



UNIVERSIDAD DE JAÉN

Escuela Politécnica Superior (Jaén)

Proyecto Fin de Carrera

MODELO DE RESOLUCIÓN DE PROBLEMAS DE TOMA DE DECISIÓN MULTICRITERIO

Alumno: Manuel Quesada Armenteros

Tutor: Luis Martínez López
Rosa M^a Rodríguez Domínguez

Dpto: Informática

Febrero, 2015

Gracias a mi familia por animarme siempre a seguir adelante.

Gracias a mi mujer porque su sacrificio y su insistencias han avivado mis fuerzas perdidas. Gracias a Rosa que, a pesar de toda la distancia, siempre ha estado ahí para ayudarme. Y muchísimas

Gracias a Luis, que como director de proyecto, me ha soportado todos estos años y, a pesar de mi cabeza olvidadiza, me ha guiado perfectamente y ha hecho posible que este proyecto llegue a su fin.

ÍNDICE

| | |
|--|-----------|
| Capítulo 1. Introducción | 1 |
| 1.1 Introducción al Proyecto | 2 |
| 1.2 Propósito del Proyecto..... | 3 |
| 1.3 Objetivos del Proyecto | 3 |
| 1.4 Estructura de la memoria del Proyecto Fin de Carrera..... | 3 |
| Capítulo 2. Antecedentes y Definiciones de Teoría de Toma de Decisión | 5 |
| 2.1 Antecedentes | 6 |
| 2.1.1 Toma de Decisión..... | 6 |
| 2.1.2 Criterios de Clasificación de los Problemas de Toma de Decisión | 9 |
| 2.1.3 Esquema de Resolución de Problemas de Toma de Decisión en Grupo | 15 |
| 2.1.4 Modelado de preferencias..... | 19 |
| 2.1.4.1 Dominio Numérico | 20 |
| 2.1.4.2 Dominio Intervalar..... | 21 |
| 2.1.4.3 Dominio Lingüístico | 21 |
| 2.1.4.4 Estructuras de representación de preferencias | 21 |
| 2.1.5 Métodos de Decisión Multicriterio Discretos | 22 |
| 2.1.5.1 Escuela Americana | 23 |
| 2.1.5.1.1 Método de la Suma Ponderada | 24 |
| 2.1.5.1.2 El Proceso Analítico Jerárquico..... | 25 |
| 2.1.5.1.3 Método SMART | 25 |
| 2.1.5.2 Escuela Europea..... | 25 |
| 2.1.5.2.1 Método ELECTRE | 26 |
| 2.1.5.2.2 Método PROMETHEE | 26 |
| 2.1.5.2.3 Método MACBETH | 26 |
| 2.1.5.3 Método ZAPROS..... | 27 |
| 2.1.5.4 Método TOPSIS..... | 27 |
| 2.2 Método TOPSIS | 28 |
| 2.2.1 El concepto de Alternativa Ideal | 28 |
| 2.2.2 El Método TOPSIS..... | 28 |
| 2.2.3 Algoritmo del Método TOPSIS..... | 31 |
| 2.2.4 Un ejemplo Método TOPSIS..... | 33 |
| Capítulo 3. El Proceso de Ingeniería del Software | 37 |
| 3.1 Descripción del Proceso | 38 |
| 3.2 Especificación de requerimientos..... | 38 |

| | | |
|---------|--|-----|
| 3.2.1 | Requerimientos funcionales | 39 |
| 3.2.2 | Requerimientos no funcionales | 42 |
| 3.3 | Análisis del Sistema | 45 |
| 3.3.1 | Perfil de usuario..... | 45 |
| 3.3.2 | Casos de Uso | 46 |
| 3.3.3 | Escenarios..... | 59 |
| 3.4 | Diseño del Sistema | 65 |
| 3.4.1 | Diagrama completo de clases | 65 |
| 3.4.2 | Diagrama de Paquetes | 66 |
| 3.4.2.1 | Paquete controllers..... | 66 |
| 3.4.2.2 | Paquete models | 68 |
| 3.4.2.3 | Paquete dialogs | 69 |
| 3.4.2.4 | Paquete panels | 70 |
| 3.4.2.5 | Paquete tablemodels | 71 |
| 3.4.2.6 | Paquete utils..... | 72 |
| 3.4.2.7 | Paquete frames..... | 72 |
| 3.4.2.8 | Paquete table.columns | 73 |
| 3.4.3 | Diseño de Datos..... | 74 |
| 3.4.3.1 | Modelo Entidad-Relación..... | 74 |
| 3.4.3.2 | Normalización en el modelo Entidad-Relación | 75 |
| 3.4.3.3 | Esquema Conceptual | 75 |
| 3.4.3.4 | Esquema Conceptual Modificado..... | 77 |
| 3.4.3.5 | Tablas de aplicación | 78 |
| 3.4.4 | Diseño de la Interfaz..... | 80 |
| 3.4.4.1 | Estilo de la Interfaz..... | 81 |
| 3.4.4.2 | Metáforas | 82 |
| 3.4.4.3 | Pantallas..... | 82 |
| 3.4.4.4 | Camino de navegación | 93 |
| 3.4.4.5 | Mensajes de error..... | 101 |
| 3.4.4.6 | Mensajes de éxito | 107 |
| 3.4.4.7 | Mensajes de confirmación | 107 |
| 3.5 | Implementación | 108 |
| 3.5.1 | Tipos de la arquitectura de la aplicación | 109 |
| 3.5.2 | Lenguajes de programación y tecnologías utilizadas | 109 |
| 3.5.2.1 | Java | 109 |
| 3.5.2.2 | SQL..... | 111 |
| 3.5.2.3 | JPA..... | 111 |
| 3.5.2.4 | Librería Hibernate..... | 113 |

| | |
|--|------------|
| 3.5.3 Herramientas de desarrollo..... | 113 |
| 3.5.3.1 Netbeans | 113 |
| 3.6 Implantación y pruebas | 114 |
| 3.6.1 Pruebas y validación..... | 114 |
| 3.6.1.1 Casos de Test..... | 115 |
| 3.6.1.2 Resultados obtenidos | 121 |
| Capítulo 4. Conclusiones | 123 |
| 4.1 Conclusiones finales..... | 124 |
| 4.2 Trabajos futuros..... | 124 |
| Bibliografía | 127 |
| Anexo I. Manual de instalación del software | 131 |
| Anexo II. Manual de usuario (Administrador) | 141 |
| Anexo III. Manual de usuario (Experto)..... | 153 |

ÍNDICE DE FIGURAS

| | |
|--|----|
| Figura 1. Etapas de la Toma de Decisión..... | 7 |
| Figura 2. Transcendencia de las Decisiones | 15 |
| Figura 3. Proceso de TDG..... | 16 |
| Figura 4. Proceso de selección de alternativas | 17 |
| Figura 5. Tendencia 5 alternativas | 29 |
| Figura 6. Algoritmo Método TOPSIS | 31 |
| Figura 7. Actores (Casos de uso) | 47 |
| Figura 8. Sistema (Casos de uso) | 48 |
| Figura 9. Caso de uso Identificación_Y_Acceso | 49 |
| Figura 10. Caso de uso Gestión_ProblemaAdm | 50 |
| Figura 11. Caso de uso Gestión_ExpertoAdm | 52 |
| Figura 12. Caso de uso Gestión_Experto | 54 |
| Figura 13. Caso de uso Gestión_Alternativas | 56 |
| Figura 14. Caso de uso Gestión_Criterios..... | 58 |
| Figura 15. Esquema de la arquitectura MVC | 65 |
| Figura 16. Relación paquetes | 66 |
| Figura 17. Paquete controllers..... | 67 |
| Figura 18. Paquete models | 68 |
| Figura 19. Paquete dialogs | 69 |
| Figura 20. Paquete panels..... | 70 |
| Figura 21. Paquete table.models | 71 |
| Figura 22. Paquete utils..... | 72 |
| Figura 23. Paquete frames | 72 |
| Figura 24. Paquete tables.columns | 73 |
| Figura 25. Esquema Conceptual E-R | 76 |
| Figura 26. Esquema Conceptual Modificado E-R..... | 77 |
| Figura 27. Pantalla acceso usuario | 85 |
| Figura 28. Pantalla principal Administrador | 86 |
| Figura 29. Pantalla nuevo problema..... | 86 |
| Figura 30. Pantalla nueva alternativa | 87 |
| Figura 31. Pantalla nuevo criterio | 87 |
| Figura 32. Pantalla vincular experto | 88 |
| Figura 33. Pantalla modificar problema | 88 |
| Figura 34. Pantalla problema resuelto..... | 89 |
| Figura 35. Pantalla nuevo experto..... | 90 |
| Figura 36. Pantalla gestión experto | 90 |

| | |
|--|-----|
| Figura 37. Consulta experto información problema..... | 91 |
| Figura 38. Pantalla valoración experto..... | 91 |
| Figura 39. Ver solución problema experto..... | 92 |
| Figura 40. Pantalla cambio información experto | 92 |
| Figura 41. Pantalla cambio contraseña experto..... | 93 |
| Figura 42. Storyboard crear problema..... | 95 |
| Figura 43. Storyboard crear un experto..... | 96 |
| Figura 44. Storyboard resolver un problema..... | 97 |
| Figura 45. Storyboard cambiar contraseña experto..... | 98 |
| Figura 46. Storyboard valorar un problema | 99 |
| Figura 47. Consultar problema resuelto | 100 |
| Figura 48.1. Mensaje error acceso | 102 |
| Figura 48.2. Mensaje error acceso | 102 |
| Figura 49.1. Mensajes error crear o modificar problema | 102 |
| Figura 49.2. Mensajes error crear o modificar problema..... | 102 |
| Figura 49.3. Mensajes error crear o modificar problema | 102 |
| Figura 50.1. Mensaje error crear o modificar alternativa..... | 103 |
| Figura 50.2. Mensaje error crear o modificar alternativa..... | 103 |
| Figura 50.3. Mensaje error crear o modificar alternativa repetida..... | 103 |
| Figura 51.1. Mensaje error crear o modificar criterio | 103 |
| Figura 51.2. Mensaje error crear o modificar criterio | 104 |
| Figura 51.3. Mensaje error crear o modificar criterio repetido | 104 |
| Figura 52.1. Mensaje error crear o modificar experto..... | 104 |
| Figura 52.2. Mensaje error crear o modificar experto..... | 104 |
| Figura 52.3. Mensaje error crear o modificar experto..... | 105 |
| Figura 52.4. Mensaje error crear o modificar experto..... | 105 |
| Figura 52.5. Mensaje error crear o modificar experto..... | 105 |
| Figura 53. Mensaje error resolver problema | 105 |
| Figura 54.1. Mensaje error cambio información experto..... | 106 |
| Figura 54.2. Mensaje error cambio información experto..... | 106 |
| Figura 55.1. Mensaje error cambio de contraseña..... | 106 |
| Figura 55.2. Mensaje error cambio de contraseña..... | 106 |
| Figura 55.3. Mensaje error cambio de contraseña..... | 106 |
| Figura 55.4. Mensaje error cambio de contraseña..... | 107 |
| Figura 56. Mensaje éxito valorar problema..... | 107 |
| Figura 57. Mensaje confirmación eliminar problema | 108 |
| Figura 58. Mensaje confirmación eliminar experto | 108 |
| Figura 59. Mensaje confirmación resolver problema..... | 108 |

| | |
|--|-----|
| Figura I.1. Instalación WampServer | 133 |
| Figura I.2. Instalación WampServer | 134 |
| Figura I.3. Instalación WampServer | 134 |
| Figura I.4. Instalación WampServer | 135 |
| Figura I.5. Instalación WampServer | 135 |
| Figura I.6. Instalación WampServer | 136 |
| Figura I.7. Instalación WampServer | 136 |
| Figura I.8. Acceso phpMyAdmin..... | 137 |
| Figura I.9. Crear BBDD | 138 |
| Figura I.10. Importar BBDD | 139 |
| Figura II.1. Acceso usuario | 142 |
| Figura II.2. Pantalla principal..... | 143 |
| Figura II.3. Crear problema..... | 143 |
| Figura II.4. Nueva alternativa | 144 |
| Figura II.5. Nuevo criterio..... | 144 |
| Figura II.6. Vincular expertos | 145 |
| Figura II.7. Crear experto..... | 146 |
| Figura II.8. Pantalla principal 2..... | 146 |
| Figura II.9. Ver/Modificar problema..... | 147 |
| Figura II.10. Ver/Modificar alternativa..... | 147 |
| Figura II.11. Ver/Modificar criterio | 148 |
| Figura II.12. Valoración experto | 149 |
| Figura II.13. Mensaje eliminar problema..... | 149 |
| Figura II.14. Resolución problema..... | 150 |
| Figura II.15. Modificar experto..... | 151 |
| Figura II.16. Mensaje eliminar experto..... | 151 |
| Figura II.17. Cerrar sesión administrador | 152 |
| Figura III.1. Acceso usuario..... | 154 |
| Figura III.2. Pantalla principal experto | 155 |
| Figura III.3. Cambiar datos experto | 155 |
| Figura III.4. Cambiar contraseña experto..... | 156 |
| Figura III.5. Pantalla principal experto 2 | 156 |
| Figura III.6. Pestaña General..... | 157 |
| Figura III.7. Pestaña Valoración | 158 |
| Figura III.8. Pestaña Resolución | 158 |
| Figura III.9. Pestaña Valoración 2 | 159 |
| Figura III.10. Cerrar sesión experto | 160 |

ÍNDICE DE TABLAS

| | |
|---|-----|
| Tabla 1. Esquema general de un problema de toma de decisión con un único criterio.... | 10 |
| Tabla 2. Esquema general de un problema de toma de decisión multicriterio..... | 11 |
| Tabla 3. Esquema general de un problema de toma de decisión con un solo experto | 13 |
| Tabla 4. Esquema general de un problema de toma de decisión multiexperto | 14 |
| Tabla 5. Esquema general de un problema de toma de decisión multicriterio..... | 14 |
| Tabla 6. Teoría matriz de decisión..... | 31 |
| Tabla 7. Ejemplo matriz de decisión..... | 34 |
| Tabla 8. Ejemplo normalización matriz de decisión | 34 |
| Tabla 9. Ejemplo matriz normalizada ponderada..... | 34 |
| Tabla 10. Ejemplo matriz normalizada y SIP Y SIN | 35 |
| Tabla 11. Ejemplo matriz distancias \bar{A}^+ | 35 |
| Tabla 12. Ejemplo matriz distancias \bar{A}^- | 35 |
| Tabla 13. Ejemplo matriz solución | 35 |
| Tabla 14. Tabla usuario..... | 78 |
| Tabla 15. Tabla problemas..... | 78 |
| Tabla 16. Tabla alternativas | 79 |
| Tabla 18. Tabla criterios..... | 79 |
| Tabla 19. Tabla Experto_Problemas | 79 |
| Tabla 20. Tabla Experto_Alt_Crit..... | 80 |
| Tabla 21. Test autenticación usuario..... | 115 |
| Tabla 22. Test autenticación usuario incorrecta..... | 115 |
| Tabla 23. Test crear un problema..... | 115 |
| Tabla 24. Test crear un problema con nombre en blanco | 116 |
| Tabla 25. Test crear un problema con descripción en blanco | 116 |
| Tabla 26. Test crear un problema con nombre igual..... | 117 |
| Tabla 27. Test añadir nueva alternativa | 118 |
| Tabla 28. Test añadir nuevo criterio..... | 118 |
| Tabla 29. Test añadir nuevo experto | 118 |
| Tabla 30. Test modificar información experto..... | 119 |
| Tabla 31. Test eliminar problema | 119 |
| Tabla 32. Test resolver problema..... | 119 |
| Tabla 33. Test consultar mejor alternativa problema..... | 120 |
| Tabla 34. Test valorar problema | 120 |
| Tabla 35. Test consulta mejor alternativa experto | 121 |
| Tabla 36. Resultados obtenidos test realizados..... | 121 |

Capítulo 1

Introducción

1.1 Introducción al Proyecto

La Toma de Decisión es una tarea habitual de la vida cotidiana. Constantemente nos enfrentamos 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. Ejemplos típicos incluyen qué comprar, decidir dónde viajar o a quién votar en unas elecciones. En algunas ocasiones, en la toma de decisiones, aparecen diferentes alternativas de elección que se evalúan atendiendo a múltiples criterios, hablándose así de Toma de Decisión Multicriterio (TDMC). Sin embargo, la Toma de Decisión no sólo ocurre para individuos aislados. Muchos problemas de decisión tienen que ser resueltos por las valoraciones de un grupo de personas, normalmente expertos, que tienen que opinar o valorar de una determinada forma las diferentes alternativas de entre todas las posibles para concluir con la mejor de ellas. Este tipo de Toma de Decisión con múltiples individuos o expertos se denomina Toma de Decisión Multiexperto (TDME).

El proceso de Toma de Decisión en Grupo (TDG) consiste en que un conjunto de expertos o individuos expresan sus valoraciones sobre un conjunto de alternativas para obtener la mejor solución.

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. En este proyecto los expertos utilizarán el mismo dominio de información para expresar sus valoraciones.

Un proceso de resolución de TDG, de forma general se compone de dos etapas:

- Proceso de consenso
- Proceso de selección

Normalmente, los procesos de TDG han sido resueltos aplicando sólo procesos de selección, donde la solución se obtiene directamente a partir de las valoraciones de los expertos. Sin embargo, puede que algunos expertos consideren que sus valoraciones no han sido tenidas en cuenta para obtener la solución, y por tanto pueden estar en desacuerdo con ella. Para evitar esta situación, se lleva a cabo un proceso de consenso donde los expertos discuten y cambian sus valoraciones para alcanzar un mínimo acuerdo antes de realizar el proceso de selección que obtenga la solución del problema. Nosotros nos centraremos solamente en el proceso de selección usado por el método TOPSIS, y no realizamos el proceso de consenso.

En este Proyecto Fin de Carrera vamos a diseñar e implementar una aplicación de escritorio que desarrolle la metodología TOPSIS (Technique for Order of Preference by Similarity to Ideal Solution) para la resolución de problemas de toma de decisión multicriterio

multiexperto. Para ello se tendrá en cuenta las valoraciones de diferentes expertos, pudiendo éstos, valorar los criterios y alternativas de los diferentes problemas generados mediante valores numéricos. Finalmente se obtendrá un ranking de alternativas como solución del problema de decisión.

En el Proyecto utilizaremos diferentes tecnologías, que serán posteriormente definidas en este documento, para la implementación del proyecto; se considerará una Base de Datos (BBDD) MySQL para el almacenamiento y gestión de toda la información, así como, el lenguaje de programación JAVA, como lenguaje ampliamente extendido y de uso actualizado para el desarrollo de tales aplicaciones.

1.2 Propósito del Proyecto

El propósito de este proyecto de fin de carrera es la resolución de problemas de Toma de Decisión Multicriterio y Multiexperto mediante el método TOPSIS, siguiendo los objetivos que a continuación se definen.

1.3 Objetivos del Proyecto

Los objetivos que se pretenden alcanzar son los siguientes:

- Buscar y revisar fuentes bibliográficas.
- Analizar el modelo existente.
- Diseñar e implementar una base de datos que recoja toda la información generada en las valoraciones de los expertos y datos de los problemas.
- Diseñar y desarrollar el método TOPSIS para la selección de la mejor alternativa en cada problema.
- Diseñar e implementar una interfaz de usuario amigable y usable.
- Realizar manuales para el uso correcto de la aplicación.

1.4 Estructura de la memoria del Proyecto Fin de Carrera

A continuación, se describe la estructura que presenta el documento de la memoria.

En el capítulo 2 vamos a revisar los conceptos básicos sobre Toma de Decisión y diferentes métodos de TDMC, en especial el modelo TOPSIS.

En el capítulo 3 se describirá el proyecto en si con las diferentes fases que forman el ciclo de vida del software como son las especificaciones de requerimientos funcionales, como no funcionales, análisis del sistema, diseño del sistema, implementación y pruebas.

En el capítulo 4 se indican unas conclusiones finales junto con posibles mejoras en trabajos futuros.

Incluiremos un apartado de bibliografía que enumerará las referencias bibliográficas utilizadas en este proyecto y al final podemos encontrar tres anexos que explican el modo de proceder para la instalación de la aplicación y los manuales de utilización de la aplicación para los diferentes usuarios (Administrador y Experto).

Capítulo 2

Antecedentes y Definiciones de Teoría de Toma de Decisión

2.1 Antecedentes

2.1.1 Toma de decisión

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.

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.

Una situación de Toma de Decisión clásica presta los siguientes elementos básicos:

- Uno o varios objetivos por resolver.
- Un conjunto de alternativas o decisiones posibles para alcanzar dichos objetivos.
- Un conjunto de factores, criterios o estados de la naturaleza que definen el contexto en el que se plantea el problema de decisión.
- Un conjunto de valores de utilidad o consecuencias asociados a los pares formados por cada alternativa y criterio de la naturaleza.

Dependiendo de las características de los elementos del problema de decisión, éstos podrán clasificarse atendiendo a distintos puntos de vista. Por otro lado, el tipo de información que define el marco del problema y su modelado influirían también en el modelo de resolución del problema de decisión.

La Toma De Decisión está formada por las etapas que se enumeran a continuación como se puede ver en la Figura 1:



Figura 1. Etapas de la Toma de Decisión

1. Identificación y diagnóstico del problema

En esta fase identificamos el problema que deseamos resolver. En ella se debe recolectar la información del problema, es decir, se debe analizar los diferentes elementos que influyen en la resolución del problema que se está llevando a cabo para extraer la máxima cantidad de información que permita obtener una solución lo más precisa posible.

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:

- **FACTORES CUANTITATIVOS:** Son factores que se pueden medir en términos numéricos, como es el tiempo, o los diversos costos fijos o de operación.
- **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; ésta debe ser implementada. Bien puede ser que quienes participen en la toma de 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 éstos 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 ésto ocurriera, sin duda tendríamos más información y probablemente más sugerencias que nos ayudarían a evitar los errores cometidos en el primer intento.

Además de los elementos que forman parte de un problema de Toma de Decisión se deben tener en cuenta los siguientes factores que influyen en este tipo de problemas:

a) **Información:** Es fundamental recoger información a la hora de solucionar cualquier tipo de problema. Dicha información guiará el proceso de solución para obtener la alternativa que mejor se adapte a las características del problema que se está solucionando.

b) **Conocimientos:** Se trata de los conocimientos de las personas que serán las encargadas de tomar la decisión. Estos conocimientos pueden ser de circunstancias relacionadas con el problema o de situaciones similares.

c) **Experiencia:** Se corresponde con la experiencia que el experto adquiere al solucionar diferentes problemas a los que se enfrenta a lo largo del tiempo. Dicha experiencia puede ser de mucha utilidad para solucionar problemas parecidos en el futuro.

d) **Análisis:** Se trata de analizar los diferentes elementos que influyen en el problema para obtener toda la información posible de ellos, con el objetivo de obtener la mejor solución para resolver el problema planteado.

e) **Juicio:** Es necesario saber elegir entre las diferentes alternativas presentadas para obtener la mejor solución posible, para ello, habrá que evaluar los diferentes aspectos que influyen en el problema.

2.1.2 Criterios de Clasificación de los Problemas de Toma de Decisión

Ante la gran variedad de situaciones o problemas de decisión que se pueden presentar en la vida real, la teoría de la decisión ha establecido una serie de criterios que permiten clasificar los problemas de decisión atendiendo a diferentes puntos de vista:

A. Según el Número de Criterios

El número de criterios o atributos que se tienen en cuenta en los procesos de decisión para obtener la solución permite clasificar a los problemas de decisión en dos tipos:

- **Problemas con un solo criterio o atributo.**

Problemas de decisión en los que, para evaluar las alternativas, se tiene en cuenta un único criterio o atributo que representa la valoración dada a esa alternativa. La solución se obtiene como la alternativa que mejor resuelve el problema teniendo en cuenta este único criterio.

Supongamos un problema de decisión en el que nos planteamos cambiar de coche y nos ofrecen tres posibles alternativas, siendo el criterio de optimización de la decisión el criterio precio. Este problema de decisión sería muy simple de resolver puesto que escogeríamos la alternativa con el menor precio.

En los problemas de decisión de un único criterio, cada alternativa es caracterizada por un único valor. Sea $X = \{x_1, x_2, \dots, x_n\}$ el conjunto de alternativas del problema. Una forma de representación de la información del problema se muestra en la Tabla 1:

| Alternativa | Valoración |
|-------------|------------|
| x_1 | y_1 |
| ... | ... |
| x_n | y_n |

Tabla 1: Esquema general de un problema de toma de decisión con un único criterio

Cada entrada, y_i , de la tabla indica la valoración de la alternativa, x_i . Según el marco de definición del problema, cada, y_i , estaría valorada en un dominio de expresión determinado (numérico, lingüístico, etc.).

- **Problemas multicriterio o multiatributo.**

Problemas de decisión en los que, para evaluar las alternativas, se tienen en cuenta dos o más criterios o atributos que definen cada alternativa. La alternativa solución sería aquella que mejor resuelva el problema considerando todos estos criterios o atributos.

Supongamos el problema presentado anteriormente. Este se complicaría y el proceso para resolverlo sería diferente si, además de considerar el precio, también tuviésemos en cuenta otros criterios o atributos como el diseño y el consumo del coche.

En este caso, nos enfrentaríamos a un problema en el que se consideran varios criterios o atributos para tomar una decisión y, por lo tanto, hablamos de un problema de decisión multicriterio o multiatributo. El número de criterios en un problema de decisión multicriterio se asume que es finito. Sean $X = \{x_1, x_2, \dots, x_n\}$ y $C = \{c_1, c_2, \dots, c_h\}$ el conjunto de alternativas y el conjunto de criterios respectivamente, que caracterizan una situación de decisión determinada, entonces una representación de la información del problema puede expresarse mediante la Tabla 2:

| Alternativas | Criterios | | | |
|--------------|-----------|----------|-----|----------|
| | c_1 | c_2 | ... | c_h |
| x_1 | y_{11} | y_{12} | ... | y_{1h} |
| ... | ... | ... | ... | ... |
| x_n | y_{n1} | y_{n2} | ... | y_{nh} |

Tabla 2: Esquema general de un problema de toma de decisión multicriterio

Cada entrada, y_{ij} , indica la preferencia de la alternativa, x_i , respecto del criterio, c_j . Los problemas de toma de decisión multicriterio son más complejos de resolver que los problemas en los que hay un solo criterio. Cada criterio puede establecer un orden de preferencia particular y diferente sobre el conjunto de alternativas. A partir del conjunto de órdenes de preferencia particulares, será necesario establecer algún mecanismo que permita construir un orden global de preferencia. En la literatura, podemos encontrar varios ejemplos.

B. Según el Ambiente de 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***

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 qué 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, como la Teoría de los Conjuntos Difusos, para modelar la incertidumbre.

C. Según el Número de Expertos

Finalmente, otro punto de vista, a la hora de clasificar los problemas de decisión, hace referencia al número de expertos o fuentes de información que toman parte en el problema de

decisión. Un problema de toma de decisión, en el que participan varios expertos, es más complejo que otro en el que la toma de decisión se realiza de forma individual. Sin embargo, el hecho de que intervengan varios expertos con puntos de vista diferentes puede ofrecer una solución más satisfactoria al problema.

Atendiendo al número de expertos que toman parte en el proceso de toma de decisión, los problemas de decisión se pueden clasificar en dos tipos:

- ***Unipersonales o individuales.***

Las decisiones son tomadas por un único experto. En los problemas de decisión unipersonales o individuales, cada alternativa es valorada por un único experto. Sea $X = \{x_1, x_2, \dots, x_n\}$ el conjunto de alternativas que son valoradas por el experto, "e", entonces una forma de representación de la información del problema es la mostrada en la siguiente Tabla 3:

| Alternativas | Experto |
|--------------|---------|
| (x_i) | e |
| x_1 | y_1 |
| ... | ... |
| x_n | y_n |

Tabla 3: Esquema general de un problema de toma de decisión con un solo experto

Cada entrada, y_i , de la tabla, indica la valoración dada por el experto, e , sobre la alternativa, x_i . Según el marco de definición del problema, cada y_i estará valorada en un dominio de expresión determinado (numérico, lingüístico, etc.).

- ***En Grupo o Multiexperto.***

Las decisiones son tomadas en conjunto por un grupo de expertos que intentan alcanzar una solución, en común, al problema. El número de expertos en problemas de decisión multiexperto se asume que es finito. Sean $X = \{x_1, x_2, \dots, x_n\}$ y $E = \{e_1, e_2, \dots, e_m\}$ el conjunto de alternativas y el conjunto de expertos respectivamente, que valoran cada alternativa que caracteriza una situación de decisión determinada, entonces una forma de representación de la información del problema puede verse en la siguiente Tabla 4:

| Alternativas | Expertos | | | |
|--------------|----------|----------|-----|----------|
| | e_1 | e_2 | ... | e_m |
| x_1 | y_{11} | y_{12} | ... | y_{1m} |
| ... | ... | ... | ... | ... |
| x_n | y_{n1} | y_{n2} | ... | y_{nm} |

Tabla 4: Esquema general de un problema de toma de decisión multiexperto

Cada entrada, y_{ij} , de la tabla indica la preferencia del experto, e_j , sobre la alternativa, x_i . Una situación de decisión habitual son los problemas de decisión multiexperto multicriterio, en los que cada experto expresa las preferencias sobre distintos criterios que definen cada alternativa. Estas situaciones están relacionadas, entre otras, con los procesos de evaluación sensorial, donde un conjunto de jueces expresan sus valoraciones sobre las características sensoriales de un producto y es necesario fusionar todas ellas para obtener la valoración global del producto. Lo vemos en las siguientes Tablas 5.

| Experto (e_1) | Criterios | | | |
|-------------------|------------|------------|-----|------------|
| Alternativas | c_1 | c_2 | ... | c_h |
| x_1 | y_{11}^1 | y_{12}^1 | ... | y_{1h}^1 |
| ... | ... | ... | ... | ... |
| x_n | y_{n1}^1 | y_{n2}^1 | ... | y_{nh}^1 |

| Experto (e_m) | Criterios | | | |
|-------------------|------------|------------|-----|------------|
| Alternativas | c_1 | c_2 | ... | c_h |
| x_1 | y_{11}^m | y_{12}^m | ... | y_{1h}^m |
| ... | ... | ... | ... | ... |
| x_n | y_{n1}^m | y_{n2}^m | ... | y_{nh}^m |

Tabla 5: Esquema general de un problema de toma de decisión multicriterio

Cada experto, e_j , proporciona la entrada de la tabla, y_{jk}^j , que indica la preferencia de la alternativa, x_j , respecto del criterio, c_k .

En este proyecto nos centraremos en la resolución de un problema de toma de decisión caracterizado por ser: multicriterio, en un ambiente de incertidumbre y multiexperto.

2.1.3 Esquema de Resolución de problemas de Toma de Decisión en Grupo

Tomar decisiones en grupo (TDG), 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, ver Figura 2.

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.



Figura 2. Trascendencia de las Decisiones

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.

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

- 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.

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

En la literatura podemos encontrar muchos trabajos donde se aborda la resolución de problemas de TDG aplicando procesos de selección de alternativas. 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 [19]. 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. Posteriormente se realizará el proceso de selección, para que, aquellas opiniones consensuadas como mejores se clasifiquen para dar la mejor decisión posible. Se puede ver en la Figura 3.

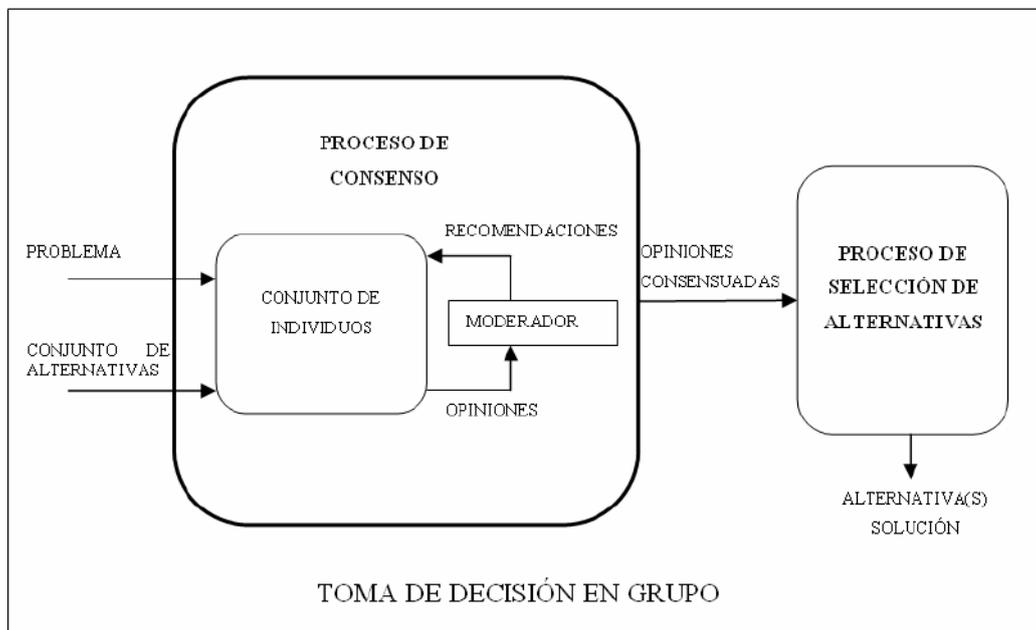


Figura 3. Proceso de TDG

A continuación se describen brevemente ambos procesos.

a) *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.

b) *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 4. Proceso de selección de alternativas

En los problemas de TDG, tradicionalmente el proceso de selección ha estado compuesto por las dos fases que se muestran en la Figura 4.

- ***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 [9, 10, 11, 12, 13, 14, 15, 16]. 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.

- **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:

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.

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 vectores de preferencia o vectores de utilidad que se mencionarán más adelante en esta memoria.

2.1.4 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 [18].

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, o bien problemas en los que los expertos utilizan dominios de información diferentes, conocidos como problemas definidos en contextos heterogéneos [17].

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.

Dentro del contexto de los problemas de decisión se entiende por dominio de expresión al conjunto de valores utilizado por los diferentes expertos para realizar las distintas valoraciones, es decir, para expresar sus preferencias. La elección del dominio adecuado al problema que se desea resolver puede deberse a varios motivos [7]:

- 1- **Naturaleza cuantitativa o cualitativa de la información:** La naturaleza del atributo puede condicionar el dominio utilizado. De esta forma, dominios numéricos se adecuan más a atributos cuantitativos, mientras que para atributos cualitativos es más recomendable usar dominios lingüísticos.

- 2- ***Pertenencia de los expertos a diferentes áreas de conocimiento:*** Los expertos tenderán a utilizar el dominio de información que más se adapte a su área de trabajo, ya que será el que maneje mejor y con el que estará más familiarizado.
- 3- ***Expertos con diferente grado de conocimiento sobre el problema:*** La experiencia de los expertos en resolución de problemas similares puede llevar a elegir determinados dominios que le hayan aportado mejores soluciones en el pasado o descartar otros que le hubiesen proporcionado malos resultados.

En la literatura encontramos [22] 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. Nosotros trabajaremos con el Dominio Numérico para representar la información en nuestra aplicación.

2.1.4.1 Dominio Numérico

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

- ***Numérico Binario.*** Se caracteriza por utilizar exclusivamente dos valores para cuantificar la utilidad de cada una de las alternativa. Normalmente se utilizan los valores $\{0, 1\}$, donde el 0 representa la valoración negativa de la alternativa y el 1 representa la valoración positiva.

Un ejemplo:

Sea $E = \{e_1, \dots, e_m\}$ ($m \geq 2$) un conjunto finito de expertos que han de expresar sus preferencias sobre un conjunto finito de alternativas $X = \{x_1, \dots, x_n\}$ ($n = 4$). Cada experto i utiliza un vector de utilidades definido $U^i = \{u_1^i, \dots, u_n^i\}$ para expresar sus preferencias, donde a cada alternativa de X se le asigna un valor del dominio binario, $u_j^i \in \{0, 1\}$. Los valores dados por e_1 y e_2 pueden ser los siguientes:

$$U^1 = \{1, 0, 0, 1\}$$

$$U^2 = \{0, 0, 1, 0\}$$

donde las alternativas x_1 y x_4 son valoradas positivamente por el experto 1 y las alternativas x_1 , x_2 y x_4 reciben una valoración negativa por parte del experto 2.

- **Numérico normalizado en el intervalo [0, 1].** Los expertos utilizan un valor dentro del intervalo [0, 1] para modelar la preferencia sobre cada alternativa. Se pueden utilizar valores reales dentro de este intervalo que permiten establecer un orden de preferencia entre las distintas alternativas en función de la utilidad asignada a cada una de ellas.

Un ejemplo:

Sea $E = \{e_1, \dots, e_m\}$ ($m \geq 2$) un conjunto finito de expertos que han de expresar sus preferencias sobre un conjunto finito de alternativas $X = \{x_1, \dots, x_n\}$ ($n = 4$). Cada experto i utiliza un vector de utilidades definido $U^i = \{u_1^i, \dots, u_n^i\}$ para expresar sus preferencias, donde cada alternativa X utiliza un dominio de expresión dentro del intervalo [0,1] y los vectores de utilidad podrían ser los siguientes:

$$U^1 = \{1, 0.2, 0, 0.6\}$$

$$U^2 = \{0, 0.4, 0.7, 0.9\}$$

Podemos comprobar cómo el experto 1 considera que la alternativa x_1 es la mejor y le asigna una utilidad máxima. Por el contrario, considera la alternativa x_3 peor que la alternativa x_2 asignándole una utilidad de 0 y 0.2 respectivamente. Para el experto 2, la mejor alternativa sería x_4 y la peor x_1 .

2.1.4.2 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.

2.1.4.3 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. 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.

2.1.4.4 Estructuras de representación de preferencias

Una vez explicados los dominios de expresión, se va a revisar brevemente las estructuras más comunes utilizadas en los problemas de decisión para representar las preferencias de los distintos expertos:

- 1- **Órdenes de preferencia:** Este formato de representación de preferencias establece un ranking u orden de alternativas que representa la idoneidad de cada una de ellas, como solución al problema de decisión según el punto de vista de cada experto, es decir, cada experto proporciona un vector de alternativas ordenándolas de mejor a peor.
- 2- **Vectores de preferencia o de utilidad:** Esta forma ha sido ampliamente utilizada. En este caso, las preferencias de un experto sobre un conjunto de alternativas se describe como un conjunto de valores de utilidad dentro de un dominio D . Este formato de representación es más fino que el anterior, pero es necesario que cada experto valore de manera global cada alternativa con respecto a las demás.

Ésta es la estructura que nosotros utilizaremos en nuestra aplicación para representar las preferencias de los expertos.

Un vector de utilidad es una estructura de representación de información ampliamente utilizada en la literatura clásica [34, 35] para representar las preferencias de los expertos. Es una estructura muy simple basada en un vector donde cada elemento se interpreta como la preferencia o utilidad de una de las alternativas del problema.

Sea $E = \{e_1, \dots, e_m\}$ ($m \geq 2$) un conjunto finito de expertos que han de expresar sus preferencias sobre un conjunto finito de alternativas $X = \{x_1, \dots, x_n\}$ ($n \geq 2$). Las preferencias dadas por los expertos sobre un conjunto de alternativas X utilizando vectores de utilidades U^i serían las siguientes:

$$U^i = \{u_1^i, \dots, u_n^i\}$$

donde u_j^i representa la utilidad o valoración dada por el experto i a la alternativa j . Se asume que cuanto mayor sea el valor de u_j^i , más satisface la alternativa j el objetivo del problema según la opinión del experto i .

- 3- **Relaciones de preferencia:** Este formato de representación establece relaciones binarias entre las distintas alternativas de la forma:

$$x_1 R x_2 \leftrightarrow "x_1 \text{ no es peor que } x_2".$$

Esta definición considera una relación binaria como una relación de preferencia débil e implica que dicha relación es reflexiva, dicha relación refleja en cierto sentido el grado de preferencia de una alternativa sobre otra cualquiera.

2.1.5 Métodos de Decisión Multicriterio Discretos

Los métodos de Decisión Multicriterio son poderosas herramientas que ayudan a generar consenso y selección en contextos complejos de decisión. Se pueden aplicar estas técnicas a casos en los que sea necesaria la confluencia de intereses y puntos de vista de

diferentes grupos o personas, o sea necesaria la selección por mejor valoración de una alternativa. Esto permite que todas las partes interesadas participen en el proceso de toma de decisiones [2, 3].

"Considerando un grupo formado por varios agentes, cada uno con sus propias preferencias, como único agente colectivo que tiene varios criterios de elección diferentes, recaemos en la problemática multicriterio" [20].

En el presente apartado se va a considerar el problema de la toma de decisión multicriterio cuando el conjunto de alternativas factibles es numerable. En este caso se dice que el problema es un *problema de decisión multicriterio discreto*. En adelante se supondrá que el número de alternativas no es muy elevado y que además se conocen explícitamente.

Los problemas multicriterio se clasifican en dos categorías, los de programación múltiple objetivo y los de evaluación multicriterio. En nuestro caso, nos vamos a centrar en los de la segunda categoría.

Los Métodos de Decisión Multicriterio son Discretos cuando las alternativas de decisión son finitas. Sin embargo, para que los Métodos de Decisión Multicriterio Discretos se apliquen en el mundo real, es necesario que tengan una solidez teórica, pero sobre todo, es imprescindible que puedan ser comprendidos por los decisores y aplicados por ellos de forma fácil. La experiencia demuestra que las técnicas conceptualmente más fáciles de entender y de aplicar son más utilizadas que aquellas más abstractas teóricamente. Puede llegar el caso en el que no se empleen técnicas multicriterio porque el decisor no acaba de comprenderlas y siente que el proceso de toma de decisiones se le escapa de las manos. El decisor tiene que sentir que controla el proceso en todo momento.

2.1.5.1 Escuela Americana

En 1944 Von Neumann y Morganstern [21] axiomatizaron la teoría de utilidad esperada y así asentó las bases de la Teoría de la Utilidad Multiatributo (MAUT), como aplicación en econometría.

Por un lado, los métodos integrados dentro de la Teoría de la Utilidad Multiatributo. Estos métodos parten del supuesto de que el decisor trata de maximizar una función de utilidad que agrega los distintos criterios que intervienen en el problema. Cuando el problema es discreto y no existe una situación de incertidumbre, esta función se denomina *función valor*.

MAUT asume que un problema de decisión puede modelizarse mediante funciones valoradas reales que pueden ser maximizadas/minimizadas entre las alternativas.

Los métodos basados en la función de valor consisten en construir una función (v) que asocia un número real a cada una de las alternativas posibles. Este número refleja el valor o la utilidad que cada alternativa tiene para el decisor.

La principal dificultad de estos métodos consiste precisamente en encontrar dicha función de valor, pero una vez obtenida, el problema de decidir la mejor de las alternativas se reduce a obtener el máximo/mínimo de todos los valores calculados.

Basados en la existencia de la función valor, la Escuela Americana propone varios métodos prácticos, como son los métodos de la suma ponderada, el método de las Jerarquías Analíticas o Proceso Analítico Jerárquico (AHP), que según sus autores constituyen una teoría, y finalmente el método SMART (*The Simple Multi-attribute Rating Technique*).

2.1.5.1.1 Método de la Suma Ponderada

El método de la suma ponderada asume que la función valor buscada se puede descomponer y asimilar a un modelo aditivo, es decir, presentarse de la forma:

$$v = \lambda_1 \cdot v_1 + \lambda_2 \cdot v_2 + \dots + \lambda_n \cdot v_n$$

Los datos de partida del método son los expresados en la matriz de valoración (x_{ij}) de forma que se evalúa, para cada alternativa, el grado de cumplimiento de cada uno de los criterios. Se supone que los juicios que evalúan cada alternativa según cada criterio admiten representaciones numéricas sobre una escala de valores reales.

Una vez obtenida la matriz de valoración, ésta debe ser normalizada, de forma que los valores de los criterios, generalmente expresados en escalas distintas, se puedan comparar y no se produzcan sesgos.

Una vez obtenidos los valores normalizados r_{ij} para cada alternativa A_i , y conocidos los pesos w_j asociados a cada uno de los criterios que se consideran, el método de la suma ponderada construye la función valor de la siguiente forma:

$$v(A_i) = \frac{\sum_{j=1}^n (w_j \cdot r_{ij})}{\sum_{j=1}^n w_j} ; i = 1, 2, \dots, m$$

Donde $v(A_i)$ es un valor promedio ponderado para cada alternativa A_i , que refleja el valor que cada alternativa tiene para el decisor. Así pues, con la ordenación de las alternativas

en base a los valores $v(A_1)$, $v(A_2)$,..., $v(A_m)$ es posible resolver el problema de decisión y determinar la mejor alternativa de entre las posibles, que será la de suma ponderada mayor/menor.

2.1.5.1.2 El Proceso Analítico Jerárquico

Desarrollado por Thomas Saaty en 1980 [22], consiste esencialmente en formalizar nuestra comprensión intuitiva de problemas complejos utilizando una estructura jerárquica. El propósito del Proceso Analítico Jerárquico es permitir que el decisor pueda estructurar un problema multicriterio en forma visual, dándole la forma de una jerarquía de atributos, la cual contendría mínimamente tres niveles: el propósito u objetivo global del problema, ubicado en la parte superior, los varios criterios que definen las alternativas en el medio, y las alternativas concurrentes en la parte inferior del diagrama. En la medida que los criterios sean muy abstractos, tal como bienestar humano, o capacidad, por ejemplo, pueden incluirse sub-criterios más operativos en forma secuencial entre el nivel de los criterios y el de las alternativas, lo que da origen entonces a una jerarquía multinivel.

2.1.5.1.3 Método SMART

El método SMART fue desarrollado por Edwards y Barron [23], donde se juzga la actuación de la alternativa mediante la elección de un apropiado valor entre un límite inferior predeterminado para la peor (real o imaginaria) alternativa y un límite superior para la mejor (real o ideal) alternativa.

La ventaja del modelo SMART es que es independiente de las alternativas, Lootsma y Schuijt [24] . Cuando las evaluaciones de las alternativas no son relativas, cambiando el número de alternativas considerado, no cambiarán las puntuaciones de la decisión con respecto a las alternativas. Esta característica es particularmente útil cuando se añaden nuevas alternativas a la comparación existente.

2.1.5.2 Escuela Europea

La Escuela Europea, con el trabajo de B. Roy en los años 70 [25] y la contribución de varios científicos europeos, fue la fundadora de la metodología de Ayuda a la Decisión Multicriterio por reflejar una actitud dentro de la línea del pensamiento constructivista. Esta familia de métodos persigue ayudar al decisor a resolver el problema teniendo en cuenta las dificultades que se derivan para la construcción de la función de valor.

2.1.5.2.1 Método ELECTRE

Familia de métodos basados en relaciones de superación para decidir acerca de la determinación de una solución, que sin ser óptima pueda considerarse satisfactoria; además de obtener una jerarquización de las acciones, alternativas bajo análisis [25, 26]. Originada inicialmente y desarrollada por la escuela francófona (principalmente en Francia, Bélgica, Suiza, aunque pueden considerarse continental, ya que se verifican muy importantes contribuciones de los Países Bajos y Polonia, entre otros, a tal esquema), en la actualidad han sido desarrollados los procedimientos ELECTRE I, II, III IV, IS y ELECTRE TRI, los que brindan procedimientos para resolver diferentes tipos de problemas suscitados en el tratamiento de la teoría de la decisión.

2.1.5.2.2 Método PROMETHEE

Este método (Brans y Vincke, 1985) consiste [27], como en ELECTRE III, en la construcción de relaciones de superación valorizadas, incorporando conceptos y parámetros que poseen alguna interpretación física o económica fácilmente comprensibles por el decisor. PROMETHEE hace uso abundante del concepto de pseudocriterio ya que construye el grado de superación entre cada par de acciones ordenadas, tomando en cuenta la diferencia de puntuación que esas acciones poseen respecto a cada atributo. La valoración de esas diferencias pueden realizarse mediante funciones de valor posibles y que son utilizadas de acuerdo a las preferencias del decisor, quien además debe proporcionar los umbrales de indiferencia y de preferencia asociados a estos pseudocriterios.

Otras variantes del método plantean situaciones más sofisticadas de decisión, en particular problemas con un componente estocástico. Así se han desarrollado las versiones Promethee II, Promethee III, Promethee IV y Promethee V. En Promethee V (Brans y Mareschal, 1992) [28], se incorpora una filosofía de optimización entera a efectos de abordar problemas de selección de inversiones con restricciones presupuestarias.

2.1.5.2.3 Método MACBETH (Measuring Attractiveness by a Categorical Based Evaluation Technique)

MACBETH es un método interactivo que mide el grado de preferencia de un decisor sobre un conjunto de alternativas. Fue desarrollado por Bana et al [29]. Construye una función criterio desde un punto de vista fundamental y determina los parámetros relacionados con la información entre criterios (pesos) en la fase de agregación.

El método utiliza un procedimiento mediante un cuestionario inicial iterativo que compara dos niveles al mismo tiempo, requiriendo solamente un juicio de preferencia cualitativo. Empieza con la comparación de la opción más atractiva y la menos atractiva. La opción más atractiva se compara entonces con el resto de opciones y el siguiente paso considera la comparación de la segunda opción más atractiva con la tercera, y así con todas.

2.1.5.3 Método ZAPROS

La técnica ZAPROS fue desarrollado por Larichev y Moshkovich [30, 31] utiliza evaluaciones cualitativas mediante las que el decisor elabora ordenes parciales en un posible gran conjunto de alternativas, pero el método solo proporciona un orden parcial y no garantiza una ordenación completa.

El método está basado en operaciones psicológicamente validas para la elicitación de información de un decisor: comparaciones de dos distancias entre las evaluaciones en las escalas ordinales de dos criterios. La información recibida por el decisor se utiliza para la construcción de relaciones binarias entre un par de alternativas que producen relaciones de preferencia, indiferencia o incompatibilidad.

2.1.5.4 Método TOPSIS

Éste es el método principal de nuestro sistema. Por tanto, será explicado en más detalle en el siguiente apartado.

2.2 MÉTODO TOPSIS

2.2.1 El concepto de Alternativa Ideal

El concepto de alternativa ideal tiene una larga tradición en diversos campos científicos, especialmente en la literatura psicométrica, en donde se maneja una noción absoluta de ideal. Pero es Zeleny [6] quien lo erige en pieza central de su propuesta de *solución de compromiso*, en el sentido de la alternativa más *próxima* al ideal. Para Zeleny el concepto de ideal y de solución de compromiso incluso, es una hipótesis sobre la racionalidad subyacente en los procesos humanos de decisión. El concepto se ha difundido desde entonces, pasando a otros campos como el de las decisiones en grupo [4].

Pero es en el método TOPSIS, Technique for order Preference by Similarity to Ideal Solution (Técnica para ordenar preferencias por similitud a la solución ideal), donde se contemplan las sutilezas que el concepto de ideal tiene y se construye un método operativo [34].

Formalicemos algunas definiciones básicas. Partimos de tener unas alternativas A_i , donde $i = 1, 2, \dots, m$ y una matriz de decisión, con $x_{ij} = U_j(A_i)$, $j = 1, 2, \dots, n$. Donde U es la función utilidad del decisor. Podemos, sin pérdida de generalidad, transformar las utilidades de manera que todos los criterios sean a maximizar/minimizar y que todos los $x_{ij} > 0$.

Definición 1: Se denomina *punto ideal* en (\mathbb{R}^n) al punto $A^+ = (A_1^+, A_2^+, \dots, A_n^+)$, donde $A_i^+ = \text{Max}_i x_{ij}$ para el caso de criterios de beneficio y $A_i^- = \text{Min}_i x_{ij}$ para el caso de criterios de coste. La alternativa A^+ se llama *alternativa ideal*.

Definición 2: Se denomina *punto anti-ideal* en (\mathbb{R}^n) al punto $A^- = (A_1^-, A_2^-, \dots, A_n^-)$, donde $A_i^- = \text{Min}_i x_{ij}$ para el caso de criterios de beneficio y $A_i^+ = \text{Max}_i x_{ij}$ para el caso de criterios de coste. La alternativa A^- se llama *alternativa anti-ideal*.

2.2.2 El Método TOPSIS

El método TOPSIS fue desarrollado por Hwang y Yoon en el año 1981 [32] y se basa en el concepto de que es deseable que una alternativa determinada se ubique a la menor distancia respecto de una alternativa ideal que representa lo mejor (ideal positiva o simplemente ideal), y a la mayor distancia respecto a una alternativa ideal que representa lo peor (ideal negativa o anti-ideal).

Para ver que ésto es realmente un dilema, pues puede conducir a resultados diferentes, basta observar la siguiente Figura 5, en la que se han representado cinco alternativas (A, B, C, D y E) para un problema de dos criterios. También aparecen en la figura los puntos ideal y anti-ideal, respecto a los que es inmediato observar que C es la más próxima al ideal mientras que D es la más lejana del anti-ideal (en ambos casos utilizando una métrica de distancia euclídea y pesos iguales).

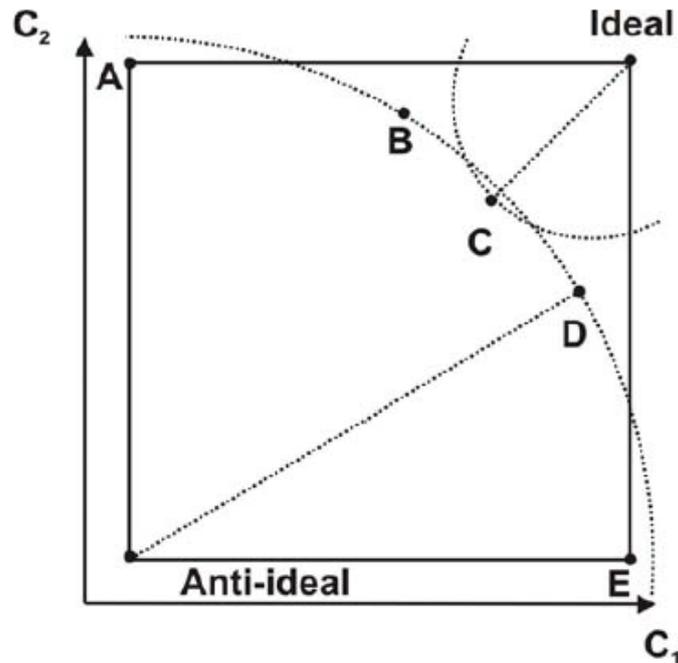


Figura 5. Tendencia 5 alternativas

TOPSIS resuelve el problema, inspirándose en una idea que Dasarathy [33] aplicó en un contexto de análisis multivariante de datos. Para cada alternativa $A_i = (x_{i1}, x_{i2}, \dots, x_{in})$, se calculan $d_p^+(A_i)$ y $d_p^-(A_i)$, las distancias ponderadas al ideal y al anti-ideal según la métrica p escogida:

$$d_p^+(A_i) = \left[\sum_j w_j^p |x_{ij} - A_j^+|^p \right]^{1/p}$$

$$d_p^-(A_i) = \left[\sum_j w_j^p |x_{ij} - A_j^-|^p \right]^{1/p}$$

Donde p , es el parámetro de distancia, y w , el peso de importancia. Los valores más utilizados son $p=1$, $p=2$ y $p=\infty$.

Para $p=1$, se obtiene la denominada distancia ciudad:

$$d^+(A_i) = \sum_j w_j |x_{ij} - A_j^+|^2$$

Para $p=2$, se obtiene la distancia euclídea:

$$d_p^+(A_i) = \left[\sum_j w_j^p |x_{ij} - A_j^+|^2 \right]^{1/2}$$

Para $p=\infty$, la distancia resulta:

$$d^+(A_i) = \text{Max}_j (w_j |x_{ij} - A_j^+|)$$

A partir de las distancias ponderadas al ideal, $d_p^+(A_i)$, y al anti-ideal, $d_p^-(A_i)$, se determina lo que se denomina “ratio de similaridad” al “ideal”:

$$D_p(A_i) = \frac{d_p^-(A_i)}{d_p^+(A_i) + d_p^-(A_i)}$$

Que varía desde $D_p(A^-) = 0$ para la solución anti-ideal, hasta $D_p(A^+) = 1$ para la solución ideal.

Finalmente, $D_p(A_i)$ se utiliza para la ordenación final de las alternativas, según las distancias que en él se reflejan hacia la solución ideal o ideal positiva y la solución anti-ideal o ideal negativa.

Es importante aclarar el sentido de “ratio de similaridad”. En rigor, similaridad es un abuso de la lengua, sería de similitud. Implica lo que se puede denominar “proximidad relativa” respecto a la alternativa ideal, combinando la proximidad a la solución “ideal” y la lejanía a la solución “anti-ideal”, seleccionándose luego la alternativa que se ubica lo más cerca posible a la máxima similaridad respecto a la ideal (de todos modos, tanto la ideal como la anti-ideal son virtuales).

2.2.3 Algoritmo del Método TOPSIS

El método TOPSIS consta de los siguientes pasos para su realización. Ver la Figura 6 [5, 7].

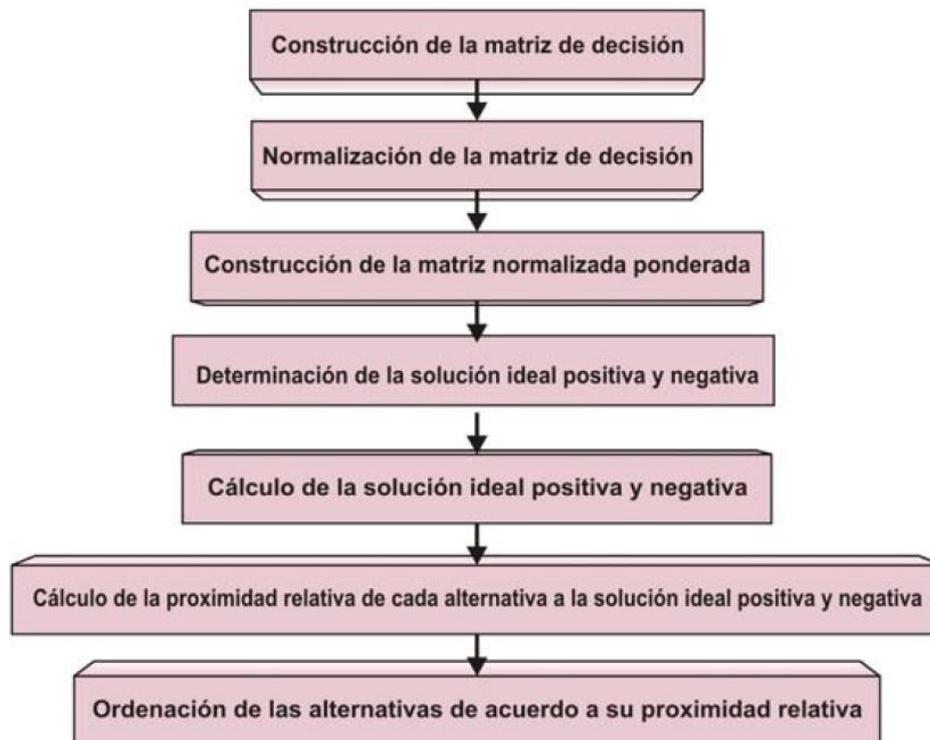


Figura 6. Algoritmo Método TOPSIS

Paso 1: Establecimiento de la matriz de decisión.

El método TOPSIS evalúa la siguiente matriz de decisión que se refiere a m alternativas $A_i, i = 1, \dots, m$, las cuales son evaluadas en función de n criterios $C_j, j = 1, \dots, n$.

Matriz de decisión

| | w_1 | w_2 | ... | w_j | ... | w_n |
|-------|----------|----------|-----|----------|-----|----------|
| | C_1 | C_2 | ... | C_j | ... | C_n |
| A_1 | x_{11} | x_{12} | ... | x_{1j} | ... | x_{1n} |
| A_2 | x_{21} | x_{22} | ... | x_{2j} | ... | x_{2n} |
| ... | ... | ... | ... | ... | ... | ... |
| A_m | x_{m1} | x_{m2} | ... | x_{mj} | ... | x_{mn} |

Tabla 6: Teoría matriz de decisión

Donde x_{ij} denota la valoración de la i -ésima alternativa en términos del j -ésimo criterio. Y donde $W = [w_1, w_2, \dots, w_n]$ es el vector de pesos asociado con C_j .

Paso 2: Normalización de la matriz de decisión.

El método TOPSIS primero convierte las dimensiones de los distintos criterios en criterios no dimensionales. Un elemento \bar{n}_{ij} de la matriz de decisión normalizada definida $N = [\bar{n}_{ij}]_{m \times n}$ se calcula como sigue:

$$\bar{n}_{ij} = \frac{x_{ij}}{\sqrt{\sum_{i=1}^m (x_{ij})^2}}, j = 1, \dots, n; i = 1, \dots, m$$

Paso 3: Construir la matriz de decisión normalizada ponderada.

El valor normalizado ponderado \bar{v}_{ij} de la matriz de decisión normalizada ponderada $N = [\bar{v}_{ij}]_{m \times n}$ se calcula como:

$$\bar{v}_{ij} = w_j \otimes \bar{n}_{ij}, j = 1, \dots, n, i = 1, \dots, m$$

(Producto de cada w_j por cada n_{ij})

Donde, w_j es el peso del j -ésimo criterio, tal que $\sum_{j=1}^n w_j$ puede ser igual a 1, si hablamos del caso general que verifica la igualdad. Se puede decir que los pesos de los criterios, en un problema de decisión, no tienen el mismo significado y no todos tienen la misma importancia. Estos pesos pueden obtenerse de diferentes modos: mediante asignación directa, mediante el método AHP, etc.

Paso 4: Determinar la solución ideal positiva (SIP) y la solución ideal negativa (SIN).

El conjunto de valores ideal positivo \bar{A}^+ y el conjunto de valores ideal negativo \bar{A}^- se determina como:

$$\bar{A}^+ = \{\bar{v}_1^+, \dots, \bar{v}_n^+\} = \{(max_i \bar{v}_{ij}, j \in J)(min_i \bar{v}_{ij}, j \in J')\} i = 1, 2, \dots, m$$

$$\bar{A}^- = \{\bar{v}_1^-, \dots, \bar{v}_n^-\} = \{(min_i \bar{v}_{ij}, j \in J)(max_i \bar{v}_{ij}, j \in J')\} i = 1, 2, \dots, m$$

Donde J está asociado con los criterios de beneficio, atributos deseables, y J' está asociado con los criterios de costes, atributos indeseables.

Paso 5: Cálculo de las medidas de distancia.

La separación de cada alternativa de la solución ideal positiva \bar{A}^+ está dada como:

$$\bar{d}_i^+ = \left\{ \sum_{j=1}^n (\bar{v}_{ij} - \bar{v}_j^+)^2 \right\}^{1/2}, i = 1, \dots, m$$

Y la separación de cada alternativa de la solución ideal negativa \bar{A}^- es como sigue:

$$\bar{d}_i^- = \left\{ \sum_{j=1}^n (\bar{v}_{ij} - \bar{v}_j^-)^2 \right\}^{1/2}, i = 1, \dots, m$$

En este caso se utiliza la distancia Euclídea m-multidimensional.

Paso 6: Cálculo de la proximidad relativa a la solución ideal.

La proximidad relativa \bar{R}_i a la solución ideal puede expresarse como sigue:

$$\bar{R}_i = \frac{\bar{d}_i^-}{\bar{d}_i^+ + \bar{d}_i^-}, i = 1, \dots, m$$

- Si $\bar{R}_i = 1 \rightarrow A_i = \bar{A}^+$. Solución ideal
- Si $\bar{R}_i = 0 \rightarrow A_i = \bar{A}^-$. Solución anti-ideal

Cuanto más próximo es el valor de \bar{R}_i a 1, implica una mayor prioridad de la alternativa i-ésima.

Paso 7: Ordenación de preferencias.

Se ordenan las mejores alternativas de acuerdo con \bar{R}_i en orden descendente.

2.2.4 Un ejemplo Método TOPSIS

Supóngase que hay tres candidatos para ocupar un determinado cargo y debemos seleccionar uno de ellos. Con tal fin, a cada uno de ellos se le han realizado dos cuestionarios para su evaluación. Cada uno de los cuestionarios tiene el mismo peso $w_1 = w_2 = 0,5$, por tanto $\sum w = 1$. De acuerdo con esto, se supone un problema multicriterio con tres alternativas y 2 criterios.

Sean los tres candidatos A_1 , A_2 y A_3 , los que obtuvieron en cada evaluación las valoraciones que se presentan en la Tabla 7, las que provienen de una escala numérica 0,1, 2, 3, 4, 5, donde $5 > 4 > 3 > 2 > 1 > 0$.

| | C₁ | C₂ |
|----------------------|----------------------|----------------------|
| A₁ | 1 | 5 |
| A₂ | 4 | 2 |
| A₃ | 3 | 3 |
| Pesos | 0,5 | 0,5 |

Tabla 7: Ejemplo matriz de decisión

Aplicación del método TOPSIS:

Paso 2: Normalización de la Matriz de Decisión

| | C₁ | C₂ |
|------------------------------------|----------------------|----------------------|
| $\sqrt{\sum_{j=1}^m (x_{ij})^2} =$ | 5,09902 | 6,164414 |

| | C₁ | C₂ |
|----------------------|----------------------|----------------------|
| A₁ | 0,19612 | 0,81111 |
| A₂ | 0,78446 | 0,32444 |
| A₃ | 0,58835 | 0,48666 |
| Pesos | 0,5 | 0,5 |

Tabla 8: Ejemplo normalización matriz de decisión

Paso 3: Construcción de la matriz normalizada ponderada

| | C₁ | C₂ |
|----------------------|----------------------|----------------------|
| A₁ | 0,09806 | 0,40555 |
| A₂ | 0,39223 | 0,16222 |
| A₃ | 0,29417 | 0,24333 |

Tabla 9: Ejemplo matriz normalizada ponderada

Paso 4: Determinación de la SIP y la SIN.

| | C_1 | C_2 |
|-------------|----------------|----------------|
| A_1 | 0,09806 | 0,40555 |
| A_2 | 0,39223 | 0,16222 |
| A_3 | 0,29417 | 0,24333 |
| \bar{A}^+ | 0,39223 | 0,40555 |
| \bar{A}^- | 0,09806 | 0,16222 |

Tabla 10: Ejemplo matriz normalizada y SIP Y SIN

Paso 5: Cálculo de las medidas de distancia euclídea.

| | C_1 | C_2 | \bar{d}_i^+ |
|-------|----------|----------|---------------|
| A_1 | 0,086538 | 0,000000 | 0,294174 |
| A_2 | 0,000000 | 0,059211 | 0,243332 |
| A_3 | 0,009615 | 0,026316 | 0,189555 |

Tabla 11: Ejemplo matriz distancias \bar{A}^+

| | C_1 | C_2 | \bar{d}_i^- |
|-------|----------|----------|---------------|
| A_1 | 0,000000 | 0,059211 | 0,243332 |
| A_2 | 0,086538 | 0,000000 | 0,294174 |
| A_3 | 0,038462 | 0,006579 | 0,212227 |

Tabla 12: Ejemplo matriz distancias \bar{A}^-

Paso 6 y 7: Cálculo de la proximidad relativa a la solución ideal y Ordenación.

| | \bar{R}_i | Ranking |
|-------|-----------------|----------|
| A_1 | 0,452706 | 3 |
| A_2 | 0,547294 | 1 |
| A_3 | 0,528215 | 2 |

Tabla 13: Ejemplo matriz solución

Siendo la mejor alternativa A_2 en el ranking. Por tanto, el mejor candidato es el candidato 2 para el puesto de trabajo.

Capítulo 3

El Proceso de Ingeniería del Software

3.1 Descripción del Proceso

Una vez presentado el propósito y los objetivos del proyecto y explicado el modelo teórico de TOPSIS, llega el momento de pasar a detallar el desarrollo del proyecto que se va a realizar. El proyecto aborda el desarrollo de una aplicación de escritorio con tecnología JAVA que implementa un Sistema de Apoyo a la Toma de Decisión mediante el método TOPSIS para resolver problemas multicriterio y multiexperto.

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 [8].
- **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 cómo 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. Éste es, sin duda, una etapa 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, en el momento de 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 una aplicación de escritorio de Apoyo a la Toma de Decisión Multicriterio mediante el método TOPSIS 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:

- *Requerimientos funcionales*: aquellos que se refieren específicamente al funcionamiento de la aplicación o sistema.
- *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.

Esta aplicación se ha diseñado para satisfacer las necesidades de dos perfiles:

- **Administrador:** podrá realizar tareas necesarias para la configuración de los problemas y la resolución de los mismos. Podrá consultar la información de todos los problemas con acceso total.
- **Experto:** se limitará a evaluar los problemas según su opinión sobre las alternativas y criterios. Podrá consultar las mejores alternativas para cada problema resuelto.

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 un experto.
 - f) Modificar un experto.
 - g) Consultar un experto.
 - h) Eliminar un experto.
 - i) Asignar expertos a un problema.
 - j) Resolver un problema.
 - k) Consultar la solución y la mejor alternativa de un problema.
- Para el usuario Experto:
 - a) Introducir la valoración de un problema asignado.
 - b) Consultar el estado de un problema asignado.
 - c) Consultar la mejor alternativa de un problema asignado ya resuelto.

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 cuando un usuario entra al 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 (BBDD), y si son correctas, el usuario accede a la aplicación.

2. RF-02: Salir del sistema

Dentro de la aplicación, 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 un nombre identificador, una breve descripción del mismo, crear las alternativas y crear los criterios.

4. RF-04: Modificar un problema

El sistema debe permitir modificar un problema almacenado en la BBDD, en el que se pueda añadir, modificar o eliminar alternativas, añadir, modificar o eliminar criterios, 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 nombre, definición, alternativas, criterios, expertos asignados, estados, valoraciones establecidos, mejor alternativa.

6. RF-06: Eliminar un problema

Debe permitir eliminar de la BBDD un problema existente.

7. RF-07: Crear un *Experto*

El sistema debe proporcionar al *Administrador* la posibilidad de crear un experto en el que se especifique su identificador, nombre, contraseña y especialidad.

8. RF-08: Modificar un *Experto*

El sistema debe permitir modificar los datos de un experto en la BBDD.

9. RF-09: Consultar un *Experto*

Debe permitir consultar toda la información referente a un experto, como puede ser nombre, identificador y especialidad.

10. RF-10: Eliminar un *Experto*

El sistema permitirá eliminar de la BBDD un experto existente.

11. RF-11: Asignar *Expertos* a un problema.

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

12. RF-12: Resolver un problema

Debe permitir que el *Administrador* pueda resolver un problema, siempre que, tenga una valoración superior al 50% de las valoraciones totales de los *Expertos*.

13. RF-13: Consultar la solución y la mejor alternativa de un problema

Debe permitir consultar un ranking de las mejores soluciones, de las mejores alternativas, de aquellos problemas resueltos, para la toma de decisión.

14. RF-14: Introducir la valoración de un problema asignado

El sistema debe proporcionar al *Experto* la posibilidad de introducir las valoraciones en un problema en el que esté asignado y visualizar el estado de dicho problema.

15. RF-15: Consultar el estado de un problema.

El sistema debe proporcionar al *Experto* la posibilidad de consultar la descripción de un problema asignado, las valoraciones introducidas por dicho *Experto*, así como visualizar el estado de resuelto o no del problema asignado.

16. RF-16: Consultar la mejor alternativa de un problema resuelto.

El sistema debe proporcionar al *Experto* la posibilidad de ver la mejor alternativa, de entre todas las posibles, como mejor opción según las valoraciones de los *Expertos* y el modelo de resolución TOPSIS, de un problema asignado.

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 a llegar a ser críticos para la aceptación del sistema. Estos requerimientos normalmente especifican propiedades del sistema o del producto en si (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 de desarrollo de la aplicación es local, indicaremos a continuación los requerimientos del equipo en el que se instalará la aplicación.

Los requerimientos del equipo 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 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*: debe soportar una resolución de 800x600 y superiores.

2. Software

- *Sistema Operativo*: el servidor de la aplicación trabaja sobre un sistema operativo Windows en su últimas versiones (Vista, 7, 8).
- *Maquina virtual Java*: Versión de la máquina virtual de Java JRE 1.7 o superiores.
- *Sistema Gestor de Bases de Datos*: la aplicación trabaja con la base de datos mysql versión 5.6.12.
- .jar que contiene la aplicación.
- 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:

1. **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 qué 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.
2. **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.
3. **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 quiénes 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 modelo de resolución TOPSIS.
- ✓ *Uso de equipos/programas informáticos:* es necesario que tenga conocimientos sobre otros programas informáticos, además de nuestra aplicación pero, por lo general, 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,... También deberá tener conocimientos de BBDD como MySQL y administración de PHPMyAdmin para gestionar las bases de datos, en caso que sea necesario.
- ✓ *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 y criterios de un problema de una forma acertada.
- ✓ *Uso de equipos/programas informáticos:* no es necesario que tenga conocimientos sobre otros programas informáticos, sólo sobre nuestra aplicación, 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 por qué 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 [1].

Los casos de uso describen cómo 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 cuáles 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.

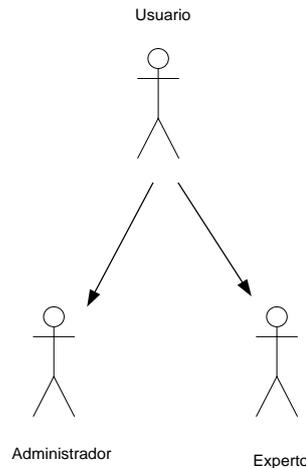


Figura 7. Actores (Casos de uso)

En nuestro sistema contamos con los dos siguientes actores como se ve en la Figura 7:

- **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 selección dando su valoración 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 (ver Figura 8), es decir, un diagrama que describe completamente la funcionalidad de un sistema:

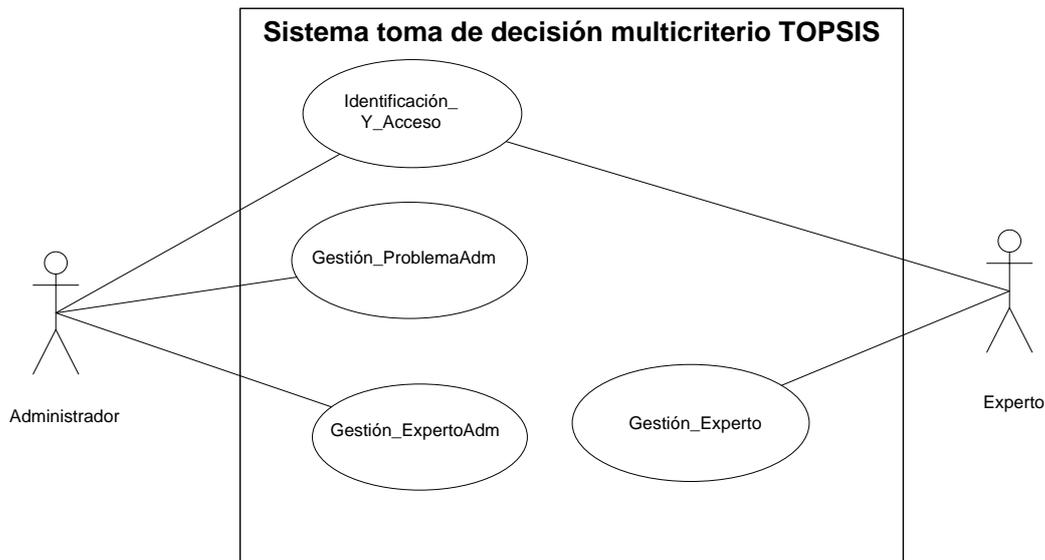


Figura 8. Sistema (Casos de uso)

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 "Identificación_Y_Acceso", "Gestión_ProblemaAdm", "Gestión_ExpertoAdm", "Gestión_Experto", "Gestión_Alternativas" y "Gestión_Criterios" requieren ser detallados en más profundidad.

- **Caso de uso 1: Identificación_Y_Acceso**

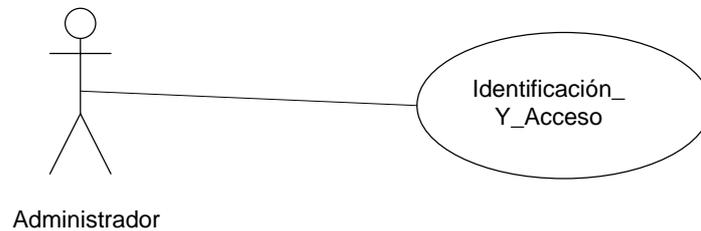


Figura 9. Caso de uso Identificación_Y_Acceso

Actores participantes: Administrador y Experto.

Condición de entrada: Que existan usuarios en el sistema que tengan nombre de usuario y contraseña. Que el administrador haya creado una cuenta con usuario y contraseña correctos.

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 la pantalla de gestión y 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: Gestión_ProblemaAdm**

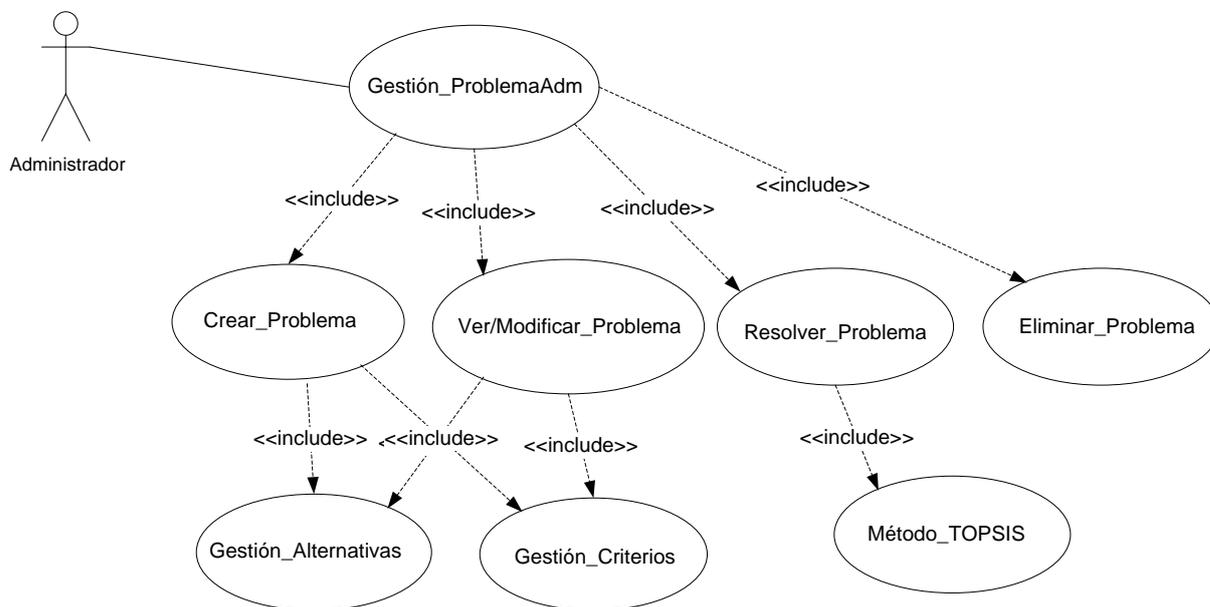


Figura 10. Caso de uso Gestión_ProblemaAdm

Actores participantes: Administrador.

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

Flujo de eventos:

1. El sistema muestra una zona para la gestión de problemas.
2. El usuario administrador puede pulsar el botón "Nuevo". Se pasa al caso de uso Crear_Problema, para generar un nuevo problema, se realiza S-1.
3. El usuario administrador puede observar un listado de problemas, extraídos de la BBDD del sistema, y ver el estado en el que se encuentra cada uno.
4. El usuario administrador puede buscar un determinado problema escribiendo el "Nombre" o "Estado" deseado y pulsará el botón "Buscar".
5. El usuario administrador puede elegir una de las tres opciones que muestra el sistema para realizar sobre un problema concreto mostrado:
 - Si pulsa "Ver/Modificar", nos lleva al caso de uso Ver/Modificar_Problema. Se realiza S-2.
 - Si pulsa "Eliminar", nos lleva al caso de uso Eliminar_Problema. Se realiza S-3.
 - Si pulsa "Resolver", nos lleva al caso de uso Resolver_Problema. Se realiza S-4.

Subflujo de eventos:**S-1: Crear_Problema**

1. El sistema mostrará el formulario para generar un nuevo problema.
2. El usuario administrador introducirá el nombre y la descripción detallada del problema.
3. El usuario administrador creará y añadirá nuevas alternativas al problema. Caso de uso Gestión_Alternativas.
4. El usuario administrador creará y añadirá nuevos criterios al problema. Caso de uso Gestión_Criterios.
5. El usuario administrador vinculará expertos al problema, a partir de una lista de expertos ya existentes en el sistema.
6. El usuario administrador aceptará, creándose el problema en el sistema, o cancelará la información introducida.

S-2: Ver/Modificar_Problema

Condiciones de entrada: El problema estará almacenado previamente en el sistema.

1. El sistema muestra la información de un problema.
2. Si el problema ya está resuelto, solo muestra la información.
3. El usuario administrador puede modificar la descripción del problema.
4. El usuario administrador puede modificar las alternativas. Caso de uso Gestión_Alternativas.
5. El usuario administrador puede modificar los criterios. Caso de uso Gestión_Criterios.
6. El usuario administrador puede modificar los expertos vinculados al problema.
7. El usuario administrador aceptará o cancelará para salir de la interfaz.

S-3: Eliminar_Problema

Condiciones de entrada: El problema estará almacenado previamente en el sistema.

1. El sistema muestra la opción de eliminar un problema.
2. Si el usuario administrador elimina el problema, aparecerá una pantalla de confirmación de eliminación.

3. El usuario administrador acepta y el problema queda eliminado del sistema.

S-4: Resolver_Problema

Condiciones de entrada: El problema estará almacenado previamente en el sistema. Se podrá hacer la resolución cuando el problema tenga más del 50% de los valores introducidos por los expertos asignados y tenga expertos asociados.

1. El sistema muestra la opción de resolver un problema.
2. Si el usuario administrador resuelve el problema, aparecerá una pantalla de confirmación de resolución del problema, (E-1).
3. El usuario administrador acepta y el sistema modifica sus valores y aplica el método TOPSIS (Subflujo Método_TOPSIS) para ese problema, quedando todo almacenado en el sistema.

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

Excepciones:

E-1: Para que un problema pueda resolverse mediante el método TOPSIS necesita estar valorado en más del 50% de sus opciones por los expertos asignados y tener asociados expertos.

• **Caso de uso 3: Gestión_ExpertoAdm**

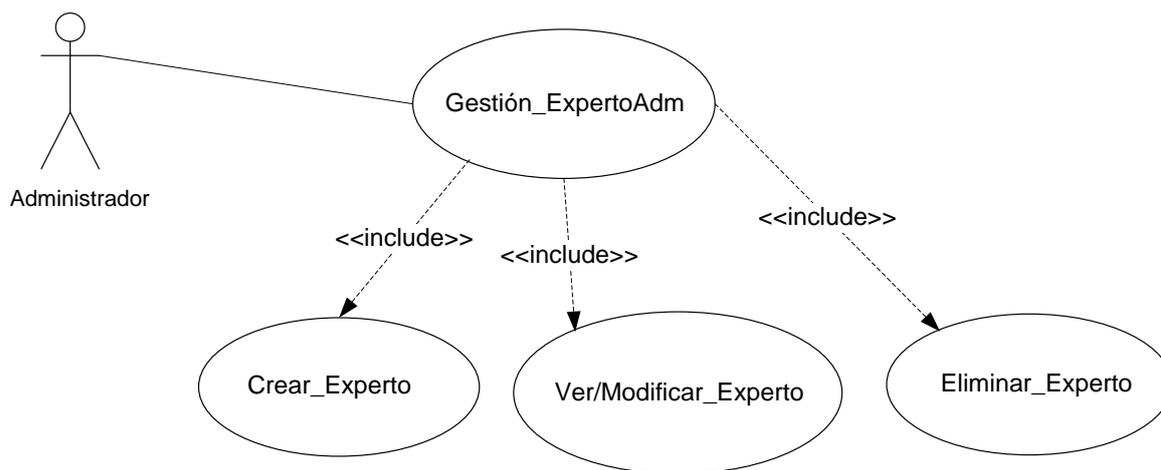


Figura 11. Caso de uso Gestión_ExpertoAdm

Actores participantes: Administrador.

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

Flujo de eventos:

1. El sistema muestra una zona para la gestión de expertos.
2. El usuario administrador puede pulsar el botón "Nuevo". Se pasa al caso de uso Crear_Experto, para generar un nuevo experto, se realiza S-1.
3. El usuario administrador puede buscar un determinado experto escribiendo el "Nombre" o "Especialidad" deseados y pulsará el botón "Buscar".
4. El usuario administrador puede elegir una de las dos opciones que muestra el sistema para realizar sobre un experto concreto dentro de la lista de expertos mostrados, que han sido buscados en la BBDD del sistema:
 - Si pulsa "Ver/Modificar", nos lleva al caso de uso Ver/Modificar_Experto. Se realiza S-2.
 - Si pulsa "Eliminar", nos lleva al caso de uso Eliminar_Experto. Se realiza S-3.

Subflujo de eventos:

S-1: Crear_Experto

1. El sistema mostrará el formulario para generar un nuevo experto.
2. El usuario Administrador introducirá la información del experto, (E-1).
3. El usuario Administrador aceptará, creándose el experto en el sistema, o cancelará la información introducida.

S-2: Ver/Modificar_Experto

Condición de entrada: El experto estará almacenado previamente en el sistema.

1. El sistema muestra la información de un experto.
2. El usuario administrador puede modificar la información del experto.
3. El usuario administrador aceptará o cancelará para salir de la interfaz.

S-3: Eliminar_Experto

Condición de entrada: El experto estará almacenado previamente en el sistema.

1. El sistema muestra la opción de eliminar un experto.
2. Si el usuario administrador elimina el experto, aparecerá una pantalla de confirmación de eliminación.

3. El usuario administrador acepta y el experto queda eliminado del sistema.

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

Excepciones:

E-1: El identificador introducido para el nuevo experto ya existe; el sistema muestra un mensaje de error y comunica al administrador que lo vuelva a intentar.

• **Caso de uso 4: Gestión_Experto**

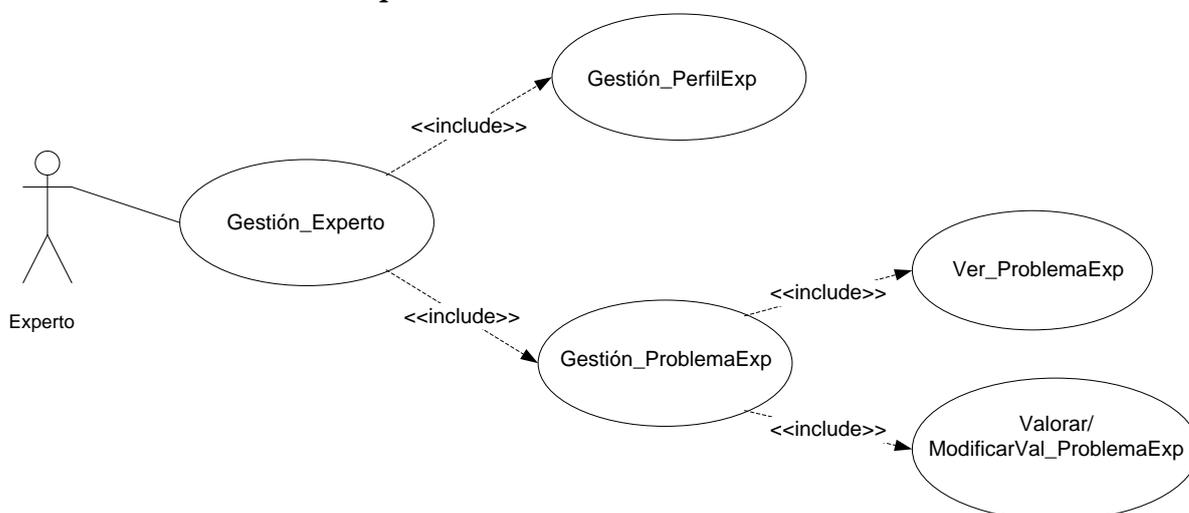


Figura 12. Caso de uso Gestión_Experto

Actores participantes: Experto.

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

Flujo de eventos:

1. El usuario experto puede modificar la información de su perfil de usuario, se lleva a cabo el caso de uso Gestión_PerfilExp, se realiza S-1.
2. El sistema muestra la pantalla con una lista de los problemas a los que está vinculado el experto.
3. El usuario experto puede buscar un determinado problema escribiendo el "Nombre" o "Estado" deseados y pulsará el botón "Buscar".
4. El usuario Experto podrá realizar dos opciones sobre un problema dependiendo del estado del problema:

- Si está resuelto el problema, el usuario Experto elegirá la opción de Ver_ProblemaExp. Se realiza S-2.
- Si no está resuelto el problema, el usuario Experto elegirá la opción de Valorar/ModificarVal_ProblemaExp. Se realiza S-3.

Subflujo de eventos:**S-1: Gestión_PerfilExp**

1. El sistema mostrará el formulario para modificar la información del experto.
2. El usuario experto introducirá la nueva información.
3. El sistema dará la posibilidad al experto de poder cambiar la contraseña.
4. El usuario experto cambia la contraseña por una nueva.
5. El usuario experto acepta los cambios y éstos se almacenan en el sistema.

S-2: Ver_ProblemaExp

Condición entrada: El problema ha sido resuelto previamente.

1. El sistema muestra la información general de un problema.
2. El usuario experto puede consultar su valoración de dicho problema.
3. El usuario experto puede consultar la mejor solución para ese problema, viendo los valores medios y la mejor alternativa como la mejor decisión.

S-3: Valorar/ModificarVal_ProblemaExp

Condición entrada: El problema no ha sido resuelto previamente. El problema aparece en la lista de problemas asignados al experto. El problema posee alternativas y criterios para valorar.

1. El sistema muestra la información general de un problema.
2. El usuario Experto debe valorar cada una de las alternativas y criterios de dicho problema.
3. El usuario Experto introducirá la valoración por cada opción, introduciendo un número mayor que 0 y con dos dígitos decimales como máximo.
4. El usuario Experto acepta los cambios realizados y dichos cambios se almacenan en el sistema.

5. El sistema confirma que los datos han sido cambiados y almacenados.

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

• **Caso de uso 5: Gestión_Alternativas**

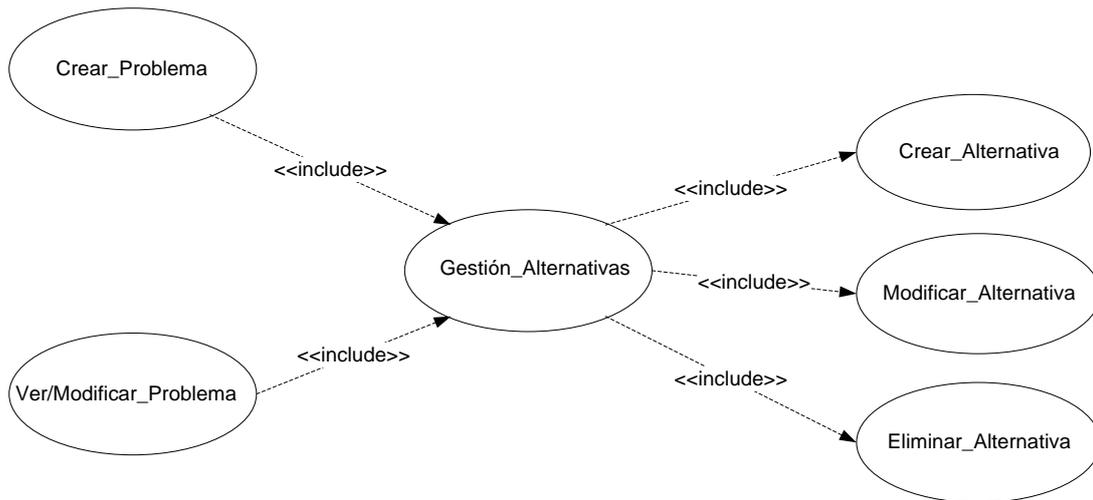


Figura 13. Caso de uso Gestión_Alternativas

Actores participantes: Administrador.

Condición de entrada: Que el administrador esté identificado en el sistema. Que haya sido llamada desde los casos de uso Crear_Problema o Ver/Modificar_Problema.

Flujo de eventos:

1. El usuario administrador podrá realizar tres opciones sobre una alternativa:
 - El sistema muestra un formulario para introducir los datos de una nueva alternativa. Se realiza S-1.
 - El sistema muestra un formulario relleno con la información de una alternativa y ésta puede ser cambiada. Se realiza S-2.
 - El usuario administrador desea eliminar la alternativa. Se realiza S-3.

Subflujo de eventos:

S-1: Crear_Alternativa

1. El sistema mostrará el formulario para generar una nueva alternativa.

2. El usuario administrador introducirá la información de la alternativa, (E-1, E-2).
3. El usuario administrador aceptará, creándose la alternativa en el sistema para un determinado problema, o cancelará la información introducida.

S-2: Ver/Modificar_Alternativa

Condición de entrada: La alternativa estará almacenada previamente en el sistema.

1. El sistema muestra la información de una alternativa.
2. El usuario administrador puede modificar la información de la alternativa, (E-1, E-2).
3. El usuario administrador aceptará o cancelará para salir de la interfaz.

S-3: Eliminar_Alternativa

Condición de entrada: La alternativa estará almacenada previamente en el sistema.

1. El sistema muestra la opción de eliminar una alternativa.
2. Si el usuario administrador elimina la alternativa, aparecerá una pantalla de confirmación de eliminación.
3. El usuario administrador acepta y la alternativa queda eliminada del sistema.

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

Excepciones:

E-1: La información introducida no puede ser cadena vacía.

E-2: El nombre de la alternativa no puede estar repetido.

- **Caso de uso 6: Gestión_Criterios**

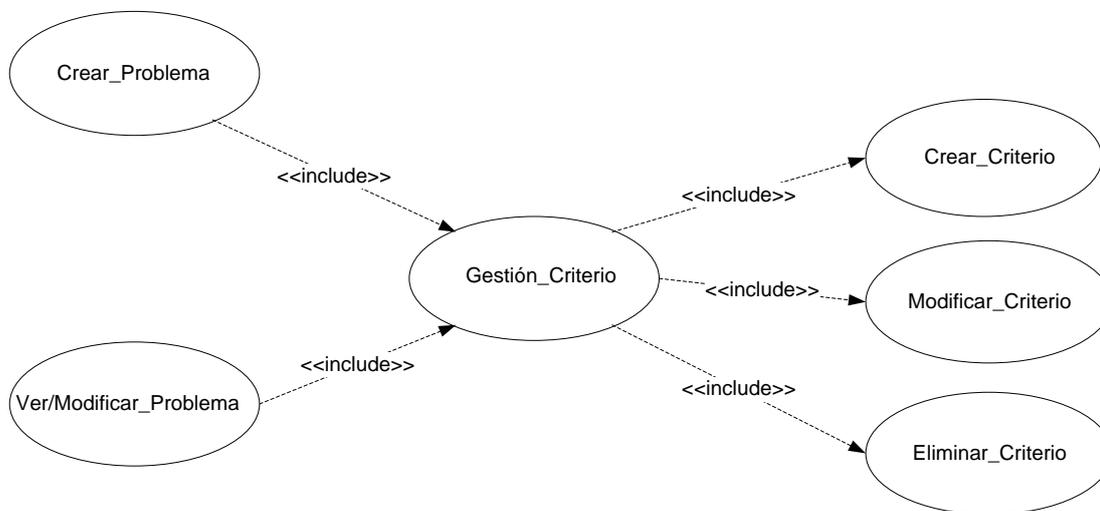


Figura 14. Caso de uso Gestión_Criterios

Actores participantes: Administrador.

Condición de entrada: Que el administrador esté identificado en el sistema. Que haya sido llamada desde los casos de uso Crear_Problema o Ver/Modificar_Problema.

Flujo de eventos:

1. El usuario administrador podrá realizar tres opciones sobre un criterio:
 - El sistema muestra un formulario para introducir los datos de un nuevo criterio. Se realiza S-1.
 - El sistema muestra un formulario relleno con la información de un criterio y ésta puede ser cambiada. Se realiza S-2.
 - El usuario administrador desea eliminar el criterio. Se realiza S-3.

Subflujo de eventos:

S-1: Crear_Criterio

1. El sistema mostrará el formulario para generar un nuevo criterio.
2. El usuario Administrador introducirá toda la información del criterio, (E-1, E-2).
3. El usuario Administrador aceptará, creándose el criterio en el sistema para un determinado problema, o cancelará la información introducida.

S-2: Ver/Modificar_Criterio

Condición de entrada: El criterio estará almacenado previamente en el sistema.

1. El sistema muestra la información de un criterio.
2. El usuario Administrador puede modificar la información del criterio, (E-1, E-2).
3. El usuario Administrador aceptará o cancelará para salir de la interfaz.

S-3: Eliminar_Criterio

Condición de entrada: El criterio estará almacenado previamente en el sistema.

1. El sistema muestra la opción de eliminar un criterio.
2. Si el usuario Administrador elimina el criterio, aparecerá una pantalla de confirmación de eliminación.
3. El usuario Administrador acepta y el criterio queda eliminado del sistema.

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

Excepciones:

E-1: La información introducida no puede ser cadena vacía.

E-2: El nombre del criterio no puede estar repetido.

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. 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 está 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.

➤ **Escenario 1**

- **Nombre:** CrearProblemaAdm
- **Descripción:** El usuario administrador desea crear un nuevo problema en el sistema.
- **Actor:** Administrador
- **Flujo de eventos:**
 1. El usuario entra en el sistema.
 2. El sistema muestra un formulario de acceso de credenciales.
 3. El usuario introduce sus credenciales. Usuario: admin, Contraseña:1234.
 4. El sistema muestra una pantalla dividida en dos, para gestionar los problemas o para gestionar los usuarios.
 5. El usuario administrador pulsa el botón de "Nuevo" en la parte de gestión de problemas.
 6. El sistema muestra un formulario para poder introducir información del nuevo problema.
 7. El usuario administrador rellena los diferentes campos
 - Nombre del problema: Problema palas pádel
 - Descripción del problema: Definir, entre las diferentes marcas de palas de pádel, cuál es la mejor marca para una correcta elección de material por parte de los usuarios.
 8. El usuario administrador pulsará el botón "Nueva Alternativa" para agregar una nueva alternativa al problema.

9. El sistema muestra un formulario para introducir información de la nueva alternativa.
10. El usuario administrador rellena los campos del formulario de creación de la nueva alternativa.
 - Nombre: Dunlop
 - Descripción: Marca comercial Dunlop
11. El usuario administrador pulsará el botón "Nueva Alternativa" para agregar una nueva alternativa al problema.
12. El sistema muestra un formulario para introducir información de la nueva alternativa.
13. El usuario administrador rellena los campos del formulario de creación de la nueva alternativa.
 - Nombre: Artengo
 - Descripción: Marca comercial Artengo
14. El usuario administrador pulsará el botón "Nuevo Criterio" para agregar un nuevo criterio al problema.
15. El sistema muestra un formulario para introducir información del nuevo criterio.
16. El usuario administrador rellena los campos del formulario de creación del nuevo criterio.
 - Nombre: Potencia
 - Peso: 8 (Elige entre un rango de 1 a 10)
 - Descripción: Describe la potencia o fuerza con la que sale despedida la pelota.
17. El usuario administrador pulsará el botón "Nuevo Criterio" para agregar un nuevo criterio al problema.
18. El sistema muestra un formulario para introducir información del nuevo criterio.
19. El usuario administrador rellena los campos del formulario de creación del nuevo criterio.
 - Nombre: Peso
 - Peso: 6 (Elige entre un rango de 1 a 10)
 - Descripción: Describe el peso que posee el material para poder ser más o menos manejable.
20. El usuario administrador pulsará el botón "Vincular Experto" para agregar expertos al problema para que puedan valorar posteriormente las opciones.
21. El sistema muestra una lista de expertos.

22. El usuario administrador pulsa "Vincular" sobre el experto "Manolo", con especialidad "Juegos de Raqueta".
23. El usuario administrador pulsa cerrar para salir de la pantalla de vinculación de expertos.
24. El usuario administrador comprueba que todos los datos introducidos son correctos y pulsa "Aceptar".
25. El sistema muestra el nuevo problema en el listado de problemas en la parte de gestión de problemas del administrador.

➤ **Escenario 2**

- **Nombre:** CrearExpertoAdm
- **Descripción:** El usuario administrador desea crear un nuevo experto en el sistema.
- **Actor:** Administrador
- **Flujo de eventos:**
 1. El usuario entra en el sistema.
 2. El sistema muestra un formulario de acceso de credenciales.
 3. El usuario introduce sus credenciales. Usuario: admin, Contraseña:1234.
 4. El sistema muestra una pantalla dividida en dos, para gestionar los problemas o para gestionar los usuarios.
 5. El usuario administrador pulsa el botón de "Nuevo" en la parte de gestión de expertos.
 6. El sistema muestra un formulario para poder introducir información del nuevo experto.
 7. El usuario administrador rellena los diferentes campos
 - Login: ManuelQA
 - Nombre: Manuel Quesada
 - Contraseña: 123456
 - Especialidad: Juegos de Raqueta
 8. El usuario administrador pulsa "Aceptar".
 9. El sistema muestra el nuevo experto en el listado de expertos en la parte de gestión de expertos del administrador.

➤ **Escenario 3**

- **Nombre:** ResolverProblemaAdm
- **Descripción:** El usuario administrador desea resolver un problema que previamente ha sido valorado por los expertos en más de un 50% de todas sus valoraciones, y se quiere saber cuál es la mejor alternativa a elegir para la toma de decisión.

- **Actor:** Administrador
- **Flujo de eventos:**
 1. El usuario entra en el sistema.
 2. El sistema muestra un formulario de acceso de credenciales.
 3. El usuario introduce sus credenciales. Usuario: admin, Contraseña:1234.
 4. El sistema muestra una pantalla dividida en dos, para gestionar los problemas o para gestionar los usuarios.
 5. En la parte de gestión de problemas, el sistema muestra una lista de problemas al administrador, donde se puede ver el "Nombre" y "Estado" en el que se encuentra.
 6. El usuario administrador pulsa el botón de "Resolver" del problema "Problema palas pádel", de la lista de problemas.
 7. El sistema muestra un mensaje de confirmación preguntando si se quiere resolver el problema.
 8. El usuario administrador pulsa "Si" para realizar la resolución del problema.
 9. El sistema muestra una pantalla con la información general del problema, así como, de las valoraciones medias de todos los expertos vinculados y el ranking de las mejores alternativas.
 10. El usuario administrador pulsará "Cerrar".
 11. El sistema mostrara el problema "Problema palas pádel" con el estado "Resuelto" en el listado de problemas de la parte gestión problemas del administrador.

➤ **Escenario 4**

- **Nombre:** ValorarProblemaExp
- **Descripción:** El usuario Experto desea valorar las alternativas y criterios de un problema que tiene asignado.
- **Actor:** Experto
- **Flujo de eventos:**
 1. El usuario entra en el sistema.
 2. El sistema muestra un formulario de acceso de credenciales.
 3. El usuario introduce sus credenciales.
Usuario: Manolo, Contraseña:123456.
 4. El sistema muestra una pantalla con un listado de problemas asignados al experto.

5. El usuario experto pulsará el botón "Valorar" del problema "Problema pala pádel".
6. El sistema muestra en la parte derecha la información general asociada al problema a valorar.
7. El usuario experto pulsará la pestaña, en la parte de arriba, "Valoración".
8. El sistema muestra una tabla organizada en alternativas y criterios del problema "Problema palas pádel".
9. El usuario experto va introduciendo una puntuación en cada celda, dentro del rango permitido, cruce de una alternativa con un criterio. Si el usuario valora fuera del rango permitido se inicializa la valoración. El usuario experto introduce para cada celda:
 - Dunlop+Potencia: 80
 - Dunlop+Peso: 60
 - Artengo+Potencia: 70,3
 - Artengo+Peso: 50,5
10. El usuario experto pulsará el botón de "Aceptar" cuando haya terminado de valorar las diferentes opciones.
11. El sistema muestra un mensaje de éxito para indicar que las valoraciones han sido almacenadas correctamente.

➤ **Escenario 5**

- **Nombre:** CambioContraseñaExp
- **Descripción:** El usuario experto desea cambiar su contraseña de acceso al sistema.
- **Actor:** Experto
- **Flujo de eventos:**
 1. El usuario entra en el sistema.
 2. El sistema muestra un formulario de acceso de credenciales.
 3. El usuario introduce sus credenciales.
Usuario: Manolo, Contraseña:123456.
 4. El sistema muestra una pantalla con un listado de problemas asignados al experto.
 5. El usuario experto pulsará el botón "Perfil de usuario" de la parte superior izquierda de la pantalla del experto.
 6. El sistema muestra un formulario para poder modificar información del experto.
 7. El usuario experto pulsará el botón "Cambiar contraseña".
 8. El sistema mostrará un formulario para cambiar la contraseña del experto.

9. El experto introduce la siguiente información:
 - Contraseña actual: 123456
 - Nueva contraseña: ManoloQA83
 - Confirmar nueva contraseña: ManoloQA83
10. El usuario experto pulsará el botón "Aceptar".

3.4 Diseño del Sistema

3.4.1 Diagrama completo de clases

Antes de mostrar el diagrama de clases, vemos apropiado explicar cómo va a ser el funcionamiento o la lógica 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) [36]. En este esquema se definen tres roles bien diferenciados, como se aprecia en la Figura 15:

- **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.

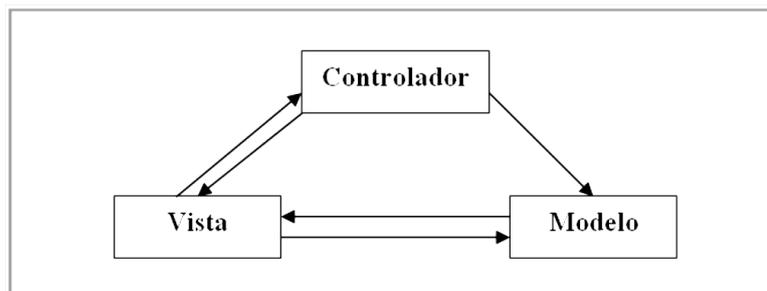


Figura 15. Esquema de la arquitectura MVC

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.

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

- **Paquete controllers:** Es la capa controladora, llevará a cabo todo el control del sistema y permitirá al usuario realizar cualquier operación sobre éste. Esta capa, 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 JPA.
- **Paquete models y models.tablemodels:** Implementa la lógica de la aplicación, es decir, almacena y organiza todos los datos, el estado de la aplicación y tiene los métodos que manipulan esos datos.
- **Paquete views.panels, views.dialogs, views.tables.columns y views.frames:** Implementa todo lo relacionado con la interfaz y la forma de mostrar la información al usuario.
- **Paquete utils:** Contienen todo lo necesario para el manejo de información.

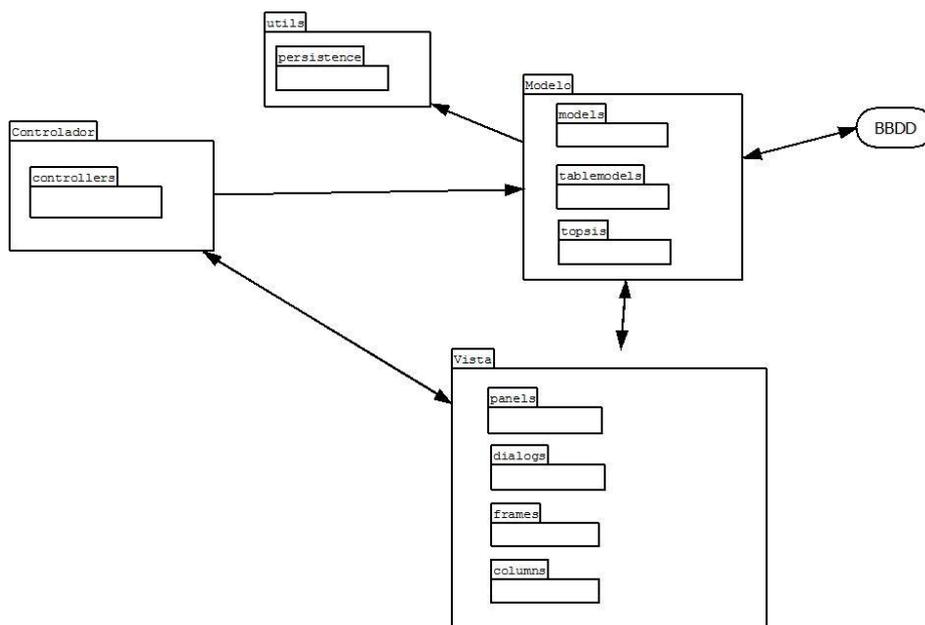


Figura 16. Relación de paquetes

3.4.2 Diagrama de Paquetes

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

3.4.2.1 Paquete controllers

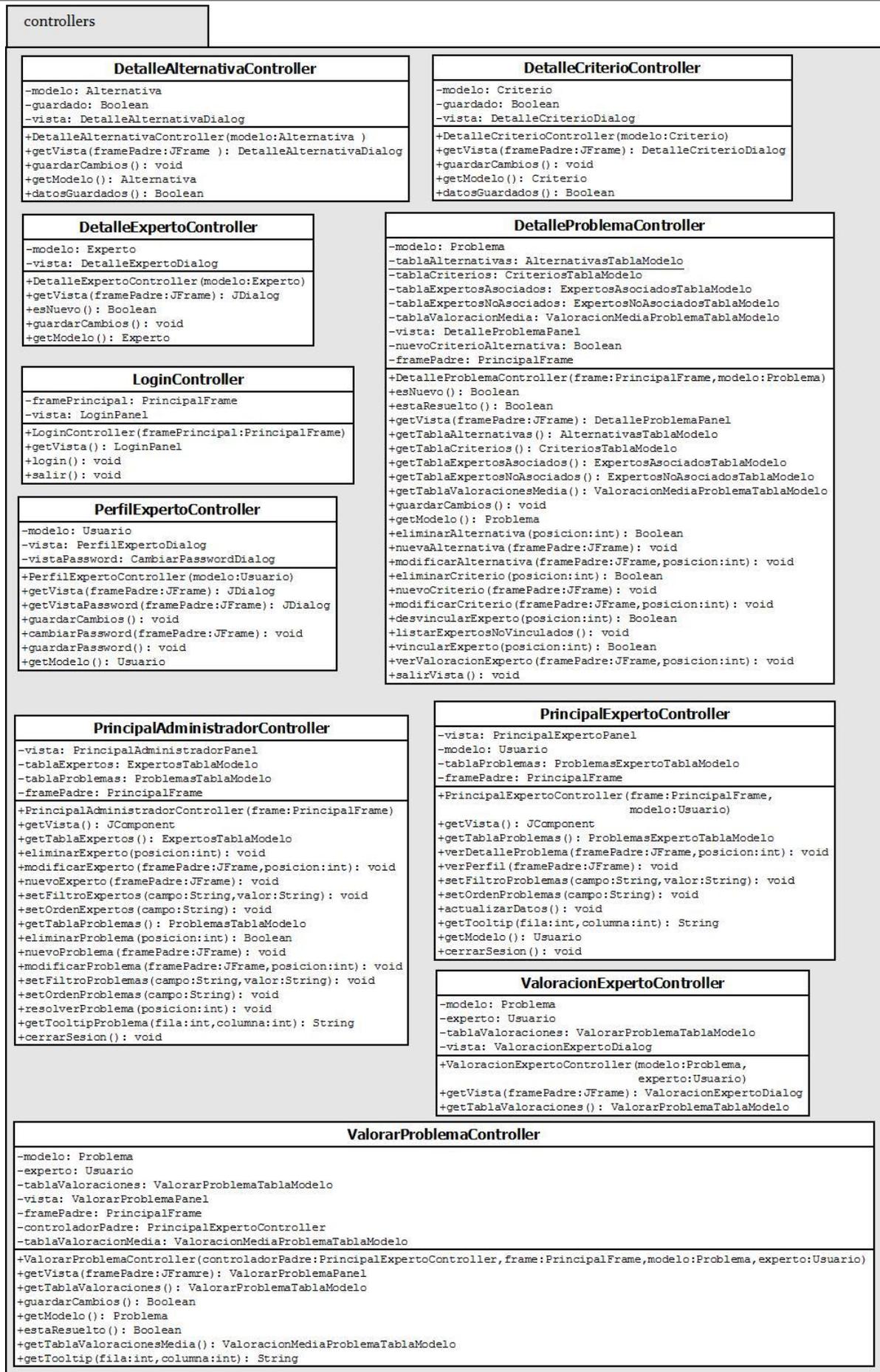


Figura 17. Paquete controllers

3.4.2.2 Paquete models

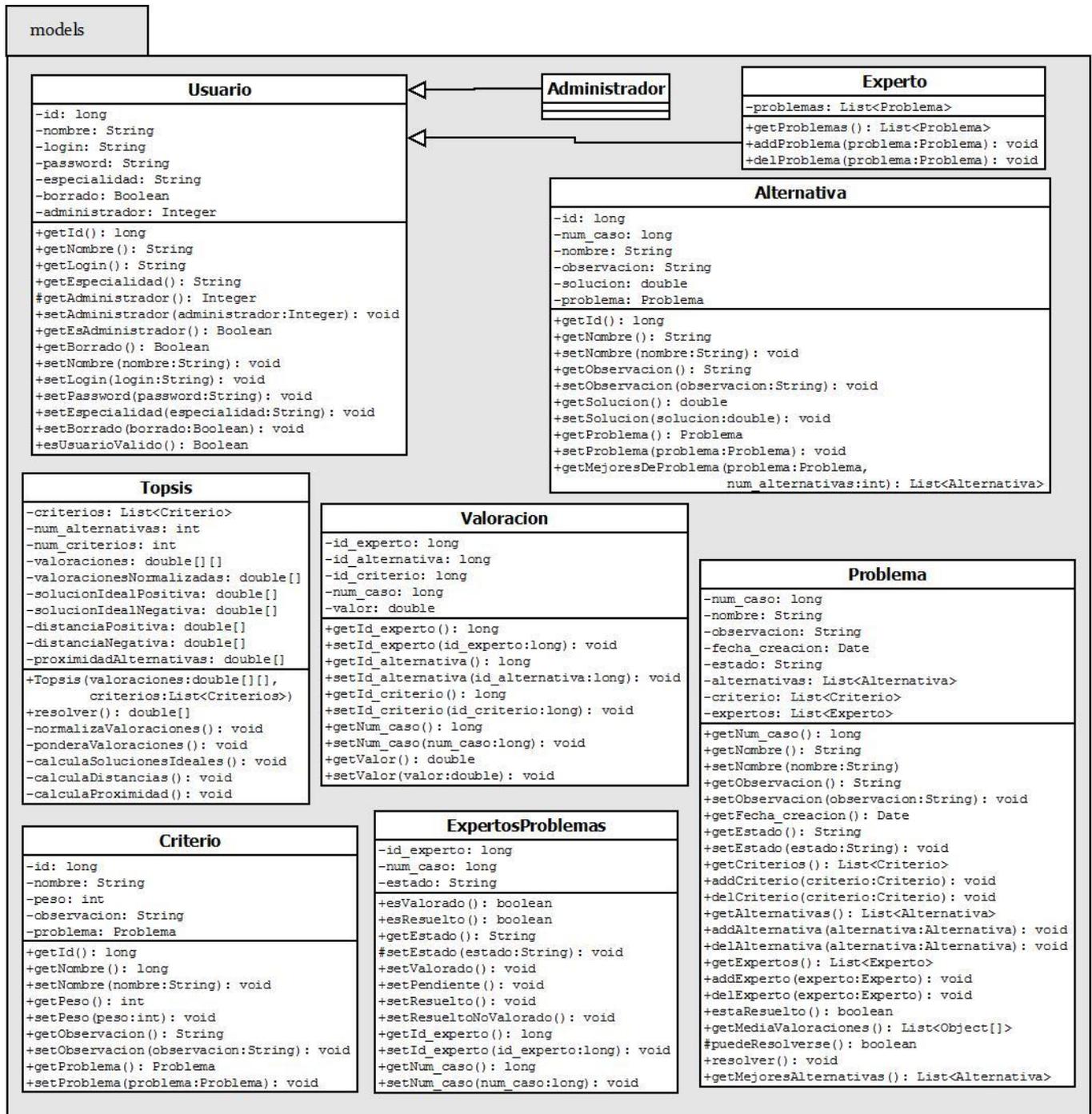


Figura 18. Paquete models

3.4.2.3 Paquete dialogs

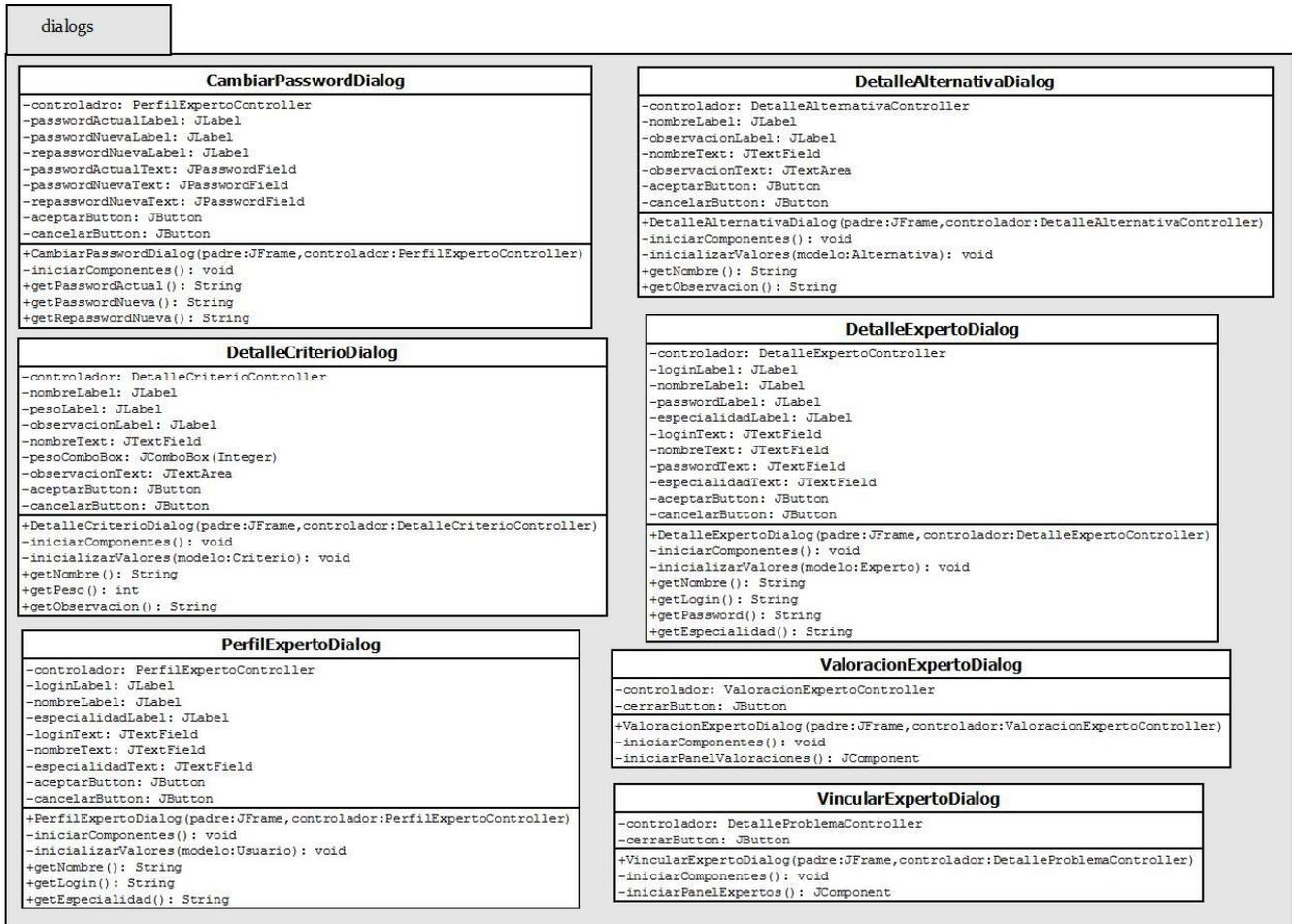


Figura 19. Paquete dialogs

3.4.2.4 Paquete panels



Figura 20. Paquete panels

3.4.2.5 Paquete tablemodels

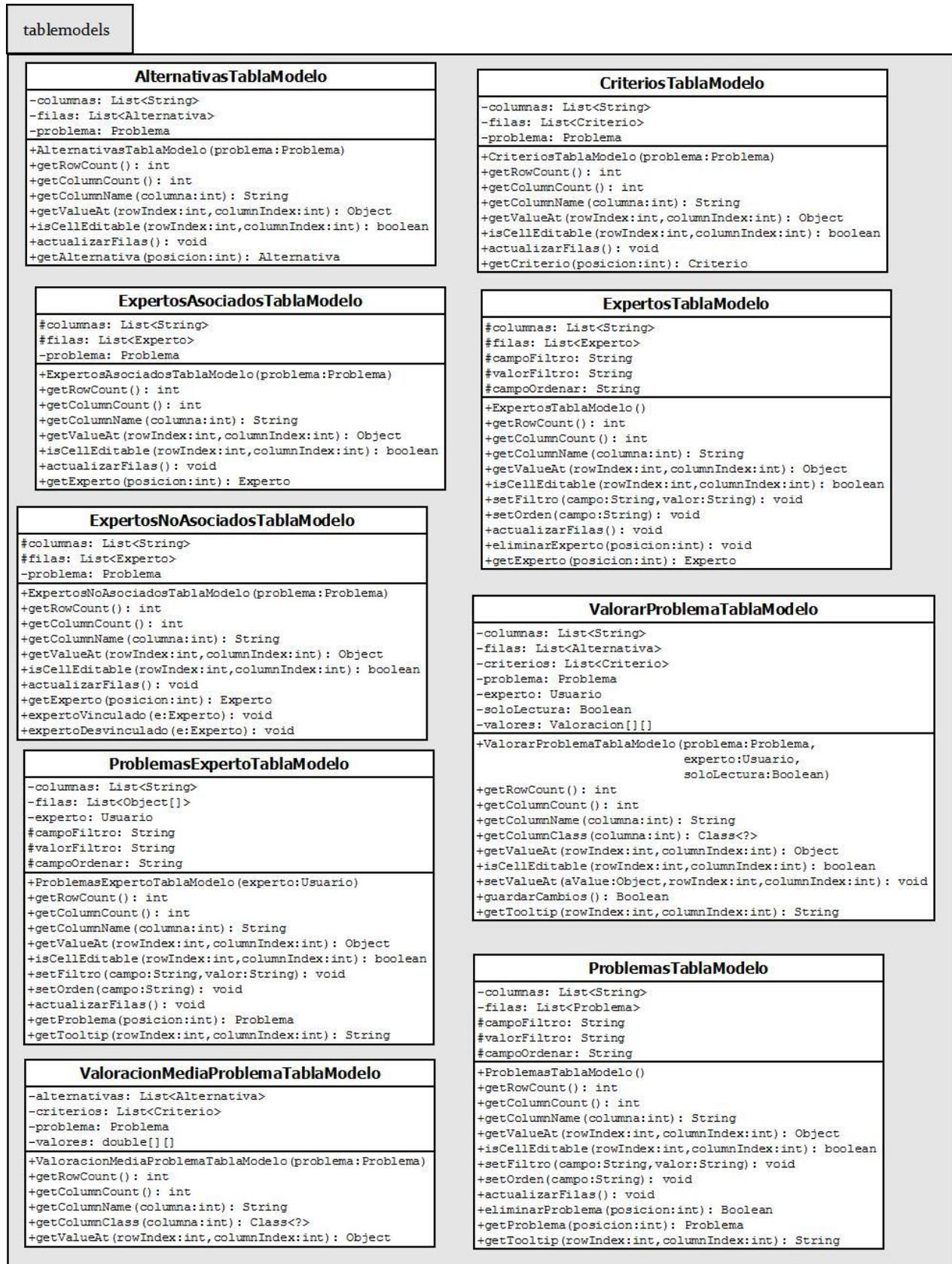


Figura 21. Paquete tablemodels

3.4.2.6 Paquete utils

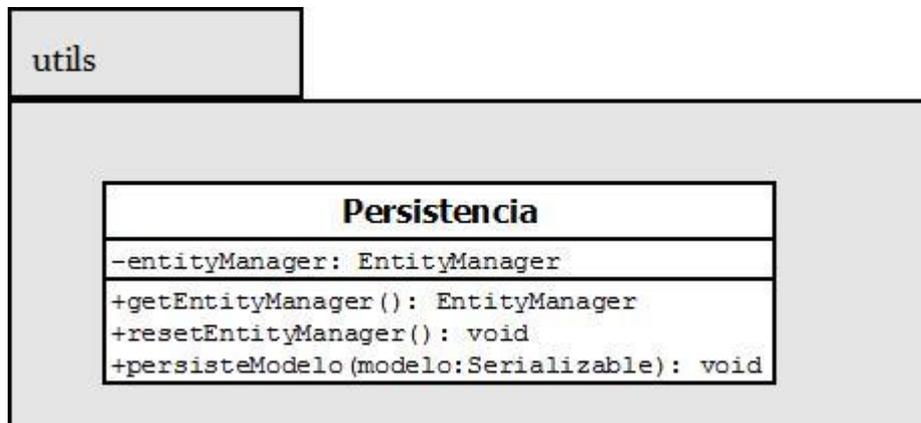


Figura 22. Paquete utils

3.4.2.7 Paquete frames

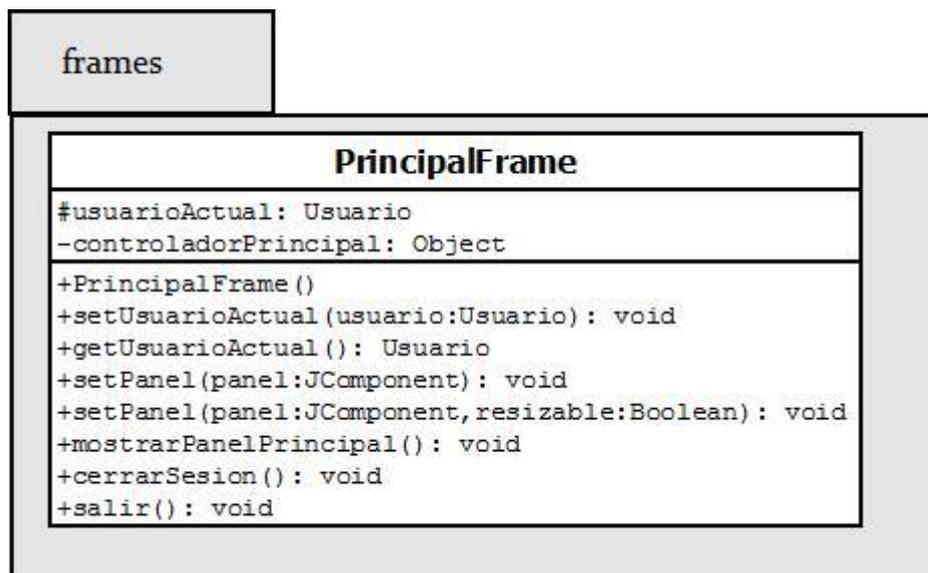


Figura 23. Paquete frames

3.4.2.8 Paquete tables.columns

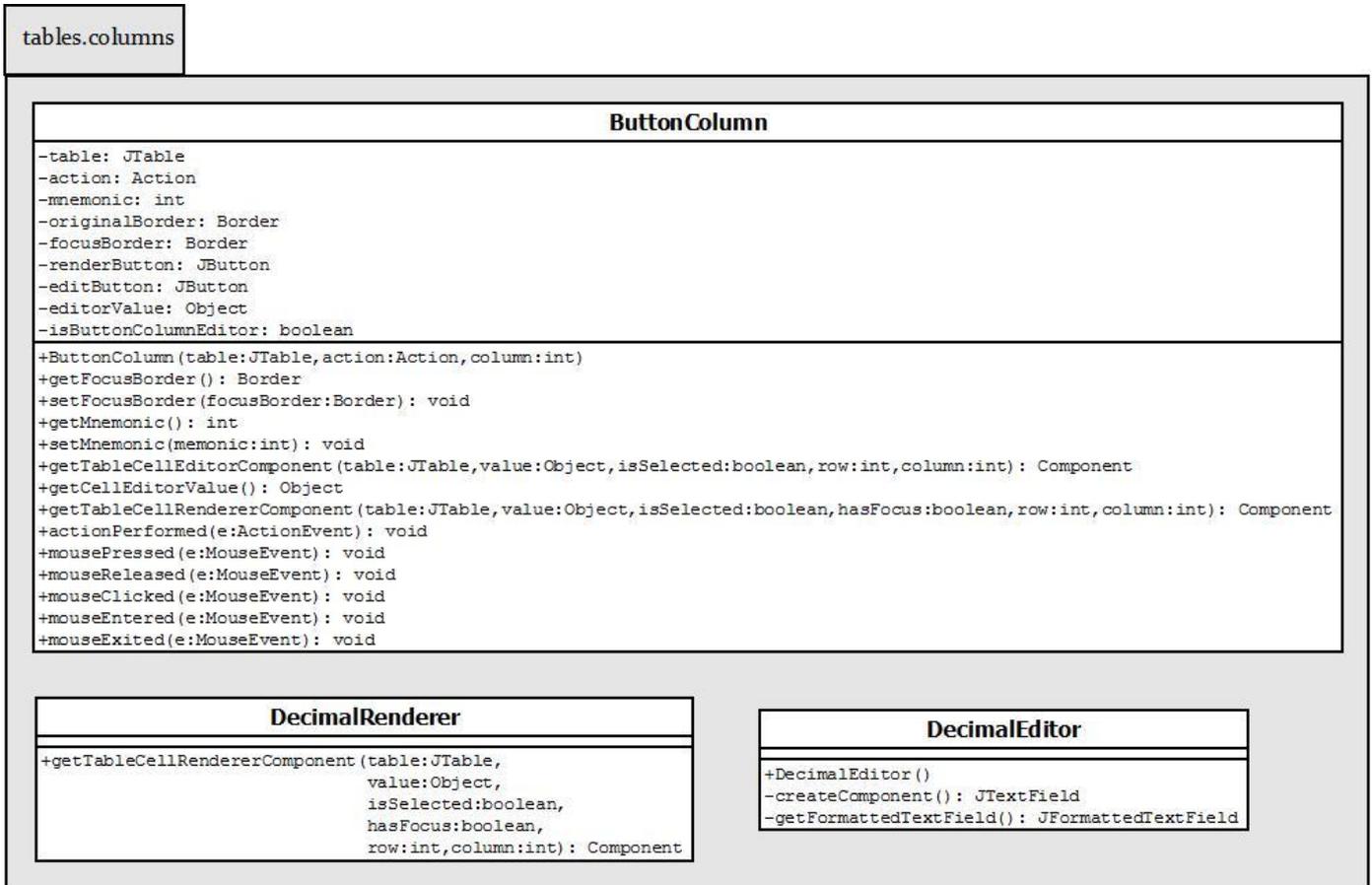


Figura 24. Paquete tables.columns

3.4.3 Diseño de 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.

3.4.3.1 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 una 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 con un nombre en el interior del más pequeño.
- **Cardinalidad de las relaciones:** existen tres tipos de cardinalidades de 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, debe estar formada por uno o más de sus atributos.

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.

3.4.3.2 Normalización en el modelo Entidad-Relación

La normalización es un proceso que consiste 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 BBDD (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 está compuesta por un único atributo esta 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.

3.4.3.3 Esquema Conceptual

Necesitamos convertir nuestros elementos del sistema en entidades o relaciones. Es obvio que Usuarios, Problemas, Alternativas y Criterios se convertirán en entidades en nuestro Esquema Conceptual.

En cuanto a las relaciones:

- **R1:** Un Usuario puede estar asignado a varios Problemas, y un Problema puede tener asociado varios Usuarios.
- **R2:** Un Problema puede tener varias Alternativas, y una Alternativa pertenece a un Problema.
- **R3:** Una Problema posee varios Criterios, y un Criterio pertenece a un Problema.
- **R4:** Un Usuario puede valorar varios Criterios de varias Alternativas con un valor concreto, un Criterio y una Alternativa son valorados por varios Usuarios, y una Alternativa y un Criterio están relacionados.

Estas entidades con sus atributos y sus relaciones se pueden observar en la Figura 25.

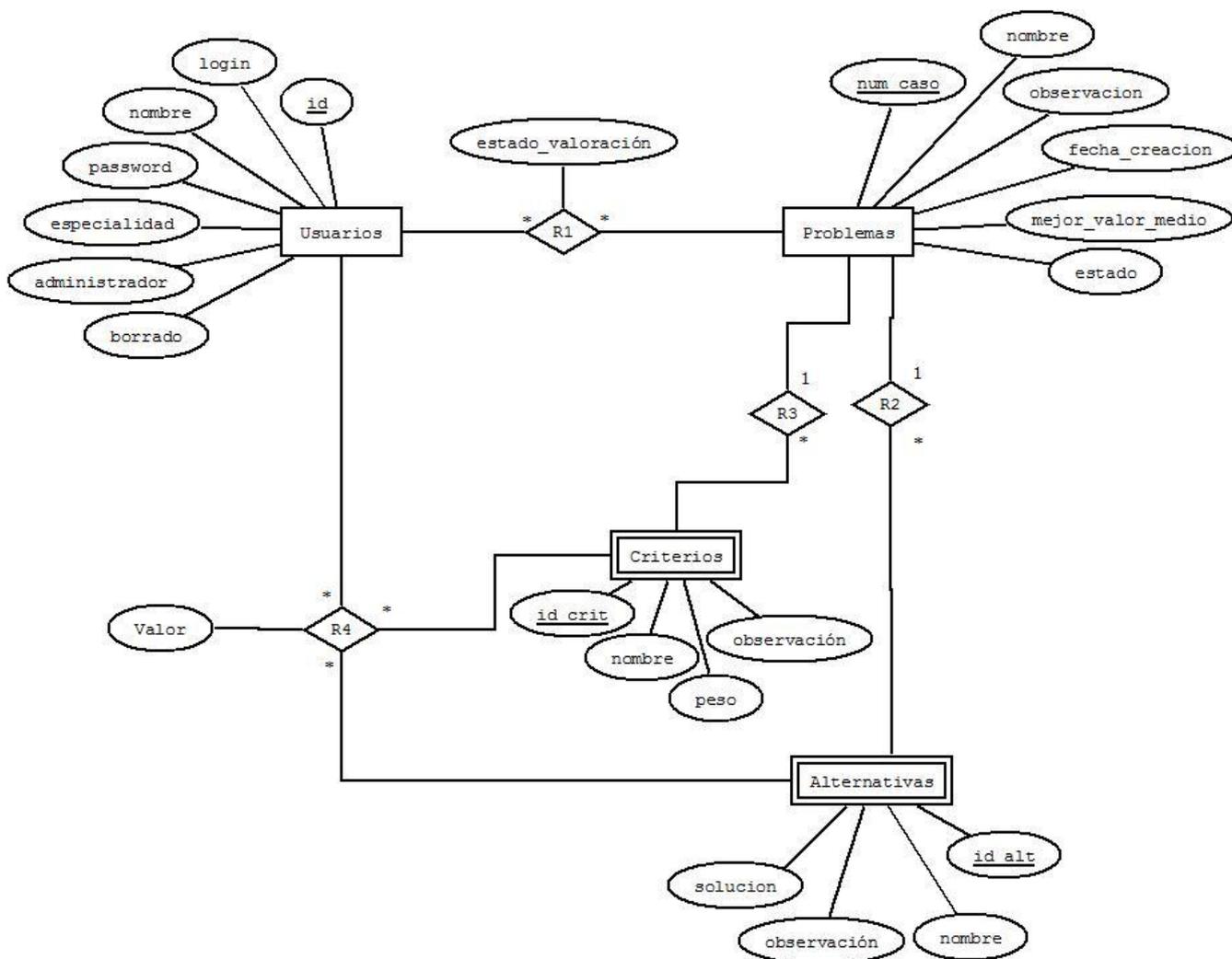


Ilustración 25. Esquema Conceptual E-R

3.4.3.4 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 26:

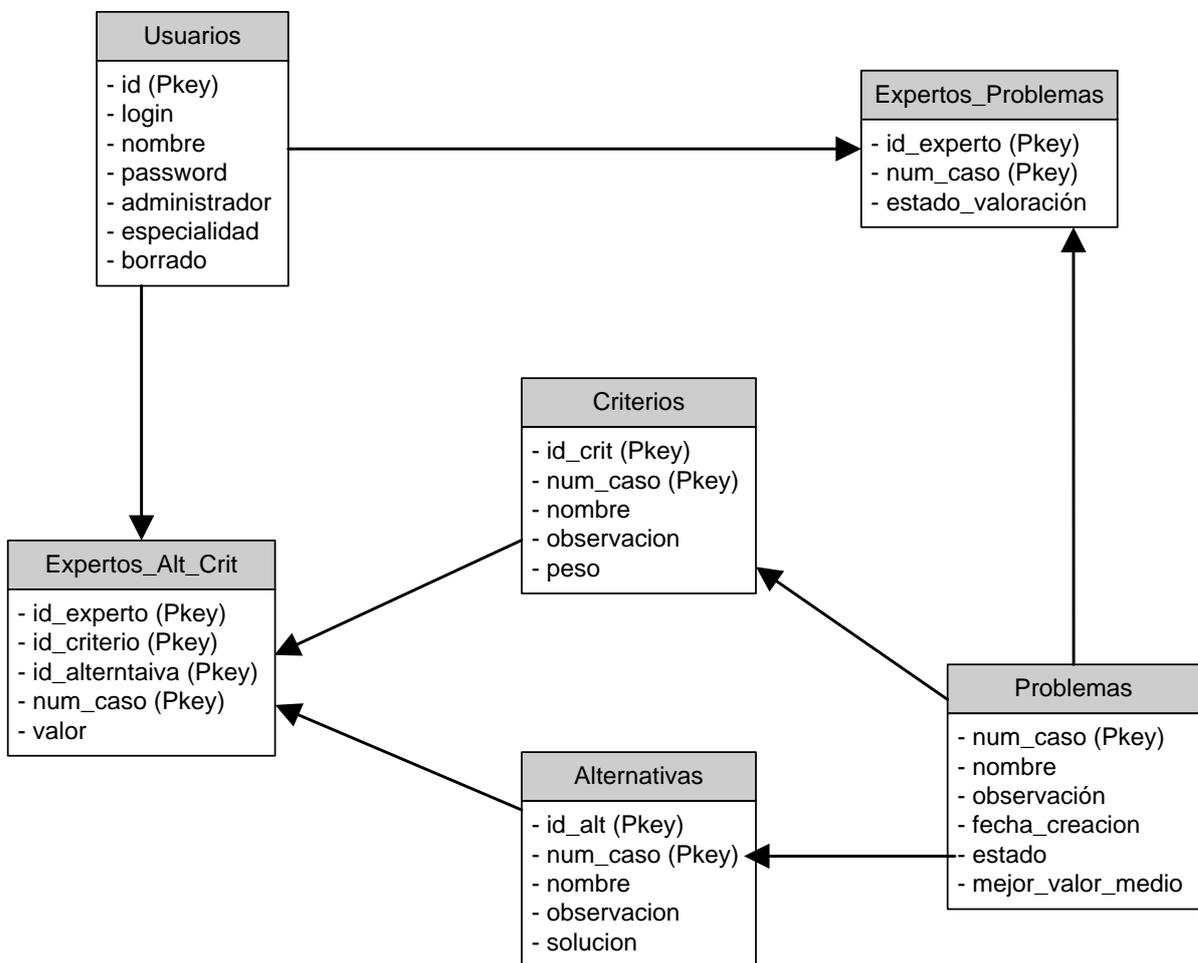


Figura 26. Esquema Conceptual Modificado E-R

3.4.3.5 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: *Usuarios*, *Problemas*, *Alternativas*, *Criterios*, *Expertos_Problemas*, *Expertos_Alt_Crit*. A continuación se detallan cada una de estas tablas.

- **Usuarios:** Tabla que contiene el conjunto de usuarios, tanto administradores (gestionan el programa) como expertos (consultan y valoran los problemas), del sistema.

| Campo | Tipo | Descripción | Clave | Requerido |
|----------------------|--------------|--|-------|-----------|
| id | Integer | Identificador de usuario | * | Si |
| nombre | Varchar(255) | Nombre del usuario | | Si |
| login | Varchar(150) | Login de acceso de usuario | | Si |
| password | Varchar(32) | Contraseña de acceso de usuario | | Si |
| administrador | Byte | Identificador de administrador o experto | | Si |
| especialidad | Varchar(255) | Especialidad del usuario | | Si |
| borrado | Byte | Identificador de borrado lógico de usuario | | Si |

Tabla 14. Tabla Usuario

- **Problemas:** Tabla que contiene la información del conjunto de todos los problemas que existen en el sistema.

| Campo | Tipo | Descripción | Clave | Requerido |
|--------------------------|--------------|---|-------|-----------|
| num_caso | Integer | Identificador del problema | * | Si |
| nombre | Varchar(255) | Nombre del problema | | Si |
| observación | Texto | Descripción del problema | | Si |
| fecha_creacion | Fecha | Fecha de creación del problema | | Si |
| estado | Enumerado | Estado del problema; Pendiente o Resuelto | | Si |
| mejor_valor_medio | Integer | Identificador de mejor alternativa | | Si |

Tabla 15. Tabla Problemas

- **Alternativas:** Tabla que contiene el conjunto de alternativas, para los diferentes problemas, en el sistema.

num_caso es clave foránea de *num_caso* de la tabla Problemas.

| Campo | Tipo | Descripción | Clave | Requerido |
|--------------------|-------------|--------------------------------|--------------|------------------|
| id_alt | Integer | Identificador de alternativa | * | Si |
| num_caso | Integer | Número de caso de problema | * | Si |
| nombre | Varchar(50) | Nombre de la alternativa | | Si |
| observacion | Texto | Descripción del la alternativa | | Si |
| solucion | Double | Valor de la alternativa | | Si |

Tabla 16. Tabla Alternativas

- **Criterios:** Tabla que contiene el conjunto de criterios, para los diferentes problemas, en el sistema.

num_caso es clave foránea de *num_caso* de la tabla Problemas.

| Campo | Tipo | Descripción | Clave | Requerido |
|--------------------|-------------|----------------------------|--------------|------------------|
| id_crit | Integer | Identificador del criterio | * | Si |
| num_caso | Integer | Número de caso de problema | * | Si |
| nombre | Varchar(50) | Nombre del criterio | | Si |
| observacion | Texto | Descripción del criterio | | Si |
| peso | Integer | Peso del criterio | | Si |

Tabla 18. Tabla Criterios

- **Expertos_Problemas:** Tabla que contiene la vinculación que existe entre experto y problema y el estado en el que se encuentra un experto respecto un problema concreto al que está vinculado en el sistema.

id_experto es clave foránea de *id* de la tabla Usuarios.

num_caso es clave foránea de *num_caso* de la tabla Problemas.

| Campo | Tipo | Descripción | Clave | Requerido |
|--------------------------|-------------|--|--------------|------------------|
| id_experto | Integer | Identificador del experto | * | Si |
| num_caso | Integer | Identificador del problema | * | Si |
| estado_valoración | Enumerado | Estado de la valoración; Pendiente, Valorado, Resuelto o Resuelto (No Valorado) | | Si |

Tabla 19. Tabla Experto_Problemas

- **Expertos_Alt_Crit:** Tabla que contiene las valoraciones de los expertos para un determinado criterio y una determinada alternativa, relacionados a su vez con un problema concreto, en el sistema.

id_experto es clave foránea de *id* de la tabla Usuarios.

id_criterio es clave foránea de *id_crit* de la tabla Criterios.

id_alternativa es clave foránea de *id_alt* de la tabla Alternativas.

num_caso es clave foránea de *num_caso* de la tabla Problemas.

| Campo | Tipo | Descripción | Clave | Requerido |
|-----------------------|-------------|---|--------------|------------------|
| id_experto | Integer | Identificador del experto | * | Si |
| id_criterio | Integer | Identificador del criterio | * | Si |
| id_alternativa | Integer | Identificador de alternativa | * | Si |
| num_caso | Integer | Identificador del problema | * | Si |
| valor | Double | Valoración del experto en una alternativa y un criterio concretos | | Si |

Tabla 20. Tabla Experto_Alt_Crit

3.4.4 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 resultar fácil de entender y trabajar sobre ella.

La ingeniería de la usabilidad aporta importantes beneficios referentes a los costes de desarrollo, la calidad del producto y la satisfacción del cliente. Define en esta fase los siguientes aspectos:

- Definir estilo
- Metáforas
- Pantallas
- Caminos de navegación
- Secuencias de dialogo
- Mensajes de error

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

3.4.4.1 *Estilo de la Interfaz*

Antes de diseñar una interfaz de usuario, se debe definir el estilo de la misma. Ésto 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. Ésto 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 especificamos que la guía de estilo utilizada por nuestra aplicación de escritorio es el estilo por defecto de Java para las versiones de Windows más actuales. Se usa el tema Vista Aero que es el tema común a las ventanas de Windows. Nuestra aplicación tendrá:

- Fuentes:
 - Los títulos y cabeceras: Serif. Si es un título principal va en negrita a 18.
 - Botones y texto: Arial.
- Colores:
 - La aplicación usa el color gris RGB(238, 238, 238), por defecto del tema Vista Aero.
 - Los botones usan un degradado entre blanco RGB (255, 255, 255) y un azul RGB(217, 229, 241).
- Diseño de las ventanas:
 - Todas las ventanas llevarán un título que indique donde se encuentra el usuario, y en determinadas veces, cuál es la tarea que se llevará a cabo en dicha ventana.

La estructura será similar para un mejor aprendizaje de la herramienta y siempre se utilizarán las mismas imágenes para los botones que tengan la misma funcionalidad.

3.4.4.2 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 de éste 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).

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

3.4.4.3 Pantallas

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

El diseño gráfico de la interfaz de usuario juega un papel fundamental, ya que una buena distribución, ordenación, y codificación de colores de los elementos de la interfaz hace que ésta sea más clara y por tanto más fácil de usar para el usuario. Por el contrario una colocación incorrecta de los elementos puede hacer muy compleja la utilización de una aplicación por parte de un usuario.

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.

Una interfaz de usuario visualmente atractiva hace que sea percibida por el usuario como una mejor interfaz, mientras que una interfaz compleja con sus elementos desordenados, provoca en el usuario insatisfacción y desorientación. No hay que olvidar que una buena interfaz es aquella que encuentra el equilibrio entre la funcionalidad y la estética.

Los prototipos son una forma de crear una imagen palpable de lo que será una futura aplicación de escritorio. 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 o bocetos rápidos y sin esfuerzo en escribir código.

Los prototipos nos ayudarán a ver que los contenidos de las pantallas 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 serán:

- *El administrador*: es la persona que dirige la sesión y deberá tener conocimientos de usabilidad y diseño centrado en el usuario.
- *Usuarios (los expertos)*: 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 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.
- *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 el 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:

- Pantalla de acceso de usuarios (para acceso a administradores y expertos)
- Pantalla principal Administrador (para problemas y expertos)
- Pantalla nuevo problema
- Pantalla modificar problema
- Pantalla consulta problema resuelto
- Pantalla nueva alternativa
- Pantalla ver alternativa
- Pantalla modificar alternativa
- Pantalla nuevo criterio
- Pantalla ver criterio
- Pantalla modificar criterio
- Pantalla vincular experto al problema
- Pantalla nuevo usuario
- Pantalla problema resuelto
- Pantalla gestión Experto
- Pantalla consulta experto información general problema

- Pantalla valoración y modificación de valoración de experto
- Pantalla ver solución problema de experto
- Pantalla cambio información experto
- Pantalla cambio contraseña experto

A continuación se mostrarán algunas pantallas que se han construido.

Pantalla de acceso de usuarios: Se puede observar fácilmente en la Figura 27 que nos encontramos en la página principal de acceso, en la que hay que autenticarse como usuario *Experto* o *Administrador*.

The image shows a login interface titled "Bienvenido al programa de Toma de Decisión Multicriterio". It contains two input fields: "Usuario:" and "Contraseña:". Below the fields are two buttons: "Acceder" and "Salir".

Figura 27. Pantalla acceso usuario

Pantalla principal Administrador: En esta pantalla el administrador puede controlar la creación, consulta, modificación y borrado de los diferentes problemas y expertos. Se puede ver en la Figura 28.

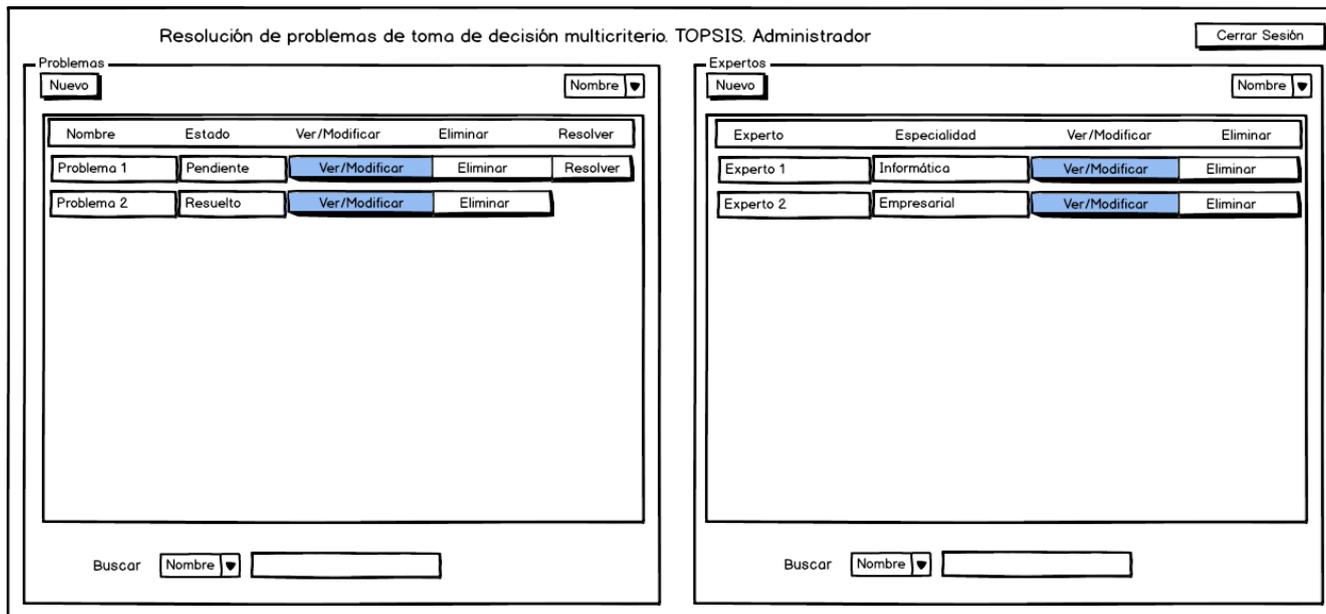


Figura 28. Pantalla principal Administrador

Pantalla nuevo problema: En esta pantalla el administrador introducirá toda la información referente a un problema nuevo. Se puede ver que también se pueden crear alternativas y criterios para el problema y vincular a expertos que luego podrán valorarlo. Ver Figura 29.

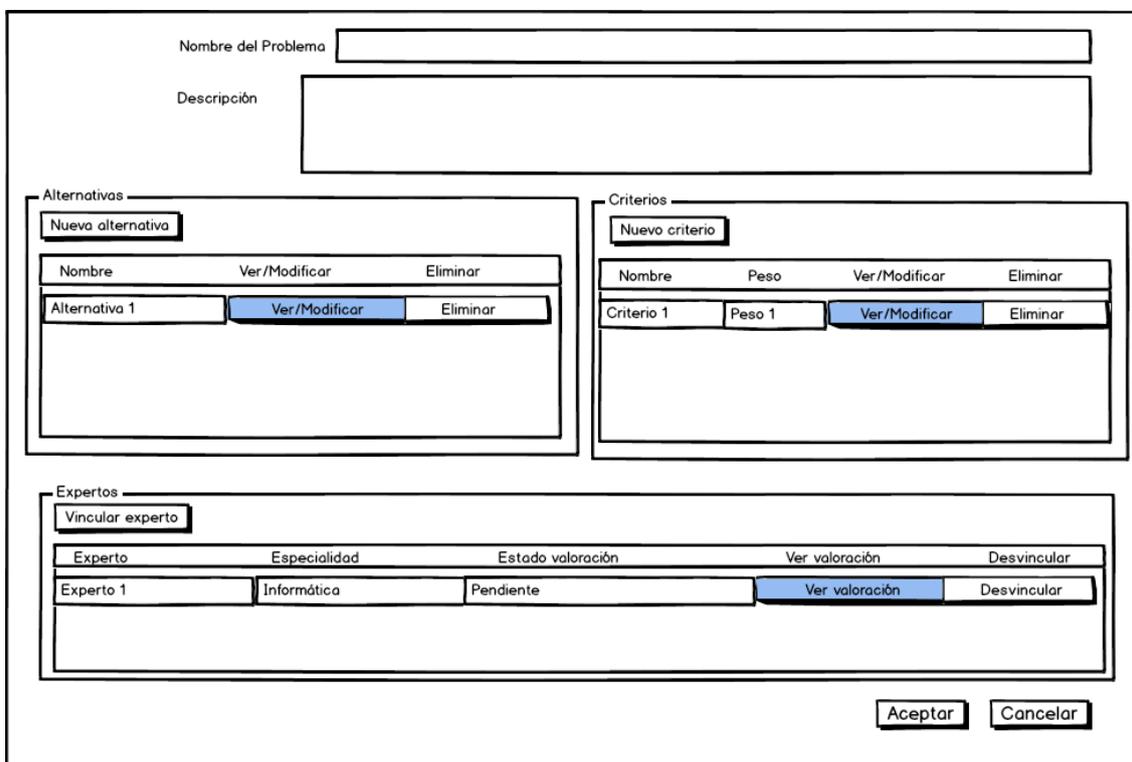
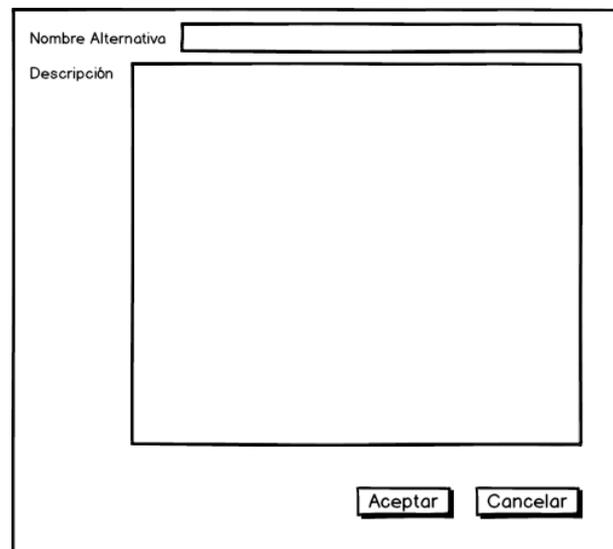


Figura 29. Pantalla nuevo problema

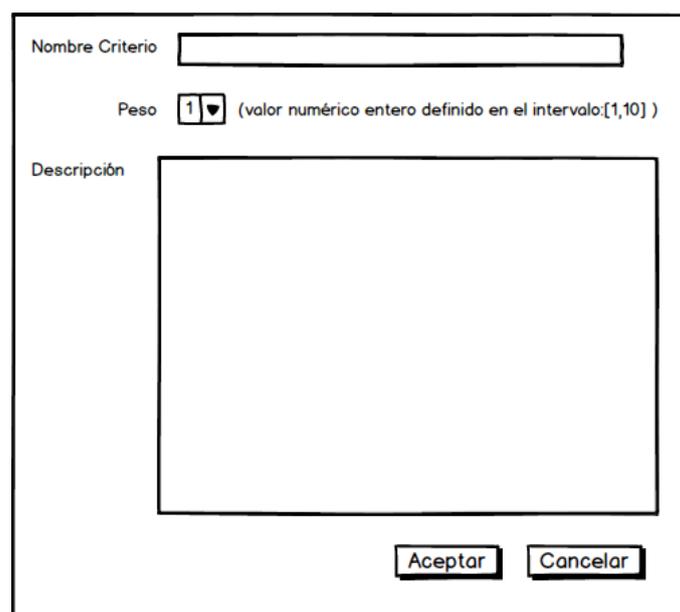
Pantalla nueva alternativa: Pantalla que el administrador rellena con la información de la nueva alternativa. Esta pantalla es similar para la pantalla de **ver** y **modificar** la información de la alternativa (en la pantalla de **ver** hay solo un botón, en vez de dos, "Cerrar"). Se puede ver en la Figura 30.



The screenshot shows a web form for creating a new alternative. It features a text input field for 'Nombre Alternativa' and a larger text area for 'Descripción'. At the bottom right, there are two buttons: 'Aceptar' and 'Cancelar'.

Figura 30. Pantalla nueva alternativa

Pantalla nuevo criterio: Pantalla que el administrador rellena con la información del nuevo criterio. Esta pantalla es similar para la pantalla de **ver** y **modificar** la información del criterio (en la pantalla de **ver** hay solo un botón, en vez de dos, "Cerrar"). Se puede ver en la Figura 31.



The screenshot shows a web form for creating a new criterion. It includes a text input field for 'Nombre Criterio', a numeric input field for 'Peso' (with a value of 1 and a dropdown arrow), and a text area for 'Descripción'. A note next to the 'Peso' field states '(valor numérico entero definido en el intervalo:[1,10])'. At the bottom right, there are two buttons: 'Aceptar' and 'Cancelar'.

Figura 31. Pantalla nuevo criterio

Pantalla vincular experto al problema: Pantalla que el administrador utilizará para asignarle expertos al problema. Ver Figura 32.

| Experto | Especialidad | Asociar |
|-----------|--------------|----------|
| Experto 1 | Informática | Vincular |

Cerrar

Figura 32. Pantalla vincular experto

Pantalla modificar información de un problema: Como se puede observar en la Figura 33, esta pantalla muestra toda la información que el administrador puede modificar en un problema.

Nombre del Problema

Descripción

Alternativas

| Nombre | Ver/Modificar | Eliminar |
|---------------|---------------|----------|
| Alternativa 1 | Ver/Modificar | Eliminar |

Criterios

| Nombre | Peso | Ver/Modificar | Eliminar |
|------------|--------|---------------|----------|
| Criterio 1 | Peso 1 | Ver/Modificar | Eliminar |

Expertos

| Experto | Especialidad | Estado valoración | Ver valoración | Desvincular |
|-----------|--------------|-------------------|----------------|-------------|
| Experto 1 | Informática | Pendiente | Ver valoración | Desvincular |

Figura 33. Pantalla modificar problema

Pantalla consulta problema resuelto y pantalla de problema resuelto: En esta pantalla, de la Figura 34, aparece toda la información de un problema ya resuelto por el administrador. La información en dicha pantalla no puede ser modificada.

The screenshot displays a web interface for viewing a solved problem. It is organized into several sections:

- Problem Information:**
 - Nombre del Problema: Problema 1
 - Descripción: Esta es la descripción del problema 1 con todos sus datos
- Alternativas:** A table with columns 'Nombre' and 'Ver'.

| Nombre | Ver |
|---------------|-----|
| Alternativa 1 | Ver |
- Criterios:** A table with columns 'Nombre', 'Peso', and 'Ver'.

| Nombre | Peso | Ver |
|------------|--------|-----|
| Criterio 1 | Peso 1 | Ver |
- Expertos:** A table with columns 'Experto', 'Especialidad', 'Estado valoración', and 'Ver valoración'.

| Experto | Especialidad | Estado valoración | Ver valoración |
|-----------|--------------|-------------------|----------------|
| Experto 1 | Informática | Pendiente | Ver valoración |
- Valoraciones Medias:** A placeholder for a table: (Tabla de valoraciones medias según el modelo TOPSIS)
- Resolución del problema:** Three input fields for 'Mejor alternativa n°1', 'Mejor alternativa n°2', and 'Mejor alternativa n°3'.
- Buttons:** A 'Cerrar' button is located at the bottom right.

Figura 34. Pantalla problema resuelto

Pantalla nuevo experto: Según la Figura 35, en esta pantalla el administrador crea un nuevo experto introduciendo su información.

The screenshot shows a registration form with the following elements:

- Labels: Login, Nombre, Contraseña, Especialidad
- Input fields: Four horizontal text boxes corresponding to the labels.
- Buttons: Two buttons labeled 'Aceptar' and 'Cancelar' at the bottom right.

Figura 35. Pantalla nuevo experto

Pantalla gestión experto: Como se puede observar en la Figura 36, en esta pantalla el experto puede ver los problemas que tiene asignados para valorar o consultar sus resultados, en caso de estar resueltos.

The screenshot shows a dashboard titled 'Resolución de problemas de toma de decisión multicriterio. TOPSIS. Experto'. It contains the following elements:

- Header: 'Resolución de problemas de toma de decisión multicriterio. TOPSIS. Experto'
- Left Panel:
 - 'Bienvenido' message
 - 'Perfil Usuario' button
 - 'Cerrar sesión' button
 - 'Ordenar por: Nombre' dropdown menu
 - 'Problemas' table with columns: Nombre, Estado, Ver
 - 'Buscar' section with a 'Nombre' dropdown and a search input field.
- Right Panel: A large empty rectangular area.

Figura 36. Pantalla gestión experto

Pantalla consulta de experto de información general de un problema: En esta pestaña, de la pantalla, el experto puede ver la información general de un problema asignado. Ver Figura 37.

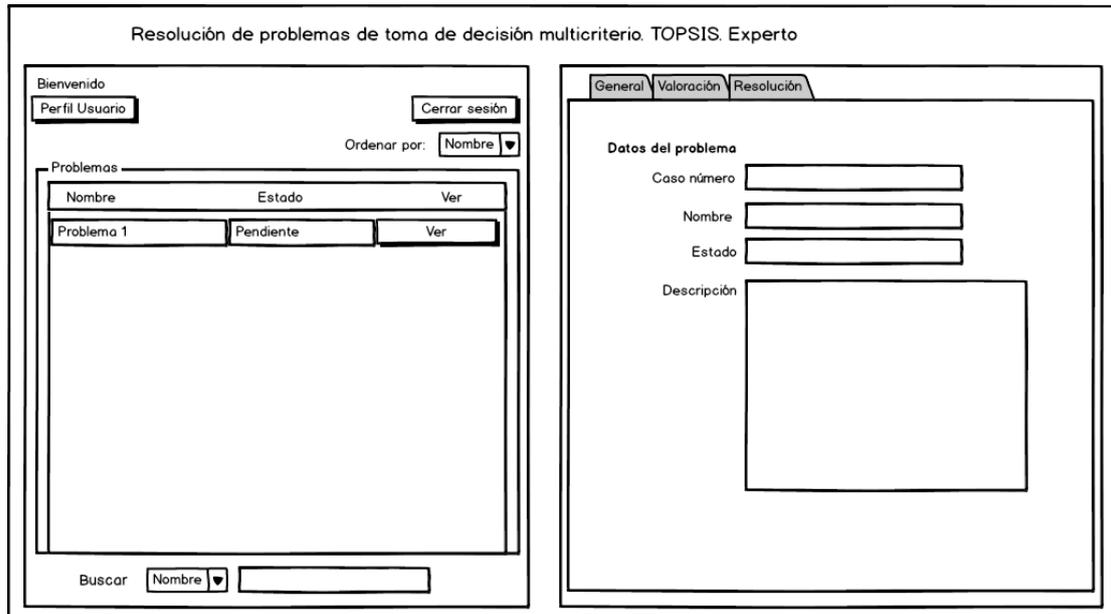


Figura 37. Consulta experto información problema

Pantalla de valoración de experto: En esta pestaña de la pantalla, el experto puede realizar la valoración numérica asignando valores a las diferentes combinaciones de alternativas y criterios de un problema asignado. Esta pestaña será de solo consulta cuando el problema esté resuelto, pudiendo así el experto ver su valoración pasada (ver Figura 38). Ésta es similar para la **pantalla modificación de valoración** de un experto cuando añade valoraciones que no había introducido o modifica las ya realizadas.

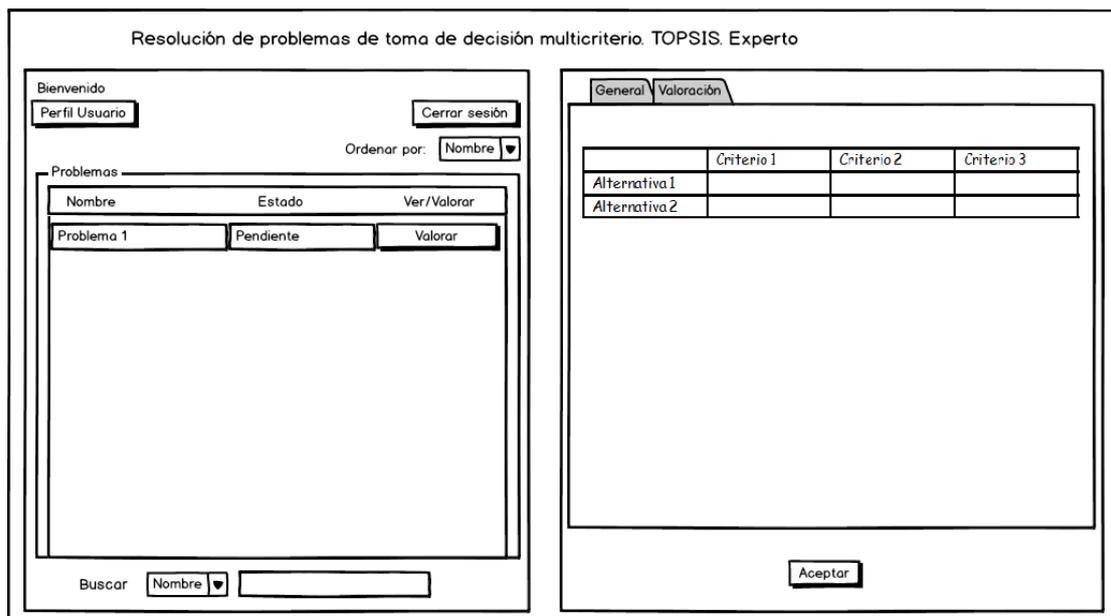


Figura 38. Pantalla valoración experto

Pantalla ver solución del problema de un experto: En esta pestaña, de la pantalla, el experto puede consultar cual ha sido la solución final del problema una vez que se le aplicó el método de resolución TOPSIS. El experto puede consultar cuál es la mejor alternativa del problema asignado. Se puede observar en la Figura 39.

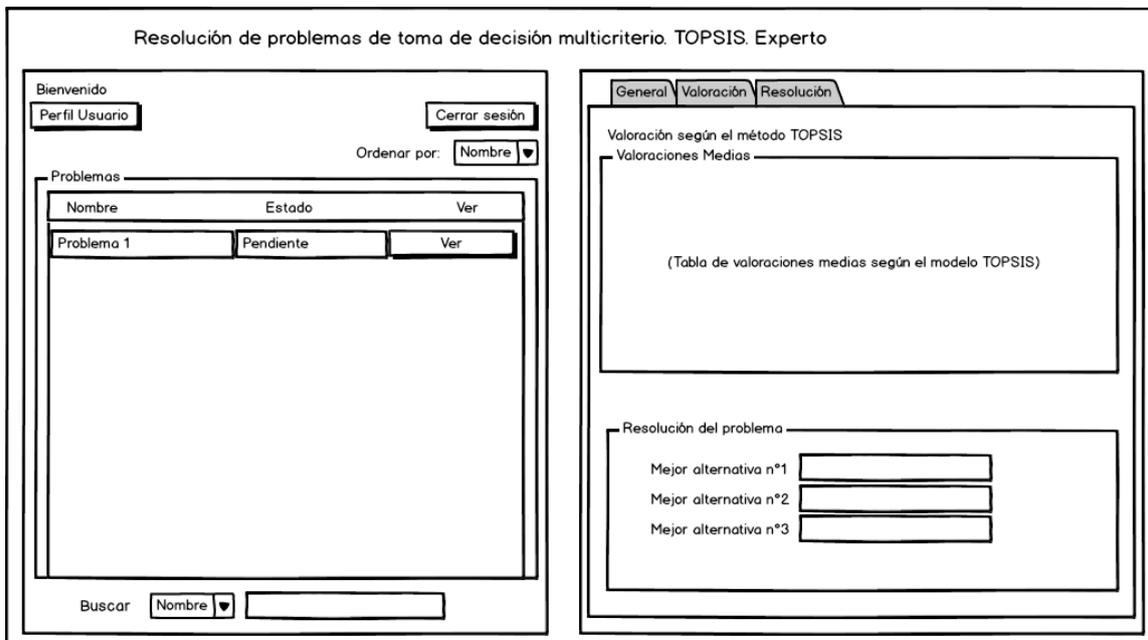


Figura 39. Ver solución problema experto

Pantalla cambio información experto: En esta pantalla el experto puede modificar su información básica. Ver Figura 40.

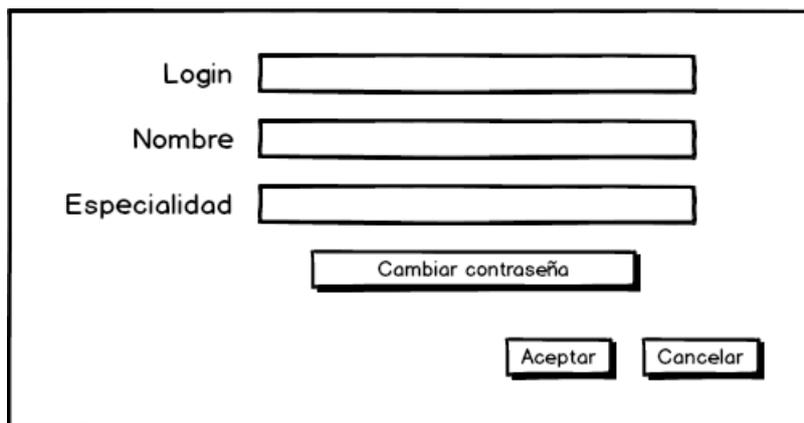
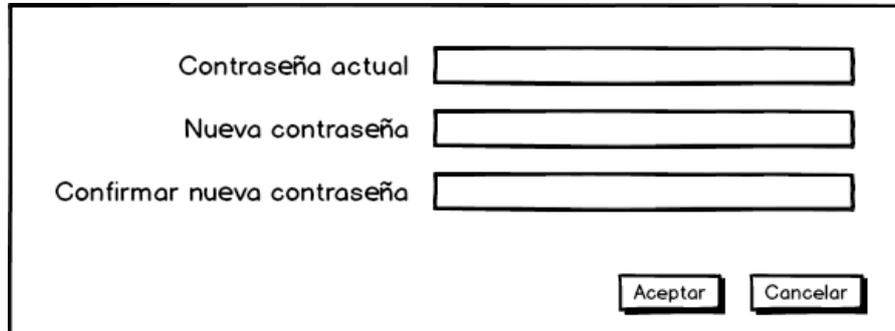


Figura 40. Pantalla cambio información experto

Pantalla cambio contraseña experto: En esta pantalla el experto puede modificar su contraseña o password. Ver Figura 41.



El diagrama muestra una interfaz de usuario para cambiar la contraseña. Está organizada en tres filas de entrada de texto, cada una con un label a la izquierda: 'Contraseña actual', 'Nueva contraseña' y 'Confirmar nueva contraseña'. En la parte inferior derecha del cuadro, hay dos botones rectangulares: 'Aceptar' y 'Cancelar'.

Figura 41. Pantalla cambio contraseña experto

3.4.4.4 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 qué 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 *storyboards* nos facilitan esta tarea ya que nos permiten ver juntas todas las pantallas asociadas a una tarea o escenario.

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 Crear un experto (Administrador)
- Storyboard Resolver un problema (Administrador)
- Storyboard Cambiar contraseña experto (Experto)
- Storyboard Valorar un problema (Experto)
- Storyboard Consultar problema resuelto (Experto)

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

Storyboard Crear un problema (Administrador)

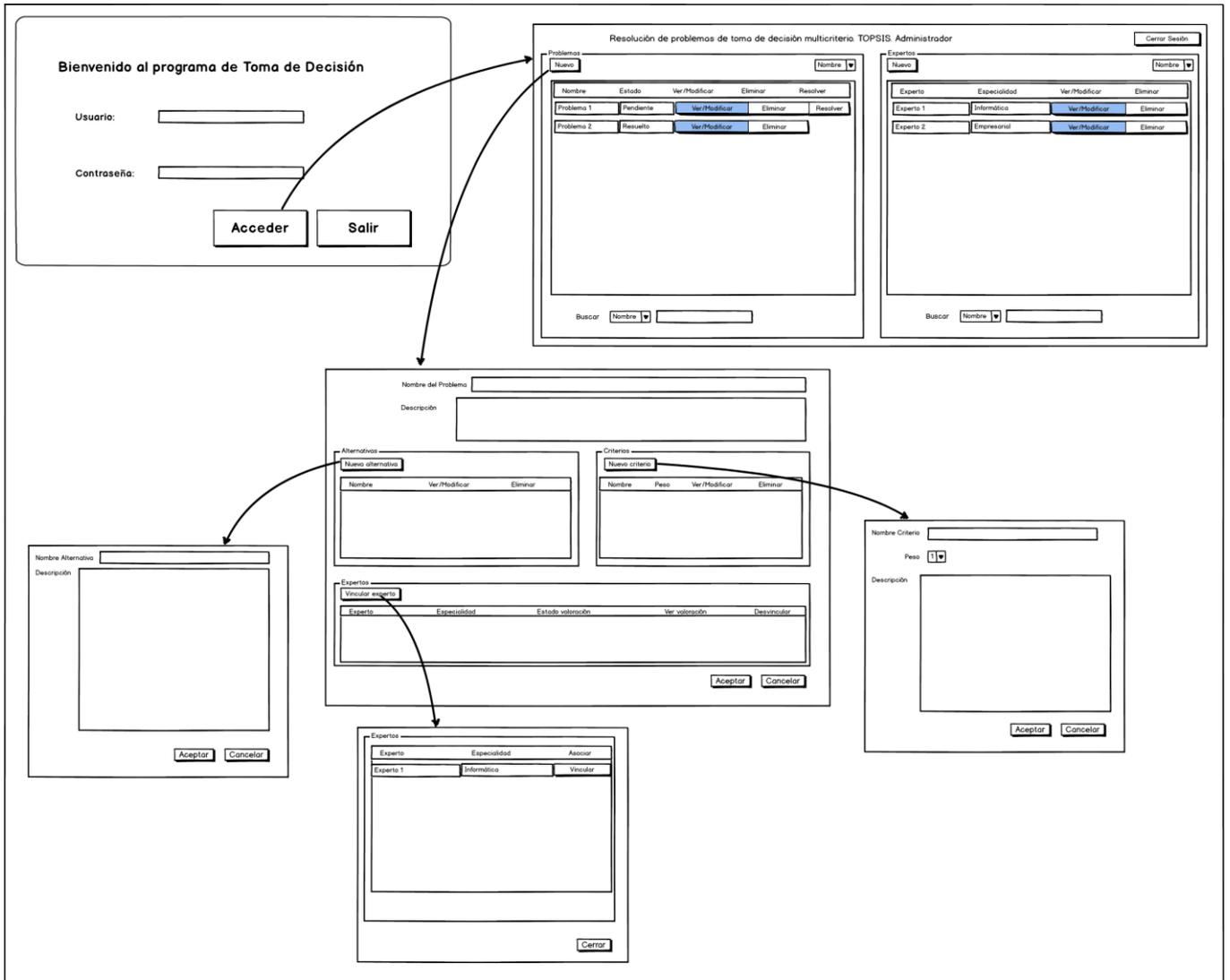


Figura 42. Storyboard crear problema

Storyboard Crear un experto (Administrador)

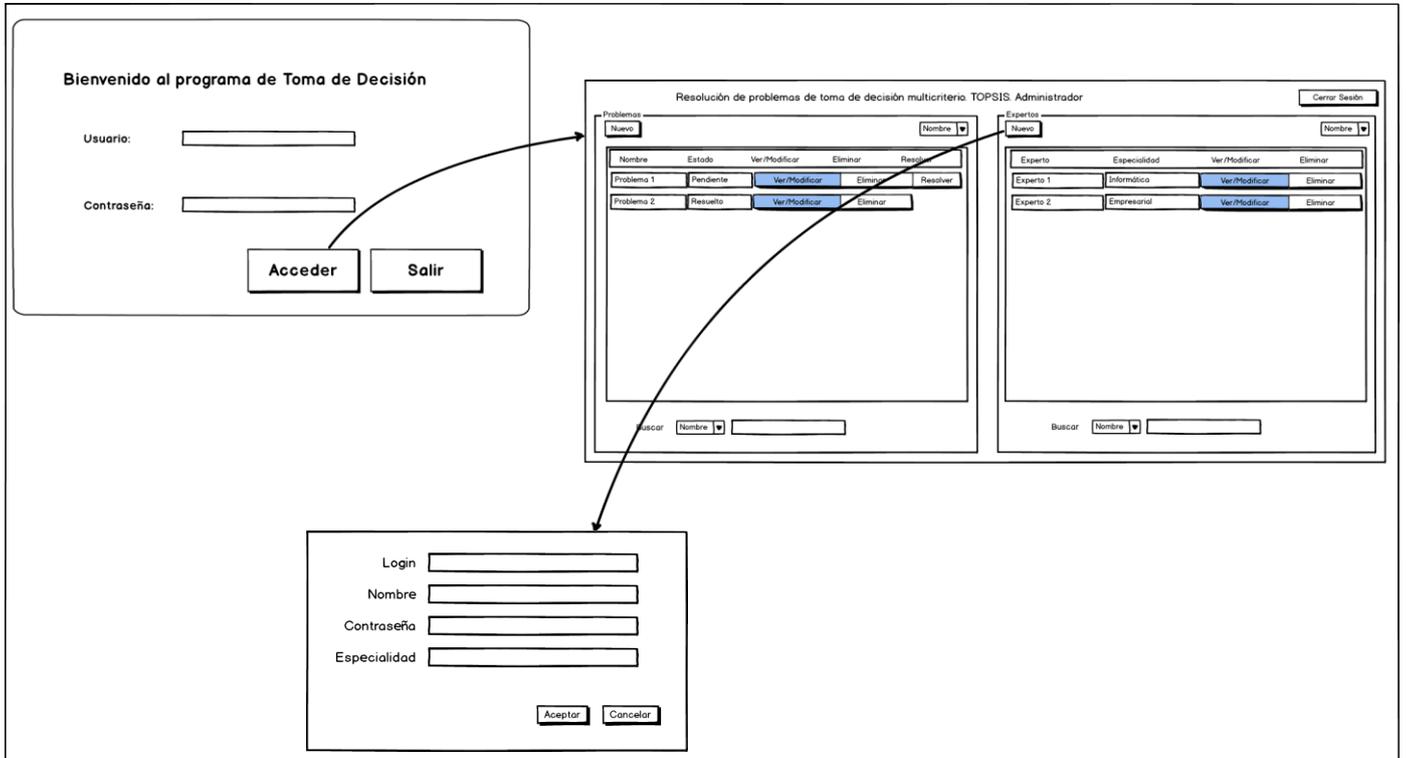


Figura 43. Storyboard crear un experto

Storyboard Resolver un problema (Administrador)

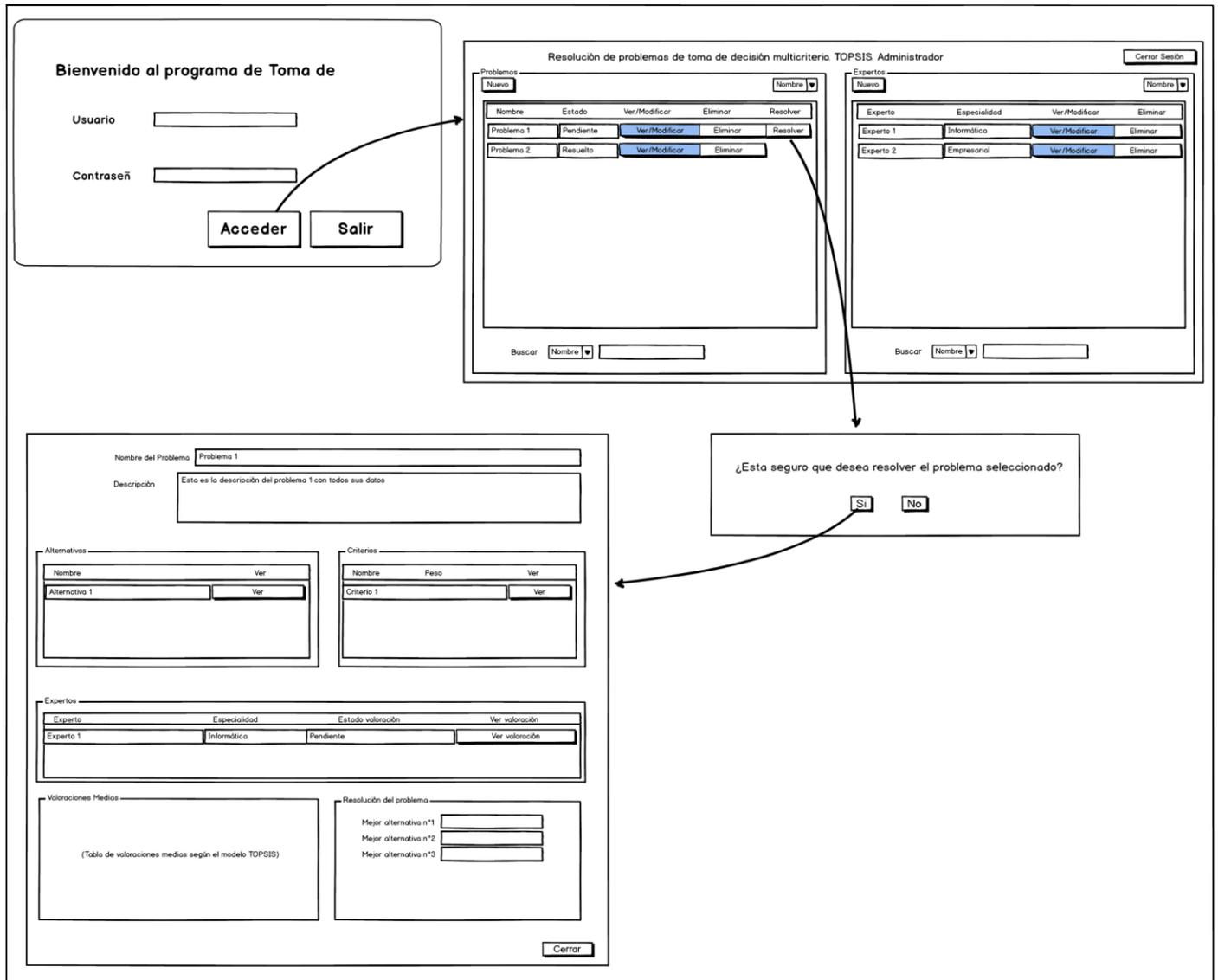


Figura 44. Storyboard resolver un problema

Storyboard Cambiar contraseña experto (Experto)

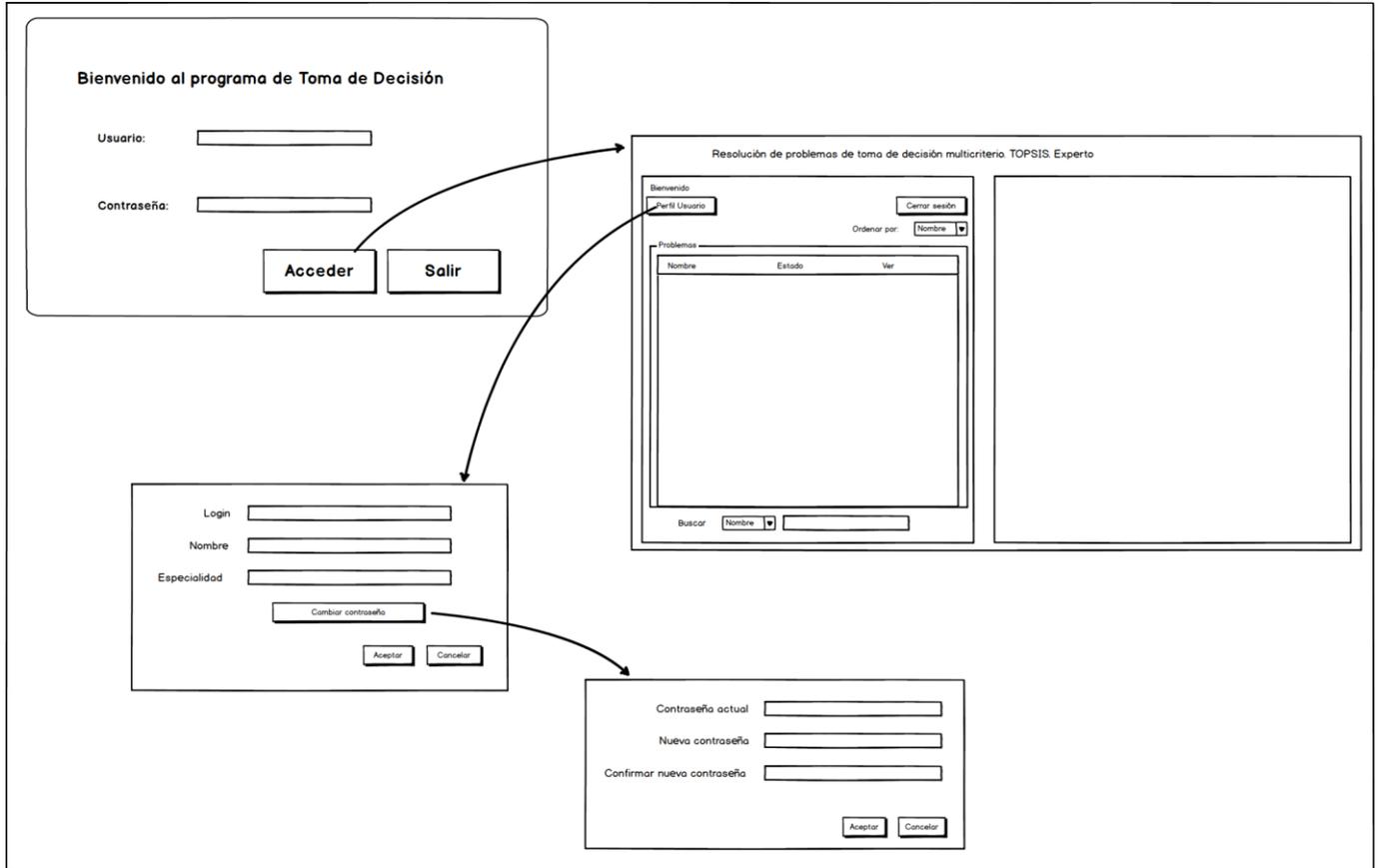


Figura 45. Storyboard cambiar contraseña experto

Storyboard Valorar un problema (Experto)

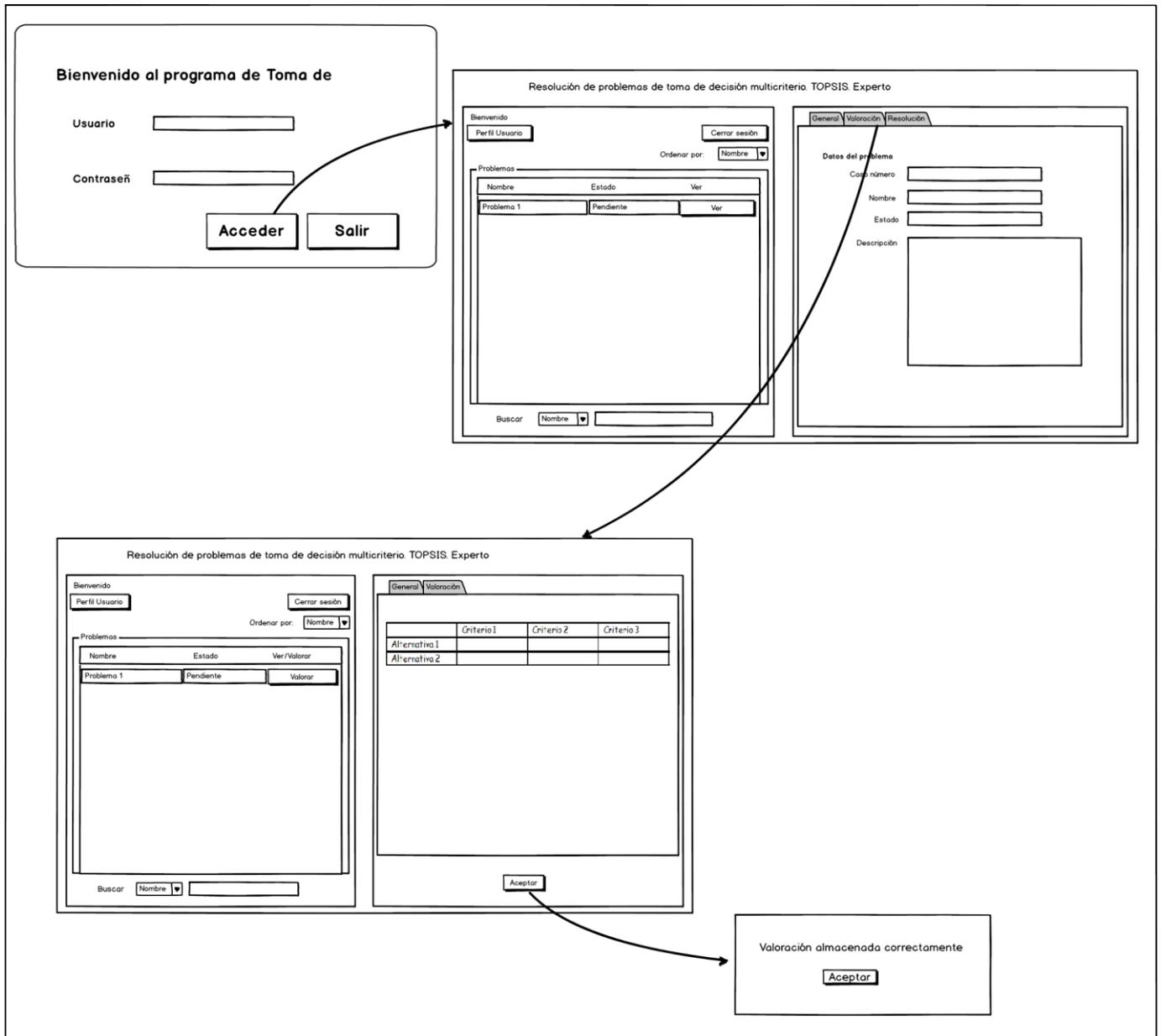


Figura 46. Storyboard valorar un problema

Storyboard Consultar problema resuelto (Experto)

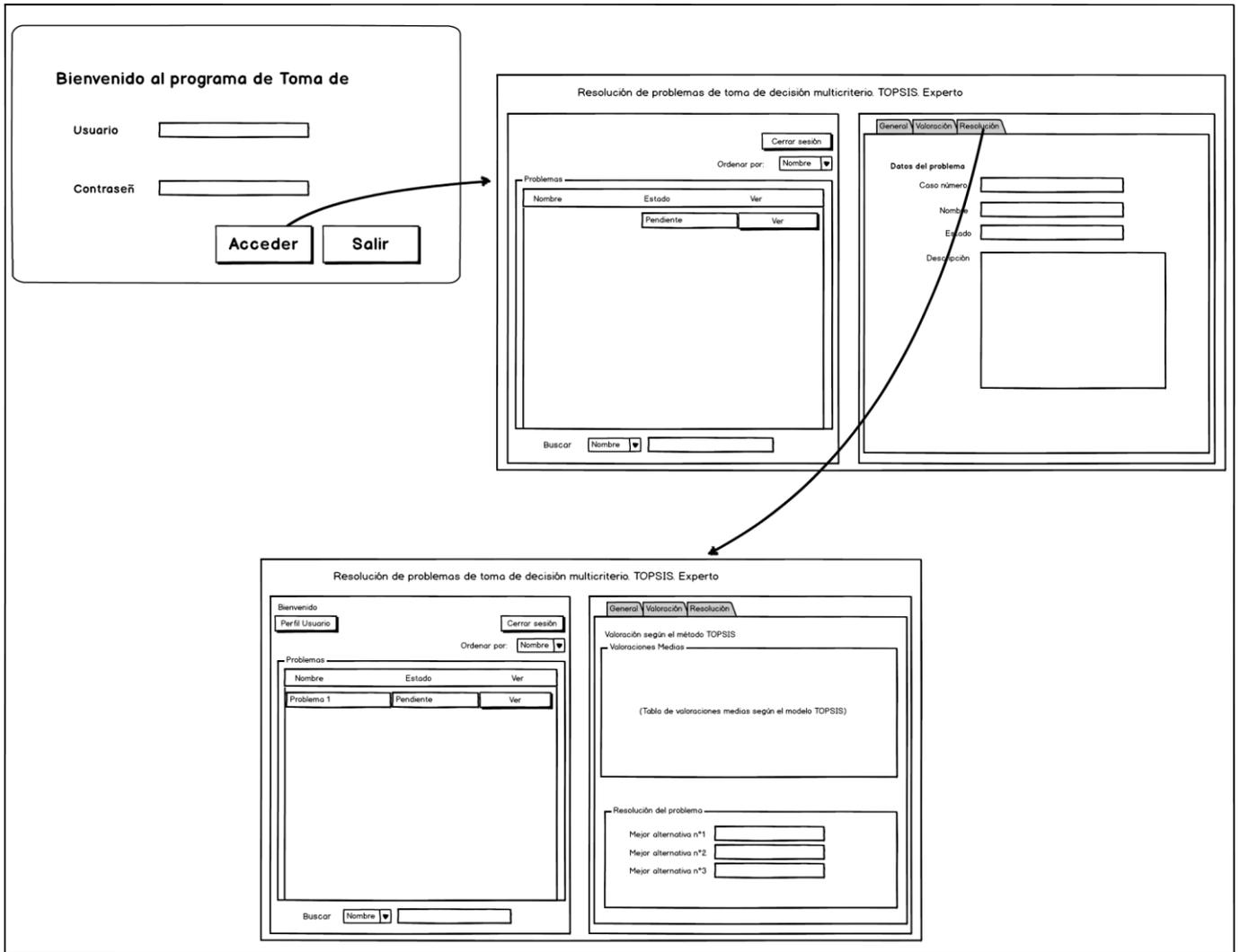


Figura 47. Consultar problema resuelto

3.4.4.5 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 las diferentes pantallas creadas anteriormente, e iremos analizando los mensajes que pueden aparecer en cada pantalla. Los mensajes detectados son los siguientes:

Pantalla de acceso de usuarios: Existe la posibilidad de que el usuario se identifique de forma incorrecta o no se identifique.



Figura 48.1 y Figura 48.2. Mensaje error acceso

Pantalla nuevo problema y pantalla modificar problema: Existe la posibilidad de crear un nuevo problema de forma incorrecta.

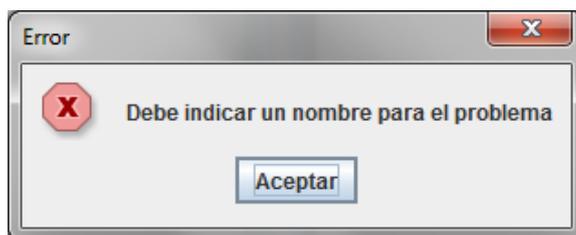


Figura 49.1. Mensaje de error crear o modificar problema

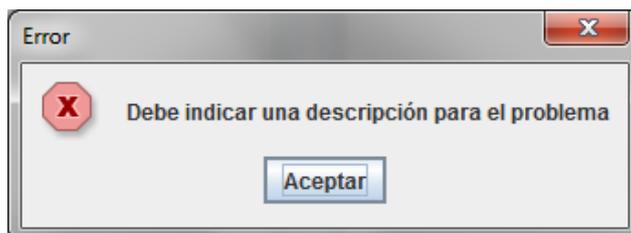


Figura 49.2. Mensaje de error crear o modificar problema

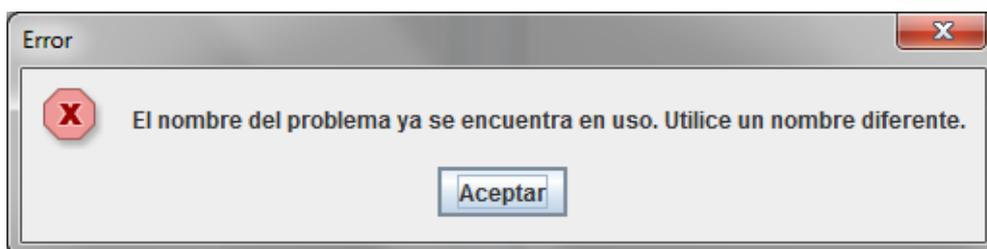


Figura 49.3. Mensajes error crear o modificar problema

Pantalla nueva alternativa y pantalla modificar alternativa: Existe la posibilidad de crear un nueva alternativa de forma incorrecta.

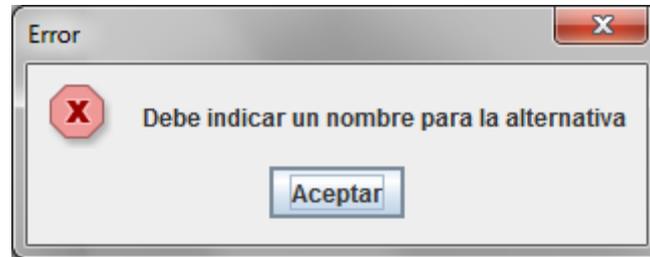


Figura 50.1. Mensaje error crear o modificar alternativa

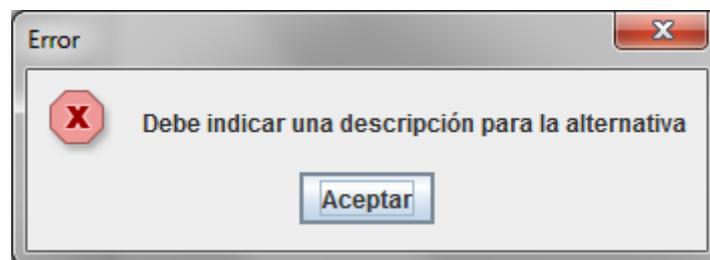


Figura 50.2. Mensaje error crear o modificar alternativa

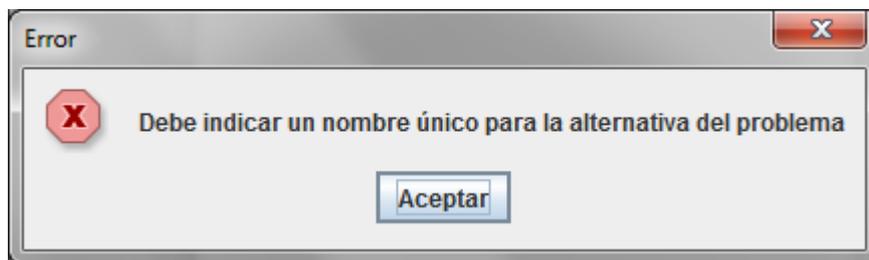


Figura 50.3. Mensaje error crear o modificar alternativa repetida

Pantalla nuevo criterio y pantalla modificar criterio: Existe la posibilidad de crear un nuevo criterio de forma incorrecta.

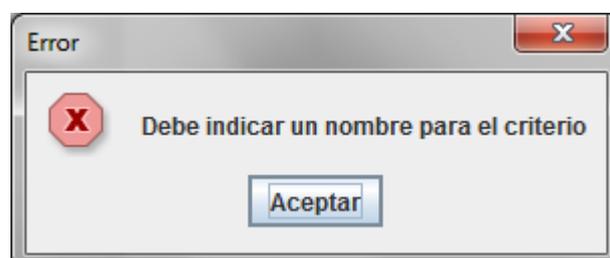


Figura 51.1. Mensaje error crear o modificar criterio

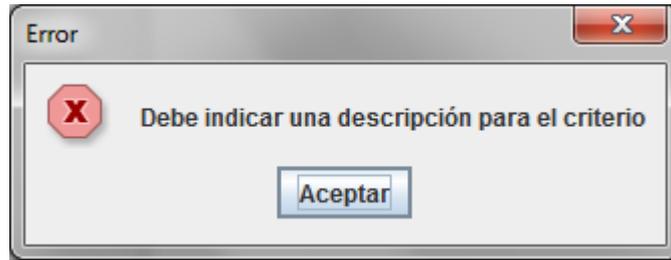


Figura 51.2. Mensaje error crear o modificar criterio

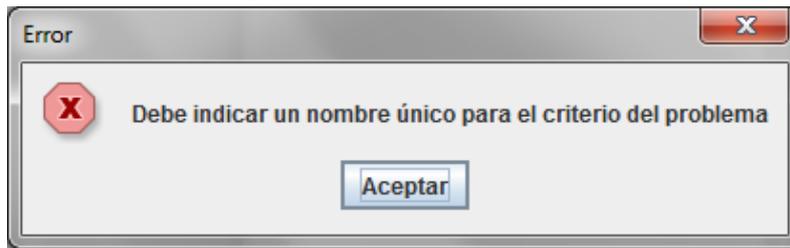


Figura 51.3. Mensaje error crear o modificar criterio repetido

Pantalla nuevo experto y pantalla modificar experto: Existe la posibilidad de crear un nuevo experto de forma incorrecta.

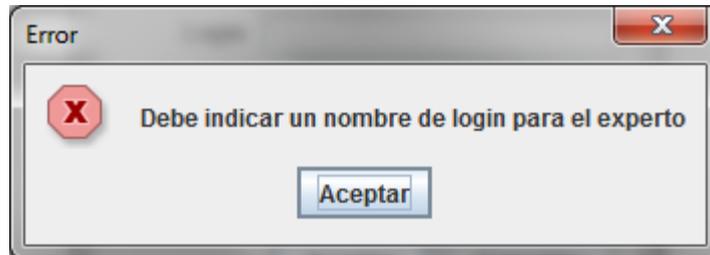


Figura 52.1. Mensaje error crear o modificar experto

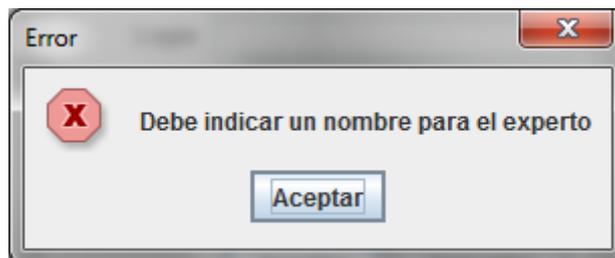


Figura 52.2. Mensaje error crear o modificar experto

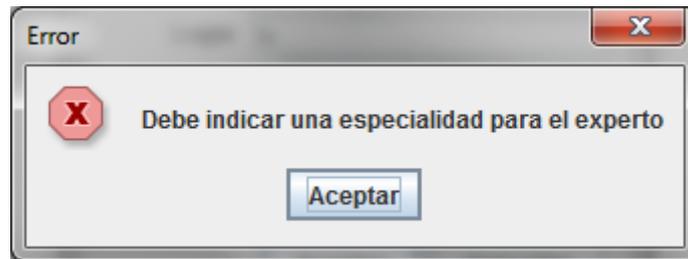


Figura 52.3. Mensaje error crear o modificar experto

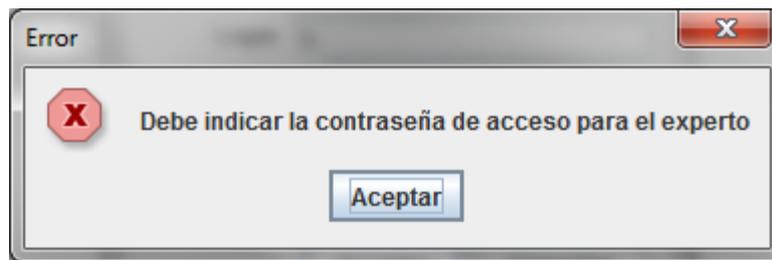


Figura 52.4. Mensaje error crear o modificar experto

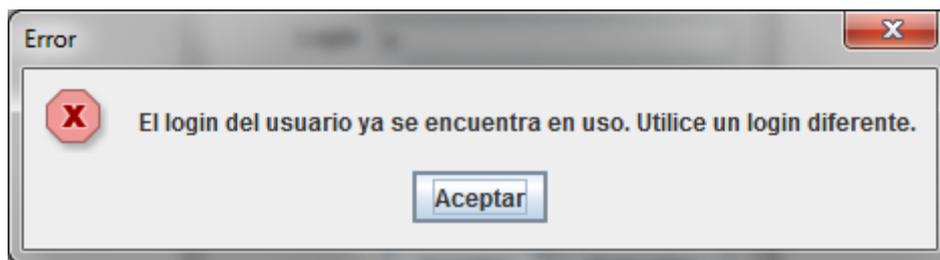


Figura 52.5. Mensaje error crear o modificar experto

Pantalla resolver un problema: Puede ocurrir que el problema no tenga el cincuenta por ciento de las valoraciones realizadas, por tanto, no se podrá resolver mediante el método TOPSIS.

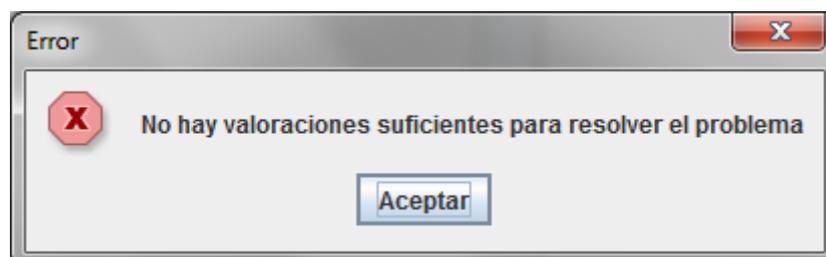


Figura 53. Mensaje error resolver problema

Pantalla cambio de información de experto: Cabe la posibilidad que al introducir los nuevos datos, el experto cometa incongruencias. Ver Figuras 54.

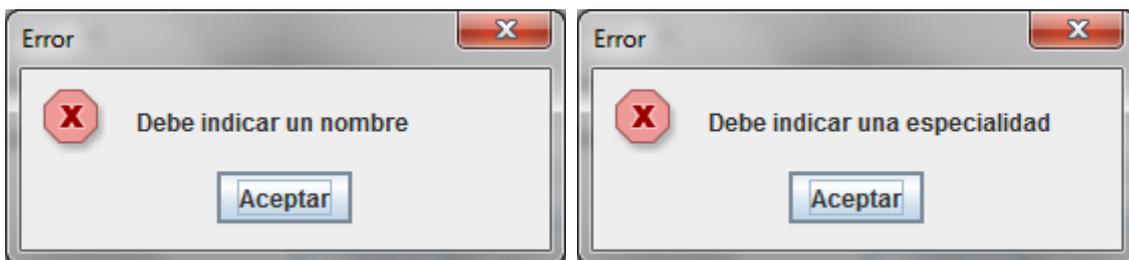


Figura 54.1 y Figura 54.2. Mensaje error cambio información experto

Pantalla cambio de contraseña de experto: Existe la posibilidad que al cambiar la contraseña el experto, lo haga de forma incorrecta. Ver Figuras 55.

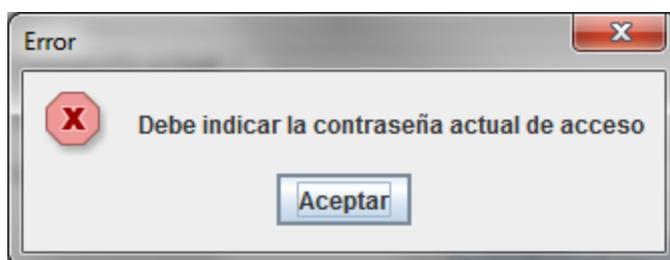


Figura 55.1. Mensaje error cambio de contraseña

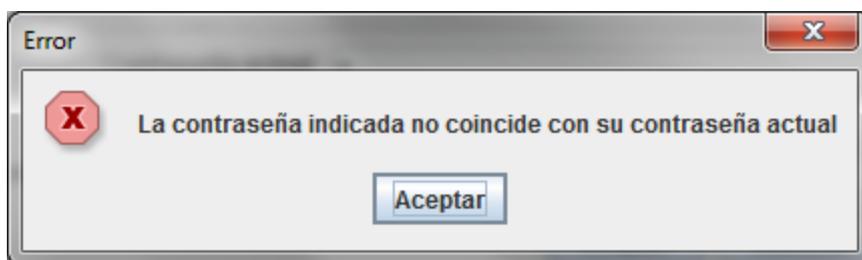


Figura 55.2. Mensaje error cambio de contraseña

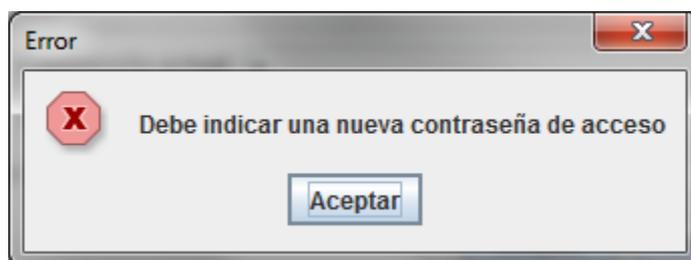


Figura 55.3. Mensaje error cambio de contraseña

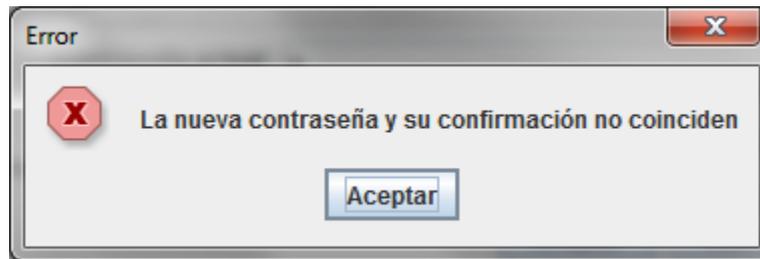


Figura 55.4. Mensaje error cambio de contraseña

3.4.4.6 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 el siguiente mensaje de éxito en la pantalla de valoración de un experto una vez introducidos los valores.

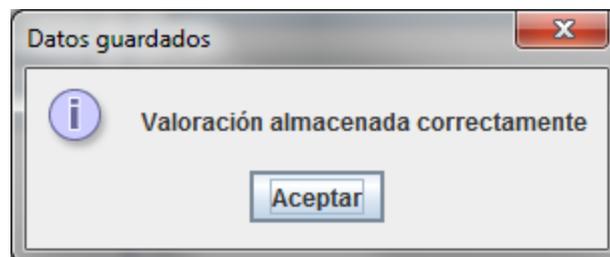


Figura 56. Mensaje éxito valorar problema

3.4.4.7 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:

Mensaje de confirmación cuando un problema va a ser borrado, ya que, una vez que se haya eliminado el problema, no se podrá recuperar. Ver Figura 57.

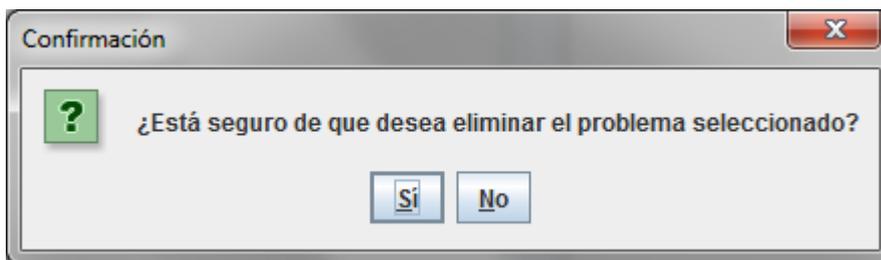


Figura 57. Mensaje confirmación eliminar problema

Mensaje de confirmación cuando un experto va a ser borrado, ya que, una vez que se haya eliminado el experto, no se podrá recuperar.

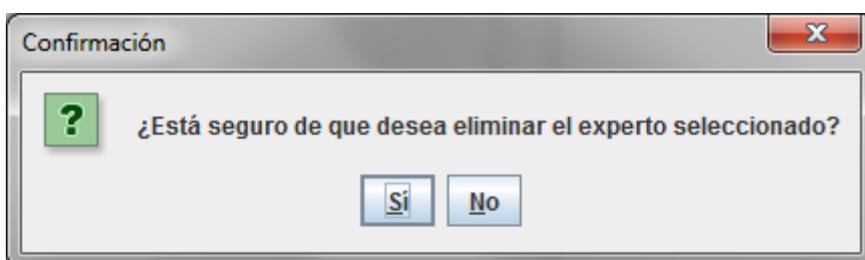


Figura 58. Mensaje confirmación eliminar experto

Mensaje de confirmación cuando se va a resolver un problema, ya que, una vez que se haya resuelto el problema, no se podrá volver a otro estado anterior.

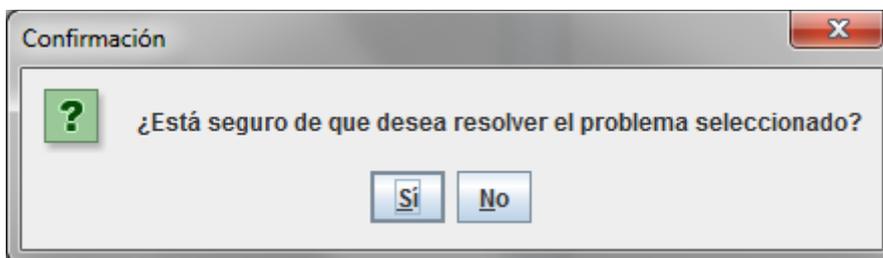


Figura 59. Mensaje confirmación resolver problema

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 arquitectura de la aplicación

En nuestro caso, vamos a desarrollar una aplicación de escritorio. El funcionamiento de las arquitecturas de este tipo es sencilla: la aplicación se encuentra instalada en el ordenador a la que los usuarios acceden de forma local, sin necesidad de disponer de una conexión a Internet ni acceder a un servidor local o externo.

3.5.2 Lenguajes de programación y tecnologías utilizadas

3.5.2.1 Java

Desarrollado en los laboratorios de Sun, es uno de los lenguajes de programación orientado a objetos que mayor repercusión ha tenido en los últimos años. Basado en C++ pero simplificado, mucho más fácil de usar, de más alto nivel y menos propenso a errores.

Cuenta con las siguientes características:

- Una amplísima biblioteca estándar de clases predefinidas.
- Las aplicaciones Java pueden ser ejecutadas indistintamente en cualquier plataforma sin necesidad de recompilación.
- Gestión avanzada de memoria mediante un recolector de basura.
- Gestión avanzada de errores, tanto en tiempo de compilación como de ejecución.
- Distribuido y multihebra.
- Lenguaje abierto. Paquete de desarrollo y documentación gratuitos en la red.

Existen distintas “ediciones” de Java para el desarrollo de aplicaciones en distintos ámbitos:

- Aplicaciones de propósito general (J2SE).
- Aplicaciones de gestión en entornos empresariales (J2EE).
- Aplicaciones para teléfonos móviles, PDAs y otros dispositivos electrónicos que permitan aplicaciones empotradas (J2ME).

La más utilizada es sin duda la edición estándar J2SE. El código desarrollado en este proyecto corresponden a este tipo.

Para la edición J2SE existen dos kits diferentes que pueden ser descargados de la página oficial de Sun:

- El Java Development Kit (JDK) permite desarrollar y ejecutar aplicaciones Java.
- El Java Runtime Environment (JRE) permite únicamente la ejecución.

J2SE incluye bibliotecas muy extensas y completas, que permiten la implementación de casi cualquier tipo de aplicación:

- Seguridad
- EEDDs
- Componentes (JavaBeans)
- Internacionalización
- E/S
- XML
- Redes y acceso a Internet
- Programación distribuida (RMI, CORBA)
- Matemática de precisión arbitraria
- Sonido
- Interfaz de usuario (AWT, SWING)
- Gráficos 2D
- Manipulación, carga y descarga de imágenes
- Impresión
- Acceso a bases de datos (JDBC)
- Gestión de preferencias y configuraciones

En cuanto a la compilación/ejecución Java no sigue un esquema tradicional. La compilación genera un ejecutable en bytecode o código intermedio independiente. Para su ejecución se requiere un JRE específico de la plataforma. El JRE está formado por una máquina virtual java (JVM) y las librerías de clases.

La JVM interpreta el bytecode o realiza su compilación justintime para que su ejecución sea más eficiente.

Ventajas de este sistema:

- Se compila la aplicación una única vez y los ejecutables en bytecode obtenidos son válidos para cualquier plataforma. El código fuente queda a salvo.
- Es muy robusto. La máquina virtual Java es capaz de detectar y notificar gran cantidad de errores durante la ejecución de la aplicación (como accesos a elementos fuera de un vector).
- El recolector de basura no ocupa espacio en el ejecutable, ya que viene integrado en la JVM.
- Los ejecutables son pequeños porque las librerías de clases vienen proporcionadas junto a la JVM en el JRE de la plataforma concreta.

- Velocidad. Evidentemente la interpretación o incluso compilación justintime del bytecode produce aplicaciones más lentas que en el caso de la ejecución directa de un binario. El recolector de basura puede suponer una sobrecarga adicional al procesador.
- La generalidad tiene como inconveniente que no se aprovecha totalmente la potencia de la maquina y del sistema operativo. Por ejemplo, el aspecto de una aplicación Java puede resultar simple y poco atractivo en comparación con las aplicaciones nativas.

3.5.2.2 SQL

El SQL (Structured query language), lenguaje de consulta estructurado, es un lenguaje surgido de un proyecto de investigación de IBM para el acceso a bases de datos relacionales. Actualmente se ha convertido en un estándar de lenguaje de bases de datos, y la mayoría de los sistemas de bases de datos lo soportan, desde sistemas para ordenadores personales, hasta grandes ordenadores.

Como su nombre indica, el SQL nos permite realizar consultas a la base de datos. Pero el nombre se queda corto ya que SQL además realiza funciones de definición, control y gestión de la base de datos. Las sentencias SQL se clasifican según su finalidad dando origen a tres sublenguajes:

- *DDL* (Data Description Language), lenguaje de definición de datos, incluye ordenes para definir, modificar o borrar las tablas en las que se almacenan los datos y de las relaciones entre éstas.
- *DCL* (Data Control Language), lenguaje de control de datos, contiene elementos útiles para trabajar en un entorno multiusuario, en el que es importante la protección de los datos, la seguridad de las tablas y el establecimiento de restricciones en el acceso, así como elementos para coordinar la compartición de datos por parte de usuarios concurrentes, asegurando que no interfieren unos con otros.
- *DML* (Data Manipulation Language), lenguaje de manipulación de datos, nos permite recuperar los datos almacenados en la base de datos y también incluye órdenes para permitir al usuario actualizar la base de datos añadiendo nuevos datos, suprimiendo datos antiguos o modificando datos previamente almacenados.

3.5.2.3 JPA

La persistencia se define, en computación, como la capacidad de los datos de sobrevivir a la ejecución de la aplicación que los creo. La persistencia se consigue almacenando los datos en ficheros o en bases de datos.

La forma más sencilla de hacer persistir un objeto Java es a través del mecanismo de serialización o a través de ficheros de texto que almacenen el estado de un objeto para que pueda ser posteriormente recuperado. Para aplicaciones que manejen grandes cantidades de datos interrelacionados entre sí, el uso de una base de datos es fundamental.

JPA es un nuevo estándar para la persistencia de datos en Java. Está basado en POJOs (Plain Old Java Objects), lo que quiere decir que puede persistir clases Java que no tengan nada de particular (no heredan de una clase en particular, no implementan un determinado interface, etc.).

Forma parte de la especificación EJB 3.0 (Enterprise Java Beans) perteneciente a la plataforma Java EE 5.0. También puede ser usado como *framework* de persistencia para proyectos Java SE. La implementación de referencia se basa en el API TopLink de Oracle.

JPA se basa en las anotaciones introducidas en la versión 5 de Java. Las anotaciones son metadatos, es decir, información sobre los elementos del programa. Vienen a ser algo similar a las etiquetas del javadoc (@author) pero aplicadas al código, no a los comentarios.

Por defecto Java incluye tres anotaciones:

- **Override:** anota a un método e indica que sobrescribe a otro de la superclase.
- **Deprecated:** indica que un método o clase esta anticuado y no debe usarse.
- **SupressWarnings:** el compilador no avisa de warnings ocurridos sobre el elemento anotado.

Las anotaciones no son más que interfaces por lo que pueden declarar métodos o constantes. La potencia de las anotaciones viene de la posibilidad de crear anotaciones personalizadas. Pueden aplicarse a clases, métodos o atributos y se agrupan en dos categorías:

- **Lógicas:** Describen las entidades del modelo desde el punto de vista del modelo de objetos.
- **Físicas:** Describen un modelo de datos concreto en la base de datos (especificando tablas, columnas, restricciones, etc.).

Pueden evitarse las anotaciones usando un fichero de mapeo XML. Generalmente se prefieren las anotaciones por la cercanía entre la propia anotación y el código al que se refieren. Pueden incluso coexistir las anotaciones con un fichero de mapeo XML (en caso de conflicto la información de este ultimo prevalecerá).

EntityManager es una interfaz que es implementada por el proveedor de persistencia. Las operaciones de persistencia se delegan fundamentalmente en él. Para obtener instancias se utiliza la clase `EntityManagerFactory` que hace referencia a una unidad de persistencia (Persistence Unit) que contiene aspectos de configuración (librería usada, conexión JDBC,...), clases de entidades que utilizaran los `EntityManagers`.

3.5.2.4 Librería Hibernate

Hibernate es una herramienta de Mapeo objeto-relacional (ORM) para la plataforma Java (y disponible también para .Net con el nombre de NHibernate) que facilita el mapeo de atributos entre una base de datos relacional tradicional y el modelo de objetos de una aplicación, mediante archivos declarativos (XML) o anotaciones en los beans de las entidades que permiten establecer estas relaciones.

Como todas las herramientas de su tipo, Hibernate busca solucionar el problema de la diferencia entre los dos modelos de datos coexistentes en una aplicación: el usado en la memoria de la computadora (orientación a objetos) y el usado en las bases de datos (modelo relacional). Para lograr esto, permite al desarrollador detallar cómo es su modelo de datos, qué relaciones existen y qué forma tienen. Con esta información Hibernate le permite a la aplicación manipular los datos en la base de datos operando sobre objetos, con todas las características de la POO. Hibernate convertirá los datos entre los tipos utilizados por Java y los definidos por SQL. Hibernate genera las sentencias SQL y libera al desarrollador del manejo manual de los datos que resultan de la ejecución de dichas sentencias, manteniendo la portabilidad entre todos los motores de bases de datos con un ligero incremento en el tiempo de ejecución.

Hibernate está diseñado para ser flexible en cuanto al esquema de tablas utilizado, para poder adaptarse a su uso sobre una base de datos ya existente. También tiene la funcionalidad de crear la base de datos a partir de la información disponible.

3.5.3 Herramientas de desarrollo

3.5.3.1 Netbeans

El IDE NetBeans es un entorno de desarrollo integrado - una herramienta para programadores pensada para escribir, compilar, depurar y ejecutar programas. Está escrito en Java - pero puede servir para cualquier otro lenguaje de programación. Existe además un número importante de módulos para extender el IDE NetBeans. Se trata de un producto libre y

gratuito sin restricciones de uso. Sun Microsystems fundó el proyecto de código abierto NetBeans en junio de 2000 y continua siendo el patrocinador principal de los proyectos.

La plataforma NetBeans permite que las aplicaciones sean desarrolladas a partir de un conjunto de componentes de software llamados módulos. Un módulo es un archivo Java que contiene clases de java escritas para interactuar con las APIs de NetBeans y un archivo especial (manifest file) que lo identifica como módulo. Las aplicaciones construidas a partir de módulos pueden ser extendidas agregándole nuevos módulos. Debido a que los módulos pueden ser desarrollados independientemente, las aplicaciones basadas en la plataforma NetBeans pueden ser extendidas fácilmente por otros desarrolladores de software.

La plataforma ofrece servicios comunes a las aplicaciones de escritorio, permitiéndole al desarrollador enfocarse en la lógica específica de su aplicación. Entre las características de la plataforma están:

- Administración de las interfaces de usuario (en este proyecto para el desarrollo de la interfaz gráfica se ha utilizado la API Swing).
- Administración de las configuraciones del usuario.
- Administración del almacenamiento.
- Administración de ventanas.
- Framework basado en asistentes.

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 Aceptar. |
| Checkpoint 1 | El sistema debe mostrar el menú principal de dicho usuario. |

Tabla 21. Test autenticación usuario

Test 2: Autenticación de usuarios incorrecta.

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

Tabla 22. Test autenticación usuario incorrecta

Test 3: Crear un problema.

| | |
|------------------------|--|
| Requisitos testeados | RF-03 |
| Precondiciones | Un usuario Administrador identificado, se encuentra en la pantalla principal del administrador. |
| Acción Checkpoint 1 | Accede a 'Nuevo' dentro del apartado Problemas. El sistema debe mostrar un formulario para completar la información del problema. |
| Acción Checkpoint 2 | Accede a 'Nueva Alternativa' dentro del formulario de nuevo problema. El sistema debe mostrar un formulario para completar la información de una alternativa. |
| Acción Checkpoint 3 | Accede a 'Nuevo Criterio' dentro del formulario de nuevo problema. El sistema debe mostrar un formulario para completar la información de un criterio. |
| Acción Checkpoint 4 | Accede a 'Vincular Experto' dentro del formulario de nuevo problema. El sistema debe mostrar una lista de expertos que podrán ser vinculados al problema. |
| Acción Checkpoint 5 | Aceptar toda la información generada para un nuevo problema. Se almacena toda esta información en el sistema. |

Tabla 23. Test crear un problema

Test 4: Crear un problema con el nombre en blanco.

| | |
|------------------------|---|
| Requisitos testeados | RF-03 |
| Precondiciones | Un usuario Administrador identificado, se encuentra en la pantalla principal del administrador. |
| Acción Checkpoint 1 | Accede a 'Nuevo' dentro del apartado Problemas. El sistema debe mostrar un formulario para completar la información del problema. |
| Acción Checkpoint 2 | Accede a 'Nueva Alternativa' dentro del formulario de nuevo problema. El sistema debe mostrar un formulario para completar la información de una alternativa. |
| Acción Checkpoint 3 | Accede a 'Nuevo Criterio' dentro del formulario de nuevo problema. El sistema debe mostrar un formulario para completar la información de un criterio. |
| Acción Checkpoint 4 | Accede a 'Vincular Experto' dentro del formulario de nuevo problema. El sistema debe mostrar una lista de expertos que podrán ser vinculados al problema |
| Acción Checkpoint 5 | Aceptar toda la información generada para un nuevo problema, dejando en blanco el nombre del problema. El sistema muestra un mensaje de error indicando que se debe introducir un nombre para el problema. |

Tabla 24. Test crear un problema con nombre en blanco

Test 5: Crear un problema con la descripción blanco.

| | |
|------------------------|--|
| Requisitos testeados | RF-03 |
| Precondiciones | Un usuario Administrador identificado, se encuentra en la pantalla principal del administrador. |
| Acción Checkpoint 1 | Accede a 'Nuevo' dentro del apartado Problemas. El sistema debe mostrar un formulario para completar la información del problema. |
| Acción Checkpoint 2 | Accede a 'Nueva Alternativa' dentro del formulario de nuevo problema. El sistema debe mostrar un formulario para completar la información de una alternativa. |
| Acción | Accede a 'Nuevo Criterio' dentro del formulario de nuevo problema. |

| | |
|------------------------|--|
| Checkpoint 3 | El sistema debe mostrar un formulario para completar la información de un criterio. |
| Acción Checkpoint 4 | Accede a 'Vincular Experto' dentro del formulario de nuevo problema. El sistema debe mostrar una lista de expertos que podrán ser vinculados al problema |
| Acción Checkpoint 5 | Aceptar toda la información generada para un nuevo problema, dejando en blanco la descripción del problema. El sistema muestra un mensaje de error indicando que se debe introducir una descripción para el problema. |

Tabla 25. Test crear un problema con descripción en blanco

Test 6: Crear un problema con un nombre de problema que ya se encuentra en el sistema.

| | |
|------------------------|---|
| Requisitos testeados | RF-03 |
| Precondiciones | Un usuario Administrador identificado, se encuentra en la pantalla principal del administrador. |
| Acción Checkpoint 1 | Accede a 'Nuevo' dentro del apartado Problemas. El sistema debe mostrar un formulario para completar la información del problema. |
| Acción Checkpoint 2 | Accede a 'Nueva Alternativa' dentro del formulario de nuevo problema. El sistema debe mostrar un formulario para completar la información de una alternativa. |
| Acción Checkpoint 3 | Accede a 'Nuevo Criterio' dentro del formulario de nuevo problema. El sistema debe mostrar un formulario para completar la información de un criterio. |
| Acción Checkpoint 4 | Accede a 'Vincular Experto' dentro del formulario de nuevo problema. El sistema debe mostrar una lista de expertos que podrán ser vinculados al problema |
| Acción Checkpoint 5 | Aceptar toda la información generada para un nuevo problema habiendo introducido el nombre del problema ya repetido en el sistema. El sistema muestra un mensaje de error indicando que ese nombre de problema ya existe en el sistema y que debe introducirse otro. |

Tabla 26. Test crear un problema con nombre igual

Test 7: Añadir una nueva alternativa.

| | |
|------------------------|--|
| Requisitos testeados | RF-03 |
| Precondiciones | Un usuario Administrador identificado, se encuentra en la pantalla principal del administrador. |
| Acción Checkpoint 1 | Accede a 'Nuevo' dentro del apartado Problemas. El sistema debe mostrar un formulario para completar la información del problema. |
| Acción Checkpoint 2 | Accede a 'Nueva Alternativa' dentro del formulario de nuevo problema. El sistema debe mostrar un formulario para completar la información de una alternativa. |
| Acción Checkpoint 3 | Aceptar toda la información generada para una nueva alternativa. Se almacena toda la información en el sistema. |

Tabla 27. Test añadir nueva alternativa

Test 8: Añadir un nuevo criterio.

| | |
|------------------------|---|
| Requisitos testeados | RF-03 |
| Precondiciones | Un usuario Administrador identificado, se encuentra en la pantalla principal del administrador. |
| Acción Checkpoint 1 | Accede a 'Nuevo' dentro del apartado Problemas. El sistema debe mostrar un formulario para completar la información del problema. |
| Acción Checkpoint 2 | Accede a 'Nuevo Criterio' dentro del formulario de nuevo problema. El sistema debe mostrar un formulario para completar la información de un nuevo criterio. |
| Acción Checkpoint 3 | Aceptar toda la información generada para un nuevo criterio. Se almacena toda la información en el sistema. |

Tabla 28. Test añadir nuevo criterio

Test 9: Añadir un nuevo experto.

| | |
|------------------------|--|
| Requisitos testeados | RF-07 |
| Precondiciones | Un usuario Administrador identificado, se encuentra en la pantalla principal del administrador. |
| Acción Checkpoint 1 | Accede a 'Nuevo' dentro del apartado Expertos. El sistema debe mostrar un formulario para completar la información del experto. |

| | |
|--------------|---|
| Acción | Se rellena la información del experto y se acepta la información introducida. |
| Checkpoint 2 | Se almacena toda la información en el sistema. |

Tabla 29. Test añadir nuevo experto

Test 10: Modificar la información de un nuevo experto.

| | |
|----------------------|---|
| Requisitos testeados | RF-07 |
| Precondiciones | Un usuario Administrador identificado, se encuentra en la pantalla principal del administrador. |
| Acción | Accede a 'Ver/Modificar' dentro del apartado Expertos. |
| Checkpoint 1 | El sistema debe mostrar un formulario para completar o modificar la información del experto. |
| Acción | Se rellena la información del experto y se acepta la información introducida. |
| Checkpoint 2 | Se almacena toda la información en el sistema. |

Tabla 30. Test modificar información experto

Test 11: Eliminar un problema.

| | |
|----------------------|---|
| Requisitos testeados | RF-06 |
| Precondiciones | Un usuario Administrador identificado, se encuentra en la pantalla principal del administrador. |
| Acción | El usuario Administrador pulsa 'Eliminar' dentro del apartado Problemas. |
| Checkpoint 1 | El sistema debe mostrar mensaje de confirmación para borrar el problema. |
| Acción | Se acepta el borrado del problema. |
| Checkpoint 2 | El problema se elimina del sistema. |

Tabla 31. Test eliminar problema

Test 12: Resolver un problema.

| | |
|----------------------|--|
| Requisitos testeados | RF-09 |
| Precondiciones | Un usuario Administrador identificado, se encuentra en la pantalla principal del administrador. |
| Acción | El usuario Administrador pulsa 'Resolver' dentro del apartado Problemas a un problema no resuelto. |
| Checkpoint 1 | El sistema debe mostrar mensaje de confirmación para resolver el problema. |

| | |
|--------------|---|
| Acción | Se acepta la resolución del problema, y éste debe tener más del 50% de valoraciones de los expertos asignados realizadas. |
| Checkpoint 2 | El sistema muestra una pantalla con la información de la resolución, es decir, nos muestra un ranking de las mejores alternativas según las valoraciones. |

Tabla 32. Test resolver problema

Test 13: Consultar la mejor alternativa de un problema resuelto.

| | |
|----------------------|---|
| Requisitos testeados | RF-10 |
| Precondiciones | Un usuario Administrador identificado, se encuentra en la pantalla principal del administrador. |
| Acción | El usuario Administrador pulsa 'Ver/Modificar' dentro del apartado Problemas a un problema que ya está resuelto. |
| Checkpoint 1 | El sistema muestra una pantalla con la información de la resolución, es decir, nos muestra un ranking de las mejores alternativas según las valoraciones. |

Tabla 33. Test consultar mejor alternativa problema

Test 14: Valorar las opciones (alternativas y criterios) de un problema.

| | |
|----------------------|--|
| Requisitos testeados | RF-11 |
| Precondiciones | Un usuario Experto identificado, se encuentra en la pantalla principal del administrador. |
| Acción | El usuario Experto pulsa 'Valorar' dentro del listado de Problemas. |
| Checkpoint 1 | El sistema debe mostrar en la parte derecha de la pantalla las diferentes opciones para un problema concreto. |
| Acción | Se pulsa la pestaña de 'Valoración' dentro de las opciones para un problema seleccionado. |
| Checkpoint 2 | El sistema muestra una pantalla con una matriz de alternativas y criterios para poder introducir las valoraciones numéricas mayores que 0. |
| Acción | Se valoran las diferentes casillas de la matriz de valoración de un experto y se pulsa aceptar. |
| Checkpoint 3 | El sistema muestra un mensaje de éxito confirmando que se ha realizado la valoración correctamente. |

Tabla 34. Test valorar problema

Test 15: Consultar la mejor alternativa de un problema, resuelto por el método TOPSIS para un experto.

| | |
|------------------------|---|
| Requisitos testeados | RF-13 |
| Precondiciones | Un usuario Experto identificado, se encuentra en la pantalla principal del administrador. |
| Acción Checkpoint 1 | El usuario Experto pulsa 'Ver' dentro del listado de Problemas. El sistema debe mostrar en la parte derecha de la pantalla las diferentes opciones para un problema concreto. |
| Acción Checkpoint 2 | Se pulsa la pestaña de 'Resolución' dentro de las opciones para un problema seleccionado. El sistema muestra una pantalla con las valoraciones medias según el método TOPSIS y un ranking de las mejores alternativas. |

Tabla 35. Test consulta mejor alternativa experto

3.6.1.2 Resultados obtenidos

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

| TEST | Resultado |
|--------------|------------------|
| Test 1 | |
| Checkpoint 1 | Ok |
| Test 2 | |
| Checkpoint 1 | Ok |
| Test 3 | |
| Checkpoint 1 | Ok |
| Checkpoint 2 | Ok |
| Checkpoint 3 | Ok |
| Checkpoint 4 | Ok |
| Checkpoint 5 | Ok |
| Test 4 | |
| Checkpoint 1 | Ok |
| Checkpoint 2 | Ok |
| Checkpoint 3 | Ok |
| Checkpoint 4 | Ok |
| Checkpoint 5 | Ok |
| Test 5 | |
| Checkpoint 1 | Ok |
| Checkpoint 2 | Ok |

| | |
|--------------|----|
| Checkpoint 3 | Ok |
| Checkpoint 4 | Ok |
| Checkpoint 5 | Ok |
| Test 6 | |
| Checkpoint 1 | Ok |
| Checkpoint 2 | Ok |
| Checkpoint 3 | Ok |
| Checkpoint 4 | Ok |
| Checkpoint 5 | Ok |
| Test 7 | |
| Checkpoint 1 | Ok |
| Checkpoint 2 | Ok |
| Checkpoint 3 | Ok |
| Test 8 | |
| Checkpoint 1 | Ok |
| Checkpoint 2 | Ok |
| Checkpoint 3 | Ok |
| Test 9 | |
| Checkpoint 1 | Ok |
| Checkpoint 2 | Ok |
| Test 10 | |
| Checkpoint 1 | Ok |
| Checkpoint 2 | Ok |
| Test 11 | |
| Checkpoint 1 | Ok |
| Checkpoint 2 | Ok |
| Test 12 | |
| Checkpoint 1 | Ok |
| Checkpoint 2 | Ok |
| Test 13 | |
| Checkpoint 1 | Ok |
| Test 14 | |
| Checkpoint 1 | Ok |
| Checkpoint 2 | Ok |
| Checkpoint 3 | Ok |
| Test 15 | |
| Checkpoint 1 | Ok |
| Checkpoint 2 | Ok |

Tabla 36. Resultados obtenidos test realizados

Capítulo 4

Conclusiones

4.1 Conclusiones finales

El objetivo de este proyecto fin de carrera es realizar una aplicación de escritorio capaz de automatizar el proceso de Toma de Decisión mediante el método TOPSIS para la resolución de problemas, es decir, que sea capaz de realizar cálculos sobre unos valores introducidos por unos expertos, dentro de un dominio numérico, y generar un ranking entre las alternativas para ayudar a la toma de decisión.

Hay que destacar, la importancia que tiene implantar un sistema como éste en la gestión de empresas, pues permite que los expertos puedan dar su valoración desde su experiencia y conocimiento, y obtener un ranking con las mejores alternativas para la toma de decisión. Estas opiniones serán expresadas en el dominio de expresión numérico para mostrar el ranking.

Además, el sistema se apoya en una base de datos que permitirá a los usuarios administradores y expertos poder consultar en cualquier momento las valoraciones realizadas pertenecientes a los diferentes casos o problemas.

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.

4.2 Trabajos futuros

Dado que nos encontramos en un tiempo en que cada vez llevamos a cabo mas tareas a través de Internet, una mejora de este proyecto es, desarrollar una aplicación web a la que se pudiese acceder desde diversos dispositivos móviles, desarrollando para ello una interfaz flexible siguiendo las últimas tendencias en *responsive design*, utilizando para ello estándares web como HTML5 y CSS3.

Otra mejora posible es extender el campo de la toma de decisión con información heterogénea, pudiendo realizar valoraciones con dominios diferentes para los expertos. También se pueden elegir diferentes estructuras de representación de preferencias.

BIBLIOGRAFÍA

- [1] Jim Arlow, Ila Neustadt. UML 2 and the unified process. Ed. Anaya Multimedia, 84-415-2033-X, 2006.
- [2] Alessio Ishizaka, Philippe Nemery. Multi-criteria Decision Analysis. Methods and Software. John Wiley & Sons, Ltd, 978-1-119-97407-9, 2013.
- [3] Diego Edher Maurtua Ollaguez. Criterios de Selección de Personal mediante el uso del proceso de análisis jerárquico. Aplicación en la selección de personal para la Empresa Exotic Foods S.A.C. E.A.P. de Investigación Operativa, Universidad Nacional Mayor de San Marcos, 2006.
- [4] Alberto C y Carignano C. Apoyo Cuantitativo a las Decisiones. Asociación Cooperadora de la FCE de la UNC, 2007.
- [5] Zeleny, M. Compromise programming, in Multicriteria decision making. J.-L. Cochrane and M. Zeleny, Eds. Columbia: University of South Carolina, 1973.
- [6] Zeleny, M. Multiple criteria decision making. McGrawHill, 1982.
- [7] Hwang C. y Yoon K. Multiple Attribute Decision Making an Introduction. Sage. USA, 1995.
- [8] R. Anthony. Planning and Control Systems: A Framework for Analysis. Harvard Business School Division or Research Press, 1965.
- [9] K. J. Arrow. Social Choice and Individual Values. Yale University Press, 1963.
- [10] 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.
- [11] T. Calvo, R. Mesiar, and RR. Yager. Quantitative weights and aggregation. IEEE. Transactions on Fuzzy Systems, 12(1):62-69, 2004.
- [12] V. Cutello and J. Montero. Hierarchies of aggregation operators. International Journal of Intelligent Systems, 9(11):1025-1045, 1994.
- [13] D. Dubois and H. Prade. A review of fuzzy set aggregation connectives. Information Science, 36:85-121, 1985.
- [14] D. Dubois and H. Prade. Weighted minimum and maximum operations in fuzzy set theory. Information Science, 39:205-210, 1986.
- [15] R.R. Yanger. An approach to ordinal decision making. International Journal of Approximate Reasoning, 12:237-261, 1995.
- [16] R.R. Yanger. Fusion of ordinal information using weighted median aggregation. International Journal of Approximate Reasoning, 18:32-35, 1998.

- [17] 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.
- [18] R. Duncan and H. Raiffa. *Games and Decision. Introduction and Critical Survey*. Dover Publications, 1985.
- [19] M. Roubens. Fuzzy sets and decision analysis. *Fuzzy Sets and Systems*, 90:199-206, 1997.
- [20] S. Barba-Romero and J. Pomerol, *Multicriterion Decision in Management: Principles and Practice* Kluwer Academic Publishers, 2000.
- [21] J. Von Neumann and O. Morgenstern, *Theory of games and economic behaviour*. Princeton: Princeton University Press, 1944.
- [22] T. Saaty, *The analytic hierarchy process* McGraw-Hill, New York, 1980.
- [23] Edwards W. y Barron F. H. "SMARTS and SMARTER: Improved simple methods for multiattribute utility measurements", *Organizational Behavior and Human Decision Process*, 60, 1994
- [24] Lootsma F and Schuijt, "The Multiplicative AHP, SMART and ELECTRE in a common context," *Journal of Multicriteria Decision Analysis*, vol. 6, pp. 185-196, 1997.
- [25] Roy B, "Decision-Aid and Decision Making," *European Journal of Operational Research*, vol. 45, pp. 324-331, 1991.
- [26] B. Roy and D. Vanderpooten, "An overview on "The European school of MCDA: Emergence, basic features and current works"," *European Journal of Operational Research*, vol. 99, no. 1, pp. 26-27, May1997.
- [27] J. P. Vinke and J. P. Brans, "A preference ranking organization method (the PROMETHEE method for multiple criteria decision-making)," *Management science*, vol. 31, no. 6, pp. 647-656, 1985.
- [28] J. P. Brans and B. Mareschal, "The PROMETHE methods for MCDM; The PROMCALC, GAIA and BANKADVISER software. *Readings in MCDA*,". Bana e Costa, Ed. Berlin: Springer-Verlang, 1990.
- [29] C. Bana, C. A. Costa, and Vansnick J.C., "The MACBETH approach: Basic ideas, software and an application," in *Advances in Decision Analysis*. N. Meskens and M. Roubens, Eds. Dordrecht: Kluwer Academic Publishers, 1999, pp. 131-157.
- [30] O. Larichev and H. Moshkovich, *ZAPROS: A method and system for ordering multiattribute alternatives on the basis of a decision-maker's preferences*. Moscow: Preprint of All Union Institute for Systems Studies, 1991.

- [31] O. Larichev, H. Moshkovich, A. Mechitov, and D. Olson, "Experiments comparing qualitative approaches to rank ordering of multiattribute alternatives," *Journal of Multicriteria Decision Analysis*, vol. 2, pp. 5-26, 1993.
- [32] C. L. Hwang and K. Yoon, *Multiple Attribute Decision Methods and Applications*, Springer, Berlin Heidelberg ed 1981.
- [33] B. V. Dasarathy, "Smart: Similarity measured anchored ranking technique for the analysis of multidimensional data," *IEEE Transactions on Systems, Man and Cybernetics*, vol. SMC-6, no. 10, pp. 708-711, 1976.
- [34] L. Dombi. *Fuzzy Logic and Soft Computing*, chapter A General Framework for the Utility-Based and Outranking Methods, pages 202-208. World Scientific, 1995.
- [35] R. D. Luce and P. Suppers. *Handbook of Mathematical Psychology*, chapter Preferences, Utility and Subject Probability, pages 249-410. Wiley, 1965.
- [36] Trygve Reenskaug. *THING-MODEL-VIEW-EDITOR* an Example from planning system. 12 May 1979.

ANEXO I. Manual de Instalación del Software

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 CD-ROM que acompaña a esta memoria. Sitúese en el dispositivo de CD-ROM D:\TomaDecisionTOPSIS y compruebe que se encuentran los siguientes archivos:

- wampserver2.2e-php5.4.3-httpd2.2.22-mysql5.5.24-32b.exe
- BBDD
 - bbdd_topsis.sql
- TDM-TOPSIS
 - lib
 - TOPSIS.jar
- Java
 - jxpiinstall.exe

Paso 1: Instalación de la Máquina Virtual Java

Para comprobar la versión del entorno de ejecución instalado en el equipo ejecutar el siguiente test :

<https://www.java.com/es/download/installed.jsp?detect=jre&try=1>

Si no dispone de Java o la versión es anterior a la versión 8 Update 25, deberá instalar el archivo jxpiinstall.exe. Para ello vaya al CD-ROM, ejecute dicho archivo dentro de la carpeta Java y siga los sencillos pasos de instalación.

Paso 2: Instalar Wamp (MySQL y PHP)

WAMP es el acrónimo usado para describir un sistema de infraestructura de internet que usa las siguientes herramientas:

- Windows, como sistema operativo
- Apache, como servidor web
- MySQL, como gestor de bases de datos
- PHP (generalmente), Perl, como lenguajes de programación.

El uso de WAMP permite servir páginas html a internet, además de poder gestionar datos en ellas. Al mismo tiempo, WAMP proporciona lenguajes de programación para desarrollar aplicaciones web.

WampServer es un entorno de desarrollo web para Windows con el que se puede crear aplicaciones web con Apache, PHP y bases de datos MySQL database. También incluye PHPMyAdmin y SQLiteManager para manejar tus bases de datos de forma rápida.

Vamos a ver la instalación de Wamp:

- Ejecutaremos el archivo wampserver2.2e-php5.4.3-httpd2.2.22-mysql5.5.24-32b.exe que encontraremos en el CD-ROM.
- Iremos siguiendo la instalación y pulsaremos "Next" para seguir hacia delante:

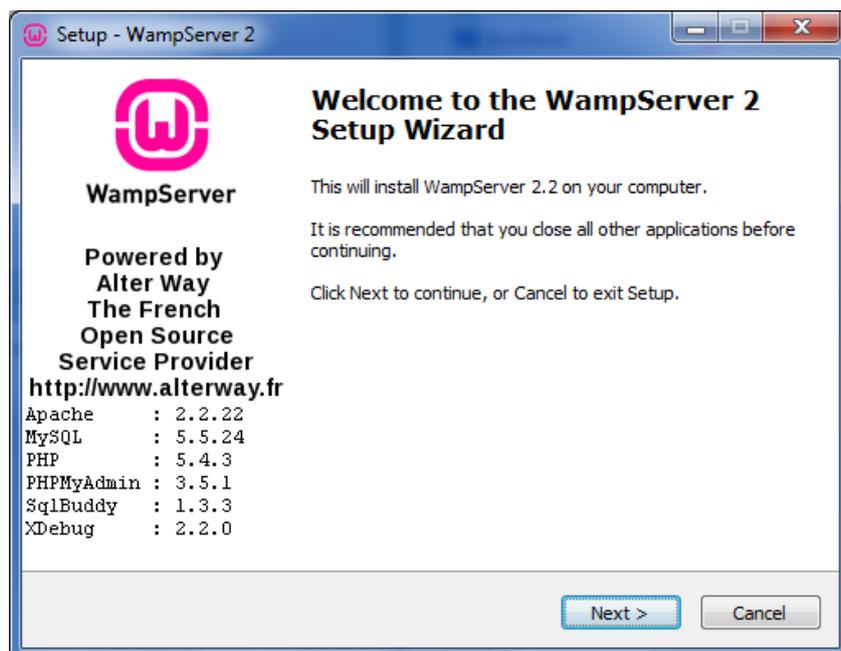


Figura I.1. Instalación WampServer

Aceptamos los términos generales y la licencia, como se ve en la Figura I.2.

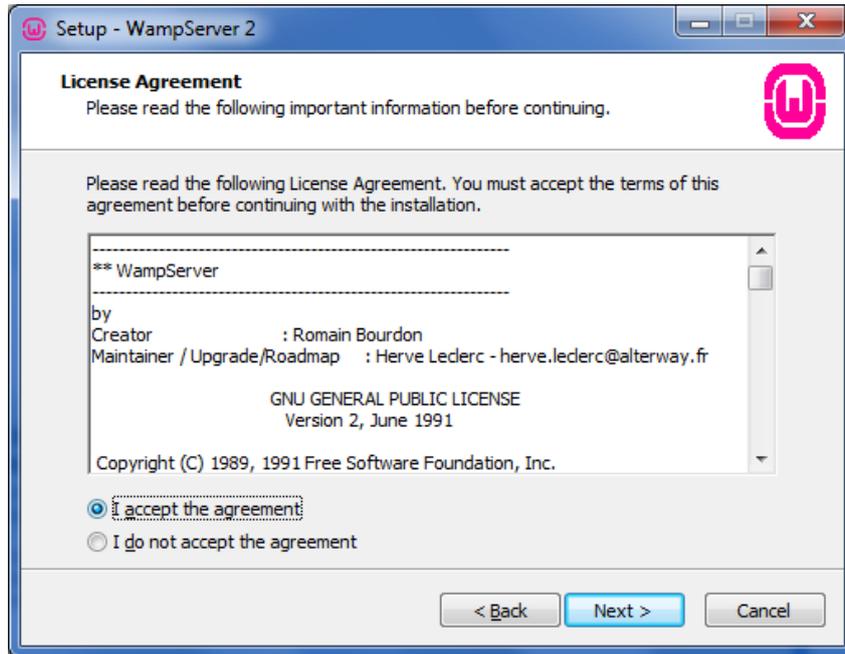


Figura I.2. Instalación WampServer

Elegimos una ruta para instalar el software.

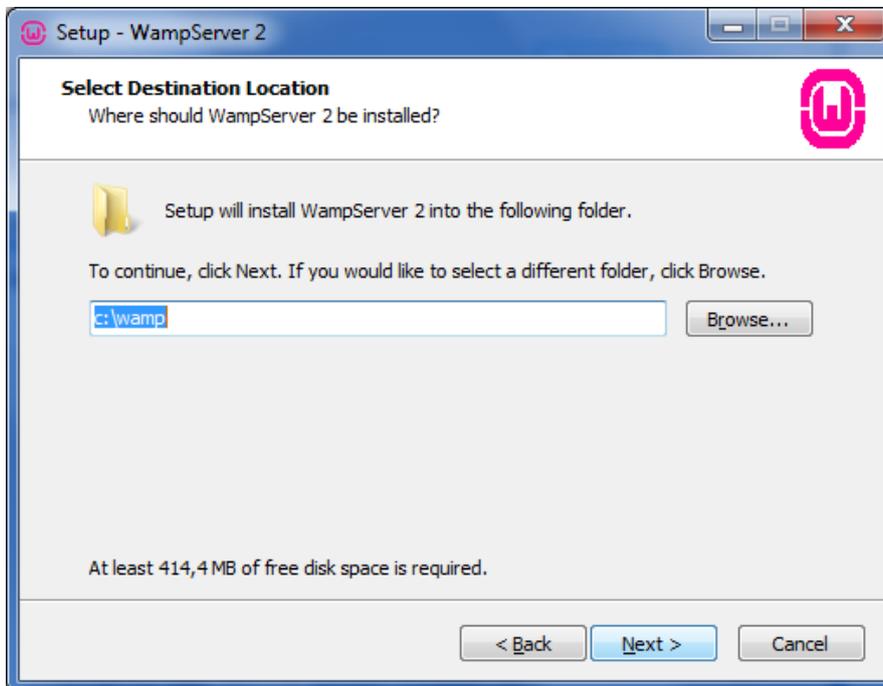


Figura I.3. Instalación WampServer

Tenemos la posibilidad de crear un icono en el escritorio y un icono de inicio rápido en la barra de herramientas. Ver Figura I.4.

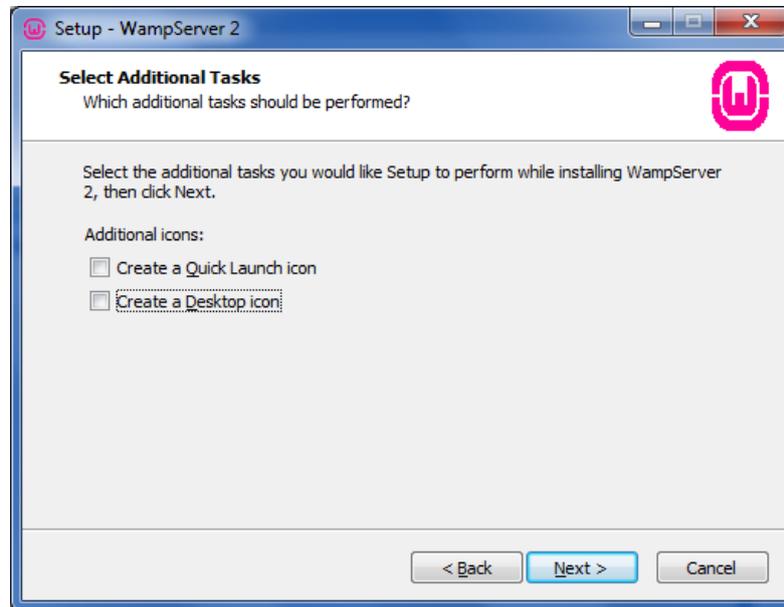


Figura I.4. Instalación WampServer

Pulsamos "Install" para ejecutar la instalación, mostrándonos el software un resumen de la misma antes de realizarse.

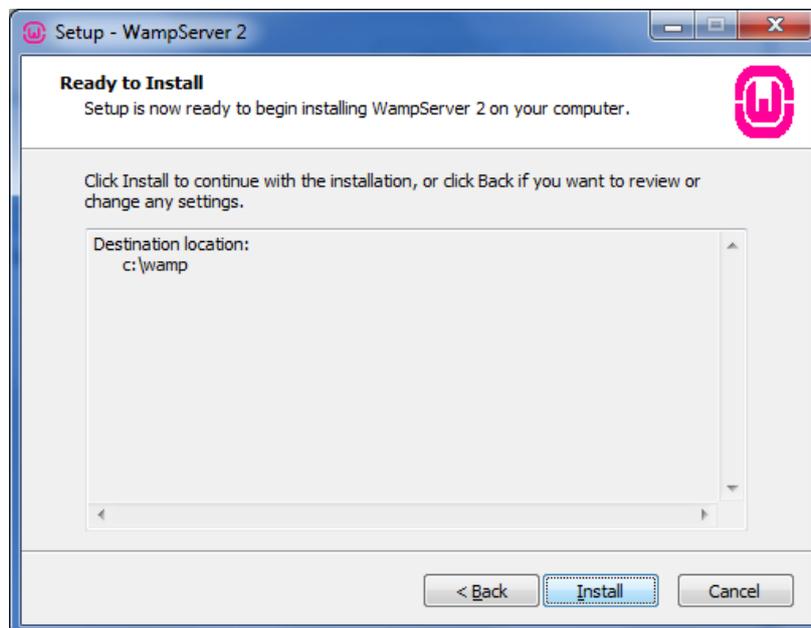


Figura I.5. Instalación WampServer

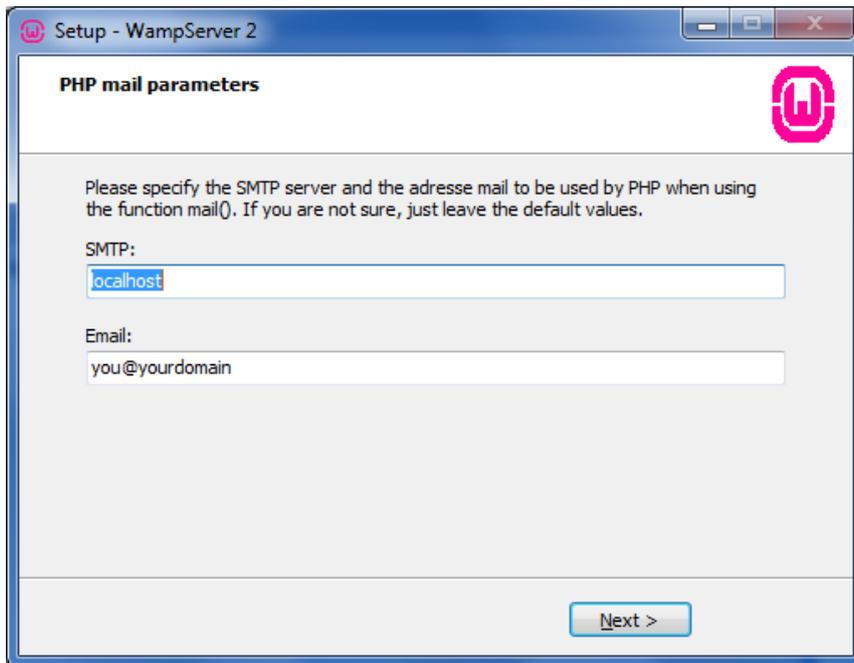


Figura I.6. Instalación WampServer

Concluimos la instalación y podemos ejecutar la aplicación de WampServer.

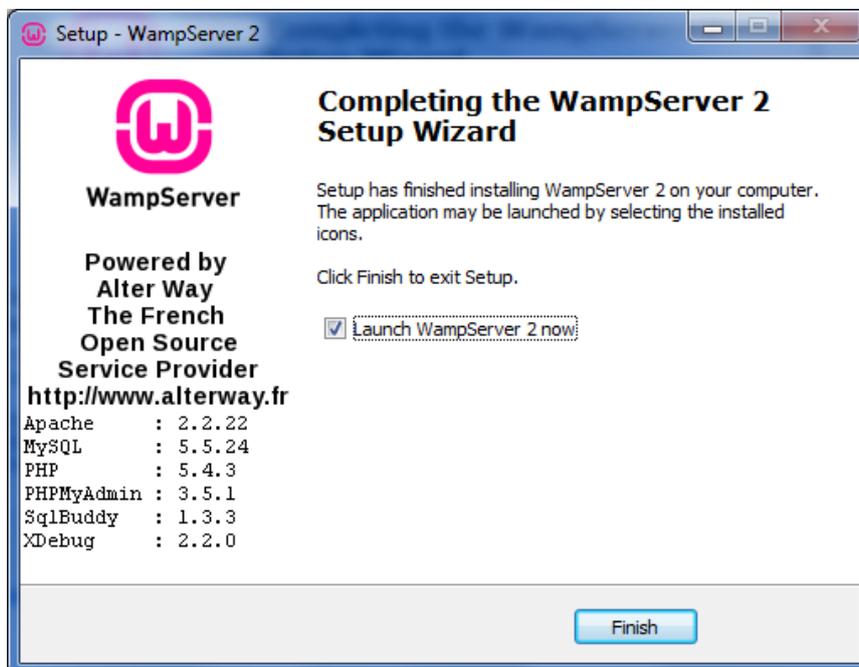


Figura I.7. Instalación WampServer

Paso 3: Instalar la Base de Datos de nuestra aplicación

Una vez instalada la aplicación de WampServer, la ejecutaremos y accederemos a su phpMyAdmin para cargar la BBDD establecida para nuestra aplicación.

Para acceder a phpMyAdmin, accedo desde el icono de WampServer y desde el menú que se despliega pulso "phpMyAdmin".

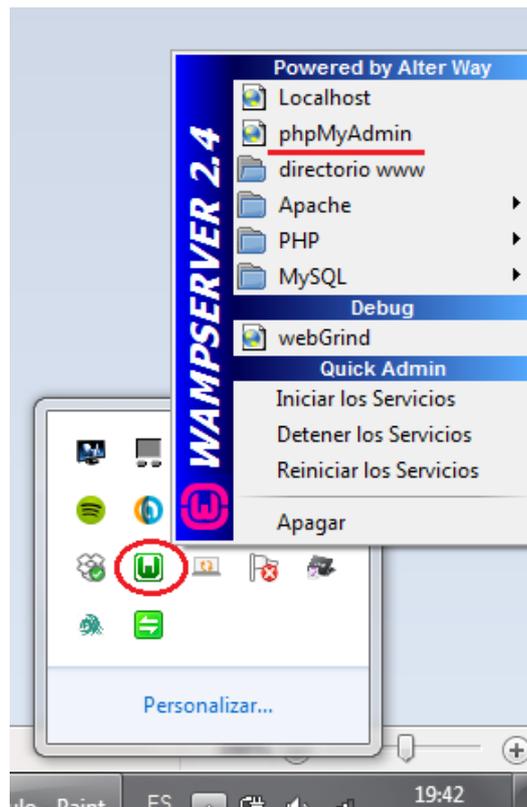


Figura I.8. Acceso phpMyAdmin

De esta forma accederemos al phpMyAdmin de WampServer. Pulsaremos la pestaña superior de "Bases de datos" donde crearemos una base de datos con el nombre "bbdd_topsis" y con el "Cotejamiento" de "utf8_general_ci". Pulsaremos crear y ya tendremos la base de datos creada, como muestra la Figura I.9.

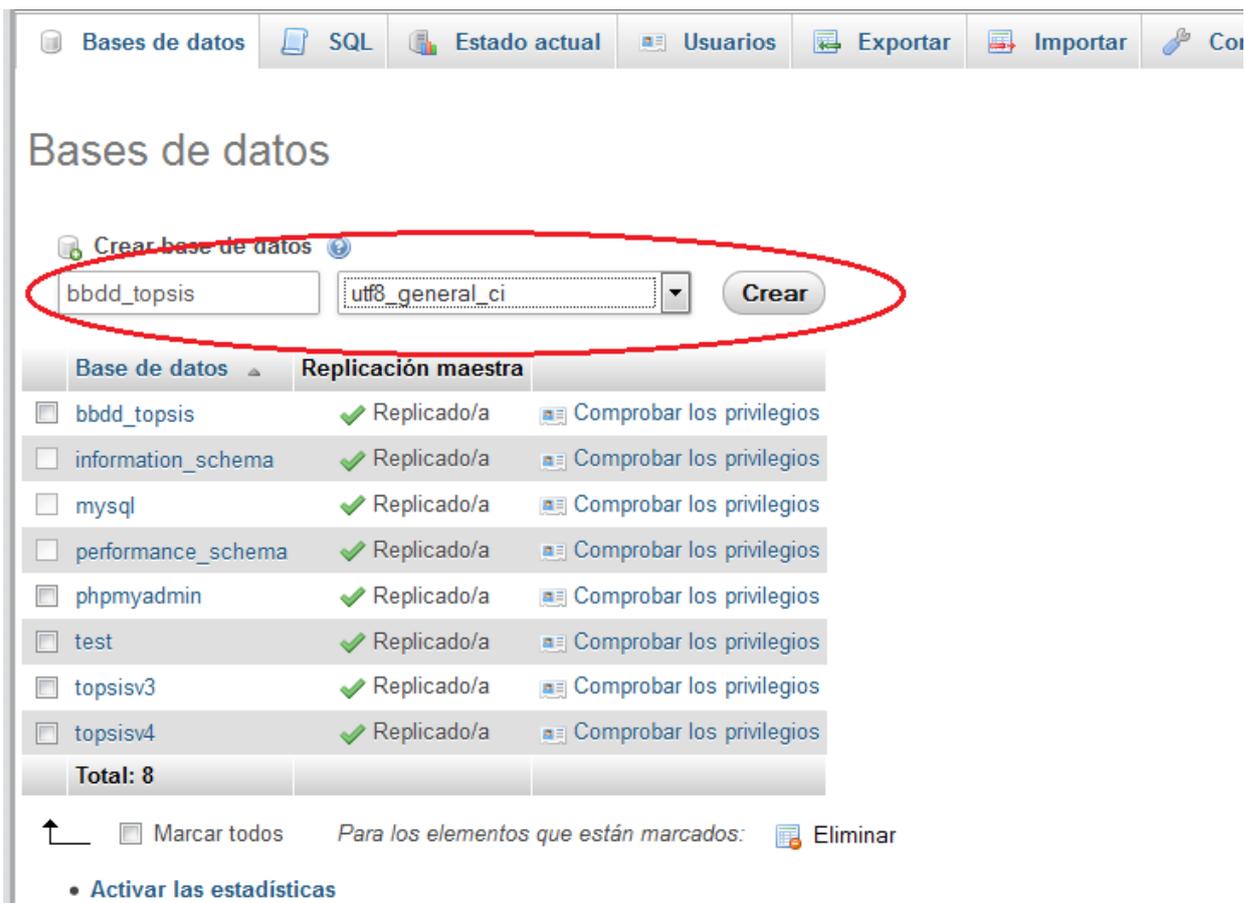


Figura I.9. Crear BBDD

Ahora importaremos el archivo "bbdd_topsis.sql" de base de datos que está en nuestro CD-ROM, localizado en la carpeta "BBDD" como se muestra en la Figura I.10, y daremos a "Continuar". Si todo ha ido bien, nos saldrá un mensaje de éxito. De esta forma, ya tenemos lista nuestra base de datos.

localhost

Bases de datos SQL Estado actual Usuarios Exportar **Importar** Configuración Reg

Importando al servidor actual

Archivo a importar:

El archivo puede ser comprimido (gzip, zip) o descomprimido.
Un archivo comprimido tiene que terminar en **[formato].[compresión]**. Por ejemplo: **.sql.zip**

Buscar en su ordenador: **Examinar...** bbdd_topsis.sql (Máximo: 2,048KB)

Conjunto de caracteres del archivo: utf-8

Importación parcial:

Permitir la interrupción de una importación en caso que el script detecte que se ha acercado al límite de tiempo PHP. (Esto podría ser un bus transacciones.)

Número de filas a omitir, iniciando de la primer fila: 0

Formato:

SQL

Opciones específicas al formato:

Modalidad SQL compatible: NONE

No utilizar AUTO_INCREMENT con el valor 0

Continuar

Figura I.10. Importar BBDD

Paso 4: Puesta en marcha de la aplicación

Ya solo nos queda copiar el directorio TMD-TOPSIS en el equipo y hacer doble clic sobre el fichero .jar que hay dentro para ejecutar la aplicación. Se adjunta también un directorio llamado "lib" donde se encuentran las librerías para la ejecución del programa.

NOTA: Por defecto los credenciales del administrador para la aplicación son

- Usuario: admin
- Password: 1234

Anexo II. Manual de usuario (Administrador)

En este Anexo II vamos a ver el modo de utilización de la aplicación de escritorio de Toma de Decisión Multicriterio mediante el método TOPSIS para el usuario Administrador. Éste se encargará de gestionar la información y datos de la aplicación.

Pasemos a ver las diferentes acciones posibles del usuario administrador:

Acceso

El usuario administrador deberá introducir su login de Usuario y su Contraseña para acceder al sistema. Posteriormente pulsar "Aceptar" para entrar, o "Salir" para salir de la aplicación.



Figura II.1. Acceso usuario

Crear Problema

El usuario administrador deberá pulsar el botón de "Nuevo" en el apartado gestión de problemas de la pantalla principal. Ver Figura II.2.

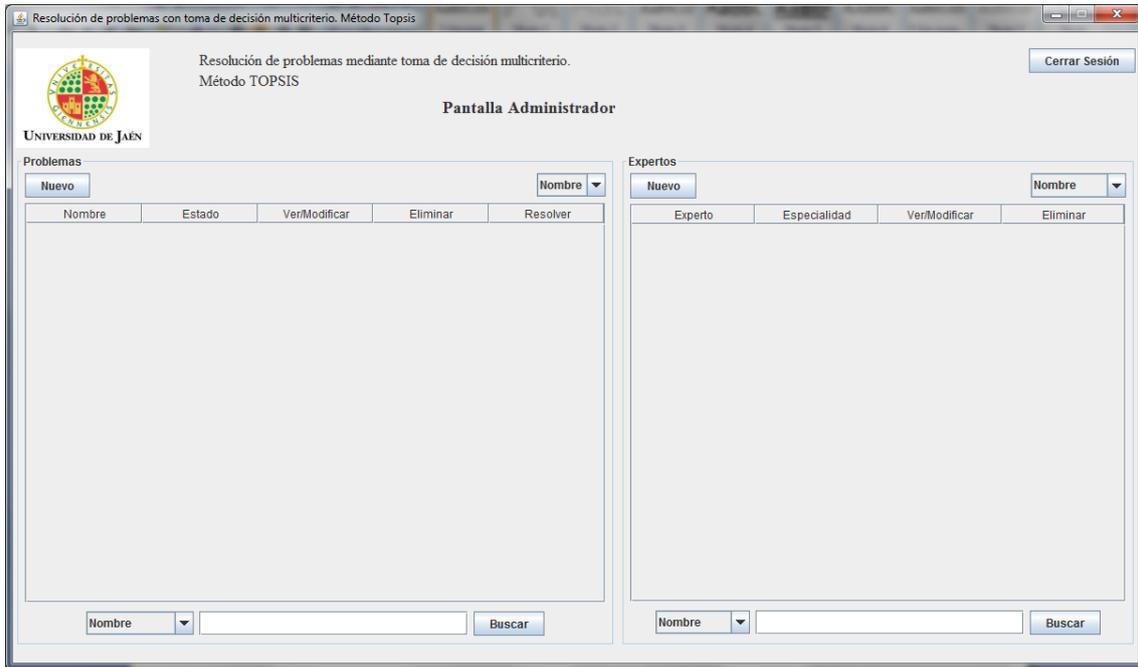


Figura II.2. Pantalla principal

El usuario administrador deberá rellenar la información en "Nombre del problema" y "Descripción del problema" (Ver Figura II.3). No se debe dejar vacios dichos campos.

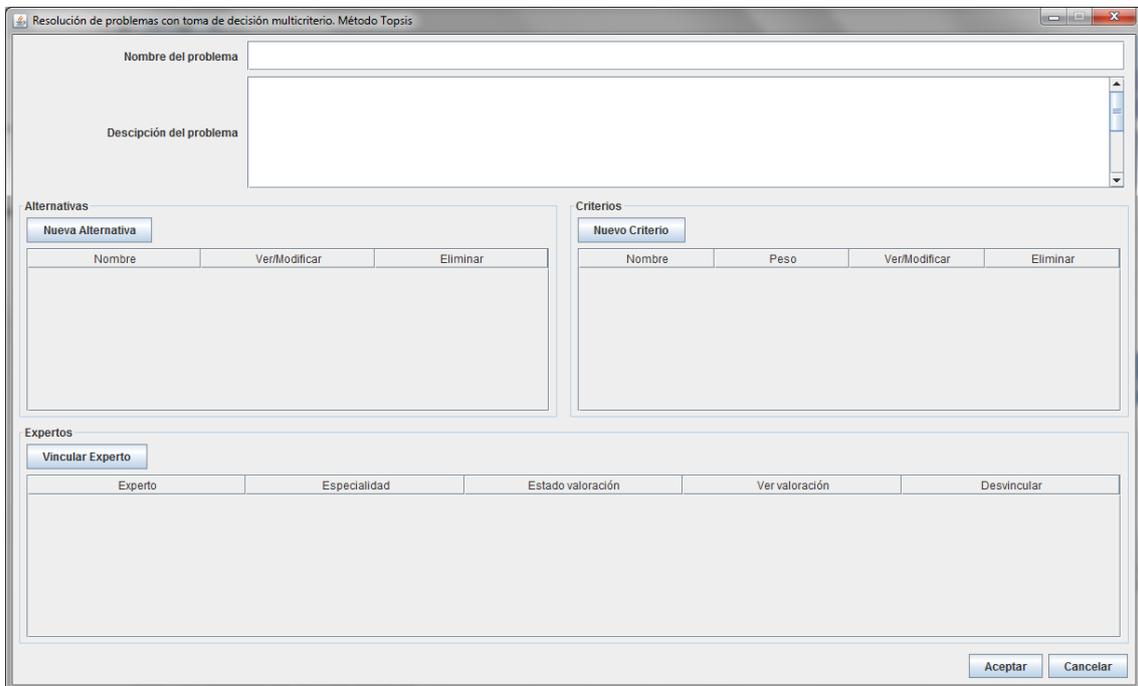


Figura II.3. Crear Problema

El usuario administrador pulsará "Nueva Alternativa", si desea añadir una alternativa nueva. Para incluir una nueva alternativa al problema, el usuario rellenará la información de la alternativa; "Nombre" y "Descripción". Pulsar aceptar para almacenar la información introducida. No se debe dejar vacíos dichos campos.

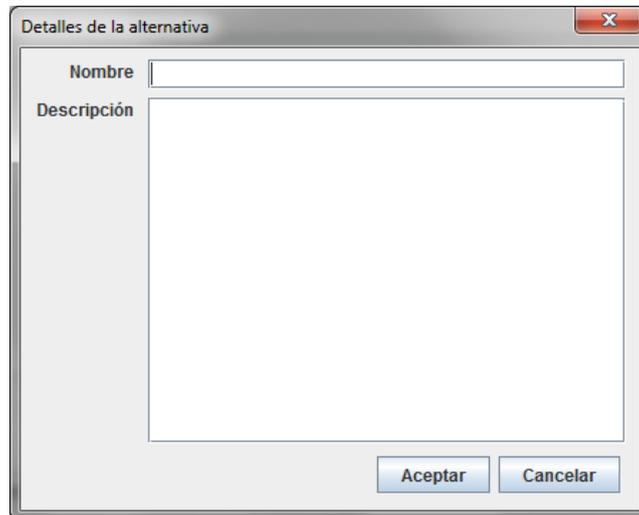


Figura II.4. Nueva alternativa

El usuario administrador pulsará "Nuevo Criterio", si desea añadir un criterio nuevo. Para incluir un nuevo criterio al problema, el usuario rellenará la información del criterio; "Nombre", "Peso" y "Descripción". Pulsar aceptar para almacenar la información introducida. No se debe dejar vacíos dichos campos.

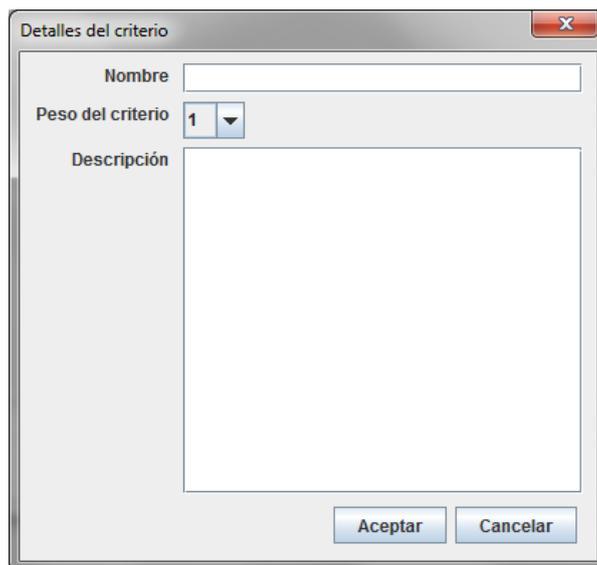


Figura II.5. Nuevo criterio

El usuario administrador vinculará expertos al problema para que puedan valorar las diferentes alternativas y criterios posteriormente. El usuario pulsará "Vincular Experto". El usuario administrador vinculará a los expertos que desee de la lista que aparece pulsando el botón "Vincular". Para salir de la pantalla de vinculación, el usuario administrador pulsará "Cerrar" (Ver Figura II.6).

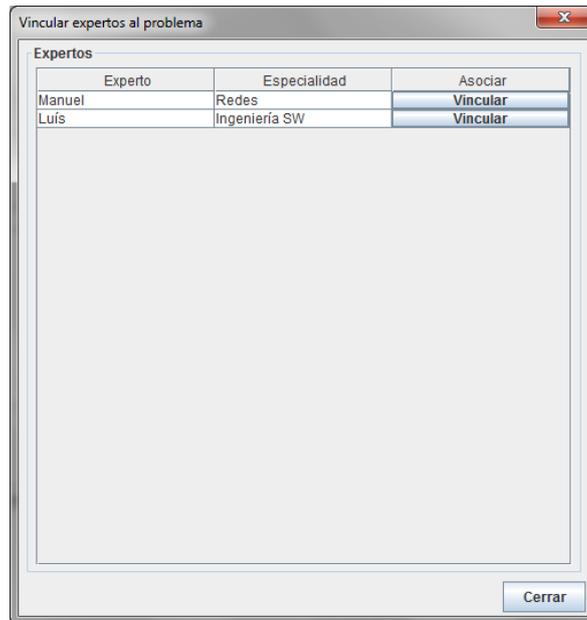


Figura II.6. Vincular expertos

Una vez que el usuario administrador ha completado toda la información del nuevo problema, pulsará el botón "Aceptar" para almacenar la información en el sistema.

Crear Experto

El usuario administrador deberá pulsar el botón de "Nuevo" en el apartado gestión de expertos de la pantalla principal. Se puede observar en la Figura II.2. Pantalla principal.

El usuario completará la información de un nuevo experto. Se escribirá el "Login", "Nombre", "Contraseña" y "Especialidad". El usuario pulsará el botón "Aceptar" para almacenar la información en el sistema. No se debe dejar vacíos dichos campos. (Ver Figura II.7. Crear experto)

Detalles del experto

Login

Nombre

Contraseña

Especialidad

Aceptar Cancelar

Figura II.7. Crear experto

Ver/Modificar un Problema

El usuario deberá pulsar el botón de "Ver/Modificar" del problema seleccionado en el listado de problemas que aparece en el apartado gestión de problemas de la pantalla principal.

Resolución de problemas con toma de decisión multicriterio. Método Topsis

Resolución de problemas mediante toma de decisión multicriterio.
Método TOPSIS

Pantalla Administrador

UNIVERSIDAD DE JAÉN

| Problemas | | | | |
|----------------------|-----------|---------------|----------|----------|
| Nombre | Estado | Ver/Modificar | Eliminar | Resolver |
| Problema palas pá... | Pendiente | Ver/Modificar | Eliminar | Resolver |

| Expertos | | | |
|----------|---------------|---------------|----------|
| Experto | Especialidad | Ver/Modificar | Eliminar |
| Luis | Ingeniería SW | Ver/Modificar | Eliminar |
| Manuel | Redes | Ver/Modificar | Eliminar |

Figura II.8. Pantalla principal 2

El usuario administrador podrá ver la información de un problema. El usuario administrador tiene la posibilidad de modificar "Nombre del problema" y "Descripción del problema". No se debe dejar vacíos dichos campos (Ver Figura II.9)

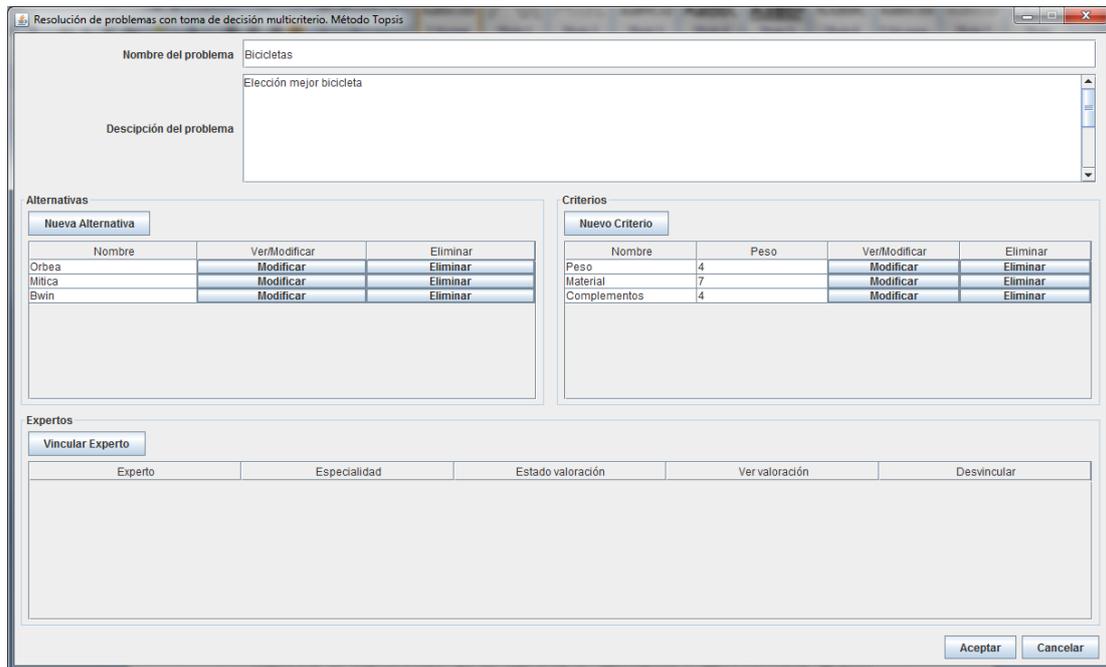


Figura II.9. Ver/Modificar problema

El usuario administrador pulsará "Nueva Alternativa", si desea añadir una alternativa nueva, "Modificar" para cambiar la información de la alternativa o "Eliminar" para quitarla del sistema y, por lo tanto, del problema.

Para incluir una nueva alternativa al problema, el usuario administrador rellenará la información de la alternativa; "Nombre" y "Descripción". Pulsar aceptar para almacenar la información introducida. No se debe dejar vacíos dichos campos.

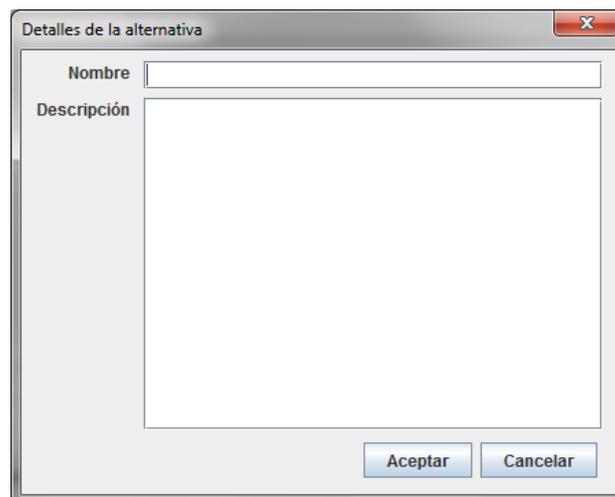


Figura II.10. Ver/Modificar alternativa

El usuario administrador pulsará "Nuevo Criterio", si desea añadir un criterio nuevo, "Modificar" para cambiar la información del criterio o "Eliminar" para quitarlo del sistema y, por lo tanto, del problema.

Para incluir un nuevo criterio al problema, el usuario administrador rellenará la información del criterio; "Nombre", "Peso" y "Descripción". Pulsar aceptar para almacenar la información introducida. No se debe dejar vacíos dichos campos.

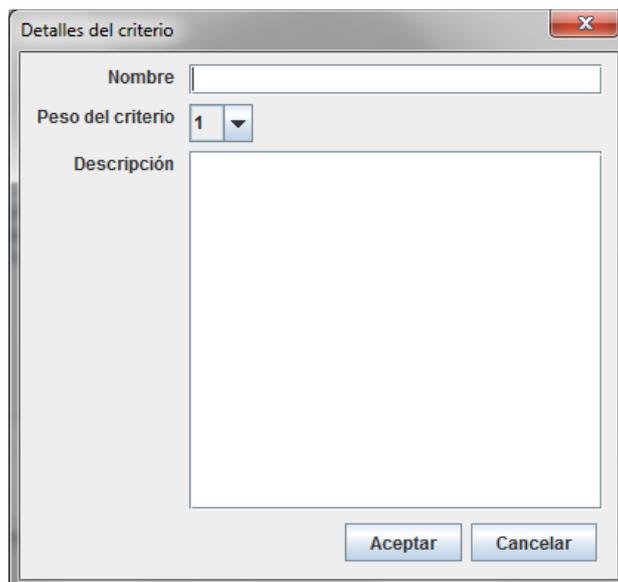
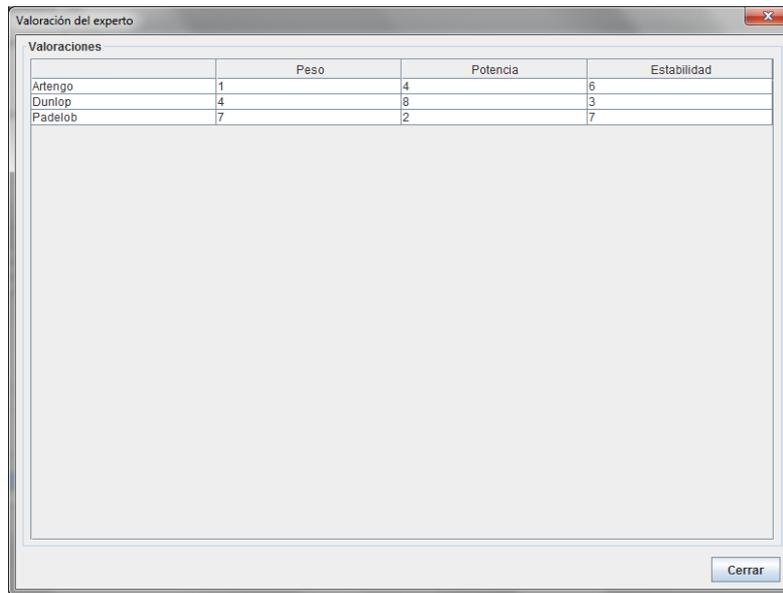
The image shows a software dialog box titled "Detalles del criterio". It has a standard window frame with a close button (X) in the top right corner. Inside the dialog, there are three main input areas: 1. "Nombre": A single-line text input field. 2. "Peso del criterio": A dropdown menu currently displaying the value "1". 3. "Descripción": A large, empty text area for entering details. At the bottom of the dialog, there are two buttons: "Aceptar" (Accept) and "Cancelar" (Cancel).

Figura II.11. Ver/Modificar criterio

El usuario administrador podrá vincular expertos al problema para que puedan valorar las diferentes opciones posteriormente. El usuario administrador pulsará "Vincular Experto". El usuario administrador vinculará a los expertos que aparecen en la lista de expertos del sistema mostrado pulsando el botón "Vincular". Para salir de la pantalla de vinculación, el usuario administrador pulsará "Cerrar". Ver Figura II.6. Vincular expertos.

El usuario administrador podrá ver la valoración de un experto vinculado pulsando el botón "Ver valoración" dentro de la zona de lista de expertos vinculados al problema. para salir de la pantalla, pulsar "Cerrar".



Valoración del experto

| Valoraciones | Peso | Potencia | Estabilidad |
|--------------|------|----------|-------------|
| Artengo | 1 | 4 | 6 |
| Dunlop | 4 | 8 | 3 |
| Padelob | 7 | 2 | 7 |

Cerrar

Figura II.12. Valoración experto

El usuario administrador podrá desvincular a un experto de un problema pulsando el botón "Desvincular" dentro de la zona de lista de expertos vinculados al problema.

Una vez que el usuario administrador ha completado, revisado o modificado toda la información del problema, pulsará el botón "Aceptar" para almacenar la información en el sistema.

Eliminar un Problema

El usuario administrador deberá pulsar el botón de "Eliminar" del problema seleccionado en el listado de problemas que aparece en el apartado gestión de problemas de la pantalla principal (Ver Figura II.8. Pantalla principal 2). El sistema pedirá confirmación de la eliminación del problema al usuario administrador.

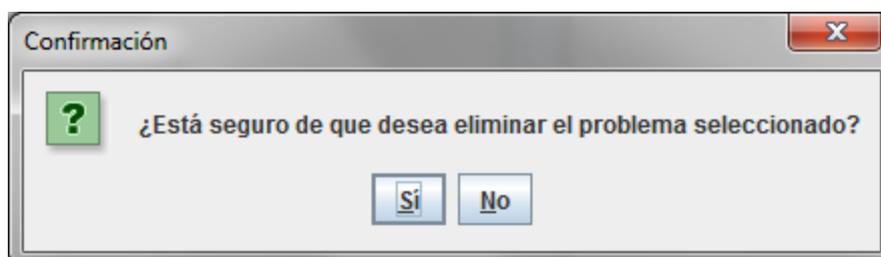


Figura II.13. Mensaje eliminar problema

Resolver un Problema con el Método TOPSIS

El usuario administrador deberá pulsar el botón de "Resolver" del problema seleccionado en el listado de problemas que aparece en el apartado gestión de problemas de la pantalla principal (Ver Figura II.8. Pantalla principal 2). El sistema pedirá confirmación para la resolución del problema con el Método TOPSIS al usuario administrador.

El usuario administrador pulsará "Si" para resolver el problema y el sistema mostrará la pantalla con toda la información de la solución. En la pantalla el usuario administrador podrá conocer un ranking de las primeras tres mejores alternativas. Ver Figura II.14.

NOTA: Para poder resolver un problema, éste debe tener valorado más del 50% de las valoraciones totales.

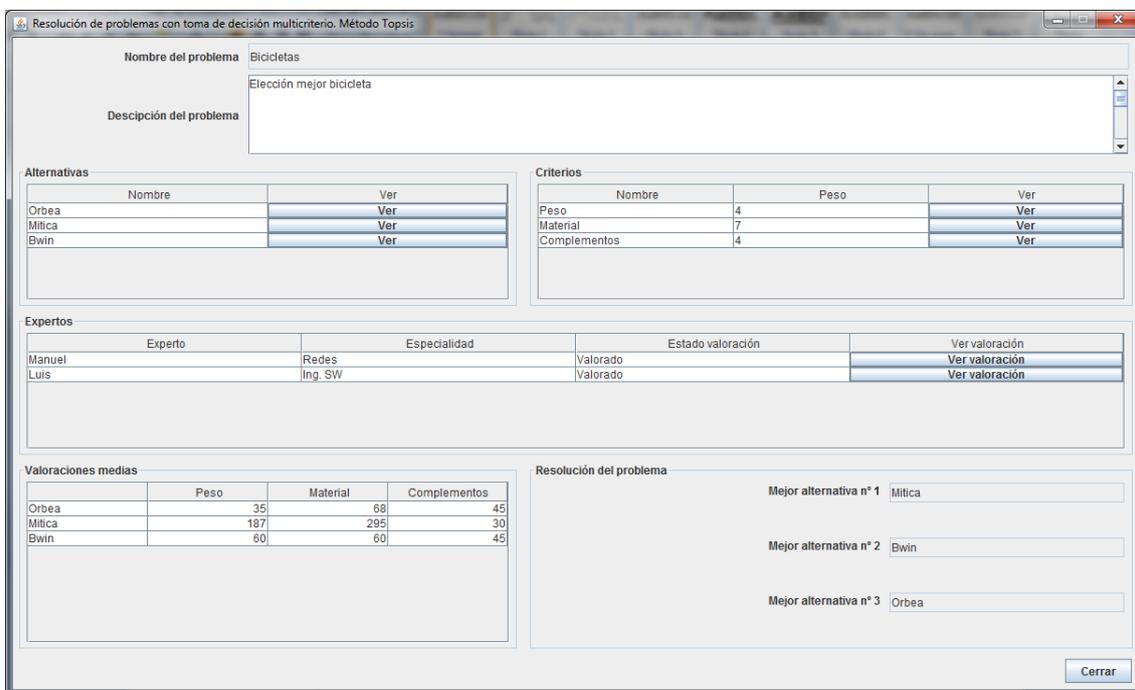
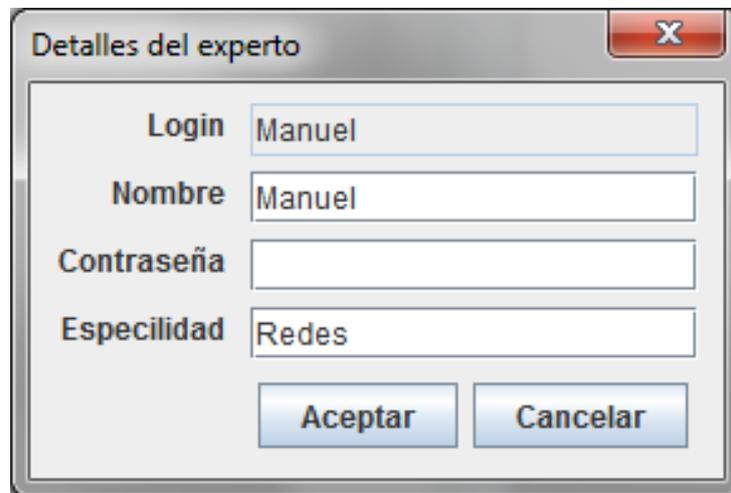


Figura II.14. Resolución problema

Ver/Modificar un Experto

El usuario administrador deberá pulsar el botón de "Ver/Modificar" del experto seleccionado en el listado de expertos que aparece en el apartado gestión de expertos de la pantalla principal. Ver Figura II.8, Pantalla principal 2.

El usuario administrador podrá ver la información de un experto. El usuario administrador tiene la posibilidad de modificar "Nombre", "Contraseña" y "Especialidad" (Ver la Figura II.15).



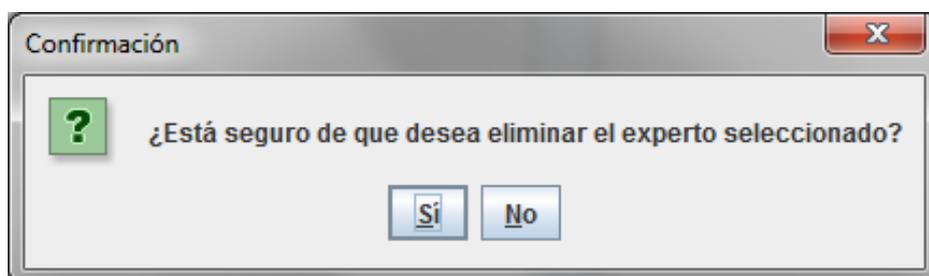
The image shows a dialog box titled "Detalles del experto". It has a close button (X) in the top right corner. Inside the dialog, there are four text input fields. The first is labeled "Login" and contains the text "Manuel". The second is labeled "Nombre" and also contains "Manuel". The third is labeled "Contraseña" and is empty. The fourth is labeled "Especialidad" and contains "Redes". At the bottom of the dialog, there are two buttons: "Aceptar" and "Cancelar".

Figura II.15. Modificar experto

Una vez que el usuario administrador ha completado, revisado o modificado toda la información del experto, pulsará el botón "Aceptar" para almacenar la información en el sistema.

Eliminar un Experto

El usuario administrador deberá pulsar el botón de "Eliminar" del experto seleccionado en el listado de expertos que aparece en el apartado gestión de expertos de la pantalla principal (Ver Figura II.8. Pantalla principal 2). El sistema pedirá confirmación de la eliminación del experto al usuario administrador. Ver Figura II.16.



The image shows a dialog box titled "Confirmación". It has a close button (X) in the top right corner. Inside the dialog, there is a green square icon with a white question mark. To the right of the icon is the text "¿Está seguro de que desea eliminar el experto seleccionado?". Below this text, there are two buttons: "Sí" and "No".

Figura II.16. Mensaje eliminar experto

Cerrar Sesión

El usuario administrador podrá salir de su sesión pulsando el botón "Cerrar Sesión" de la pantalla principal. El sistema mostrará la pantalla de acceso. Se puede ver en la Figura II.17.



Figura II.17. Cerrar sesión administrador

Anexo III. Manual de usuario (Experto)

En este Anexo III vamos a ver el modo de utilización de la aplicación de escritorio de Toma de Decisión Multicriterio mediante el método TOPSIS para el usuario Experto. Éste se encargará de dar sus opiniones o valoraciones y consultar la información de los diferentes problemas asociados para una toma de decisión correcta.

Pasemos a ver las diferentes acciones posibles del usuario experto:

Acceso

El usuario experto deberá introducir su login de Usuario y su Contraseña para acceder al sistema. Posteriormente pulsar "Aceptar" para entrar, o "Salir" para salir de la aplicación.



Figura III.1. Acceso usuario

Perfil de usuario

El usuario experto podrá modificar sus datos (Nombre, Especialidad, Contraseña). Para poder cambiarlo, pulsará el botón "Perfil de Usuario" de la pantalla principal de usuarios expertos (Ver Figura III.2)

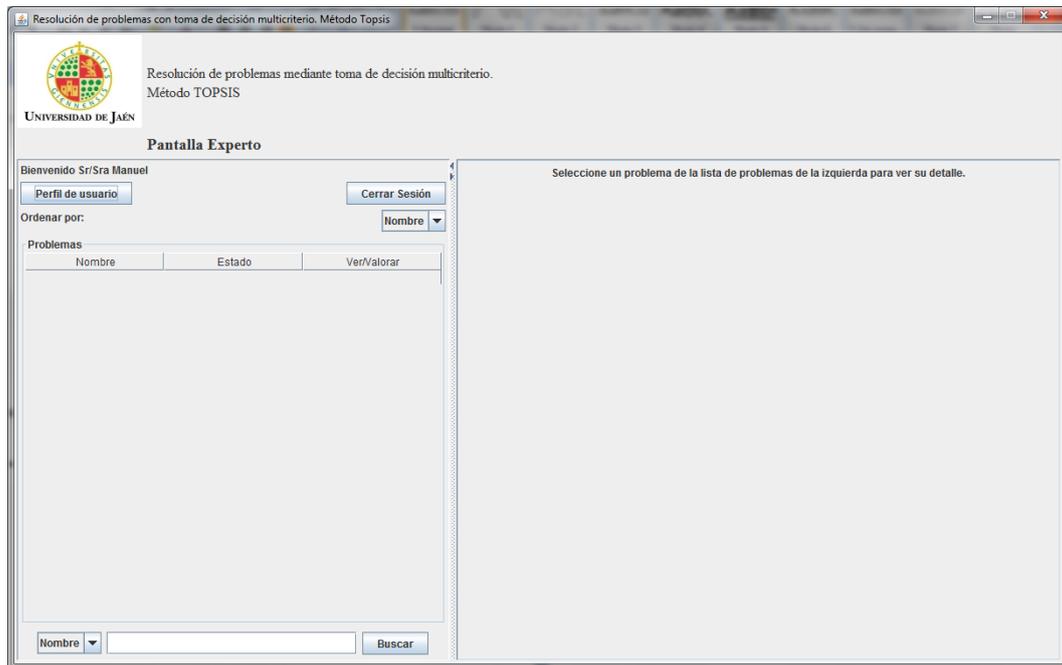


Figura III.2. Pantalla principal experto

Aparecerá una pantalla para cambiar el "Nombre" y la "Especialidad". Si el usuario experto lo desea, puede cambiar su contraseña pulsando el botón de "Cambiar contraseña".

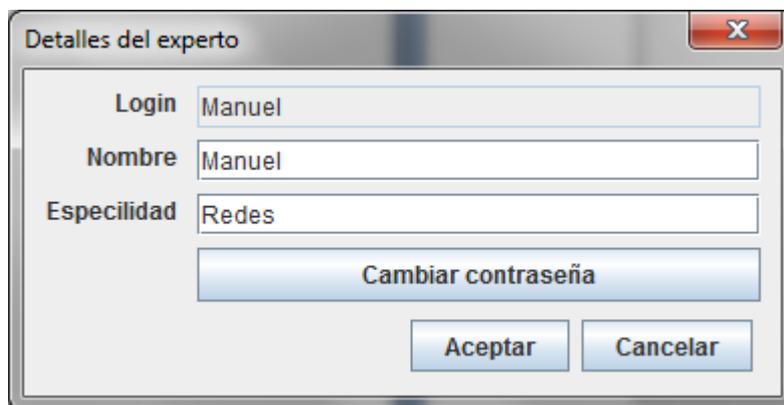


Figura III.3. Cambiar datos experto

El usuario experto podrá cambiar la contraseña introduciendo la "Contraseña actual" y escribiendo dos veces la "Nueva contraseña" como medida de prevención y seguridad (Ver Figura III.4)

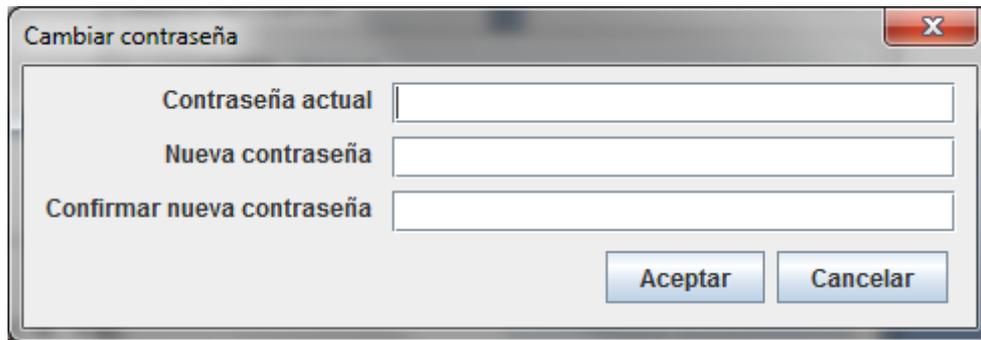


Figura III.4. Cambiar contraseña experto

Para confirmar las modificaciones, el usuario experto pulsará el botón "Aceptar", quedando todo almacenado en el sistema.

Visualización de un problema

El usuario experto podrá consultar la lista de problemas que tiene vinculados, mostrados en la pantalla principal. Ver Figura III.5.

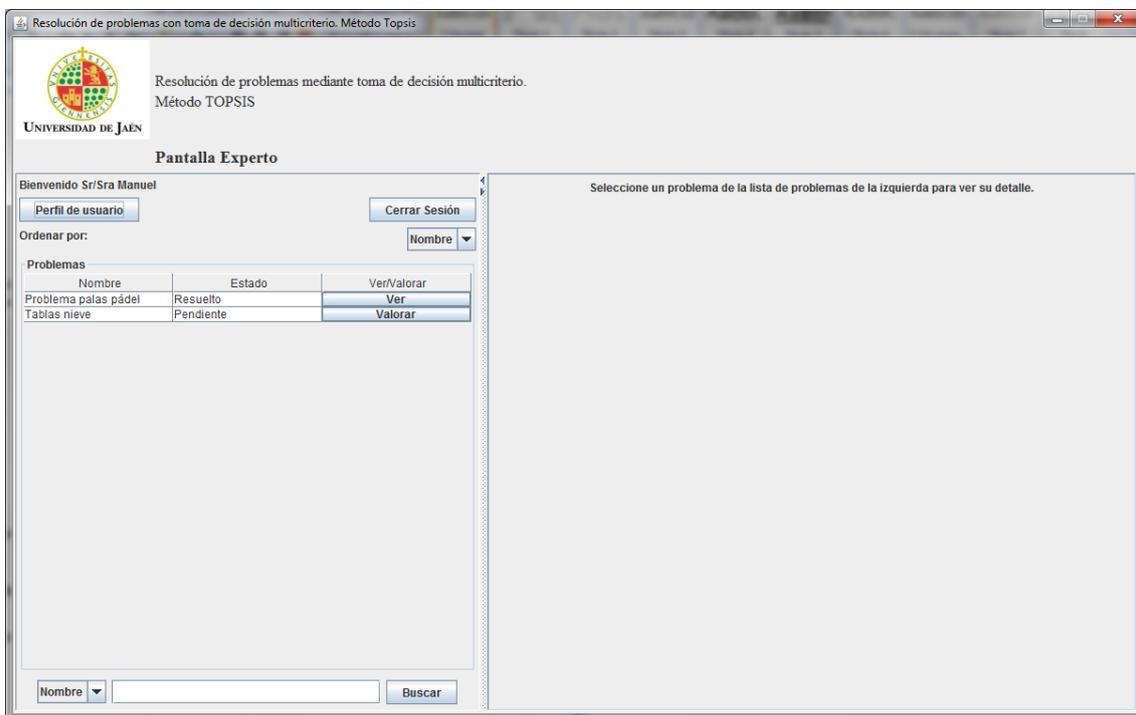


Figura III.5. Pantalla principal experto 2

Para poder consultar la información de un problema, el usuario experto pulsará el botón "Ver" (para problemas resueltos) o "Valorar" (para problemas no resueltos). Seguidamente en la pantalla principal se mostrará en la parte derecha, los detalles principales de un problema en la pestaña "General" (Ver Figura III.6).

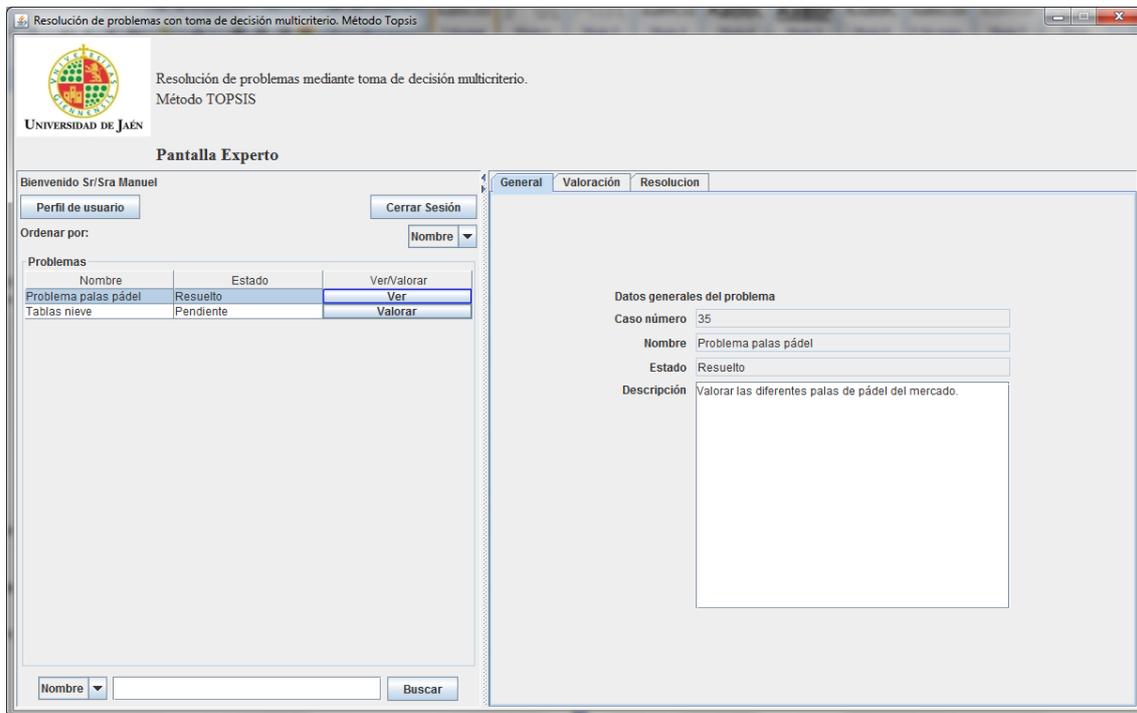


Figura III.6. Pestaña General

Consulta de la valoración y solución de un problema resuelto

El usuario experto podrá consultar la valoración introducida, la valoración media según el Método TOPSIS, y las 3 mejores alternativas de un problema para la Toma de Decisión.

El usuario experto pulsará el botón "Ver" de aquellos problemas que posean dicha opción (problemas resueltos), de la lista de problemas de la pantalla principal (problemas que tiene vinculado el experto). En la pantalla principal se mostrará, en la parte derecha, los detalles principales de un problema en la pestaña "General", la valoración realizada por el experto en la pestaña "Valoración"; y las valoraciones medias según el Método TOPSIS y las tres mejores alternativas del problema en la pestaña "Resolución" (Ver Figura III.7 y Figura III.8).

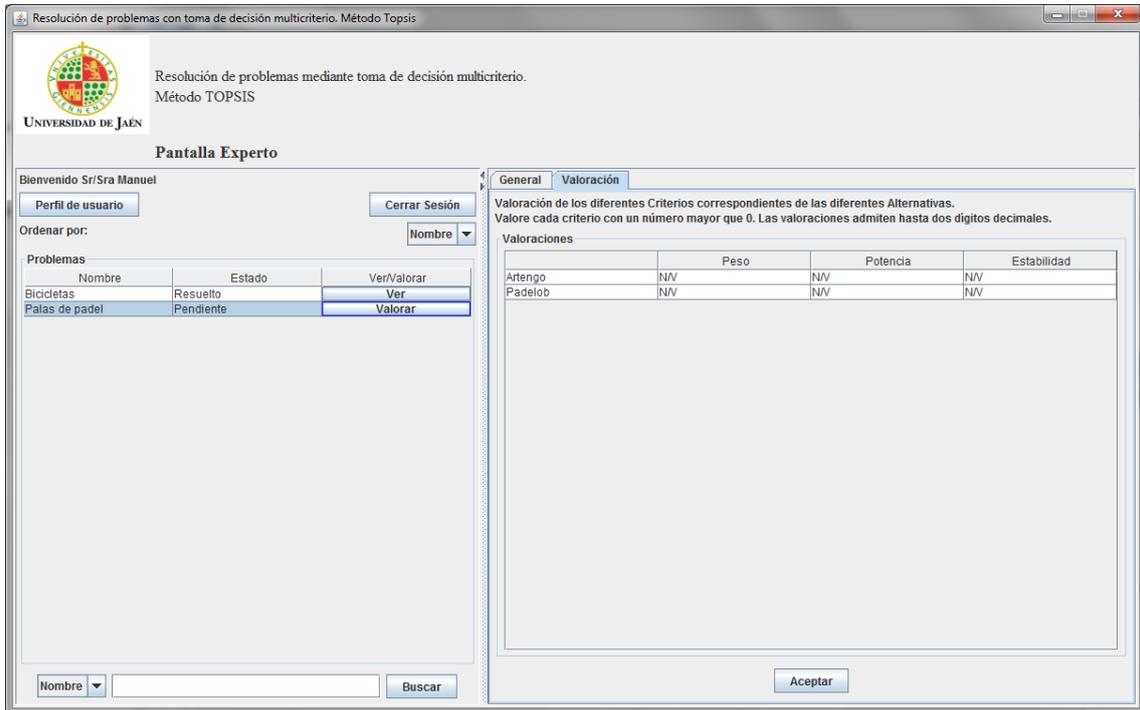


Figura III.7. Pestaña Valoración

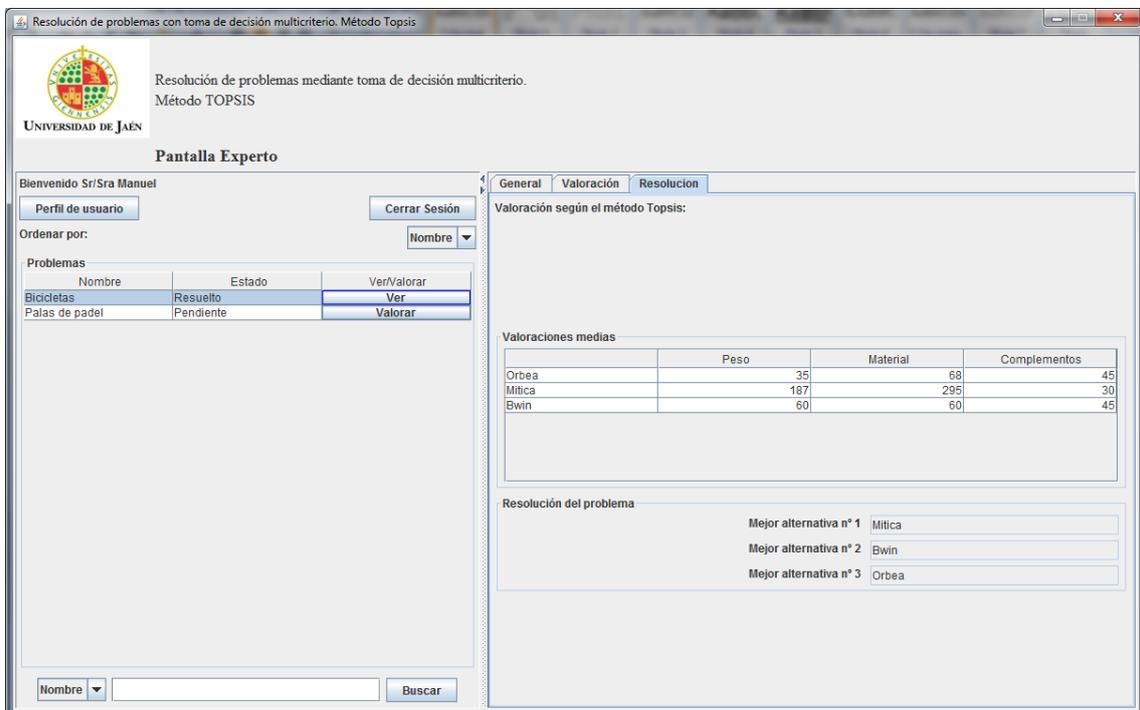


Figura III.8. Pestaña Resolución

Valorar un problema

El usuario experto podrá valorar un problema. Para ello, el usuario experto pulsará el botón "Valorar" de aquellos problemas que posean dicha opción, de la lista de problemas de la pantalla principal (problemas que tiene vinculado el experto). En la pantalla principal se mostrará, en la parte derecha, los detalles principales de un problema en la pestaña "General"; y la opción de valorar un problema en la pestaña "Valoración".

Una vez pulsada la pestaña "Valoración", el usuario experto podrá ir asignando valores mayores que 0 y con dos decimales máximo a las diferentes alternativas y criterios. Si el usuario introduce cualquier dato erróneo, el sistema lo cambia por un N/V y se considera no valorado. El usuario deberá pulsar fuera de la celda una vez que haya introducido la valoración para que el sistema admita las modificaciones de la celda.

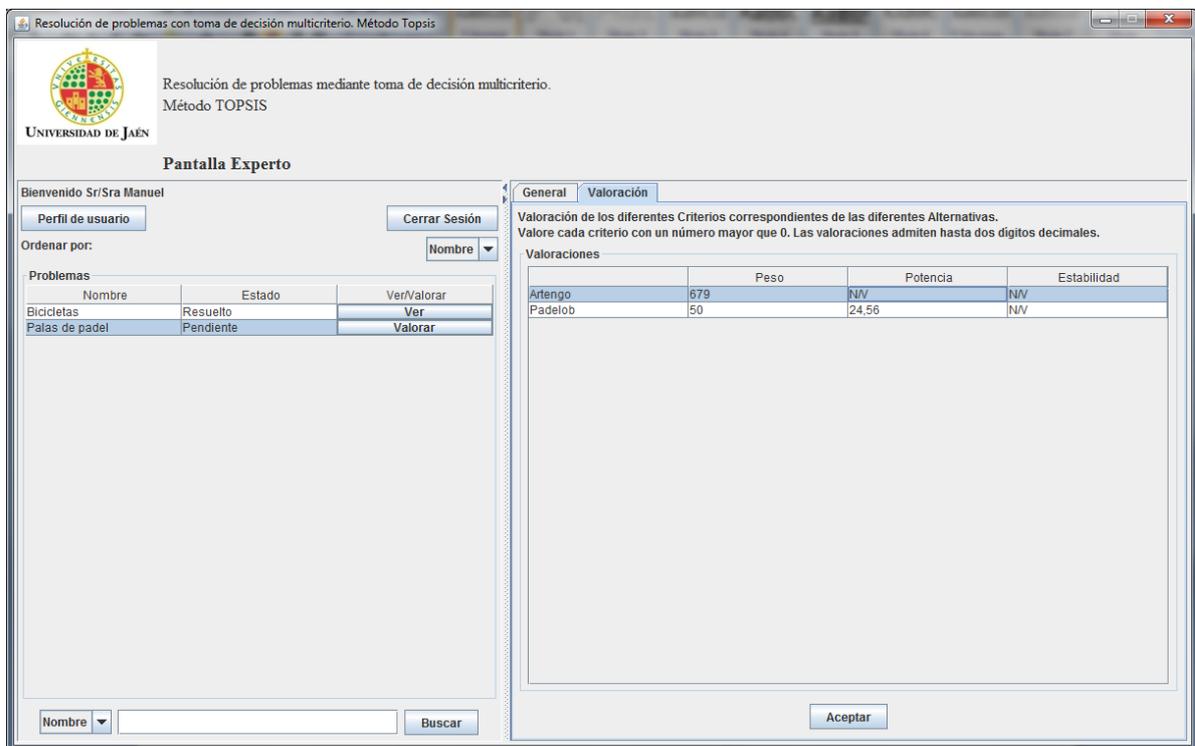


Figura III.9. Pestaña Valoración 2

El usuario experto pulsará el botón "Aceptar" para almacenar su valoración parcial o total en el sistema.

Cerrar Sesión

El usuario experto podrá salir de su sesión pulsando el botón "Cerrar Sesión" de la pantalla principal. El sistema mostrará la pantalla de acceso (Ver Figura III.10).

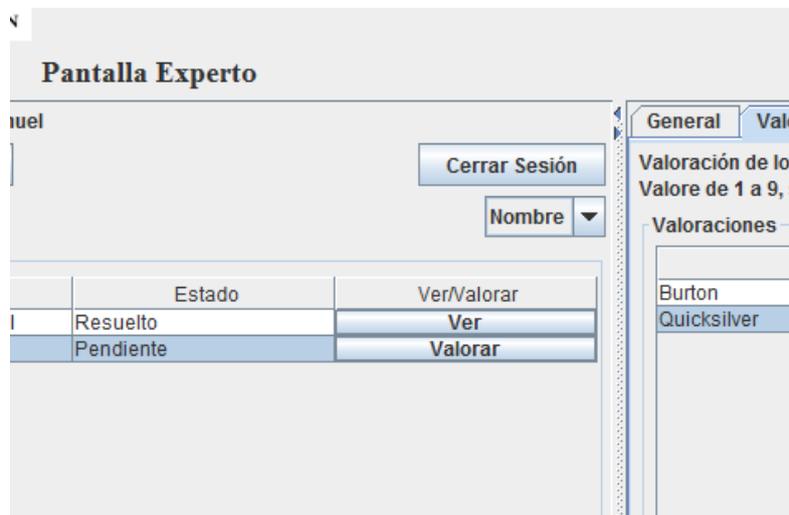


Figura III.10. Cerrar sesión experto

