



Trabajo de Fin de Grado

Software de Ayuda a la Decisión mediante el Proceso Analítico Jerárquico

*Decision Support Software through Analytic Hierarchy
Process*

Sergio Leopoldo Benítez Delgado

La Laguna, 7 de julio de 2022

D. **Christopher Expósito Izquierdo**, con N.I.F. 78.851.649-J profesor Ayudante Doctor adscrito al Departamento Ingeniería Informática y de Sistemas de la Universidad de La Laguna como tutor

D. **Israel López Plata**, con N.I.F. 42.193.801-W profesor Ayudante Doctor adscrito al Departamento Ingeniería Informática y de Sistemas de la Universidad de La Laguna como cotutor

C E R T I F I C A N

Que la presente memoria titulada:

"Software de Ayuda a la Decisión mediante el Proceso Analítico Jerárquico"

ha sido realizada bajo su dirección por D. **Sergio Leopoldo Benítez Delgado**, con N.I.F. 43.486.602-J.

Y para que así conste, en cumplimiento de la legislación vigente y a los efectos oportunos firman la presente en La Laguna a 7 de julio de 2022

Agradecimientos

Me gustaría comenzar agradeciendo a la oportunidad brindada por la Universidad de La Laguna y, en especial, al conjunto de profesores, que durante estos cuatro años, han sabido transmitir los conocimientos necesarios para formarme tanto profesionalmente como personalmente.

También agradecer a mi tutor D. Christopher Expósito Izquierdo y cotutor D. Israel López Plata, por el excelente trato recibido en estos casi cuatro meses, donde en todo momento han estado a disposición para la resolución y consulta de cualquier tipo de duda. Logrando así un entorno de trabajo muy bueno para el desarrollo de este proyecto.

Por último, pero no menos importante, agradecer a mi familia y a mi pareja, uno de los pilares más importantes en el transcurso de la carrera, por apoyarme en este largo proceso, y sobre todo en estos últimos meses en los que se acerca el final de esta dura, pero bonita etapa.

Resumen

A día de hoy, a casi todas las empresas y personas se les ha presentado la difícil tarea de tomar una determinada decisión, cuyo resultado tendrá multitud de consecuencias. Es por ello que a todas estas les ha surgido, alguna vez, la necesidad de disponer de una ayuda a la hora de tomar dicha decisión. Es aquí, donde entra en juego el software de ayuda a la toma de decisiones, desarrollado en este proyecto como una aplicación web, gratuita, y accesible para toda persona que disponga de conexión a internet.

A la hora de la toma de la decisión, en la aplicación se ha implementado la técnica del Proceso Analítico Jerárquico, un método cuantitativo para la toma de decisiones multicriterio que permite generar escalas de prioridades basándose en juicios de expertos, manifestados a través de comparaciones por pares mediante una escala de preferencia.

En cuanto al desarrollo, este ha sido etiquetado bajo el concepto de desarrollo de una aplicación Full Stack, abarcando el desarrollo de un back-end que actuará como API-REST y un front-end en el que implementar la interfaz de usuario. Esta API-REST será la encargada de recibir las peticiones desde la interfaz de usuario y validarlas para su posterior introducción a la base de datos, o comunicación con un módulo externo encargado de la resolución del Proceso Analítico Jerárquico.

Palabras clave: Toma de decisiones, Sistemas de ayuda a la decisión, Proceso Analítico Jerárquico

Abstract

Actually, for most of companies and people the difficult task of taking a particular decision have been presented, whose result will have multitude of consequences. That is why all of them sometimes needed a help when making such decisions. Here is where a decision support software is necessary, developed in this project as a web application, free, and accessible to anyone with an internet connection.

The application implements an Analytic Hierarchy Process technique to take the decisions, which is a quantitative method for multi-criteria decision making that allows the generation of priority scales based on expert judgements, expressed through pairwise comparisons using a preference scale.

In terms of development, this has been categorized under the concept of Full Stack application development. This includes the development of a back-end published as an API-REST and a front-end, in which the user interface is presented. This API-REST will be in charge of receiving the requests from the user interface and validate them before insert the data into the database, or communicate with the external module with Analytic Hierarchy Process algorithm.

Keywords: *Decision making, Decision suport system, Analytic Hierarchy Process*

Índice general

1. Introducción	1
1.1. Justificación del proyecto	2
1.2. Objetivos	3
2. Motivación y Estado del arte	5
2.1. Teoría de la decisión	5
2.2. Decisión multicriterio	6
2.3. Proceso Analítico Jerárquico	7
2.3.1. Fundamentos Psicológicos	8
2.3.2. Fundamentos matemáticos	9
2.3.3. Fases para la resolución del AHP	9
2.3.4. Contrastes empíricos	13
2.3.5. Aplicaciones para la toma de decisiones	15
3. Tecnologías y arquitectura de software	18
3.1. Tecnologías	18
3.1.1. MongoDB	18
3.1.2. Spring	19
3.1.3. Vue.js	19
3.1.4. Resto de tecnologías	20
3.2. Arquitectura	21
3.3. Composición de la aplicación	23
3.3.1. Back-end	24
3.3.2. Front-end	26
3.3.3. AHP	28
4. Funcionamiento	29
4.1. Despliegue	29
4.1.1. Despliegue de manera local	29
4.1.2. Despliegue haciendo uso de Docker	29
4.2. Funcionamiento	30
4.2.1. Home	30
4.2.2. Escenarios	33
4.2.3. Expertos	35
4.2.4. Lanzamiento del AHP	38
5. Experimentación	41
5.1. Procedimiento para selección del jugador	41
5.1.1. Selección de criterios	41

5.1.2. Selección de alternativas	42
5.2. Valoración de los expertos	42
5.3. Selección de opiniones	44
6. Presupuesto	48
6.1. Infraestructuras tecnológicas	48
6.2. Recursos humanos	48
6.3. Coste total	49
7. Conclusiones	50
7.1. Líneas Futuras	50
8. Conclusions	52
8.1. Future Lines	52
Bibliografía	53

Índice de figuras

1.1. Estructura jerárquica del AHP.	2
2.1. Estructura jerárquica del caso	14
2.2. Vector de prioridades resultante	15
2.3. Interfaz gráfica de SpiceLogic AHP	16
3.1. Estructura de la aplicación	21
3.2. Flujo de ejecución del AHP.	23
3.3. Estructura directorios del back-end	24
3.4. Estructura directorios del front-end	26
3.5. Diagrama de casos de uso del funcionamiento del front-end.	28
3.6. Estructura del módulo AHP	28
4.1. Pantalla de inicio.	31
4.2. Pantalla para el manejo de criterios.	32
4.3. Pantalla para el mostrar, modificar, eliminar o añadir un criterio.	32
4.4. Pantalla para el manejo de árboles.	33
4.5. Pantalla para la visualización gráfica de los árboles.	34
4.6. Pantalla para la definición de los expertos.	36
4.7. Pantalla para la realización de una opinión.	37
4.8. Pantalla para la visualización del árbol pesado.	37
4.9. Pantalla para la selección de opiniones y pesos de los expertos.	38
4.10 Componente para la visualización de la solución obtenida.	39
4.11 Informe PDF con la información del proceso de toma de decisión.	40
5.1. Estructura jerárquica para la selección de un delantero.	41
5.2. Estructura jerárquica ponderada por el presidente.	43
5.3. Selección de opiniones y ajuste del peso de los expertos.	45
5.4. Solución al proceso de selección de un jugador.	46
5.5. Informe PDF con la información del proceso de selección de un delantero.	47

Índice de tablas

2.1. Escala fundamental de comparación por pares (Saaty, 1980)	8
2.2. Porcentajes máximos del ratio de consistencia CR	11
2.3. Índice aleatorio RI	11
2.4. Vectores de prioridades de los criterios	12
2.5. Vector propio de un determinado criterio	13
2.6. Conjunto de vectores propios	13
5.1. Jugadores seleccionados (Alternativas).	42
5.2. Matriz comparaciones pareadas (Criterios 1º Nivel).	43
5.3. Matriz comparaciones pareadas (Criterios 2º Nivel).	43
5.4. Matriz comparaciones pareadas (Criterios 3º Nivel).	43
5.5. Vector de prioridades asignadas por el presidente.	44
6.1. Coste recursos materiales	48
6.2. Coste recursos humanos	49
6.3. Coste recursos humanos	49

Capítulo 1

Introducción

A lo largo del proceso productivo de una empresa, uno de los puntos fundamentales y a la vez más complejos es la toma de decisiones. En múltiples ocasiones, tomar una decisión de forma correcta puede suponer un beneficio crítico para la empresa, pero incluso las decisiones más triviales, en las cuales la decisión se simplifica a la elección de una alternativa u otra, pueden afectar notablemente al funcionamiento normal de la misma.

Es por ello que la toma de decisiones es un punto de inflexión en multitud de ámbitos de la vida, uno de ellos podría ser el ámbito empresarial, donde la competitividad entre empresas es un hecho que nadie duda. Independientemente del sector al que pertenezca cada empresa, estas siempre buscan obtener el mayor beneficio, siendo lo más eficientes posibles. Es en este punto donde las empresas deben tomar buenas decisiones, las cuales les permitan competir contra el resto de empresas.

En muchos ámbitos de la vida, las decisiones que se deben tomar son muy complejas y de vital importancia, teniendo estas unas consecuencias inmediatas sobre la entidad o persona encargada de tomar dicha decisión. Un ejemplo de este tipo de decisiones podría ser la elección de la carrera universitaria a estudiar, tal y como se expone en [1]. Esta es una decisión de gran trascendencia, en la que intervienen diversos factores de forma directa, como el prestigio social de la carrera, salidas laborales, coeficiente de tradicionalidad familiar y las necesidades vitales.

Es por ello que cada vez existen más aplicaciones focalizadas en ayudar al usuario a tomar decisiones correctas en función de los datos disponibles, llamadas sistemas de ayuda a la decisión.

Por normal general, un sistema de ayuda a la decisión suele estar construido para resolver una problemática en concreto, como por ejemplo el utilizado por un GPS a la hora de seleccionar la mejor ruta entre dos puntos, valorando de manera automática criterios específicos como la distancia, el tráfico, tipo de vía, etc. Esto implica que su desarrollo es específico y difícilmente aplicable en otros contextos ajenos al de la empresa que lo utiliza [9]. Por ello, la creación de un sistema de ayuda a la decisión genérico permitirá la toma de decisiones en ámbitos muy diversos.

Una de las técnicas más frecuentes para la ayuda a la decisión de forma genérica es el Proceso Analítico Jerárquico (AHP, por sus siglas en inglés) [10]. Es un método de decisión

multicriterio que ayuda a seleccionar entre distintas alternativas en función de una serie de criterios o variables de selección, normalmente jerarquizadas y que suelen entrar en conflicto entre sí. Esta estructura jerarquizada, como la representada gráficamente en la Figura 1.1, habitualmente en forma de árbol, describe de arriba a abajo las siguientes partes:

- *Objetivo global o final.* Nodo raíz de la estructura, y cuyo nombre debe expresar claramente el problema a resolver.
- *Criterios y subcriterios.* Grupo de factores o características que relacionan las alternativas con el objetivo.
- *Alternativas.* Conjunto factible de soluciones que dan solución al objetivo final planteado.

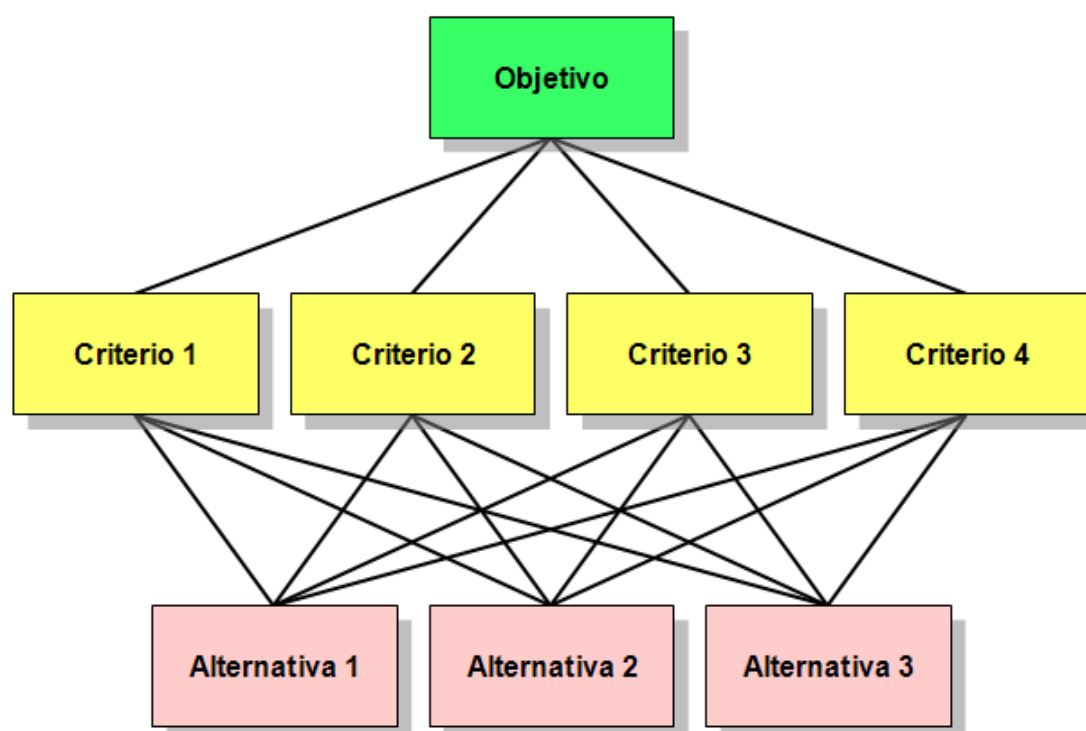


Figura 1.1: Estructura jerárquica del AHP¹

1.1. Justificación del proyecto

En múltiples ocasiones, tomar una decisión correcta puede suponer un beneficio crítico, por lo que el desarrollo de este software viene a cubrir la necesidad que tienen muchas personas y organizaciones a la hora de disponer de una ayuda a la hora de tomar la decisión correcta.

¹<https://upload.wikimedia.org/wikipedia/commons/2/20/AHPHierarchy1Spanish.png>

Es por ello que cada día existen más aplicaciones focalizadas en ayudar al usuario a tomar las decisiones correctas, en función de los datos disponibles. Esto es lo que se conoce como un sistema de ayuda a la decisión, siendo esto lo que se trata de desarrollar a lo largo de este Trabajo Fin de Grado.

Estos sistemas de ayuda a la decisión han sido creados para satisfacer las necesidades que le surgen a muchas personas y empresas en el proceso de toma de decisiones, como se ha comentado anteriormente. Casi todos estos sistemas son desarrollados de manera meramente matemática, mediante la aplicación de herramientas para procesamiento de datos, capaz de arrojar como resultado la decisión óptima. Además, casi todos estos sistemas se encuentran implementados para resolver casos muy acotados de toma de decisión. En este Trabajo de Fin de Grado se persigue desarrollar un sistema de ayuda a la decisión en forma de plataforma web, pero con ámbito genérico, permitiendo que aquellos problemas relacionados con la toma de decisiones que puedan ser descompuestos de manera jerárquica, permitiendo identificar sus objetivos, sus criterios y sus alternativas, puedan ser resueltos de manera sencilla por el usuario. Además, el resultado de dicha decisión, podrá ser determinado por un conjunto de expertos ponderados según la importancia de los mismos en el proceso de la toma de dicha decisión. Dicho resultado podrá ser observado en la interfaz gráfica de dicha aplicación, pudiendo además obtener informes, estadísticas y resultados por cada uno de los expertos.

En cuanto a la técnica usada para la toma de dicha decisión, una de las más frecuentes es el AHP el cual es un método cuantitativo para la toma de decisiones multicriterio que permite generar escalas de prioridades basándose en juicios de expertos manifestados a través de comparaciones por pares mediante una escala de preferencia.

1.2. Objetivos

El objetivo general de este proyecto es desarrollar una herramienta de apoyo, en forma de plataforma web, para el proceso de definición y evaluación de alternativas asociadas a un determinado árbol de decisión, con el objetivo de obtener como resultado la mejor alternativa de acuerdo a un conjunto de expertos. De esta manera, se crea un sistema de ayuda a la toma de decisiones mediante la aplicación de la técnica del Proceso Analítico Jerárquico.

En cuanto a los objetivos más específicos a conseguir en el desarrollo de este software, podemos listar algunos como:

- *Creación del anteproyecto.* Elaboración de un documento donde se expone una introducción, antecedentes y estado actual, así como un listado de tareas a realizar y el plan de trabajo establecido para la realización de este Trabajo Fin de Grado.
- *Creación y configuración de la estructura principal del proyecto.* Creación de los directorios en los que se alojará todo el código, distribuido en los directorios *front-end*, *back-end* y *AHP*
- *Creación de la API-REST.* Desarrollo de un *back-end* capaz de comunicar la interfaz

de usuario con una base de datos como lo es MongoDB, haciendo uso del lenguaje de programación Java, y su framework Spring.

- *Creación de la interfaz de usuario.* Desarrollo de un *front-end* desde el que el usuario pueda interactuar con la aplicación. Para el desarrollo de esta interfaz se hizo uso del framework VueJS y Vuetify.
- *Implementación e integración en la aplicación de la técnica AHP.* Implementación de un módulo, aparte de los dos anteriores, capaz de resolver la técnica del Proceso Analítico Jerárquico.
- *Funcionalidad de importación de datos.* Permitir al usuario que introduzca los datos de entrada sobre los que tomar una decisión desde fichero que siga un formato determinado.
- *Creación de una pantalla de resultados.* Desarrollo de una pantalla en la que se muestre de manera gráfica y detallada la salida obtenida al aplicar el AHP.
- *Generación de informes sobre los resultados.* Implementación de una opción en la interfaz de usuario que permita obtener las estadísticas y análisis de datos en PDF.
- *Experimentación.* Validar el correcto funcionamiento de la aplicación, mediante la experimentación con un caso real de toma de decisiones.
- *Redacción de la memoria.* Se describe el Trabajo Fin de Grado desarrollado.

Capítulo 2

Motivación y Estado del arte

2.1. Teoría de la decisión

La teoría de la decisión se encarga de analizar como una persona elige una acción que, de entre un conjunto de posibles acciones, le conduce al mejor resultado dadas sus preferencias. Desde una perspectiva formal, lo único que importa es que dichas preferencias satisfagan ciertos criterios básicos de consistencia lógica, entre los que cabe destacar por su importancia los siguientes, donde x , y y z representan diferentes alternativas:

- *Transitividad.* Para todo x , y , y z , si x es preferida estrictamente a y e y es preferida estrictamente a z , x será preferida a z . Cumplir con este requisito asegura que no se tome una decisión incoherente, como sería la producida al elegir en un principio que se prefiere x frente a y , e y frente a z , para posteriormente elegir z frente a x .
- *Compleitud.* Para todo x y todo y , o bien x es preferida a y , o y es preferida a x , o el individuo es indiferente entre ellas. Cumplir con este requisito exige que el experto compare entre sí todas sus opciones y se decida por una de ellas o manifieste su indiferencia.
- *Asimetría.* Si x es preferida estrictamente a y , y no es preferida estrictamente a x .
- *Simetría de la indiferencia.* Para todo x e y , si x es indiferente a y , y es indiferente a x . Cumplir con estos dos últimos requisitos implica poder establecer una escala, pudiendo ordenar de los menos a los más preferidos.

El incumplimiento de estos requisitos producirá la imposibilidad de saber lo que la persona quiere, no pudiendo ordenar sus preferencias, y la teoría de la decisión considerará que dicha persona no elige racionalmente [4].

El proceso de toma de decisión puede entenderse como la elección de lo *mejor* entre lo *posible*. Dependiendo de qué se defina como lo mejor y qué es lo posible, se estará enfrente de distintas situaciones de decisión[15].

Una de las situaciones de decisión es la optimización clásica, la cual tiene como característica principal que lo mejor está claramente determinado y que las posibilidades no vienen expresadas explícitamente, sino en forma de restricciones y sin incertidumbre.

Pero además de estos contextos de decisión de optimización clásica, existen otros tres grandes bloques según se recoge en [15] sobre los que abordar el análisis de la teoría de la decisión:

- a) *La teoría de la decisión con incertidumbre o riesgo*. En la que la toma de decisiones podría analizarse con aleatoriedad o incertidumbre en los resultados, produciendo que los resultados estén sujetos al azar.
- b) *La decisión multicriterio*. Una decisión tiene perfectamente determinadas sus consecuencias, no definiendo claramente que es lo mejor y existiendo, por lo tanto, varios objetivos en conflicto.
- c) *La teoría de juegos*. Una decisión en la que las consecuencias del resultado están sujetas a la decisión adoptada y a la decisión de los otros jugadores.

En este proyecto se considerará lo "*posible*", como el conjunto de alternativas asociadas a un determinado árbol, mientras que lo "*mejor*", será el conjunto de alternativas ordenadas según lo que más se parece a las preferencias del decisor.

2.2. Decisión multicriterio

Según es definido en [2], la decisión multicriterio (DMC), es un campo interdisciplinario de estudio tradicionalmente descuidado por el economista. Este tipo de decisiones, a su vez, aparece situada entre otras dos grandes áreas: la *Teoría del Consumidor*, desde el punto de vista del unicriterio, y la *Teoría de la Elección Social y de las Decisiones de Grupo*.

De forma general, según [15], dicho problema de decisión multicriterio vendría formulado como:

$$\begin{aligned} \text{opt } z(x) &= (z_1(x), \dots, z_p(x)) \\ x &\in F \end{aligned} \tag{2.1}$$

donde F es el *espacio de decisiones o soluciones* (si es continuo, se denomina región factible, $F \subseteq \mathbb{R}^n$). Al conjunto $z(F)$ se le denomina *espacio de objetivos o resultados*.

Como se recoge en [15], existen tres grandes bloques sobre los que abordar el análisis de la teoría de la decisión. Dentro de los cuales se encuentra la denominada *Decisión multicriterio* (Multiple Criteria Decision), que será objeto de estudio en este apartado. La decisión multicriterio recoge aquellas decisiones en las que un decisor debe seleccionar entre una serie de posibilidades, denominadas alternativas, sobre las cuales el decisor tendrá un conjunto de criterios a considerar a la hora de elegir.

Como se define en [14] "la toma de decisiones multicriterio es un problema crítico de la vida real. Cualquier actividad involucra, de una u otra manera, la evaluación de un conjunto de alternativas en términos de un conjunto de criterios de decisión, donde muy frecuentemente estos criterios están en conflicto unos con otros".

Como dice Thomas L. Saaty [12], "las personas hacen tres tipos generales de juicios para expresar la importancia, la preferencia o la probabilidad y los utilizan para elegir la mejor entre las alternativas en presencia de influencias ambientales, sociales, políticas y de otro tipo". Además, destaca que dichos juicios son basados en los conocimientos de la memoria o el análisis de los beneficios, costes y riesgos a la hora de seleccionar la mejor decisión, es por ello que es muy importante disponer de la información adecuada para obtener la mejor solución dentro del conjunto de posibles alternativas.

En resumen, la toma de decisiones multicriterio es un proceso complejo en el cual la subjetividad y la dependencia de la información disponible juegan un papel esencial. Y por esta razón es necesario disponer herramientas como un "Software de ayuda a la decisión" que mejoren y permitan analizar el conjunto de alternativas mediante un análisis más exhaustivo y sistemático.

2.3. Proceso Analítico Jerárquico

Dentro de los métodos de toma de decisión multicriterio, se encuentra el Proceso Analítico Jerárquico (Analytic Hierarchy Process), el cual será objeto de estudio e implementación en este proyecto. Este método es original del profesor Thomas L. Saaty, y fue presentado por él en el año 1980 [11]. Saaty ha sido un matemático economista de los más importantes del siglo XX y siglo XXI, una vez acabados sus estudios académicos comenzó a trabajar en el departamento de defensa de los Estados Unidos. Tras ello pasó a la universidad, donde empezó a trabajar en cómo podían haber mejorado las decisiones que se habían tomado en el departamento de defensa, y de esta reflexión nació la metodología, llamada *Proceso Analítico Jerárquico* (AHP).

El también conocido como AHP en su origen es un método cuyo objetivo es ayudar a mejorar la toma de decisiones en el mundo de las empresas y de las organizaciones. En cuanto al método, este tiene un gran rigor académico, pues se encuentra muy bien fundamentado tanto psicológicamente como matemáticamente, además de poseer numerosos contrastes empíricos.

En comparación con el resto de técnicas de decisión multicriterio, AHP presenta una serie de ventajas con respecto al resto de técnicas, como son las siguientes:

- Permite desglosar y analizar un problema por partes.
- Está fundamentado psicológica y matemáticamente.
- Permite medir tanto criterios cuantitativos como cualitativos.
- Fomenta la participación de diversos expertos a la hora de tomar una decisión, con una importancia determinada para cada uno de ellos.

- Realiza la verificación del índice de consistencia de la comparación pareada.
- Es capaz de generar una síntesis y un análisis de sensibilidad.

AHP permite organizar de forma gráfica, sencilla y eficiente la información acerca de un problema de decisión, generando de esta manera un modelo jerárquico. Sobre este se aplicarán una serie de pasos necesarios para obtener como solución el conjunto de alternativas ponderadas, según su importancia. Como dice su Thomas L. Saaty, (1980), "*Trata de desmenuzar un problema y luego unir las soluciones de los subproblemas en una conclusión.*"[13]

2.3.1. Fundamentos Psicológicos

Dentro de los fundamentos psicológicos, el profesor Saaty, basándose en los estudios de varios psicólogos del siglo XIX y principios del XX, llega a la conclusión de que la mente humana hay cosas que hace muy bien, como las comparaciones por pares y cosas que las hace regular, como las comparaciones globales. Es por ello que se define una escala numérica como la de la Tabla 2.1, que permite cuantificar la comparación pareada. Esta escala está definida en el rango $[1 - 9]$, donde un 1 representa que dos criterios tienen igual de importancia en la selección de un determinado criterio, mientras que un 9 representa la importancia extrema de un criterio A frente a un criterio B .

Este proceso de comparación pareada no consiste en asignar un número para ordenar las alternativas. Puesto que, por un lado, se tendría la asignación de un número a una magnitud medible, mientras que por el otro, la utilización de la escala de Saaty, la cual tiene como objetivo capturar la realidad percibida por el experto, siendo este un proceso muy diferente a una simple asignación y normalización arbitraria de números [6].

Valor	Definición	Comentarios
1	Igual importancia	El criterio A es igual de importante que el criterio B
3	Importancia moderada	La experiencia y el juicio favorecen ligeramente al criterio A sobre el B
4	Importancia grande	La experiencia y el juicio favorecen fuertemente al criterio A sobre el B
7	Importancia muy grande	El criterio A es mucho más importante que el B
9	Importancia extrema	La importancia del criterio A sobre el B está fuera de toda duda
2, 4, 6 y 8	Valores intermedios a las anteriores, cuando es necesario matizar	

Tabla 2.1: Escala fundamental de comparación por pares (Saaty, 1980)

2.3.2. Fundamentos matemáticos

Este método también se encuentra fundamentado matemáticamente, gracias a los grandes conocimientos matemáticos del profesor Thomas L. Saaty. El método presenta una gran robustez matemática en referencia a la matriz de comparaciones pareada (Fórmula 2.2)

$$A = \begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{pmatrix} \quad (2.2)$$

Esta matriz, formada por los valores obtenidos de la comparación por pares, es decir, el valor de a_{12} será el valor de la escala de Saaty asignado por el experto a la pareja formada por los criterios 1 y 2. Esta debe cumplir con los siguientes principios para poder aplicar el algoritmo que genere como solución el conjunto de alternativas con un peso asociado a la preferencia de decisión:

- *Reciprocidad.* Si $a_{ij} = x$, entonces $a_{ji} = 1/x$, con $1/9 \leq x \leq 9$
- *Homogeneidad.* Si los elementos i y j son considerados igualmente importantes, entonces $a_{ij} = a_{ji} = 1$ además $a_{ii} = 1$ para todo i .
- *Consistencia.* Se satisface que $a_{ik} * a_{kj} = a_{ij}$ para todo $1 \leq i, j, k \leq q$

2.3.3. Fases para la resolución del AHP

Estructuración del modelo jerárquico

Esta primera fase de estructuración constituye una de las partes más importantes del método AHP, permitiendo organizar las ideas, obteniendo una visión gráfica y jerarquizada de la información general del problema a resolver. Esta estructura jerárquica está constituida por tres partes principales: un objetivo, unos criterios y una serie de alternativas. A continuación se exponen los pasos necesarios para la construcción de la estructura jerárquica del AHP, como la mostrada en la Figura 1.1:

1. *Definición del objetivo.* Este paso está constituido por la definición del objetivo que se pretende alcanzar con la resolución del problema, y será definido por el decisor del proceso.
2. *Selección de criterios.* Se deben de definir una serie de criterios y subcriterios que permitan valorar el conjunto de alternativas, estableciendo preferencias de los involucrados en el proceso de la toma de decisión, impactando directamente al objetivo a determinar.
3. *Selección de alternativas.* Se deben definir un conjunto de opciones factibles, también conocidas como alternativas. Sobre las cuales se determinará el objetivo final.

Valoraciones

Esta segunda fase tiene como objetivo establecer la importancia que cada experto asigna a cada uno de los criterios. Para ello, cada experto deberá realizar una valoración de cada par de criterios, formando de esta manera una matriz de comparaciones pareadas (Matriz 2.2). Esta matriz de comparaciones pareada representará el conjunto de valoraciones de un experto para una determinada jerarquía, donde cada valoración (a_{ij}) deberá estar contenida en el rango [1-9] correspondiente a la escala numérica, definida por Saaty en la Tabla 2.1, para la comparación de pares.

Una vez conocida la matriz de comparaciones pareadas, esta debe ser normalizada para determinar los pesos determinados para cada criterio. Para la normalización de esta matriz basta dividir cada valor de una columna "j", por la suma de los valores de dicha columna, como se muestra en la Matriz 2.3. El sumatorio de los valores de cada columna de la matriz Aw debe ser 1.

$$Aw = \begin{pmatrix} \frac{a_{11}}{\sum a_{i1}} & \frac{a_{12}}{\sum a_{i2}} & \dots & \frac{a_{1n}}{\sum a_{in}} \\ \frac{a_{21}}{\sum a_{i1}} & \frac{a_{22}}{\sum a_{i2}} & \dots & \frac{a_{2n}}{\sum a_{in}} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{a_{n1}}{\sum a_{i1}} & \frac{a_{n2}}{\sum a_{i2}} & \dots & \frac{a_{nn}}{\sum a_{in}} \end{pmatrix} \quad (2.3)$$

Cálculo de prioridades

Una vez determinada la matriz de comparaciones pareadas y su correspondiente normalización, se debe calcular el vector de prioridades. Este vector de prioridades, el cual denominaremos C , se encuentra representado en la Matriz 2.4, indicará el peso o importancia relativa asignada por cada experto a cada uno de los diferentes criterios. En cuanto al cálculo de C , esta es:

$$C_i = \frac{\sum_{i=1}^n Aw_i}{n} \quad (2.4)$$

Donde n es el número de criterios y C_i es la prioridad del criterio i otorgada por un determinado experto.

Análisis de consistencia

Una condición importante a la hora de obtener una decisión, en términos de calidad, es la que hace referencia a la consistencia de las valoraciones realizadas por el experto. Es por ello que el AHP proporciona un método capaz de medir el grado de consistencia de dichas valoraciones. Para determinar dicho grado de consistencia, se debe calcular la multiplicación de AxC , obteniendo como resultado un vector final total (x_i).

$$A \times C = \begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{pmatrix} \times \begin{pmatrix} C_1 \\ C_2 \\ \vdots \\ C_n \end{pmatrix} = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix} \quad (2.5)$$

Una vez se disponga del vector X , podremos estimar el valor de λ_{max} . Dicha estimación puede ser calculada haciendo uso de la Fórmula 2.6:

$$\lambda_{max} = \frac{\sum_{i=1}^n \frac{x_i}{C_i}}{n} \quad (2.6)$$

La consistencia de la matriz A , será obtenida mediante el *índice de consistencia* (CI), cuya Fórmula 2.7, viene caracterizada por el λ_{max} , que es el máximo autovalor y n que es la dimensión de la matriz de comparaciones pareadas.

$$CI = \frac{\lambda_{max} - n}{n - 1} \quad (2.7)$$

Una vez obtenido el índice de consistencia (CI), se debe de obtener la *proporción de consistencia* (CR), el cual determinará si dicha matriz es consistente o no, pues la matriz será aceptada siempre que no supere los valores indicados en la Tabla 2.2. En el caso contrario, es decir, cuando el resultado supere los valores de la tabla, se recomienda al experto revisar sus valoraciones.

Tamaño de la matriz (n)	Ratio de consistencia (CR)
3	5 %
4	9 %
5 o mayor	10 %

Tabla 2.2: Porcentajes máximos del ratio de consistencia CR

Para el cálculo del CR se hace uso de la Fórmula 2.8. La cual viene determinada por el *índice de consistencia* (CR) y el *índice aleatorio* (IR), que indica la consistencia de una matriz aleatoria (Tabla 2.3).

$$CR = \frac{CI}{RI} \quad (2.8)$$

Tamaño de la matriz (n)	2	3	4	5	6	7	8	9	10
Índice Aleatorio (IR)	0	0.58	0.9	1.12	1.24	1.32	1.41	1.45	1.49

Tabla 2.3: Índice aleatorio RI

Síntesis

Una vez se repitan las fases anteriores con cada una de las opiniones de los expertos, obtendremos por cada experto un *vector de prioridades* de los criterios y un ratio de consistencia (CR). Dichos vectores de prioridades serán introducidos en una matriz (Tabla 2.4) donde cada fila i representa a un determinado criterio, y cada columna j representa a un determinado experto. Obteniendo, por lo tanto, que a_{i1} correspondiente a la columna 1, será el vector de prioridades calculado anteriormente para el experto 1.

	Experto 1	Experto 2	...	Experto n
Criterio 1	b_{11}	b_{12}	...	b_{1n}
Criterio 2	b_{21}	b_{22}	...	b_{2n}
\vdots	\vdots	\vdots	\vdots	\vdots
Criterio n	b_{n1}	b_{n2}	...	b_{nn}

Tabla 2.4: Vectores de prioridades de los criterios

Una vez creada la matriz anterior, se debe calcular el *vector de prioridades global*. Este puede ser determinado de dos maneras distintas: la primera de ellas, se debe aplicar si todos los expertos tienen la misma importancia en el proceso de selección, por lo que dicho vector sería calculado como la media aritmética de los pesos de cada fila. Mientras que la segunda manera, se aplicará si queremos asignar a cada experto un determinado peso, dichos pesos vendrán determinados en el vector P (Vector 2.9).

$$P = \begin{pmatrix} p_1 \\ p_2 \\ \vdots \\ p_n \end{pmatrix} \quad (2.9)$$

Una vez asignados dichos pesos, el *vector de prioridades global* será calculado como la media ponderada de cada una de las filas de la Tabla 2.4. Dicho cálculo se hará siguiendo la siguiente Fórmula 2.10, donde V_i representa el vector de prioridades global, la i hace referencia al índice del criterio, y la j al índice del experto.

$$V_i = \frac{\sum_{j=1}^n b_{ij} \cdot p_j}{n} \quad (2.10)$$

El vector de prioridades global (V), representará las prioridades, asignadas por un conjunto de expertos, a la hora de posteriormente seleccionar una determinada alternativa de su conjunto factible.

Toma de la decisión

En esta última fase se procede a tomar la decisión, es decir, seleccionar la alternativa que más se adecue a los criterios. Para ello se deben generar un vector propio por cada

	Criterio X	Vector Propio
Alternativa 1	Valor 1	Valor1 / suma
⋮	⋮	⋮
Alternativa n	Valor n	Valorn / suma
Suma	Suma	1

Tabla 2.5: Vector propio de un determinado criterio

criterio, dicho vector propio será calculado siguiendo la Tabla 2.5, si dicho criterio es una variable cuantitativa.

Si el criterio para el cual se debe obtener su vector propio, es una variable cualitativa, entonces se debe plantear al experto una encuesta de igual manera que en la fase 2 (2.3.3), en la que se comparen en lugar de los criterios, las alternativas y obteniendo de igual manera un vector de prioridades que se corresponderá con el vector propio de dicho criterio.

Una vez obtenido el vector propio de cada criterio, en función de las diversas alternativas, tendremos como resultado una Tabla 2.6 donde cada columna representará a un criterio y, por lo tanto, a su vector propio, y cada fila a una determinada alternativa.

	Criterio 1	Criterio 2	⋯	Criterio m
Alternativa 1	vp_{11}	vp_{12}	⋯	vp_{1m}
⋮	⋮	⋮	⋮	⋮
Alternativa n	vp_{n1}	vp_{n2}	⋯	vp_{nm}

Tabla 2.6: Conjunto de vectores propios

Dicha Tabla 2.6 puede ser tratada como una matriz con nombre VP , y la cual será utilizada para calcular el *vector de prioridades de las alternativas*, este representará una clasificación del conjunto factible de soluciones y, por lo tanto, resultando esta clasificación como resultado a la toma de decisión. Para la obtención de dicha clasificación basta con multiplicar la matriz VP por el vector de prioridades global (V), obteniendo como resultado un vector de prioridades de las alternativas 2.11 (R).

$$R = VP \times V = \begin{pmatrix} vp_{11} & vp_{12} & \cdots & vp_{1n} \\ vp_{21} & vp_{22} & \cdots & vp_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ vp_{n1} & vp_{n2} & \cdots & vp_{nm} \end{pmatrix} \times \begin{pmatrix} v_1 \\ v_2 \\ \vdots \\ v_m \end{pmatrix} = \begin{pmatrix} r_1 \\ r_2 \\ \vdots \\ r_m \end{pmatrix} \quad (2.11)$$

2.3.4. Contrastes empíricos

El Proceso Analítico Jerárquico, desde su creación, ha sido implementado como soporte para resolver problemas de decisión de diferente tipo, nivel y alcance. Aplicándose esta técnica a una amplia gama de situaciones, como la selección de alternativas competitivas en un entorno con múltiples objetivos, la asignación de recursos escasos y la generación

de pronósticos. A su vez, resulta sencillo, intuitivo y fácil de comprender, lo que hace que cuente con numerosos contrastes empíricos, en forma de aplicaciones o publicaciones del mismo en una gran diversidad de ámbitos. A continuación se exponen algunas de las que se han considerado interesantes:

Este primer caso de aplicación, planteado en el artículo [8], y con título “Seleccionar una técnica de modelamiento para el diseño de un modelo inventario administrado por el proveedor para una compañía retail colombiana”, contó inicialmente con 19 expertos, con un alto nivel profesional. Estos expertos fueron consultados con el objetivo de obtener un conjunto de alternativas sobre las que tomar la decisión, todas estas representadas en el nivel inferior de la Figura 2.1. Dichas alternativas fueron evaluadas según los cuatro criterios seleccionados y representados en el nivel intermedio de la figura anterior. Una vez construida la estructura jerárquica del problema, y la posterior valoración de cada par de criterios por parte de los expertos, se procedió a la resolución del AHP. Obteniendo como resultado que la alternativa que mejor se ajusta a los requerimientos, fue el *Modelo Matemático Analítico* con un porcentaje de preferencia del 49,1 %, superior a la *Teoría de Juegos* y la *Programación No Lineal* con un 23,8 % y 27,1 %.

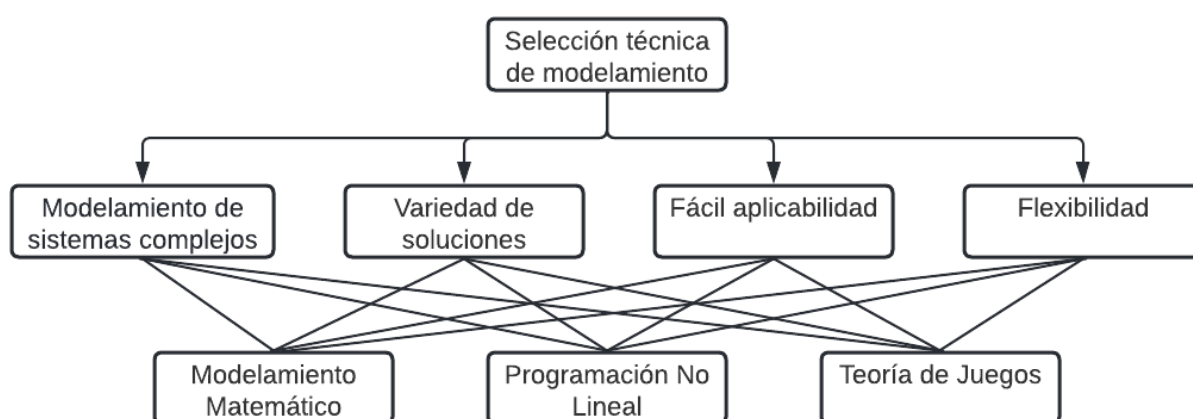


Figura 2.1: Estructura jerárquica del caso

Este segundo caso de estudio planteado en el artículo [7], y con título “Criterios de decisión para el análisis de la contaminación lumínica en España”, trata de recurrir al AHP para obtener como resultado el peso o vector de preferencias de los criterios más influyentes con respecto a la perspectiva de la contaminación lumínica. Una vez conocido el objetivo de dicho estudio se definen los siguientes criterios: C_1 - Flujo natural, C_2 - Potencia emitida por superficie artificial, C_3 - Potencia emitida por habitante, C_4 - Puntos de luz por superficie artificial, C_5 - Puntos de luz por habitante, C_6 - Puntos de luz por municipio. Tras ello, se procedió a la valoración de cada par de criterios por parte de los expertos para la posterior aplicación del AHP. Obteniendo como resultado, los representados en la figura 2.2, donde se observa como los expertos se han decantado por el criterio C_2 , correspondiente con la Potencia emitida por superficie artificial, con una prioridad del 41.19 %, como el más importante.

	Vector de pesos global (%)
C ₁	26.25
C ₂	41.19
C ₃	10.93
C ₄	9.36
C ₅	6.59
C ₆	5.68

Figura 2.2: Vector de prioridades resultante

2.3.5. Aplicaciones para la toma de decisiones

AHP Online System - AHP-OS

Esta aplicación web gratuita [3], descrita como una herramienta de soporte para el proceso de toma de decisiones, la cual puede ser de ayuda en el trabajo diario, permitiendo abordar desde problemas de decisión simple hasta problemas muy complejos. Dicha aplicación dispone de 5 funciones principales¹:

- *My AHP Projects*. Sección desde la cual una vez registrado permite administrar tus proyectos AHP, permitiendo crear, abrir, modificar y eliminar jerarquías.
- *AHP Priority Calculator*. Sección desde la que ingresando un conjunto de criterios permite realizar la valoración pareada de los mismos, obteniendo como resultados la consistencia, prioridades, matrices de comparaciones pareadas, etc.
- *AHP Hierarchies*. Sección desde la que definir la estructura jerárquica del problema, permitiendo además introducir alternativas, obteniendo como resultado la prioridad de las mismas.
- *AHP Group Session*: sección de para la toma de decisiones en grupo totalmente en línea.
- *Group Consensus Cluster Analysis*. Sección experimental, que permite el análisis de conglomerados de consenso grupal. Agrupando a los expertos en subgrupos más pequeños con mayor consenso.

SpiceLogic Analytic Hierarchy Process

El *SpiceLogic Analytic Hierarc Process* ² es un software basado en un asistente capaz de modelar el proceso de jerarquía analítica paso a paso. Y descrito como un "Software AHP rápido, fácil y hermoso como aplicación de escritorio en Windows".

¹<https://bpmsg.com/ahp/>

²<https://www.spicelogic.com/Products/ahp-software-30>

Este Software ofrece algunas características muy interesantes como:

- Experiencia de usuario destacada e intuitiva.
- Tutorización gratuita en vivo.
- Modelar de un nivel infinito de criterios y subcriterios.
- Toma de decisiones en grupo sin ninguna nube.
- Análisis de sensibilidad de criterios.
- Selección del método de cálculo deseado.
- Obtención de informes en PDF o Excel.
- Posibilidad de trabajo 100 % sin conexión.
- Compatible con cualquier sistema operativo Windows que tenga Microsoft.

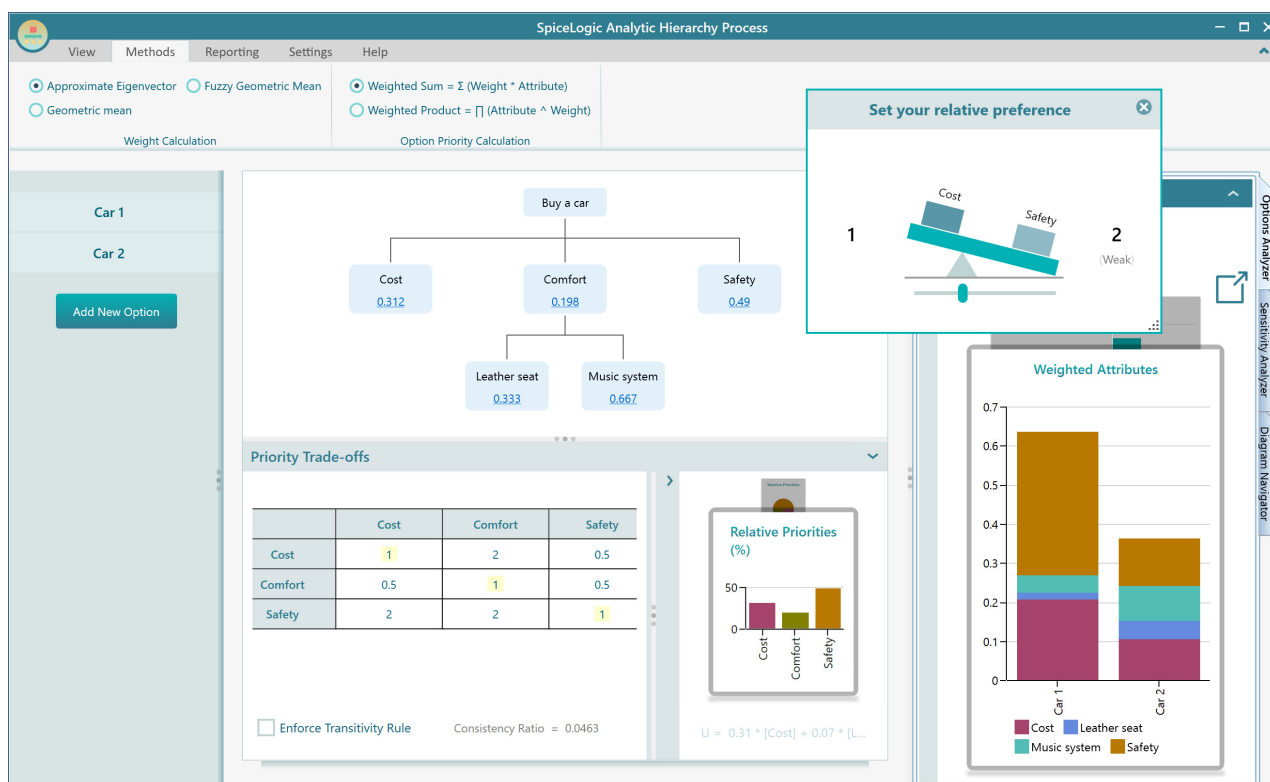


Figura 2.3: Interfaz gráfica de SpiceLogic AHP.³

En este TFG se plantea la creación de una aplicación, la cual se diferencia con las comparadas anteriormente en algunas características, como que esta se trata de una aplicación web y gratuita, mientras que *SpiceLogic*, es una aplicación de escritorio solo disponible para el sistema operativo Windows y de pago, con solo un mes de prueba gratuito. Además, nuestra aplicación está pensada para resolver el AHP de una forma sencilla y muy gráfica, permitiendo al usuario una rápida familiarización con la misma,

³<https://spicelogicstorage.blob.core.windows.net/product/ahpsoftware/featureImages/37/ahp-software-spicelogic.jpg>

característica no destacable en este caso de la aplicación *AHP-OS*, la cual presenta una estructura en forma de artículo de investigación, en la que abunda el texto y carece de diseño amigable y accesible para el usuario.

Capítulo 3

Tecnologías y arquitectura de software

En esta sección se expondrá aquellos aspectos más importantes relacionados con las Tecnologías utilizadas para el desarrollo de esta aplicación web, así como su arquitectura a nivel de software y composición de la misma.

3.1. Tecnologías

Para el desarrollo de este proyecto se han hecho uso de las siguientes tecnologías:

3.1.1. MongoDB

MongoDB¹ es un sistema de base de datos NoSQL orientado a documentos, desarrollado bajo el concepto de código abierto y escrito en C++. Esta base de datos destaca por su método de almacenamiento de datos, ya que en lugar de almacenar los datos en tablas como lo hacen las bases de datos relacionales, almacenan estructuras de datos en documentos con una estructura BSON (similar a JSON), y siguiendo un esquema dinámico, proporcionando de esta manera una mayor rapidez y facilidad de integración de los datos.

Existen otras opciones como *CouchDB*², *Elasticsearch*³ y *RavenDB*⁴, también NoSQL o incluso otras SQL como podrían ser *PostgreSQL*⁵, *MariaDB*⁶, *MySQL*⁷ y muchas más. La elección de dicha base de datos ha sido apoyada por las ventajas que aporta este tipo de datos, destacando características como su escalabilidad, flexibilidad, además de su extensa documentación y conocimiento previo de la misma, ya que se trata de una de las bases de datos más utilizadas a lo largo de la carrera.

¹<https://www.mongodb.com/es>

²<https://couchdb.apache.org/>

³<https://www.elastic.co/es/what-is/elasticsearch>

⁴<https://ravendb.net/>

⁵<https://www.postgresql.org/>

⁶<https://mariadb.org/>

⁷<https://www.mysql.com/>

3.1.2. Spring

Spring⁸ es un marco de trabajo o Framework más popular para el desarrollo de aplicaciones con el lenguaje de programación Java, permitiendo de la creación de código de alto rendimiento, liviano y reutilizable. Entre sus características destacables se encuentra la de permitir que los equipos de desarrollo puedan enfocarse directamente en la lógica que requiere la aplicación, haciendo el proceso más corto, rápido y eficaz.

Si bien es cierto que Spring es muy potente, la configuración inicial y la preparación de las aplicaciones para la producción son tareas bastante tediosas. Es por ello que se opto por le uso de *Spring Boot*, ya que este simplifica el proceso de configuración para el despliegue del proyecto al máximo. Permitiendo compilar aplicaciones web como un archivo .jar que podemos ejecutar como una aplicación Java normal.

3.1.3. Vue.js

Vue.js⁹ es un framework progresivo Javascript para construir interfaces de usuario, de una forma rápida, sencilla y muy práctica. Que sea progresivo, quiere decir, que permitirá el desarrollo desde componentes sencillos que implementan una determinada parte de una aplicación, hasta aplicaciones front-end completas.

Entre las características por las cuales nos decantamos por el uso de este framework son algunas como su curva de aprendizaje, pues esta es con diferencia la menos pronunciada con respecto a la del resto de frameworks más populares. Además de una de las características fundamentales, como es el trabajo con componentes, estos, son elementos que encapsulan código reutilizable, dividido en tres fragmentos de código encargados de formar la estructura, aportar la lógica y dar estilo, a dicho elemento con HTML, Javascript y CSS. También destacar su flexibilidad y rendimiento, sin olvidarnos del rápido crecimiento que ha experimentado dicho framework en los últimos años, en lo que a número de aplicaciones que hacen uso del mismo se refiere.

Vuetify

Vuetify¹⁰ es una librería de VueJS destinada a la creación de interfaces de usuario que implementa componentes basados en la filosofía de *Material Design*. Dicha librería nos permitirá implementar diversas pantallas de nuestra aplicación de una manera mucho más sencilla y rápida, pues esta nos ofrece una gran variedad de componentes, estilos y animaciones ya desarrollados y preparados para ser utilizados directamente en nuestra interfaz.

⁸<https://spring.io>

⁹<https://vuejs.org/>

¹⁰<https://vuetifyjs.com/en/>

i18n

I18n¹¹ es otra librería de VueJS, la cual nos permite diseñar la aplicación capaz de adaptarse a diferentes idiomas.

3.1.4. Resto de tecnologías

Docker

Docker¹² es una herramienta de código abierto que automatiza el despliegue de aplicaciones en contenedores virtuales. Para ello, proporciona un modelo de implementación basado en imágenes, que es lo que se permite compartir una aplicación o un conjunto de servicios con todas sus dependencias en varios entornos. Permitiendo crear contenedores ligeros y portables para las aplicaciones software que pueden ejecutarse en cualquier máquina con Docker instalada, independiente del sistema operativo que la máquina tenga por debajo, facilitando así el despliegue de la misma.

Docker Hub

Docker Hub¹³ es la biblioteca y comunidad de imágenes de contenedores más grande del mundo. Permitiendo así almacenar en esta biblioteca las imágenes creadas para este proyecto, y que estas sean descargadas e implementadas por cualquier persona con acceso a dicho repositorio para la puesta en marcha de la aplicación.

Docker Compose

Docker Compose¹⁴ es una herramienta la cual nos permitirá definir el conjunto de contenedores que conforman nuestra aplicación y ejecutarla. Todo ello, mediante la configuración de un archivo YAML, en el que se configurarán todos los servicios de nuestra aplicación. Posteriormente, con el comando *docker-compose up*, se podrá desplegar toda la infraestructura de nuestra aplicación.

Git & GitHub

Git¹⁵ es un sistema de control de versiones, de código abierto e instalado localmente para gestionar todo el código fuente del proyecto. GitHub¹⁶ es una plataforma basada en la nube, la cual los desarrolladores usan como alojamiento para los repositorios de Git.

¹¹<https://kazupon.github.io/vue-i18n/>

¹²<https://www.docker.com/>

¹³<https://hub.docker.com/>

¹⁴<https://docs.docker.com/compose/>

¹⁵<https://git-scm.com/>

¹⁶<https://github.com/>

3.2. Arquitectura

En lo que respecta en la arquitectura software de la aplicación, esta viene representada en la Figura 3.1, y ha sido dividida en tres módulos principales: un back-end (API-REST), un front-end (Interfaz de usuario) y el AHP (Simulación de una caja negra para la resolución de todas las fases del proceso analítico jerárquico).

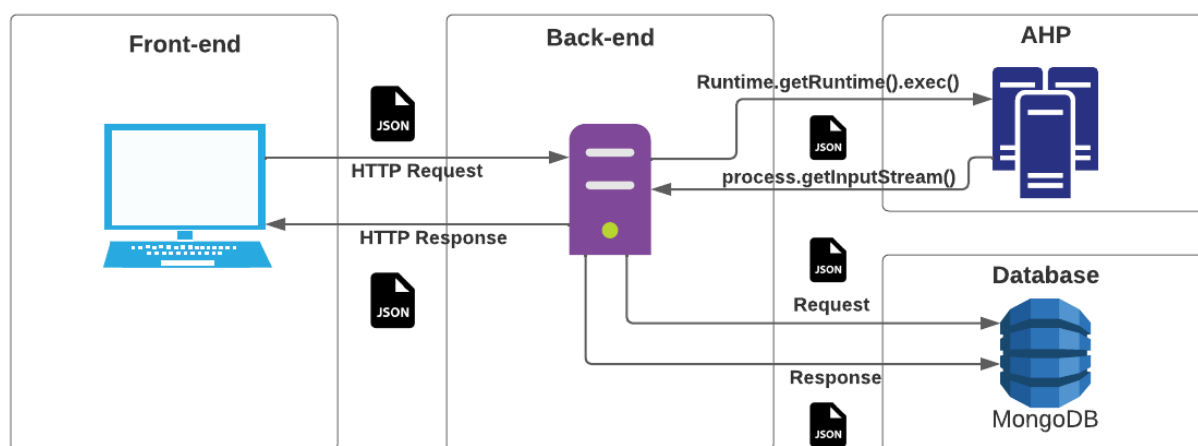


Figura 3.1: Estructura de la aplicación

Siguiendo el esquema representado en la Figura 3.1, podemos observar como cada módulo está encerrado en una especie de caja, que representa a los contenedores de docker que se han construido para cada uno de ellos. Estos contenedores serán creados en el proceso de despliegue de la aplicación haciendo uso de *Docker Compose*. Una herramienta que permite definir el conjunto de contenedores que componen la aplicación y crearlos de una sola vez.

Para que el despliegue funcione correctamente, las imágenes del front-end y back-end deben estar publicadas en *Docker Hub* previamente, de tal manera que sea posible acceder a ellas. En este caso, tenemos una cuenta que es *softwarebuild*, cedida por el profesorado, donde publicar las imágenes del proyecto.

Una vez creado el `docker-compose.yml` y lanzado mediante `docker-compose up`, ya tendremos desplegados y con conexión entre los contenedores del front-end, back-end y MongoDB.

Creados todos los contenedores, y comprobado su funcionamiento, el siguiente paso sería la conexión entre el front-end y el back-end para el envío de peticiones (*HTTP Request*) y respuestas (*HTTP Response*) entre ellos. Para ello se instala la librería de VueJS, llamada *Axios*, una librería cliente HTTP, para la realización de peticiones basado en promesas. Para la instalación de esta librería se debe ejecutar el Comando 3.1.

```
1 $ yarn add axios vue-axios
```

Listing 3.1: Comando para la instalación de Axios en el front-end

Tras la instalación, para el uso de esta, se debe crear el directorio `front-end/src/service`. En este directorio serán creadas las distintas clases encargadas de realizar las peticiones

a cada uno de los endpoints del back-end. Estas clases definen un método por cada tipo de petición, como se observa en el Código 3.2, declarando en su cuerpo la petición, con una llamada a axios continuado del tipo de petición a realizar, su URL y el cuerpo de dicha petición, si procede, y retornando el valor devuelto por el back-end en forma de promesa.

```

1 import axios from 'axios';
2
3 export default class TreeService {
4
5 // Metodo para la realizacion de una peticion de tipo GET a la URL del back-end
6 getAll() {
7     return axios.get("http://localhost:8080/v1/trees");
8 }
9
10 save(data) {
11     return axios.post("http://localhost:8080/v1/trees", data)
12 }
13
14 update(id, data) {
15     return axios.put("http://localhost:8080/v1/trees/" + id, data)
16 }
17
18 delete(id) {
19     return axios.delete("http://localhost:8080/v1/trees/" + id)
20 }
21
22 getId(id) {
23     return axios.get("http://localhost:8080/v1/trees/" + id)
24 }
25 }

```

Listing 3.2: Clase TreeService para la realización peticiones al endpoint /trees

Estos servicios son definidos para cada endpoint, a los que queremos realizar una petición, y usados desde cada una de las vistas, como se muestra en el código 3.3. Este código corresponde a una petición lanzada desde una de las vistas de nuestro proyecto, en la que se trata realizar una petición del tipo POST, pasando como cuerpo un objeto JSON (trees), con la información del árbol a almacenar. Una vez lanzada dicha petición hacia el back-end, este debe de enviar un mensaje de vuelta (HTTP Response) que será retornado por el servicio en forma de promesa, para el posterior análisis del código de estado de dicha petición o manejar los errores producidos mediante un catch.

```

1 <script>
2 import TreeService from '@/service/TreeService';
3 export default {
4   name: 'addCriterion',
5   treeService: null,
6
7   created() {
8     this.treeService = new TreeService();
9   },
10
11   methods: {
12     save() {
13       this.treeService.save(this.tree).then(data => {
14         if (data.status === 201) {
15           console.log("Arbol almacenado correctamente")

```

```

16     }
17     }).catch((error) => {
18         console.log(error)
19     }
20 }
21 }
22 }
23 </script>

```

Listing 3.3: Petición de POST al endpoint /trees

Una vez configurada y comprobada la conexión entre el front-end y el back-end, se debe configurar otra petición de tipo POST de igual manera que se hizo con la anterior, pero en este caso para el lanzamiento del AHP. Este procedimiento se lleva a cabo siguiendo el orden presente en la Figura 3.2, donde partimos del front-end con el paso número uno en el que se realiza una petición POST al endpoint /ahps, una vez la petición llegue al back-end, este se encargará de extraer los datos necesarios para la ejecución del AHP. Concretamente valoraciones, alternativas y expertos. Una vez se disponga de estos datos en el back-end, este debe validar los datos y lanzar el algoritmo AHP, mediante el uso de `Runtime.getRuntime().exec()`, para así, ejecutar un comando de shell externo. En este caso, el Comando 3.4, es el que se debe ejecutar para el lanzamiento del AHP.

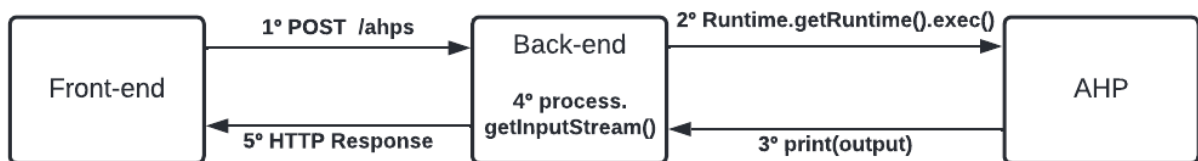


Figura 3.2: Flujo de ejecución del AHP.

```

1 python3 main.py -t trees.json -a alternatives.json -e experts.json

```

Listing 3.4: Comando para el lanzamiento del AHP

Tras el lanzamiento del comando anterior desde el back-end, se ejecutará todo el proceso de resolución del AHP, imprimiendo por consola la solución obtenida, como se muestra en el paso tres de la Figura 3.2. Es aquí, donde el back-end, debe capturar el flujo de salida del proceso creado al lanzar el Comando 3.4. Para ello hace uso del `process.getInputStream()` en el paso cuatro, conectándose a la salida estándar del proceso anterior, y almacenando cada línea en una variable que será retornada en el paso cinco, como una respuesta a la petición que se realizó en el paso uno. Es en este punto, donde ya se tiene la respuesta del AHP, en nuestro front-end y podemos pasar a mostrarla en la interfaz de usuario.

3.3. Composición de la aplicación

En este proyecto se desarrolló una aplicación web *Full Stack* desde cero, esto quiere decir que en el proceso de desarrollo se ha abarcado desde la parte más visual de la

aplicación (front-end), hasta por el desarrollo de un servidor (back-end) capaz de recibir y procesar las peticiones realizadas por el usuario, todo ello pasando por otro módulo (AHP) capaz de resolver el proceso analítico jerárquico. En resumen, esta aplicación web dinámica, está compuesta por tres principales componentes descritos a continuación a nivel estructural y en el orden en el que fueron implementados:

3.3.1. Back-end

Este módulo desarrolla una API-REST, es decir, define un conjunto de normas que determinan como en este caso el front-end se comunicará con la base de datos. Esta API-REST ha sido desarrollada con Spring Boot, una tecnología de Spring, la cual nos facilita la configuración de dependencias, así como el despliegue de la misma. Este módulo fue inicializado mediante el *spring initializr*¹⁷, permitiendo descargar la estructura principal, representada en la Figura 3.3.

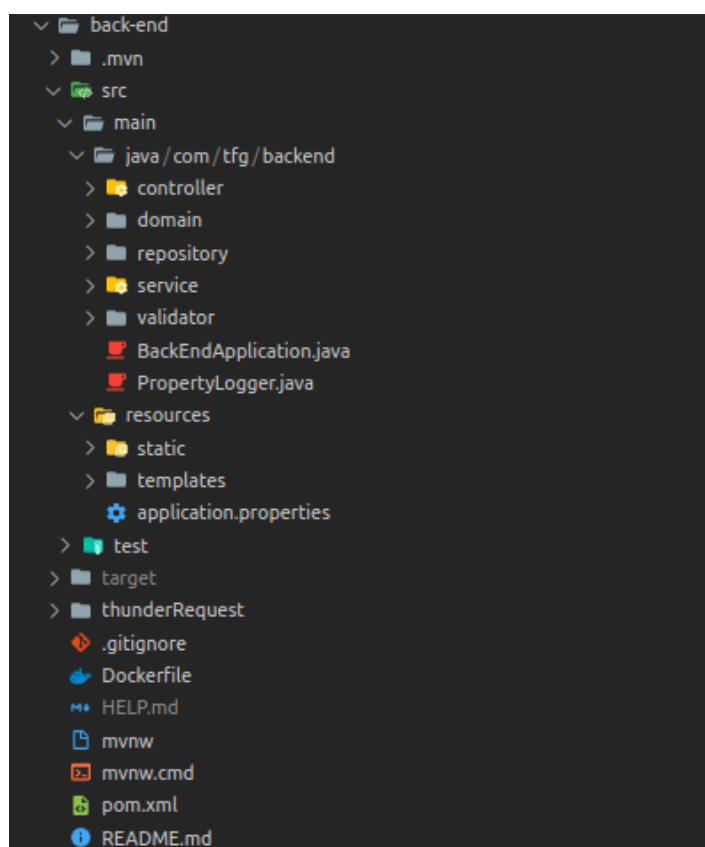


Figura 3.3: Estructura directorios del back-end

Endpoints definidos

Dicho back-end define un conjunto de endpoints, en el directorio `/back-end/src/main/java/com/tfg/backend`. Cada endpoint cuenta con una clase que define el esquema del objeto que representará dicho endpoint, un validador para validar el contenido de dicho

¹⁷<https://start.spring.io/>

objeto, unos servicios para el manejo de la base de datos y un controlador, o *controller* para el manejo de los distintos verbos HTTP a los que responde cada endpoint. El conjunto de endpoints definidos para esta API-REST, son los siguientes:

- **/trees:** *endpoint* encargado de la representación de un árbol del AHP, el cual tiene como atributos un *Id* para la identificación del recurso, un nombre, una descripción y una raíz que será el nodo raíz del árbol y, por lo tanto, un elemento de tipo Criterio.
- **/criteria:** *endpoint* encargado de la representación de un criterio del AHP, este al igual que el anterior, dispone de atributos como un *Id*, un nombre, una descripción y un conjunto de criterios hijos del criterio en cuestión.
- **/experts:** *endpoint* encargado de la representación de cada uno de los expertos que participarán en la toma de decisión, este dispone de atributos como un *Id*, un nombre, una descripción y una imagen de perfil.
- **/ratings:** *endpoint* encargado de la representación de la valoración concreta hecha por un experto para un par de criterios, permitiendo indicar, con un rango de [-10,10], cuál de los dos criterios se considera más importante. Este endpoint dispone de atributos como el *Id*, el identificador del primer criterio que conforman la pareja, el identificador del segundo criterio, el identificador del experto que realizo dicha valoración y el valor entre -10 y 10 asignado por el experto a este par de criterios.
- **/opinions:** *endpoint* encargado de la representación de la opinión que tiene un experto sobre el conjunto de criterios a evaluar, este dispone de algunos atributos como el *Id*, una nota, el identificador del experto que realizo dicha opinión y un conjunto de valoraciones expresadas por el experto para cada par de criterios.
- **/alternatives:** *endpoint* encargado de la representación de cada una de las alternativas a ser evaluadas mediante el AHP. Dicha alternativa tendrá como atributos un identificador, un identificador del árbol al que está asociado dicha alternativa, un nombre, una descripción, y un atributo llamado *Data* que representa al conjunto de pares que representan los datos asociados a la alternativa, dicha pareja a su vez está formada por el identificador del criterio y el valor asignado para el mismo.

Peticiones permitidas

Todos estos *endpoints*, cuentan con una serie de métodos definidos en el directorio */controller*. Estos controladores definen el código a ejecutar como respuesta a una petición HTTP:

- **GET:** permite obtener todas entidades existentes asociadas a un determinado *endpoint* existente.
- **POST:** permite crear una nueva entidad.
- **PUT:** permite modificar una determinada entidad.
- **DELETE:** permite eliminar una determinada entidad.

- **GET /ratings/:id**: permite obtener una determina entidad a partir de su id.

Todos estos métodos en los que se definen las respuestas a las peticiones HTTP están anotados con anotaciones en las que se indica el tipo de petición y la URL a al que se responde. Además, una vez realizada recibida la petición y antes de responder a esta, nos encontramos ante dos opciones, una de ellas es la de aplicar una validación mediante las clases definidas en */validator* y las cuales se encargan de verificar que el cuerpo de la petición es correcto y, por lo tanto, se puede continuar con la implementación de la lógica de negocio de la aplicación. Es decir, realizar todo el procesamiento lógico de la petición mediante los métodos definidos en las clases del directorio */services*.

3.3.2. Front-end

En este módulo se desarrolla la interfaz de usuario para nuestra aplicación, haciendo uso del framework *VueJS*, y de sus librerías *Vuetify* e *I18n*. Para la creación de la estructura de este módulo, se hizo uso del comando 3.5, el cual generó la estructura representada en la Figura 3.4.

```
1 $ vue create front-end
```

Listing 3.5: Comando para crear el proyecto front-end.

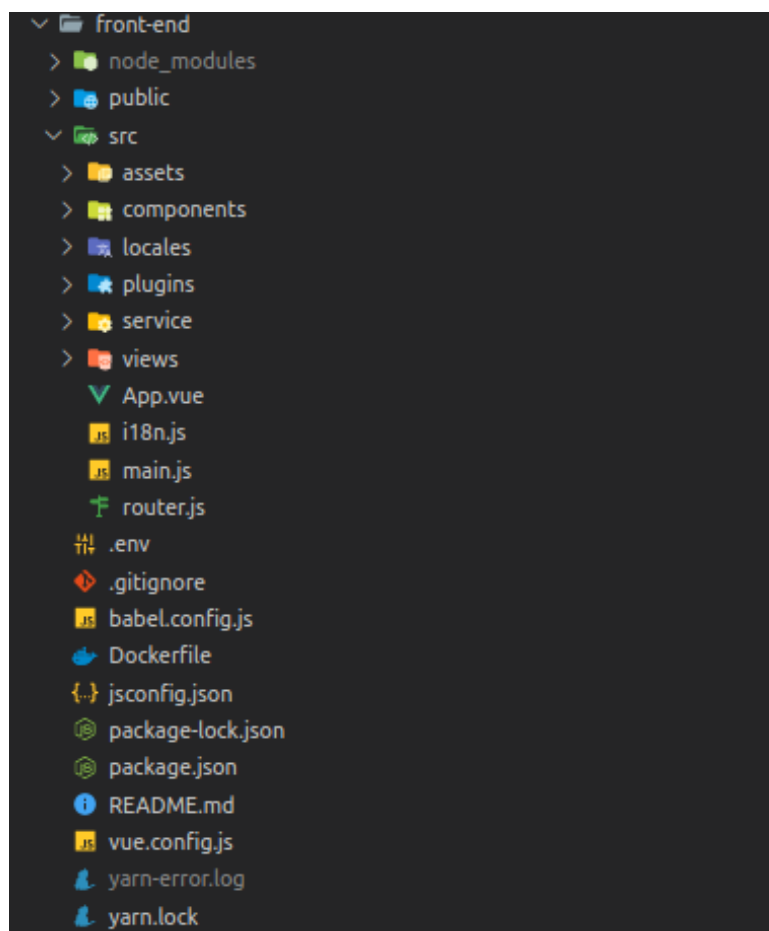


Figura 3.4: Estructura directorios del front-end

En esta estructura de directorios se han implementado cada una de las funcionalidades con las que cuenta la interfaz de usuario de la aplicación desarrollada, y cuyos casos de uso se encuentran recogidos en el diagrama de casos de uso de la Figura 3.5. Se puede observar en dicha figura, como el actor principal, en este caso, el usuario que haga uso de la aplicación, dispondrá de una serie de casos de uso, los cuales reflejan el comportamiento que debe seguir ese usuario para tomar una determinada decisión. Entre estos casos de uso podemos encontrar:

- *Definir criterios*: este caso de uso trata de representar al primer paso en el proceso de la toma de decisiones, donde el usuario haciendo uso de un formulario, podrá definir cada uno de los criterios que formarán el árbol de decisiones. A este caso de uso, también se le incluye el comportamiento descrito en los casos de uso de *Crear*, *Eliminar*, *Actualizar* y *Visualizar*, representando, estos últimos, cada una de las vistas definidas para el manejo de estos criterios, permitiendo así realizar cada una de los tipos peticiones que se definieron para el endpoint `/criteria` en el back-end.
- *Definir árbol de criterios*: caso de uso que trata de representar el segundo paso de la toma de decisiones, en el que el usuario, mediante otro formulario, deberá indicar el nombre del árbol y el nombre del criterio que actuará como raíz del mismo, formando de esta manera la estructura jerárquica del AHP. Para el manejo de esta estructura, se incluye el comportamiento expresado en los cuatro casos de uso nombrados anteriormente, permitiendo así, crear, eliminar, actualizar y visualizar dicha árbol de criterios.
- *Definir alternativas*: caso de uso, que representa el paso en el que el usuario accede a la visualización de un determinado árbol y, define un conjunto de alternativas asociadas a ese árbol. Al igual que los dos casos anteriores, este caso de uso también incluye el comportamiento expresado por los casos de uso de *Crear*, *Eliminar*, *Actualizar* y *Visualizar*.
- *Opinar*: este caso de uso representa el proceso en el que los distintos expertos a participar en el proceso de toma de decisiones, realizan la valoración de cada uno de los pares de criterios, que conforman el árbol creado anteriormente. Dicha valoración será recogida mediante un slider con el que ajustar la preferencia entre cada par de criterios.
- *Seleccionar opiniones*: caso de uso extendido del caso de uso anterior, en el que el usuario podrá seleccionar, las opiniones que desee de las realizadas anteriormente por los expertos. Una vez seleccionadas dichas opiniones, asociadas todas a un mismo árbol, se deberá asignar un peso a cada uno de los expertos participantes. Todo ello antes de pasar al comportamiento incluido mediante el caso de uso de *Lanzar AHP*, el cual trata de representar el proceso en el que se obtiene la solución a una determinada decisión, para finalizar mostrando dicha solución, mediante el comportamiento incluido por el caso de uso *Visualizar solución*. Este último hará uso de un componente de Vue, con un carrusel horizontal de Vuetify, en el que mostrar ordenadas según la prioridad establecida, cada una de alternativas asociadas al árbol.

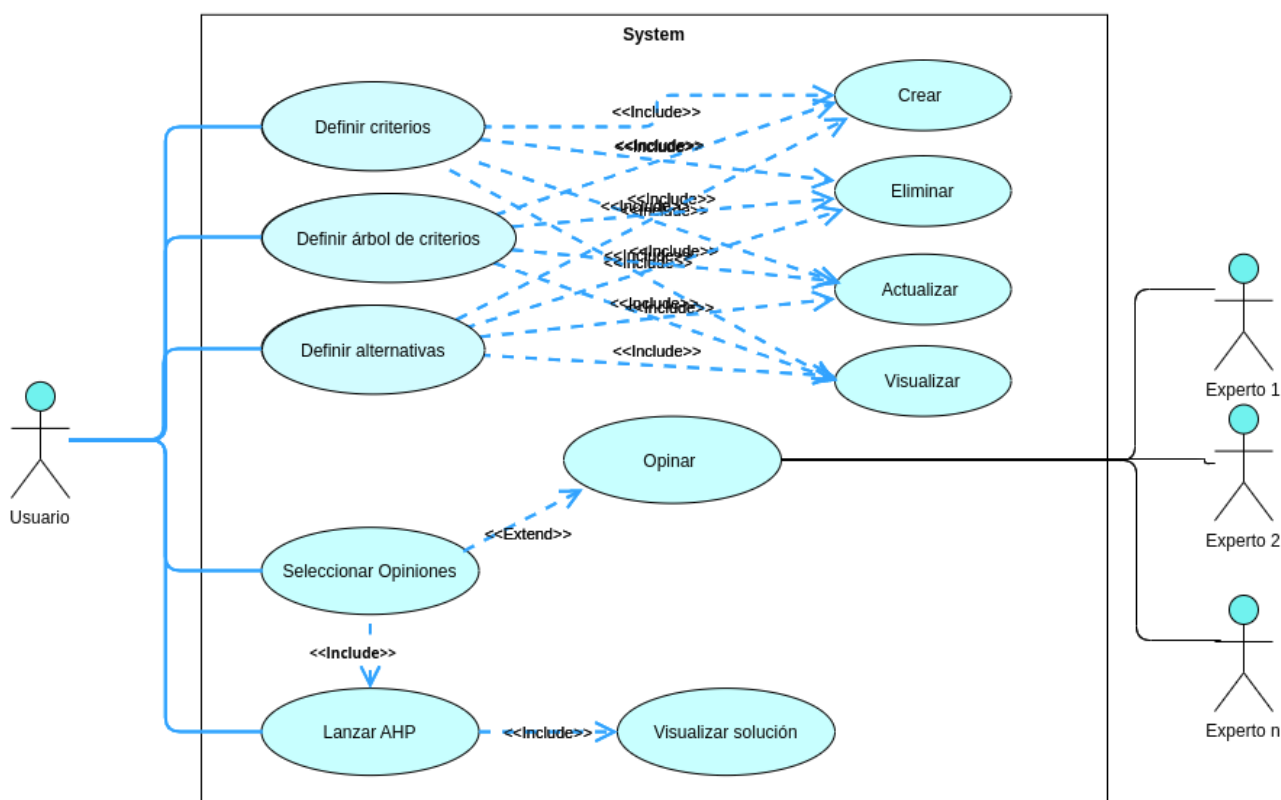


Figura 3.5: Diagrama de casos de uso del funcionamiento del front-end.

3.3.3. AHP

Este módulo es el encargado, de una vez recibido el conjunto de datos en el back-end, lanzar el proceso analítico jerárquico, implementado en el directorio `/ahp` mediante el lenguaje de programación *python*. En cuanto a la estructura, esta viene representada en la Figura 3.6, de la que podemos destacar un programa principal (`main.py`) encargado de recibir como parámetros, en formato JSON, datos sobre el árbol, los expertos y las alternativas. Estos parámetros serán pasados al constructor de la clase implementada en el directorio `ahp.py`, creando un objeto, el cual será utilizado para aplicación de cada una de las fases descritas en la sección 2.3.3. Estas fases han sido implementadas como métodos de la clase AHP, que tras su ejecución en el orden correcto, generan como salida la solución el conjunto de alternativas ponderadas.



Figura 3.6: Estructura del módulo AHP

Capítulo 4

Funcionamiento

4.1. Despliegue

Para el despliegue de nuestra aplicación tenemos dos vías:

4.1.1. Despliegue de manera local

Esta primera manera es la utilizada en durante el desarrollo del proyecto, puesto que permite observar los cambios realizados en el mismo instante en el que se realizan. Para ello, primero debemos desplegar nuestra API-REST, mediante el Comando 4.1, este debe ejecutar dentro del directorio `/back-end`. Posteriormente, se pone en marcha la interfaz de usuario, mediante el comando 4.2, desde el directorio `/front-end`. Una vez ambos se encuentren desplegados podremos acceder a la dirección `http://localhost:8081`, correspondiente a la interfaz de usuario, y verificar si todo está en orden.

```
1 $ mvn spring-boot:run
```

Listing 4.1: Comando para despliegue del back-end

```
1 $ yarn serve
```

Listing 4.2: Comando para despliegue del front-end

4.1.2. Despliegue haciendo uso de Docker

Esta segunda manera trata de desplegar la aplicación haciendo uso de los contenedores de Docker. Para ello, se deben seguir los siguientes pasos:

- 1) Dockerización del módulo del back-end y del front-end, mediante la creación de un Dockerfile, en cada uno de ellos.

- 2) Desarrollo de las *GitHub Actions* encargadas de la creación de las imágenes Docker de la aplicación, cada vez que se realice un commit sobre las carpetas `/front-end` y `/back-end`. Además de la creación de las imágenes, estas acciones también serán las encargadas de publicar en *Docker Hub*, las correspondientes imágenes.
- 3) Creación del *docker-compose.yml*, encargado de definir el conjunto de contenedores que compondrán nuestra aplicación y crearlos de una sola vez. En este caso, la ejecución del comando `docker-compose up`, permitirá obtener las imágenes del front-end y back-end, del Docker Hub, además de la imagen ya predefinida de MongoDB.

Si todo va bien, en este momento tendríamos desplegada nuestra aplicación haciendo uso de los contenedores de Docker. Pudiendo acceder a las siguientes direcciones para verificar su correcto despliegue.

- Front-end: `localhost:8081`
- Back-end: `localhost:8080`
- MongoDB: `localhost:27017`

4.2. Funcionamiento

Una vez desplegada nuestra aplicación y posterior acceso a la dirección `localhost:8081`, toca verificar el correcto funcionamiento de cada una de las páginas.

4.2.1. Home

El usuario, una vez acceda a la dirección `localhost:8081`, verá una pantalla como la mostrada en la Figura 4.1. Desde esta página de inicio, podemos destacar varias partes:

- *Una cabecera*, desde la que poder acceder a cada una de las páginas principales de la aplicación.
- *Un menú desplegable*, en el lado izquierdo de la pantalla, desde el que acceder, al igual que desde la cabecera, a las páginas principales de la aplicación.
- *Un pie de página*, donde se muestra información de contacto con el autor.
- *Selector del idioma*, desplegable que hace uso de la librería *i18n* para ajustar el idioma de dicha página a inglés o español.

Todos estos son componentes que estarán presentes en todas las pantallas que se verán a continuación. Esto quiere decir que estos tres componentes han sido desarrollados en el archivo principal (`src/App.vue`).

Además de los elementos comunes comentados, desde la página de inicio, también se muestra información básica de nuestra aplicación, así como los pasos necesarios para llevar a cabo la resolución de un problema de toma de decisiones, haciendo uso de esta Web.

Decision Support System

Decision Support System

HOME STAGES CRITERIA EXPERTS

Home Stages Criteria Experts

We help you make the best decision !

This software, in the form of a free web application, and accessible to any type of user with an internet connection, has been developed with the aim of providing companies or users with a simple and generic decision support system, on which they can rely when making all kinds of decisions.

What is AHP?

The AHP, or Analytical Hierarchy Process, is the technique implemented in this application for decision making. This is a quantitative method for making multi-criteria decisions that allows the generation of priority scales based on the judgement of experts, expressed through pairwise comparisons, using a preference scale.

User manual:

1° Define Criteria
In this first step, the set of criteria must be defined, which will represent the set of qualities to be assessed in order to make the decision.

2° Define Tree
In this second step, the user has to create a tree that represents the hierarchical structure necessary for the application of the AHP. For the generation of this structure, it is sufficient to define the name of the tree and the criterion that will act as the root of the tree.

3° Define Alternatives
In this third step, it is now possible to visualise the tree defined in the previous step, and to add to it the set of alternatives on which the solution is to be obtained.

4° Define Experts
In this fourth step, the user must define the set of experts who will carry out the evaluation process of the pairs of criteria that make up a given tree.

5° Opine
In this fifth step, the user must select the set of opinions previously made by the experts, and assign a certain weight to each of them, in this decision-making process.

6° Run AHP
Finally, if all has gone well, the solution obtained can be seen as a set of alternatives ordered according to the overall priority calculated by the AHP, as well as the priority for each alternative, according to each of the experts.

TFG - Sergio Leopoldo Benítez Delgado

2022 - AHP





















Figura 4.1: Pantalla de inicio.

Crterios

Desde esta pantalla, mostrada en la Figura 4.2, el usuario podrá visualizar todos los criterios almacenados en la base de datos. Pudiendo realizar operaciones como:

- *Consultar* de un determinado criterio: haciendo uso del icono del ojo.
- *Eliminar* de un determinado criterio: haciendo uso del icono rojo de la papelera, disponible al consultar un determinado criterio.
- *Modificar* de un determinado criterio: haciendo uso del icono del lápiz.
- *Añadir* un nuevo criterio: también es posible añadir un nuevo criterio, pinchando en el botón "New Criterion".



Available criteria:

Search by Name, Description, N° of chil... <input type="text"/>				SEARCH BY ID	NEW CRITERION
Name ↑	Description	Childrens	Actions		
Altura	Altura en cm	0	 		
Asistencias	Asistencias	0	 		
Cabeza	Goles de cabeza	0	 		
Contrato	Numero de años de contrato	0	 		
Defensa	Mejor centrocampista	2	 		
Delantero	Seleccionar el mejor delantero	3	 		
Deportivas	Infracciones deportivas	2	 		
Derecho	Goles marcados con la pierna deracha	0	 		
Directa	Implicación directa en el juego por parte del jugador	4	 		
Duelos Aéreos	Duelos ganados	0	 		

Rows per page: 10 1-10 of 28 < >

Figura 4.2: Pantalla para el manejo de criterios.


Todas estas operaciones hacen uso de un formulario, como el mostrado en la Figura 4.3. En este formulario, dependiendo del tipo de operación seleccionada, aparecerá ya relleno con los datos, en el caso de que se haya seleccionado la opción de consultar o modificar, y vacío para la introducción de datos, si se ha seleccionado opción de añadir.


Criterion
 


Id
62bd7b4cd01c1f351bcbe7e3


Name
Delantero 9 / 100

Description
Seleccionar el mejor delantero 30 / 2000

Childrens 

Name
Goles 

Name
Personales 

Name
Implicación 

SAVE

Figura 4.3: Pantalla para el mostrar, modificar, eliminar o añadir un criterio.

4.2.2. Escenarios

Pantalla para el manejo de árboles, mostrada en la Figura 4.4, donde al igual que en la pantalla anterior, se dispone de una tabla que muestra todos los árboles disponibles con la posibilidad de consultarlo, eliminarlo, modificarlo y añadir uno nuevo. Todas estas opciones siguen un esquema muy similar al del formulario de la Figura 4.3, pero adaptado a los atributos de un árbol.

Name	Description	Root	Actions
Mejor centrocampista	2022/2023	Defensa	
Árbol de prueba	1.0	GolesPrueba	
Mejor delantero	Elegir el mejor delantero 2021/2022	Delantero	

Rows per page: 5 1-3 of 3

Figura 4.4: Pantalla para el manejo de árboles.

Además de las opciones comentadas anteriormente, en esta pantalla también se tiene las siguientes dos opciones:

- **Visualizar árbol:** pinchando sobre el icono con forma de árbol, se redirigirá a una nueva vista, como la mostrada en la Figura 4.5. En esta nueva vista se mostrará gráficamente el árbol seleccionado y el conjunto de alternativas asociadas a este árbol. Para la representación gráfica del árbol, se hizo uso de una librería llamada `@ssthouse/vue-tree-chart`, la cual permite el uso de un componente llamado `<vue-tree>`, y como observamos en el fragmento de Código 4.3, pasando como parámetros: el estilo, un dataset a representar y la configuración, representará en forma de árbol el dataset. Dicho dataset debe seguir el formato presentado en el código 4.4. Además, desde esta pantalla, el usuario podrá obtener información de cada criterio que conforman el árbol pinchando sobre el nodo a consultar, también podrá añadir, modificar, actualizar y visualizar las alternativas asociadas al árbol que está visualizando.
- **Lanzar AHP:** pinchando sobre el icono en forma de círculo con la palabra AHP en su interior, se redirigirá a una pantalla en la que seleccionar las opiniones necesarias para lanzar el AHP. Esta pantalla será explicada más en detalle posteriormente.



Figura 4.5: Pantalla para la visualización gráfica de los árboles.

```

1 <template>
2   <vue-tree
3     style="width: 1000px; height: 600px; border: 1px solid gray;"
4     :dataset="defaultTree"
5     :config="treeConfig"
6   >
7     <template v-slot:node="{ node, collapsed }">
8       <div class="rich-media-node" :style="{ border: collapsed ? '2px solid grey' :
9         '' }" @click="onclickNode(node)">
10        <span style="padding: 4px 0; font-weight: bold;">{{node.name}}</span>
11      </div>
12    </template>
13  </vue-tree>
</template>
  
```

Listing 4.3: Utilización del componente <vue-tree>

```

1 {
2   "id": "628b79055447e762d2eea50e",
  
```

```

3  "name": "Centrocampista",
4  "description": "",
5  "children": [
6      {
7          "id": "6267d3af7347197ff638e96d",
8          "name": "Falta",
9          "description": "Goles de falta",
10         "children": []
11     },
12     {
13         "id": "628b79445447e762d2eea50f",
14         "name": "Tarjetas",
15         "description": "Media de tarjetas",
16         "children": []
17     },
18     {
19         "id": "628b79735447e762d2eea510",
20         "name": "Asistencias",
21         "description": "Asistencias ultima temporada",
22         "children": []
23     }
24 ]
25 }

```

Listing 4.4: JSON del árbol a representar

4.2.3. Expertos

Una vez se pueda crear y visualizar un determinado árbol, es hora de poder realizar la valoración del conjunto de nodos de dicho árbol, Y es aquí donde entran en juego los expertos. Para ello, en la pantalla, mostrada en la Figura 4.6, se puede observar como se encuentran listados cada uno de los expertos, y que cada uno de ellos tiene las siguientes opciones:

- *Editar*: ofrece la opción de editar la información asociada a un experto.
- *Eliminar*: posibilidad de eliminar de la base de datos a un experto.
- *Opinar*: esta opción nos redirigirá a la pantalla mostrada en la Figura 4.7, en la que el usuario podrá seleccionar el árbol sobre el que desea opinar, y posteriormente realizar la valoración de cada uno de los pares de criterios, mediante un *slider* y con una escala de [-10, 10], analizando antes de guardar dicha opinión, la consistencia de la misma.
- *Opiniones*: esta opción nos redirigirá a una pantalla desde la cual podremos observar todas las opiniones realizadas por un determinado experto, y pinchando en un icono con forma de pesa, nos redirigirá a una nueva vista, como la mostrada en la Figura

4.8, en la que se mostrará el árbol pesado, correspondiente a una determinada opinión. De la representación gráfica del árbol, destacar dos datos: "Parcialz "Final", el primero de ellos representa el peso asignado por el experto a ese determinado criterio, mientras que el segundo representa al peso utilizado para la aplicación del AHP, y este es calculado como el valor del parcial del propio criterio, por el parcial de sus antecesores.

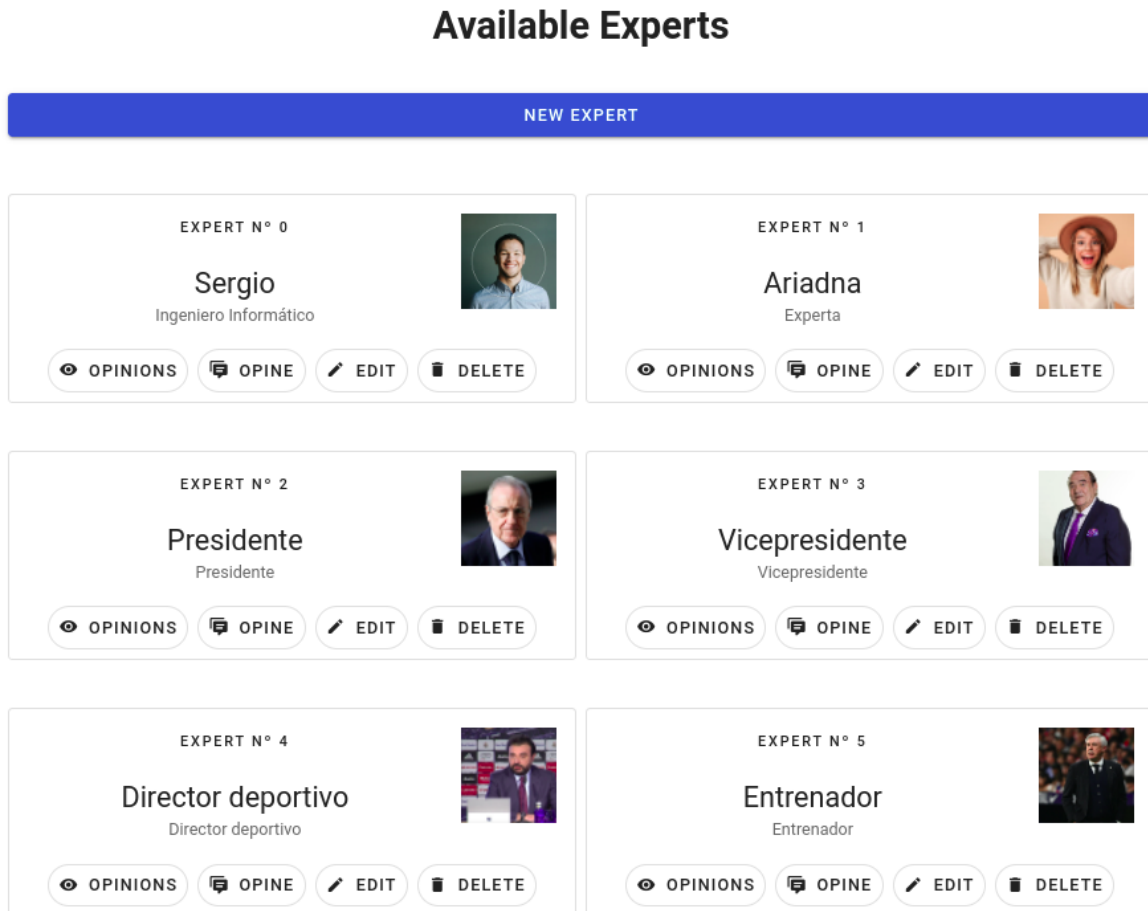


Figura 4.6: Pantalla para la definición de los expertos.

Available Trees:

Name	Actions
Mejor centrocampista	
Árbol de prueba	
Mejor delantero	

Rows per page: 5 1-3 of 3 < >

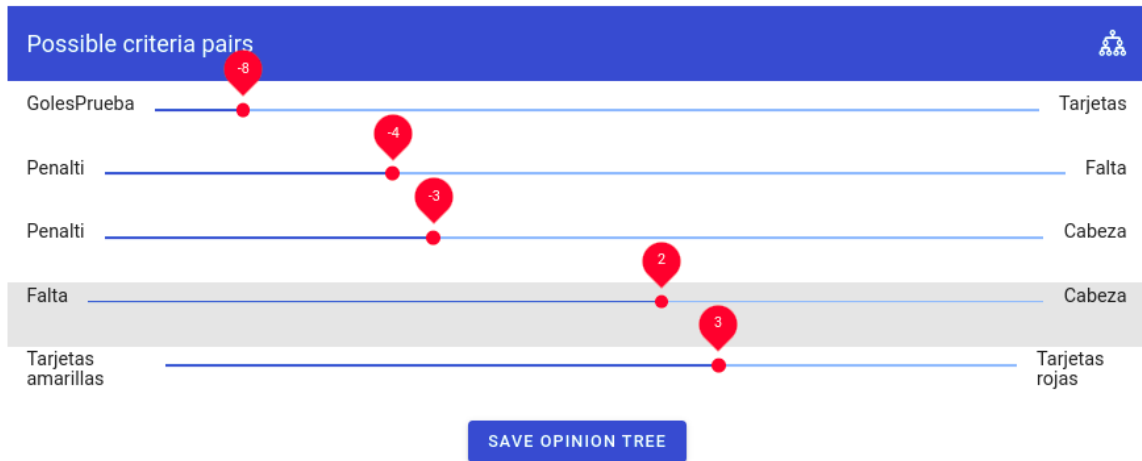


Figura 4.7: Pantalla para la realización de una opinión.

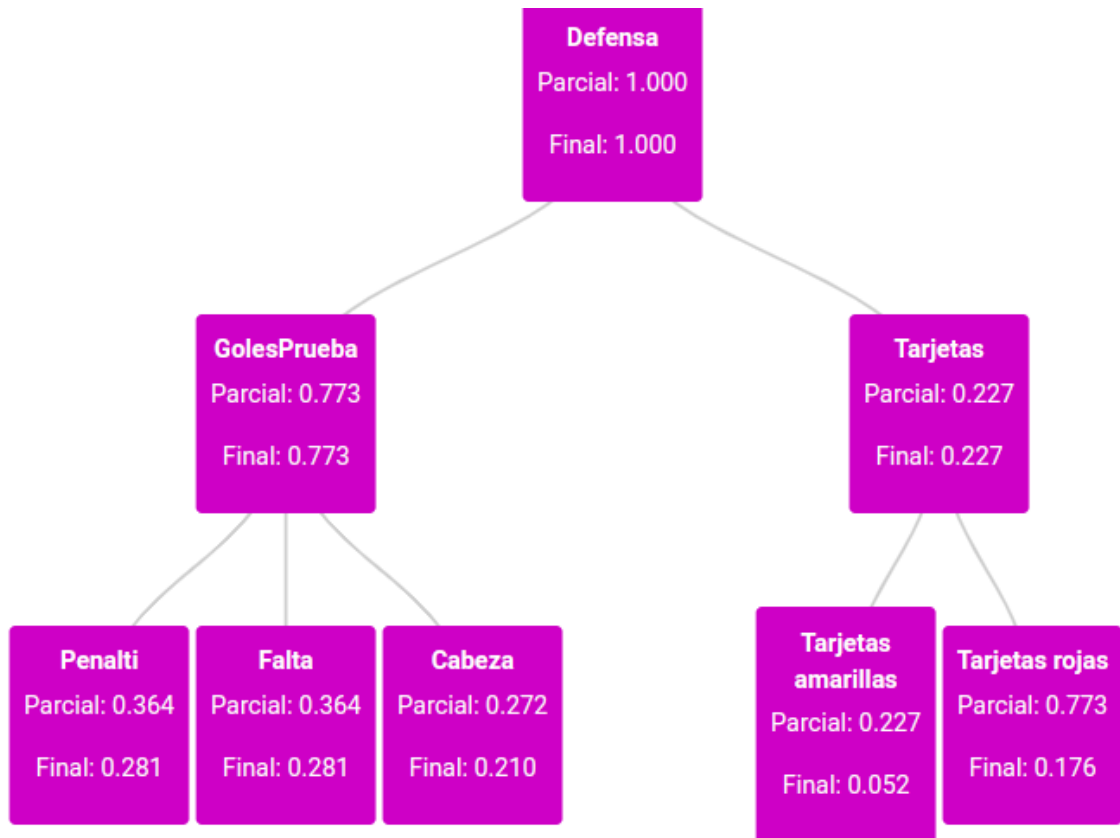


Figura 4.8: Pantalla para la visualización del árbol pesado.

4.2.4. Lanzamiento del AHP

Una vez completados los pasos de creación de los criterios, creación del árbol, para la posterior opinión de cada experto, llega la hora de aplicar el proceso analítico jerárquico para obtener las alternativas ponderadas. Para ello, desde la pantalla mostrada en la Figura 4.4, debemos pulsar sobre el icono violeta con la palabra "AHP" en su interior, para ser redirigidos a la pantalla de la Figura 4.9. Desde esta pantalla se nos presenta las opiniones disponibles para el árbol seleccionado, tras la selección de las que deseemos, se debe pulsar en "SELECT THESE OPINIONS". Cada opinión de las seleccionadas habrá sido realizada por un experto, es por ello que tras pinchar en ese botón, se desplegarán tantos *sliders* como expertos haya, para seleccionar así la importancia que tendrá cada experto dentro del Proceso Analítico Jerárquico. Esta peso, viene a determinar cuanto más de importante es la opinión de un experto frente a la opinión de otro.

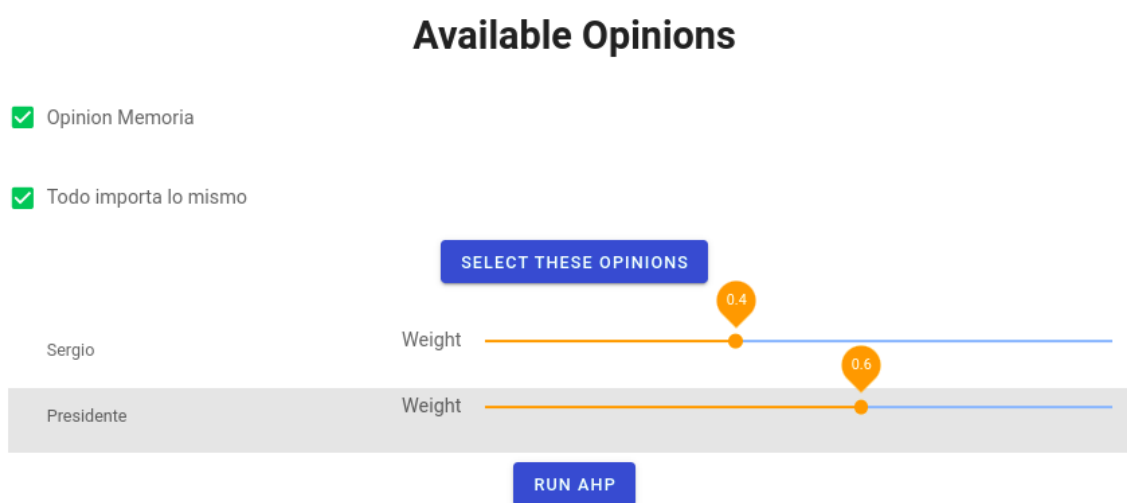


Figura 4.9: Pantalla para la selección de opiniones y pesos de los expertos.

Tras seleccionar las opiniones, ajustar los pesos de cada experto, es hora de lanzar el proceso analítico jerárquico, pinchando el botón "RUN AHP". Pinchar este botón hace que se inicie el flujo de ejecución del AHP, mostrado en la Figura 3.2. Obteniendo como respuesta el conjunto de alternativas factibles ponderadas según su importancia. La salida del AHP, es mostrada mediante un componente, al cual mediante sus *props*, se le pasa la respuesta obtenida por la API y el identificador del árbol. El identificador del árbol será utilizado para obtener las alternativas asociadas a dicho árbol, y una vez dispongamos de dichas alternativas, basta con recorrer la salida obtenida (con el formato representado en el JSON 4.5) y relacionar el campo "alternativeId" con las alternativas asociadas al árbol, y añadir a dicha alternativa un nuevo campo ("peso global"), con el valor asociado al campo "value". Además del nuevo campo con el peso global, a dicha alternativa también le será asociado el peso de obtenido para cada uno de los expertos que colaboraron en la toma de esta decisión. Obteniendo como resultado, el conjunto de alternativas ordenadas, según su peso global, y mostrado en la Figura 4.10. Toda esta información puede ser exportada pinchando sobre el botón "Generate PDF Report", generando mediante el uso de la librería jsPDF, un informe en formato PDF como el mostrado en la Figura 4.11.

```

1 {
2   "assessment": {
3     "alternatives": [
4       "alternative": {
5         "alternativeId": ""
6         "value": ""
7         "expertsAssessment": [
8           "expert": {
9             "expertId": ""
10            "value": ""
11          }
12        ]
13      }
14    ]
15  }
16 }

```

Listing 4.5: Formato JSON para la salida del algoritmo AHP

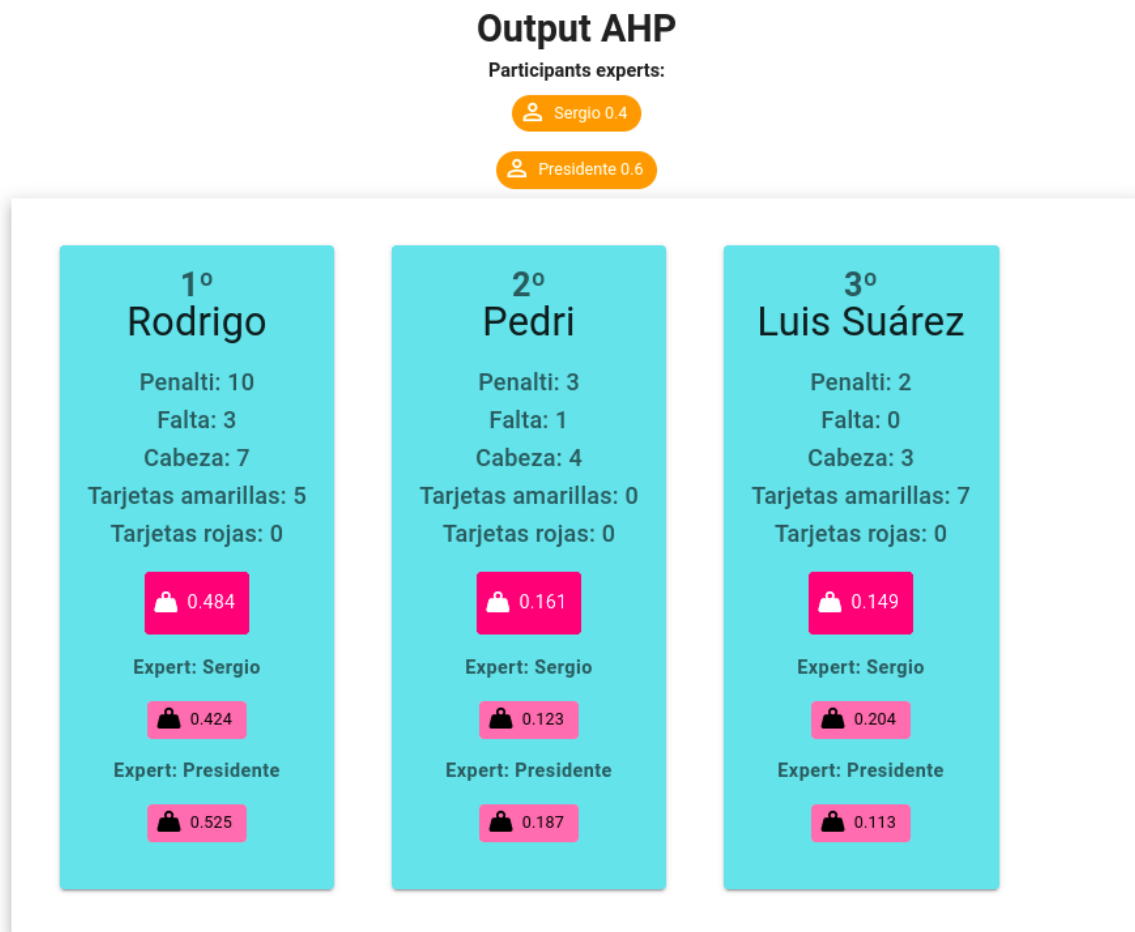


Figura 4.10: Componente para la visualización de la solución obtenida.

Software de Ayuda a la Decisión mediante el Proceso Analítico Jerárquico



Informe de resultados

Árbol:

Árbol seleccionado: Mejor defensa
Criterio raíz: Defensa

Expertos participantes:

Experto	Peso
Sergio	0.40
Presidente	0.60

Resultados:

Alternativas	Global	Sergio	Presidente
Rodrigo	0.484	0.424	0.525
Pedri	0.161	0.123	0.187
Luis Suárez	0.149	0.204	0.113

Informe generado por el Sistema de Ayuda a la decisión
Trabajo Fin de Grado 2021/2022
Autor: Sergio Leopoldo Benítez Delgado

Figura 4.11: Informe PDF con la información del proceso de toma de decisión.

Capítulo 5

Experimentación

En este capítulo se trata de exponer un caso real, como el de la *selección del jugador a fichar por un determinado equipo*, en el que sería de gran ayuda el uso de este software. Este problema surge cuando el entrenador de un determinado equipo le plantea a su presidente y a todos los mandatarios del club, el fichaje de un delantero para la próxima temporada. Una vez recibida esta propuesta, estos últimos se podrán en marcha para la búsqueda del delantero que más se acerque a sus preferencias. Sería en este punto, donde entraría en escena el “Software de Ayuda a la Decisión”. Teniendo esta decisión una gran importancia tanto en el aspecto económico, como en el deportivo. Pues la toma de una mala decisión, tras otra, podría desencadenar unas consecuencias muy perjudiciales en el ecosistema del club.

5.1. Procedimiento para selección del jugador

5.1.1. Selección de criterios

En este primer paso, los mandatarios del club han de seleccionar las cualidades, que han de valorar para tomar dicha decisión. En este caso, los criterios seleccionados son los representados en el árbol de la Figura 5.1

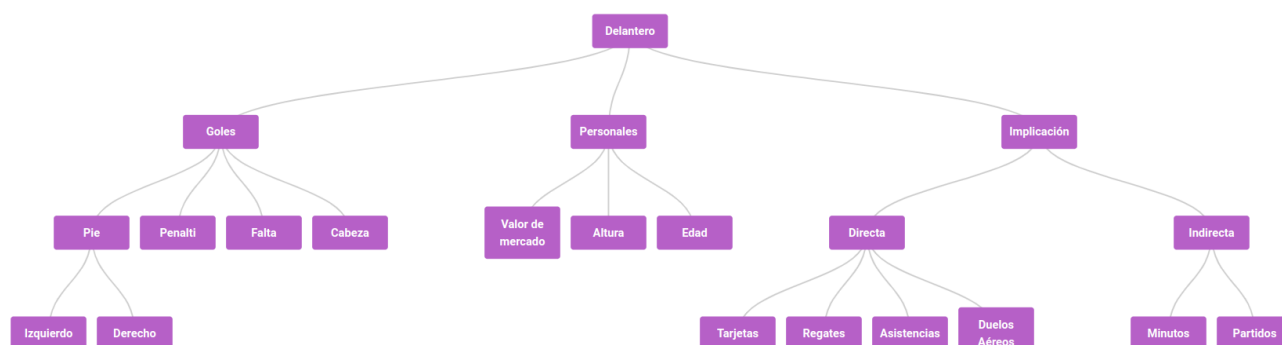


Figura 5.1: Estructura jerárquica para la selección de un delantero.

5.1.2. Selección de alternativas

Para poder tomar una decisión lo segundo que se necesita es la elección de las alternativas sobre las que poder tomar una decisión. Es por ello, que en este paso los mandatarios del club, deben de introducir en la aplicación el conjunto de jugadores, con los datos asociados a cada uno, que formarán el conjunto factible de alternativas con las que lograr el objetivo final.

Tras consultar el mercado de fichajes disponible, y los datos estadísticos de cada uno de ellos, las alternativas seleccionadas son las expuestas en la Tabla 5.1.

	Benzema	Joao Felix	Depay	Vinicius	RDT	Iago Aspas
Izquierdo	3	1	1	5	5	14
Derecho	20	6	10	10	10	4
Penalti	7	0	4	0	5	3
Falta	0	0	0	0	1	0
Cabeza	3	0	1	2	1	1
Tarjetas	0	1	3	6	5	8
Regates	36	31	45	95	50	51
Asistencias	12	4	2	10	3	6
Duelos Aéreos	44	10	20	20	38	30
Minutos	2596	1264	1847	2699	2923	3093
Partidos	32	24	28	35	34	37
Valor (M)	30	70	35	100	25	7
Altura	185	181	176	176	180	176
Edad	34	22	28	21	27	34

Tabla 5.1: Jugadores seleccionados (Alternativas).

5.2. Valoración de los expertos

Una vez se tenga se haya completado la jerarquía de la Figura 5.1, con las alternativas expuestas en la Tabla 5.1, se procede a la valoración por cada experto de cada uno de los pares de criterio que forman el árbol. Esta valoración será llevada a cabo, en este caso, por cuatro expertos, entre los que se encuentran el presidente, el vicepresidente, el director deportivo y el entrenador.

En las siguientes Tablas 5.2, 5.3 y 5.4, se muestran las matrices de comparaciones pareadas de generadas por las valoraciones realizadas por el presidente para cada uno de los niveles de la jerarquía, teniendo que repetir este proceso recorriendo el árbol desde el nodo objetivo hasta los nodos hijos, y a su vez repetirlo por cada uno de los expertos. Cada componente a_{ij} de las matrices pareadas deben estar en el rango [1-9].

Tras la valoración de todos los pares, es posible observar el árbol pesado de dicha opinión en la Figura 5.2. De este árbol se puede destacar dos valores, el primero de ellos representa la ponderación de un criterio respecto a su padre, y el otro valor representa la

Delantero	Goles	Implicación	Personales	Prioridades
Goles	1	3.4	1.8	0.54
Implicación	0.29	1	0.55	0.16
Personales	0.55	1.79	1	0.29

Tabla 5.2: Matriz comparaciones pareadas (Criterios 1º Nivel).

Goles	Pie	Penalti	Falta	Cabeza	Prioridades
Pie	1	1.8	0.55	1	0.24
Penalti	0.55	1	0.55	0.55	0.15
Falta	1.79	1.79	1	1	0.32
Cabeza	1	1.79	1	1	0.27

Tabla 5.3: Matriz comparaciones pareadas (Criterios 2º Nivel).

Pie	Izquierdo	Derecho	Prioridades
Izquierdo	1	1	0.5
Derecho	1	1	0.5

Tabla 5.4: Matriz comparaciones pareadas (Criterios 3º Nivel).

ponderación final respecto al árbol completo, siendo esta ponderación la utilizada en el Proceso Analítico Jerárquico.

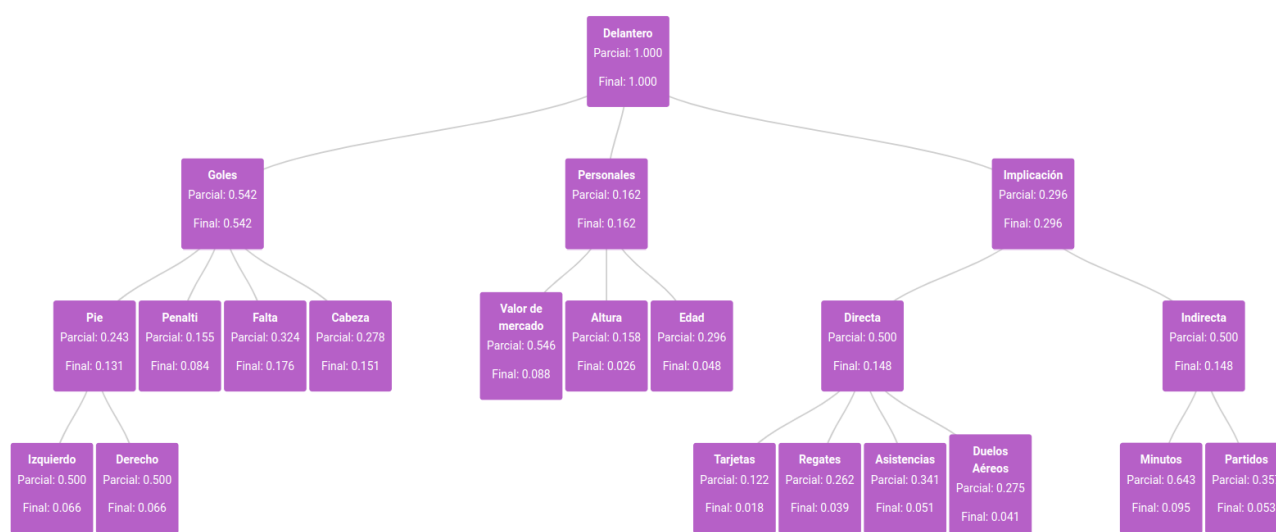


Figura 5.2: Estructura jerárquica ponderada por el presidente.

Una vez visualizada la estructura jerárquica de la imagen anterior, se puede obtener los pesos finales de cada uno de sus nodos hojas, obteniendo como resultado la Tabla 5.5.

Criterio	Peso Final
Izquierdo	0.066
Penalti	0.066
Falta	0.084
Cabeza	0.176
Valor de mercado	0.151
Altura	0.088
Edad	0.026
Derecho	0.048
Tarjetas	0.018
Regates	0.039
Asistencias	0.051
Duelos Aéreos	0.041
Minutos	0.095
Partidos	0.053
Total	1

Tabla 5.5: Vector de prioridades asignadas por el presidente.

5.3. Selección de opiniones

En este paso, se deben de seleccionar las opiniones realizadas anteriormente por cada uno de los expertos, y asignarle el peso que tendrá cada experto en esta decisión. En este caso los pesos asignados son:

- *Presidente*: peso de 0.5, siendo el de mayor peso en esta decisión, al tratarse de la máxima autoridad del club.
- *Vicepresidente*: peso de 0.2.
- *Director deportivo*: peso de 0.15.
- *Entrenador*: peso de 0.15.

Una vez seleccionadas las opiniones de cada uno de ellos, y ajustados sus pesos, como se muestra en la Figura 5.3, se lanzará el Proceso Analítico Jerárquico para obtener la solución a dicho problema.

La solución obtenida por parte de esta aplicación será el conjunto de jugadores, ordenados según su prioridad, siendo el primero, el jugador con mayor prioridad para ser fichado. En este caso, y observando la Figura 5.4, podemos concluir que el jugador con mayor prioridad de ser fichado es *Raúl de Tomás* (RDT), con una prioridad del 0.292, lo que equivale a un 29.2%. Dicha solución, recogida en un informe, puede ser exportada en formato PDF pinchando sobre el botón "Generate PDF Report", generando mediante el uso de la librería jsPDF, un informe como el mostrado en la Figura 5.5.

Available Opinions

- Opinion Presidente
- Opinion Vicepresidente
- Opinión Director deportivo
- Opinión Entrenador

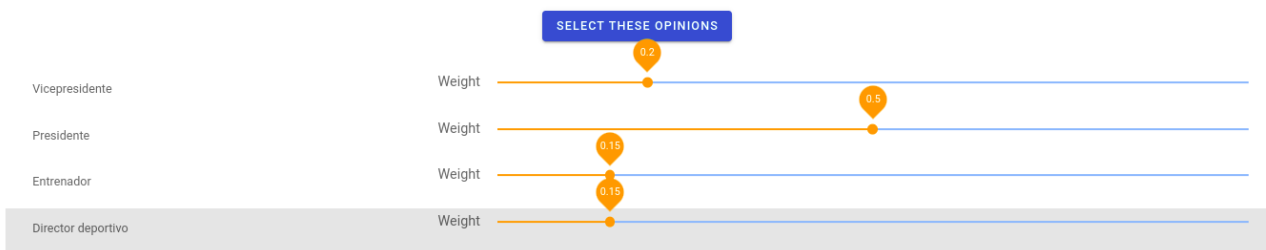


Figura 5.3: Selección de opiniones y ajuste del peso de los expertos.

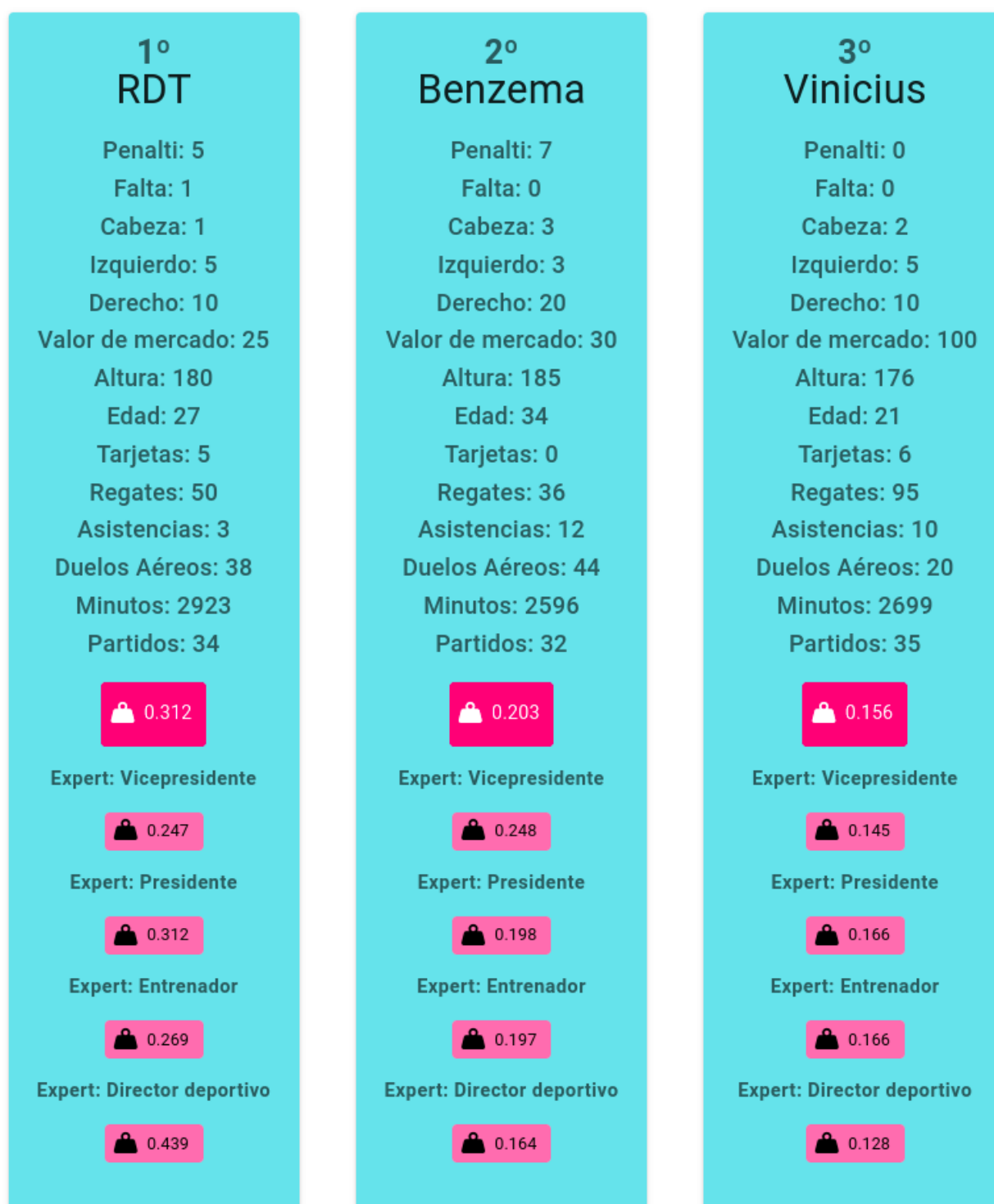


Figura 5.4: Solución al proceso de selección de un jugador.

Software de Ayuda a la Decisión mediante el Proceso Analítico Jerárquico



Informe de resultados

Árbol:

Árbol seleccionado: Mejor delantero

Criterio raíz: Delantero

Expertos participantes:

Experto	Peso
Vicepresidente	0.20
Presidente	0.50
Entrenador	0.15
Director deportivo	0.15

Resultados:

Alternativas	Global	Vicepresidente	Presidente	Entrenador	Director deportivo
RDT	0.312	0.247	0.312	0.269	0.439
Benzema	0.203	0.248	0.198	0.197	0.164
Vinicius	0.156	0.145	0.166	0.166	0.128
Iago Aspas	0.155	0.141	0.144	0.237	0.133
Depay	0.110	0.145	0.109	0.085	0.087
Joao Felix	0.064	0.073	0.070	0.047	0.048

Informe generado por el Sistema de Ayuda a la decisión
Trabajo Fin de Grado 2021/2022
Autor: Sergio Leopoldo Benítez Delgado

Figura 5.5: Informe PDF con la información del proceso de selección de un delantero.

Capítulo 6

Presupuesto

Este capítulo trata de presentar el presupuesto estimado para el desarrollo de este Trabajo Fin de Grado, teniendo en cuenta los honorarios de los desarrolladores y los gastos en material. Por ello, este presupuesto se ha desglosado en tres apartados diferentes, distinguiendo entre los costes de las infraestructuras tecnológicas, los costes de personal para desarrollo, y el coste total.

6.1. Infraestructuras tecnológicas

Respecto al presupuesto necesario para paliar los gastos producidos por el pago de materiales y/o equipos utilizados, debemos destacar que los utilizados para el desarrollo son un ordenador portátil, un segundo monitor, además de recursos gratuitos, en este caso, como los repositorios de *Docker Hub* cedidos por los tutores y el repositorio de *GitHub*.

Descripción	Precio de compra	Tiempo de uso	Amortización	Coste
Ordenador portátil	900€	4 meses	3 años	139€
Monitor	100€	4 meses	3 años	16€
Coste total				155€

Tabla 6.1: Coste recursos materiales

6.2. Recursos humanos

Respecto a la mano de obra empleada, para su cálculo se ha consultado la página web *Talent.com*¹ que actúa como buscador de empleos, además de permitir obtener el salario promedio de este año 2022, en nuestro caso para un desarrollador full stack en España, y nos dice que es de 35.000€ al año o 17,95€ por hora.

¹<https://es.talent.com/salary?job=full>

Tarea	Horas	Coste hora	Coste Total
Documentación	80	17.95€	1.436€
Formación	20	17.95€	359€
Front-end	180	17.95€	3.231€
Back-end	150	17.95€	2.692€
AHP	30	17.95€	538€
Coste total			8.256€

Tabla 6.2: Coste recursos humanos

6.3. Coste total

Recursos	Costes
Infraestructuras tecnológicas	155€
Recursos humanos	8.256€
Coste total	8.411€

Tabla 6.3: Coste recursos humanos

Capítulo 7

Conclusiones

Al concluir este proyecto se cuenta con una aplicación web que permite resolver una gran multitud de casos de toma de decisión multicriterio, consiguiendo en esta un comportamiento completamente genérico en cuanto a los ámbitos de aplicación se refiere. Este caso de decisión multicriterio es resuelto en base a una serie de alternativas y aplicando la técnica del Proceso Analítico Jerárquico, siendo capaz de obtener como solución un conjunto de alternativas ponderadas y generar un informe con los datos asociados a la solución obtenida.

Esta aplicación web ha sido dividida en dos componentes principales, un front-end y un back-end, desarrollados con dos frameworks diferentes: el primero, el front-end, ha sido desarrollada con el framework VueJS y su librería Vuetify, mientras que el back-end, ha sido desarrollado con el framework Spring. Ambos completamente nuevos para mí, suponiendo, por lo tanto, el aprendizaje de estos uno de los principales retos de este proyecto, así como el trabajo en una de las partes que me llamaba bastante la atención, como es el desarrollo Full Stack.

En este proceso de desarrollo de una aplicación Full Stack, se han ido encontrando varias dificultades, muchas de las cuales vinculadas con la novedad que suponen la gran mayoría de las tecnologías y herramientas utilizadas. Todas estas dificultades han sido superadas, consiguiendo ampliar los conocimientos y la visión de este mundo tan amplio como es la Ingeniería Informática.

7.1. Líneas Futuras

Las líneas futuras de este proyecto están orientadas sobre todo a la finalización y mejora de algunas características no acabadas, así como añadir nuevas características no consideradas en el desarrollo de este TFG. Las propuestas de mejora para el futuro son:

- *Introducción de datos en crudo*: añadir opción para introducir datos en crudo, como podría ser: los criterios, estructura jerárquica y alternativas.
- *Toma de decisiones en línea*: permitir a los usuarios registrarse en la plataforma,

y formar un grupo, donde cada usuario represente a un determinado experto del proceso de valoración de los criterios.

- *Aplicación móvil*: adaptar la aplicación para su funcionamiento como aplicación móvil.

Capítulo 8

Conclusions

At the end of this project, we have a web application that allows to solve many cases of multi-criteria decision making, this application with a completely generic behavior concerning the application areas. The multi-criteria decision case is solved based on a series of alternatives and applying the Analytic Hierarchy Process, which is able to obtain a solution with a set of weighted alternatives and also generates a report with the data associated with the solution obtained.

This application has been divided into two main components, a front-end and a back-end, each of them developed with two different frameworks. The first component, the front-end, has been developed with the Vue framework and its library Vuetify, while the back-end has been developed using Spring framework. Both frameworks were a novelty for me, which implies that learning how to use them was one of the main challenges of this project. Besides, working on a Full Stack development in this project was one of the main motivations of this project.

Several difficulties have appeared when developing the Full Stack application, most part of them related with the novelty of using several technologies and tools. All these difficulties have been surpassed, increasing my knowledge and my vision of the world as a Computer Engineer.

8.1. Future Lines

The future lines of this project are mainly oriented to the completion and improvement of some unfinished features, as well as to add new features not considered in the development of this TFG. The proposals for future improvements are the following:

- *Raw data input*: add an option to import input data on raw format, such as: criteria, hierarchical structure and alternatives.
- *Online decision making*: allow users to register on the platform and form groups, where each user represents a specific expert in the criteria assessment process.

- *Mobile application*: adapting the application for working on mobile devices.

Bibliografía

- [1] E.F. De Amorós. La elección de carrera: una decisión de gran trascendencia. *Educación*, 2(3):5–13, 1993.
- [2] S. Barba-Romero. Panorámica actual de la decisión multicriterio discreta. *Investigaciones Económicas*, 11(2):279–308, Mayo 1987.
- [3] K.D. Goepel. Implementation of an online software tool for the analytic hierarchy process (ahp-os). *International Journal of the Analytic Hierarchy Process*, 10(3), 2018.
- [4] F. Aguiar González. Teoría de la decisión e incertidumbre: modelos normativos y descriptivos. *Empiria. Revista De metodología De Ciencias Sociales*, 8:139–160, 2004.
- [5] J.C. Osorio Gómez and J.P. Orejuela Cabrera. El proceso de análisis jerárquico (ahp) y la toma de decisiones multicriterio. ejemplo de aplicación. *Scientia Et Technical*, 2(39):1, Septiembre 2008.
- [6] J.M. Moreno Jiménez. El proceso analítico jerárquico (ahp). fundamentos, metodología aplicaciones. *Dpto. Métodos Estadísticos. Facultad de Económicas. Universidad de Zaragoza*.
- [7] J.M Sánchez Lozano, A. Dogarganes Nodar, and A. Sánchez De Miguel. Criterios de decisión para el análisis de la contaminación lumínica en España. un enfoque ahp. *Centro Universitario de la Defensa de San Javier. Sección de Técnicas de Apoyo a la Decisión. University of Exeter*, 2021.
- [8] A. Mendoza, C. Solano, D. Palencia, and D. Garcia. Aplicación del proceso de jerarquía analítica (ahp) para la toma de decisión con juicios de expertos. *Ingeniare. Revista chilena de ingeniería*, 27(3):348–360, 2019.
- [9] T. Olavsrud. Decision support systems: Sifting data for better business decisions. 2020.
- [10] S. Berumen & F. Llamazares Redondo. La utilidad de los métodos de decisión multicriterio (como el ahp) en un entorno de competitividad creciente. *Cuadernos de Administración*, 20:65 – 87, 12 2007.
- [11] T.L. Saaty. The analytic hierarchy process: Planning, priority setting, resource allocation. 1980.
- [12] T.L. Saaty. How to make a decision: The analytic hierarchy process. *INFORMS Journal on Applied Analytics*, 24(6):19–43, Diciembre 1994.

- [13] T.L. Saaty, P. Rogers, and R. Pell. Portfolio selection through hierarchies. *Journal of Portfolio Management*, 6(3):16–21, 1988.
- [14] R. Sánchez. La toma de decisiones con múltiples criterios. un resumen conceptual y teórico. *Centro de Planificación y gestión, Universidad Mayor de San Simón*, 2001.
- [15] B. Vitoriano. Teoría de la decisión: Decisión con incertidumbre, decisión multicriterio y teoría de juegos. *Universidad Complutense de Madrid*, 107, 2007.