



TRABAJO FIN DE GRADO  
GRADO EN INGENIERÍA INFORMÁTICA

# Sistema de ayuda a la decisión en grupo basado en el paradigma de la computación granular

---

**Autor**

Juan Carlos González Quesada

**Director**

Francisco Javier Cabrerizo Lorite



ESCUELA TÉCNICA SUPERIOR DE INGENIERÍAS INFORMÁTICA Y DE  
TELECOMUNICACIÓN

Granada, julio de 2021









# Sistema de ayuda a la decisión en grupo basado en el paradigma de la computación granular

---

## **Autor**

Juan Carlos González Quesada

## **Director**

Francisco Javier Cabrerizo Lorite



DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN E INTELIGENCIA  
ARTIFICIAL

---

Granada, julio de 2021



# **Sistema de ayuda a la decisión en grupo basado en el paradigma de la computación granular**

Juan Carlos González Quesada

**Palabras clave:** coherencia, computación granular, consenso, sistema de ayuda a la decisión, web.

## **Resumen**

Cuando la adopción de una decisión depende de las opiniones de un grupo de personas es esencial tanto que estas sean coherentes como que se alcance un consenso antes de tomar la decisión. Para ello, es usual que los integrantes del grupo tengan que modificar sus opiniones iniciales. Sin embargo, si tienen que modificar mucho sus opiniones, es posible que sean reacios a ello. El objetivo de este trabajo fin de grado es el de desarrollar e implementar, como una aplicación web, un sistema de ayuda a la decisión en grupo basado en el paradigma de la computación granular que permita mejorar la coherencia individual y el consenso alcanzado a la vez que mantiene la máxima similitud posible entre las opiniones iniciales de los integrantes del grupo y las modificadas. Este sistema será capaz de modelar problemas de toma de decisiones en grupo realizados en entornos multicriterio (se consideran varios criterios) y heterogéneos (los integrantes del grupo tienen distinto nivel de conocimiento sobre el problema). Este nuevo sistema también permitirá automatizar los procesos de toma de decisiones en grupo y facilitará a sus integrantes la participación en el proceso de decisión.

# **A group decision support system based on the granular computing paradigm**

Juan Carlos González Quesada

**Keywords:** consistency, granular computing, consensus, decision support system, web.

## **Abstract**

When the adoption of a decision depends on the opinions given by a group of individuals, these opinions must be consistent and the consensus achieved must be as higher as possible before making the decision. To do this, it is usual that the individuals have to adjust their initial opinions. However, if they have to modify their opinions a great deal, they may be reluctant to do so. The objective of this Bachelor's Degree Final Project is to develop and implement, as a web application, a group decision support system based on the granular computing paradigm allowing to improve the individual consistency and the consensus achieved while maximizing as high as possible the similarity between the inicial opinions given by the individuals and the adjusted ones. It is able to model group decision-making problems carried out in multi-criteria and heterogenous environments (those in which several criteria are considered and in which the individuals have different knowledge levels about the problem). This new decision support system also allows to automatize the group decision-making processes and makes easy the participation of the individuals in the decision-making process.



---

Yo, **Juan Carlos González Quesada**, alumno de la titulación INGENIERÍA INFORMÁTICA de la **Escuela Técnica Superior de Ingenierías Informática y de Telecomunicación de la Universidad de Granada**, con DNI 77765614-S, autorizo la ubicación de la siguiente copia de mi Trabajo Fin de Grado en la biblioteca del centro para que pueda ser consultada por las personas que lo deseen.

Fdo: Juan Carlos González Quesada

Granada, a 9 de julio de 2021.



---

D. **Francisco Javier Cabrerizo Lorite**, Profesor del Área de Ciencias de la Computación e Inteligencia Artificial del Departamento de Ciencias de la Computación e Inteligencia Artificial de la Universidad de Granada.

**Informa:**

Que el presente trabajo, titulado *Sistema de ayuda a la decisión en grupo basado en el paradigma de la computación granular*, ha sido realizado bajo su supervisión por **Juan Carlos González Quesada**, y autoriza la defensa de dicho trabajo ante el tribunal que corresponda.

Y para que conste, expide y firma el presente informe en Granada, a 9 de julio de 2021.

**El director:**

**Francisco Javier Cabrerizo Lorite**



# Agradecimientos

Si puedo escribir estas palabras es porque he llegado al final de este largo camino. Ha tenido muchas complicaciones, altibajos y momentos de dificultad, pero los he podido superar gracias a mi madre, M<sup>a</sup> Encarnación Quesada. Gracias por todo lo que has hecho por mí, siempre estaré en deuda contigo, como ya te dije en primaria, eres mi heroína favorita. Por supuesto, agradecer a mi hermana, María González, y a mis abuelos, M<sup>a</sup> Carmen y Francisco Quesada, el apoyo incondicional que me han dado durante todos estos años.

Como no, agradecer a mi director de proyecto, Francisco Javier Cabrerizo Lorite, por brindarme la oportunidad de realizar un proyecto conjunto. Desde pequeño, siempre me han apasionado las matemáticas y haber podido realizar mi proyecto final uniendo las matemáticas con la informática ha sido una experiencia muy reconfortante. Agradecer también las enseñanzas, consejos e indicaciones que me has ido dando, debido a que han sido fundamentales para enfocar el sistema y lograr los objetivos marcados. Ojalá podamos realizar muchos más proyectos y trabajos conjuntamente, pues lo que he aprendido y puedo aprender es increíble.

De la universidad, me llevo que no todo lo que reluce es oro, y que quienes verdaderamente están contigo, te lo demuestran día a día. Agradecer a nuestro grupo «Los Cocos»: Marina Hernández, Pedro Jiménez y Román Larrosa, toda la ayuda que nos hemos prestado, los momentos y las risas vividas. También agradecer a mi compañero de viaje, Víctor Rubia, su ayuda incondicional y querer estar siempre a mi lado.

Finalizaré con una frase de Gandhi: «No hay que apagar la luz del otro para hacer que brille la tuya propia».

Gracias a todos por el apoyo, este proyecto tiene un poquito de vosotros.



# Índice general

|  |           |
|--|-----------|
| <b>1. Introducción</b>   | <b>17</b> |
| 1.1. Planteamiento . . . . .   | 17        |
| 1.2. Objetivos . . . . .   | 20        |
| 1.3. Estructura de la memoria . . . . .  | 20        |
| <b>2. Modelo de toma de decisiones en grupo basado en computación granular</b> | <b>23</b> |
| 2.1. La toma de decisiones en grupo . . . . .                                  | 23        |
| 2.2. La computación granular en la toma de decisiones . . . . .                | 25        |
| 2.3. Modelo propuesto . . . . .  | 27        |
| 2.3.1. Expresión de las evaluaciones . . . . .                                 | 27        |
| 2.3.2. Mejora de la coherencia y el consenso con mínimo ajuste                 | 28        |
| 2.3.3. Proceso de selección . . . . .  | 34        |
| <b>3. Desarrollo del sistema web de ayuda a la decisión en grupo</b>           | <b>39</b> |
| 3.1. Planificación, metodologías y costes asociados . . . . .                  | 39        |
| 3.1.1. Planificación . . . . .   | 39        |
| 3.1.2. Metodologías . . . . .  | 41        |
| 3.1.3. Metodología Kanban . . . . .  | 41        |
| 3.1.4. Desarrollo de prototipos . . . . .                                      | 43        |
| 3.1.5. Costes . . . . .  | 43        |
| 3.2. Análisis . . . . .  | 45        |
| 3.2.1. Requisitos no funcionales . . . . .                                     | 45        |
| 3.2.2. Requisitos funcionales . . . . .  | 45        |
| 3.2.3. Requisitos de información . . . . .                                     | 49        |
| 3.2.4. Casos de uso . . . . .  | 50        |
| 3.3. Diseño . . . . .  | 55        |
| 3.3.1. Diagrama arquitectónico del sistema . . . . .                           | 55        |
| 3.3.2. Diagramas de secuencia . . . . .  | 55        |
| 3.3.3. Diagrama Entidad-Relación . . . . .                                     | 57        |
| 3.3.4. Diagrama de clases . . . . .  | 58        |
| 3.3.5. Patrones de diseño . . . . .  | 58        |
| 3.3.6. Prototipado final del proceso de diseño . . . . .                       | 60        |

|   |           |
|---|-----------|
| 3.4. Implementación . . . . .                                     | 63        |
| 3.4.1. Herramientas utilizadas . . . . .                          | 63        |
| 3.4.2. Seguridad y prevención de errores . . . . .                | 64        |
| 3.5. Pruebas . . . . .  | 68        |
| 3.5.1. Versiones del sistema . . . . .                            | 68        |
| 3.5.2. Pruebas realizadas . . . . .                               | 69        |
| <b>4. Aplicación y análisis</b>                                   | <b>77</b> |
| 4.1. Ejemplo ilustrativo . . . . .                                | 77        |
| 4.2. Análisis . . . . .   | 82        |
| 4.2.1. Nivel de granularidad de la información $\alpha$ . . . . . | 83        |
| 4.2.2. Parámetro $\gamma$ . . . . .                               | 84        |
| 4.2.3. Parámetro $\beta$ . . . . .                                | 84        |
| <b>5. Conclusiones y vías futuras</b>                             | <b>87</b> |
| 5.1. Conclusión . . . . .   | 87        |
| 5.2. Vías futuras . . . . .                                       | 88        |
| <b>A. Manual de usuario</b>                                       | <b>89</b> |
| A.1. Instalación . . . . .  | 89        |
| A.2. Administrador . . . . .                                      | 89        |
| A.3. Usuario . . . . .  | 91        |
| A.4. Algunos ejemplos . . . . .                                   | 91        |
| <b>B. Conocimientos utilizados</b>                                | <b>95</b> |
| <b>Bibliografía</b>   | <b>97</b> |



# Índice de figuras

|  |    |
|--|----|
| 3.1. Diagrama de Gantt del trabajo fin de grado. . . . .                   | 40 |
| 3.2. Ejemplo de diagrama Kanban. . . . .                                   | 42 |
| 3.3. Proceso del desarrollo de prototipos. . . . .                         | 43 |
| 3.4. Diagrama de casos de uso del sistema. . . . .                         | 51 |
| 3.5. Diagrama arquitectónico del sistema. . . . .                          | 55 |
| 3.6. Diagrama de secuencia para la creación de un problema. . . .          | 56 |
| 3.7. Diagrama de secuencia para opinar en un problema. . . . .             | 56 |
| 3.8. Diagrama de secuencia para el cálculo de un problema. . . . .         | 57 |
| 3.9. Diagrama de entidad-relación de la base de datos. . . . .             | 57 |
| 3.10. Diagrama de clases del sistema. . . . .                              | 58 |
| 3.11. Diagrama de clases del patrón Strategy. . . . .                      | 59 |
| 3.12. <i>Mockup</i> de la acción de opinar. . . . .                        | 60 |
| 3.13. <i>Mockup</i> de la acción de visualizar la solución de un problema. | 61 |
| 3.14. MoodBoard del sistema. . . . .                                       | 62 |
| 3.15. Mapa del sitio web para el administrador. . . . .                    | 62 |
| 3.16. Mapa del sitio web para los usuarios. . . . .                        | 63 |
| 4.1. Ejemplo de evaluaciones expresadas usando la aplicación web.          | 79 |
| 4.2. Visualización de la solución del problema en la aplicación web.       | 82 |
| A.1. Página principal del sistema. . . . .                                 | 90 |
| A.2. Página principal del administrador. . . . .                           | 90 |
| A.3. Página principal de experto. . . . .                                  | 91 |
| A.4. Página de opinar. . . . .   | 92 |
| A.5. Página de visualización de las soluciones. . . . .                    | 92 |
| A.6. Página de creación de problemas. Parte 1. . . . .                     | 93 |
| A.7. Página de creación de problemas. Parte 2. . . . .                     | 93 |
| A.8. Página de creación de problemas. Parte 3. . . . .                     | 93 |
| A.9. Selección de un problema para calcular su solución. . . . .           | 93 |
| A.10. Selección de un problema para visualizarlo. . . . .                  | 94 |



# Índice de tablas

|      |  |    |
|------|--|----|
| 3.1. | Tabla descriptiva de los costes del proyecto. . . . .  | 44 |
| 4.1. | Valores de $f$ , $f_1$ , $f_2$ y $f_3$ , para distintos valores de $\alpha$ . . . . .          | 83 |
| 4.2. | Valores de $f_1$ y $f_2$ para determinados valores de $\gamma$ . . . . .                       | 84 |
| 4.3. | Valores de $0.5 \cdot f_1 + 0.5 \cdot f_2$ y $f_3$ para distintos valores de $\beta$ . . . . . | 84 |



# Capítulo 1

## Introducción

### 1.1. Planteamiento

Tanto los animales como los seres humanos disponen de cierta capacidad de decisión que supedita su conducta cuando se enfrentan a determinadas situaciones. Sin embargo, la particularidad que permite diferenciar a los seres humanos en relación con los animales es la complejidad con la que los primeros desarrollan la toma de decisiones [5]. Esta se define como el proceso cognitivo que se basa en seleccionar la alternativa, opción, curso de acción, etc., que se considera mejor de entre un conjunto de estas.

Los seres humanos se enfrentan a todas horas a procesos de toma de decisiones. Por ejemplo, seleccionar qué se va a comer en el desayuno, el tipo de ropa para una cierta ocasión, a quién votar para un cargo determinado, etc. En particular, en la mayoría de los procesos de toma de decisiones, son varias personas las que deben adoptar la decisión, y no una única. Por ejemplo, pensemos en un grupo de amigos que deben decidir a qué restaurante ir. Esta clase de procesos de decisión se denomina toma de decisiones en grupo [13], ya que es un grupo de personas las que deben ponerse de acuerdo para elegir la alternativa más conveniente.

Vemos, por tanto, que los procesos de toma de decisiones se pueden realizar individualmente o en grupo. Sin embargo, de acuerdo con las investigaciones sobre el rendimiento grupal llevadas a cabo en psicología social, se comprueba que, por un lado, el grupo es más eficiente que la unión directa de las selecciones individuales de los miembros del grupo y, por otro, que el grupo tiende a tomar mejores decisiones que su integrante más preparado [55]. No obstante, al intervenir un cierto número de personas en la selección de la mejor alternativa, debemos tener en cuenta una serie de consideraciones adicionales como, por ejemplo, el consenso alcanzado entre el grupo. De esta forma, es recomendable que se lleve a cabo un proceso de consenso [7], en el cual los integrantes del grupo discuten sus motivaciones y tratan de acercar posturas con el objetivo de llegar a una decisión que sea suficiente-

mente aceptable para todos. El consenso, en esencia, trata de conseguir el consentimiento, pero no el acuerdo total, de las personas que deben tomar la decisión, la cual tratará de ser útil para todo el grupo y no para una parte de este. Esto implica que algunos integrantes del grupo podrían dar su consentimiento a una decisión que no es su primera opción, ya que quieren cooperar con el grupo. Ahora bien, el consentimiento no significa que se esté totalmente de acuerdo con la decisión tomada [7]. Resumiendo, el hecho de que se consiga un acuerdo no implica que todos los integrantes del grupo estén totalmente de acuerdo, situación compleja de conseguir en grupos formados por personas que se asumen inteligentes y que tienen conocimientos diversos.

Un problema de toma de decisiones en grupo se compone generalmente de dos procesos que se ejecutan de forma secuencial para obtener la solución del problema. El primero de ellos se denomina proceso de consenso [23] y, como hemos descrito, consiste en que los integrantes del grupo argumenten y razonen sus opiniones, ideas, evaluaciones, etc., con el fin de lograr el mayor nivel de acuerdo posible. El segundo de ellos se denomina proceso de selección [16] y consiste en seleccionar la mejor alternativa teniendo en cuenta las evaluaciones proporcionadas por los integrantes del grupo.

A la hora de modelar de forma correcta un problema de toma de decisiones en grupo, además del nivel de acuerdo alcanzado entre los integrantes del grupo, algo que ya hemos visto que es de suma importancia, hay otros aspectos que se deben tener en cuenta:

- Similitud entre las evaluaciones originales y las modificadas. Por definición, el consenso implica que los integrantes del grupo tienen que estar dispuestos a modificar sus evaluaciones iniciales con el objetivo de acercar posturas y mejorar el consenso. Esto implica que deben admitir un cierto grado de flexibilidad, dentro del cual sus evaluaciones iniciales se pueden ajustar. Aún así, si las nuevas evaluaciones sugeridas para mejorar el consenso son muy diferentes de las proporcionadas inicialmente por el integrante del grupo, este podría rechazarlas. Por lo tanto, es importante minimizar el ajuste, es decir, incrementar la similitud entre las evaluaciones iniciales y las ajustadas [59].
- La coherencia de los integrantes del grupo. Por un lado, es recomendable que los integrantes del grupo posean opiniones diversas, aun siendo opuestas. Esto supone un mayor análisis del problema e implica la discusión entre los integrantes del grupo. Pero, por otro, la incoherencia en las propias opiniones de una persona no son aceptables [14]. Las personas que expresan opiniones incoherentes no suelen ser creíbles para el resto. Además, pueden generar decisiones poco fiables. Por tanto, el estudio de la coherencia en las evaluaciones individuales de los integrantes del grupo debe considerarse a la hora de resolver un problema de toma de decisiones en grupo [31].

- Los criterios que se tienen en mente cuando se evalúan las distintas alternativas. En general, los modelos desarrollados para resolver problemas de toma de decisiones en grupo consideran que los integrantes del grupo evalúan las diferentes alternativas en su conjunto. Aunque esto es posible, la evaluación de las alternativas de acuerdo con una serie de criterios es más realista y permite ofrecer más información a la hora de obtener la mejor alternativa. Como ejemplo, una casa puede ser mejor que otra respecto al precio, pero no respecto a la ubicación. En estos casos, nos encontramos con problemas de toma de decisiones en grupo definidos en ambientes multicriterio [16, 38].
- Heterogeneidad. Dentro de los problemas de toma de decisiones en grupo podemos encontrar heterogeneidad en el sentido de que, por un lado, los integrantes del grupo poseen distinto conocimiento sobre el problema a resolver. Esto se suele modelar asignando diferentes niveles de importancia a los integrantes del grupo [30]. Y, por otro, si estamos ante un problema de toma de decisiones en grupo multicriterio, los criterios también pueden tener distintos niveles de importancia de cara a la evaluación de las alternativas [27].

En la última década, se ha comenzado a emplear el paradigma de la computación granular para modelar procesos de toma de decisiones en grupo. Este es un paradigma de computación que tiene un menor coste computacional para el procesamiento de información que el necesitado cuando se realiza un procesamiento numérico preciso [2]. Ha surgido como una metodología sólida y consistente de construcción, descripción y procesamiento de gránulos de información, los cuales son el concepto subyacente que provoca que entidades no específicas y semánticamente significativas sean esenciales para la comprensión del mundo real, caracterizando sus fenómenos y apoyando los procesos de toma de decisiones [40]. Aunque se han desarrollado diferentes modelos de toma de decisiones en grupo basados en computación granular, estos, como se verá más adelante en esta memoria, solo tienen en cuenta algunos de los aspectos descritos anteriormente, pero no existe un modelo que sea capaz de considerar todos esos aspectos al mismo tiempo.

La hipótesis general de partida de este trabajo fin de grado se basa en el hecho de que se pueden mejorar los actuales modelos de toma de decisiones en grupo basados en computación granular al desarrollar un nuevo modelo que tenga en cuenta todos los aspectos anteriores. Además, la implementación de este modelo como una aplicación web permitirá, por un lado, automatizar los procesos de toma de decisiones en grupo y, por otro, que las personas que intervienen en el problema de toma de decisiones no tengan que estar físicamente en la misma ubicación, facilitándoles así el proporcionar sus evaluaciones en cualquier momento y lugar. Esto originará un modelo más realista y coherente con los ambientes reales y actuales de decisión.

## 1.2. Objetivos

El objetivo de este trabajo fin de grado es el desarrollo e implementación de un sistema de ayuda a la decisión en grupo basado en el paradigma de la computación granular que permita, por un lado, modelar problemas de toma de decisiones en grupo efectuados en entornos multicriterio y heterogéneos y, por otro, mejorar la coherencia individual y el consenso conseguido entre los integrantes del grupo con el menor ajuste posible en las evaluaciones proporcionadas por estos. Este objetivo general se conseguirá a través de la obtención de los objetivos específicos que enumeramos a continuación:

- Desarrollo de un método para calcular el consenso entre las personas que participan en un problema de toma de decisiones en grupo definido en un entorno multicriterio y heterogéneo.
- Desarrollo de un método para calcular la similitud entre las evaluaciones proporcionadas por los integrantes del grupo y las sugeridas por el modelo con el objetivo de mejorar la coherencia individual y el consenso en un problema de toma de decisiones en grupo definido en un entorno multicriterio y heterogéneo.
- Desarrollo de un método basado en computación granular que incremente tanto la coherencia individual como el consenso lo máximo posible mientras que modifica las evaluaciones originales proporcionadas por los integrantes del grupo lo mínimo posible.
- Desarrollo de un método de selección de alternativas para problemas de toma de decisiones en grupo llevados a cabo en entornos multicriterio y heterogéneos.
- Implementación del modelo de toma de decisiones en grupo basado en computación granular como un sistema o aplicación web con el objetivo de automatizar todo el proceso de toma de decisiones.

## 1.3. Estructura de la memoria

Para alcanzar los objetivos planteados, esta memoria de trabajo fin de grado se estructura en cinco capítulos, cuyo contenido es el siguiente:

- En el capítulo 1, se ha planteado el problema que se pretende resolver, la hipótesis de partida y los objetivos que se persiguen con el desarrollo de este trabajo fin de grado.
- En el capítulo 2, se describe de forma detallada el modelo de toma de decisiones en grupo basado en computación granular que se ha desarrollado. Previamente, se introducen brevemente los procesos de



toma de decisiones en grupo y se describen los desarrollos basados en el paradigma de la computación granular que se han presentado en esta área.

- En el capítulo 3, se realiza el desarrollo del sistema web de ayuda a la decisión basado en el modelo de toma de decisiones en grupo anterior. Para ello, se detallan las diferentes etapas llevadas a cabo: planificación, análisis, diseño, implementación y pruebas.
- En el capítulo 4, se resuelve un problema de toma de decisiones en grupo para ilustrar el funcionamiento del sistema de ayuda a la decisión desarrollado. Además, se analizan sus características.
- En el capítulo 5, se indican las conclusiones obtenidas a partir del desarrollo llevado a cabo en este trabajo fin de grado. También se esbozan algunos trabajos futuros.

La memoria concluye con dos apéndices, en los que se describen el manual de usuario de la aplicación web desarrollada (ver apéndice A) y los conocimientos necesarios que se han adquirido en diversas asignaturas para poder realizar este trabajo fin de grado (ver apéndice B), respectivamente, y con una colección bibliográfica de las publicaciones en congresos, libros, artículos, etc., más importantes para el desarrollo del presente trabajo fin de grado.



## Capítulo 2

# Modelo de toma de decisiones en grupo basado en computación granular

Este capítulo comienza con una breve introducción a los problemas de toma de decisiones en grupo. A continuación, se describe brevemente el paradigma de la computación granular y los modelos basados en este que se han desarrollado para modelar problemas de toma de decisiones en grupo. Finalmente, se presenta el modelo de toma de decisiones en grupo basado en computación granular desarrollado en este trabajo fin de grado.

### 2.1. La toma de decisiones en grupo

En la naturaleza de los seres humanos se encuentra la necesidad de tomar decisiones asiduamente. Todos los días nos enfrascamos en situaciones en las cuales tenemos que escoger una opción, alternativa, curso de acción, etc., en función del escenario concreto en el que nos encontramos [16]. Por ejemplo, seleccionar qué ruta coger para llegar a una determinada ubicación. Por tanto, parece coherente pensar que la investigación de procesos de toma de decisiones sea interesante en áreas como la inteligencia artificial, la economía y la ingeniería, por citar algunas, y no solo en teoría de la decisión.

Dentro de los procesos de toma de decisiones, de particular interés son aquellos desarrollados en grupo. En estos, un número de personas (usualmente expertos en el problema a resolver, aunque no siempre tiene por qué ser así) actúan colectivamente para escoger la mejor solución (alternativa, curso de acción, opción, etc.) de entre un conjunto de estas [42]. Para ello, los expertos deben proporcionar sus evaluaciones de las alternativas, siendo habitual la consideración de varios criterios para ello [16]. En tal caso, nos encontramos ante un problema de toma de decisiones en grupo con múltiples criterios o multicriterio [27], el cual está formalmente caracterizado por:

- Un conjunto finito de alternativas,  $A = \{a_1, a_2, \dots, a_n\}$ , siendo  $n \geq 2$ .
- Un grupo finito de expertos,  $E = \{e_1, e_2, \dots, e_m\}$ , siendo  $m \geq 2$ .
- Una colección finita de criterios,  $C = \{c_1, c_2, \dots, c_q\}$ , siendo  $q \geq 2$ , que se utilizan para evaluar las alternativas.
- Un peso de importancia  $w_l \in [0, 1]$  asociado a cada criterio  $c_l \in C$ . Generalmente, los pesos están normalizados, es decir,  $\sum_{l=1}^q w_l = 1$ .

Para resolver un problema de toma de decisiones en grupo con múltiples criterios se suelen llevar a cabo las siguientes acciones:

- Expresión de evaluaciones. Los expertos que participan en el proceso de decisión proporcionan sus evaluaciones de las alternativas utilizando un dominio de representación particular y un formato de representación concreto [25].
- Proceso de consenso. Para alcanzar un acuerdo suficiente antes de obtener la solución al problema, se debe llevar a cabo un proceso dinámico e iterativo de debate y revisión de las evaluaciones por parte de los integrantes del grupo [23, 48].
- Proceso de selección. Consiste en obtener una evaluación colectiva a partir de la agregación de todas las evaluaciones individuales expresadas por los integrantes del grupo. A continuación, la información contenida en esta evaluación colectiva se utiliza para clasificar las alternativas de mejor a peor como solución al problema. La primera alternativa de esta clasificación se selecciona como solución del problema de decisión [44].

La expresión de las evaluaciones de las alternativas por parte de un experto puede significar dos cosas diferentes. Por un lado, la evaluación puede determinar el grado de preferencia de una alternativa sobre otra según un cierto criterio. Por otro lado, la evaluación puede determinar el grado en el cual la alternativa satisface un determinado criterio. En el primer caso, se usa una comparación entre pares como formato de representación. En el segundo caso, se emplea un valor de utilidad como formato de representación. Aunque ambos formatos de representación han sido empleados en procesos de toma de decisiones, las comparaciones entre pares modelan mejor los procesos de toma de decisiones [34]. Sin embargo, este formato de representación produce más información de la que en realidad es necesaria. Además, la comprensión global de las alternativas por parte de los expertos se ve restringida. Esto puede dar lugar a que algunas evaluaciones sean contradictorias (incoherentes) [18]. Por tanto, es importante analizar la coherencia (consistencia) de las evaluaciones proporcionadas por los integrantes del grupo [31].

Una vez elegido el formato de representación, hay que establecer el dominio de representación en el que se modelarán las evaluaciones. En los procesos de toma de decisiones en grupo, al ser procesos cognitivos en los que participan seres humanos, la teoría de conjuntos difusos ha demostrado ser una buena herramienta para modelar las evaluaciones, las cuales están impregnadas de incertidumbre humana [3]. En particular, esta teoría, junto con sus extensiones [6], ha sido ampliamente utilizada como dominio de representación [25].

Por último, otro de los aspectos que hay que tener en cuenta es aquel de la diferente experiencia y nivel de conocimiento sobre los criterios y las alternativas de los expertos. Esto se suele modelar asignando a cada experto,  $e_z \in E$ , un peso o nivel de importancia,  $v_{zl} \in [0, 1]$ , que puede ser diferente para cada criterio,  $c_l \in C$ . En este caso, nos encontraríamos ante un proceso de toma de decisiones en grupo definido en un ambiente multicriterio y heterogéneo (o no homogéneo) [35].

## **2.2. La computación granular en la toma de decisiones en grupo**

La computación granular es un paradigma de computación emergente de procesamiento de información que abarca todas las herramientas, técnicas, metodologías y teorías, que emplean entidades de información complejas, denominadas gránulos de información, en la resolución de problemas complejos. Los gránulos de información, en términos generales, son colecciones de entidades, originadas generalmente en el nivel numérico, que se organizan de forma conjunta según su coherencia, adyacencia física o funcional, similitud o similares [2].

Los gránulos de información se pueden formalizar mediante distintos enfoques, por ejemplo, con los conjuntos sombreados [60], con conjuntos aproximados [21], con conjuntos probabilísticos [26], con conjuntos difusos [57, 58], o con intervalos [36], por citar algunos de ellos. A pesar de la evidente variedad de formalismos, existen varias características compartidas subyacentes que se articulan con respecto a los dominios de aplicación, desarrollos algorítmicos y fundamentos. Es interesante comprobar que los gránulos de información no se forman sin razón, sino que existe una clara motivación en su creación, es decir, surgen como una clara materialización del paradigma esencial de abstracción [40].

Para procesar los gránulos de información, independientemente del enfoque con el que se formalicen, es necesario buscar formas generales para formarlos [41]. Por un lado, el principio de granularidad justificable ofrece una opción viable. Basándose en pruebas experimentales disponibles, este principio se refiere a la creación de un gránulo de información con significado. Por otro lado, al entrar en escena resultados o datos no numéricos, la

granularidad de la información ayuda a lograr una conexión más adecuada con la realidad al cuantificar la naturaleza de los datos mediante gránulos de información. La granularidad de la información se puede ver como un activo de diseño fundamental, cuya utilización resulta esencial para desarrollar modelos más realistas. Este punto de vista da lugar a otro principio esencial de la computación granular, es decir, una distribución de la granularidad de la información junto con la optimización del proceso de distribución. En modelado de sistemas, los modelos existentes, independientemente de su origen, se elevan a un nuevo nivel de abstracción mediante una distribución de la granularidad de la información, conociéndose entonces como modelos granulares [40].

Estos dos fundamentos algorítmicos y conceptuales de la computación granular son esenciales para el desarrollo de modelos granulares. En particular, en los procesos de toma de decisiones en grupo [20, 37], la distribución de la granularidad de la información se ha utilizado para construir modelos que reflejen mejor la realidad y capturen la variedad de fuentes de conocimiento y puntos de vista expresados por los diferentes integrantes del grupo.

Un proceso de toma de decisiones en grupo es un claro ejemplo de distribución de la granularidad de la información en el que la asignación de la granularidad surge como un elemento crucial para construir, por ejemplo, el consenso, una cuestión, que como ya hemos visto, es muy importante en este tipo de procesos de toma de decisiones. Llegar a un consenso consiste en obtener una solución con la que cada integrante del grupo se siente a gusto [7, 19]. Para llegar a esta situación, obviamente, cada integrante del grupo debe permitir un cierto grado de flexibilidad y estar dispuesto a «suavizar» su postura (evaluación) y, aquí, la granularidad de la información puede entrar en juego [42]. En pocas palabras, al admitir una realización granular de las evaluaciones de los integrantes del grupo, en lugar de evaluaciones numéricas precisas, se facilita la colaboración entre ellos y que se llegue a un consenso.

En la literatura, podemos encontrar diferentes modelos de toma de decisiones en grupo desarrollados desde la perspectiva del paradigma de la computación granular. En concreto, el fundamento algorítmico y conceptual de la granularidad de la información ha sido utilizado para ayudar a llegar a un consenso en problemas de toma de decisiones en grupo. Pero la granularidad de la información no ha sido usada solamente para mejorar el consenso, sino que también se ha empleado para mejorar la coherencia [14, 18], otro aspecto cuya importancia en los procesos de toma de decisiones en grupo ya se ha descrito. Así, por ejemplo, y por citar algunos ejemplos, en [42] se presenta un modelo para mejorar tanto el consenso como la coherencia individual en el contexto del proceso de jerarquía analítica (conocido como AHP por sus siglas en inglés); en [11] se desarrolla un modelo para mejorar tanto el consenso como la coherencia pero, en este caso, cuando se utilizan relaciones de preferencia difusas para modelar las evaluaciones proporciona-

das por los integrantes del grupo; en [10] se describe un modelo para mejorar la consistencia multiplicativa, un tipo de coherencia, en relaciones de preferencia recíprocas; en [12] se presenta un modelo para mejorar el consenso cuando se utilizan relaciones de preferencia recíprocas intuicionistas; en [32] se introduce un modelo mejorado del presentado en [42] para mejorar el consenso y la coherencia en el contexto del proceso de jerarquía analítica; en [9] se presenta un modelo para mejorar el consenso con un ajuste mínimo de las evaluaciones en un entorno multicriterio; y en [8] se desarrolla un modelo para mejorar la coherencia en contextos multicriterio y heterogéneos.

No hace falta decir que existe un gran número de modelos basados en el paradigma de la computación granular, en concreto, en el concepto de la granularidad de la información, que tratan diferentes características o aspectos de los procesos de toma de decisiones en grupo. Sin embargo, ninguno de ellos es capaz de tratar con todas esas características al mismo tiempo.

## **2.3. Modelo propuesto**

En este trabajo fin de grado, haciendo uso del paradigma de la computación granular, en concreto, del concepto de distribución de la granularidad de la información, se desarrolla un nuevo modelo granular para modelar los procesos de toma de decisiones en grupo. Primero, este nuevo modelo granular tiene la capacidad de trabajar en entornos multicriterio, es decir, en procesos de decisión en los que se deben tener en cuenta diferentes criterios para evaluar las alternativas. Segundo, este nuevo modelo es capaz de manejar entornos heterogéneos, es decir, procesos de decisión en los que cada integrante del grupo tiene un conocimiento diferente sobre las alternativas y los criterios. Tercero, es capaz de mejorar tanto la coherencia de los integrantes del grupo como el consenso alcanzado entre ellos. Y, cuarto, es capaz de realizar esas mejoras con el ajuste mínimo, es decir, modificando lo menos posible las evaluaciones proporcionadas por los integrantes del grupo.

Este nuevo modelo basado en el paradigma de la computación granular se estructura en tres etapas:

- Expresión de las evaluaciones.
- Mejora de la coherencia y el consenso con mínimo ajuste.
- Proceso de selección.

A continuación, se describen detalladamente cada una de las etapas anteriores.

### **2.3.1. Expresión de las evaluaciones**

Al inicio de este capítulo, hemos mencionado que existen diferentes dominios y formatos de representación para caracterizar las evaluaciones pro-

porcionadas por los expertos que participan en el problema de decisión [25]. En nuestro modelo, nos centraremos en valores en el intervalo  $[0, 1]$  como dominio de representación y en las relaciones de preferencia como formato de representación. Esto implica que, para modelar las evaluaciones expresadas por los integrantes del grupo, se usarán relaciones de preferencia difusas [39]. La elección de las relaciones de preferencia difusas para modelar las evaluaciones se debe a que son las más utilizadas en procesos de toma de decisiones en grupo debido, entre otras razones, a su utilidad cuando se agregan las evaluaciones individuales de los expertos para obtener la evaluación colectiva [24].

**Definición 2.1** *La relación de preferencia difusa  $P^{zl}$  proporcionada por el experto  $e_z$  sobre el conjunto de alternativas  $A = \{a_1, \dots, a_n\}$  para el criterio  $c_l$  se caracteriza mediante la función  $\mu_{P^{zl}} : A \times A \rightarrow [0, 1]$ .*

La relación de preferencia difusa  $P^{zl}$  se suele simbolizar, para hacerla comprensible, a través de la matriz  $P^{zl} = (p_{jk}^{zl})$ , siendo  $n \times n$  la dimensión de la matriz:

$$P^{zl} = \begin{bmatrix} p_{11}^{zl} & \cdots & p_{1n}^{zl} \\ \vdots & \ddots & \vdots \\ p_{n1}^{zl} & \cdots & p_{nn}^{zl} \end{bmatrix}$$

Aquí,  $p_{jk}^{zl} = \mu_{P^{zl}}(a_j, a_k)$  indica el grado de preferencia de la alternativa  $a_j$  sobre la alternativa  $a_k$  ( $j \neq k$ ) según el experto  $e_z$  para el criterio  $c_l$ . En particular, si el experto  $e_z$  prefiere a la alternativa  $a_j$  sobre la alternativa  $a_k$  para el criterio  $c_l$ , se le asigna a  $p_{jk}^{zl}$  un valor igual o superior a 0.5; si el experto  $e_z$  prefiere igualmente a las alternativas  $a_j$  y  $a_k$  para el criterio  $c_l$ , se le asigna a  $p_{jk}^{zl}$  un valor igual a 0.5; y si el experto  $e_z$  prefiere a la alternativa  $a_k$  sobre la alternativa  $a_j$  para el criterio  $c_l$ , se le asigna a  $p_{jk}^{zl}$  un valor inferior a 0.5. Además, generalmente no se definen los elementos que forman parte de la diagonal principal. La comparación de una alternativa consigo misma no es lógica. Esto se suele simbolizar con un guion, es decir, usando el símbolo «—». Con todo, llegado el caso, la comparación de una alternativa consigo misma no debería tener ninguna influencia en el resultado obtenido. En este caso, esto implica que se tendría que asignar un valor igual a 0.5.

En resumen, el punto de partida de nuestro modelo basado en el paradigma de la computación granular es una colección de  $m \times q$  relaciones de preferencia difusas,  $P^{zl}$ ,  $z = 1, \dots, m$ ,  $l = 1, \dots, q$ , de dimensión  $n \times n$ .

### 2.3.2. Mejora de la coherencia y el consenso con mínimo ajuste

Antes de aplicar el proceso de selección para obtener la solución al problema, es importante que tanto la coherencia individual de cada experto como el consenso alcanzado entre estos sean tan altos como sea posible. La



mejora de la coherencia y del consenso usualmente requiere que los expertos modifiquen sus evaluaciones iniciales. Esto implica que, en cierta medida, deben estar dispuestos a aceptar un ajuste o modificación en sus evaluaciones iniciales. Sin embargo, si la modificación sugerida dista mucho de la inicialmente proporcionada por el experto, este podría rechazarla. Por ejemplo, supongamos que, usando relaciones de preferencia difusas para modelar las evaluaciones, un integrante del grupo expresa que su grado de preferencia de una cierta alternativa sobre otra es 0.7. Entonces, si la preferencia se ajusta a un valor igual a 0.8, este experto podría aceptarla, pero si la preferencia se ajusta a un valor igual a 0.3, el cual es bastante diferente del proporcionado por el experto, este podría rechazarla. Por tanto, es importante minimizar el ajuste, es decir, reducir la disimilitud entre la preferencia original proporcionada por el experto y la ajustada por el modelo [59].

La flexibilidad que admiten los integrantes del grupo en sus evaluaciones se puede modelar mediante el concepto de granularidad de la información, el cual transforma los elementos de las relaciones de preferencia difusas, los cuales se componen de valores numéricos precisos en el intervalo  $[0, 1]$ , en gránulos de información de mayor nivel de abstracción [33], dando lugar a relaciones de preferencia difusas granulares [11].

Para destacar que estamos considerando relaciones de preferencia difusas granulares, la relación de preferencia difusa  $P$  se denota como  $\mathbf{G}(P)$ . Además, como el formalismo de granulación que se va a utilizar es el de los intervalos (el cual ha sido usado en estudios similares [11, 42]), esto significa que  $\mathbf{G}(P) = \mathbf{INT}(P)$ , siendo  $\mathbf{INT}(\cdot)$  una familia de intervalos. Es decir, los elementos de las relaciones de preferencia difusas se interpretan ahora como intervalos, en lugar de como valores numéricos precisos, dando lugar a relaciones de preferencia difusas intervalares. En concreto, el nivel de granularidad de la información,  $\alpha$ , determina la longitud de los intervalos y se utiliza para mejorar tanto la coherencia como el consenso. Vemos así que la granularidad de la información juega un papel fundamental como sinónimo de flexibilidad inyectado en los elementos de las relaciones de preferencia difusas que se puede usar para optimizar una función determinada.

En resumen, al modificar las evaluaciones dadas por los expertos dentro de los límites establecidos por los intervalos formados según el nivel de granularidad de la información  $\alpha$ , pretendemos maximizar (optimizar) tanto la coherencia individual como el consenso, a la vez que pretendemos obtener unas evaluaciones modificadas que sean lo más parecidas posibles a las proporcionadas inicialmente por los expertos. Para realizar esta tarea de optimización se utilizará el algoritmo de optimización por enjambre de partículas [29] (PSO, por sus siglas en inglés). Aunque se podría haber elegido cualquier otra técnica de optimización (algoritmos genéticos, evolución diferencial, etc.), se ha escogido el algoritmo PSO porque tiene pocos parámetros, su convergencia es rápida y es fácil de implementar [50]. Además, ya ha sido usado exitosamente en problemas similares [8, 11, 41].

A continuación, se describen en detalle los aspectos más importantes de esta técnica de optimización: el algoritmo concreto utilizado, la representación de la partícula y la función de aptitud o función objetivo (también conocida como función *fitness*).

### Algoritmo

El objetivo del algoritmo PSO, creado por Eberhart y Kennedy [15, 29], es el de emular el comportamiento social. Para ello, intenta modelar el movimiento de una serie de organismos como los pertenecientes a una bandada de pájaros o a un banco de peces. Por ejemplo, pensemos que hay un alimento en una ubicación determinada y un banco de peces trata de encontrarlo. Los peces saben lo lejos que está el alimento en cada movimiento, pero no son conscientes de su ubicación. Para encontrar el alimento, el enfoque más prometedor es desplazarse siguiendo al pez que está más próximo al alimento.

Teniendo en cuenta lo anterior, el algoritmo PSO es un método de computación que, en una serie de iteraciones, intenta optimizar un problema de optimización. En cada iteración, este método intenta encontrar las soluciones candidatas más prometedoras según una medida de calidad determinada (función de aptitud). Para ello, comienza considerando un enjambre (población) de partículas (soluciones candidatas en un espacio de búsqueda  $d$ -dimensional). Para optimizar el problema, las partículas se van desplazando en las diferentes direcciones del espacio de búsqueda según una serie de fórmulas matemáticas simples. En cada iteración, el desplazamiento de cada partícula a una nueva posición se guía según su mejor ubicación hasta el momento en el espacio de búsqueda, según la mejor ubicación alcanzada por cualesquiera del resto de partículas de la población y por una determinada velocidad. El objetivo es dirigir a las partículas hacia la mejor ubicación (solución) [29].

Aunque existen diversas variantes del algoritmo PSO [50], las cuales se podrían aplicar aquí, en nuestro modelo utilizaremos la versión genérica [29], cuyos pasos se describen en el algoritmo 1 (suponemos que el criterio de parada es que se alcance el número máximo de iteraciones, aunque se podrían haber escogido otros criterios como el hecho de haber encontrado una solución satisfactoria al problema de optimización). Formalmente se define como sigue. Sea  $d$  la dimensión del espacio de búsqueda y  $n$  el número de partículas que componen la población. Sean  $\mathbf{v}_i = (v_{i,1}, v_{i,2}, \dots, v_{i,d})$ ,  $\mathbf{x}_i = (x_{i,1}, x_{i,2}, \dots, x_{i,d})$  y  $\mathbf{xb}_i = (xb_{i,1}, xb_{i,2}, \dots, xb_{i,d})$ , tres vectores  $d$ -dimensionales que representan la velocidad, la posición actual y la mejor posición alcanzada hasta el momento de la partícula  $i$ ,  $i \in \{1, \dots, n\}$ , respectivamente. Sea  $f : \mathbb{R}^d \rightarrow \mathbb{R}$  la función de aptitud que queremos optimizar (imaginemos aquí que queremos maximizarla). Esta función recibe como argumento el vector posición  $\mathbf{x}_i$  de la partícula  $i$  y devuelve como resultado un número real, el cual determina la calidad de la partícula  $i$  para optimizar el

---

**Algoritmo 1** PSO

---

```

1: para cada partícula  $i = 1, \dots, n$  hacer
2:   Iniciar su posición de acuerdo con un vector aleatorio uniformemente
     distribuido:  $\mathbf{x}_i \sim U(l_{\inf}, l_{\sup})$ 
3:   Iniciar su mejor posición con su posición inicial:  $\mathbf{xb}_i \leftarrow \mathbf{x}_i$ 
4:   si ( $f(\mathbf{xb}_i) > f(\mathbf{xg})$ ) entonces
5:     Actualizar la mejor posición global:  $\mathbf{xg} \leftarrow \mathbf{xb}_i$ 
6:   fin si
7:   Iniciar su velocidad:  $\mathbf{v}_i \sim U(-|l_{\sup} - l_{\inf}|, |l_{\sup} - l_{\inf}|)$ 
8: fin para
9: para cada iteración  $t = 1, \dots, m$  hacer
10:  para cada partícula  $i = 1, \dots, n$  hacer
11:    para cada dimensión  $h = 1, \dots, d$  hacer
12:      Generar dos números aleatorios:  $r_h, s_h \sim U(0, 1)$ 
13:      Actualizar su velocidad:  $v_{i,h} \leftarrow \omega(t) \cdot v_{i,h} + c_1 \cdot r_h \cdot (xb_{i,h} - x_{i,h}) +$ 
         $c_2 \cdot s_h \cdot (xg_h - x_{i,h})$ 
14:    fin para
15:    Actualizar su posición:  $\mathbf{x}_i \leftarrow \mathbf{x}_i + \mathbf{v}_i$ 
16:    si ( $f(\mathbf{x}_i) > f(\mathbf{xb}_i)$ ) entonces
17:      Actualizar su mejor posición:  $\mathbf{xb}_i \leftarrow \mathbf{x}_i$ 
18:    si ( $f(\mathbf{xb}_i) > f(\mathbf{xg})$ ) entonces
19:      Actualizar la mejor posición global:  $\mathbf{xg} \leftarrow \mathbf{xb}_i$ 
20:    fin si
21:  fin para
22: fin para
23: fin para
24: devolver  $\mathbf{xg}$ 

```

---

problema. La finalidad es identificar la partícula  $i$  que cumple la condición  $f(\mathbf{x}_i) \geq f(\mathbf{x}_j), \forall j \in \{1, \dots, n\}$  y  $i \neq j$ . Sea  $\mathbf{xg} = (xg_1, xg_2, \dots, xg_d)$  el vector que caracteriza la mejor posición global alcanzada por cualquier partícula. Sean  $l_{\inf}$  y  $l_{\sup}$ , respectivamente, los límites inferior y superior del espacio de búsqueda. Sean  $c_1$  y  $c_2$  dos coeficientes de aceleración que influyen el desplazamiento de la partícula en la dirección de su mejor posición y en la dirección de la mejor posición global, respectivamente (sus valores suelen pertenecer al intervalo  $[0, 2]$ , aunque el valor que se recomienda que tengan es 2 [43]). Sean  $\mathbf{r} = (r_1, r_2, \dots, r_d)$  y  $\mathbf{s} = (s_1, s_2, \dots, s_d)$  son dos vectores de valores de dos secuencias aleatorias en  $[0, 1]$ , los cuales dan un comportamiento aleatorio al desplazamiento de las partículas (esto mejora la capacidad de las partículas para huir de mínimos locales). Sea  $\omega$  la componente de inercia que se presentó en [46] para regular la velocidad de las partículas, teniendo una cierta influencia en el rendimiento del algoritmo. Por un lado, la asignación de valores grandes a  $\omega$  implica exploración (búsqueda global).

Por otro, la asignación de valores pequeños implica explotación (búsqueda local). Debido a esto, el valor asignado a  $\omega$  suele decrementarse con el paso de las iteraciones. Esto asegura una búsqueda global al inicio del algoritmo y una búsqueda local al final. Su valor suele modificarse de acuerdo con la siguiente expresión:

$$\omega(t) = (\omega(1) - \omega(m)) \cdot \frac{m - t}{m} + \omega(m), \quad (2.1)$$

siendo  $\omega(1)$ ,  $\omega(t)$ , y  $\omega(m)$ , los valores inicial, de la iteración actual y final asignados a  $\omega$ ;  $m$  y  $t$  representan el máximo número de iteraciones y la iteración actual, respectivamente.

### Partícula

De acuerdo con los rasgos distintivos de los procesos de toma de decisiones en grupo en entornos multicriterio y heterogéneos que estamos modelando, cada partícula se representa por un vector cuyos componentes pertenecen al intervalo  $[0, 1]$ , ya que estamos utilizando relaciones de preferencia difusas para modelar las evaluaciones de los integrantes del grupo. En particular, si el problema de decisión que se plantea se compone de  $q$  criterios,  $m$  expertos y  $n$  alternativas, cada partícula consiste en un vector de  $q \cdot (n - 1) \cdot n \cdot m$  dimensiones.

Asumiendo que  $p_{jk}^{zl} \in [0, 1]$  es un elemento de la relación de preferencia difusa  $P^{zl}$  expresada por el experto  $e_z$  para el criterio  $c_l$ , si  $\alpha$  es el nivel de granularidad de la información admitido, el elemento,  $p_{jk}^{zl}$ , en  $\text{INT}(P_{jk}^{zl})$  puede tomar valores dentro del intervalo  $[L_{jk}, U_{jk}]$ , cuyos límites se calculan de la siguiente manera:

$$[L_{jk}, U_{jk}] = [\max(0, p_{jk}^{zl} - \frac{\alpha}{2}), \min(p_{jk}^{zl} + \frac{\alpha}{2}, 1)] \quad (2.2)$$

Como ejemplo aclaratorio, supongamos que  $p_{jk}^{zl}$  y  $\alpha$  valen 0.6 y 0.3, respectivamente. Sea  $x_{i,h} = 0.4$  la componente correspondiente de la partícula  $i$ . Aplicando (2.2), el intervalo correspondiente a  $p_{jk}^{zl}$  es  $[0.45, 0.75]$ . Finalmente, aplicando (2.3), obtenemos que el valor ajustado  $\bar{p}_{jk}^{zl}$  de  $p_{jk}^{zl}$  es igual a 0.57.

$$\bar{p}_{jk}^{zl} = L_{jk} + (U_{jk} - L_{jk}) \cdot x_{i,h} \quad (2.3)$$

### Función de aptitud

La finalidad de las partículas es maximizar el valor devuelto por la función de aptitud en sus posiciones dentro del espacio de búsqueda. En nuestro caso, esto implica que, al modificar los valores iniciales de los elementos de las relaciones de preferencia difusas proporcionadas por los integrantes del grupo dentro de los límites de los intervalos obtenidos según el nivel

de granularidad de la información  $\alpha$ , en primer lugar, se pretende maximizar la coherencia individual de los expertos, en segundo lugar, se pretende maximizar el consenso alcanzado entre ellos y, en tercer lugar, se pretende maximizar la similitud entre los valores modificados de las relaciones de preferencia difusas y los inicialmente proporcionados por los expertos. Por tanto, la función de aptitud  $f$  se define así:

$$f = \beta \cdot (\gamma \cdot f_1 + (1 - \gamma) \cdot f_2) + (1 - \beta) \cdot f_3 \quad (2.4)$$

siendo  $\gamma \in [0, 1]$  un parámetro que establece un equilibrio entre la coherencia,  $f_1$ , y el consenso,  $f_2$ , y  $\beta \in [0, 1]$  un parámetro que establece un equilibrio entre esos dos valores,  $f_1$  y  $f_2$ , y  $f_3$ , el cual mide la similitud entre las relaciones de preferencia difusas proporcionadas por los integrantes del grupo,  $P^{zl}$ , y las relaciones de preferencia difusas modificadas,  $\bar{P}^{zl}$ ,  $z = 1, \dots, m$ ,  $l = 1, \dots, q$ .

Según (2.4), se necesitan calcular tres valores,  $f_1$ ,  $f_2$  y  $f_3$ , para obtener el valor de la función de aptitud  $f$ . En primer lugar, para obtener el valor de  $f_1$ , se calcula la media de la coherencia individual de los expertos:

$$f_1 = \frac{1}{m} \sum_{z=1}^m cl_z \quad (2.5)$$

donde  $cl_z$  simboliza el nivel de coherencia global del experto  $e_z$ . Este valor se calcula como la media ponderada de los niveles de coherencia de ese experto en cada criterio  $c_l$ :

$$cl_z = \frac{1}{\sum_{l=1}^q v_{zl}} \sum_{l=1}^q v_{zl} \cdot cl_{zl} \quad (2.6)$$

donde  $cl_{zl}$  simboliza el nivel de coherencia del experto  $e_z$  en el criterio  $c_l$ . Este valor se calcula mediante el procedimiento desarrollado en [24]. Remítase a esa referencia bibliográfica para una descripción detallada del procedimiento.

En segundo lugar, para calcular  $f_2$ , se utiliza un nuevo enfoque basado en el concepto de coincidencia [22], el cual se desarrolla como se describe a continuación:

- Cálculo de una matriz de similitud,  $SM = (sm_{jk}^{zyl})$ , para cada par de expertos,  $e_z$  y  $e_y$ , y cada criterio,  $c_l$ . Esta matriz determina la similitud entre las evaluaciones comunicadas por ambos expertos sobre ese criterio particular:

$$sm_{jk}^{zyl} = 1 - |p_{jk}^{zl} - p_{jk}^{yl}| \quad (2.7)$$

- Cálculo de una matriz de consenso,  $CM^l = (cm_{jk}^l)$ , para cada criterio,  $c_l$ . Esta matriz determina el consenso alcanzado por el grupo de expertos en ese criterio particular:

$$cm_{jk}^l = \frac{2}{m \cdot (m - 1)} \sum_{z=1}^{m-1} \sum_{y=z+1}^m sm_{jk}^{zyl} \quad (2.8)$$

- Cálculo, para cada matriz de consenso  $CM^l$ , de una medida de consenso global,  $gc_l$ , asociada al criterio  $c_l$ :

$$gc_l = \frac{1}{n \cdot (n-1)} \sum_{j=1}^n \sum_{k=1; k \neq j}^n cm_{jk}^l \quad (2.9)$$

- Cálculo de  $f_2$  mediante la media ponderada de las medidas de consenso global asociadas a los distintos criterios:

$$f_2 = \sum_{l=1}^q w_l \cdot gc_l \quad (2.10)$$

En tercer lugar, para calcular  $f_3$ , se debe obtener la similitud entre las relaciones de preferencia difusas expresadas por los expertos y las modificadas por el algoritmo PSO. Esto se realiza como se describe a continuación:

- Cálculo de un índice de similitud,  $si_{zl}$ , relacionado con cada experto,  $e_z$ , en cada criterio,  $c_l$ . Este índice determina la similitud entre la relación de preferencia difusa comunicada por ese experto y la sugerida por el algoritmo PSO en ese criterio:

$$si_{zl} = \frac{1}{n \cdot (n-1)} \sum_{j=1}^n \sum_{\substack{k=1 \\ k \neq j}}^n 1 - |p_{jk}^{zl} - \bar{p}_{jk}^{zl}| \quad (2.11)$$

- Cálculo de un índice de similitud,  $si_l$ , para cada criterio,  $c_l$ . Este índice determina la similitud entre las relaciones de preferencia difusas comunicadas por el grupo de expertos y las sugeridas por el algoritmo PSO en ese criterio:

$$si_l = \frac{1}{m} \sum_{z=1}^m si_{zl} \quad (2.12)$$

- Cálculo de  $f_3$  mediante la media ponderada de los índices de similitud asociados a los criterios:

$$f_3 = \sum_{l=1}^q w_l \cdot si_l \quad (2.13)$$

### 2.3.3. Proceso de selección

La tercera y última etapa del modelo tiene por finalidad devolver la mejor alternativa (o alternativas) a través de un proceso de agregación y de un proceso de explotación, los cuales se describen a continuación.

### Agregación

Mediante el proceso de agregación se obtiene una evaluación colectiva al fusionar las evaluaciones individuales de los integrantes del grupo. Como se están utilizando relaciones de preferencia difusas para modelar las evaluaciones, esto significa que, a partir de la agregación de las relaciones de preferencia difusas individuales ajustadas por el algoritmo PSO,  $\bar{P}^{zl}$ ,  $z = 1, \dots, m$ ,  $l = 1, \dots, q$ , se obtendrá una relación de preferencia difusa colectiva,  $\bar{PC}$ , que integra las evaluaciones de todos los expertos. Esta agregación se realiza utilizando una función de agregación [4]. En nuestro modelo, estamos considerando que tanto los criterios como los expertos tienen pesos de importancia. Por tanto, se puede utilizar un operador de agregación de la clase IOWA [54] (Induced Ordered Weighted Averaging, en inglés).

Los operadores de agregación de la clase IOWA toman una serie de argumentos formados por pares de elementos, en los que los primeros se usan para constituir el orden en el cual se agregarán los segundos.

**Definición 2.2** [54] *La función  $\Phi : \mathbb{R}^n \times \mathbb{R}^n \rightarrow \mathbb{R}$  es un operador IOWA de dimensión  $n$  si, asociado a  $\Phi$ , existe un vector de pesos,  $w = (w_1, w_2, \dots, w_n)$ , tal que  $w_i \in [0, 1]$ ,  $\sum_{i=1}^n w_i = 1$ , y donde:*

$$\Phi(\langle u_1, a_1 \rangle, \langle u_1, a_2 \rangle, \dots, \langle u_n, a_n \rangle) = \sum_{j=1}^n w_j \cdot b_j \quad (2.14)$$

siendo  $b_j$  el componente  $a_i$  del par  $\langle u_i, a_i \rangle$  que tiene el  $j$ -ésimo mayor valor de  $u_i$ . Debido a la función que realizan los componentes de los pares,  $a_i$  se conoce como la variable argumento y  $u_i$  como la variable inductora del orden.

Según la definición de un operador de agregación de la clase IOWA, vemos que es necesario disponer de un procedimiento adecuado para calcular el vector de pesos  $w$  [52, 53]. Un procedimiento para obtenerlo es mediante la utilización de los cuantificadores lingüísticos [56], los cuales utilizó Yager para calcular el vector de pesos de la siguiente forma:

$$w_i = Q\left(\frac{i}{n}\right) - Q\left(\frac{i-1}{n}\right), \quad i = 1, \dots, n \quad (2.15)$$

donde  $Q$  simboliza el cuantificador lingüístico modelado [52]. Si el vector de pesos del operador de agregación IOWA se determina mediante este procedimiento, esto se simboliza mediante  $\Phi_Q$ .

Una vez introducidos los operadores de agregación de la clase IOWA, el procedimiento para obtener la relación de preferencia difusa colectiva es el siguiente:

- Primero, se obtiene una relación de preferencia difusa colectiva,  $\overline{PC}^l = (\overline{pc}_{jk}^l)$ , para cada criterio  $c^l$  utilizando para ello un operador de agregación de clase IOWA donde el peso de importancia de cada experto para cada criterio actúa como variable inductora del orden:

$$\overline{pc}_{jk}^l = \Phi_Q(\langle v_{1l}, \overline{p}_{jk}^{1l} \rangle, \langle v_{2l}, \overline{p}_{jk}^{2l} \rangle, \dots, \langle v_{ml}, \overline{p}_{jk}^{ml} \rangle) \quad (2.16)$$

- Segundo, utilizando la información contenida en las relaciones de preferencia difusas colectivas de los criterios, la relación de preferencia difusa colectiva,  $\overline{PC} = (\overline{pc}_{jk})$ , se calcula también mediante el uso de un operador de agregación de clase IOWA donde el peso de importancia de cada criterio actúa como variable inductora del orden:

$$\overline{pc}_{jk} = \Phi_Q(\langle w_1, \overline{pc}_{jk}^1 \rangle, \langle w_2, \overline{pc}_{jk}^2 \rangle, \dots, \langle w_q, \overline{pc}_{jk}^q \rangle) \quad (2.17)$$

### Explotación

Mediante el proceso de explotación se selecciona la mejor alternativa (o alternativas) a través de la explotación de la información incluida en la relación de preferencia difusa colectiva. Para llevar a cabo este proceso, se pueden aplicar diferentes funciones [24, 44]. Entre ellas, haremos uso de un grado de selección de alternativas conocido, el grado de no dominancia guiado por cuantificador, el cual se basa en un operador de agregación de la clase OWA [52] (Ordered Weighted Averaging, en inglés).

**Definición 2.3** [52] *La función  $\phi : \mathbb{R}^n \rightarrow \mathbb{R}$  es un operador OWA de dimensión  $n$  si, asociado a  $\phi$ , existe un vector de pesos,  $w = (w_1, w_2, \dots, w_n)$  tal que  $w_i \in [0, 1]$ ,  $\sum_{i=1}^n w_i = 1$ , y donde:*

$$\phi(a_1, a_2, \dots, a_n) = \sum_{j=1}^n w_j \cdot b_j$$

*siendo  $b_j$  el  $j$ -ésimo mayor elemento en la colección  $a_1, a_2, \dots, a_n$ .*

Los operadores de agregación de clase OWA determinan una familia paramétrica de operadores de agregación, la cual contiene la mayoría de funciones de agregación conocidas que se generan escogiendo los pesos adecuados [51] (por ejemplo, el máximo, la media, el mínimo y la mediana). Además, para obtener el vector de pesos, se puede seguir el mismo procedimiento que el seguido para el caso de los operadores de agregación de clase IOWA. En tal caso, se representa mediante  $\phi_Q$ .

Una vez introducidos los operadores de agregación de la clase OWA, el procedimiento para obtener el grado de no dominancia guiado por cuantificador es el siguiente:



- El grado de no dominancia guiado por cuantificador,  $QGNDD_j$ , determina el grado en que la alternativa  $a_j$  no está dominada por las restantes alternativas. Se calcula como:

$$QGNDD_j = \phi_Q(1 - d_{1j}, 1 - d_{2j}, \dots, 1 - d_{(j-1)j}, \\ 1 - d_{(j+1)j}, \dots, 1 - d_{nj}) \quad (2.18)$$

siendo  $d_{kj} = \max\{\overline{pc}_{kj} - \overline{pc}_{jk}, 0\}$ , que determina el grado en que la alternativa  $a_j$  es dominada por la alternativa  $a_k$ .



## Capítulo 3

# Desarrollo del sistema web de ayuda a la decisión en grupo

A lo largo de este capítulo se explica de forma detallada todo el proceso que se ha seguido para desarrollar el sistema web de ayuda a la decisión en grupo basado en el modelo descrito en el capítulo anterior. En concreto, se detallan la planificación, metodología, costes asociados, análisis, diseño, implementación y pruebas.

### 3.1. Planificación, metodologías y costes asociados

#### 3.1.1. Planificación

Para realizar este trabajo de fin de grado, se ha llevado a cabo una planificación de desarrollo, la cuál empezó a principios de febrero de 2021 y finalizó a principios de julio de 2021 (5 meses).

Para la planificación, se ha tenido en cuenta que habrá un par de reuniones con el cliente (director del trabajo fin de grado) cada mes, aunque esto se intensificará conforme se acerque el plazo de finalización de este. Las tareas en las que se estructura el desarrollo del trabajo fin de grado y el tiempo que se dedica a cada una de ellas se visualiza en un diagrama de Gantt (ver figura 3.1), herramienta gráfica que nos permite planificar el trabajo de forma rápida y visual. También permite visualizar las tareas que tenemos que llevar a cabo y realizar su seguimiento más eficientemente.

Teniendo en cuenta el diagrama de Gantt de la figura 3.1, la planificación descrita con más detalle es la siguiente:

- Base Matemática. En este punto, entra todo lo que está asociado al cálculo matemático: consistencia, consenso, operadores de agregación

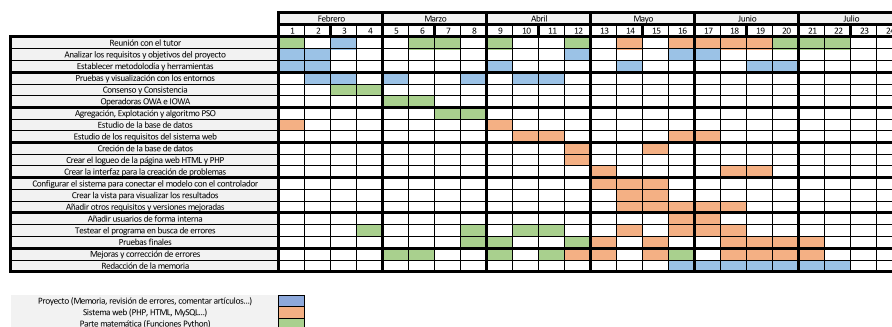


Figura 3.1: Diagrama de Gantt del trabajo fin de grado.

de la clase OWA y de la clase IOWA, agregación, explotación, algoritmo PSO, etc. La planificación de esta parte es la siguiente:

- Semanas 1 y 2. Se realizarán varias reuniones con el director, en las que se tratarán los temas sobre el diseño, herramientas que utilizar, modo de trabajo y los artículos en los que nos basaremos.
- Semanas 3 y 4. Se desarrollará el cálculo de la consistencia y del consenso. Se podrá realizar de forma global para todos los expertos o de forma individual. De esta forma, será más fácil la detección de errores.
- Semanas 5 a 9. Se desarrollarán los operadores de agregación de la clase OWA y de la clase IOWA y comenzaremos con los métodos de explotación y agregación. Con la mayoría de la parte matemática casi terminada, se empezará con la implementación del patrón de diseño Strategy. Se añade el cálculo con los pesos para los criterios y expertos, y se comienza con la implementación del algoritmo PSO. Durante el desarrollo de esta parte, hemos tenido que realizar varias reuniones, porque los resultados no eran los esperados, sin embargo, y tras varias horas de prueba y error, nos dimos cuenta de que se estaba realizando mal la copia de los elementos que intervenían en el proceso porque no se le realizaba en profundidad. Una vez cambiada y chequeada esta parte, el algoritmo quedó implementado.
- Desarrollo del sistema. Una vez finalizada la parte matemática del modelo, comenzaremos con el diseño e implementación del sistema y con la mejora de algunas funciones junto a la corrección de errores.

- Semanas 10 y 11. Comenzaremos con el diseño del sistema, la base de datos y corrección de errores en pruebas externas.
  - Semana 12. Junto a un proyecto de licencia gratis y abierta, implementaremos el panel de control junto a la creación de problemas por parte del administrador. Será necesario familiarizarnos con las clases, el entorno y las sentencias necesarias.
  - Semanas 13 a 18. Implementación de diferentes métodos que permitan añadir usuarios, visualizar las soluciones y juntar nuestro sistema en PHP con las funciones escritas en Python. Para ello, necesitamos realizar los cálculos, los cuales pueden conllevar hasta 5 minutos de espera, de forma transparente a los usuarios. Una vez que esté la solución a cualquier problema, se le notificará mediante un correo electrónico a los usuarios correspondientes.
  - Semanas 18 a 21. Implementación de una mejora propuesta por el director, junto a diferentes arreglos al aspecto y diseño y funcionalidad de algunas de las partes.
- Desarrollo de la memoria y arreglos finales. Con aproximadamente cinco semanas de margen, se comenzará con el desarrollo de la memoria, junto a la revisión de fallos y aspectos finales para ser entregada. Su fecha máxima de finalización será el 9 de julio.

### **3.1.2. Metodologías**

En esta sección, describimos las metodologías de trabajo que se han llevado a cabo durante el desarrollo del proyecto.

### **3.1.3. Metodología Kanban**

Conforme se avanza en el desarrollo de software es normal que aparezcan nuevos requisitos e incluso que algunos de ellos que ya estaban aceptados acaben siendo desechados. Por ello, el uso de metodologías ágiles es conveniente, debido a que permiten dar respuestas rápidas a los clientes y realizar cambios sin que supongan horas en exceso o grandes complicaciones. También permiten verificar de forma más rápida los errores y cambiarlos de forma instantánea. En general, el uso de metodologías ágiles lo podemos resumir en las siguientes ventajas:

- Incremento en la satisfacción del cliente. Al involucrar desde el inicio del proyecto al cliente, permitiéndole participar en las revisiones del producto que se va desarrollando, este se sentirá más satisfecho.
- Ahorro en coste y tiempo. La realización del proyecto mediante entregas periódicas de nuevas versiones del producto, las cuales se someten a revisión, hace que se obtenga una versión cada vez más funcional que

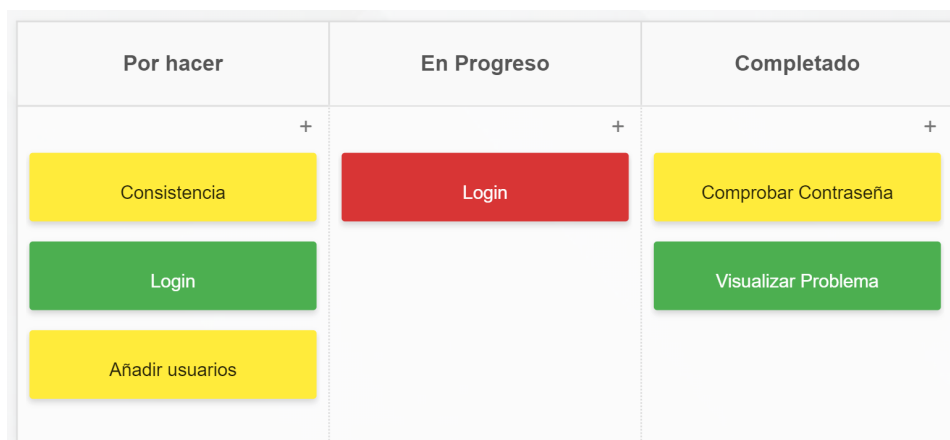


Figura 3.2: Ejemplo de diagrama Kanban.

permite que, en el caso de encontrar problemas, sean más sencillos de resolver.

- Eliminación de requisitos no deseados. Esas mismas entregas periódicas de nuevas versiones del producto ayudan a que las características no necesarias del producto se eliminen.

De entre las principales metodologías ágiles existentes (SCRUM, eXtreme Programming, etc.), nosotros haremos uso de la metodología Kanban [1]. Aunque esta metodología no la hemos estudiado en nuestra rama, es algo que hemos realizado durante toda nuestra vida. Consiste en clasificar diferentes tareas en tres columnas: «Por hacer», «En progreso» y «Completado», pero en vez de añadir la última columna, se tachaba en nuestro papel. Un ejemplo del diagrama se encuentra en la figura 3.2.

De acuerdo con la metodología Kanban tenemos que construir un diagrama con las tareas pendientes, las tareas en proceso y las tareas finalizadas. Por un lado, esto evita que se nos olvide realizar alguna tarea y, por otro, al llevar a cabo una división del proyecto en tareas pequeñas, es mucho más sencillo volver atrás para modificar algún requisito o para revisar alguna parte. Como resumen, la metodología Kanban nos da los siguientes beneficios:

- Mayor rendimiento. La organización visual del estado de desarrollo en el cual se encuentra el proyecto permite saber qué tareas debemos priorizar.
- Ahorro de tiempo. La organización de las tareas importantes nos evita distracciones innecesarias en tareas extras o secundarias.

- Flexibilidad. La división del proyecto en tareas pequeñas permite llevar a cabo cambios sin tener que realizar grandes modificaciones.

#### 3.1.4. Desarrollo de prototipos

Por otro lado, también haremos uso de una metodología basada en el desarrollo de prototipos iterativos [47], entendiendo al prototipo como una versión funcional de un sistema de información que forma parte de un modelo preliminar. Los distintos prototipos han sido las sucesivas entregas que se han ido realizando para ir haciendo el seguimiento y avance del sistema. De esta forma, se va recibiendo retroalimentación en cada entrega de forma que podemos comprobar si estamos cumpliendo las expectativas del cliente y si se están implementado las funcionalidades previstas y, en caso necesario, quitar o añadir funcionalidades (ver figura 3.3).

Hemos utilizado también esta metodología debido a que nos aporta, entre otros, los siguientes beneficios:

- En cada entrega del prototipo, obtención de requisitos y validación por parte del cliente. Esto provoca que los cambios tengan menor impacto, reduciendo costes y aumentando la fiabilidad.
- Al haber estado el cliente inmerso en el desarrollo del producto, facilita su aprobación.
- El producto final estará adaptado totalmente a los requerimientos del cliente, al haber estado este, como ya hemos comentado, proporcionando su retroalimentación durante todo el desarrollo del producto.

#### 3.1.5. Costes

En cuanto a los costes, se desglosará el total de precios, según la mano de obra y todos los dispositivos utilizados durante el desarrollo del sistema:

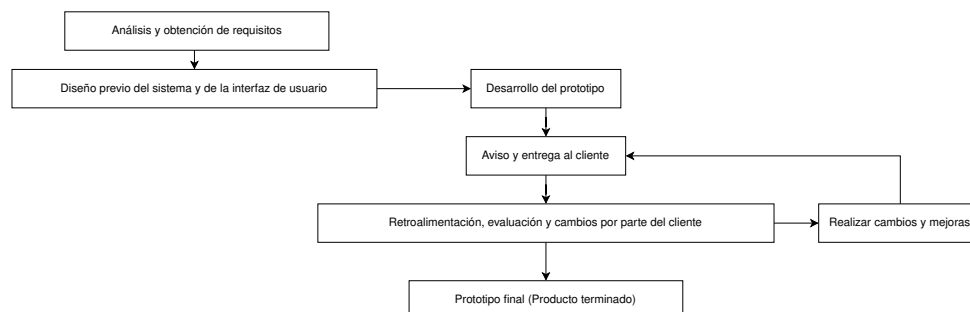


Figura 3.3: Proceso del desarrollo de prototipos.

| Material y recurso     | Coste total |
|------------------------|-------------|
| Salario                | 9864.35€    |
| Dispositivos           | 1674€       |
| Material               | 50€         |
| Gastos Extras          | 250€        |
| Licencias              | 0€          |
| Coste total: 11838.35€ |             |

Tabla 3.1: Tabla descriptiva de los costes del proyecto.

- El proyecto se ha realizado en cinco meses, y se le ha asignado un sueldo neto mensual de 1539.15€. Como son cinco meses, el saldo total sería 7695.75€. Esto supondría una cuantía de dinero en bruto de 9864.35€.
- Para la realización del sistema, hemos utilizado un ordenador de 985€ y una Microsoft Surface de 689€.
- Se han consultado diversos artículos sobre el tema, algunos proporcionados por el director y otros obtenidos en la web, y como se prefiere realizar anotaciones sobre ellos, se añadirá un presupuesto de 50€ en material educativo.
- Se ha obtenido un sistema de gestión de usuarios obtenido del GitHub<sup>1</sup> de unos desarrolladores que lo ofrecen de manera gratuita, para cualquier tipo de actividad. Por tanto, su coste será de 0€. Se ha escogido este sistema debido a que es un sistema de inicio de sesión muy completo, y no es parte de nuestro proyecto realizar algo similar que debido a que ya está desarrollado. Además, tiene licencia MIT<sup>2</sup> que tiene como características: compatible con DFSG (directrices de software libre de Debian), de software libre, aprobado por la OSI (Open Systems Interconnection), compatible con la GPL (General Public License), no posee Copyleft y es utilizable junto con otras licencias.
- Como se ha realizado un gasto de luz y electricidad, se añadirán 50€ por mes durante la realización del trabajo. Lo que hace una cantidad de 250€.

El desglose del proyecto y su coste total se reflejan en la tabla 3.1.

<sup>1</sup><https://github.com/therecluse26/PHP-Login>

<sup>2</sup><https://opensource.org/licenses/MIT>



## 3.2. Análisis

En esta sección, se detallan los requisitos no funcionales y funcionales del sistema, así como los requisitos de información y los casos de uso del sistema.

### 3.2.1. Requisitos no funcionales

En este punto, vamos a tratar los requisitos no funcionales más importantes de nuestro sistema:

- Seguro. Debe garantizar la protección de los datos alojados en la base de datos frente a ataques externos.
- Regulable. Debe cumplir la Ley Orgánica 3/2018, del 5 de diciembre, de Protección de Datos Personales y garantía de los derechos digitales.
- Eficiente y escalable. Tiene que dar respuestas rápidas también en caso de sobrecarga de uso. En caso de un alto nivel de espera, deberá ser totalmente transparente al cliente.
- Evolutivo. Debe usar patrones de diseño que permitan la introducción de mejoras y nuevas implementaciones de manera sencilla para los próximos desarrolladores, al mismo tiempo que nuevas acciones, distintos datos o nuevas métricas.
- Sobre la interfaz de usuario. Debe usar un patrón de diseño que permita la introducción de mejoras y nuevas implementaciones de manera sencilla para los próximos desarrolladores, al mismo tiempo que nuevas acciones, distintos datos o nuevas métricas.
  - Amigable. La GUI debe ser sencilla de usar e intuitiva.
  - Adaptable. Debe poder adaptarse a distintos dispositivos.

### 3.2.2. Requisitos funcionales

En este punto, vamos a tratar los requisitos funcionales más importantes de nuestro sistema. En nuestro caso, el sistema tendrá dos tipos de usuarios:

- Administrador. Será quien gestione el sistema, creando los problemas, verificando usuarios, modificando problemas, etc.
- Cliente (experto). Será un usuario estándar que opinará sobre los problemas que se le asocien y consultará los resultados.

Los requisitos funcionales (RF) asociados a los dos tipos de usuarios anteriores son los siguientes:

| Detalle            | Descripción  |
|--------------------|--|
| RF (Administrador) | 1  |
| Nombre             | Crear problema   |
| Descripción        | Permite crear un nuevo problema para que sea opinado por los clientes de cara a obtener sus resultados   |
| Entrada            | Todos los datos relacionados con los problemas (Número de expertos, numero de criterios, nombre, descripción, umbral, valores del problema, nombres de los factores que intervienen, etc.) |
| Procesamiento      | Comprobación previa de los datos y posterior inserción a la base de datos de forma externa al sistema  |
| Salida             | Mensaje emergente para indicar la correcta inserción   |

| Detalle            | Descripción  |
|--------------------|--|
| RF (Administrador) | 2  |
| Nombre             | Calcular problema  |
| Descripción        | Permite calcular un problema que haya recibido todas las opiniones. De forma interna, se realizarán los cálculos de la consistencia, consenso, algoritmo PSO |
| Entrada            | Nombre del problema que se va a calcular   |
| Procesamiento      | Consulta por el nombre a la base de datos y ejecución  |
| Salida             | Un correo electrónico a todos los participantes indicando que ya pueden visualizar la solución   |

| Detalle            | Descripción   |
|--------------------|---|
| RF (Administrador) | 3   |
| Nombre             | Modificar problema  |
| Descripción        | Permite cambiar los parámetros (umbral, $\alpha$ , $\beta$ y $\gamma$ ) para estudios individuales. |
| Entrada            | Nombre del problema que se va a modificar   |
| Procesamiento      | Consulta por el nombre a la base de datos y actualización de los resultados                         |
| Salida             | Mensaje emergente de una correcta modificación  |

| Detalle            | Descripción  |
|--------------------|--|
| RF (Administrador) | 4  |
| Nombre             | Borrar problema  |
| Descripción        | Permite eliminar un problema   |
| Entrada            | Nombre del problema que se va a eliminar                             |
| Procesamiento      | Consulta por el nombre a la base de datos y eliminación del problema |
| Salida             | Mensaje emergente de una correcta eliminación                        |

| Detalle            | Descripción   |
|--------------------|---|
| RF (Administrador) | 5   |
| Nombre             | Visualizar problema   |
| Descripción        | Permite ver los resultados de un problema                                   |
| Entrada            | Nombre del problema que se va a visualizar                                  |
| Procesamiento      | Consulta por el nombre a la base de datos y visualización de sus resultados |
| Salida             | Datos de la consulta realizada  |

| Detalle            | Descripción  |
|--------------------|--|
| RF (Administrador) | 6  |
| Nombre             | Añadir usuario   |
| Descripción        | Permite añadir un usuario al sistema   |
| Entrada            | Datos del usuario (Usuario, contraseña, correo electrónico). Posteriormente, el usuario deberá cambiar la contraseña |
| Procesamiento      | Inserción de los datos en la base de datos   |
| Salida             | Mensaje emergente de confirmación  |

| Detalle            | Descripción  |
|--------------------|--|
| RF (Administrador) | 7  |
| Nombre             | Verificar usuario  |
| Descripción        | Permite aceptar a los usuarios que se han registrado en el sistema   |
| Entrada            | Nombre del usuario a verificar   |
| Procesamiento      | El administrador tendrá una tabla donde se encuentran todos los usuarios pendientes de validación. Podrá aceptarlos o denegarlos |
| Salida             | Correo electrónico que indica que ya es usuario del sistema  |

| Detalle       | Descripción  |
|---------------|--|
| RF (Cliente)  | 8  |
| Nombre        | Identificarse  |
| Descripción   | Permite acceder al sistema   |
| Entrada       | Nombre de usuario y contraseña   |
| Procesamiento | Se hace una consulta a la base de datos para comprobar los datos introducidos y se inicia una nueva sesión |
| Salida        | Acceso a la página web   |

| Detalle       | Descripción  |
|---------------|--|
| RF (Cliente)  | 9  |
| Nombre        | Opinar   |
| Descripción   | Permite que los clientes realicen la opinión sobre su problema asociado y así poder resolverlo |
| Entrada       | Nombre del problema a resolver y su opinión de forma numérica                                  |
| Procesamiento | Comprobación de la validez de datos e inserción en la base de datos                            |
| Salida        | Mensaje emergente para indicar la correcta inserción   |

| Detalle       | Descripción   |
|---------------|---|
| RF (Cliente)  | 10  |
| Nombre        | Identificarse   |
| Descripción   | Permite acceder al sistema  |
| Entrada       | Nombre de usuario y contraseña  |
| Procesamiento | Se hace una consulta a la base de datos para comprobar los datos introducidos y se realiza una nueva sesión |
| Salida        | Acceso a la página web  |

| Detalle       | Descripción                                |
|---------------|--|
| RF (Cliente)  | 11   |
| Nombre        | Visualizar problema                        |
| Descripción   | Permite ver los resultados de un problema  |
| Entrada       | Nombre del problema que se va a visualizar |
| Procesamiento | Consulta por el nombre en la base de datos |
| Salida        | Datos de la consulta realizada             |

| Detalle       | Descripción  |
|---------------|--|
| RF (Cliente)  | 12   |
| Nombre        | Registrarse  |
| Descripción   | Permite añadirse al sistema  |
| Entrada       | Nombre de usuario, contraseña y correo electrónico   |
| Procesamiento | Tras introducir los datos, el usuario desde su correo o desde administración deberá esperar a ser verificado |
| Salida        | Correo electrónico para confirmar su validación  |

### 3.2.3. Requisitos de información

A continuación, detallaremos todos los atributos que intervienen en cada una de las acciones de nuestro sistema y el tipo de dato que tiene asignado cada uno de los atributos.

1. Para la creación de los problemas:

| Atributo                   | Tipo    |
|----------------------------|---------|
| Id                         | Entero  |
| Nombre                     | Varchar |
| Número de Expertos         | Entero  |
| Número de Criterios        | Entero  |
| Descripción                | Varchar |
| Opiniones                  | Varchar |
| Usuarios                   | Varchar |
| Número de Alternativas     | Entero  |
| Nombre de las Alternativas | Varchar |
| $\alpha$                   | Float   |
| $\beta$                    | Float   |
| $\gamma$                   | Float   |
| Pesos de los Criterios     | Varchar |
| Pesos de los Expertos      | Varchar |
| Umbral                     | Float   |
| Nombre de los Criterios    | Varchar |

2. Para visualizar los problemas

| Atributo | Tipo    |
|----------|---------|
| Nombre   | Varchar |

3. Para el registro o creación de usuarios:

| Atributo           | Tipo    |
|--------------------|---------|
| Nombre             | Varchar |
| Correo electrónico | Varchar |
| Contraseña         | Varchar |

4. Para opinar:

| Atributo | Tipo        |
|----------|-------------|
| Nombre   | Varchar     |
| Opinión  | Float [0,1] |

5. Para modificar un problema:

| Atributo    | Tipo    |
|-------------|---------|
| Nombre      | Varchar |
| Descripción | Varchar |
| $\alpha$    | Float   |
| $\beta$     | Float   |
| $\gamma$    | Float   |
| Umbral      | Float   |

#### 3.2.4. Casos de uso

A continuación, se expondrán los distintos casos de uso del sistema. De forma global, se podrán ver en la figura 3.4 todos los casos de uso y su correspondiente actor. Una explicación más detallada de cada caso de uso, con su nombre, actores, descripción, precondiciones y postcondiciones, se desarrolla acto seguido.

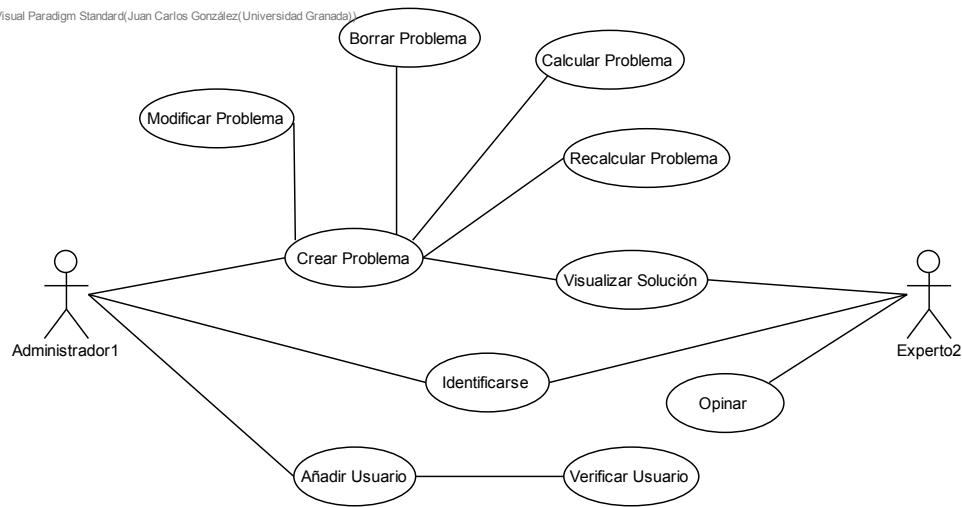


Figura 3.4: Diagrama de casos de uso del sistema.

| Detalle         | Descripción   |
|-----------------|---|
| Nombre          | Crear problema  |
| Actores         | Administrador   |
| Descripción     | El administrador añadirá los datos necesarios para la creación del problema. Los datos han sido detallados en la sección anterior.            |
| Precondiciones  | El problema no puede estar creado con anterioridad. Se necesitan conocer los expertos que participarán en el problema y todos los datos       |
| Postcondiciones | El sistema creará el problema y se lo asignará a cada experto, notificándolo con un correo, para que pueda proceder a proporcionar su opinión |

| Detalle         | Descripción  |
|-----------------|--|
| Nombre          | Calcular problema  |
| Actores         | Administrador  |
| Descripción     | El administrador procederá a dar solución a los problemas que hayan sido opinados  |
| Precondiciones  | Todos los expertos asociados al problema deberán haber proporcionado su opinión. Se le notificará con un correo cuando esto ocurra   |
| Postcondiciones | El sistema asociará la solución al problema y lo marcará como solucionado. Se podrá visualizar la solución y se notificará a los usuarios de ello mediante un correo electrónico |

| Detalle         | Descripción  |
|-----------------|--|
| Nombre          | Modificar problema   |
| Actores         | Administrador  |
| Descripción     | El administrador cambiará los valores internos de un problema creado, y previamente calculado, para obtener nuevas mediciones y estudios                             |
| Precondiciones  | Todos los expertos asociados al problema deberán haber realizado su opinión y haberse calculado  |
| Postcondiciones | El sistema asociará la solución al problema principal y lo añadirá como opción de visualización. Se notificará a los usuarios de ello mediante un correo electrónico |

| Detalle         | Descripción   |
|-----------------|---|
| Nombre          | Borrar problema   |
| Actores         | Administrador   |
| Descripción     | El administrador eliminará un problema del sistema  |
| Precondiciones  | El problema debe haber sido creado  |
| Postcondiciones | El sistema borrará el problema, junto a todos los registros asociados a él (problemas modificados, opiniones, soluciones, etc.) |

| Detalle         | Descripción   |
|-----------------|---|
| Nombre          | Recalcular problema   |
| Actores         | Administrador   |
| Descripción     | El administrador podrá calcular nuevamente un problema ya terminado, porque considere que ha habido un error en los cálculos o no se haya generado bien alguno de los datos |
| Precondiciones  | Todos los expertos asociados al problema deberán haber proporcionado su opinión al problema y haberse calculado   |
| Postcondiciones | El sistema asociará la solución al problema, sobrescribiendo los datos anteriores, y notificará a los usuarios asociados de su resolución                                   |



| Detalle         | Descripción  |
|-----------------|--|
| Nombre          | Visualizar solución  |
| Actores         | Administrador y expertos   |
| Descripción     | Se podrá consultar los resultados de los problemas que han sido calculados                   |
| Precondiciones  | El problema se ha calculado y, por tanto, finalizado el proceso                              |
| Postcondiciones | Se mostrarán los datos finales del problema, junto a una gráfica descriptiva de su evolución |

| Detalle         | Descripción   |
|-----------------|---|
| Nombre          | Opinar  |
| Actores         | Experto   |
| Descripción     | El experto dará su opinión sobre un problema en términos numéricos en el intervalo $[0, 1]$ .                                     |
| Precondiciones  | El problema tiene que estar creado y asociado al experto  |
| Postcondiciones | Se almacenará la opinión al problema y se pondrá en espera, hasta que se calcule y todos los demás expertos hayan dado su opinión |

| Detalle         | Descripción  |
|-----------------|--|
| Nombre          | Identificarse  |
| Actores         | Administrador y expertos   |
| Descripción     | Los usuarios podrán identificarse en el sistema para acceder a los datos |
| Precondiciones  | Deberán estar dados de alta en el sistema                                |
| Postcondiciones | No hay   |

| Detalle         | Descripción  |
|-----------------|--|
| Nombre          | Añadir usuario   |
| Actores         | Administrador  |
| Descripción     | El administrador añadirá un usuario al sistema                         |
| Precondiciones  | El usuario no podrá encontrarse ya registrado en el sistema            |
| Postcondiciones | El sistema creará un nuevo usuario al cuál se le notificará por correo |

| Detalle         | Descripción   |
|-----------------|---|
| Nombre          | Verificar usuario   |
| Actores         | Administrador   |
| Descripción     | El administrador aceptará a un usuario que se ha registrado de forma remota para que pueda acceder al sistema |
| Precondiciones  | El usuario deberá haberse registrado  |
| Postcondiciones | El sistema validará al usuario y se lo notificará por correo  |

| Detalle         | Descripción   |
|-----------------|---|
| Nombre          | Registrarse   |
| Actores         | Expertos  |
| Descripción     | Un usuario podrá registrarse en el sistema para poder convertirse en experto y poder opinar sobre problemas |
| Precondiciones  | El usuario no puede tener una cuenta anteriormente  |
| Postcondiciones | Se quedará a la espera de que el administrador le verifique la cuenta y recibir la notificación por correo  |

### 3.3. Diseño

En esta sección, se describen el diagrama arquitectónico del sistema, los diagramas de secuencia, el diagrama arquitectónico, el diagrama entidad-relación, el diagrama de clases, los patrones de diseño utilizados y los *mocups* de la interfaz de usuario.

#### 3.3.1. Diagrama arquitectónico del sistema

El diseño arquitectónico está compuesto por dos sistemas (ver figura 3.5):

- Uno para la gestión de los problemas, que es con el que interactúan los clientes.
- Otro para el cálculo de los problemas, que es llamado y utilizado por el sistema gestor.

Ambos sistemas se conectan a la base de datos para la inserción, modificación, actualización o consulta de datos.

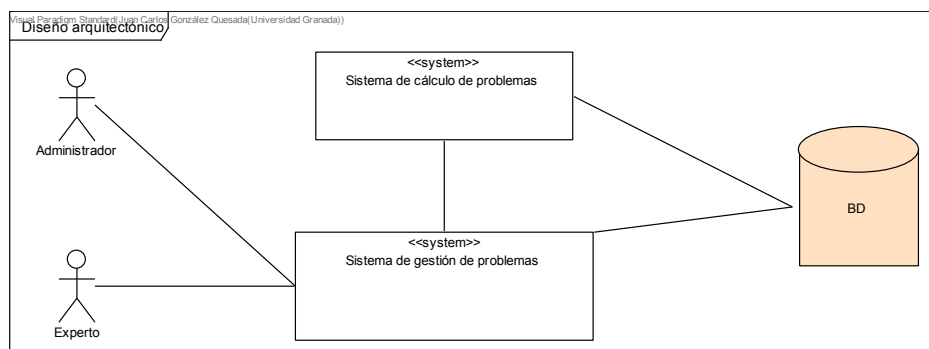


Figura 3.5: Diagrama arquitectónico del sistema.

#### 3.3.2. Diagramas de secuencia

Se van a desarrollar tres diagramas de secuencia, que corresponden con los más importantes del sistema:

- La secuencia que se lleva a cabo para crear un problema (ver figura 3.6).
- El proceso que se realiza cuando un experto proporciona su opinión en un problema (ver figura 3.7).
- El proceso para calcular un problema (ver figura 3.8).

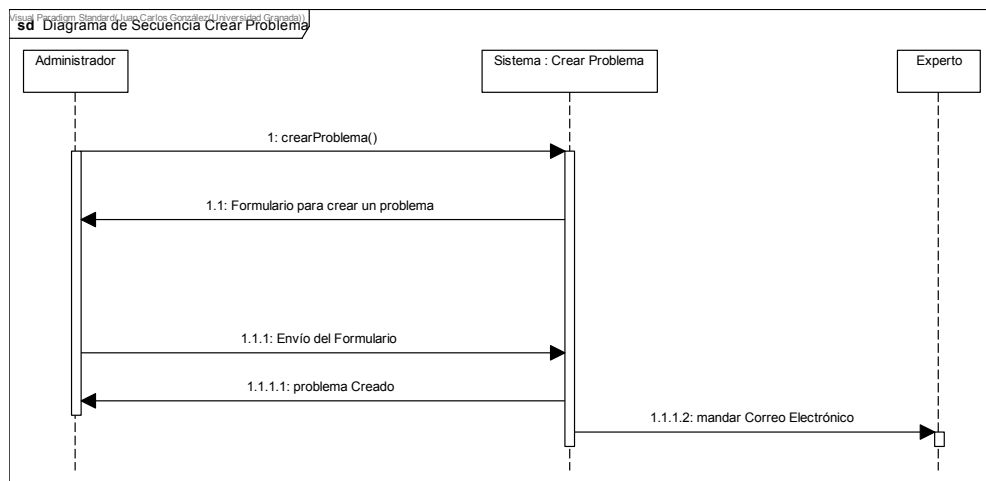


Figura 3.6: Diagrama de secuencia para la creación de un problema.

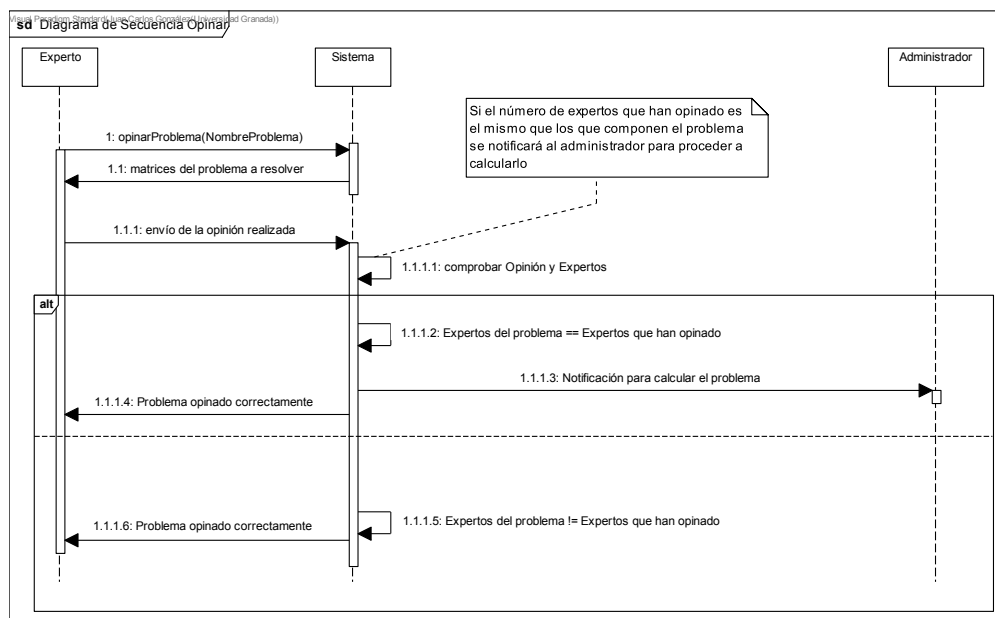


Figura 3.7: Diagrama de secuencia para opinar en un problema.

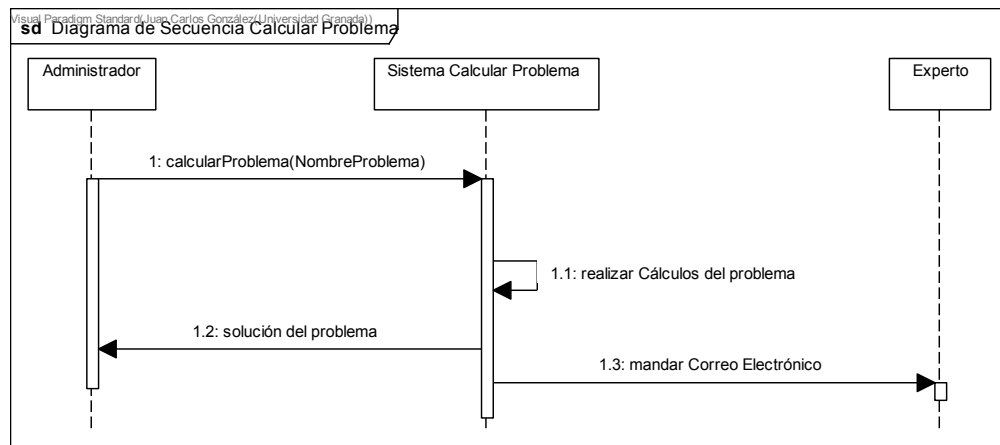


Figura 3.8: Diagrama de secuencia para el cálculo de un problema.

### 3.3.3. Diagrama Entidad-Relación

En la figura 3.9 podemos observar el diagrama entidad-relación de la base de datos. Se visualizan las tablas y sus relaciones.

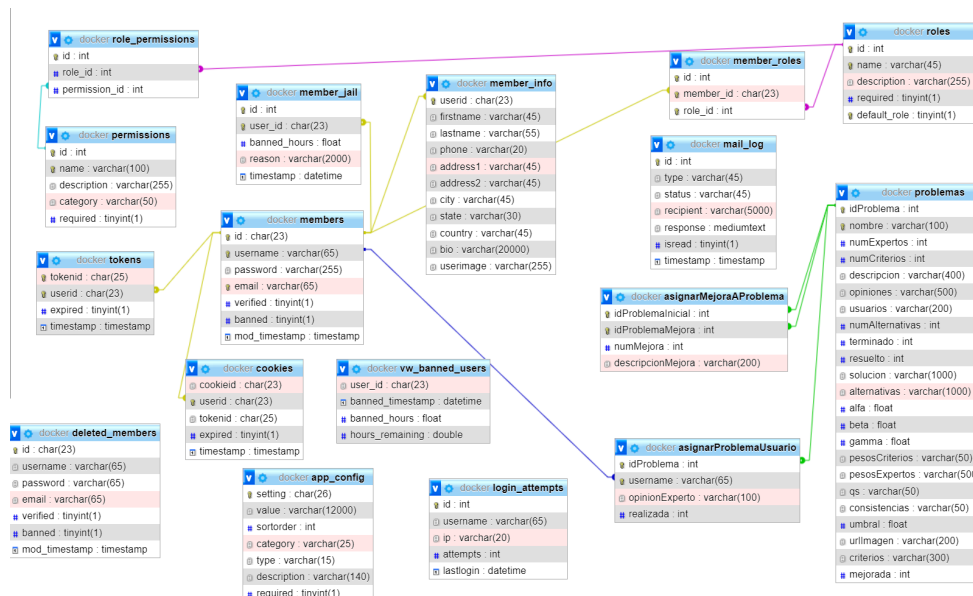


Figura 3.9: Diagrama de entidad-relación de la base de datos.

### 3.3.4. Diagrama de clases

El diagrama de clase del sistema se muestra en la figura 3.10. La clase principal es Problema, la cual tiene relación con todas las demás.

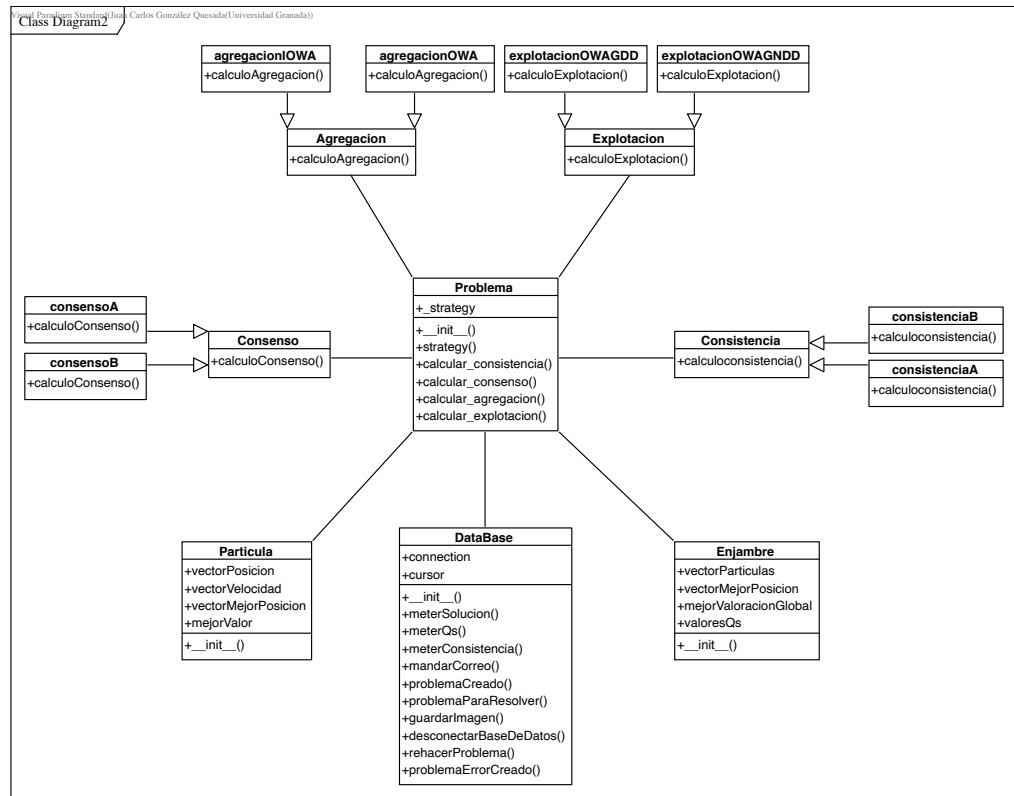


Figura 3.10: Diagrama de clases del sistema.

### 3.3.5. Patrones de diseño

Vamos a utilizar dos patrones de diseño, por un lado el patrón de diseño Strategy y, por otro, el patrón de diseño modelo-vista-controlador (MVC), los cuales describimos a continuación.

#### Strategy

Este patrón de diseño permite almacenar un conjunto de estrategias que el usuario podrá intercambiar dinámicamente según su necesidad. En nuestro caso, por ejemplo, la agregación tiene dos formas de ser calculada, con el operador de agregación de clase OWA o con el operador de agregación

de clase IOWA. De esta forma, el usuario puede elegir cual de ellos utilizar. A esto, se añade la facilidad de reutilizar el código de cara a futuras mejoras, puesto que si se quiere añadir otra forma de calcular la agregación (otra estrategia al sistema) solo hará falta insertarla, realizándose de forma intuitiva y sin necesidad de conocer en profundidad todo el código. No solo la agregación, sino que se pueden introducir de forma sencilla otros métodos (estrategias) de calcular el consenso, la coherencia, y la similitud. No obstante, a pesar de que este diseño pueda contener numerosas estrategias, solo cargará aquellas que vayan a ser utilizadas [17, 45].

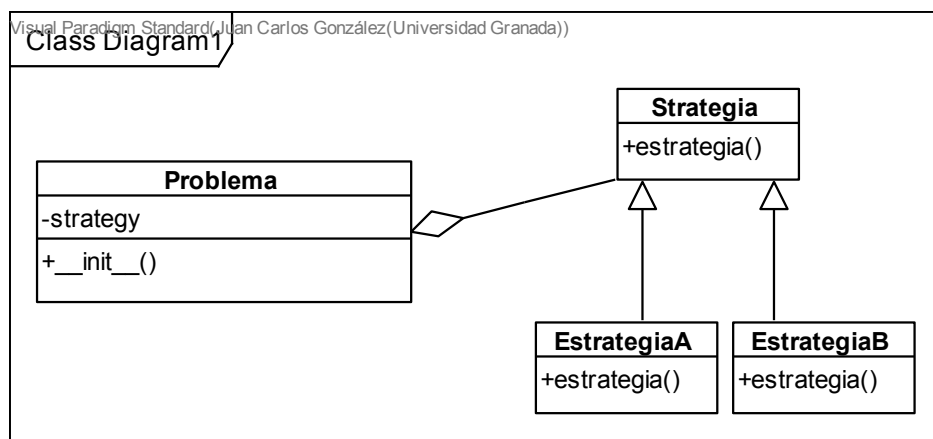


Figura 3.11: Diagrama de clases del patrón Strategy.

### Patrón de diseño MVC

Este patrón de diseño permite separar los datos de la aplicación, la interfaz de usuario y la lógica de control [17]. Entre sus ventajas, tenemos [17, 28]:

- Facilita el mantenimiento del código.
- Permite una mayor escalabilidad y cambios futuros.
- Sencillez a la hora de crear las representaciones de los distintos datos.
- Independencia de funcionamiento.
- Facilidad de realización de pruebas unitarias y corrección eficiente de posibles errores.

Una vez presentado el patrón de diseño, trataremos de forma individual cada una de sus partes:

- **Modelo.** Se encarga del tratamiento de los datos. Permite el acceso a los datos, actualizarlos, insertarlos. En nuestro caso, utilizaremos el lenguaje de MYSQL, donde utilizaremos las sentencias de inserción, borrado y selección para almacenar los datos en nuestra base de datos.
- **Vista.** Conocida como la interfaz de usuario, usaremos HTML junto con CSS para los estilos. Como objetivos tendrá: recibir los datos y mostrarlos al usuario.
- **Controlador.** Es el responsable de hacer de puente entre la vista y el modelo, por lo que se encargará de gestionar el flujo de información entre ellos. Recibirá los eventos que se van a realizar en todo momento y realizará las peticiones y entregas de datos entre el modelo y la vista. Para esta parte se utilizarán los lenguajes de programación PHP y Python.

### 3.3.6. Prototipado final del proceso de diseño

A continuación, se muestran algunos de los *mockups* de la interfaz de usuario más relevantes del sistema:

- **Proceso de opinar.** Estará compuesto por tres procesos: acceso a la página de opinar, selección del problema y rellenar las matrices con las opiniones. Se puede ver gráficamente en la figura 3.12.

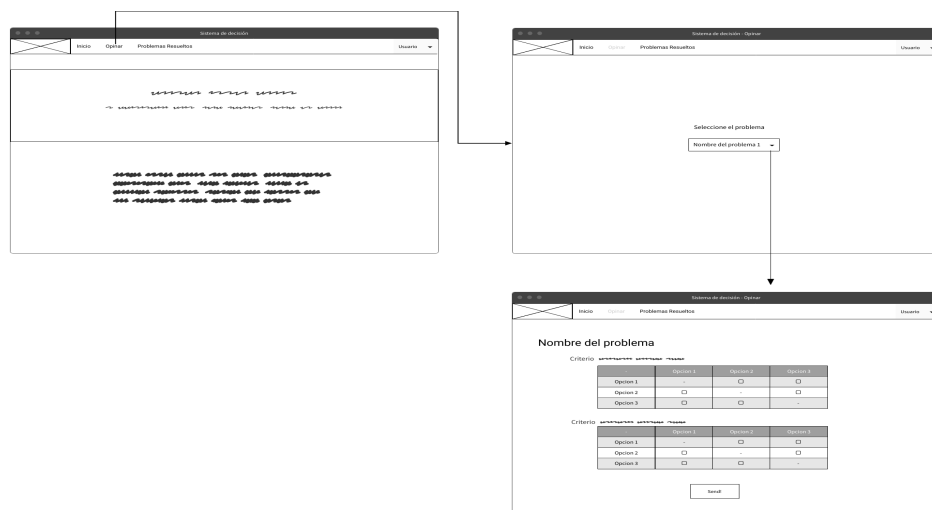


Figura 3.12: *Mockup* de la acción de opinar.



- Proceso de visualizar solución. Estará compuesto por tres procesos: acceso a la página de visualizar solución, selección del problema y obtención de los datos finales del problema. Se puede ver gráficamente en la figura 3.13.

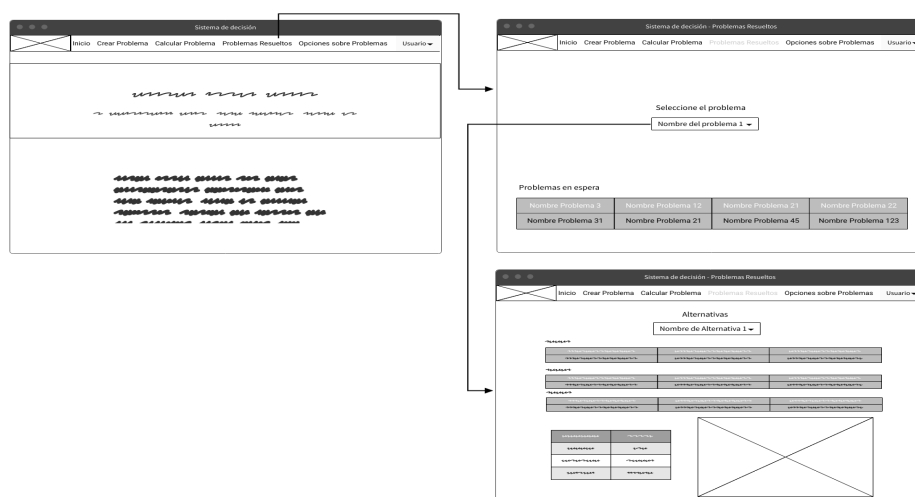


Figura 3.13: *Mockup* de la acción de visualizar la solución de un problema.

## Tipografía

Solo se utilizará una fuente para todo el sistema: «Verdana». La utilizamos debido a que es más abierta y agrega espacio entre letras para darles una sensación más pareja. También utiliza pequeños matices para garantizar las letras «ambiguas». Una de las grandes ventajas es que es muy similar a la letra «Roboto» que es la principal de Google, pero en algunos navegadores provoca problemas de visualización. No se han utilizado más tipografías puesto que no queremos sobrecargar la visualización del usuario.

## Colores

El color principal es más saturado y llamativo, por lo que será el color característico de nuestra aplicación. El secundario será para las zonas o partes de la página que se repetirán en todas como el menú o la identificación. Para cada color, se generan también versiones más claras y más oscuras, debido a que se usan como color de selección de botón, deselección, etc. No obstante, se utilizarán otros colores para resaltar las soluciones finales y el proceso de rellenar las opiniones, para no dotarla de monotonía pero sin llegar a sobrecargar al usuario.

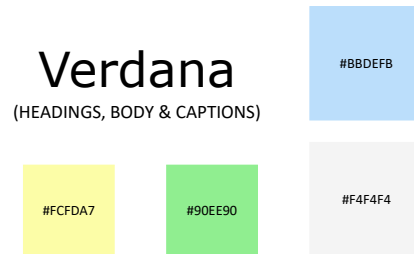


Figura 3.14: MoodBoard del sistema.

El MoodBoard del sistema se puede ver en la figura 3.14.

Pale Cornflower Blue: #BBDEFB.

White Smoke: #F4F4F4.

Ligth Green: #90EE90.

Shalimar: #FCFDA7.

### Sitemap

El mapa del sitio web del proyecto lo dividiremos en dos partes: el del administrador (figura 3.15) y el del usuario (figura 3.16)

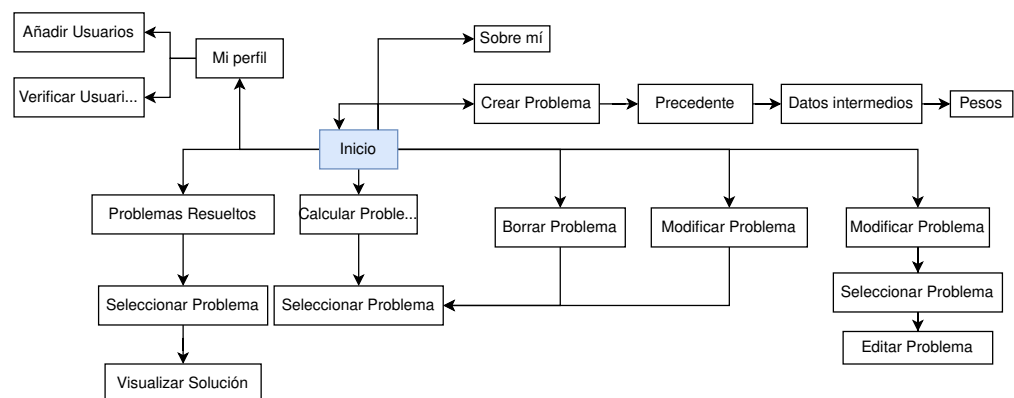


Figura 3.15: Mapa del sitio web para el administrador.

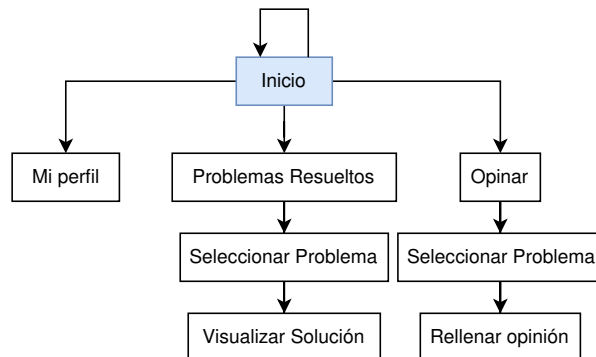


Figura 3.16: Mapa del sitio web para los usuarios.

### 3.4. Implementación

A continuación, describiremos las herramientas utilizadas para realizar la implementación del sistema y algunas cuestiones relativas a la seguridad y prevención de errores.

#### 3.4.1. Herramientas utilizadas

Las herramientas que hemos usado para la implementación del sistema, así como las razones para su elección, son las siguientes:

- Python, Jupyter Notebook y Visual Studio Code. La parte matemática del modelo se implementará mediante el lenguaje de programación Python. Se ha elegido este lenguaje de programación por la cantidad de recursos y ventajas que ofrece. Por un lado, permite una programación orientada a objetos muy sencilla, al igual que el uso de funciones, y declaraciones sencillas. Por otro, existen numerosas librerías y funciones que facilitan el trabajo con matrices, vectores y conversión de datos. Utilizaremos Jupyter Notebook con Python ya que nos ofrece una interfaz muy amigable y se pueden ir realizando pruebas por bloques, sin necesidad de realizar una ejecución total del programa. Paralelamente, y con la ayuda de Visual Studio Code, se realizará una librería de funciones, en un archivo *.py*, para que las funciones que estén ya terminadas vayan guardándose en la librería para ir modularizando el código.
- PhpStorm 2021. Para el desarrollo del sistema web usaremos un docker (ver Apéndice A) que nos hará de servidor. Para trabajar con el código, utilizaremos la aplicación PhpStorm 2021 debido a su asistencia inteligente a la codificación, navegación y detección de errores, a lo que se añade también una interfaz amigable.

- phpMyAdmin. Para el manejo de la base de datos, usaremos phpMyAdmin debido a su interfaz y facilidad de uso, a que permite realizar la creación, modificación y borrado de datos, tablas y columnas de manera más rápida, y porque permite verificar con mayor seguridad la creación y consultas del sistema a la base de datos.
- TWIG. Para realizar el patrón de diseño MVC, nos ayudaremos del motor de plantillas conocido como TWIG. Aparte de ser el más utilizado por los desarrolladores, permite una gran facilidad a la hora del desarrollo web y el uso del patrón de diseño MVC [49]. Algunas de sus ventajas son:
  - Debido al uso de *templates*, compila el código PHP de forma muy optimizada.
  - Ofrece módulos de seguridad como «sandbox» para evaluar los códigos no verificados.
  - Posee plantillas orientadas a sintaxis.
  - Muy fácil de aprender y de utilizar.
  - Permite un sistema de herencia para eliminar la repetición de código. Para ello, se utilizará el sistema por bloques.
- HTML, CSS y MySQL. Para la vista se utilizará HTML y CSS mientras que la base de datos (modelo) se implementará en MySQL.

### 3.4.2. Seguridad y prevención de errores

A continuación, se detallarán las distintas características del sistema que lo dotan de mayor seguridad y protección contra fallos o errores:

1. URL's semánticas. Son aquellas direcciones que son más entendibles para el usuario.

URL: *https://www.mi-proyecto/loguear.php*

URL Limpia: *https://www.mi-proyecto/logueo*

Esta forma de escribir y mostrar las urls, no solo deja que el usuario pueda entender mejor qué realiza la página que está visualizando, sino que ayudará al sistema a la prevención de ataques por inyección de código, como, por ejemplo:

- Consulta a la base de datos. Supongamos que a nuestra url le pasamos el id del problema que queremos calcular para luego realizar una consulta como, por ejemplo, “*SELECT \* FROM problemas WHERE idProblema=(id)*”. Si, ahora, en nuestra url que es *https://www.mi-proyecto/seleccionarProblema.php?id=121*, le añaden “*DELETE FROM problemas*”, esto haría que se puedan borrar las tablas y tener un resultado poco satisfactorio. Este inconveniente se elimina usando este tipo de urls.

- Script malicioso. Ocurre lo mismo si, en vez de añadir alguna sentencia por inyección, lo que añaden es la ruta a un archivo que contiene unas líneas maliciosas. Eso pondría en grave peligro a nuestro sistema o al servidor donde estemos alojados. Sin embargo, no sucederá este problema puesto que al usar url limpias, la composición de las paginas, o el nombre de los archivos es totalmente opaco al usuario.
2. Comprobación de variables en formularios. Cuando se realiza una opinión o se crea un problema, se hace uso del parámetro 'require' para que el usuario no pueda avanzar sin completar todos los campos. Sin embargo, puede eliminarlo dándole a inspeccionar y enviando el formulario sin ningún dato. Para ello, podríamos establecer un segundo control con un *script* de JavaScript. No obstante, puede desactivar este lenguaje, permitiendo así el envío del formulario vacío. Por lo que solo comprobamos varios formularios, para una mayor seguridad, pero la verdadera validación se hace internamente desde PHP. Para evitar que se creen problemas, realizamos una función en PHP que comprueba si estas variables están o no vacías y si contienen el tipo y estructura de dato requerido. Si se obtiene un resultado positivo, es decir, se ha intentado evadir los formularios, se cancela el proceso. A continuación, se describe a grandes rasgos sus funciones de seguridad y validación de datos:
- Rellenar opiniones. Para realizar la opinión sobre un problema, será necesario añadir:

$$alternativas = numeroAlternativas^2 - numeroAlternativas$$

$$opiniones = alternativas * numCriterios$$

La primera comprobación será conocer si el número de variables no vacías es igual al número de datos que debería haber introducido. Si pasa ese filtro, comprobaremos si es un número, y no letras o caracteres extraños, por lo que utilizaremos una expresión regular:

```
1 preg_match("/[\d]/", $opinion)
```

De esta forma rechazaremos todo lo que no sea numérico. Una vez comprobado este paso, necesitaremos que tenga el formato de 0 o 1, la coma del decimal y dos valores más: (0.87, 1.0, 0.5, 0.01, ...) En otro caso, habrá metido (2, 123, 12.4, ...). Para ello, haremos uso de otra expresión regular. Antes, comprobamos si el tamaño es uno, en ese caso, significa que el usuario introdujo 1 o 0:

```

1 if(strlen($opinión)==1)
2   if($opinión==1 or $opinión==0)

```

En otro caso, solamente podrá valer tres (con un decimal) o cuatro (con dos decimales) y, por supuesto, lo mencionado anteriormente sobre el formato:

```

1 if(strlen($opinión)==3 or strlen($opinión)==4)
2 if(preg_match("/[0-1]\.\.[\d]/", $opinión))

```

Llegados a este punto, no se guardará nada en la base de datos que no sea un número válido.

- Crear problema. Para crear el problema habrá tres formularios:
  - a) Primer paso. Se obtiene el número de expertos, criterios y alternativas. Estos números tienen que ser enteros y superiores a un cierto número:

```

1 validarPredecesor($expertos, $criterios,
   $alternativas)

```

Comprobamos que:

$$\text{numeroExpertos} \geq 2$$

$$\text{numeroCriterios} \geq 2$$

$$\text{numeroAlternativas} \geq 3$$

En la función, se comprueba también que son números enteros, puesto que podría haber letras en los *inputs*.

- b) Segundo paso. Ahora tendremos que comprobar lo siguiente:

```

1 if(strlen($nombre)>99){
2   $correcto=False;
3 }
4 if(strlen($descripcion)>399){
5   $correcto=False;
6 }
7 if(strlen($nombreAlternativas)>999 or strlen(
   $nombreCriterios)>999){
8   $correcto=False;
9 }
10 if(empty(validarNombreCantidad($nombreAlternativas
   , $numAlternativas))){
11   $correcto=False;
12 }
13 if(empty(validarNombreCantidad($nombreCriterios,
   $numCriterios))){
14   $correcto=False;
15 }
16 if(empty(validarNombreCantidad($nombreUsuarios,
   $numUsuarios))){
17   $correcto=False;

```

```

18 }
19 if(empty(esDecimal($alfa)) or (empty(esDecimal(
    $beta)))){
20     $correcto=False;
21 }
22 if(empty(esDecimal($gamma)) or (empty(esDecimal(
    $umbral)))){
23     $correcto=False;
24 }
25 if(empty(validarPesos($pesosCriterios,
    $numCriterios))){
26     $correcto = False;
27 }

```

Ni el nombre, ni la descripción deben superar los caracteres establecidos en la base de datos. Los nombres de las alternativas y de los criterios, deben coincidir con el número de criterios y alternativas. Los valores  $\alpha$ ,  $\beta$ ,  $\gamma$  y el umbral, deben ser valores numéricos entre el cero y el uno, por último, el número asignado a los pesos debe estar entre cero y uno, y en su totalidad, deben coincidir con el número de criterios. Respecto al tamaño del nombre de las alternativas o criterios, se comprobará que el string final que contiene todos los nombres no sea superior al establecido en la base de datos, puesto que no tendría sentido limitar el tamaño del nombre de forma individual. Si estas comprobaciones no dan error, es porque el formulario está correcto y, por tanto, pasaríamos al último paso.

- c) Tercer paso. Se le pedirá al administrador que de el valor del peso de cada usuario con respecto a cada criterio. Para ello, deberá volver a introducir:

$$\text{numeroPesos} = \text{numeroExpertos} * \text{numeroCriterios}$$

Para comprobar que se realiza de forma correcta, usaremos una función ya definida anteriormente:

```

1 #En otro .php
2 $numPesos = $expertos*$criterios;
3
4 #Funcion para validar
5 function validarPesos($pesos, $numPesos){
6     $pesosAux = explode(",", $pesos);
7     $resultado = True;
8     if(sizeof($pesosAux)==$numPesos){
9         for($i=0; $i<sizeof($pesosAux) and
        $resultado=True ;$i++){
10             $resultado = esDecimal($pesosAux[$i]);
11         }
12     }

```

```
13     else{  
14         $resultado = False;  
15     }  
16     return $resultado;  
17 }
```

Tenemos que comprobar que el tamaño de pesos dados es el mismo que el de los pesos que se necesitan y que es un número válido.

Si todo lo dicho anteriormente es correcto, es porque no se ha intentado malmeter al sistema, por lo que se procederá a realizar la consulta a la base de datos para la creación del problema, y se le mandará un mensaje a los expertos para que puedan opinar sobre el problema asignado.

## 3.5. Pruebas

En esta sección, describiremos cómo han ido evolucionando las diferentes versiones del sistema y algunas de las pruebas realizadas a este.

### 3.5.1. Versiones del sistema

Durante el proceso de desarrollo del sistema se han implementado diferentes versiones sobre las que se han añadido las sucesivas mejoras requeridas para satisfacer las necesidades del cliente (director) y completar los objetivos marcados. A grandes rasgos, el sistema, el cual se ha dividido en dos partes, ha ido evolucionando durante estos meses de trabajo de la siguiente forma:

- Primera parte. Durante los dos primeros meses, se ha ido desarrollando la parte matemática, que consta de las funciones implementadas en Python, junto a las clases y el patrón de diseño, y varios datos de prueba para comparar resultados.
- Segunda parte. Se desarrolla el sistema web, que se divide en las siguientes versiones:
  - Primera versión. Se realiza el inicio de sesión y registro de usuarios, junto a la creación de los problemas. Se comprueba que los datos se almacenan de forma correcta y sin dar problemas en la conexión a la base de datos.
  - Segunda versión. Se añade la funcionalidad matemática uniendo PHP con Python. De esta forma, ya se pueden calcular las soluciones a los problemas creados. Por ahora, el único valor que se le pasa de la base de datos es la opinión de todos los expertos. Por ello, se calcula con datos generados en el propio Python. Se



añade la opción de visualización que muestra el grado de no dominancia guiado por cuantificador de cada alternativa. Para la parte del experto, se realiza la funcionalidad para poder opinar sobre un problema.

- Tercera versión. Se corrigen algunos errores en el formulario de crear problemas, se añaden otros factores como el umbral y parámetros internos y se añade a la visualización la gráfica evolutiva del proceso, los valores iniciales y finales de consenso y consistencia y los parámetros internos.
- Cuarta versión. Se añaden los nombres, pesos y descripción a la hora de opinar sobre un problema y ver la solución. Cuando se crea un problema, se le añaden los pesos de cada experto y el peso de cada criterio. Se añaden las funcionalidades de borrar un problema o recalcularlo por si ocurrió un error en el proceso. Se asocia con Gmail, para que de esta forma, cuando se realiza alguna función importante en el sistema se notifique mediante un correo electrónico al usuario indicado.
- Quinta versión. Se añade la funcionalidad de modificar un problema, la cual cambiará algunos de los parámetros internos. De esta forma, el administrador podrá ver y realizar un estudio de como afectan los parámetros a las soluciones.
- Sexta versión. Se corrigen errores ortográficos, de diseño y estilos, se comprueba que todas las funcionalidades se realizan sin problema y el envío de correos es correcto. El sistema está listo para entregarlo al cliente.

### **3.5.2. Pruebas realizadas**

A continuación, se expondrán las diferentes pruebas a las que ha sido sometido el sistema, para comprobar su fiabilidad, seguridad y rapidez.

| Prueba número 1  |   |
|------------------|---|
| Nombre           | Creación de un problema.  |
| Condiciones      | En algunos de los campos no se pueden meter más caracteres de los que se han establecido en la base de datos. En los campos numéricos solo se puede meter letras y los nombres de los usuarios no pueden ser más de los introducidos al principio.  |
| Desarrollo       | Se intenta meter mucho texto en algunos de los campos, sin embargo, solo deja añadir hasta que se va a superar el límite que se ha establecido. Si el número de usuarios no es el correcto, no deja avanzar a la siguiente página y, en los campos numéricos, no se puede añadir algo que no sea un número decimal y este debe estar entre dos valores.                                     |
| Resultado        | Éxito.  |
| Posibles errores | Todo está establecido en los input y funcionalidades que nos ofrece HTML, por lo que no se pueden meter valores extraños. El único problema que encontramos es que meta un nombre de usuario que no se encuentre en la base de datos, pero tampoco se puede hacer puesto que gracias a una función JavaScript, el usuario solo podrá elegir y añadir nombres que se encuentren registrados. |

| Prueba número 2  |  |
|------------------|--|
| Nombre           | Calcular un problema.  |
| Condiciones      | Siempre se van a extraer los datos de la misma forma, por lo que lo único que hay que comprobar es que haya algún campo vacío. En ese caso, dará un error que lo notificará al administrador.  |
| Desarrollo       | Se ejecuta un problema que está completo, y después de unos minutos da una solución y notifica a los usuarios que intervienen en el mismo. Por otro lado, eliminamos de forma manual uno de los parámetros en la base de datos para comprobar si se controla que haya errores. Como era de esperar, se notifica al administrador que hay un error en alguno de los parámetros. |
| Resultado        | Éxito.   |
| Posibles errores | A no ser que haya un ataque, o alguien de forma intencionada acceda a la base de datos y borre errores o se acceda al código fuente y modifiquen parámetros, es prácticamente imposible que se cambie algún dato, o esté vacío o con una estructura incorrecta.  |

| Prueba número 3  |   |
|------------------|---|
| Nombre           | Borrar un problema.   |
| Condiciones      | El problema que se va a eliminar debe haber sido creado.  |
| Desarrollo       | Se selecciona el nombre del problema que se quiere borrar y se pulsa en el botón de eliminar problema. Saldrá un mensaje emergente para notificar su confirmación. Se borrarán también los datos, opiniones y problemas asociados al mismo. |
| Resultado        | Éxito.  |
| Posibles errores | Que no se borre el problema seccionado. Sin embargo, al coger siempre el id asociado al nombre, no debe dar ningún problema.  |

| Prueba número 4  |  |
|------------------|--|
| Nombre           | Modificar un problema.   |
| Condiciones      | El problema que se va a modificar debe haber sido creado.  |
| Desarrollo       | Se selecciona el nombre del problema que se quiere modificar y se pulsa en el botón de modificar problema. Se mostrarán los valores que se pueden modificar con los valores actuales. Una vez que se modifiquen, se comprobará que los valores son los correctos y se realizará una consulta a la base de datos. |
| Resultado        | Éxito.   |
| Posibles errores | Que no se modifique el problema seleccionado o que los datos que se introduzcan provoquen inconsistencias. Sin embargo, al coger siempre el id asociado al nombre, no debe dar ningún problema al igual que al pasar las comprobaciones que realizamos cuando el problema se crea, tampoco puede haber errores.  |

- Prueba unitaria en Python (número 5). Ahora vamos a probar en varias funciones que utilizamos en nuestro sistema si los valores que calcula son los que nosotros hemos obtenido a mano.

```

1 def pruebaUnitariaConsenso(matriz):
2     return str(nivelConsistenciaMatriz(matriz))[0:5]
3
4 def test_func(matriz, resultado):
5     assert pruebaUnitariaConsenso(matriz) == resultado
6     print("Prueba pasada con éxito")

```

La matriz con la que haremos la prueba será la siguiente:

$$\begin{bmatrix} 0 & 0.2 & 0.6 & 0.4 & 1 \\ 0.8 & 0 & 0.9 & 0.7 & 1 \\ 0.4 & 0.1 & 0 & 0.3 & 1 \\ 0.6 & 0.3 & 0.7 & 0 & 1 \\ 0.2 & 0.8 & 0.2 & 0.4 & 0 \end{bmatrix}$$

El valor de coherencia calculado de forma manual es 0.862, y ahora lo comprobaremos con la función anteriormente mencionada:

```

1     problemas = os.listdir("problemas/")
2     opiniones=obtenerTodasOpiniones("1")
3     matriz=opiniones[0][0]
4     test_func(matriz, "0.862")
5
6     #||Prueba pasada con éxito||
7

```

Como podemos comprobar, la prueba ha sido pasada. De esta forma, hemos comprobado la mayoría de las funciones, ya que otras dependen de un número elevado de iteraciones y de una multiplicación muy grande de números decimales o de un conjunto muy largo de pasos.

- Prueba unitaria en PHP. En este apartado, realizaremos varias pruebas sobre las funciones de seguridad creadas.

1. Proceso de opinar (número 6). Comprobaremos que en el momento de enviar la opinión hay tantas opiniones como:

$$\begin{aligned} alternativas &= numeroAlternativas^2 - numeroAlternativas \\ opiniones &= alternativas * numCriterios \end{aligned}$$

Después, comprobaremos que cada uno de los valores introducidos es un número decimal con el siguiente formato:

$$[0 - 1]/./[0 - 9][0 - 9]$$

La función que validará el proceso será la siguiente:

```
1 if(comprobarOpinion($opinion, $id)==True and
   $formatoCorrecto==True)
```

Vamos a introducir de manera manual los valores: 0.1, 0.2, 0.3, 0.4, 0.5, 0.6. Corresponden con la opinión de un experto con tres alternativas y un criterio. La comparación del tamaño es éxito, y luego al comprobar de forma individual (FormatoCorrecto) cada valor, también obtenemos éxito. Por tanto, guardar esa opinión en la base de datos nos asegura un formato correcto. Usaremos estas aserciones para comprobar si efectivamente se realizan las comprobaciones:

```
1 $tamano=0;
2 if(empty(comprobarOpinion($opinion, $id))){
3     $tamano=1;
4 }
5 assert($tamano==1, 'Tamano incorrecto');
6 $formato=0;
7 if(empty($formatoCorrecto)){
8     $formato=1;
9 }
10 assert($formato==1, 'Formato de numero incorrecto');
11
```

Para comprobar que falla, meteremos los siguientes valores:

- 0.1, 0.2, 0.3, 0.4, 12.5, 0.6. Estos nos dan error en la comprobación de que el numero esté entre 0 y 1.

```
1 Warning: assert(): Formato de numero incorrecto
   failed in /var/www/html/user/procesarOpinion.
   php on line 45
```

- 0.1, ho, 0.3, 0.4, 0.5, 0.6.

```
1 Warning: assert(): Formato de numero incorrecto
   failed in /var/www/html/user/procesarOpinion.
   php on line 45
```

- 0.1, 0.2, 0.3, 0.4, 0.5.

```
1 ( ! ) Warning: assert(): Tamano incorrecto failed
   in /var/www/html/user/procesarOpinion.php on
   line 43
```

2. Proceso de crear un problema (número 7). Comprobaremos la primera parte, en la que se tiene que comprobar que haya más de un experto, más de dos alternativas y al menos un criterio. Después habrá que comprobar que cada uno de los valores introducidos es un número entero. La función que validará el proceso será la siguiente:

```
1 if(preg_match("/[\d]/", $expertos) and preg_match("/[\d]/",
   $criterios) and preg_match("/[\d]/",
   $alternativas)){
```

```

2     if(intval($expertos)>=2 and intval($criterios)>=1
    and intval($alternativas)>=3){
3         if(!strstr($expertos, '.') and !strstr(
    $criterios, '.') and !strstr($alternativas, '.')){
4             if(!strstr($expertos, ',') and !strstr(
    $criterios, ',') and !strstr($alternativas, ',')){

```

Vamos a introducir de manera manual los siguientes valores: dos expertos, tres criterios y cinco alternativas. Como cumple todas las condiciones, pasa al siguiente formulario sin ningún inconveniente. Usaremos estas aserciones para comprobar si efectivamente se realizan las comprobaciones:

```

1 $aux=0;
2 if(empty(comprobarPredecesorProblema($expertos,
    $criterios, $alternativas))){
3     $aux=1;
4 }
5 assert($aux==1, 'Error al introducir algun parametro')
    ;

```

Para comprobar que falla, meteremos los siguientes valores:

- Expertos = 0. Esto nos da error en la comprobación de que el número esté entre 0 y 1.

```

1 ( ! ) Warning: assert(): Tamano incorrecto failed
    in /var/www/html/admin/procesarProblema.php on
    line 23

```

- Criterios = "hola".

```

1 ( ! ) Warning: assert(): Tamano incorrecto failed
    in /var/www/html/admin/procesarProblema.php on
    line 23

```

- Alternativas = 2.

```

1 ( ! ) Warning: assert(): Tamano incorrecto failed
    in /var/www/html/admin/procesarProblema.php on
    line 23

```

| Prueba número 8  |   |
|------------------|---|
| Nombre           | Rehacer un problema.  |
| Condiciones      | El problema que se va a eliminar debe haber sido creado.  |
| Desarrollo       | Se selecciona el nombre del problema que se quiere rehacer y se pulsa en el botón de rehacer problema. Se modificará el parámetro de solucionado para que el administrador pueda volver a calcularlo. |
| Resultado        | Éxito.  |
| Posibles errores | Que no se permita recalcular el problema seleccionado. Sin embargo, al coger siempre el id asociado al nombre, no debe dar ningún problema.   |

| Prueba número 9  |   |
|------------------|---|
| Nombre           | Opinar.   |
| Condiciones      | El problema al que se le va a realizar la opinión tiene que haber sido creado.  |
| Desarrollo       | Se selecciona el nombre del problema que se quiere opinar y se rellenan las matrices con valores entre 0 y 1.   |
| Resultado        | Éxito.  |
| Posibles errores | Que se intenten enviar valores vacíos o no numéricos, o fuera del rango establecido. Sin embargo, al realizar comprobaciones antes del envío y antes de insertar en la base de datos, el error es mínimo. |





## Capítulo 4

# Aplicación y análisis

Este capítulo ilustra el funcionamiento del sistema de ayuda a la decisión en grupo basado en computación granular desarrollado y analizando sus características.

### 4.1. Ejemplo ilustrativo

Supongamos que un restaurante quiere añadir una nueva marca de vinos a su carta. Para ello, el restaurante recurre a cuatro expertos sumilleres,  $e_1$ ,  $e_2$ ,  $e_3$  y  $e_4$ , los cuales tienen que elegir el mejor vino de entre cuatro posibles marcas:

- $a_1$ : Hardy's (Australia).
- $a_2$ : Yellow Tail (Australia).
- $a_3$ : Sutter home (Estados Unidos).
- $a_4$ : Changyu (China).

Para la elección del mejor vino, los sumilleres tienen que tener en cuenta los siguientes cuatro criterios, cuyos pesos de importancia, según el restaurante, son 0.2, 0.3, 0.1 y 0.4, respectivamente:

- $c_1$ : aroma.
- $c_2$ : apariencia.
- $c_3$ : sabor.
- $c_4$ : precio.

Teniendo en cuenta el conocimiento y la experiencia de cada uno de los cuatro sumilleres consultados, los pesos o niveles de importancia asignados

a cada uno de los cuatro sumilleres en cada uno de los cuatro criterios son los siguientes:

$$\begin{array}{cccc}
 v_{11} = 0.2 & v_{12} = 0.3 & v_{13} = 0.3 & v_{14} = 0.2 \\
 v_{21} = 0.3 & v_{22} = 0.3 & v_{23} = 0.2 & v_{24} = 0.2 \\
 v_{31} = 0.4 & v_{32} = 0.1 & v_{33} = 0.2 & v_{34} = 0.3 \\
 v_{41} = 0.1 & v_{42} = 0.3 & v_{43} = 0.3 & v_{44} = 0.3
 \end{array}$$

Al inicio del proceso de decisión, cada sumiller tiene que proporcionar cuatro relaciones de preferencia difusas (evaluaciones), una para cada criterio. Inicialmente, las relaciones de preferencia difusas proporcionadas por los cuatro sumilleres son las siguientes:

$$\begin{array}{ll}
 P^{11} = \begin{bmatrix} - & 0.30 & 0.50 & 0.50 \\ 0.50 & - & 0.70 & 0.70 \\ 0.50 & 0.30 & - & 0.70 \\ 0.50 & 0.30 & 0.30 & - \end{bmatrix} & P^{12} = \begin{bmatrix} - & 0.30 & 0.90 & 0.90 \\ 0.70 & - & 0.10 & 0.10 \\ 0.10 & 0.90 & - & 0.70 \\ 0.10 & 0.90 & 0.30 & - \end{bmatrix} \\
 P^{13} = \begin{bmatrix} - & 0.30 & 0.30 & 0.70 \\ 0.90 & - & 0.30 & 0.30 \\ 0.90 & 0.70 & - & 0.30 \\ 0.50 & 0.70 & 0.90 & - \end{bmatrix} & P^{14} = \begin{bmatrix} - & 0.70 & 0.90 & 0.90 \\ 0.30 & - & 0.60 & 0.70 \\ 0.20 & 0.30 & - & 0.80 \\ 0.20 & 0.20 & 0.10 & - \end{bmatrix} \\
 P^{21} = \begin{bmatrix} - & 0.50 & 0.30 & 0.90 \\ 0.50 & - & 0.90 & 0.90 \\ 0.70 & 0.10 & - & 0.50 \\ 0.10 & 0.10 & 0.50 & - \end{bmatrix} & P^{22} = \begin{bmatrix} - & 0.70 & 0.70 & 0.70 \\ 0.10 & - & 0.90 & 0.30 \\ 0.50 & 0.30 & - & 0.90 \\ 0.50 & 0.50 & 0.30 & - \end{bmatrix} \\
 P^{23} = \begin{bmatrix} - & 0.50 & 0.50 & 0.30 \\ 0.50 & - & 0.50 & 0.70 \\ 0.50 & 0.50 & - & 0.50 \\ 0.90 & 0.10 & 0.50 & - \end{bmatrix} & P^{24} = \begin{bmatrix} - & 0.80 & 0.70 & 0.20 \\ 0.20 & - & 0.40 & 0.40 \\ 0.10 & 0.60 & - & 0.80 \\ 0.90 & 0.60 & 0.30 & - \end{bmatrix} \\
 P^{31} = \begin{bmatrix} - & 0.10 & 0.70 & 0.30 \\ 0.70 & - & 0.50 & 0.30 \\ 0.30 & 0.50 & - & 0.50 \\ 0.70 & 0.70 & 0.50 & - \end{bmatrix} & P^{32} = \begin{bmatrix} - & 0.90 & 0.90 & 0.70 \\ 0.30 & - & 0.70 & 0.50 \\ 0.30 & 0.10 & - & 0.70 \\ 0.30 & 0.50 & 0.30 & - \end{bmatrix} \\
 P^{33} = \begin{bmatrix} - & 0.30 & 0.30 & 0.70 \\ 0.50 & - & 0.50 & 0.70 \\ 0.30 & 0.50 & - & 0.50 \\ 0.30 & 0.70 & 0.90 & - \end{bmatrix} & P^{34} = \begin{bmatrix} - & 0.50 & 0.60 & 0.30 \\ 0.50 & - & 0.20 & 0.50 \\ 0.30 & 0.60 & - & 0.70 \\ 0.70 & 0.50 & 0.40 & - \end{bmatrix} \\
 P^{41} = \begin{bmatrix} - & 0.80 & 0.90 & 0.70 \\ 0.30 & - & 0.70 & 0.50 \\ 0.30 & 0.20 & - & 0.70 \\ 0.20 & 0.50 & 0.20 & - \end{bmatrix} & P^{42} = \begin{bmatrix} - & 0.20 & 0.20 & 0.70 \\ 0.80 & - & 0.50 & 0.60 \\ 0.70 & 0.50 & - & 0.60 \\ 0.30 & 0.40 & 0.30 & - \end{bmatrix}
 \end{array}$$

$$P^{43} = \begin{bmatrix} - & 0.50 & 0.30 & 0.70 \\ 0.50 & - & 0.90 & 0.70 \\ 0.80 & 0.10 & - & 0.50 \\ 0.10 & 0.30 & 0.50 & - \end{bmatrix} \quad P^{44} = \begin{bmatrix} - & 0.10 & 0.10 & 0.20 \\ 0.80 & - & 0.20 & 0.80 \\ 0.90 & 0.70 & - & 0.90 \\ 0.80 & 0.10 & 0.10 & - \end{bmatrix}$$

En la figura 4.1, podemos ver un ejemplo de cómo un experto, en este caso, el sumiller  $e_1$ , introduciría sus evaluaciones (relaciones de preferencia difusas) mediante la aplicación web.

Una vez que todos los sumilleres han proporcionado sus relaciones de preferencia difusas, se aplica el algoritmo PSO para mejorar la coherencia y el consenso con el menor ajuste posible en las evaluaciones de los sumilleres. Antes de mostrar los resultados obtenidos, vamos a indicar los valores asignados a los parámetros del algoritmo PSO:

- La población está formada por 100 partículas.
- El número de iteraciones se establece en 300.
- El valor de  $\omega$  se decrementa linealmente desde 0.9 a 0.4. Esto implica que  $\omega(1) = 0.9$  y  $\omega(300) = 0.4$  en (2.1).

Estos valores han sido asignados después de una intensa experimentación. Además, el valor del nivel de granularidad de la información  $\alpha$  se establece a 0.2, mientras que los valores asignados a  $\beta$  y  $\gamma$  son 0.75 y 0.5, respectivamente. Suponemos, por tanto, que, por un lado, el restaurante le da más importancia a la mejora del consenso y la coherencia sobre la similitud ( $\beta = 0.75$ ) y, por otro, le da la misma importancia al consenso y a la coherencia ( $\gamma = 0.5$ ).

UNIVERSIDAD DE GRANADA Inicio Opinar Problemas Resueltos victor

### Rellene su opinión

Problema con el nombre '**Selección de vinos**'

Criterio **Aroma** con peso : 0.2

|             | Hardys | Yellow tail | Sutter home | Changyu |
|-------------|--------|-------------|-------------|---------|
| Hardys      | -      | 0.3         | 0.5         | 0.5     |
| Yellow tail | 0.5    | -           | 0.7         | 0.7     |
| Sutter home | 0.5    | 0.3         | -           | 0.7     |
| Changyu     | 0.5    | 0.3         | 0.3         | -       |

Criterio **Apariencia** con peso : 0.3

|             | Hardys | Yellow tail | Sutter home | Changyu |
|-------------|--------|-------------|-------------|---------|
| Hardys      | -      | 0.3         | 0.9         | 0.9     |
| Yellow tail | 0.7    | -           | 0.1         | 0.1     |
| Sutter home | 0.1    | 0.9         | -           | 0.7     |
| Changyu     | 0.1    | 0.9         | 0.3         | -       |

Criterio **Sabor** con peso : 0.1

Figura 4.1: Ejemplo de evaluaciones expresadas usando la aplicación web.

Una vez descritos los valores asignados a los distintos parámetros del modelo, podemos mostrar los resultados devueltos por este. Así, una vez ejecutado el algoritmo PSO, la coherencia global obtenida,  $f_1$ , es 0.917, el consenso alcanzado entre los sumilleres,  $f_2$ , es 0.825 y la similitud entre las relaciones de preferencia difusas proporcionadas por estos y las ajustadas por el algoritmo PSO,  $f_3$ , es 0.915 (ver figura 4.2).

A continuación, se muestran las relaciones de preferencia difusas ajustadas que devuelve el algoritmo PSO:

$$\begin{aligned}
\bar{P}^{11} &= \begin{bmatrix} - & 0.40 & 0.60 & 0.60 \\ 0.60 & - & 0.60 & 0.60 \\ 0.40 & 0.40 & - & 0.60 \\ 0.40 & 0.40 & 0.40 & - \end{bmatrix} & \bar{P}^{12} &= \begin{bmatrix} - & 0.40 & 0.80 & 0.80 \\ 0.60 & - & 0.20 & 0.20 \\ 0.20 & 0.80 & - & 0.60 \\ 0.20 & 0.80 & 0.40 & - \end{bmatrix} \\
\bar{P}^{13} &= \begin{bmatrix} - & 0.40 & 0.40 & 0.80 \\ 0.80 & - & 0.40 & 0.30 \\ 0.80 & 0.68 & - & 0.40 \\ 0.40 & 0.60 & 0.80 & - \end{bmatrix} & \bar{P}^{14} &= \begin{bmatrix} - & 0.60 & 0.80 & 0.80 \\ 0.40 & - & 0.60 & 0.60 \\ 0.30 & 0.40 & - & 0.70 \\ 0.30 & 0.30 & 0.20 & - \end{bmatrix} \\
\bar{P}^{21} &= \begin{bmatrix} - & 0.60 & 0.40 & 0.80 \\ 0.40 & - & 0.80 & 0.80 \\ 0.60 & 0.20 & - & 0.60 \\ 0.20 & 0.20 & 0.40 & - \end{bmatrix} & \bar{P}^{22} &= \begin{bmatrix} - & 0.60 & 0.60 & 0.69 \\ 0.20 & - & 0.80 & 0.40 \\ 0.49 & 0.40 & - & 0.80 \\ 0.60 & 0.50 & 0.40 & - \end{bmatrix} \\
\bar{P}^{23} &= \begin{bmatrix} - & 0.60 & 0.60 & 0.40 \\ 0.40 & - & 0.60 & 0.69 \\ 0.40 & 0.49 & - & 0.60 \\ 0.80 & 0.20 & 0.40 & - \end{bmatrix} & \bar{P}^{24} &= \begin{bmatrix} - & 0.70 & 0.60 & 0.30 \\ 0.30 & - & 0.50 & 0.50 \\ 0.20 & 0.50 & - & 0.70 \\ 0.80 & 0.50 & 0.40 & - \end{bmatrix} \\
\bar{P}^{31} &= \begin{bmatrix} - & 0.20 & 0.60 & 0.40 \\ 0.60 & - & 0.50 & 0.40 \\ 0.20 & 0.50 & - & 0.60 \\ 0.60 & 0.60 & 0.40 & - \end{bmatrix} & \bar{P}^{32} &= \begin{bmatrix} - & 0.80 & 0.80 & 0.60 \\ 0.40 & - & 0.60 & 0.50 \\ 0.20 & 0.20 & - & 0.60 \\ 0.40 & 0.60 & 0.40 & - \end{bmatrix} \\
\bar{P}^{33} &= \begin{bmatrix} - & 0.30 & 0.40 & 0.60 \\ 0.50 & - & 0.50 & 0.60 \\ 0.20 & 0.50 & - & 0.60 \\ 0.40 & 0.60 & 0.80 & - \end{bmatrix} & \bar{P}^{34} &= \begin{bmatrix} - & 0.50 & 0.60 & 0.40 \\ 0.50 & - & 0.30 & 0.40 \\ 0.20 & 0.50 & - & 0.60 \\ 0.60 & 0.60 & 0.40 & - \end{bmatrix} \\
\bar{P}^{41} &= \begin{bmatrix} - & 0.70 & 0.80 & 0.60 \\ 0.40 & - & 0.60 & 0.60 \\ 0.40 & 0.30 & - & 0.80 \\ 0.30 & 0.40 & 0.30 & - \end{bmatrix} & \bar{P}^{42} &= \begin{bmatrix} - & 0.30 & 0.30 & 0.60 \\ 0.70 & - & 0.60 & 0.70 \\ 0.80 & 0.41 & - & 0.70 \\ 0.30 & 0.50 & 0.30 & - \end{bmatrix} \\
\bar{P}^{43} &= \begin{bmatrix} - & 0.45 & 0.30 & 0.60 \\ 0.60 & - & 0.80 & 0.80 \\ 0.80 & 0.20 & - & 0.60 \\ 0.20 & 0.30 & 0.40 & - \end{bmatrix} & \bar{P}^{44} &= \begin{bmatrix} - & 0.20 & 0.20 & 0.30 \\ 0.70 & - & 0.30 & 0.80 \\ 0.80 & 0.60 & - & 0.80 \\ 0.70 & 0.20 & 0.20 & - \end{bmatrix}
\end{aligned}$$

El siguiente paso consiste en aplicar el proceso de selección sobre las relaciones de preferencia difusas devueltas por el algoritmo PSO para obtener la marca de vino que debería añadir el restaurante a su carta.

Una vez que la coherencia y el consenso han sido mejorados con un ajuste mínimo, se aplica el proceso de selección para obtener el mejor vino. Utilizando (2.16) junto con el cuantificador lingüístico «mayoría», modelado por el cuantificador  $Q(r) = r^{1/2}$ , para obtener el vector de pesos del operador de agregación IOWA de acuerdo con (2.15), obtenemos las siguientes relaciones de preferencia difusas colectivas para los criterios:

$$\begin{aligned}\overline{PC}^1 &= \begin{bmatrix} - & 0.52 & 0.53 & 0.67 \\ 0.47 & - & 0.68 & 0.67 \\ 0.47 & 0.30 & - & 0.63 \\ 0.31 & 0.33 & 0.38 & - \end{bmatrix} & \overline{PC}^2 &= \begin{bmatrix} - & 0.54 & 0.62 & 0.68 \\ 0.39 & - & 0.61 & 0.42 \\ 0.44 & 0.46 & - & 0.72 \\ 0.44 & 0.57 & 0.38 & - \end{bmatrix} \\ \overline{PC}^3 &= \begin{bmatrix} - & 0.49 & 0.48 & 0.54 \\ 0.53 & - & 0.58 & 0.62 \\ 0.52 & 0.48 & - & 0.56 \\ 0.57 & 0.35 & 0.54 & - \end{bmatrix} & \overline{PC}^4 &= \begin{bmatrix} - & 0.57 & 0.58 & 0.42 \\ 0.41 & - & 0.46 & 0.55 \\ 0.31 & 0.49 & - & 0.70 \\ 0.65 & 0.42 & 0.33 & - \end{bmatrix}\end{aligned}$$

La relación de preferencia difusa colectiva,  $\overline{PC}$ , se calcula mediante la agregación de  $\overline{PC}^1$ ,  $\overline{PC}^2$ ,  $\overline{PC}^3$  y  $\overline{PC}^4$ . De nuevo, esto se realiza mediante (2.17) y el cuantificador lingüístico «mayoría», modelado por el cuantificador  $Q(r) = r^{1/2}$ , el cual genera el vector de pesos del operador de agregación IOWA de acuerdo con (2.15):

$$\overline{PC} = \begin{bmatrix} - & 0.54 & 0.56 & 0.53 \\ 0.44 & - & 0.55 & 0.57 \\ 0.40 & 0.45 & - & 0.67 \\ 0.54 & 0.41 & 0.38 & - \end{bmatrix}$$

Por último, utilizando (2.18) y el cuantificador lingüístico «mayoría», modelado por el cuantificador  $Q(r) = r^{1/2}$ , para obtener, en este caso, el vector de pesos del operador de agregación OWA de acuerdo con (2.15), obtenemos los siguientes grados de no dominancia guiados por cuantificador (ver figura 4.2):

$$QGNDD_1 = 0.998$$

$$QGNDD_2 = 0.981$$

$$QGNDD_3 = 0.976$$

$$QGNDD_4 = 0.973$$

Considerando estos valores, el restaurante debería añadir a su carta de vinos el de la marca australiana Hardy's, según las opiniones de los cuatro sumilleres.

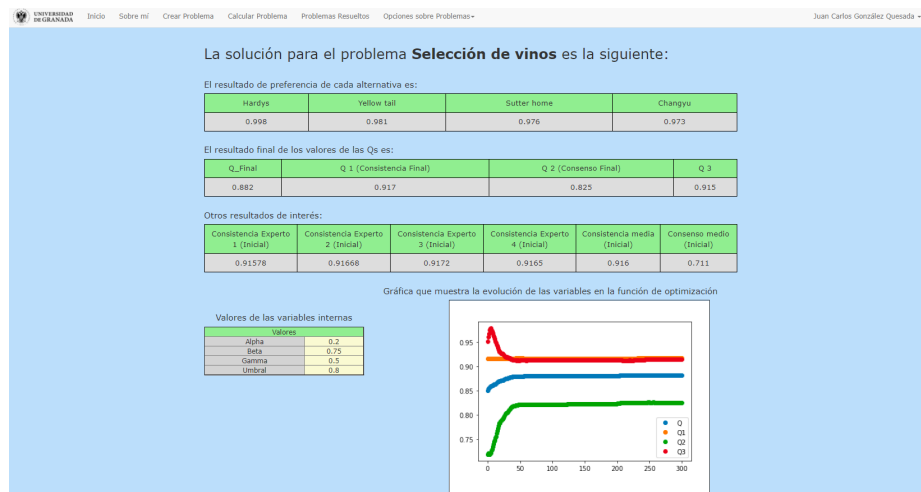


Figura 4.2: Visualización de la solución del problema en la aplicación web.

Para finalizar el ejemplo, en la figura 4.2, podemos visualizar los datos más relevantes del problema. Entre otros, se muestran los datos iniciales de la coherencia y el consenso, es decir, cuando se utilizan las evaluaciones iniciales proporcionadas por los sumilleres; se muestra la evolución del valor devuelto por la función de aptitud en sucesivas iteraciones del algoritmo PSO, así como la evolución de la coherencia, el consenso y la similitud; se muestran los valores finales devueltos por el algoritmo PSO de la coherencia, el consenso y la similitud; y se muestran los grados de no dominancia guiados por cuantificador de cada una de las alternativas del problema, en este caso concreto, de los distintos vinos.

## 4.2. Análisis

La comparación de los resultados ofrecidos por distintos modelos de toma de decisiones en grupo no es una tarea sencilla cuando estos abordan diferentes características del proceso de toma de decisiones. Los aspectos considerados por los modelos son diferentes y, en consecuencia, una comparación cuantitativa no tendría sentido. Sin embargo, como ya se ha mencionado anteriormente, el modelo propuesto basado en el paradigma de computación granular es capaz de considerar más características de los procesos de toma de decisiones en grupo que los modelos basados en computación granular existentes. No obstante, presenta un mayor número de parámetros. En cualquier caso, a continuación, analizaremos el rendimiento del modelo desde diferentes puntos de vista.

#### 4.2.1. Nivel de granularidad de la información $\alpha$

Con la finalidad de situar los resultados obtenidos por nuestro modelo en un cierto contexto, vamos a mostrar los resultados alcanzados cuando se utilizan directamente las evaluaciones iniciales proporcionadas por los expertos. Esto es lo mismo que asignar a  $\alpha$  el valor 0. En este caso, la coherencia,  $f_1$ , vale 0.916 y el consenso alcanzado,  $f_2$ , es 0.711 (ver figura 4.2). Si comparamos estos valores con los resultados obtenidos cuando  $\alpha = 0.2$  (el valor utilizado en el ejemplo ilustrado anteriormente), es decir, cuando se permite cierta flexibilidad en las evaluaciones iniciales expresadas por los expertos (sumilleres en el caso del ejemplo), tanto la coherencia como el consenso alcanzados por nuestro modelo son mayores. Por tanto, es interesante analizar cómo mejoran la coherencia y el consenso y cómo empeora la similitud entre las evaluaciones iniciales y las modificaciones por el modelo en función del valor del nivel de granularidad de la información  $\alpha$ .

La tabla 4.1 muestra los valores de  $f$ ,  $f_1$ ,  $f_2$  y  $f_3$ , para diversos valores de  $\alpha$ , usando las mismas relaciones de preferencia difusas proporcionadas inicialmente por los expertos en el ejemplo anterior y teniendo  $\beta$  y  $\gamma$  los mismos valores que los utilizados en el ejemplo anterior, es decir,  $\beta = 0.75$  y  $\gamma = 0.5$ . Podemos observar que cuanto mayor es el nivel de granularidad de la información permitido mayor es la probabilidad de que la coherencia,  $f_1$ , y el consenso,  $f_2$ , alcancen valores mayores. Sin embargo, también es mayor la probabilidad de que las evaluaciones modificadas estén más alejadas de las proporcionadas inicialmente por los expertos, es decir, la probabilidad de conseguir peores valores para  $f_3$  es mayor. Así, cuando  $\alpha = 1.0$ , tanto  $f_1$  como  $f_2$  alcanzan sus mejores valores, mientras que  $f_3$  obtiene su peor valor. Por el contrario, cuando  $\alpha = 0.2$ ,  $f_3$  alcanza su mejor valor, mientras que  $f_1$  y  $f_2$  obtienen sus valores más bajos. Esto es lógico ya que estamos asignando a las evaluaciones un cierto nivel de flexibilidad del cual podemos tomar ventaja. Esto significa que al permitir un intervalo mayor, la probabilidad de mejorar la coherencia y el consenso es mayor, pero la probabilidad de obtener una evaluación modificada más alejada de la inicial también lo es. Destaca la mejora conseguida en el consenso, ya que, en este caso concreto, la coherencia ya era alta de por sí para las evaluaciones proporcionadas inicialmente por los expertos.

Tabla 4.1: Valores de  $f$ ,  $f_1$ ,  $f_2$  y  $f_3$ , para distintos valores de  $\alpha$ .

|                | $f$   | $f_1$ | $f_2$ | $f_3$ |
|----------------|-------|-------|-------|-------|
| $\alpha = 0.2$ | 0.882 | 0.917 | 0.825 | 0.915 |
| $\alpha = 0.4$ | 0.904 | 0.917 | 0.916 | 0.868 |
| $\alpha = 0.6$ | 0.912 | 0.918 | 0.950 | 0.847 |
| $\alpha = 0.8$ | 0.915 | 0.918 | 0.961 | 0.842 |
| $\alpha = 1.0$ | 0.916 | 0.919 | 0.967 | 0.836 |

Tabla 4.2: Valores de  $f_1$  y  $f_2$  para determinados valores de  $\gamma$ .

|                 | $f_1$ | $f_2$ |
|-----------------|-------|-------|
| $\gamma = 0.00$ | 0.915 | 0.924 |
| $\gamma = 0.25$ | 0.916 | 0.917 |
| $\gamma = 0.50$ | 0.917 | 0.916 |
| $\gamma = 0.75$ | 0.918 | 0.827 |
| $\gamma = 1.00$ | 0.919 | 0.712 |

#### 4.2.2. Parámetro $\gamma$

La tabla 4.2 muestra el impacto del parámetro  $\gamma$ , el cual se encuentra en la función de aptitud  $f$ . Estos valores se obtienen usando las mismas relaciones de preferencia difusas proporcionadas inicialmente por los expertos en el ejemplo anterior y siendo  $\alpha = 0.4$  y  $\beta = 0.75$ . Como este parámetro establece un compromiso entre la coherencia y el consenso, nos vamos a focalizar en los valores alcanzados por  $f_1$  y  $f_2$ , que son los que nos interesan. Por un lado, cuando  $\gamma$  es igual a 1, el algoritmo PSO se centra en maximizar de la coherencia de los expertos. Como resultado,  $f_1$  obtiene sus valores más altos. Por otro lado, cuando  $\gamma$  va tomando valores inferiores a 1, el algoritmo PSO se focaliza tanto en la coherencia como en el consenso y, por tanto,  $f_1$  alcanza valores más bajos, lo cual es de esperar. En particular, cuando  $\gamma$  es igual a 0, el algoritmo PSO se centra solamente en el consenso alcanzado entre los expertos. Esto significa que se obtienen los valores más altos de  $f_2$  y los más bajos de  $f_1$ .

#### 4.2.3. Parámetro $\beta$

La tabla 4.3 muestra el impacto del parámetro  $\beta$ , el cual también aparece en la función de aptitud  $f$ . Estos valores también se obtienen usando las mismas relaciones de preferencia difusas proporcionadas inicialmente por los expertos en el ejemplo anterior y siendo  $\alpha = 0.4$  y  $\gamma = 0.5$ . Como este parámetro establece un compromiso entre la combinación de la coherencia y el consenso,  $\gamma \cdot f_1 + (1 - \gamma) \cdot f_2$ , y la similitud,  $f_3$ , nos centramos en estos valores. Por un lado, cuando  $\beta$  es igual a 0, el algoritmo PSO se centra solamente

Tabla 4.3: Valores de  $0.5 \cdot f_1 + 0.5 \cdot f_2$  y  $f_3$  para distintos valores de  $\beta$ .

|                | $0.5 \cdot f_1 + 0.5 \cdot f_2$ | $f_3$ |
|----------------|---------------------------------|-------|
| $\beta = 0.00$ | 0.815                           | 0.994 |
| $\beta = 0.25$ | 0.821                           | 0.992 |
| $\beta = 0.50$ | 0.841                           | 0.978 |
| $\beta = 0.75$ | 0.917                           | 0.868 |
| $\beta = 1.00$ | 0.921                           | 0.830 |



en maximizar la similitud,  $f_3$ . Como resultado, se consigue el valor más alto de  $f_3$  (0.994, en este caso) y el valor más bajo para la combinación de la coherencia y el consenso. Por otro lado, cuando a  $\beta$  se le asignan valores superiores a 0, el algoritmo PSO se centra tanto en la combinación de la coherencia y el consenso como en la similitud. Por lo tanto, como es de esperar,  $f_3$  alcanza valores más bajos. En particular, cuando  $\beta$  es igual a 1, el algoritmo PSO se centra solamente en la combinación de la coherencia y el consenso. Esto implica que se obtienen los valores más altos para esta combinación (0.921, en este caso) mientras que  $f_3$  alcanza su valor más bajo (0.830, en este caso).



## Capítulo 5

# Conclusiones y vías futuras

En este último capítulo, se presentan las principales conclusiones que hemos extraído después de haber desarrollado este trabajo fin de grado y se indican algunas líneas futuras de desarrollo.

### 5.1. Conclusión

Para empezar, hay que resaltar que la realización de este trabajo fin de grado ha sido enriquecedora gracias a los conocimientos adquiridos. Durante su desarrollo, nos hemos dado cuenta de la cantidad de asignaturas que pueden intervenir en el proceso, junto a la ampliación de conocimientos de forma autodidacta para obtener documentación, solucionar problemas y adecuarse a un cliente y a unos plazos.

Centrándonos en los objetivos del proyecto, vemos que estos se han cumplido. De esta forma, se ha desarrollado e implementado un nuevo sistema de ayuda a la decisión en grupo que, estando basado en el paradigma de la computación granular, permite, por una parte, modelar problemas de toma de decisiones en grupo realizados en entornos multicriterio y heterogéneos y, por otra, mejorar la coherencia y el consenso con el menor ajuste posible en las evaluaciones proporcionadas por los expertos.

Respecto al modelo, este genera gránulos de información en forma de intervalos mediante la asignación de un determinado nivel de granularidad de la información en las evaluaciones proporcionadas por los expertos, las cuales se modelan mediante relaciones de preferencia difusas. A diferencia de los modelos granulares existentes, hemos visto que este nuevo modelo permite alcanzar un equilibrio sólido entre la generación de evaluaciones lo más similares posible a las iniciales proporcionadas por los expertos y la mejora tanto de la coherencia como del consenso en entornos multicriterio y heterogéneos. De esta forma, este nuevo modelo es capaz de tener en cuenta más características asociadas a los procesos de toma de decisiones en grupo que los modelos granulares existentes.

Respecto a la aplicación web que implementa el modelo desarrollado, esta permite automatizar los procesos de toma de decisiones en grupo y facilita a los expertos la expresión de sus evaluaciones desde cualquier lugar y en cualquier momento. Durante el desarrollo del sistema, hemos comprobado que, aunque existen muchos modelos teóricos, no existe un sistema similar que permita, de forma online, diseñar un problema, opinar según tus preferencias y obtener cuál es la mejor solución.

Para finalizar, hay que destacar también que el desarrollo de este trabajo fin de grado ha sacado a relucir la importancia de:

- El uso de metodologías ágiles para hacernos una idea real de cuánto tiempo necesitamos invertir y cómo para poder acabar satisfactoriamente el desarrollo de un sistema.
- Estimar correctamente el tiempo, ya que es la base que sostiene todas las tareas restantes.
- Llevar a cabo un análisis correcto de los comentarios recibidos durante las primeras reuniones con el cliente (director). Esto permite ahorrar tiempo en cambios futuros debido a no haber efectuado un análisis y diseño inicial correcto.

## 5.2. Vías futuras

El desarrollo realizado en este trabajo fin de grado puede continuar de varias maneras, unas relacionadas con el modelo teórico de toma de decisiones en grupo y otras relacionadas con la aplicación web.

En relación con la parte de la aplicación web, se podría realizar un acceso y una navegación adicional para aquellas personas que posean una discapacidad, la posibilidad de ver la página en distintos idiomas o la mejora en aspectos como la seguridad, modularización e interfaz.

Respecto al modelo teórico, se podrían estudiar y trabajar los siguientes aspectos. Por un lado, para el nivel de granularidad de la información, se ha utilizado una distribución uniforme, es decir, el valor de  $\alpha$  es igual para todos los integrantes del grupo. Por tanto, la primera mejora que se podría proponer es una distribución no uniforme, asignando un determinado valor de  $\alpha$  a cada experto. Este valor, podría también optimizarse. Por otro lado, para los cálculos de los operadores de agregación de clase OWA e IOWA, el uso de un cuantificador lingüístico distinto que no penalice de forma tan acentuada a los últimos valores, puesto que si hay, por ejemplo, diez valores, el último valor se aproxima mucho al cero. Por último, trabajar en la búsqueda de una alternativa al uso de matrices para el cálculo de las opiniones, puesto que tener que dar la opinión sobre 10 criterios y 10 alternativas, hace que cada experto tenga que introducir 90 valores para cada criterio y, para este ejemplo, 900 valores para su opinión.

## Apéndice A

# Manual de usuario

A continuación se procederá a explicar los pasos que hay que realizar para instalar y utilizar el sistema.

### A.1. Instalación

La instalación la realizaremos para el sistema operativo Windows. Será necesario instalar previamente WSL 2 backend. Luego, en la página web de Docker Desktop for Windows<sup>1</sup> se puede descargar e instalar el ejecutable del docker. Una vez finalizado este proceso, tendremos que descomprimir el fichero *.zip* que contiene el sistema. Dentro de la carpeta *docker-compose*, abrimos la consola de Windows y escribimos las siguientes sentencias:

```
1 docker-compose build
2 docker-compose up
```

Una vez que hayamos realizado estos pasos, ya tendremos funcionando el docker y podremos entrar a nuestro servidor. Para ello, entramos en un navegador y escribimos *localhost*, donde nos aparecerá la pantalla principal (ver figura A.1).

### A.2. Administrador

Para acceder como administrador será necesario introducir las siguientes credenciales:

- Usuario: admin
- Contraseña: Admin1234

Accederemos de esta forma a la página inicial del sistema (ver figura A.2). Desde aquí, podremos acceder a:

---

<sup>1</sup><https://hub.docker.com/editions/community/docker-ce-desktop-windows/>



Figura A.1: Página principal del sistema.



Figura A.2: Página principal del administrador.

- Sobre mí. Página donde cuento quién soy yo.
- Crear problema. Añadir un problema al sistema.
- Calcular problema. Resolver un problema que ha recibido las opiniones de todos los usuarios.
- Problemas resueltos. Visualizar los problemas que han sido calculados, y posteriormente visualizar sus soluciones.
- Opciones sobre problemas:
  - Modificar un problema. Cambiar parámetros para poder realizar estudios con otros valores.
  - Rehacer un problema. Permitir que un problema pueda ser calculado de nuevo.
  - Borrar problema. Eliminar un problema del sistema.

En la parte del nombre, hay un menú desplegable donde se podrá:

- Añadir expertos al sistema.
- Gestionar roles a los usuarios.
- Verificar usuarios que se han dado de alta de forma externa.

### A.3. Usuario

Para acceder como usuario, será necesario introducir su usuario y contraseña. Una vez realizada la identificación, aparecerá en la página inicial del usuario (experto) (ver figura A.3).



Figura A.3: Página principal de experto.

Desde aquí, podremos acceder a:

- Opinar. Proporcionar las valoraciones sobre un problema.
- Problemas resueltos. Visualizar los problemas que han sido calculados, y posteriormente visualizar sus soluciones.

### A.4. Algunos ejemplos

A continuación, se detallarán algunas de las acciones que se pueden realizar con el sistema.

- A la hora de opinar, se accederá a la página de opinar (ver figura A.4), en donde se tendrán que rellenar las valoraciones (relaciones de preferencia difusas) por cada uno de los criterios para el problema que ha seleccionado.
- A la hora de visualizar las soluciones, se accederá a la página de visualización de resultados del problema (ver figura A.5), en donde se visualizarán las valoraciones finales asignadas a cada alternativa, las consistencias iniciales de cada experto y el consenso inicial, las variables internas y la gráfica que describe cómo han ido evolucionando el consenso alcanzado, la consistencia y la similitud.

UNIVERSIDAD DE GRANADA Inicio Opinir Problemas Resueltos Juanca

### Rellene su opinión

Problema con el nombre '**Puente en Almerimar**'

Criterio **Tiempo** con peso : 0.4

|            | Golko | Playeo | Lol | ChucuChucu |
|------------|-------|--------|-----|------------|
| Golko      | -     |        |     |            |
| Playeo     |       | -      |     |            |
| Lol        |       |        | -   |            |
| ChucuChucu |       |        |     | -          |

Criterio **Calor** con peso : 0.7

|            | Golko | Playeo | Lol | ChucuChucu |
|------------|-------|--------|-----|------------|
| Golko      | -     |        |     |            |
| Playeo     |       | -      |     |            |
| Lol        |       |        | -   |            |
| ChucuChucu |       |        |     | -          |

Figura A.4: Página de opinar.

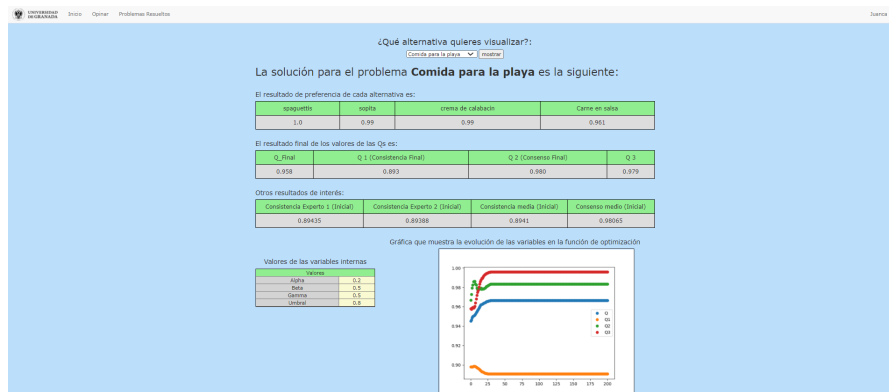


Figura A.5: Página de visualización de las soluciones.

- A la hora de crear un problema, se tendrán que rellenar varios formularios. Estos son: introducción del número de expertos, criterios y alternativas (ver figura A.6), añadir todos los valores intermedios: nombre, descripción, pesos de los criterios, nombres de las alternativas, etc. (ver figura A.7) y, por último, indicar el peso de cada experto con respecto a su criterio (ver figura A.8).



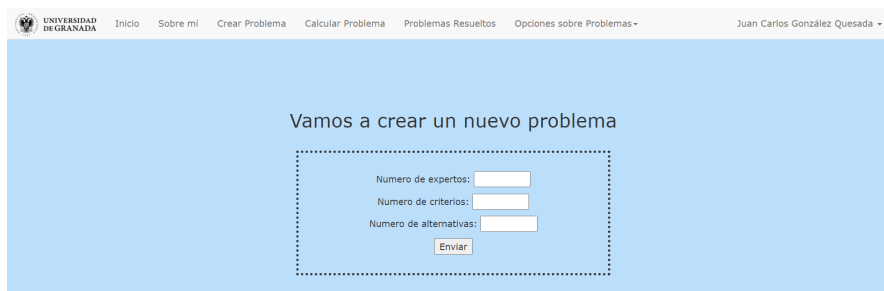


Figura A.6: Página de creación de problemas. Parte 1.

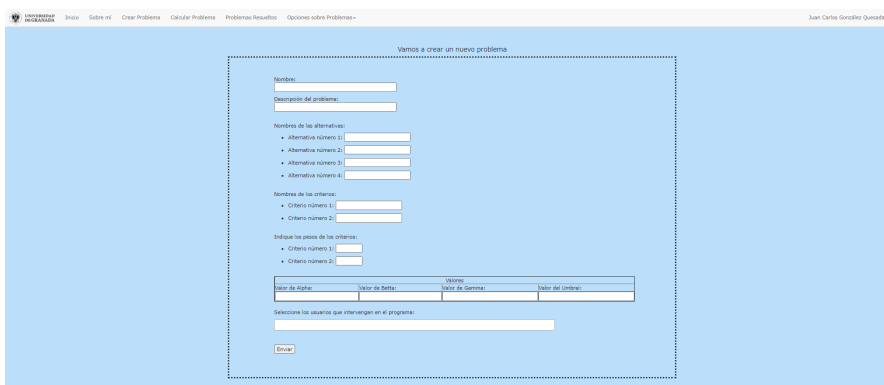


Figura A.7: Página de creación de problemas. Parte 2.

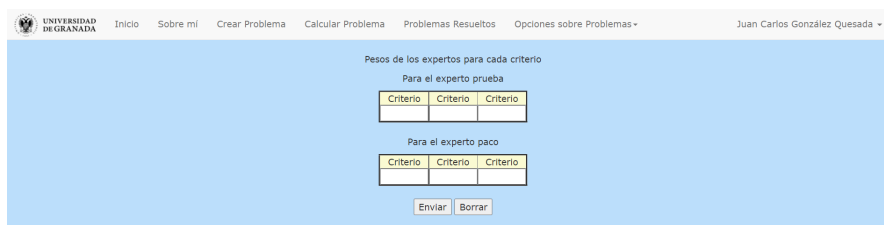


Figura A.8: Página de creación de problemas. Parte 3.

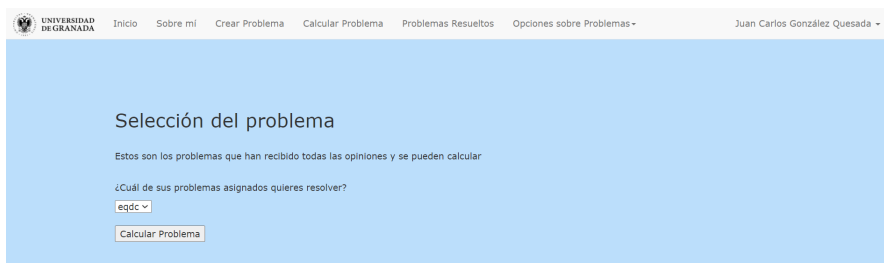


Figura A.9: Selección de un problema para calcular su solución.

Finalmente, otros ejemplos de páginas que podemos encontrar en el sistema son:

- La selección de un problema para calcular su solución (ver figura A.9), en donde se mostrarán todos los problemas que han recibido todas las opiniones y que, por tanto, se pueden calcular sus soluciones.
- La visualización de la solución de un problema (ver figura A.10), en donde, además de los problemas que ya hay resueltos y cuya solución se puede ver, se incluyen aquellos problemas que no han sido calculados o que se encuentran a la espera de que los expertos en su conjunto proporcionen todas sus valoraciones sobre los distintos criterios.

UNIVERSIDAD DE GRANADA Inicio Sobre mí Crear Problema Calcular Problema Problemas Resueltos Opciones sobre Problemas Juan Carlos González Quesada

¿De qué problema quieres visualizar las soluciones?

Que comida basura comer

Visualizar solución

Problemas en espera

Estos son los problemas que ya has opinado y están a la espera de otros expertos o de ser calculados

Juegos de mesa Musica piano Puente en Almería

Comida para la playa 1 Final rtehyrthh

Figura A.10: Selección de un problema para visualizarlo.

## Apéndice B

# Conocimientos utilizados

En relación con los conocimientos necesarios para llevar a cabo el desarrollo de este trabajo fin de grado, han sido de gran ayuda, entre otras, las siguientes asignaturas:

- Cálculo y Álgebra Lineal y Estructuras Matemáticas. Con los conocimientos adquiridos en estas dos asignaturas (visión espacial de matrices, multiplicación de vectores y el uso de funciones) se ha podido realizar de forma más rápida la parte matemática del proyecto.
- Programación y Diseño Orientado a Objetos. Uso de clases y herencia para descomponer los sistemas en módulos en lugar de por funciones, poder representar las clases y sus relaciones y poder aplicar en la resolución de problemas conceptos de los principios de diseño con el objetivo de especificar jerarquías de clases. Conocimiento sobre patrones de diseño.
- Fundamentos de Ingeniería del Software. Capacidad para concebir y desarrollar sistemas o arquitecturas informáticas y mantener sistemas o servicios empleando métodos de la ingeniería del software.
- Fundamentos de Bases de Datos. Poder plantear los diseños lógicos y físicos más adecuados en función de los requerimientos del sistema. Crear esquemas de bases de datos usando un lenguaje de definición de datos, incorporando restricciones de integridad y sabiendo realizar consultas adecuadas evaluando el sistema.
- Programación Web. Saber desarrollar aplicaciones que se ejecutan en el ámbito de un servidor. Conocer los mecanismos de interacción del usuario en las aplicaciones web y ser capaz de realizar análisis y diseño de aplicaciones web.
- Diseño y Desarrollo de Sistemas de Información. Capacidad para diseñar y desarrollar sistemas de información informáticos asegurando

fiabilidad, seguridad, etc. Conocimientos sobre características, funcionalidades y estructura de bases de datos y aplicación de sistemas para almacenar aplicaciones basadas en web.

- Metaheurísticas. Competencias para determinar la complejidad computacional de un cierto problema, tener conocimiento de diferentes métodos algorítmicos capaces de resolverlo y desarrollar e implementar aquel que proporcione mejor rendimiento según los requisitos fijados.

# Bibliografía

- [1] D. J. Anderson. *Agile Management for Software Engineering: Applying the Theory of Constraints for Business Results*. Prentice Hall, 2003.
- [2] A. Bargiela y W. Pedrycz. *Granular Computing: An Introduction*. Dordrecht: Kluwer Academic Publishers, 2003.
- [3] R. E. Bellman y L. A. Zadeh. “Decision-making in a fuzzy environment”. En: *Management Science* 17.4 (1970), págs. 141-164.
- [4] F. Blanco-Mesa, E. León-Castro y J. M. Merigó. “A bibliometric analysis of aggregation operators”. En: *Applied Soft Computing* 81 (2019), pág. 105488.
- [5] D. Bouyssou y col. *Evaluation and Decision Models: A Critical Perspective*. New York: Springer, 2000.
- [6] H. Bustince y col. “A historical account of types of fuzzy sets and their relationships”. En: *IEEE Transactions on Fuzzy Systems* 24.1 (2016), págs. 179-194.
- [7] C. T. Butler y A. Rothstein. *On Conflict and Consensus: A Handbook on Formal Consensus Decisionmaking*. Takoma Park: Food Not Bombs Publishing, 2006.
- [8] F. J. Cabrerizo y col. “A distribution of information granularity to deal with inconsistency in multi-criteria and heterogeneous group decision making”. En: *54th Hawaii International Conference on System Sciences*. Hawaii, USA, ene. de 2021, págs. 1908-1917.
- [9] F. J. Cabrerizo y col. “A granular consensus approach with minimum adjustment for multi-criteria group decision making”. En: *2020 IEEE International Conference on Systems, Man, and Cybernetics*. Toronto, Canada, oct. de 2020, págs. 76-81.
- [10] F. J. Cabrerizo y col. “An improvement of multiplicative consistency of reciprocal preference relations: A framework of Granular Computing”. En: *2017 IEEE International Conference on Systems, Man, and Cybernetics*. Banff, Canada, oct. de 2017, págs. 1262-1267.

- [11] F. J. Cabrerizo y col. "Building consensus in group decision making with an allocation of information granularity". En: *Fuzzy Sets and Systems* 255 (2014), págs. 115-127.
- [12] F. J. Cabrerizo y col. "Improving consensus in group decision making with intuitionistic reciprocal preference relations: A granular computing approach". En: *2018 IEEE International Conference on Systems, Man, and Cybernetics*. Miyazaki, Japan, oct. de 2018, págs. 1471-1476.
- [13] S. J. Chen y C. L. Hwang. *Fuzzy Multiple Attribute Decision-Making: Methods and Applications*. New York: Springer, 1992.
- [14] V. Cutello y J. Montero. "Fuzzy rationality measures". En: *Fuzzy Sets and Systems* 62.1 (1994), págs. 39-54.
- [15] R. C. Eberhart y J. Kennedy. "A new optimizer using particle swarm theory". En: *6th International Symposium on Micro Machine and Human Science*. Nagoya, Japan, oct. de 1995, págs. 39-43.
- [16] J. C. Fodor y M. R. Roubens. *Fuzzy Preference Modelling and Multi-criteria Decision Support*. Dordrecht: Springer, 1994.
- [17] E. Gamma y col. *Design Patterns: Elements of Reusable Object-Oriented Software*. Addison-Wesley, 1994.
- [18] J. L. García-Lapresta y J. Montero. "Consistency in preference modeling". En: *Modern Information Processing: From Theory to Applications*. Ed. por B. Bouchon-Meunier, G. Coletti y R. R. Yager. Amsterdam: Elsevier, 2006, págs. 87-97.
- [19] P. Gelderloos. *Consensus: A New Handbook for Grassroots Social, Political, and Environmental Groups*. Tucson, AZ: See Sharp Press, 2006.
- [20] M. Gupta. "Consensus building process in group decision making—An adaptive procedure based on group dynamics". En: *IEEE Transactions on Fuzzy Systems* 26.4 (2018), págs. 1923-1933.
- [21] R. Heradio y col. "Rough sets: A bibliometric analysis from 2014 to 2018". En: *53rd Hawaii International Conference on System Sciences*. Hawaii, USA, ene. de 2020, págs. 1747-1756.
- [22] F. Herrera, E. Herrera-Viedma y J. L. Verdegay. "Linguistic measures based on fuzzy coincidence for reaching consensus in group decision making". En: *International Journal of Approximate Reasoning* 16.3-4 (1997), págs. 309-334.
- [23] E. Herrera-Viedma y col. "A review of soft consensus models in a fuzzy environment". En: *Information Fusion* 17 (2014), págs. 4-13.
- [24] E. Herrera-Viedma y col. "Group decision-making model with incomplete fuzzy preference relations based on additive consistency". En: *IEEE Transactions on Systems, Man, and Cybernetics—Part B: Cybernetics* 37.1 (2007), págs. 176-189.

- [25] E. Herrera-Viedma y col. "Revisiting fuzzy and linguistic decision making: Scenarios and challenges for making wiser decisions in a better way". En: *IEEE Transactions on Systems, Man, and Cybernetics: Systems* 51.1 (2021), págs. 191-208.
- [26] K. Hirota. "Concepts of probabilistic sets". En: *Fuzzy Sets and Systems* 5.1 (1987), págs. 31-46.
- [27] C.-L. Hwang y M.-J. Lin. *Group Decision Making Under Multiple Criteria: Methods and Applications*. Berlin: Springer-Verlag, 1987.
- [28] Servicio de Informática. Universidad de Alicante. *Modelo vista controlador (MVC)*. URL: <https://si.ua.es/es/documentacion/asp-net-mvc-3/1-dia/modelo-vista-controlador-mvc.html>.
- [29] J. Kennedy y R. C. Eberhart. "Particle swarm optimization". En: *1995 IEEE International Conference on Neural Networks*. Perth, Australia, nov. de 1995, págs. 1942-1948.
- [30] E. Koksalmis y O. Kabak. "Deriving decision makers' weights in group decision making: An overview of objective methods". En: *Information Fusion* 49 (2019), págs. 146-160.
- [31] C.-C. Li y col. "An overview on managing additive consistency of reciprocal preference relations for consistency-driven decision making and fusion: Taxonomy and future directions". En: *Information Fusion* 52 (2019), págs. 143-156.
- [32] F. Liu, Y. Wu y W. Pedrycz. "A modified consensus model in group decision making with an allocation of information granularity". En: *IEEE Transactions on Fuzzy Systems* 26.5 (2018), págs. 3182-3187.
- [33] S. Liu y col. "Development of information granules of higher type and their applications to granular models of time series". En: *Engineering Applications of Artificial Intelligence* 71 (2018), págs. 60-72.
- [34] I. Millet. "The effectiveness of alternative preference elicitation methods in the analytic hierarchy process". En: *Journal of Multi-Criteria Decision Analysis* 6.1 (1997), págs. 41-51.
- [35] J. Montero. "Aggregation of fuzzy opinions in a non-homogeneous group". En: *Fuzzy Sets and Systems* 25.1 (1988), págs. 15-20.
- [36] R. Moore. *Interval Analysis*. Englewood Cliffs, NJ: Prentice-Hall, 1966.
- [37] J. A. Morente-Molinera y col. "A dynamic group decision making process for high number of alternatives using hesitant Fuzzy Ontologies and sentiment analysis". En: *Knowledge-Based Systems* 195 (2020), pág. 105657.

- [38] J. A. Morente-Molinera y col. "A novel multi-criteria group decision-making method for heterogeneous and dynamic contexts using multi-granular fuzzy linguistic modelling and consensus measures". En: *Information Fusion* 53 (2020), págs. 240-250.
- [39] S. A. Orlovsky. "Decision-making with a fuzzy preference relation". En: *Fuzzy Sets and Systems* 1.3 (1978), págs. 155-167.
- [40] W. Pedrycz. "Allocation of information granularity in optimization and decision-making models: Towards building the foundations of Granular Computing". En: *European Journal of Operational Research* 232.1 (2014), págs. 137-145.
- [41] W. Pedrycz. "The principle of justifiable granularity and an optimization of information granularity allocation as fundamentals of granular computing". En: *Journal of Information Processing Systems* 7.3 (2011), págs. 397-412.
- [42] W. Pedrycz y M. Song. "Analytic hierarchy process (AHP) in group decision making and its optimization with an allocation of information granularity". En: *IEEE Transactions on Fuzzy Systems* 19.3 (2011), págs. 527-539.
- [43] R. Poli, J. Kennedy y T. Blackwell. "Particle swarm optimization". En: *Swarm Intelligence* 1.1 (2007), págs. 33-57.
- [44] M. Roubens. "Fuzzy sets and decision analysis". En: *Fuzzy Sets and Systems* 90.2 (1997), págs. 199-206.
- [45] M. A. Sánchez. "Patrón Estrategia (Strategy Pattern)". En: (2020). URL: <https://medium.com/all-you-need-is-clean-code/patr%C3%B3n-estrategia-strategy-pattern-654c6e9d2abe>.
- [46] Y. Shi y R. C. Eberhart. "A modified particle swarm optimizer". En: *IEEE International Conference on Evolutionary Computation*. Ancho-rage, Estados Unidos, mayo de 1998, págs. 69-73.
- [47] M. F. Smith. *Software Prototyping: Adoption, Practice and Management*. London, UK: McGraw-Hill, 1991.
- [48] X. Tan, J. Zhu e Y. Zhang. "A consensus reaching process with quantum subjective adjustment in linguistic group decision making". En: *Information Sciences* 533 (2020), págs. 150-168.
- [49] Telefónica. *Twig, el motor de plantillas para PHP que separa el código HTML*. Jun. de 2014. URL: <https://www.acens.com/wp-content/images/2014/06/twig-plantillas-wp-acens.pdf>.
- [50] D. Wang, D. Tan y L. Liu. "Particle swarm optimization algorithm: An overview". En: *Soft Computing* 22.2 (2018), págs. 387-408.
- [51] R. R. Yager. "Families of OWA operators". En: *Fuzzy Sets and Systems* 59.2 (1993), págs. 125-148.



- [52] R. R. Yager. "On ordered weighted averaging aggregation operators in multicriteria decisionmaking". En: *IEEE Transactions on Systems, Man, and Cybernetics* 18.1 (1988), págs. 183-190.
- [53] R. R. Yager. "Using stress functions to obtain OWA operators". En: *IEEE Transactions on Fuzzy Systems* 15.6 (2007), págs. 1122-1129.
- [54] R. R. Yager y D. P. Filev. "Induced ordered weighted averaging operators". En: *IEEE Transactions on Systems, Man, and Cybernetics - Part B: Cybernetics* 29.2 (1999), págs. 141-150.
- [55] M. C. Yang. "Consensus and single leader decision-making in teams using structured design methods". En: *Design Studies* 31.4 (2010), págs. 345-362.
- [56] L. A. Zadeh. "A computational approach to fuzzy quantifiers in natural languages". En: *Computers & Mathematics with Applications* 9.1 (1983), págs. 149-184.
- [57] L. A. Zadeh. "Fuzzy sets". En: *Information and Control* 8.3 (1965), págs. 338-353.
- [58] L. A. Zadeh. "Towards a theory of fuzzy information granulation and its centrality in human reasoning and fuzzy logic". En: *Fuzzy Sets and Systems* 90.2 (1997), págs. 111-127.
- [59] B. Zhang, Y. C. Dong e Y. Xu. "Multiple attribute consensus rules with minimum adjustments to support consensus reaching". En: *Knowledge-Based Systems* 67 (2014), págs. 35-48.
- [60] X. R. Zhao e Y. Yao. "Three-way fuzzy partitions defined by shadowed sets". En: *Information Sciences* 497 (2019), págs. 23-37.



