'</i> .
Checkpoint 3	El sistema debe mostrar un formulario con todos los datos del problema, matriz de consenso, matriz de opiniones colectivas y grado de consenso obtenido en la última ronda realizada.

Test 27: Consultar el grado de consenso de un problema sin seleccionar una determinada ronda

Requisitos testeados	RF-06
Precondiciones	Un usuario Administrador identificado, se encuentra en el menú principal del administrador.
Acción	Accede a <i>'Estado problema'</i> .
Checkpoint 1	El sistema debe mostrar un listado con todos los problemas creados.
Acción	Selecciona un problema y pulsa <i>'Buscar'</i> .
Checkpoint 2	El sistema muestra algunos datos referentes al problema seleccionado y muestra las opciones que el Administrador puede realizar.
Acción	Pulsa <i>'Evolución consenso'</i> .
Checkpoint 3	El sistema muestra un formulario con todos los datos del problema, matriz de consenso, matriz de opiniones colectivas y grado de consenso obtenido en la última ronda realizada.
Acción	Pulsa <i>'Ver grado consenso'</i> .
Checkpoint 4	El sistema muestra un mensaje de error indicando que no ha seleccionado una ronda.

Test 28: Consultar el grado de consenso de una determinada ronda de un problema

Requisitos testeados	RF-06
Precondiciones	Un usuario Administrador identificado, se encuentra en el menú principal del administrador.
Acción	Accede a ' <i>Estado problema</i> '.
Checkpoint 1	El sistema debe mostrar un listado con todos los problemas creados.
Acción	Selecciona un problema y pulsa ' <i>Buscar</i> '.
Checkpoint 2	El sistema muestra algunos datos referentes al problema seleccionado y muestra las opciones que el Administrador puede realizar.
Acción	Pulsa ' <i>Evolución consenso</i> '.
Checkpoint 3	El sistema muestra un formulario con todos los datos del problema, matriz de consenso, matriz de opiniones colectivas y grado de consenso obtenido en la última ronda realizada.
Acción	Selecciona una ronda y pulsa ' <i>Ver grado consenso</i> '.
Checkpoint 4	El sistema muestra el grado de consenso, el número de cambios totales realizados en dicha ronda y el número de cambios que cada experto ha realizado.

Test 29: Consultar las preferencias introducidas por un experto en un problema sin seleccionar el identificador de un experto y una determinada ronda

Requisitos testeados	RF-06
Precondiciones	Un usuario Administrador identificado, se encuentra en el menú principal del administrador.
Acción	Accede a ' <i>Estado problema</i> '.
Checkpoint 1	El sistema debe mostrar un listado con todos los problemas creados.
Acción	Selecciona un problema y pulsa ' <i>Buscar</i> '.
Checkpoint 2	El sistema muestra algunos datos referentes al problema seleccionado y muestra las opciones que el Administrador puede realizar.
Acción	Pulsa ' <i>Evolución consenso</i> '.
Checkpoint 3	El sistema muestra un formulario con todos los datos del problema, matriz de consenso, matriz de opiniones colectivas y grado de consenso obtenido en

Acción	la última ronda realizada.
Checkpoint 4	Pulsa ' <i>Ver preferencias</i> '. El sistema muestra un mensaje de error indicando que debe seleccionar experto y ronda.

Test 30: Consultar las preferencias introducidas por un determinado experto en una ronda de un problema

Requisitos testeados	RF-06
Precondiciones	Un usuario Administrador identificado, se encuentra en el menú principal del administrador.
Acción	Accede a ' <i>Estado problema</i> '.
Checkpoint 1	El sistema debe mostrar un listado con todos los problemas creados.
Acción	Selecciona un problema y pulsa ' <i>Buscar</i> '.
Checkpoint 2	El sistema muestra algunos datos referentes al problema seleccionado y muestra las opciones que el Administrador puede realizar.
Acción	Pulsa ' <i>Evolución consenso</i> '.
Checkpoint 3	El sistema muestra un formulario con todos los datos del problema, matriz de consenso, matriz de opiniones colectivas y grado de consenso obtenido en la última ronda realizada.
Acción	Selecciona un experto y una ronda y pulsa ' <i>Ver preferencias</i> '.
Checkpoint 4	El sistema muestra la matriz de preferencias que el experto ha introducido en la ronda indicada.

Test 31: Valorar un problema en progreso

Requisitos testeados	RF-07
Precondiciones	Un usuario Experto identificado, se encuentra en el menú principal del experto, en el que se muestran todos los problemas que el Experto tiene asignados.
Acción	Selecciona un problema y pulsa ' <i>Introducir opinión</i> '.
Checkpoint 1	El sistema muestra un formulario con algunos datos referentes al problema seleccionado y una matriz de preferencias. Si es la segunda ronda o posterior el Experto podrá ver las recomendaciones que se le generaron en la anterior ronda.
Acción	Rellena la matriz de preferencias y pulsa ' <i>Enviar</i> '.
Checkpoint 2	El sistema muestra un mensaje de éxito indicando que las preferencias se han guardado correctamente.

Test 32: Valorar un problema que ya ha sido valorado

Requisitos testeados	RF-07
Precondiciones	Un usuario Experto identificado, se encuentra en el menú principal del experto, en el que se muestran todos los problemas que el Experto tiene asignados.
Acción	Selecciona un problema y pulsa ' <i>Introducir opinión</i> '.
Checkpoint 1	El sistema debe mostrar un mensaje de error indicando que debe esperar a la siguiente ronda para poder introducir sus preferencias.

Test 33: Valorar un problema que ya ha finalizado

Requisitos testeados	RF-07
Precondiciones	Un usuario Experto identificado, se encuentra en el menú principal del experto, en el que se muestran todos los problemas que el Experto tiene asignados.
Acción	Selecciona un problema y pulsa ' <i>Introducir opinión</i> '.
Checkpoint 1	El sistema debe mostrar un mensaje de error indicando que el problema ya ha finalizado.

Test 34: Valorar un problema sin seleccionar uno

Requisitos testeados	RF-07
Precondiciones	Un usuario Experto identificado, se encuentra en el menú principal del experto, en el que se muestran todos los problemas que el Experto tiene asignados.
Acción	Pulsa ' <i>Introducir opinión</i> '.
Checkpoint 1	El sistema debe mostrar un mensaje de error indicando que debe seleccionar un problema.

Test 35: Valorar un problema y cambiar de opción sin enviar las preferencias introducidas

Requisitos testeados	RF-07
Precondiciones	Un usuario Experto identificado, se encuentra en el menú principal del experto, en el que se muestran todos los problemas que el Experto tiene asignados.
Acción	Selecciona un problema y pulsa ' <i>Introducir opinión</i> '.
Checkpoint 1	El sistema muestra un formulario con algunos datos referentes al problema seleccionado y una matriz de preferencias. Si es la segunda ronda o posterior el Experto podrá ver las recomendaciones que se le generaron en la anterior ronda.
Acción	Rellena la matriz de preferencias y pulsa otra opción que no es ' <i>Enviar</i> '.
Checkpoint 2	El sistema muestra un mensaje de confirmación preguntando si está seguro de que desea cambiar de opción sin enviar sus preferencias.

Test 36: Valorar un problema introduciendo uno o varios valores fuera del dominio

Requisitos testeados	RF-07
Precondiciones	Un usuario Experto identificado, se encuentra en el menú principal del experto, en el que se muestran todos los problemas que el Experto tiene asignados.
Acción	Selecciona un problema y pulsa ' <i>Introducir opinión</i> '.
Checkpoint 1	El sistema muestra un formulario con algunos datos referentes al problema seleccionado y una matriz de preferencias. Si es la segunda ronda o posterior

Acción	el Experto podrá ver las recomendaciones que se le generaron en la anterior ronda.
Checkpoint 2	Rellena la matriz de preferencias y pulsa ' <i>Enviar</i> '. El sistema muestra un mensaje de error indicando que el valor de alguna preferencia no es correcto.

Test 37: Consultar el estado de un problema

Requisitos testeados	RF-08
Precondiciones	Un usuario Experto identificado, se encuentra en el menú principal del experto, en el que se muestran todos los problemas que el Experto tiene asignados.
Acción	Selecciona un problema y pulsa ' <i>Consultar problema</i> '.
Checkpoint 1	El sistema muestra un formulario con todos los datos del problema seleccionado, las preferencias y las recomendaciones en caso que las hubiera de la última ronda.

Test 38: Consultar el estado de un problema sin seleccionar uno

Requisitos testeados	RF-08
Precondiciones	Un usuario Experto identificado, se encuentra en el menú principal del experto, en el que se muestran todos los problemas que el Experto tiene asignados.
Acción	Pulsa ' <i>Consultar problema</i> '.
Checkpoint 1	El sistema debe mostrar un mensaje de error indicando que debe seleccionar un problema.

Test 39: Salir de la aplicación

Requisitos testeados	RF-02
Precondiciones	Un usuario identificado, se encuentra en su menú correspondiente.
Acción	Pulsa ' <i>Salir</i> '.
Checkpoint 1	El sistema debe mostrar un formulario indicando que ha cerrado la sesión y dando la posibilidad de acceder de nuevo a la aplicación.

3.6.1.2 Resultados obtenidos.

A continuación mostramos una tabla con los resultados obtenidos, después de realizar los test diseñados sobre nuestra aplicación.

<i>TEST</i>	<i>RESULTADO</i>
<i>Test 1</i>	
Checkpoint 1	OK
<i>Test 2</i>	
Checkpoint 1	OK
<i>Test 3</i>	
Checkpoint 1	OK
Checkpoint 2	OK
Checkpoint 3	OK
Checkpoint 4	OK
<i>Test 4</i>	
Checkpoint 1	OK
Checkpoint 2	OK
Checkpoint 3	OK
Checkpoint 4	OK
<i>Test 5</i>	
Checkpoint 1	OK
Checkpoint 2	OK
Checkpoint 3	OK
Checkpoint 4	OK
<i>Test 6</i>	
Checkpoint 1	OK
Checkpoint 2	OK
Checkpoint 3	OK
Checkpoint 4	OK

<i>Test 7</i>	
Checkpoint 1	OK
Checkpoint 2	OK
Checkpoint 3	OK
Checkpoint 4	OK
<i>Test 8</i>	
Checkpoint 1	OK
Checkpoint 2	OK
Checkpoint 3	OK
Checkpoint 4	OK
<i>Test 9</i>	
Checkpoint 1	OK
Checkpoint 2	OK
Checkpoint 3	OK
Checkpoint 4	OK
<i>Test 10</i>	
Checkpoint 1	OK
Checkpoint 2	OK
Checkpoint 3	OK
Checkpoint 4	OK
<i>Test 11</i>	
Checkpoint 1	OK
Checkpoint 2	OK
Checkpoint 3	OK
Checkpoint 4	OK
<i>Test 12</i>	
Checkpoint 1	OK
Checkpoint 2	OK
Checkpoint 3	OK
Checkpoint 4	OK

<i>Test 13</i>	
Checkpoint 1	OK
Checkpoint 2	OK
Checkpoint 3	OK
Checkpoint 4	OK
<i>Test 14</i>	
Checkpoint 1	OK
Checkpoint 2	OK
Checkpoint 3	OK
<i>Test 15</i>	
Checkpoint 1	OK
Checkpoint 2	OK
Checkpoint 3	OK
<i>Test 16</i>	
Checkpoint 1	OK
Checkpoint 2	OK
Checkpoint 3	OK
<i>Test 17</i>	
Checkpoint 1	OK
Checkpoint 2	OK
Checkpoint 3	OK
<i>Test 18</i>	
Checkpoint 1	OK
Checkpoint 2	OK
Checkpoint 3	OK
<i>Test 19</i>	
Checkpoint 1	OK
Checkpoint 2	OK
Checkpoint 3	OK

<i>Test 20</i>	
Checkpoint 1	OK
Checkpoint 2	OK
Checkpoint 3	OK
<i>Test 21</i>	
Checkpoint 1	OK
Checkpoint 2	OK
Checkpoint 3	OK
<i>Test 22</i>	
Checkpoint 1	OK
Checkpoint 2	OK
Checkpoint 3	OK
<i>Test 23</i>	
Checkpoint 1	OK
Checkpoint 2	OK
Checkpoint 3	OK
<i>Test 24</i>	
Checkpoint 1	OK
Checkpoint 2	OK
Checkpoint 3	OK
<i>Test 25</i>	
Checkpoint 1	OK
Checkpoint 2	OK
Checkpoint 3	OK
<i>Test 26</i>	
Checkpoint 1	OK
Checkpoint 2	OK
Checkpoint 3	OK

<i>Test 27</i>	
Checkpoint 1	OK
Checkpoint 2	OK
Checkpoint 3	OK
Checkpoint 4	OK
<i>Test 28</i>	
Checkpoint 1	OK
Checkpoint 2	OK
Checkpoint 3	OK
Checkpoint 4	OK
<i>Test 29</i>	
Checkpoint 1	OK
Checkpoint 2	OK
Checkpoint 3	OK
Checkpoint 4	OK
<i>Test 30</i>	
Checkpoint 1	OK
Checkpoint 2	OK
Checkpoint 3	OK
Checkpoint 4	OK
<i>Test 31</i>	
Checkpoint 1	OK
Checkpoint 2	OK
<i>Test 32</i>	
Checkpoint 1	OK
<i>Test 33</i>	
Checkpoint 1	OK
<i>Test 34</i>	
Checkpoint 1	OK

<i>Test 35</i>	
Checkpoint 1	OK
Checkpoint 2	OK
<i>Test 36</i>	
Checkpoint 1	OK
Checkpoint 2	OK
<i>Test 37</i>	
Checkpoint 1	OK
<i>Test 38</i>	
Checkpoint 1	OK
<i>Test 39</i>	
Checkpoint 1	OK

Figura 3.38: Resultados de las pruebas de test.

CAPÍTULO 4

CONCLUSIONES

4.1 Conclusiones finales.

El propósito de este proyecto era realizar una aplicación web capaz de automatizar el proceso de consenso en Toma de Decisión en Grupo en Contextos Heterogéneos, es decir, que sea capaz de llevar a cabo procesos de consenso donde los expertos expresen sus preferencias usando diferentes dominios de información (numérica, intervalar y lingüística) y generar recomendaciones para ayudar a conseguir un mayor grado de consenso en la siguiente ronda.

Hay que destacar, la importancia que tiene implantar un sistema como este en la gestión de empresas, pues permite que los expertos puedan dar su opinión desde su puesto de trabajo sin necesidad de realizar reuniones presenciales y sin un moderador que las guíe. El sistema presentado intenta que se tengan en cuenta las opiniones de todos los expertos, para alcanzar una solución para evitar que ninguno pueda decir que no se le ha considerado. Estas opiniones serán expresadas en el dominio de expresión más adecuado para cada experto y que podrá ser numérico, lingüístico e intervalar. Además, generará recomendaciones que los expertos deben seguir para aproximar sus opiniones y alcanzar de esta forma el consenso, independientemente del dominio de expresión utilizado por los expertos.

Podemos decir, que el propósito del proyecto y los objetivos indicados en el capítulo 1, se han alcanzado con el software desarrollado.

Por otro lado, no debemos olvidar que para diseñar y desarrollar este sistema hemos seguido detenidamente los pasos de la Ingeniería del Software. Para ello hemos comprendido los objetivos de cada etapa y se han puesto en práctica cada una de ellas, observando que todo lo estudiado en la Ingeniería del Software ha dado su fruto.

Finalmente, comentar que, la realización de este proyecto ha servido para poner en práctica muchas de las metodologías y habilidades adquiridas durante los años de formación académica y otras muchas que he tenido que adquirir durante su desarrollo.

Trabajos futuros

En el futuro se podría abordar diferentes mejoras en el software presentado, tales como:

- ✓ Que el Administrador pueda establecer el dominio en cada problema creado, ya sea éste, de tipo numérico, intervalar o lingüístico.
- ✓ Que ofrezca la posibilidad de utilizar diferentes operadores de agregación.
- ✓ Que incluya el Proceso de Selección, una vez alcanzado el consenso.

BIBLIOGRAFÍA

Bibliografía.

- [1] C. Alcalde, A. Burusco and R. Fuentes-González. A constructive method for the definition of interval-valued fuzzy implications operators. *Fuzzy Sets and Systems*, 153(2):211-227, 2005.
- [2] R. Anthony. *Planning and Control Systems: A Framework for Analysis*. Harvard Business School Division or Research Press, 1965.
- [3] K. J. Arrow. *Social Choice and Individual Values*. Yale University Press, 1963.
- [4] G. Beliakov, R. Mesiar, and L. Valaskova. Fitting generated aggregation operators to empirical data. *International Journal of Uncertainty Fuzziness and Knowledge-Based Systems*, 12(2):219-236, 2004.
- [5] G. Beliakov and J. Warren Appropriate choice of aggregation operators in fuzzy decision support systems. *IEEE Transactions on Fuzzy Systems*, 9(6):773-784, 2001.
- [6] [15P]P.P. Bonissone and K.S. Decker. *Selecting Uncertainty Calculi and Granularity: An Experiment in Trading-Off Precision and Complexity*, 217-247, North-Holland, 1986.
- [7] G. Bordogna, M. Fedrizzi and G. Pasi. A linguistic modelling of consensus in group decision making based on OWA operators. *IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans*, 27:126-132, 1997.
- [8] N. Bryson. Group decision making and the analytic hierarchy process: Exploring the consensus-relevant information content. *Computers and Operational Research*, 1(23):27-35, 1996.
- [9] T. Calvo, R. Mesiar, and RR. Yager. Quantitative weights and aggregation. *IEEE Transactions on Fuzzy Systems*, 12(1):62-69, 2004.
- [10] C. Carlsson and R. Fullér. Benchmarking and linguistic importance weighted aggregations. *Fuzzy Sets and Systems*, 114:35-41, 2000.
- [11] [34P] C.H. Cheng and Y. Lin. Evaluating the best main battle tank using fuzzy decision theory with linguistic criteria evaluation. *European Journal of Operational Research*, 142:174-186, 2002.
- [12] V. Cutello and J. Montero. Hierarchies of aggregation operators. *International Journal of Intelligent Systems*, 9(11):1025-1045, 1994.
- [13] R. Degani and G. Bortolan. The problem of linguistic approximation in clinical decision making. *International Journal of Approximate Reasoning*, 2:143-162, 1988.
- [14] M. Delgado, F. Herrera, E. Herrera-Viedma, and L. Martínez. Combining numerical and linguistic information in group decision making. *Information Sciences*, 107:177-194, 1998.
- [15] M. Delgado, J.L. Verdegay, and M.A Vila. Linguistic decision making models. *International Journal of Intelligent Systems*, 7:479-492, 1992.

- [16] M. Delgado, J.L. Verdegay, and M.A Vila. On aggregation operations of linguistic labels. *International Journal of Intelligent System*, 8:351-370, 1993.
- [17] J. Doyle. Prospects for preferences. *Computational Intelligence*, 20(2):111-136, 2004.
- [18] D. Dubois and H. Prade. A review of fuzzy set aggregation connectives. *Information Science*, 36:85-121, 1985.
- [19] D. Dubois and H. Prade. Weighted minimum and maximum operations in fuzzy set theory. *Information Science*, 39:205-210, 1986.
- [20] C. Edwards, J. Ward, and A. Bytheway. *Fundamentos de Sistemas de Información*. Prentice Hall, 1998.
- [21] Z.-P. Fan, J. Ma and Q. Zhang. An approach to multiple attribute decision making based on fuzzy preference information alternatives. *Fuzzy Sets and Systems*, 131(1):101-106, 2002.
- [22] Z. P. Fan, S.H. Xiao, and G.F. Hu. An optimization method for integrating two kinds of preference information in group decision making. *Computers Industrial Engineering*. 46(2):329-335, 2004.
- [23] J. Fodor and M. Roubens. *Fuzzy preference modelling and multicriteria decision support*. Kluwer Academic Publishers, 1994.
- [24] L. Godo and V. Torra. On aggregation operators for ordinal qualitative information. *IEEE Transactions on Fuzzy Systems*, 8(2):143-154, 2000.
- [25] F. Herrera and E. Herrera-Viedman. Choice functions and mechanisms for linguistic preference relations. *European Journal of Operational Research*, 120:144-161, 2000.
- [26] F. Herrera and E. Herrera-Viedma. Linguistic Decision Analysis: Steps for Solving Decision Problems under Linguistic Information. *Fuzzy Sets and Systems*, 115:67-82, 2000.
- [27] F. Herrera, E. Herrera-Viedman and J.L. Verdegay. A sequential selection process in group decision making with linguistic assessment. *Information Sciences*, 85:223-230, 1995.
- [28] F. Herrera, E. Herrera-Viedman and J.L. Verdegay. Direct approach processes in group decision making using linguistic OWA operators. *Fuzzy Sets and Systems*, 79:175-190, 1996.
- [29] F. Herrera and L. Martínez. A model based on linguistic 2-tuples for dealing with multigranularity hierarchical linguistic contexts in multiexpert decision-making. *IEEE Transactions on Systems, Man and Cybernetics. Part B: Cybernetics*, 31(2):227-234, 2001.
- [30] F. Herrera, L. Martinez and P. J. Sanchez. Managing heterogeneous information in group decision making. *In Proceedings Ninth International Conference IPMU 2002*, Annecy (France), 439-446, 2002.

- [31] F. Herrera, L. Martínez and P. J. Sánchez. Managing non-homogeneous information in group decision making. *European Journal of Operational Research*, 166(1):115-132, 2005.
- [32] F. Herrera, E. and J.L. Verdegay. Linguistic assessments in group decision. *In First European Congress on Fuzzy and Intelligent Technologies*, 941-948, Aachen, 1993.
- [33] E. Herrera-Viedma, L. Martínez, F. Mata and F. Chiclana. A Consensus support system model for group decision-making problems with multi-granular linguistic preference relations. *IEEE Transactions on Fuzzy Systems*, 13(5):644-658, 2005.
- [34] E. Herrera-Viedma, F. Mata, L. Martínez, F. Chiclana and L. G. Perez. Measurements of consensus in multi-granular linguistic group decision making. *Lecture Notes in Artificial Intelligence*, 3131:194-204, 2004.
- [35] J. Kacprzyk. Group decision making with a fuzzy linguistic majority. *Fuzzy Sets and Systems*, 18:105-118, 1986.
- [36] D. Kuchta. Fuzzy capital budgeting. *Fuzzy Sets and Systems*, 111:367-385, 2000.
- [37] L.I.Kuncheva. Five measures of consensus in GDM using fuzzy sets. *In International Conference on Fuzzy Sets and Applications IFSA-95*, 141-144, 1991.
- [38] S. Kundu. Min-transitivity of fuzzy leftness relationship and its application to decision making. *Fuzzy Sets and Systems*, 86:357-367, 1997.
- [39] H. M. Lee. Generalization of the group decision making using fuzzy sets theory for evaluating the rate of aggregate risk in software development. *Information Sciences*, 113:301-311, 1999.
- [40] E. Levrat, A. Voisin, S. Bombardier and J. Bremont. Subjective evaluation of car seat comfort with fuzzy set techniques. *International Journal of Intelligent Systems*, 12:891-913. 1997.
- [41] D. Li and J. B. Yang. A multiattribute decision making approach using intuitionistic fuzzy sets. *In Proceedings Eusflat 2003*, Zitaú, 183-186, 2003.
- [42] M. Marimin, M. Umamo, I. Hatono and H. Tamura. Linguistic labels for expressing fuzzy preference relation in fuzzy group decision making. *IEEE Transaction on Systems, Man and Cybernetics, Part B: Cybernetics*, 28:205-218, 1998.
- [43] S.A Orlovsky. Decision-making with a fuzzy preference relation. *Fuzzy Sets Systems*, 1:155-167, 1978.
- [44] M. H. Rasmy, S. M. Lee, W. F. A. El-Wahed, A. M. Ragab and M. M. El-Sherbiny. An expert system for multiobjective decision making: application of fuzzy linguistic preferences and goal programming. *Fuzzy Sets and Systems*, 127:209-220, 2002.
- [45] S. Rios, C. Bielza, and A. Mateos. *Fundamentos de los Sistemas de Ayuda a la Decisión*. Ra-Ma, 2002.

- [46] M. Roubens. Some properties of choice functions based on valued binary relations. *European Journal of Operational Research*, 40:309-321, 1989.
- [47] M. Roubens. Fuzzy sets and decision analysis. *Fuzzy Sets and Systems*, 90: 199-206, 1997.
- [48] S. Saint and J. R. Lawson. *Rules for Reaching Consensus. A Modern Approach to Decision Making*. Jossey-Bass, 1994.
- [49] J. F. L. Téo and B. Mareschal. An interval version of PROMETHEE for the comparison of building products' design with ill-defined data on environmental quality. *European Journal of Operational Research*, 109:522-529, 1998.
- [50] M. Tong and P.P Bonissone. A linguistic approach to decision making with fuzzy sets. *IEEE Transactions on Systems, Man and Cybernetics*, 10:716-723, 1980.
- [51] M. Tong and P.P Bonissone. Linguistic solution to fuzzy decision problems. *Studies in the Management Science*, 20:323-334, 1984.
- [52] Z.S. Xu. Uncertain linguistic aggregation operators based approach to multiple attribute group decision making under uncertain linguistic environment. *Information Sciences*, 168:171-184, 2004.
- [53] R.R. Yager. On ordered weighted averaging aggregation operators in multicriteria decision making. *IEEE Transactions on Systems, Man and Cybernetics*, 18:183-190, 1988.
- [54] R.R. Yager. Aggregation operators and fuzzy system modelling. *Fuzzy Sets and Systems*, 67:129-145, 1993.
- [55] R.R. Yager. On weighted median aggregation. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 2:101-113, 1994.
- [56] R.R. Yager. An approach to ordinal decision making. *International Journal of Approximate Reasoning*, 12:237-261, 1995.
- [57] R.R. Yager. Fusion of ordinal information using weighted median aggregation. *International Journal of Approximate Reasoning*, 18:32-35, 1998.
- [58] L. A. Zadeh. The concept of a linguistic variable and its applications to approximate reasoning. *Information Sciences*, Part I, II, III, 8,8,9:199-249,301-357,43-80, 1975..
- [59] L. A. Zadeh. Fuzzy logic = computing with words. *IEEE Transactions on Fuzzy Systems*. 4(2):103-111, 1996.
- [60] S. Zadrozny. *Consensus under Fuzziness*. An approach to the consensus reaching support in fuzzy environment, 83-109, Kluwer Academic Publishers, 1997.
- [61] Q. Zhang, J.C.H. Chen and P.P. Chong. Decision consolidation. Criteria weight determination using multiple preference formats. *Decision Support Systems*, 38(2):247-258, 2004.

ANEXO I.

MANUAL DE INSTALACIÓN

DEL SERVIDOR

Este anexo está dedicado a realizar las configuraciones necesarias para poner en marcha nuestra aplicación.

Las únicas consideraciones previas que han de tenerse en cuenta son que durante todo este manual se ha supuesto que la unidad principal de disco duro es C: y que la unidad principal de disco óptico es D:.

Hemos de decir que nuestro sistema trabajará con un servidor basado en la plataforma Microsoft Windows, por lo que todas estas instalaciones y configuraciones hay que realizarlas teniendo en cuenta dicho Sistema Operativo.

Todo el material necesario para instalar y dejar operativo el servidor se encuentra disponible en el DVD que acompaña a esta memoria. Sitúese en el dispositivo de DVD D:\TomaDecision y compruebe que se encuentran los siguientes archivos:

- ✓ jdk-6u6-windows-i586-p.exe
- ✓ apache-tomcat-6.0.16.exe
- ✓ mysql-5.0.51a-win32.zip
- ✓ mysql-gui-tools-5.0-r12-win32.msi
- ✓ ProcesoConsenso.war
- ✓ Fichero *.sql*

Si es así podemos proceder a la instalación de nuestro servidor inmediatamente. Si falta algún archivo o alguno de ellos se encuentra dañado póngase en contacto con el responsable de la aplicación para subsanar el problema.

Paso 1: Instalar la base de datos *MySQL*.

El primer paso consiste en instalar tanto el Administrador como el cliente de la Base de Datos, pero antes, instalaremos el servidor de la Base de Datos MySQL. Para ello descomprimos el fichero *mysql-5.0.51a-win32.zip* y ejecutamos el fichero *Setup.exe*. En ese momento el asistente nos mostrará la siguiente ventana, Figura I.1.



Figura I.1: Ventana principal del asistente de instalación de MySql

Pulsamos *Next* y el asistente nos preguntará por el tipo de instalación que deseamos, Figura I.2. Seleccionamos la opción *Typical* y pulsamos *Next*.



Figura I.2: Ventana de selección del tipo de instalación

A continuación, el asistente nos informará del tipo seleccionado y del directorio donde se va a instalar el servidor MySQL, Figura I.3. Pulsamos *Install* para comenzar la instalación.



Figura I.3: Ventana de iniciación de instalación del servidor MySQL

Una vez concluida la instalación, el asistente mostrará una ventana indicando que la instalación ha concluido con éxito, Figura I.4. Pulse *Finish* para terminar.



Figura I.4: Ventana de finalización de instalación

Una vez instalado el servidor de Base de Datos MySQL, hay que configurarlo. En la siguiente ventana, Figura I.5, pulsamos *Next*, en ese momento se iniciará el asistente de configuración del servidor MySQL.



Figura I.5: Ventana principal del asistente de configuración

A continuación seleccionamos el tipo de configuración *Detailed Configuration* y pulsemos *Next*, Figura I.6.

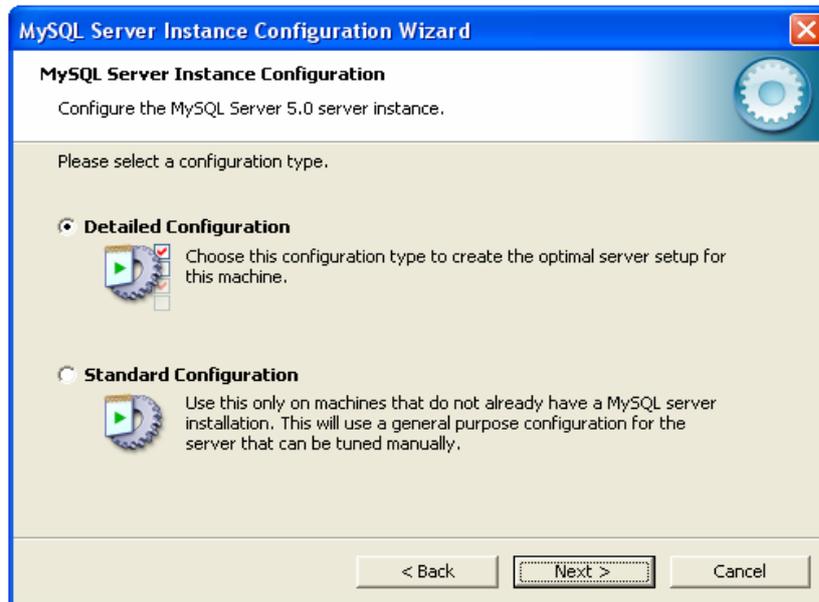


Figura I.6: Ventana de selección del tipo de configuración

En la siguiente ventana, Figura I.7, seleccionamos el tipo de servidor *Server Machine* y pulsamos *Next*.

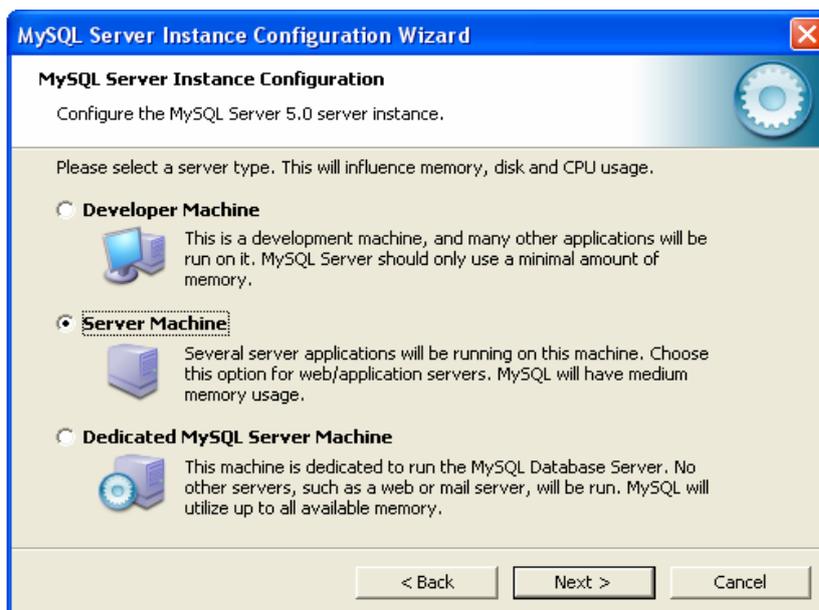


Figura I.7: Ventana de selección del tipo de servidor MySQL

El asistente nos preguntará por el tipo de uso de la base de datos, seleccionamos *Multifunctional Database*, para que nos permita tener accesos concurrentes a la Base de Datos, Figura I.8, y pulsemos *Next*.



Figura I.8: Ventana de selección del tipo de uso de la Base de Datos

En ese momento el asistente nos mostrará una ventana, Figura I.9, donde nos pide el directorio donde van a residir las tablas en el servidor. Dejamos la que viene por defecto y pulsamos *Next*.

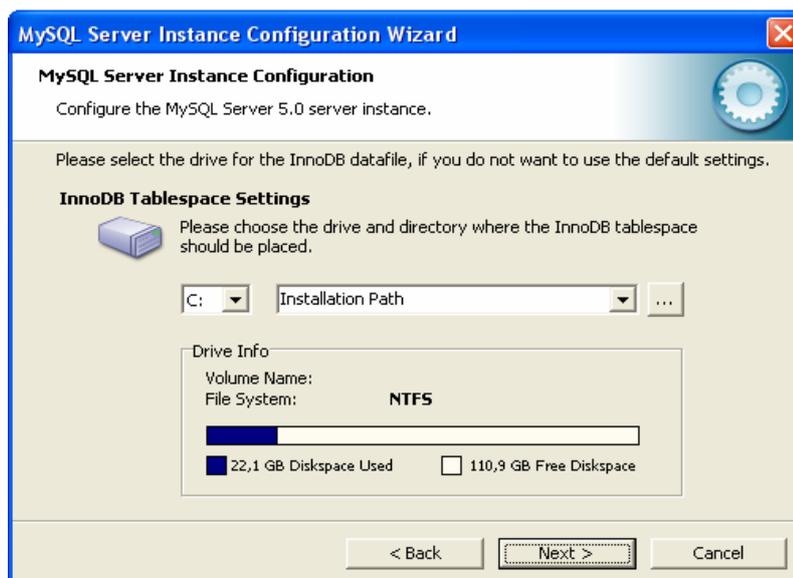


Figura I.9: Ventana de selección del directorio de la Base de Datos

En la siguiente ventana, Figura I.10, seleccionamos la primera o tercera opción y pulsamos *Next*.

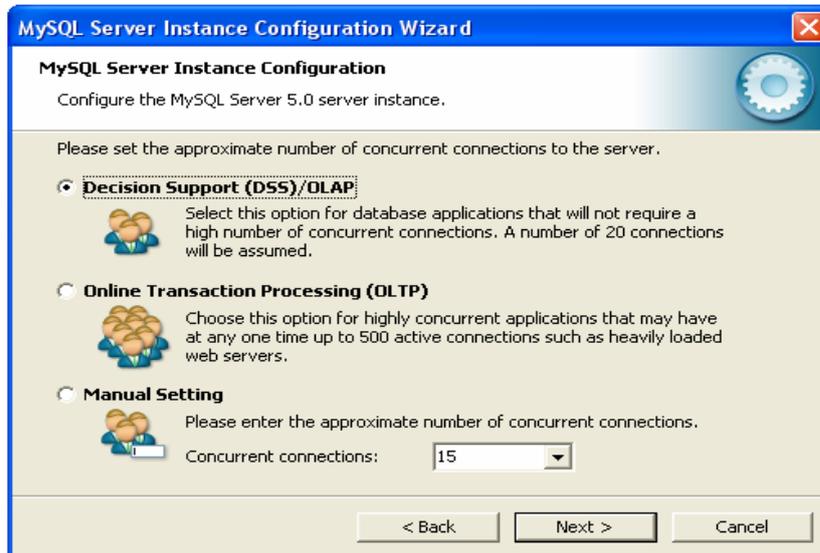


Figura I.10: Ventana de selección del número de conexiones simultáneas

A continuación, debemos indicar el puerto que vamos a usar para conectarnos con el servidor de la Base de Datos MySQL, Figura I.11. Es recomendable que dejemos el que viene por defecto 3306 y pulsamos *Next*.

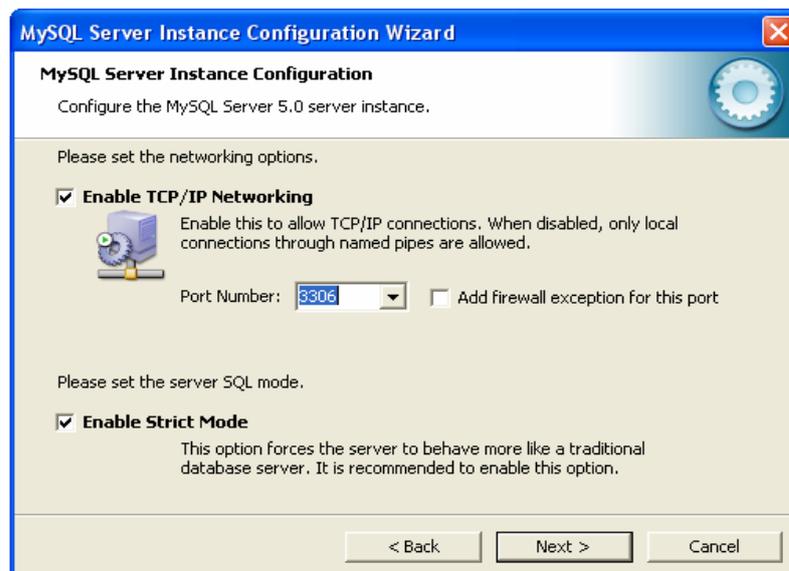


Figura I.11: Ventana de selección del puerto de conexión

Seleccionamos el conjunto de caracteres para representar la información de la Base de Datos. Dejamos la que viene por defecto, Figura I.12, y pulsamos *Next*.

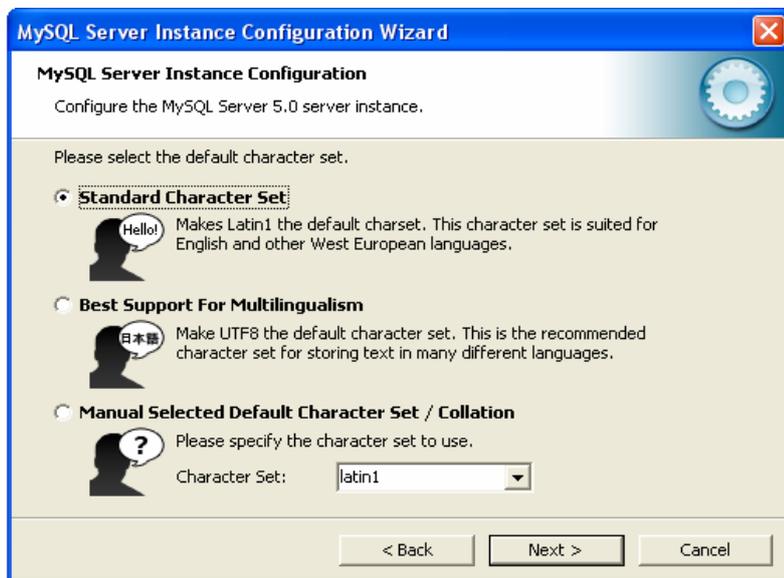


Figura I.12: Ventana de selección del conjunto de caracteres

En la siguiente ventana, Figura I.13, el asistente nos preguntará por el tipo de servicio. Podemos seleccionar las dos opciones o sólo la primera y pulsamos *Next*.



Figura I.13: Ventana de selección del tipo de servicio

Después tenemos que introducir la clave del usuario root del servidor como se muestra en la siguiente ventana, Figura I.14 y pulsemos *Next*.



Figura I.14: Ventana para introducir la contraseña

Finalmente pulsemos *Execute*, Figura I.15.

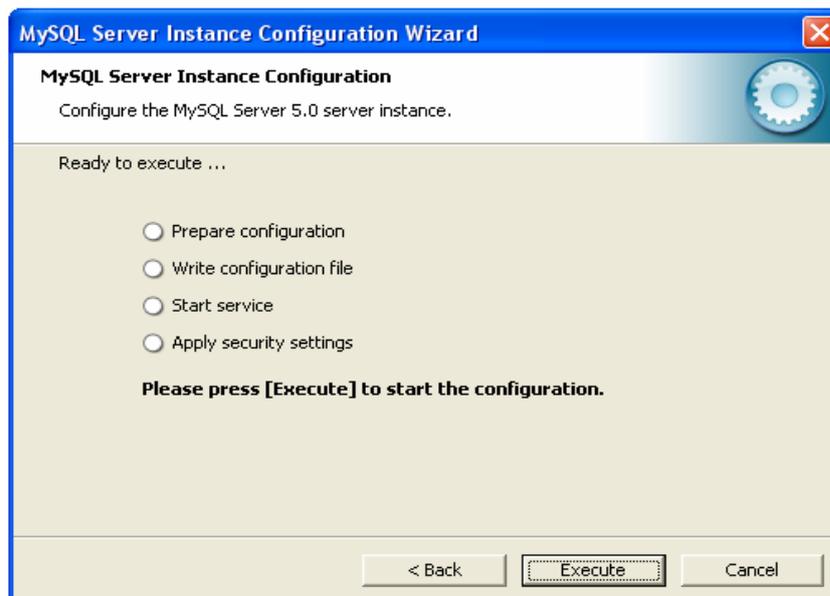


Figura I.15: Ventana de preparación para ejecutar la configuración

El asistente mostrará una ventana indicando que la configuración se ha realizado con éxito, Figura I.16. Pulsamos *Finish* para terminar.

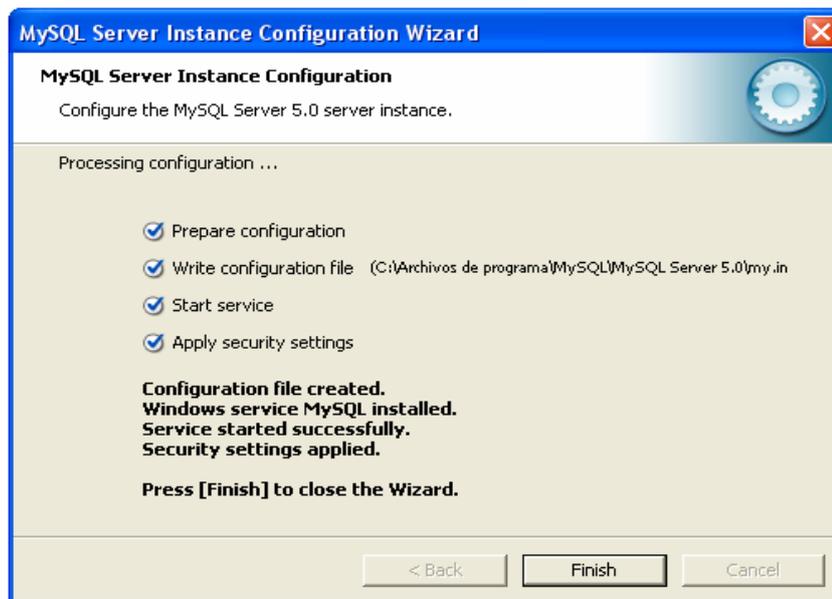


Figura I.16: Ventana de finalización de la configuración del servidor MySQL

Paso 2: Instalación del *administrador y cliente de MySQL.*

Utilizaremos un paquete de herramientas de MySQL que contiene los siguientes componentes:

- MySQL Administrator.
- MySQL Query-Browser.
- MySQL Migration Toolkit.

En nuestro caso, únicamente instalaremos los dos primeros.

Para comenzar con la instalación ejecutamos el fichero *mysql-gui-tools-5.0-r12-win32.msi* y nos aparecerá la siguiente ventana, Figura I.17.



Figura I.17: Ventana principal de instalación de MySQL Tools

Pulsamos *Next* y el asistente nos mostrará una ventana de conformidad con la licencia de software, Figura I.18. Seleccionamos *I accept the terms in the license agreement* y pulsamos *Next*.

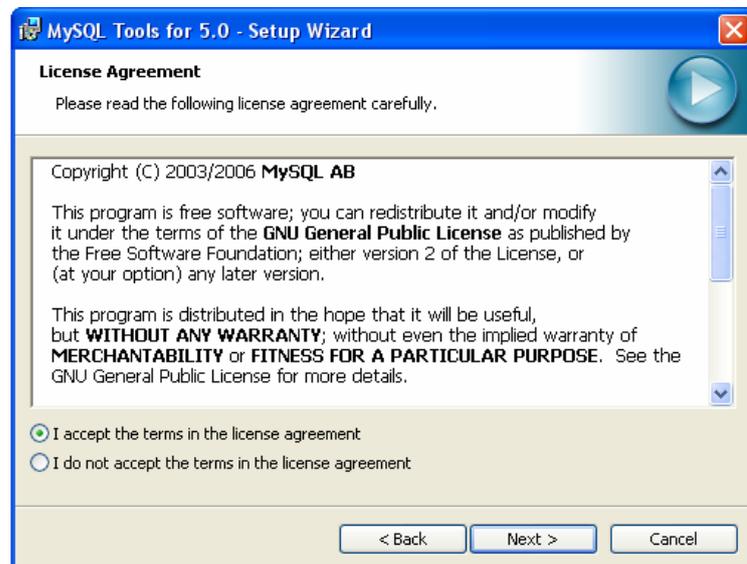


Figura I.18: Ventana de licencia MySQL Tools

A continuación el asistente mostrará una ventana, Figura I.19, indicando el directorio donde se va a instalar MySQL Tools.

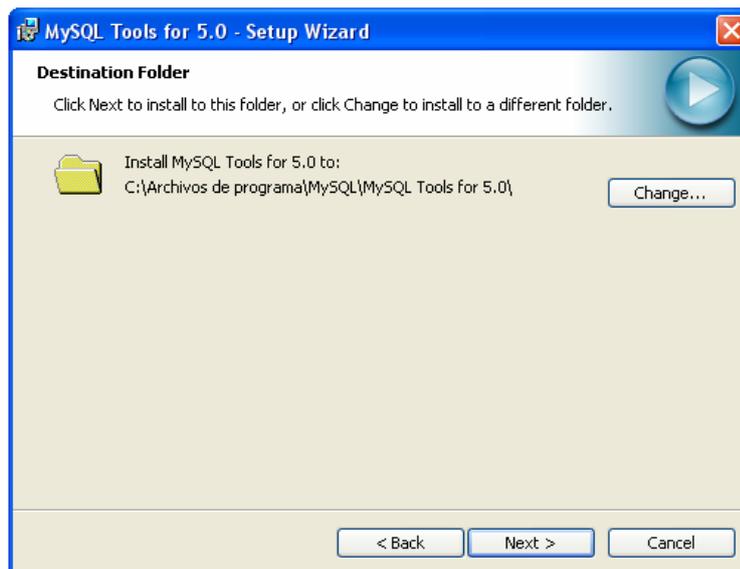


Figura I.19: Ventana de selección del directorio de instalación

Pulsamos *Next* y nos aparecerá una ventana, Figura I.20, donde debemos indicar el tipo de instalación. Seleccionamos la opción *Custom*.

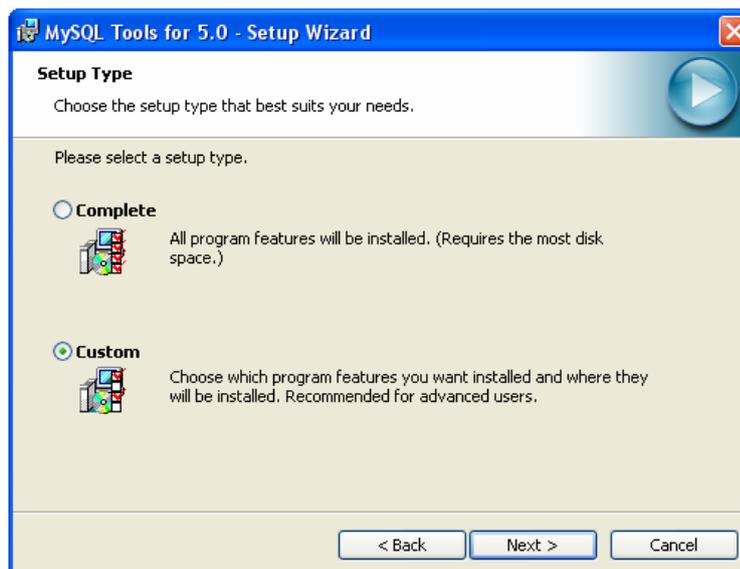


Figura I.20: Ventana de selección del tipo de instalación de MySQL Tools

Pulsemos *Next* y el asistente nos mostrará una ventana, Figura I.21, con todos los componentes que se instalarán.

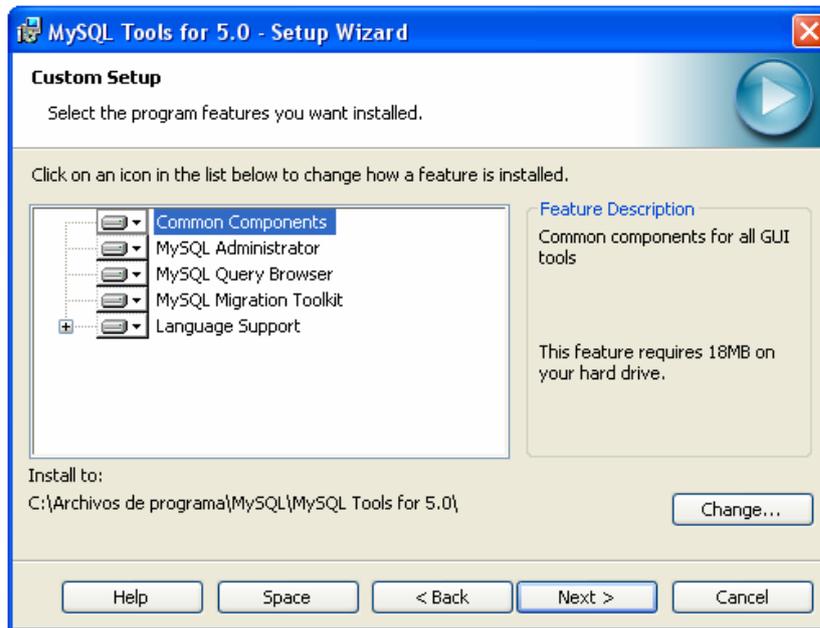


Figura I.21: Ventana con los componentes disponibles

Hacemos click en *MySQL Migration Toolkit* y seleccionamos *This feature will not be available* Figura I.22, para que no se instale este componente. A continuación pulsamos *Next*.



Figura I.22: Ventana para no instalar MySQL Migration Toolkit

El asistente nos mostrará una ventana para comenzar la instalación, Figura I.23. Pulsamos *Install* y la instalación dará comienzo.

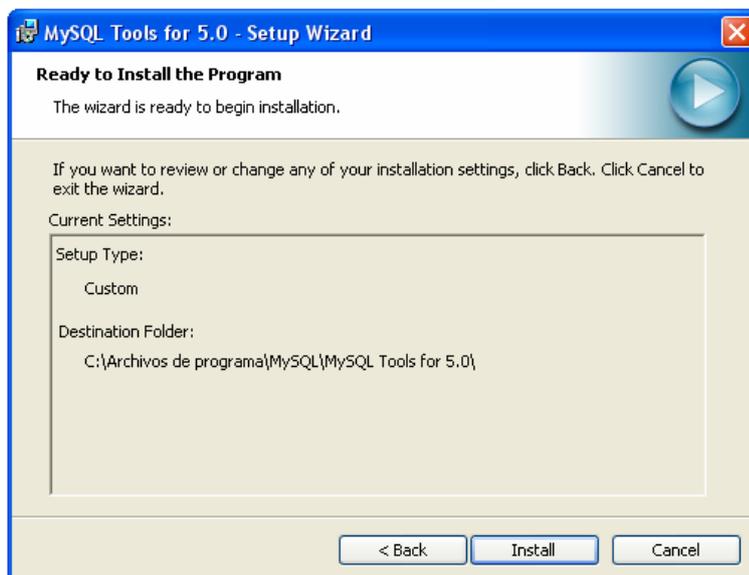


Figura I.23: Ventana para comenzar la instalación

Una vez finalizada la instalación el asistente nos mostrará la siguiente ventana, Figura I.24. Pulsamos *Finish* y la instalación terminará.

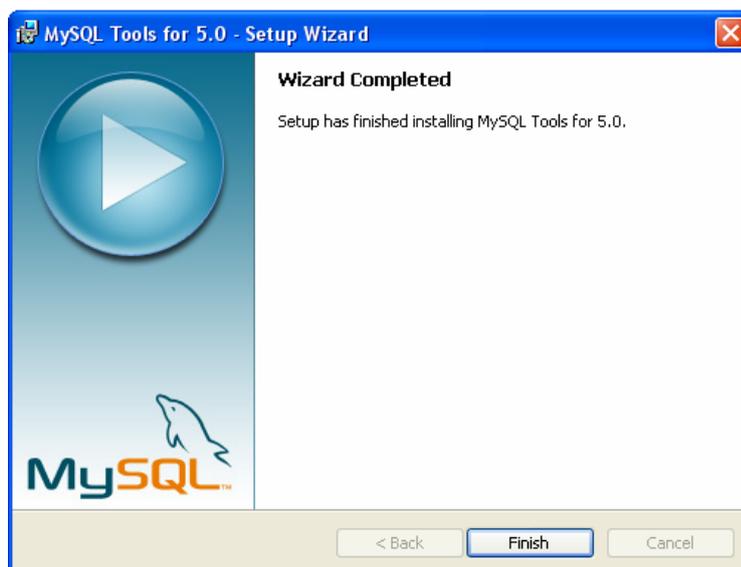


Figura I.24: Ventana de finalización de la instalación

Paso 3: Crear la Base de Datos de la aplicación.

Una vez que tenemos instalados los programas que forman la Base de Datos MySQL, el siguiente paso consiste en crear las tablas que formarán la base de datos de la aplicación ProcesoConsenso.

Para ello tenemos dos opciones:

- Crear las tablas manualmente, utilizando MySQL Administrator o MySQL Query-Browser.
- Utilizar el fichero *.sql*

Si decidimos utilizar el fichero *.sql*, tenemos que seguir los siguientes pasos:

Accedemos a MySQL Administrator y pulsamos la opción *Restore*, como podemos ver en la Figura I.25.

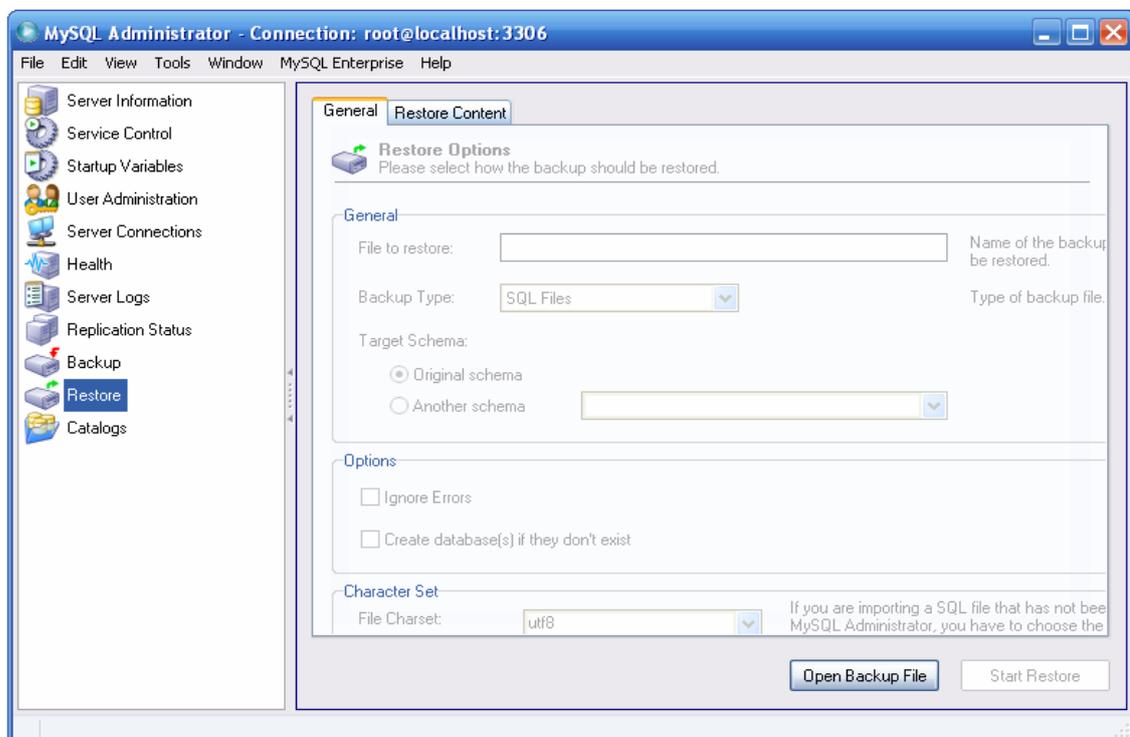


Figura I.25: MySQL Administrator

A continuación, pulsamos *Open Backup File*, cargamos el archivo .sql y pulsamos *Abrir*. Chequeamos la opción *Create DataBase(s) if they don't exist*. Pulsamos *Start Restore* y seguidamente se mostrará un mensaje de éxito.

Si ya disponemos de un servidor Apache Tomcat 6.0 sólo tendremos que copiar el archivo *ProcesoConsenso.war* en el directorio *webapp*, que se encuentra donde tengamos instalado el servidor. Una vez hecho esto podremos acceder a la aplicación tecleando en el navegador <http://www.ip-del-servidor/ProcesoConsenso/index.jsp>. En caso contrario, sigamos los pasos 4 y 5.

Paso 4: Instalar jdk

Para instalar la JVM (Java Virtual Machine) ejecutamos el archivo *jdk-6u6-windows-i586-p.exe*. Si ya tenemos instalada una JVM nos aparecerá una ventana (Figura I.26) informándonos que ya tenemos instalada una versión igual, por lo que no es necesario reinstalarla.

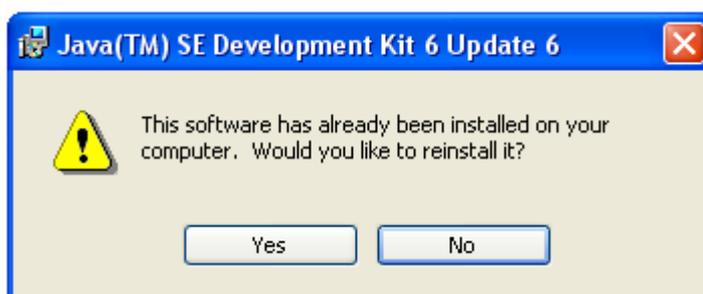


Figura I.26: Jdk instalado

Paso 5: Instalar Apache-Tomcat

Al tratarse nuestro proyecto de una aplicación Web, es lógico necesitar un servidor Web para que proporcione los servicios o responda a las peticiones realizadas por cualquier cliente que se quiera conectar al sistema. Para ello utilizaremos el servidor Web Apache Tomcat.

Para instalar el servidor Apache, ejecutamos el fichero de instalación *apache-tomcat-6.0.16.exe* que acompaña al DVD de la memoria. Éste nos mostrará la siguiente ventana, Figura I.27:

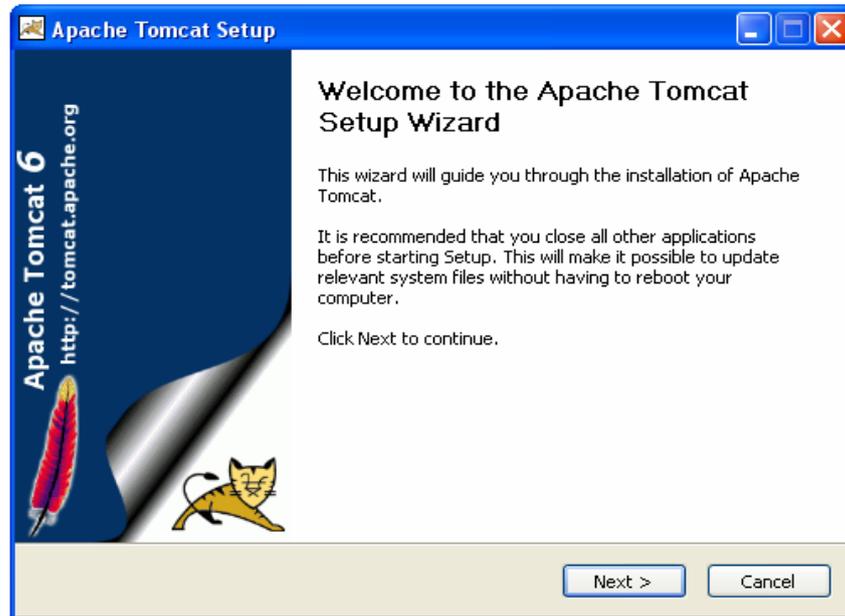


Figura I.27: Ventana principal de instalación de Apache-Tomcat

Pulsamos *Next*, para comenzar la instalación de Apache Tomcat. Acto seguido, éste nos mostrará la ventana de conformidad con la licencia de Apache Software Foundation, Figura I.28.

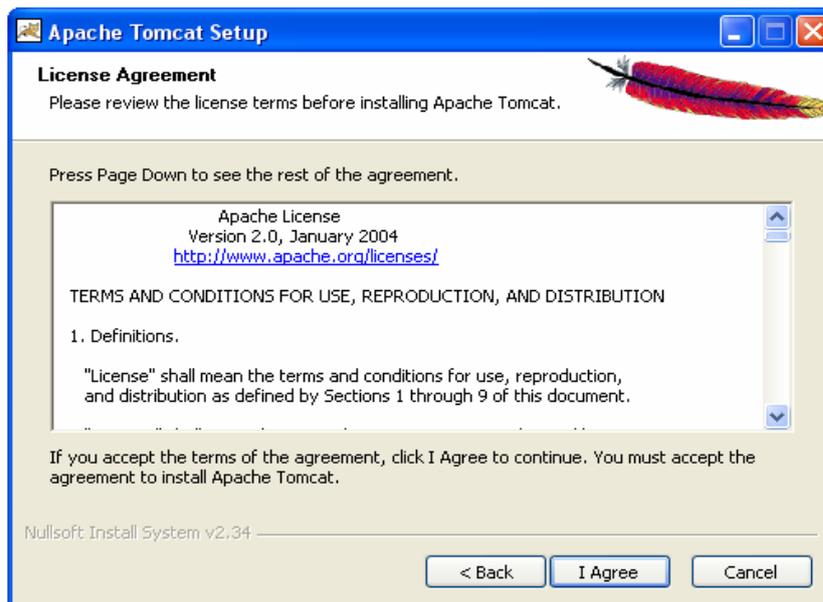


Figura I.28: Ventana de licencia de Apache Tomcat

Pulsamos *I Agree* y el asistente nos pedirá el tipo de instalación que deseamos, Figura I.29. Seleccionamos la normal y pulsamos *Next*.

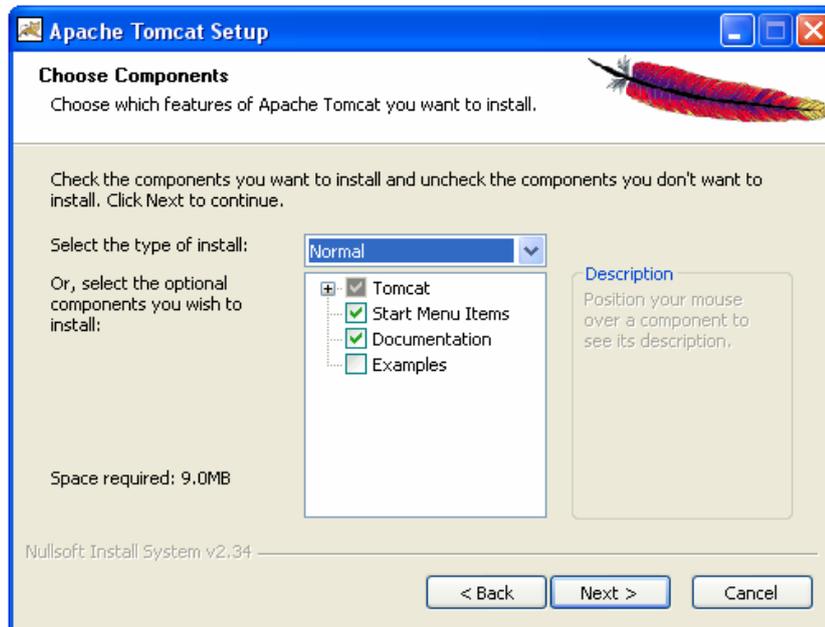


Figura I.29: Ventana de selección del tipo de instalación

Después el asistente nos indicará el directorio donde vamos a instalar el servidor, Figura I.30.

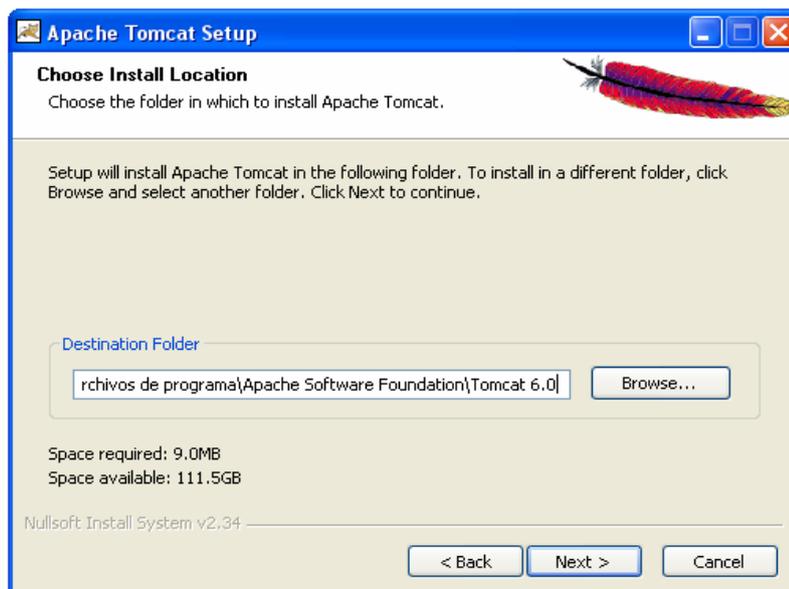


Figura I.30: Ventana de selección del directorio de instalación

Pulsamos *Next* y el asistente nos mostrará la siguiente ventana de configuración, Figura I.31. En ella nos pedirá el puerto para la conexión http, el nombre de usuario y la contraseña del administrador.

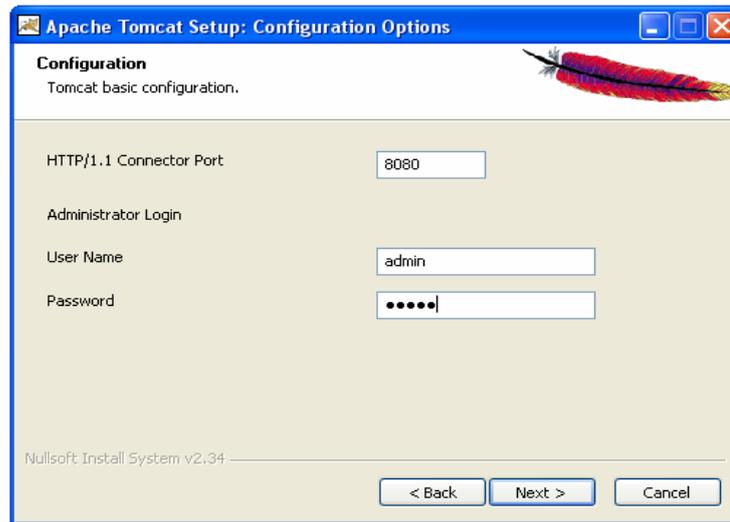


Figura I.31: Ventana de configuración

Introducimos el puerto deseado, por defecto 8080, el nombre de usuario y la contraseña del administrador. A continuación pulsamos *Next* y nos aparecerá una ventana, Figura I.32, que nos pedirá que seleccionemos el directorio donde tengamos instalada la *jre* (instalada en el paso anterior).

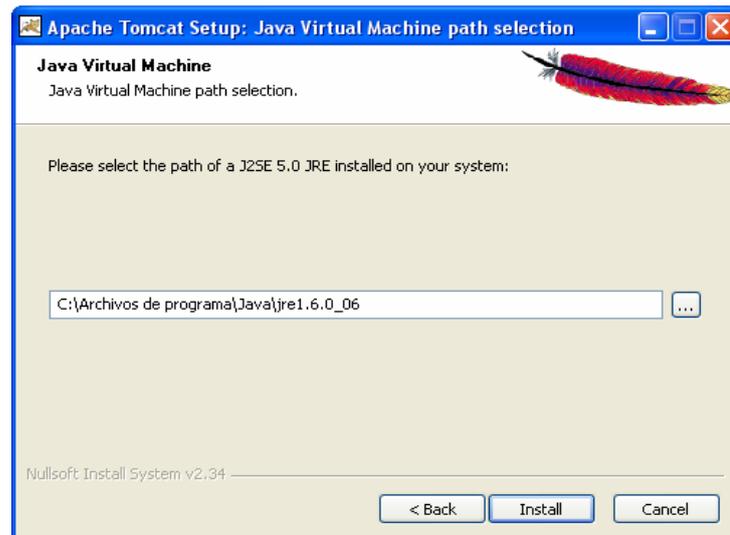


Figura I.32: Ventana de selección del directorio de *jre*

Pulsamos *Install*, y la instalación dará comienzo. Finalizada la instalación, el asistente mostrará la siguiente ventana, Figura I.33.

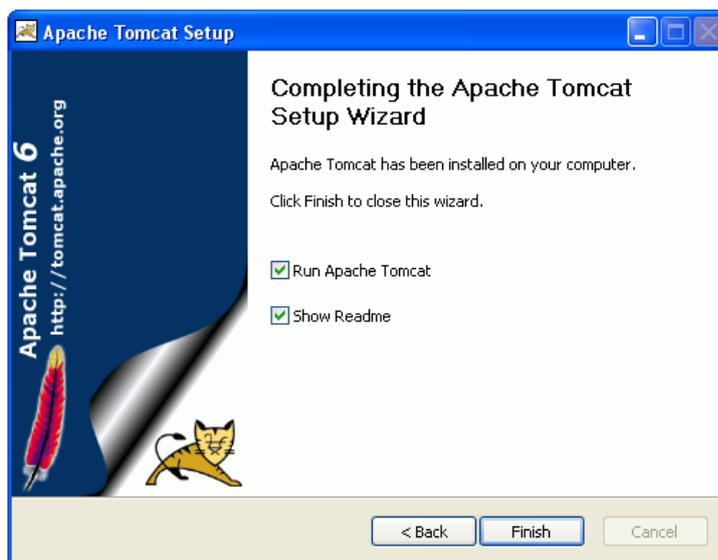


Figura I.33: Ventana de finalización de la instalación

Pulsamos *Finish* y habremos terminado con la instalación de Tomcat.

Una vez finalizada la instalación debemos poner en marcha el servidor Apache Tomcat. Para ello debemos establecer la variable de entorno `CATALINA_HOME` al directorio donde tengamos instalado Tomcat 6.0.

En Mi PC, hacemos click en el botón derecho del ratón, seleccionamos propiedades, en la pestaña opciones avanzadas, pulsamos un botón llamado *variables de entorno* y creamos una nueva variable de entorno con nombre `CATALINA_HOME` y valor, el directorio donde tengamos instalado Tomcat 6.0.

Después de iniciar, las aplicaciones incluidas en Tomcat 6.0 deben ser visualizadas en <http://localhost:8080>.

BIBLIOGRAFÍA ANEXO I.

Mysql 5.0 Reference Manual, (en línea). Consultado 14 marzo 2008. Disponible <http://dev.mysql.com/doc/refman/5.0/es/index.html>.

The Apache Tomcat 5.5 Servlet/JSP Container, (en línea). Consultado 4 abril 2008. Disponible <http://tomcat.apache.org/tomcat-5.5-doc/index.html>.

ANEXO II.

**MANUAL DE USUARIO
DEL ADMINISTRADOR**

Este manual de usuario está organizado como una visita guiada por la aplicación, cuando accedemos a la misma como Administrador.

El primer paso para utilizar la aplicación es abrir el Navegador (Internet Explorer) y teclear la dirección en la que tenemos alojada ésta, que en nuestro caso será: <http://localhost:8084/ProcesoConsenso/index.jsp>.

Al introducir dicha dirección, nos encontramos con la página de inicio de la aplicación, tal y como se ve en la Figura II.1.

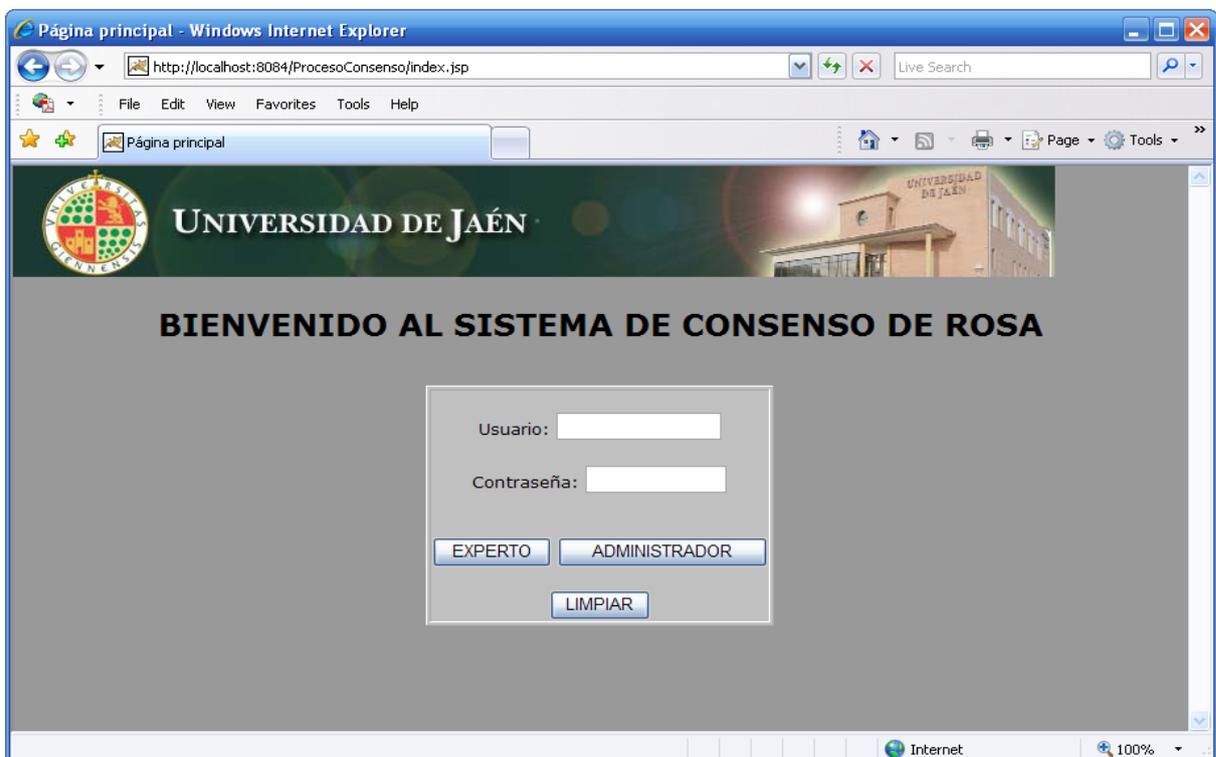


Figura II.1: Entrada al sistema

Para entrar como Administrador debemos introducir el nombre de usuario y contraseña del administrador. Si alguno de estos datos (o los dos) no son correctos, el sistema nos muestra un mensaje de error como se observa en la Figura II.2.

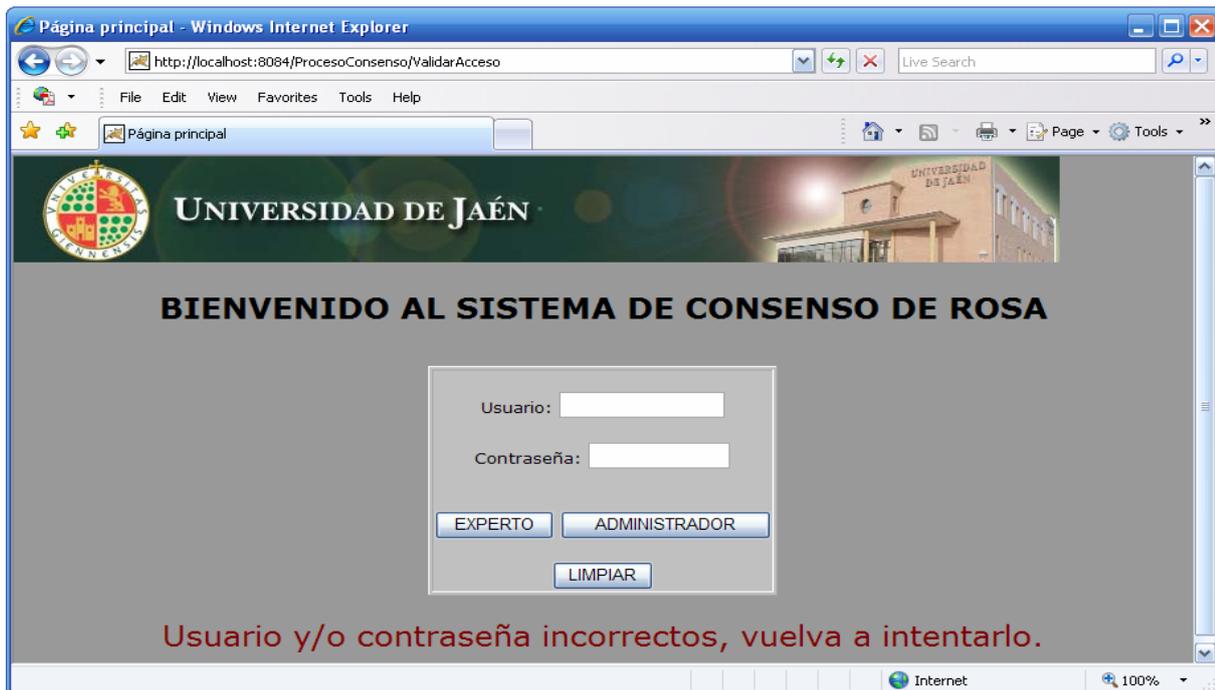


Figura II.2: Mensaje de error en la autenticación

En el caso de una autenticación correcta nos aparece el menú principal del administrador, Figura II.3.



Figura II.3: Menú principal del administrador

En este menú principal podemos encontrar las 3 opciones principales que tiene el Administrador.

A continuación veremos cada una de ellas con más detalle.

1. Gestión de problemas.

Si hacemos click en el botón *Acceder* que aparece a la derecha de *Gestión de problemas*, entraremos en el submenú de *Gestión de Problemas*, Figura II.4.

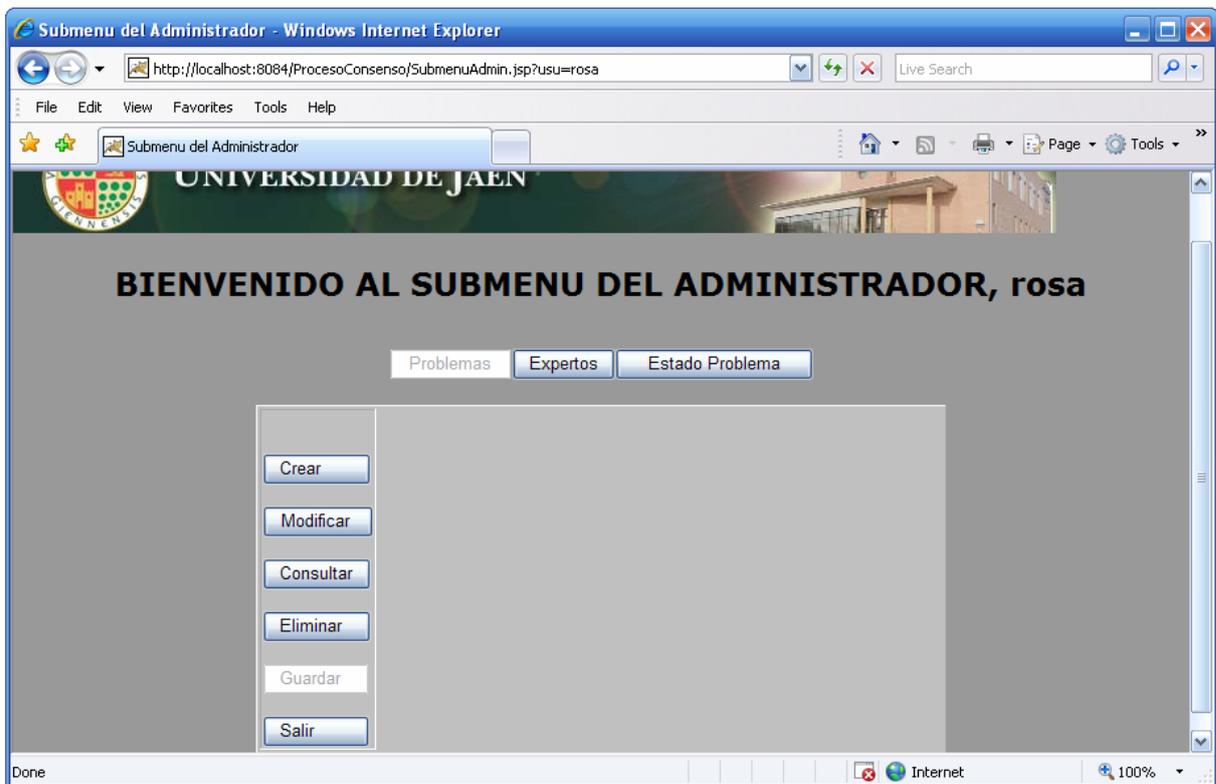


Figura II.4: Submenú para la Gestión de problemas

En este submenú aparecen todas las acciones que el Administrador puede realizar sobre *Problemas*. Destacar que en cualquier momento el Administrador puede cambiar de opción y dirigirse al menú *Expertos* o *Estado Problema*.

Veamos cada una de las posibles acciones de *Problemas*.

◆ **Crear**

Para crear un problema pulsamos el botón *Crear* y nos aparecerá un formulario como el que se muestra en la Figura II.5. Introducimos el nº de alternativas que vamos a definir en el problema y el nº de expertos que deseamos asignar. A continuación pulsamos *Continuar*.



Figura II.5: Establecer nº de alternativas y expertos

Después, debemos rellenar todos los campos que aparecen en el formulario de la Figura II.6. Una vez rellenados pulsamos *Guardar*.

The screenshot shows a web browser window titled "Crear Problema - Windows Internet Explorer". The address bar shows the URL "http://localhost:8084/ProcesoConsenso/CrearProblema.jsp?usu=rosa". The page header includes the University of Jaén logo and the text "UNIVERSIDAD DE JAÉN". Below the header, a welcome message reads "BIENVENIDO/A, rosa". There are three navigation tabs: "Problemas", "Expertos", and "Estado Problema". The main content area is a form for creating a problem. It has a sidebar on the left with buttons: "Crear", "Modificar", "Consultar", "Eliminar", "Guardar", and "Salir". The form fields are: "Id del problema:" (text input), "Descripción:" (text area), "Parámetros:" (two sub-fields: "Umbral consenso:" and "N°max de rondas:"), and "Alternativas:" (four numbered text inputs). The browser status bar at the bottom shows "Done" and "Internet" with a 100% zoom level.

Figura II.6: Creación de un problema

Si al rellenar el formulario dejamos algún campo vacío, el sistema mostrará un mensaje de error indicando el campo concreto que se ha quedado en blanco. El resto de los campos mantendrán los valores introducidos.

En la Figura II.7, podemos ver un ejemplo si dejamos en blanco el campo descripción.

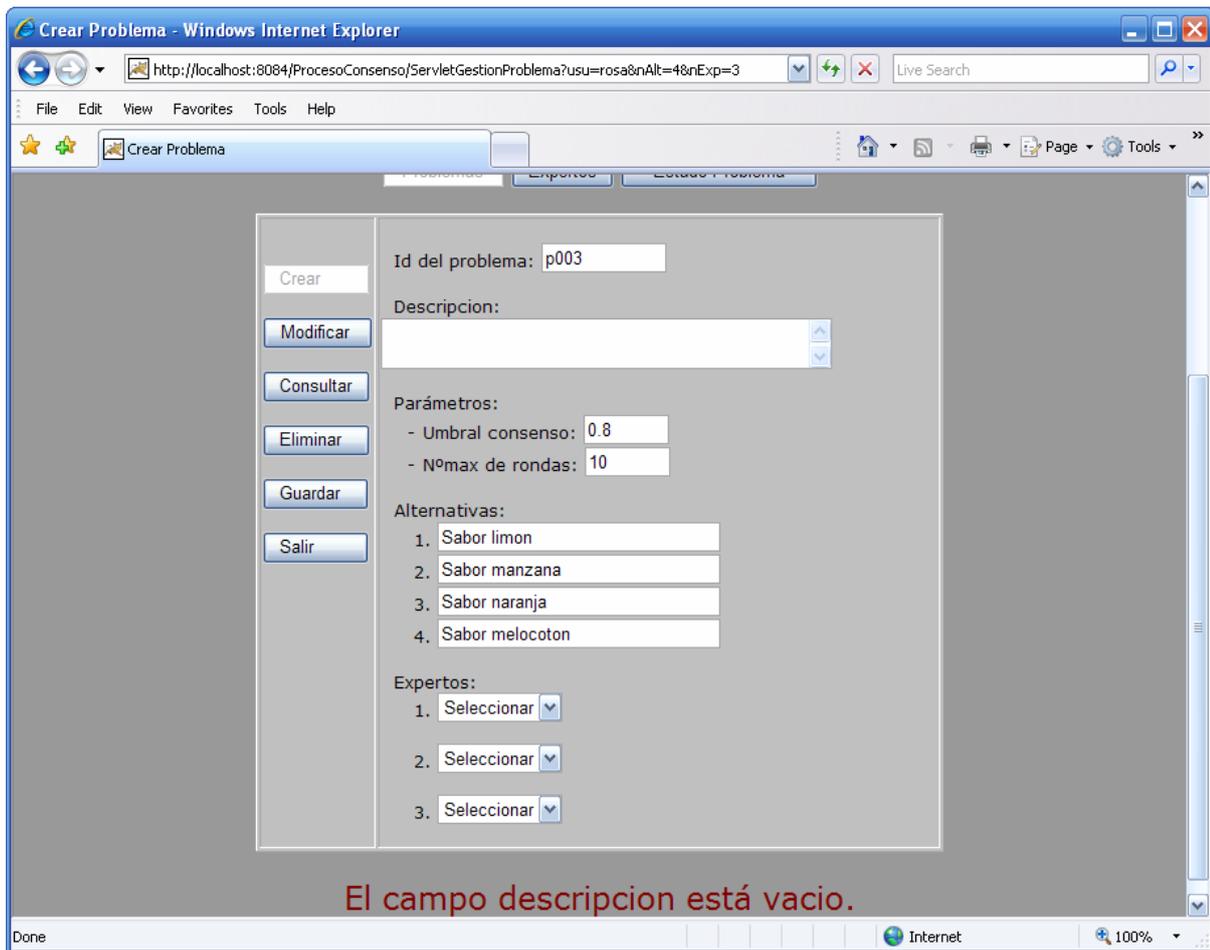


Figura II.7: Mensaje de error 'El campo descripción está vacío'

Si introducimos un identificador que ya existe para otro problema, el sistema nos mostrará un mensaje de error como el de la Figura II.8.

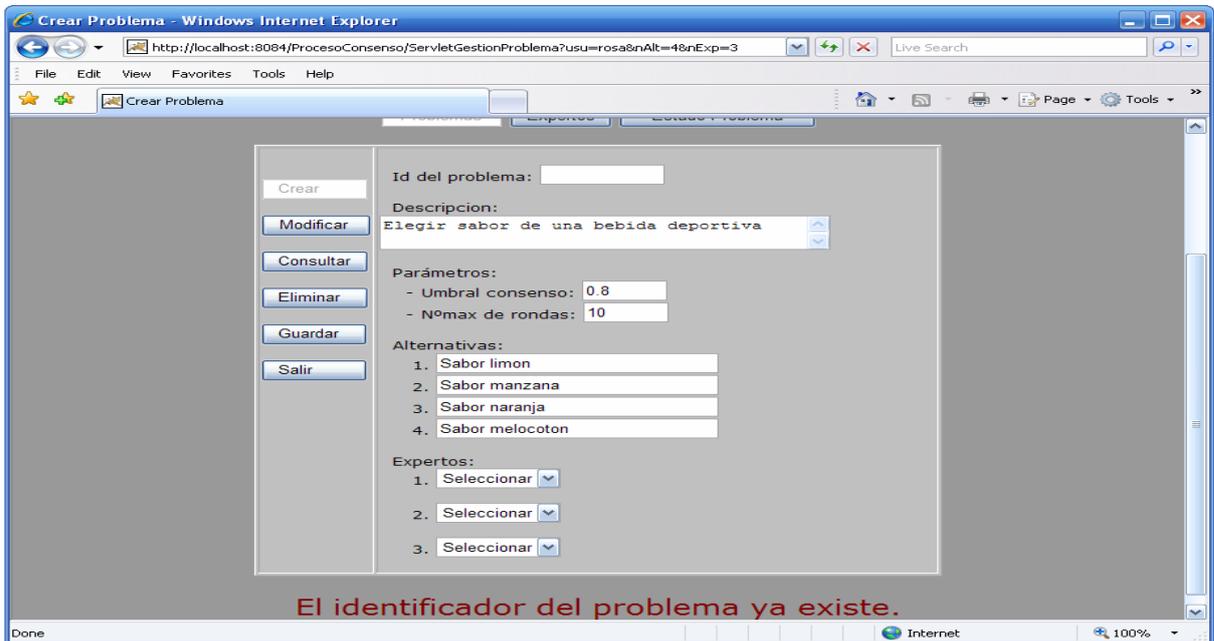


Figura II.8: Mensaje de error 'El identificador del problema ya existe'

Si el valor introducido en el campo umbral de consenso no es un número real entre 0 y 1, el sistema nos mostrará el mensaje de la Figura II.9.

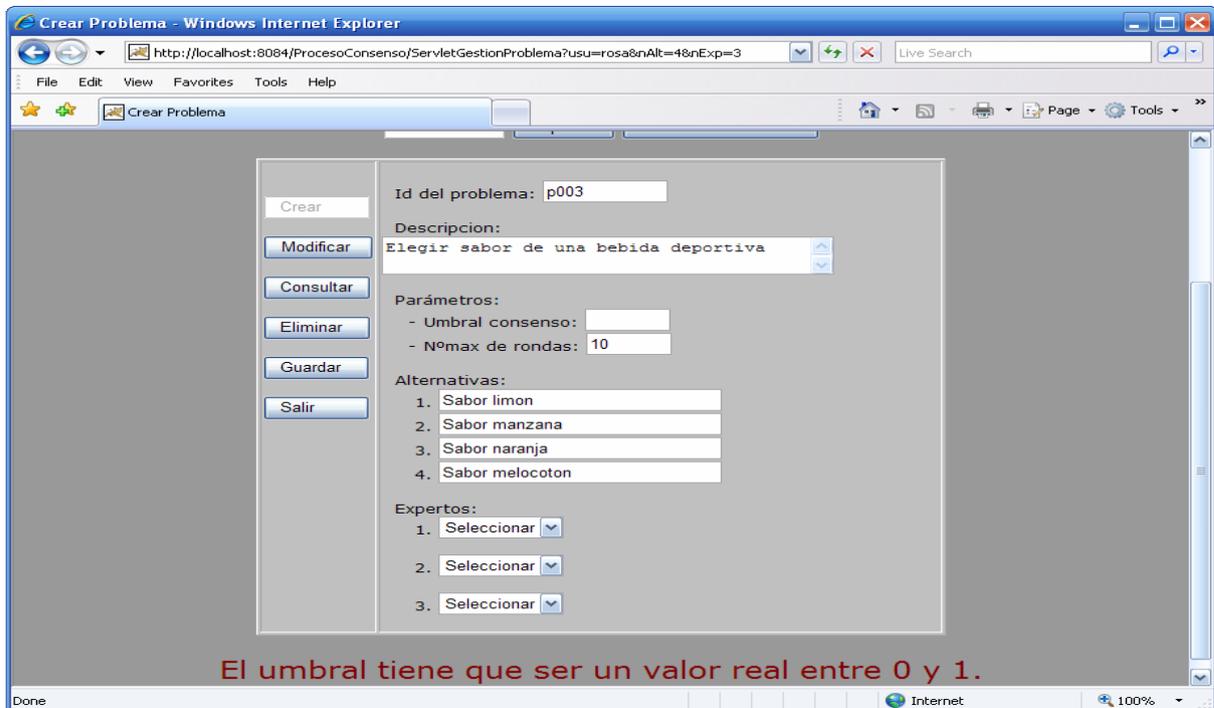


Figura II.9: Mensaje de error 'El umbral tiene que ser un valor real entre 0 y 1'

Si el valor introducido en el campo nº max de rondas no es un número entero mayor que 1, el sistema mostrará el error de la Figura II.10.

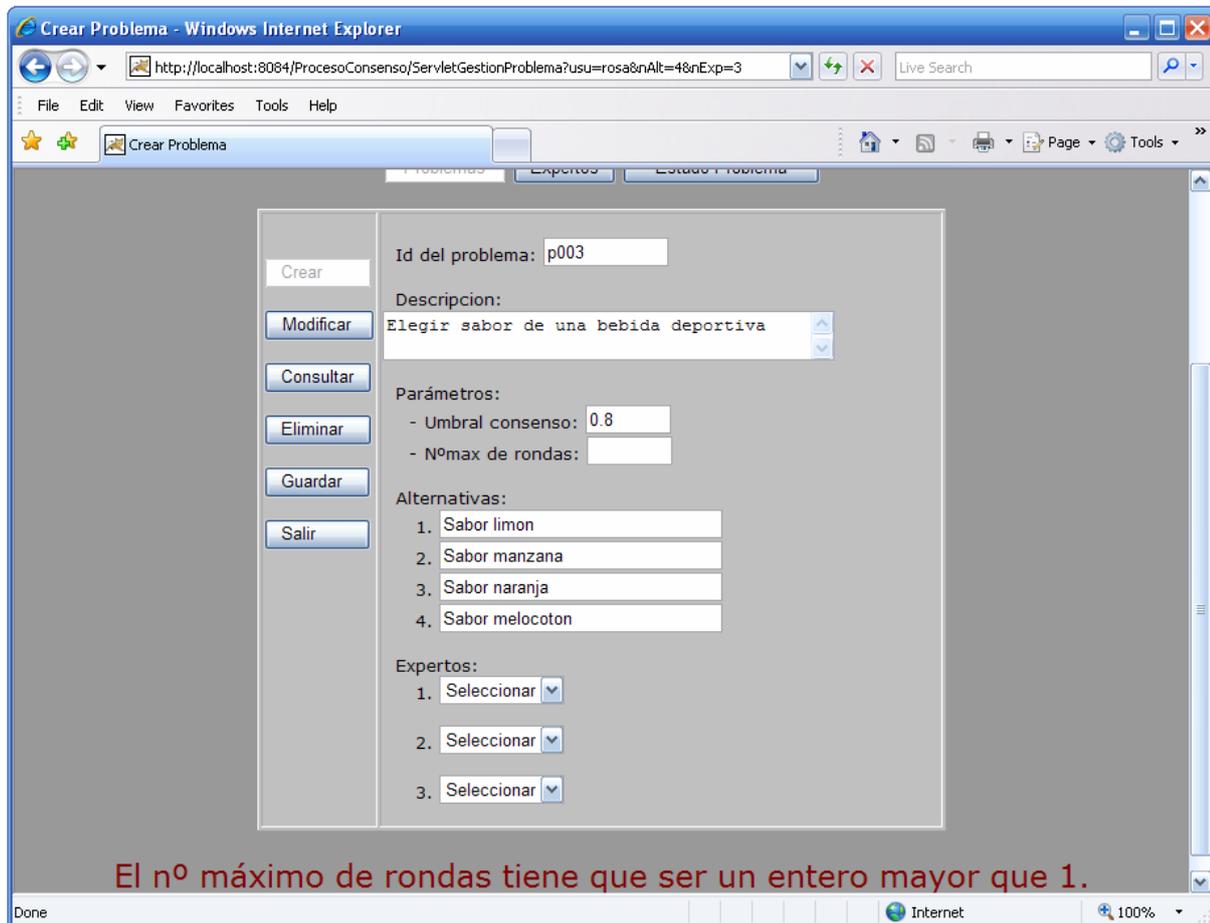


Figura II.10: Mensaje de error 'El nº máximo de rondas tiene que ser un entero mayor que 1'

Si no seleccionamos un experto en cada una de las listas desplegables, el sistema nos mostrará el mensaje de error como el de la Figura II.11.

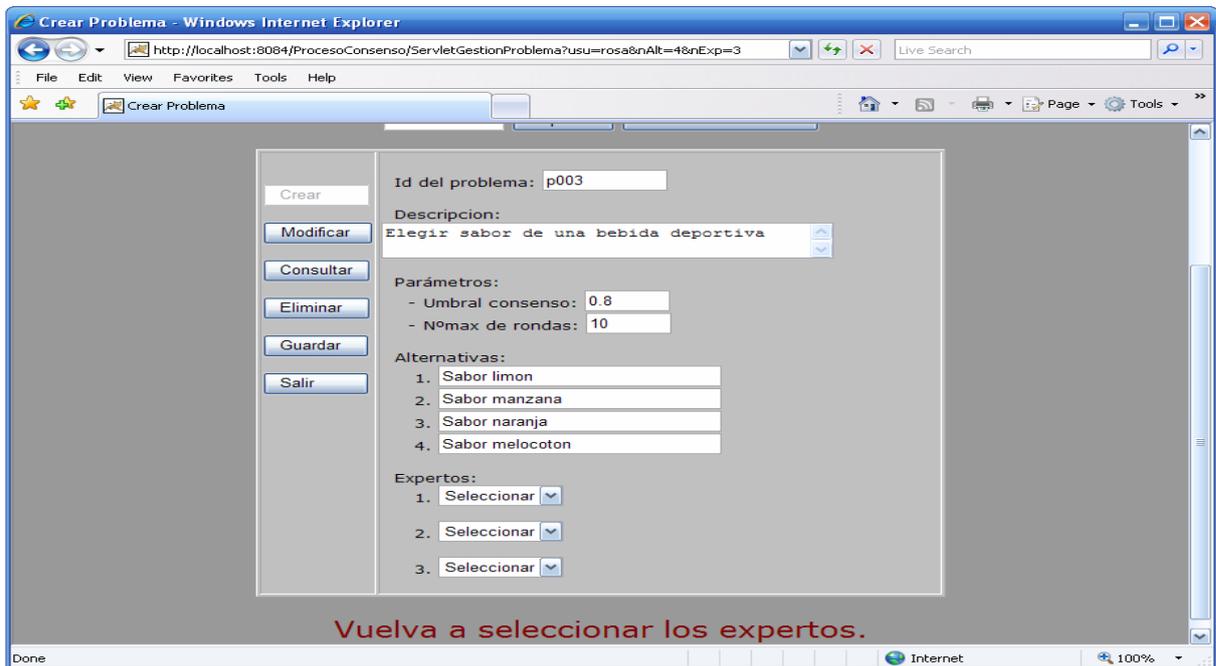


Figura II.11: Mensaje de error 'Vuelva a seleccionar los expertos'

Si pulsamos cualquier otro botón que no sea *Guardar*, el sistema nos mostrará un mensaje de confirmación, Figura II.12, para indicarnos que vamos a cambiar de opción sin guardar el problema.

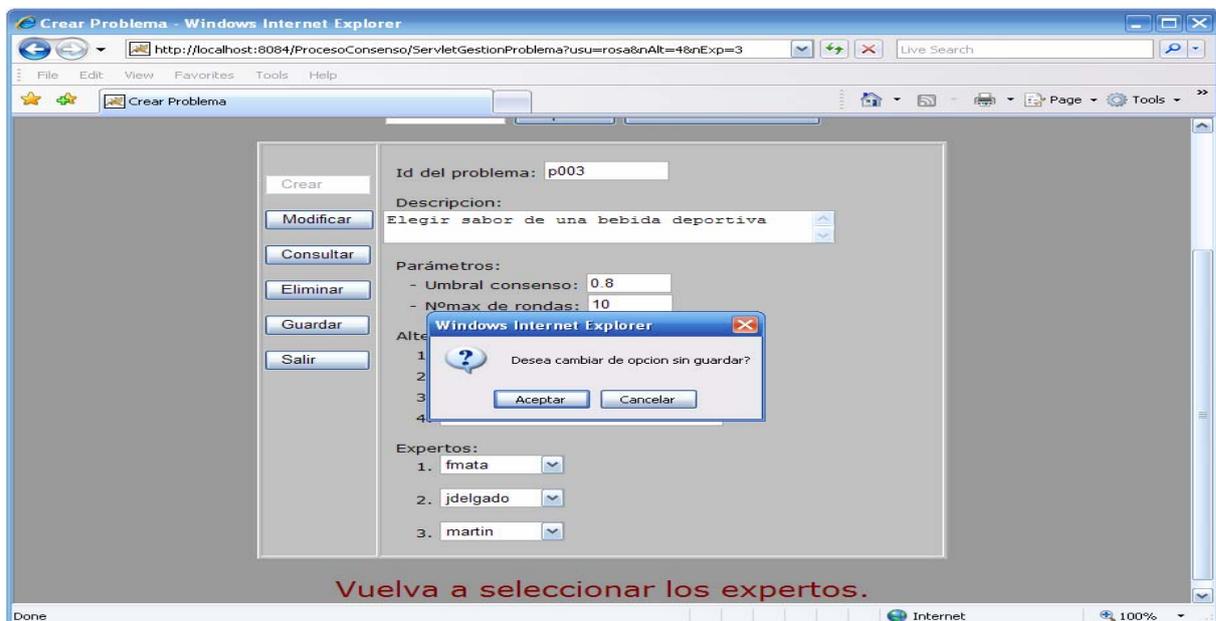


Figura II.12: Mensaje de confirmación al cambiar de opción en Crear problema

Si introducimos todos los datos correctamente y pulsamos *Guardar*, el sistema nos muestra un mensaje de éxito como podemos ver en la Figura II.13.

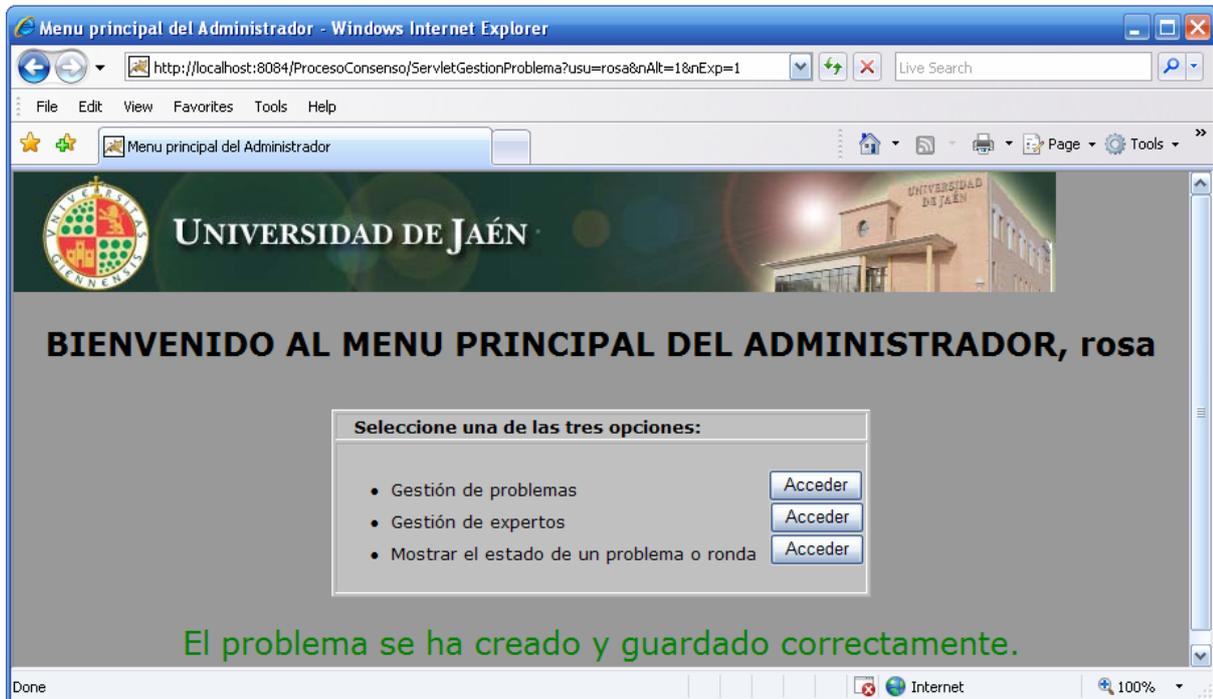


Figura II.13: Mensaje de éxito de creación de un problema

◆ **Modificar**

Una vez que nos encontramos en el submenú de *Gestión de problemas*, pulsamos la opción *Modificar* y nos aparece una lista desplegable en la que debemos seleccionar el problema que vamos a modificar, Figura II.14.



Figura II.14: Selección del problema

Una vez seleccionado el identificador del problema, pulsamos *Buscar* y el sistema nos mostrará un formulario con todos los datos del problema, Figura II.15.

Podemos cambiar cualquier campo de texto, excepto el identificador del problema. Si deseamos eliminar una alternativa, tenemos que seleccionar el checkbox correspondiente y pulsar *Eliminar*, si por el contrario lo que queremos es añadir, introduciremos la descripción de la alternativa en el campo de texto en blanco y pulsamos *Añadir*. También podemos eliminar un experto asignado al problema de la misma manera que una alternativa. Para añadir un nuevo experto, utilizamos la lista desplegable y pulsamos *Añadir*.

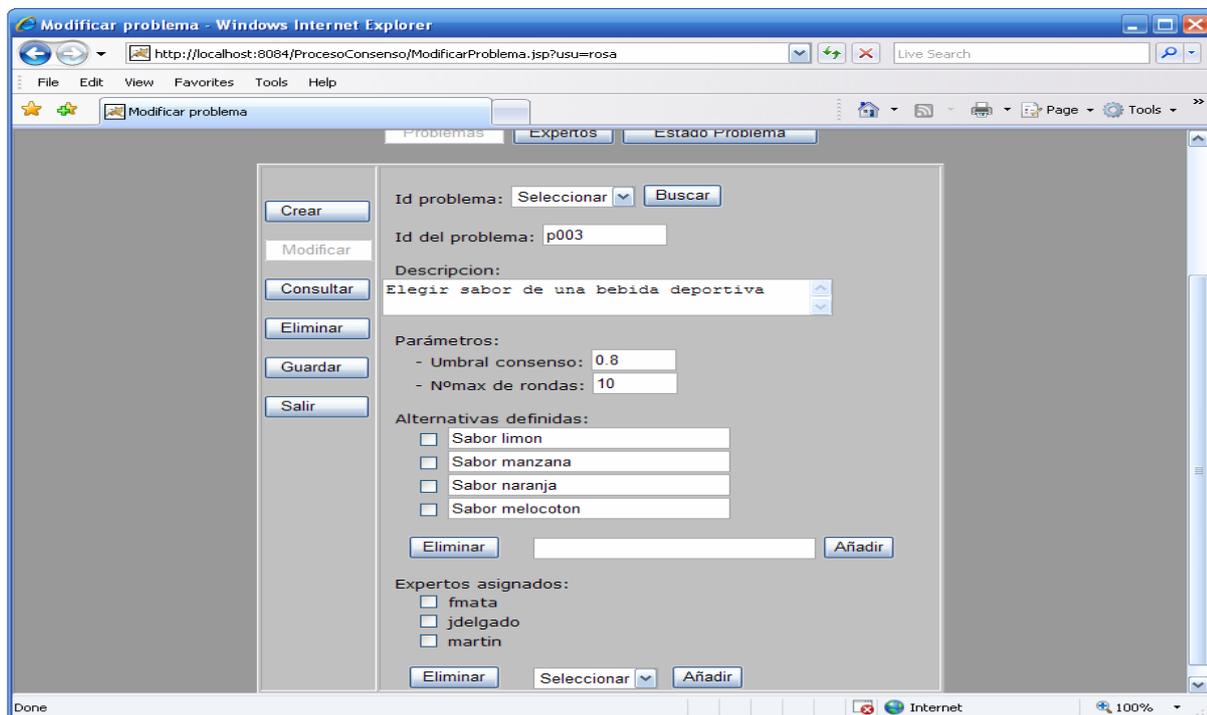


Figura II.15: Modificación de un problema

Una vez que hemos realizado todos los cambios pulsamos *Guardar*, en ese momento el sistema comprueba que todos los datos son correctos y nos muestra un mensaje de éxito como el de la Figura II.16. Si modificamos los valores de umbral de consenso o n^o máximo de rondas y éstos son erróneos, el sistema nos muestra un mensaje de error como hemos visto en la acción *Crear problema*.

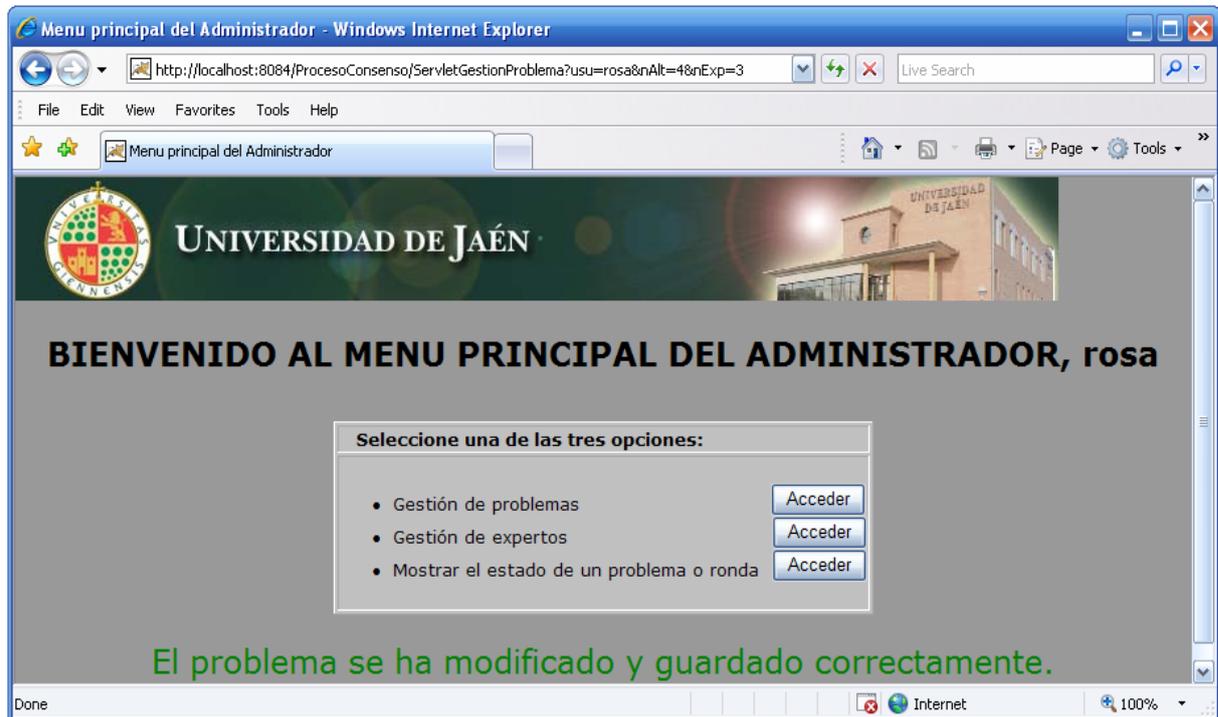


Figura II.16: Mensaje de éxito de modificación de un problema

◆ Consultar

Para consultar un problema, tenemos que pulsar *Consultar* y el sistema nos muestra una ventana como la de la Figura II.14

Una vez seleccionado el problema, pulsamos *Buscar* y el sistema nos muestra un formulario como el de la Figura II.17, con todos los datos del problema.

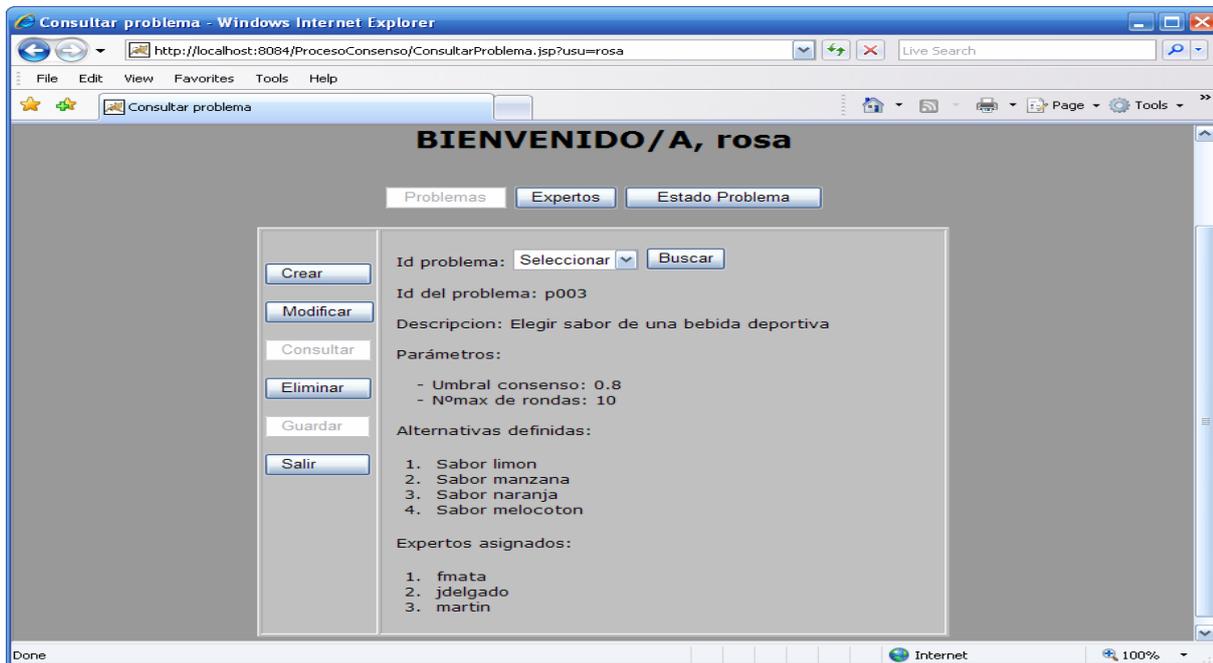


Figura II.17: Consultar un problema

◆ Eliminar

Para eliminar un problema, tenemos que pulsar *Eliminar* y el sistema nos muestra una ventana como la de la Figura II.14.

Una vez seleccionado el problema, pulsamos *Buscar* y el sistema nos muestra una ventana como en la Figura II.18.



Figura II.18: Eliminación de un problema

Pulsamos *Eliminar* y nos aparece un mensaje de confirmación como el de la Figura II.19.



Figura II.19: Mensaje de confirmación para eliminar un problema

Si estamos seguros de eliminar el problema, pulsamos *Aceptar* y el sistema nos muestra un mensaje de éxito como en la Figura II.20.



Figura II.20: Mensaje de éxito de eliminación de un problema

◆ **Salir**

Si el Administrador desea salir de la aplicación, pulsaría esta acción y aparecería la siguiente pantalla, Figura II.21.



Figura II.21: Salir de la aplicación

2. Gestión de expertos

Una vez que el Administrador se ha autenticado y nos encontramos en el menú principal, pulsamos *Acceder* y nos aparecerá un submenú con todas las acciones que el Administrador puede realizar en la opción *Gestión de expertos*, Figura II.22. Éstas, son las mismas que tiene la opción *Gestión de problemas*, vistas en el punto anterior.



Figura II.22: Submenú para la Gestión de Expertos

Veamos cada una de las distintas acciones con más detalle.

◆ Crear

Para crear un nuevo experto, pulsamos *Crear* y nos aparecerá un formulario como el de la Figura II.23.

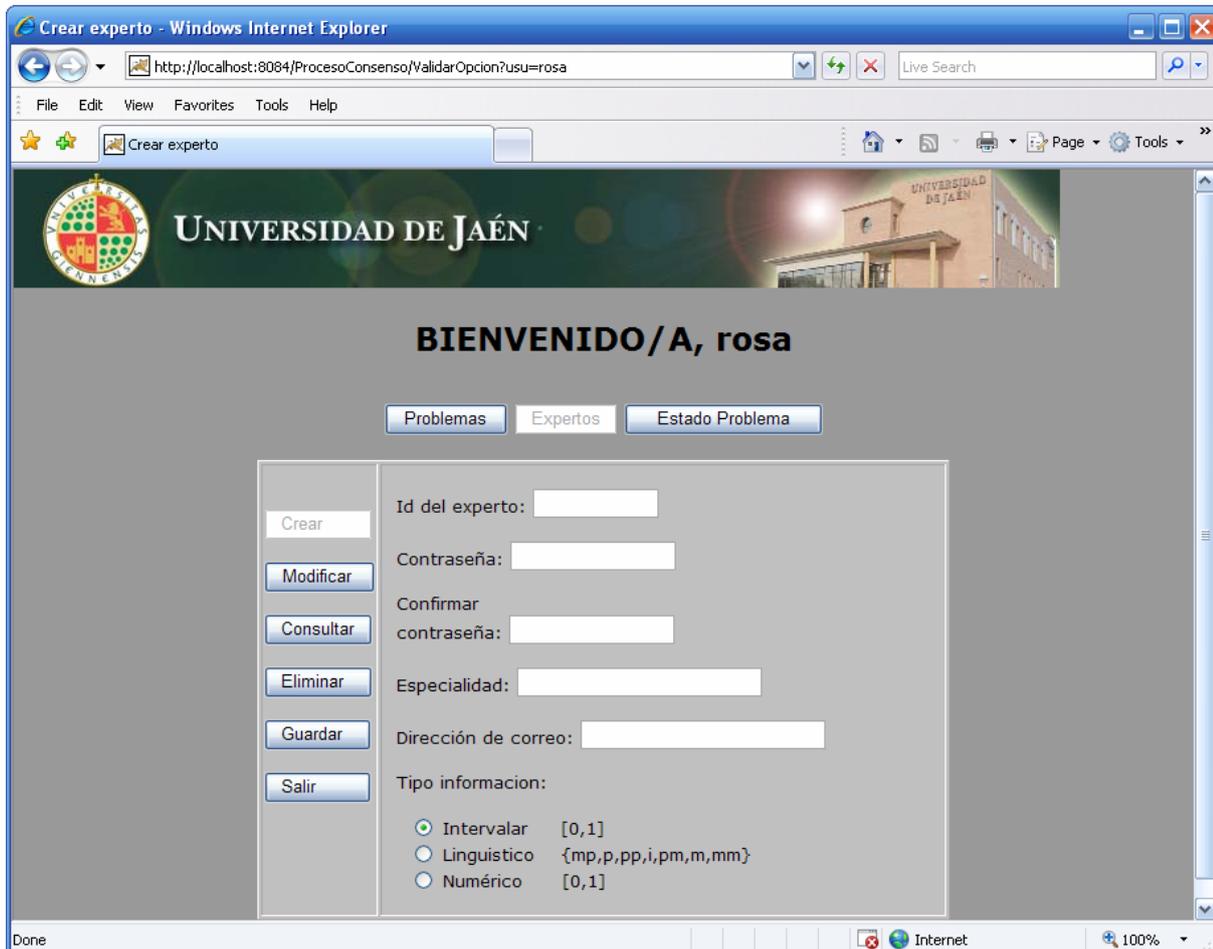


Figura II.23: Creación de un experto

Una vez rellenado el formulario, pulsamos *Guardar*. Si nos hemos dejado algún campo en blanco, el sistema nos muestra un mensaje de error indicando el campo concreto que se ha quedado vacío.

En la Figura II.24, podemos ver un ejemplo si dejamos en blanco el campo especialidad.

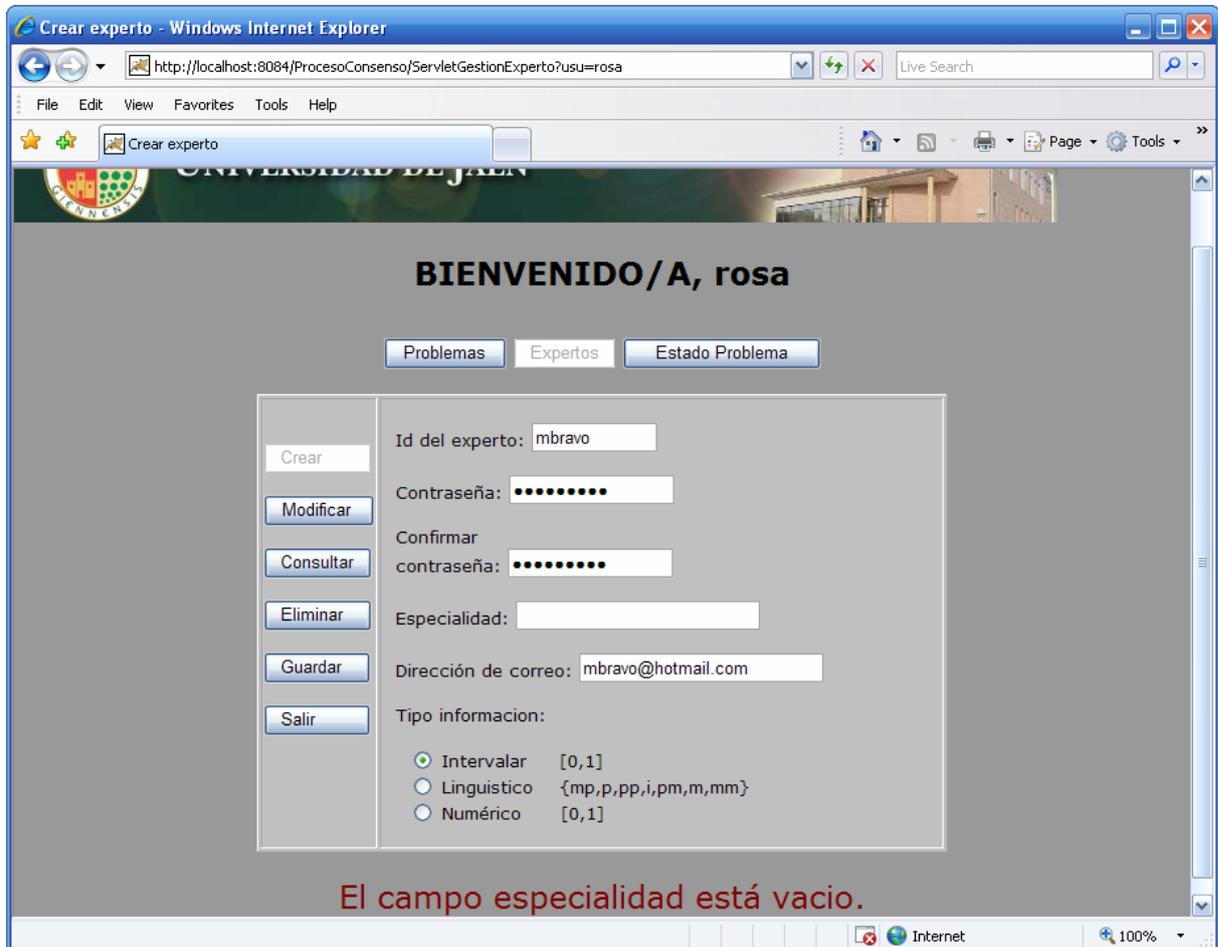


Figura II.24: Mensaje de error 'El campo especialidad está vacío'

Si introducimos un identificador que ya existe en la base de datos, el sistema nos muestra un mensaje de error como el de la Figura II.25.



Figura II.25: Mensaje de error 'El identificador del experto ya existe'

Si la contraseña y confirmación de contraseña no coinciden, nos aparecerá un mensaje de error como el de la Figura II.26.

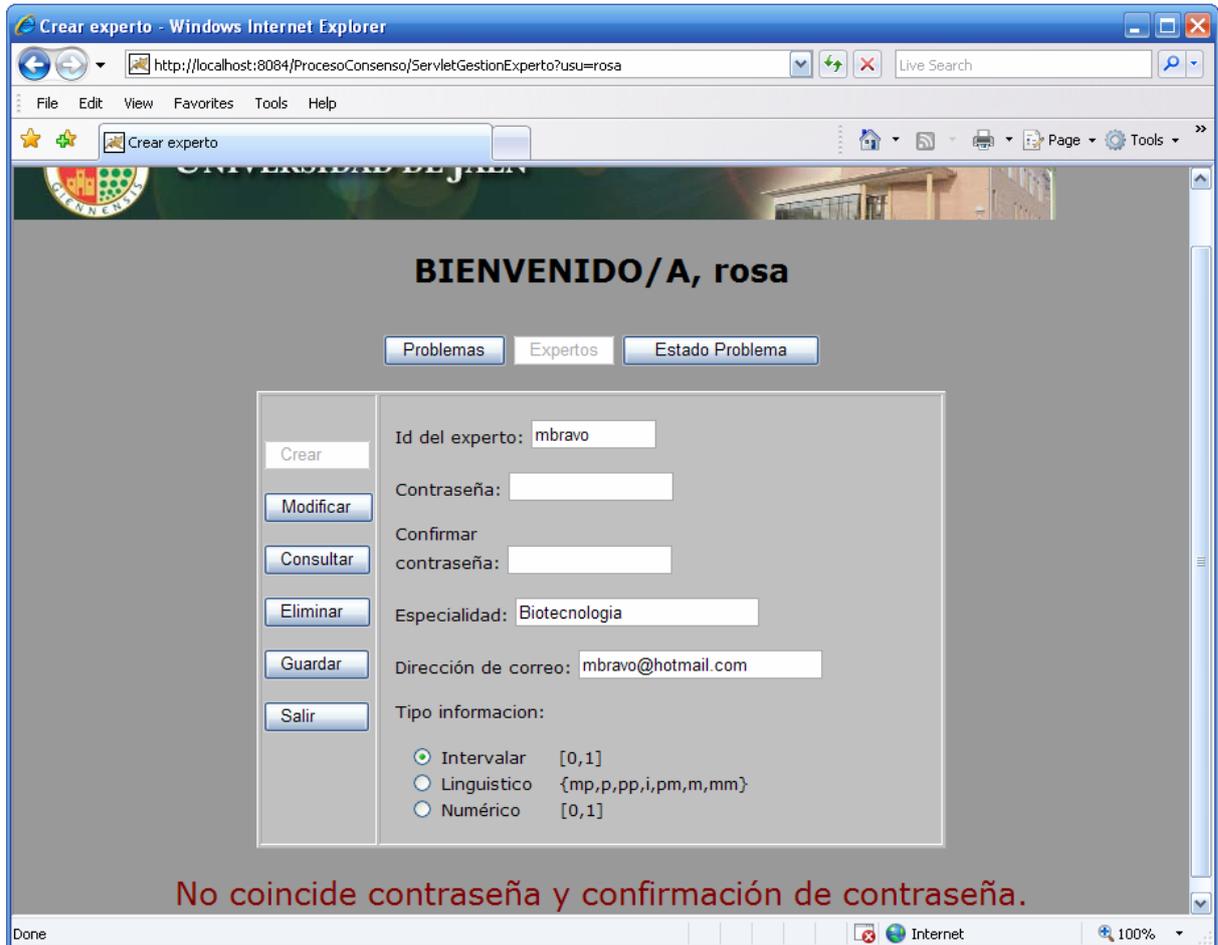


Figura II.26: Mensaje de error 'No coincide contraseña y confirmación de contraseña'

Si la dirección de correo no es válida, el sistema nos muestra un mensaje de error como el de la Figura II.27.

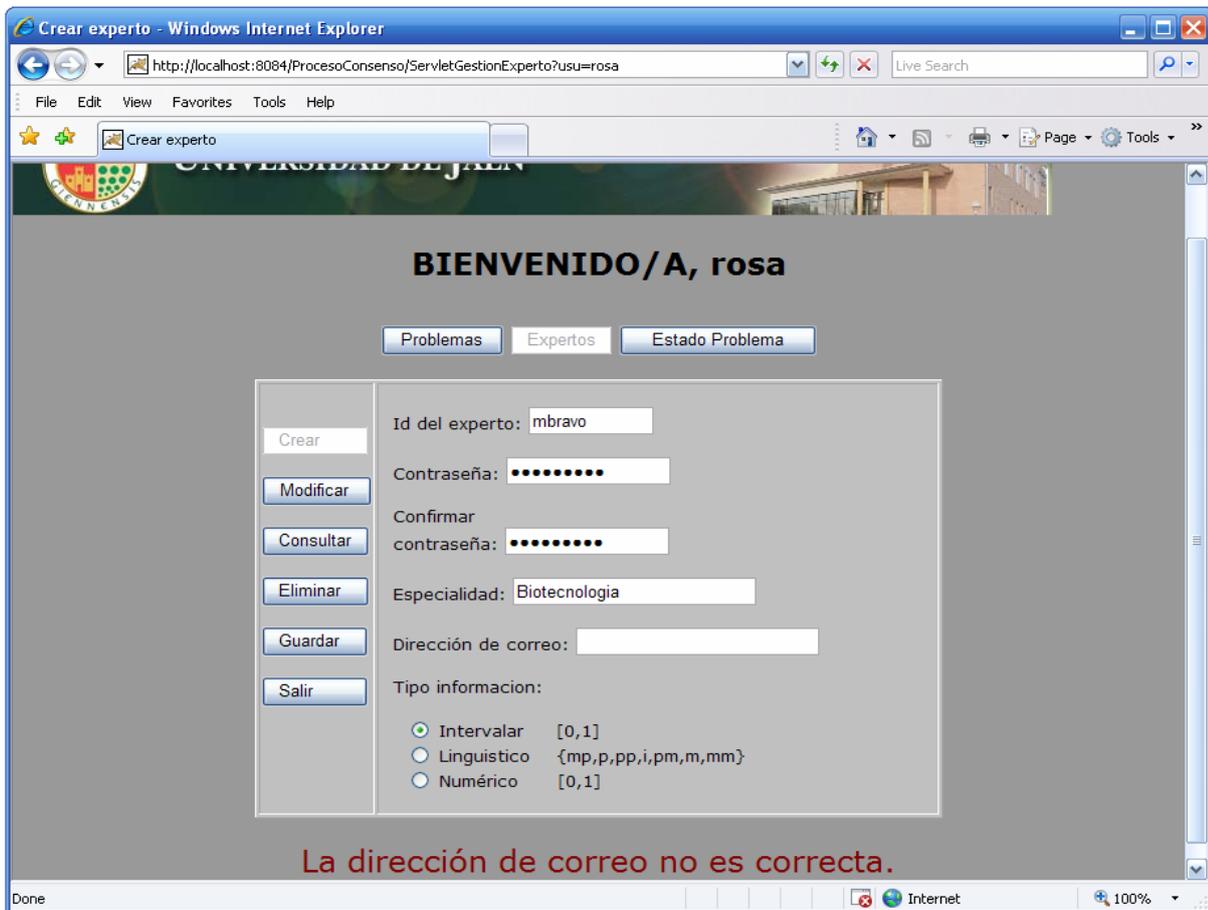


Figura II.27: Mensaje de error 'La dirección de correo no es correcta'

En el caso de pulsar cualquier otro botón que no sea *Guardar*, nos aparecerá un mensaje de confirmación, Figura II.28, para indicarnos que vamos a cambiar de opción sin guardar el experto.



Figura II.28: Mensaje de confirmación al cambiar de opción en Crear experto

Si pulsamos *Guardar* y hemos rellenado el formulario adecuadamente, se nos mostrará un mensaje de éxito como el de la Figura II.29.

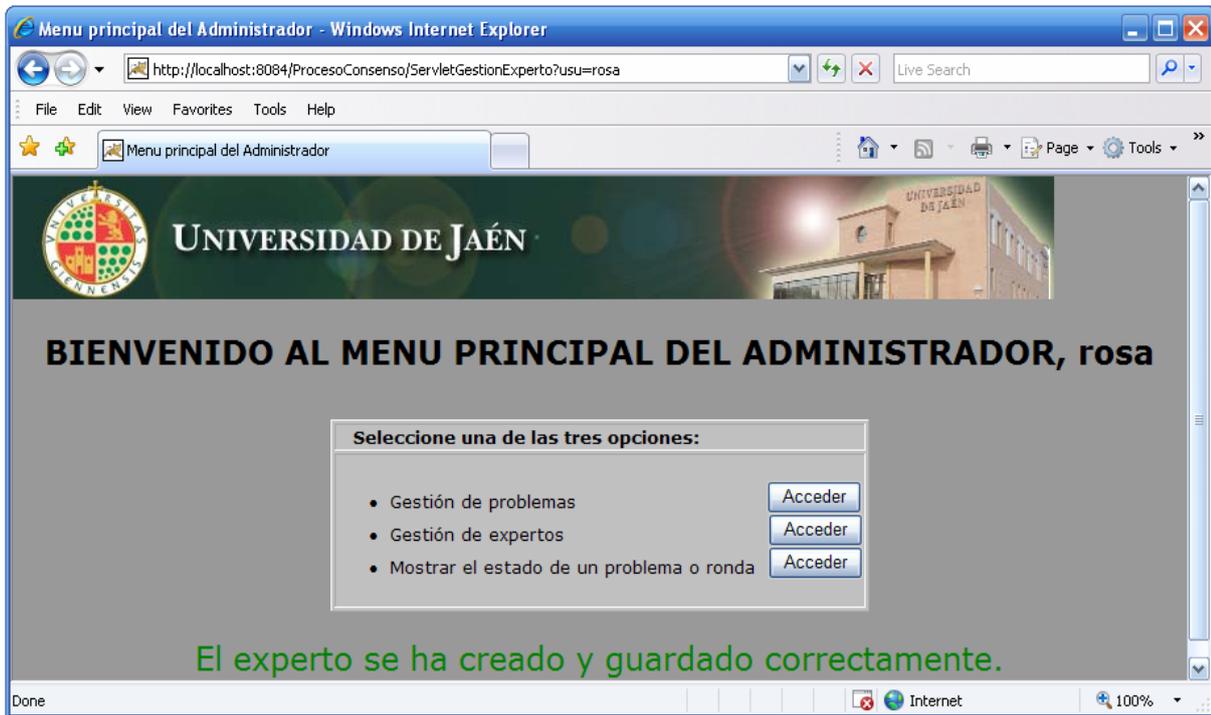


Figura II.29: Mensaje de éxito de creación de un experto

◆ **Modificar**

Para modificar un experto pulsamos *Modificar* y nos aparecerá una ventana con una lista desplegable en la que debemos seleccionar el experto que queremos modificar, Figura II.30.

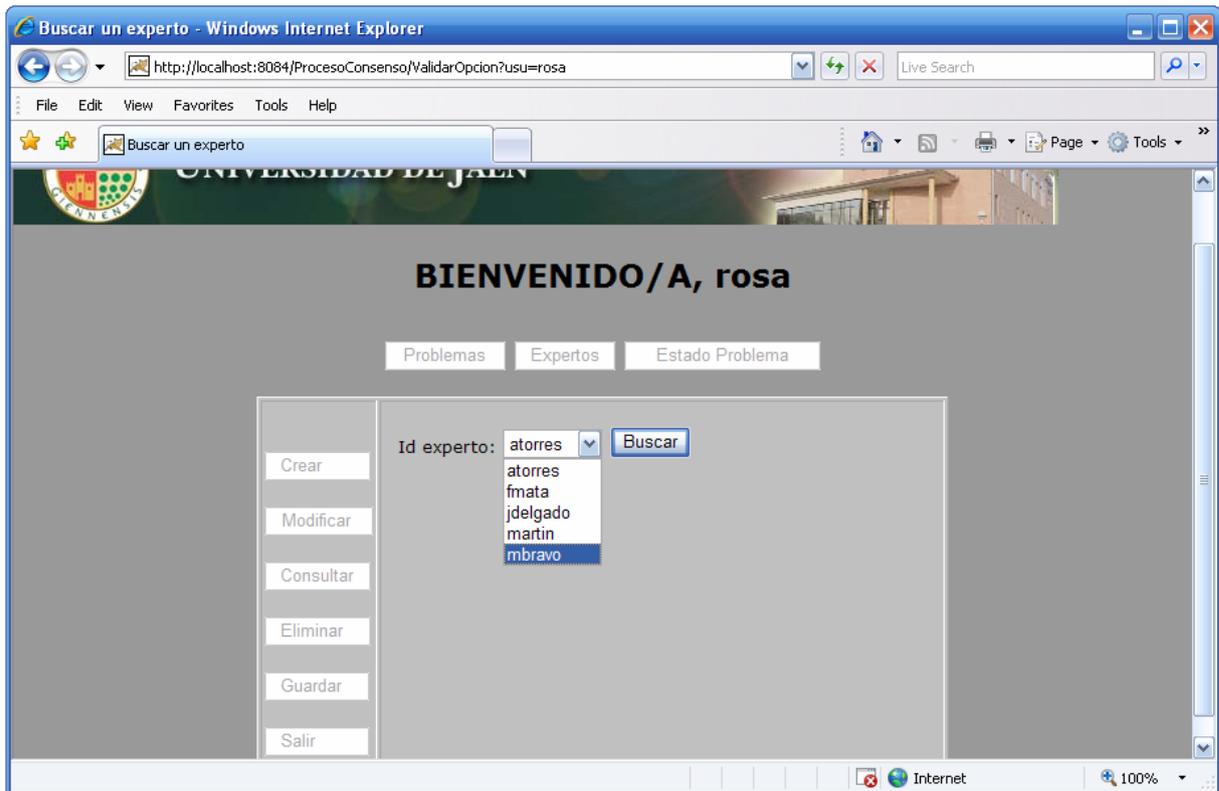


Figura II.30: Selección del experto

Una vez seleccionado el identificador del experto, pulsamos *Buscar* y el sistema nos mostrará un formulario con toda la información del experto, Figura II.31.

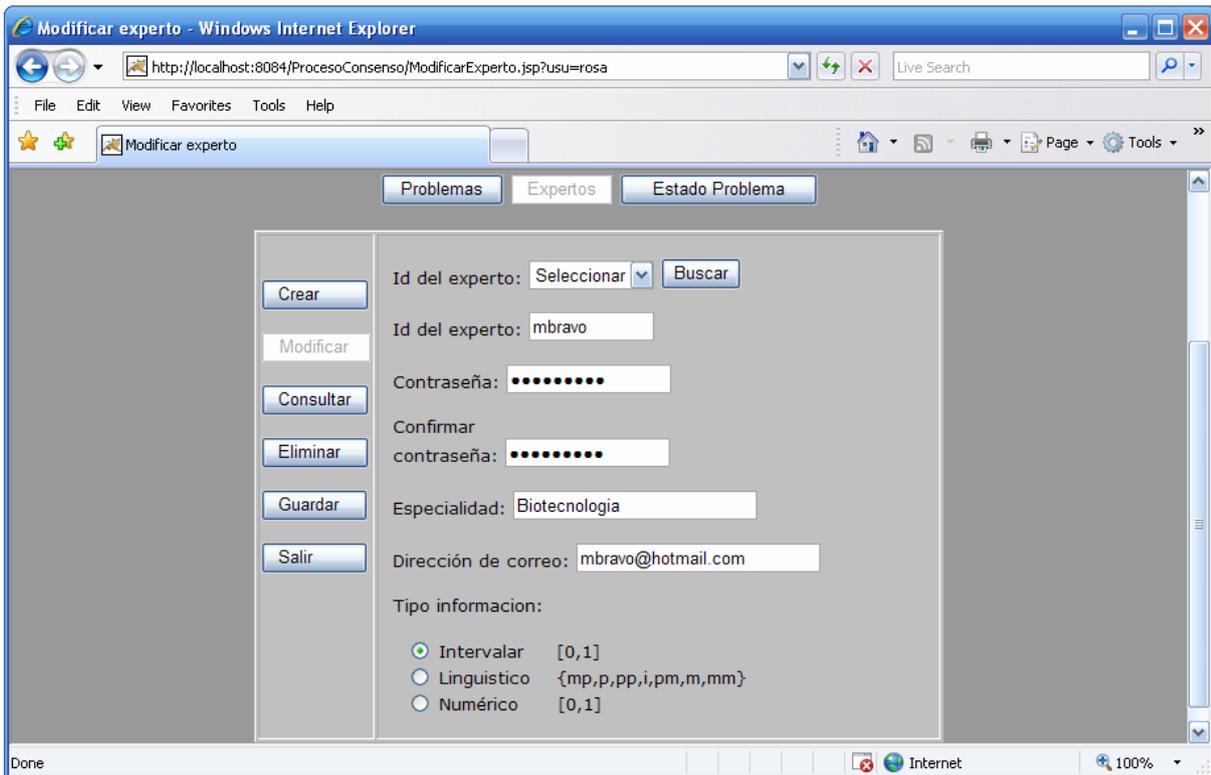


Figura II.31: Modificación de un experto

Podemos modificar cualquier campo de texto, excepto el identificador del experto.

Una vez que hemos realizado los cambios oportunos, pulsamos *Guardar*. El sistema comprobará que todos los datos son correctos y nos mostrará un mensaje de éxito como el de la Figura II.32. Si modificamos la contraseña o dirección de correo y éstos no son válidos, el sistema nos mostrará un mensaje de error como hemos visto en la acción *Crear experto*.



Figura II.32: Mensaje de éxito de modificación de un experto

◆ Consultar

Para consultar un experto, tenemos que pulsar *Consultar* y se nos mostrará una ventana como la de la Figura II.30.

Una vez que hemos seleccionado el experto, pulsamos *Buscar* y el sistema nos muestra un formulario como el de la Figura II.33, con todos los datos del experto, los problemas que tiene asignados, el estado en el que se encuentra cada uno de ellos, el n^o de ronda actual y el n^o de cambios que el experto tiene que realizar en esa ronda.

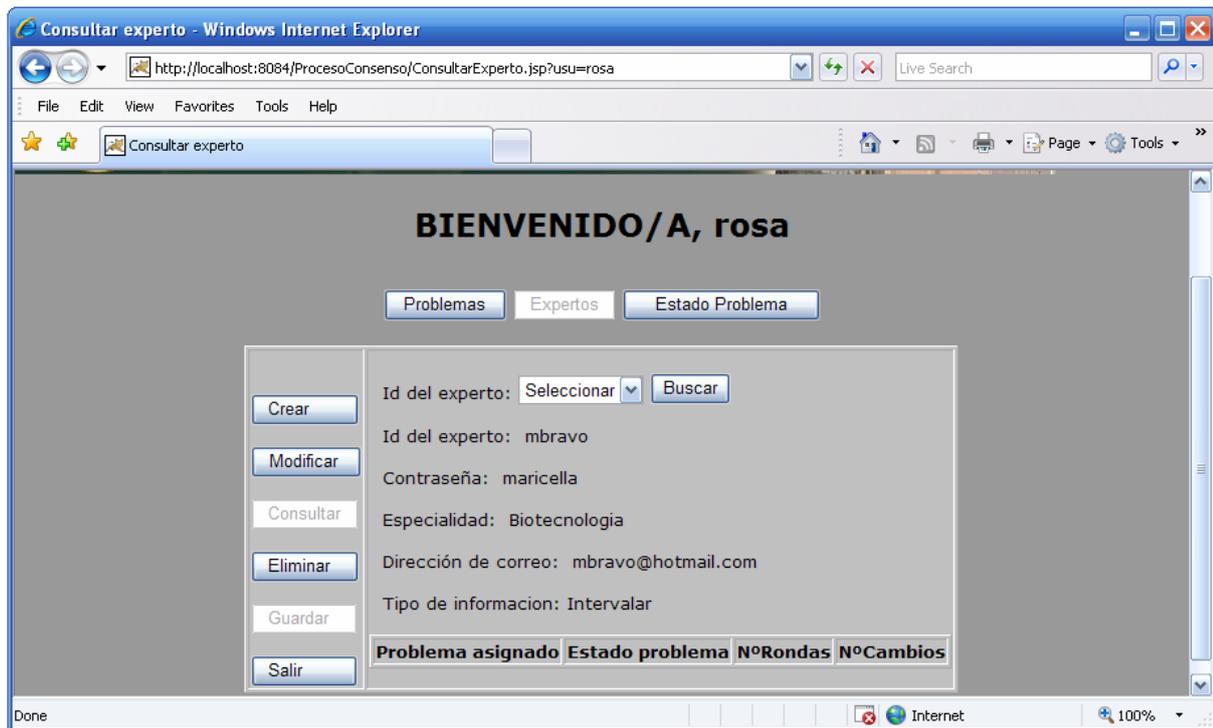


Figura II.33: Consultar un experto

◆ Eliminar

Para eliminar un experto, pulsamos *Eliminar* y el sistema nos muestra un formulario como el de la Figura II.30.

Una vez seleccionado el experto, pulsamos *Buscar* y el sistema nos muestra una ventana como en la Figura II.34.



Figura II.34: Eliminación de un experto

Cuando pulsemos *Eliminar*, se nos mostrará un mensaje de confirmación como en la Figura II.35.

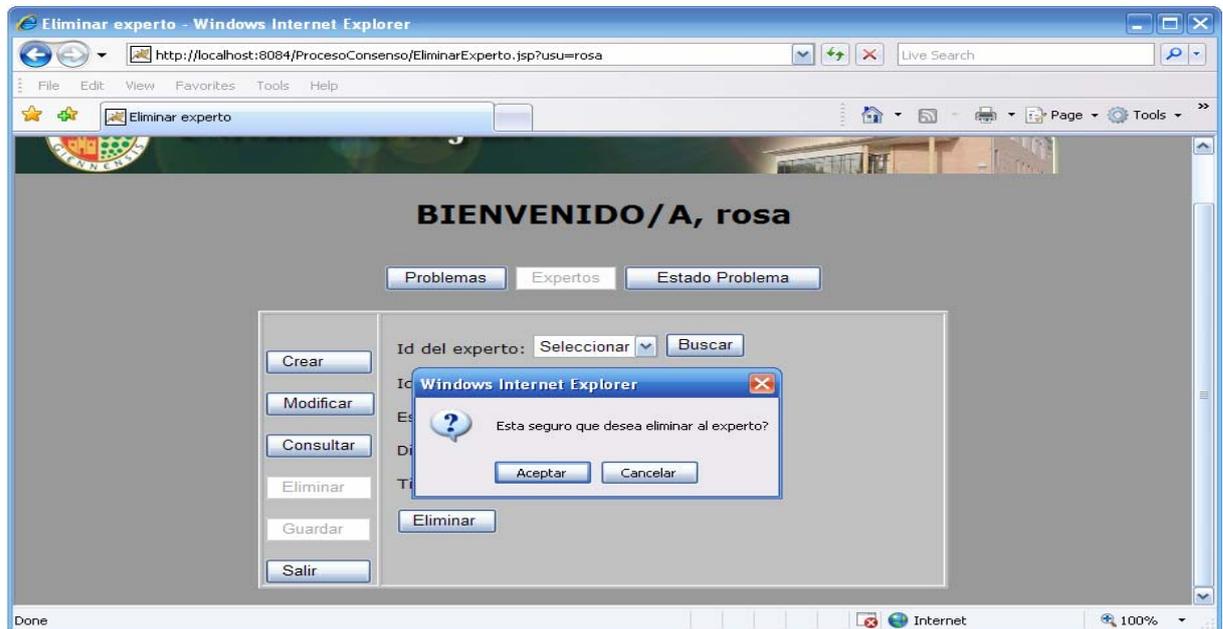


Figura II.35: Mensaje de confirmación para eliminar un experto

Si estamos seguros de eliminar el experto, pulsamos *Aceptar* y nos aparecerá un mensaje de éxito como en la Figura II.36.

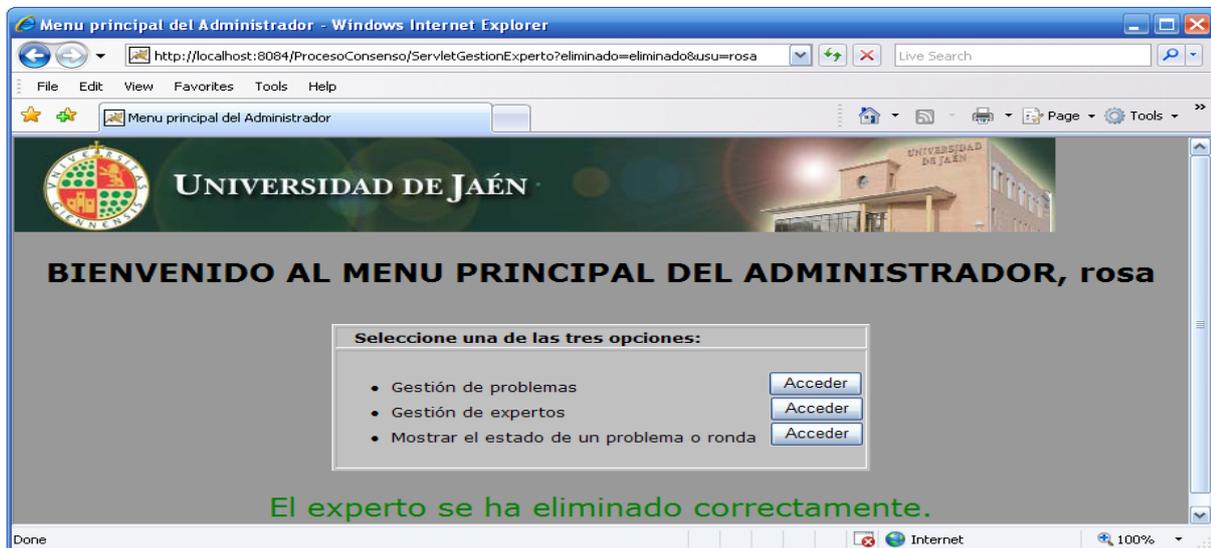


Figura II.36: Mensaje de éxito de eliminación de un experto

3. Mostrar el estado de un problema o ronda

Si en el menú principal del administrador accedemos a la opción *Mostrar el estado de un problema o ronda*, el sistema nos muestra un submenú con una tabla que contiene todos los problemas creados. Tenemos que seleccionar un problema y pulsar *Buscar*. En ese momento el sistema habilita las posibles acciones que se pueden realizar sobre dicho problema, Figura II.37.



Figura II.37: Posibles acciones en la opción Estado problema

Veamos cada una de estas acciones más detenidamente.

◆ **Calcular nivel de consenso**

Si pulsamos *Calcular nivel de consenso* y aún hay expertos que no han introducido sus preferencias, se nos mostrará una ventana como la de la Figura II.38.



Figura II.38: No se puede calcular el nivel de consenso

Si por el contrario todos los expertos han introducido sus preferencias, se calculará el nivel de consenso y se nos mostrará una ventana que nos indicará si se ha alcanzado o no el grado de consenso, Figura II.39.



Figura II.39: Se ha calculado el nivel de consenso

◆ Evolución consenso

Si pulsamos *Evolución consenso*, se nos mostrará un formulario con todos los detalles del proceso de consenso del problema previamente seleccionado, Figura II.40. Además, podemos seleccionar cualquier ronda anterior y ver el grado de consenso alcanzado y el nº de cambios totales e individuales realizados por los expertos. También podemos visualizar las preferencias de cualquier experto en cualquier ronda si seleccionamos el identificador del experto, el nº de ronda y pulsamos *Ver preferencias*.

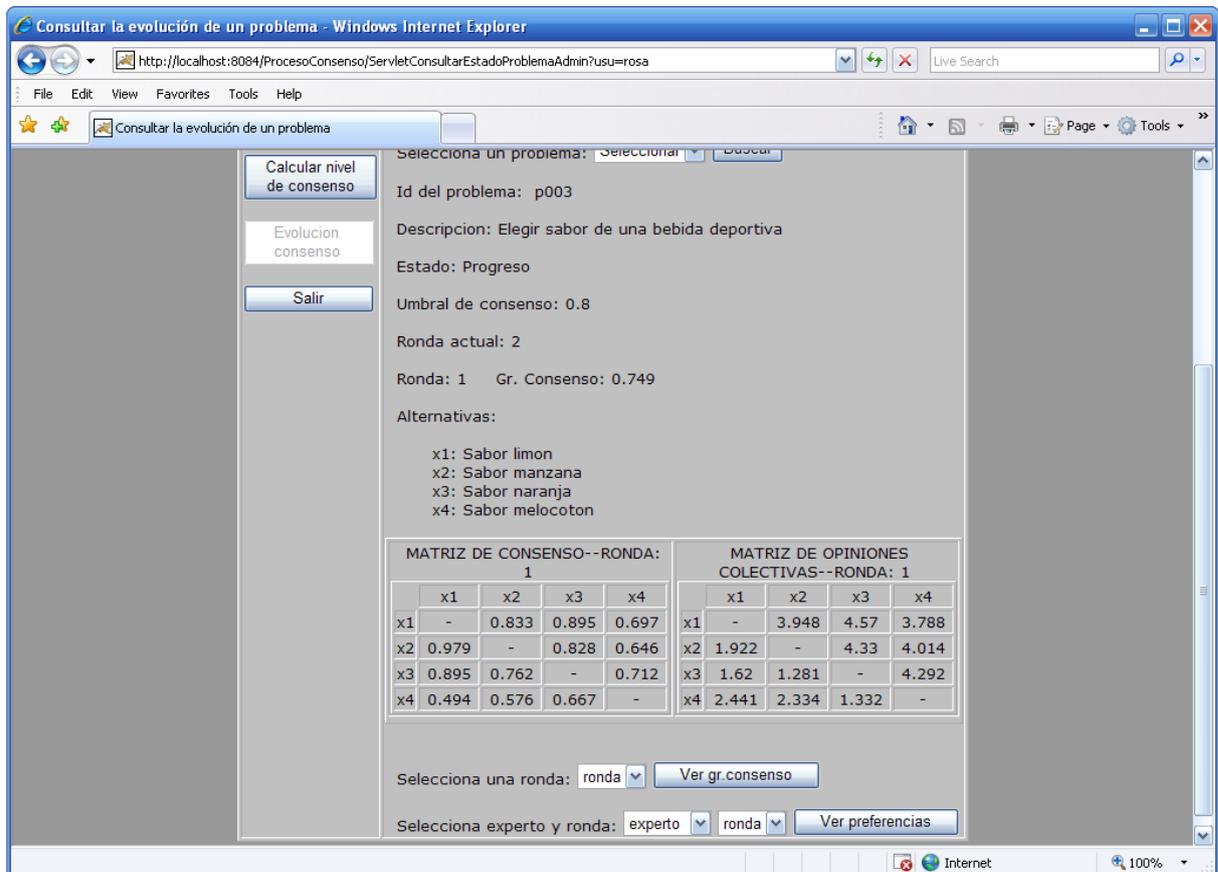


Figura II.40: Evolución de consenso

ANEXO III.

MANUAL DE USUARIO

DEL EXPERTO

Este manual de usuario está organizado como una visita guiada por la aplicación, cuando entramos como Experto.

El primer paso para utilizar la aplicación es abrir el Navegador (Internet Explorer) y teclear la dirección en la que tenemos alojada ésta, que en nuestro caso será: <http://localhost:8084/ProcesoConsenso/index.jsp>.

Al realizar lo anteriormente comentado, nos encontramos con la página de inicio, tal y como se ve en la Figura III.1.



Figura III.1: Entrada al sistema

Para entrar como Experto debemos introducir el nombre de usuario y contraseña del experto. Si alguno de estos datos (o los dos) no son correctos, el sistema nos muestra un mensaje de error como se observa en la Figura III.2.



Figura III.2: Mensaje de error en la autenticación

En el caso de una autenticación correcta nos aparece el menú principal del experto, como podemos ver en la Figura III.3



Figura III.3: Menú principal del experto

En este menú principal podemos encontrar una tabla con los problemas que el Experto tiene asignados y las opciones que puede realizar.

A continuación veremos cada una de ellas con más detalle.

◆ Introducir opinión

El primer paso consiste en seleccionar el problema que deseamos valorar y pulsar *Introducir Opinión*. Si el estado del problema seleccionado fuese *Finalizado*, el sistema nos mostraría un mensaje de error como el de la Figura III.4.

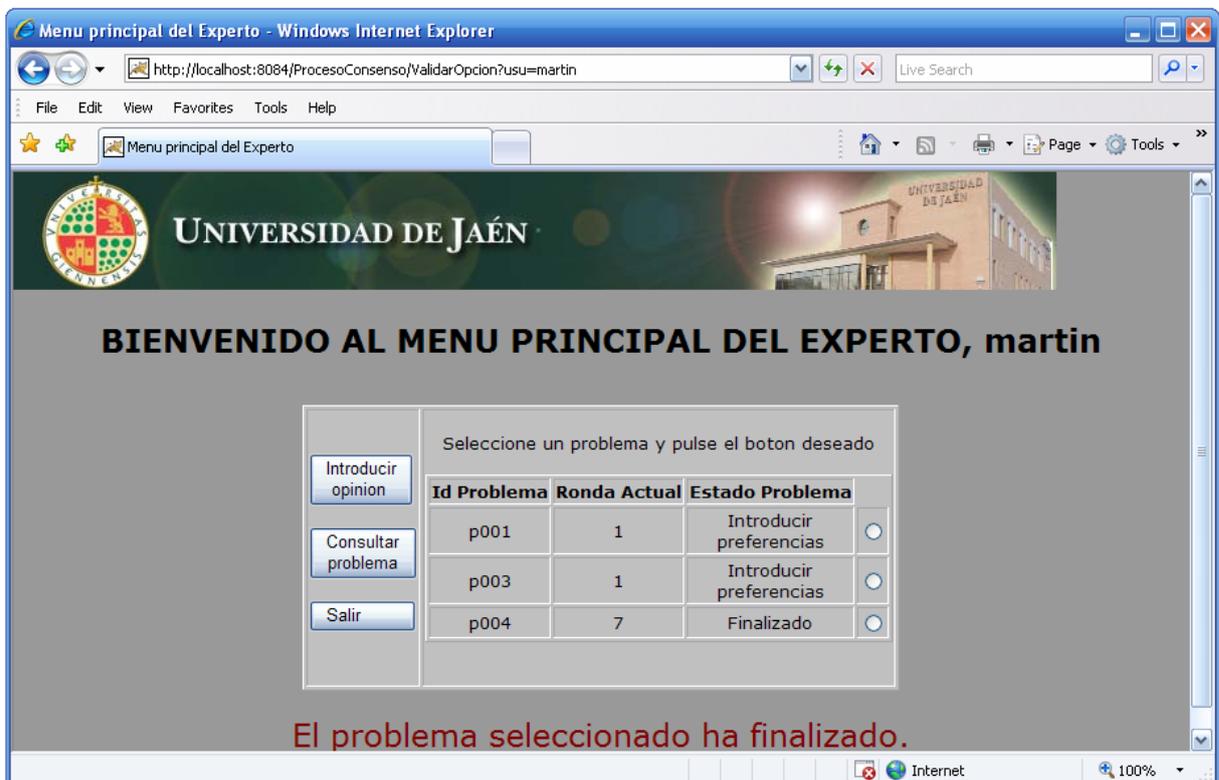


Figura III.4: Mensaje de error 'El problema seleccionado ha finalizado'

Si el estado del problema es *Introducir preferencias*, se mostraría un formulario como el de la Figura III.5.

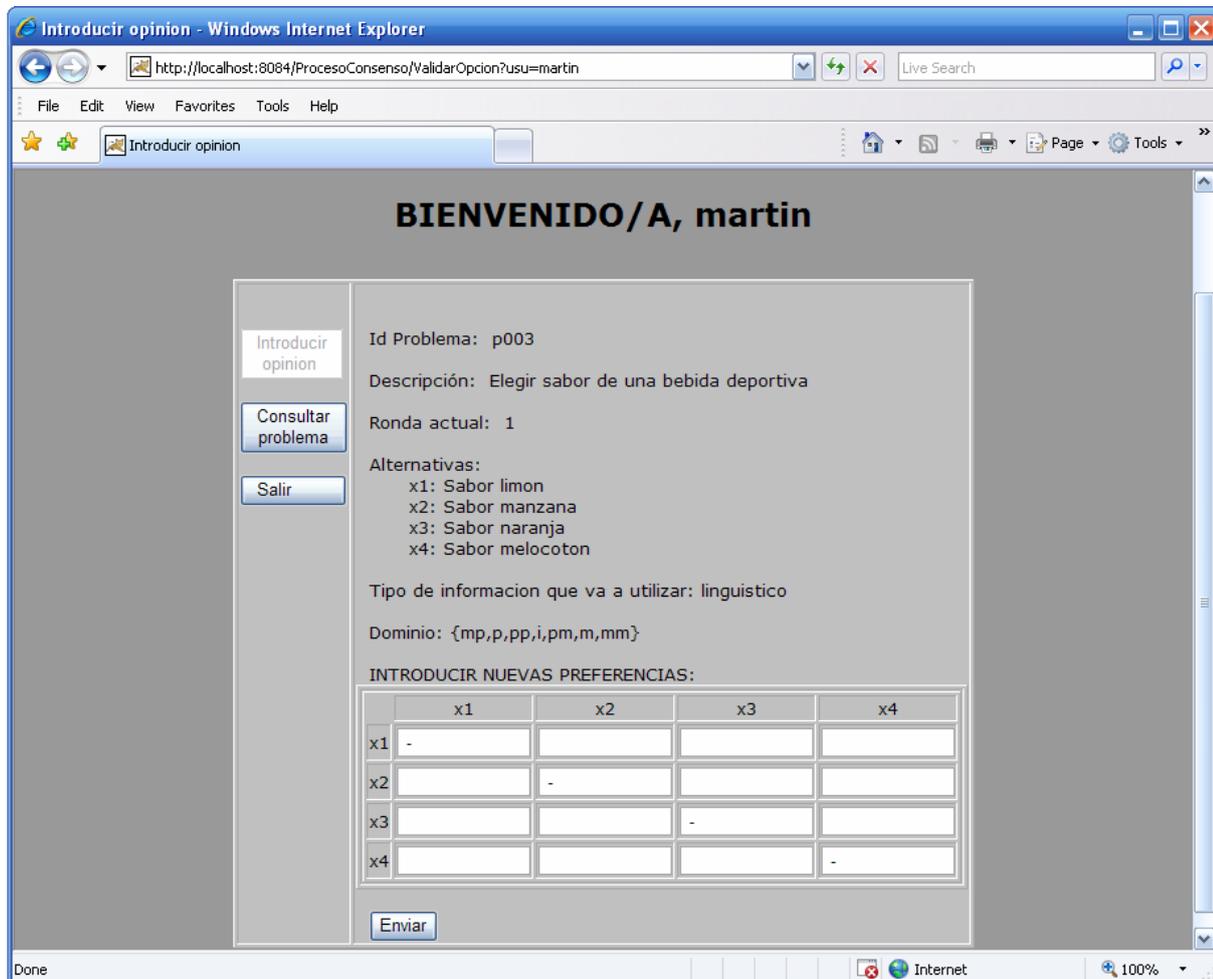


Figura III.5: Valoración de un problema

Una vez que hayamos introducido nuestras preferencias pulsamos *Enviar*.

Si hemos dejado en blanco la valoración sobre un par de alternativas o alguno de los valores introducidos está fuera del dominio, se nos mostrará un mensaje de error como el de la Figura III.6.

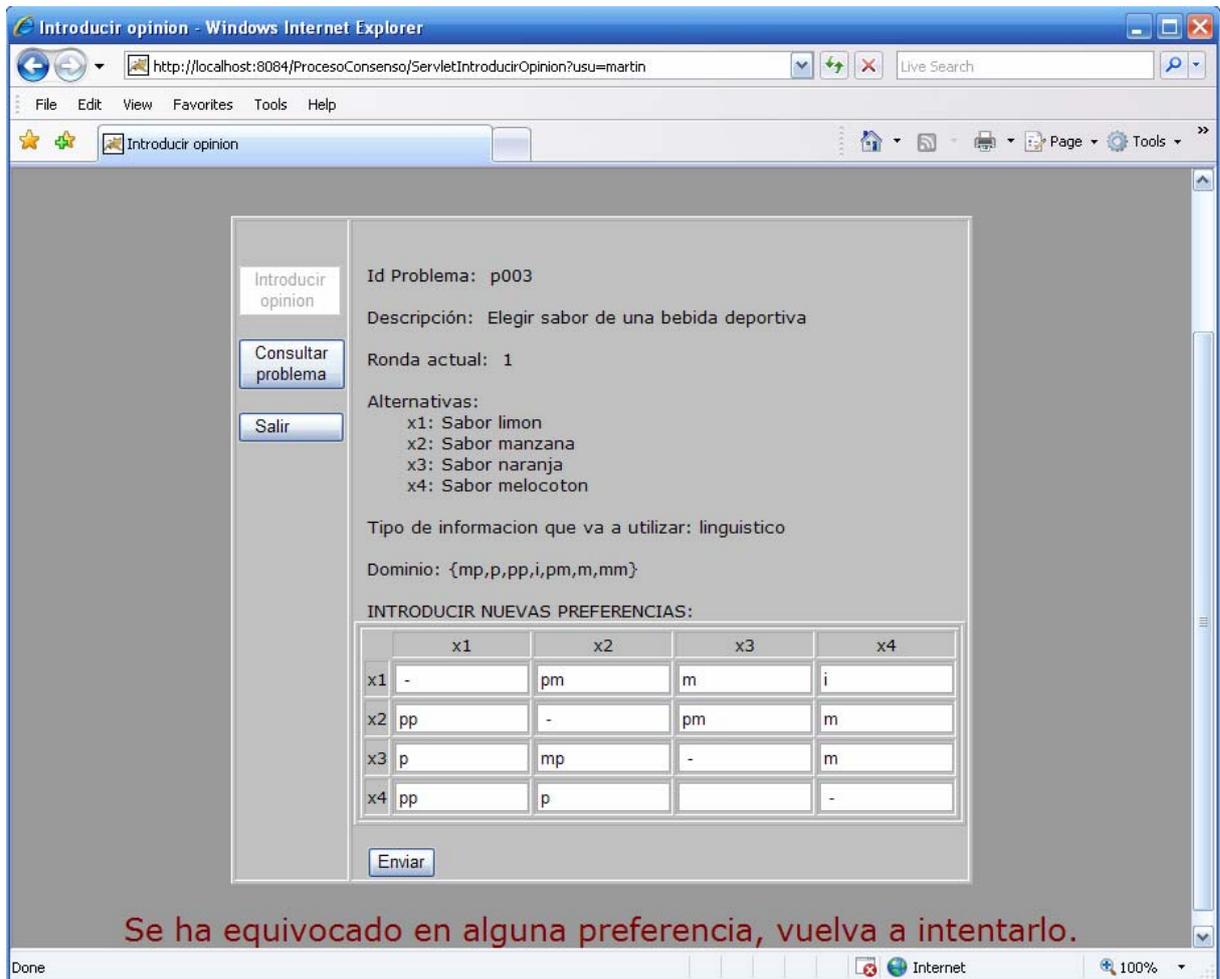


Figura III.6: Mensaje de error al introducir un valor

Si pulsamos cualquier otro botón que no sea *Enviar*, nos aparecerá un mensaje de confirmación indicándonos que vamos a cambiar de opción y no hemos enviado nuestras preferencias, Figura III.7.

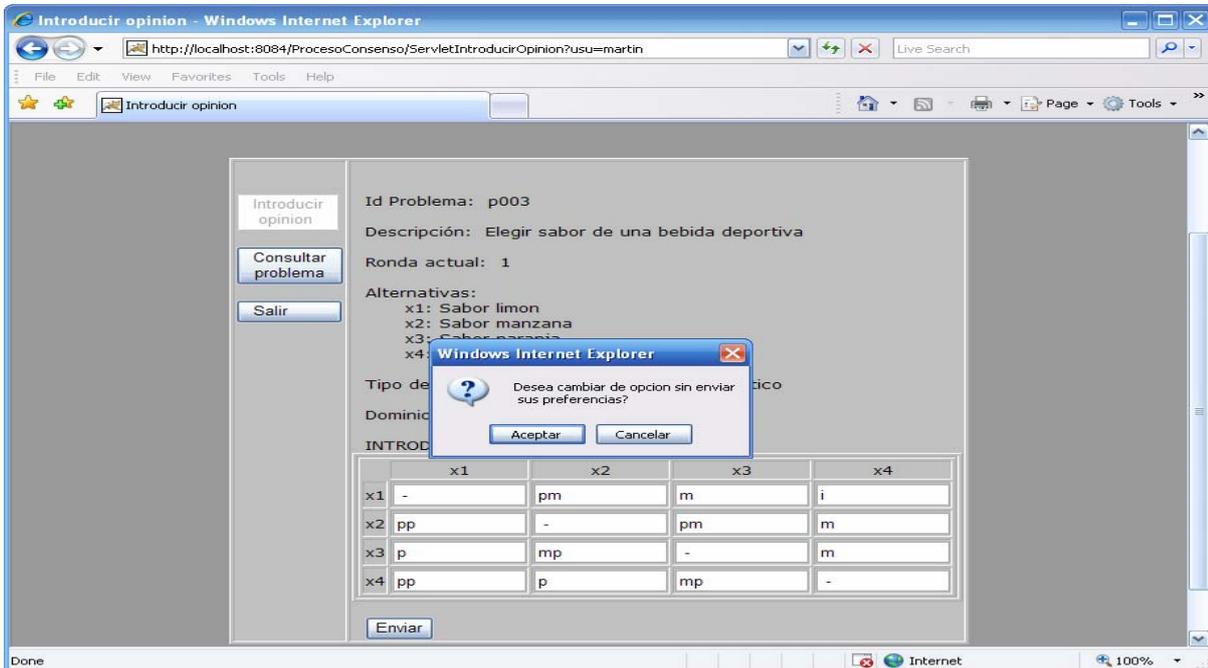


Figura III.7: Mensaje de confirmación al cambiar de opción en Introducir preferencias

Si todos los valores son correctos, el sistema nos mostrará un mensaje de éxito como el de la Figura III.8.

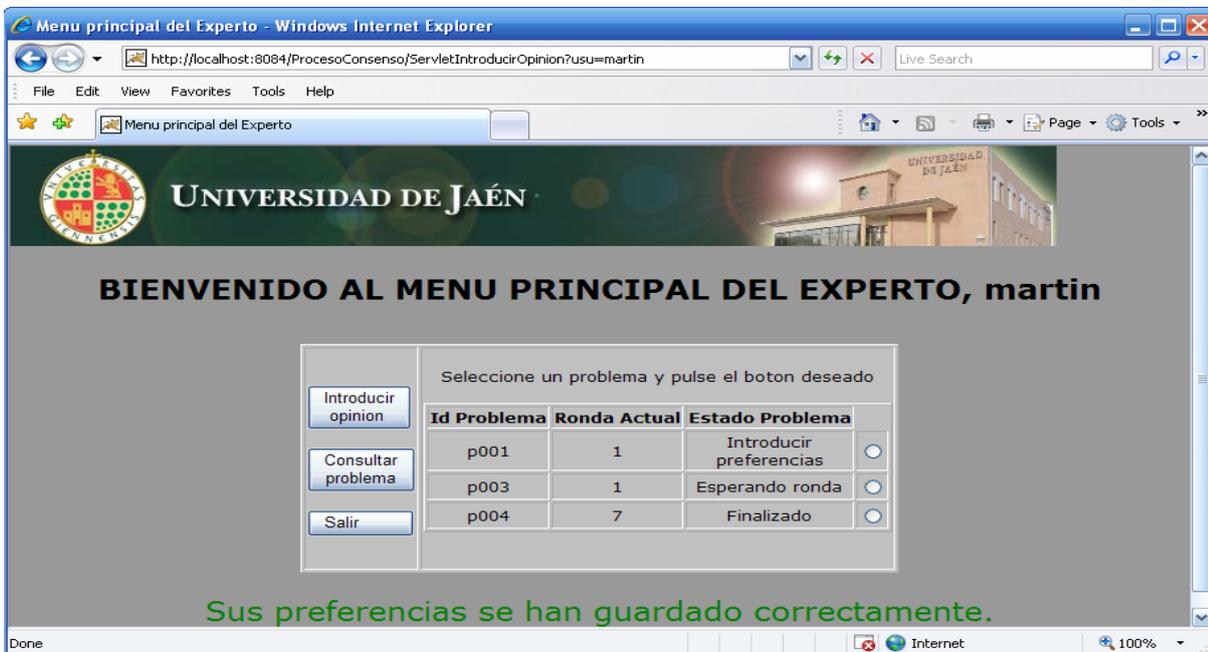


Figura III.8: Mensaje de éxito en la valoración de un problema

◆ Consultar problema

Si pulsamos *Consultar problema*, nos aparecerá un formulario con las preferencias introducidas en la última ronda, Figura III.9.

Si nos encontrásemos en la segunda ronda o posterior y se nos hubieran generado recomendaciones, éstas se visualizarían en colores (rojo decrementar y verde incrementar).

Además, podemos ver las preferencias y recomendaciones si las hubiera de cualquier ronda anterior, si la seleccionamos y pulsamos *Ver*.

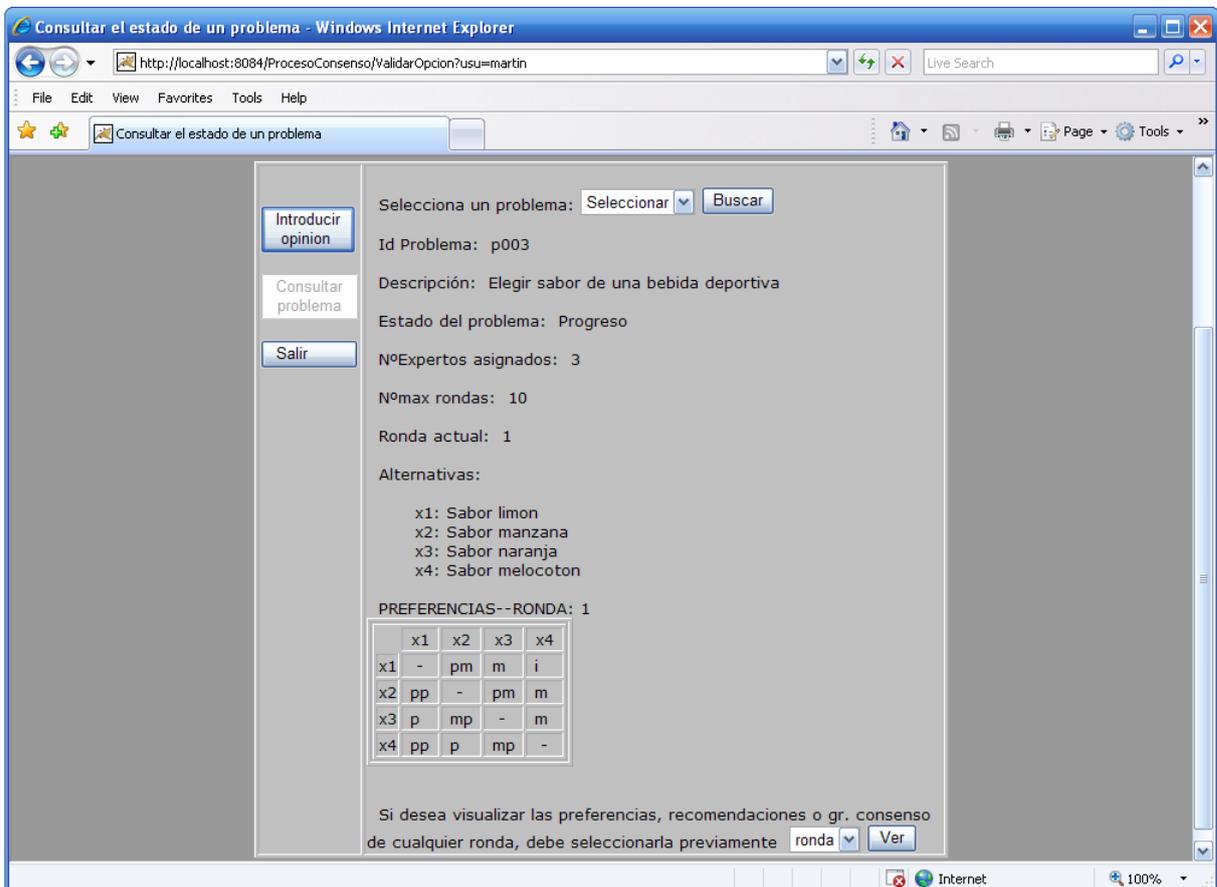


Figura III.9: Consultar las preferencias de un problema

◆ **Salir**

Si el Experto desea salir de la aplicación, pulsaría esta acción y aparecería la siguiente pantalla, Figura III.10.



Figura III.10: Salir de la aplicación