



**UNIVERSIDAD
DE GRANADA**

**TRABAJO FIN DE GRADO
INGENIERÍA INFORMÁTICA**

**Estudio e implementación de algoritmos para la agregación
de rankings**

Autor

Carmen Quiles Ramírez

Directores

David Alejandro Pelta Mochcovsky



**ESCUELA TÉCNICA SUPERIOR DE INGENIERÍAS INFORMÁTICA Y DE
TELECOMUNICACIÓN**

Granada, 13 de Junio de 2025

Estudio e implementación de algoritmos para la agregación de rankings

Carmen Quiles Ramírez

Palabras clave: ranking, algoritmos, agregación de rankings, comparación, consenso.

Resumen

Los rankings son herramientas esenciales para representar preferencias, relevancia o calidad percibida en multitud de contextos, desde motores de búsqueda hasta toma de decisiones colaborativas.

En este Trabajo Fin de Grado se lleva a cabo un estudio teórico y práctico de varios algoritmos de agregación de rankings, que permiten fusionar múltiples ordenaciones individuales en un único ranking de consenso. Se analizan métodos clásicos como Borda, Copeland o Kemeny-Young, así como enfoques más recientes basados en grafos o cadenas de Markov.

Además, se desarrolla una herramienta software que implementa estos algoritmos, permitiendo su comparación y análisis empírico. Finalmente, se ejemplifica su uso mediante un caso de estudio relacionado con los Objetivos de Desarrollo Sostenible (ODS).

El proyecto completo y el código fuente pueden consultarse en el siguiente repositorio: <https://github.com/carmenqr/TFG>

Study and Implementation of Algorithms for Ranking Aggregation

Carmen Quiles Ramírez

Keywords: ranking, algorithm, rank aggregation, consensus, comparison

Abstract

Rankings are essential tools for representing preferences, relevance, or perceived quality in many contexts, from search engines to collaborative decision-making.

This Bachelor's Thesis presents a theoretical and practical study of several ranking aggregation algorithms, which aim to merge multiple individual orderings into a single consensus ranking. Classical methods such as Borda, Copeland, and Kemeny-Young are analyzed, along with more recent approaches based on graphs and Markov chains.

Furthermore, a software tool is developed to implement these algorithms, allowing for empirical comparison and analysis. The usefulness of the tool is demonstrated through a case study related to the Sustainable Development Goals (SDGs).

The complete project and source code are available at: <https://github.com/carmenqr/TFG>

Yo, **Carmen Quiles Ramírez**, alumno de la titulación Grado en Ingeniería Informática de la **Escuela Técnica Superior de Ingenierías Informática y de Telecomunicación de la Universidad de Granada**, con DNI 77391248C, autorizo la ubicación de la siguiente copia de mi Trabajo Fin de Grado en la biblioteca del centro para que pueda ser consultada por las personas que lo deseen.

Fdo: Carmen Quiles Ramírez

Granada a 13 de Junio de 2025.

D. **David Alejandro Pelta Mochcovsky**, Profesor del Departamento de Ciencias de la Computación e Inteligencia Artificial de la Universidad de Granada.

Informan:

Que el presente trabajo, titulado ***Estudio e implementación de algoritmos para la agregación de rankings***, ha sido realizado bajo su supervisión por **Carmen Quiles Ramírez**, y autorizamos la defensa de dicho trabajo ante el tribunal que corresponda.

Y para que conste, expiden y firman el presente informe en Granada a 13 de Junio de 2025.

El director:

David Alejandro Pelta Mochcovsky

Agradecimientos

En primer lugar, quiero agradecer a mi tutor, David Alejandro Pelta, por su apoyo constante durante la realización de este Trabajo de Fin de Grado. Gracias por confiar en mí desde el principio, por guiarme con paciencia y motivarme a seguir adelante en cada fase del proyecto. Ha sido un verdadero placer trabajar contigo y aprender tanto a lo largo del camino.

Pero, sin duda, mi mayor agradecimiento es para mis padres. Mamá, papá: gracias por vuestro apoyo constante, por la confianza y todo el cariño que siempre me habéis dado. Gracias a ello, todo este camino ha sido mucho más fácil y llevadero. Siempre me habéis enseñado que con trabajo y esfuerzo se puede conseguir cualquier meta, y una vez más teníais razón: todo esfuerzo tiene su recompensa. Sé que con vuestro apoyo, como siempre, podré llegar a donde me proponga. Gracias de corazón. Os quiero.

Índice general

1. Introducción	1
1.1. ¿Qué es un ranking?	1
1.2. Aplicaciones actuales de la agregación de rankings	3
1.3. Objetivos del trabajo	5
1.4. Temporización del trabajo	5
2. El problema de agregación de rankings	7
2.1. Definición	7
2.2. Medidas para la comparación de rankings	8
2.2.1. Distancia de Kendall	8
2.2.2. Coeficiente de correlación de Spearman	10
2.2.3. Coeficiente WS	12
2.3. Variantes del Problema de Agregación de Rankings	14
2.3.1. Rankings Estrictamente Lineales	14
2.3.2. Rankings con Empate	15
2.3.3. Rankings con Ponderación	16
3. Algoritmos de Agregación de Rankings	19
3.1. Método de Borda	20
3.2. Método de Copeland	24
3.3. Método de Kemeny-Young	28
3.4. Método de Schulze	31
3.5. Método de Footrule	36
3.6. Método basado en Cadenas de Markov	41
3.7. Comparación de los Algoritmos	45
4. Herramienta para el análisis y la comparación de algoritmos	49
4.1. Funcionalidad	49
4.1.1. Manual de usuario	50
4.1.2. Instalación	60
4.2. Tecnologías utilizadas	61

5. Caso de estudio	63
5.1. Planteamiento del problema: Cumplimiento de los Objetivos de Desarrollo Sostenible (ODS)	63
5.2. Resolución del problema	66
5.3. Análisis de resultados	68
5.4. Conclusiones	83
6. Conclusiones y trabajos futuros	89
Bibliografía	90

Capítulo 1

Introducción

En numerosos contextos donde se requiere tomar decisiones basadas en múltiples opiniones o fuentes, no es raro encontrarse con listas ordenadas que reflejan preferencias, relevancia o calidad. La agregación de rankings surge como una herramienta que permite combinar esas distintas clasificaciones individuales en una sola ordenación colectiva, que pretende capturar de la forma más justa posible una visión global o consensuada.

Este proceso resulta especialmente útil cuando las fuentes de información son diversas y pueden presentar ruido, sesgos o incluso contradicciones. Aunque en un primer momento pueda parecer una técnica muy ligada al ámbito académico o a la toma de decisiones formales, en realidad su aplicación se extiende a campos tan variados como la ingeniería, la medicina, el comercio electrónico o el análisis de opiniones en redes sociales.

Un ejemplo ilustrativo podría ser el caso de un grupo de cuatro amigos que quieren alquilar una casa para pasar el verano. Cada uno de ellos propone cinco alojamientos diferentes según sus propios criterios (ubicación, precio, piscina, etc.), generando así cuatro listas individuales (rankings) con ordenaciones distintas. A partir de estas listas, el grupo necesita elegir una única casa de forma consensuada. En este escenario, aplicar una técnica de agregación de rankings permite combinar de manera justa las preferencias individuales y tomar una decisión colectiva que refleje el consenso del grupo.

1.1. ¿Qué es un ranking?

En el contexto de la toma de decisiones basada en múltiples criterios u opiniones, el concepto de *ranking* es esencial. Un ranking representa una ordenación de alternativas en función de algún atributo o conjunto de atributos, ya sea relevancia, preferencia o calidad percibida. Esta ordenación puede provenir tanto de fuentes humanas (personas expertas, grupos de

usuarios) como de sistemas automáticos (motores de búsqueda, algoritmos de recomendación) [1].

Ranking individual y de consenso

Un **ranking individual** es una lista ordenada que refleja las preferencias de una única fuente de información. Esta fuente puede ser una persona, un sistema, un modelo estadístico o incluso un subsistema dentro de una arquitectura compleja. Cada ranking individual expresa una perspectiva subjetiva y parcial del problema.

Cuando se dispone de múltiples rankings individuales sobre un mismo conjunto de elementos, surge la necesidad de sintetizar esta información. El resultado de este proceso es un **ranking de consenso**, el cual busca representar una visión agregada que minimice el desacuerdo entre las distintas fuentes [2, 3]. La generación de rankings de consenso es fundamental tanto en contextos clásicos como la elección social, como en sistemas modernos como los motores de metabúsqueda o los sistemas de recomendación colaborativa [4].

Ejemplo: Supongamos que tres personas elaboran rankings individuales sobre tres películas:

- Persona A: 1. *Inception*, 2. *Dunkirk*, 3. *Interstellar*
- Persona B: 1. *Dunkirk*, 2. *Inception*, 3. *Interstellar*
- Persona C: 1. *Interstellar*, 2. *Inception*, 3. *Dunkirk*

Encontrar un ranking de consenso implica ordenar estas películas de forma que se represente de manera justa la opinión colectiva.

Rankings completos y parciales

Dependiendo del contexto, un ranking puede ser:

- **Completo**, si incluye y ordena todas las alternativas posibles.
- **Parcial**, si no todas las alternativas han sido evaluadas u ordenadas explícitamente.

Los rankings parciales son frecuentes en entornos con información incompleta o cuando ciertas fuentes no tienen conocimiento sobre todos los elementos. En estos casos, es común trabajar con técnicas que permitan reconstruir rankings incompletos para su posterior análisis y agregación [3].

1.2. Aplicaciones actuales de la agregación de rankings

A continuación, se exploran algunos de los escenarios donde los métodos de agregación de rankings han demostrado ser especialmente útiles, ya sea como herramienta de apoyo a la toma de decisiones o como componente clave en sistemas automatizados que requieren ordenar elementos según múltiples criterios u opiniones.

Metabúsqueda en la Web

Uno de los contextos más clásicos para aplicar métodos de agregación de rankings es la metabúsqueda. Dado que distintos motores de búsqueda emplean algoritmos diversos para ordenar los resultados, fusionar sus salidas mediante agregación permite mejorar la cobertura, reducir el sesgo individual de cada sistema y aumentar la robustez frente a técnicas de manipulación como el *spam*. Este enfoque ha sido ampliamente estudiado en el ámbito académico, destacando los trabajos de Dwork et al. [5], quienes proponen métodos para lograr un ranking de consenso más justo y eficaz frente a las limitaciones individuales de los motores de búsqueda.

Por ejemplo, si una persona busca información sobre “cómo preparar una entrevista técnica” y utiliza varios motores de búsqueda (como Google, Bing y Yahoo), cada uno ofrecerá una lista distinta de resultados. Aplicar técnicas de agregación permite combinar esas listas en un ranking único que integre lo mejor de cada fuente, mejorando así la cobertura y la calidad de la información presentada.

Sistemas de recomendación

En entornos donde se generan recomendaciones, como plataformas de contenido o comercio electrónico, se suelen combinar rankings generados a partir de diferentes modelos (colaborativos, basados en contenido, híbridos, etc.). La agregación permite equilibrar las fortalezas y debilidades de cada enfoque, logrando recomendaciones más precisas y personalizadas. Según se describe en [4], esta técnica contribuye a mejorar la calidad del sistema de recomendación al integrar distintas perspectivas sobre la relevancia de los ítems.

Un ejemplo de este tipo de aplicación podría darse al buscar “mejor portátil calidad-precio” en una tienda online, los resultados pueden provenir de distintos sistemas: uno que prioriza las valoraciones de usuarios, otro que analiza especificaciones técnicas, y un tercero que tiene en cuenta el historial de navegación del usuario. Mediante agregación de rankings, es

posible combinar estas diferentes fuentes para generar una recomendación única más ajustada a las preferencias del usuario.

Toma de decisiones multicriterio

En problemas donde se evalúan alternativas desde múltiples criterios o expertos, es común que se generen rankings parciales que necesitan ser consolidados. La agregación permite sintetizar estas valoraciones de manera justa y representativa. Además, en este contexto, resulta clave considerar la penalización por errores en las primeras posiciones del ranking.

Una situación real podría darse al seleccionar el mejor proveedor para una empresa, distintos departamentos (como logística, finanzas y calidad) pueden emitir sus propios rankings según criterios específicos. La agregación de estos rankings permite llegar a una decisión global que refleje equilibradamente las prioridades de todas las áreas implicadas.

Elección social y sistemas de votación

En el ámbito de la elección social, la agregación de rankings se utiliza para transformar las preferencias individuales de los votantes en un resultado colectivo que minimice el desacuerdo global [6, 1].

Por ejemplo, en una elección para decidir el representante de estudiantes en una facultad, cada votante puede ordenar a los candidatos según su preferencia. Mediante un método de agregación de rankings, se puede obtener un resultado final que refleje de forma justa el consenso del grupo, incluso si no hay una mayoría absoluta clara.

Aplicaciones en ingeniería y diseño

En procesos industriales y de ingeniería, distintos expertos suelen emitir juicios sobre propuestas técnicas o decisiones de diseño. La agregación permite integrar estas valoraciones en un ranking consensuado, facilitando la toma de decisiones en contextos multidisciplinares.

Para entender mejor este tipo de aplicaciones, podemos pensar en el diseño de un nuevo producto, como un dron, donde distintos equipos técnicos (aerodinámica, electrónica, materiales, etc.) pueden establecer sus propios rankings sobre diferentes prototipos en función de criterios específicos. La agregación de estos rankings permite obtener una clasificación final que refleje de forma equilibrada las prioridades técnicas de todos los departamentos implicados.

Estos casos ilustran cómo la agregación de rankings no solo tiene un

interés académico, sino también un impacto directo en aplicaciones reales. Desde motores de búsqueda hasta sistemas colaborativos y de toma de decisiones, su uso aporta valor al integrar información dispersa y, a menudo, contradictoria, en un resultado más robusto y representativo.

1.3. Objetivos del trabajo

Dada la importancia que tienen los rankings como herramienta para representar preferencias en contextos muy diversos, y considerando la variedad de fuentes de las que pueden proceder —humanas o sistemas automáticos—, el problema de su agregación adquiere un papel central en muchas aplicaciones prácticas. En este Trabajo de Fin de Grado se plantean, por tanto, los siguientes objetivos:

1. Estudiar las variantes del problema.
2. Analizar un conjunto de algoritmos de agregación de rankings.
3. Diseñar una herramienta para implementar los algoritmos con el objetivo de facilitar la resolución de los problemas de agregación de rankings.
4. Ejemplificar su uso en un caso de estudio.

1.4. Temporización del trabajo

Para alcanzar cada uno de los objetivos propuestos, se han definido una serie de tareas concretas que estructuran y guían el desarrollo del proyecto. A continuación, se presenta la temporización del mismo.

Objetivo 1: Estudiar las variantes del problema

1. Revisión bibliográfica sobre el problema y sus variantes.
2. Identificación de casos de uso relevantes.
3. Análisis de aplicaciones prácticas del problema.

Objetivo 2: Analizar un conjunto de algoritmos de agregación de rankings

1. Investigación sobre algoritmos de agregación de rankings existentes.
2. Análisis y comparación teórica de los algoritmos seleccionados.

Objetivo 3: Diseñar una herramienta para implementar los algoritmos con el objetivo de facilitar la resolución de los problemas de agregación de rankings

1. Diseño de la arquitectura de la herramienta.
2. Implementación de los algoritmos en la herramienta.
3. Evaluación de los algoritmos.
4. Comparación de los algoritmos.

Objetivo 4: Ejemplificar su uso en un caso de estudio

1. Selección del caso de estudio.
2. Aplicación de los algoritmos en el caso de estudio.
3. Análisis y evaluación de los resultados obtenidos.
4. Elaboración de conclusiones basadas en los experimentos realizados.

Además de las tareas asociadas a los objetivos establecidos, se debe tener en cuenta el tiempo destinado a la redacción de esta memoria y a las sesiones de tutoría con el director del proyecto. A continuación, en la Tabla 1.1, se presenta la planificación temporal del proyecto, donde cada número corresponde a una tarea específica. Por ejemplo, "1.1" se refiere a la primera tarea del Objetivo 1, y así sucesivamente.

Mes/Semana	Semana 1	Semana 2	Semana 3	Semana 4
Febrero	1.1, 1.2, Tutoría	1.3, 2.1	2.2	3.1
Marzo	3.1	3.2	3.2, Tutoría	3.3
Abril	3.3	3.4	3.4, 4.1, Tutoría	4.2
Mayo	4.2, Tutoría	4.3	4.3, Tutoría	4.4
Junio	Memoria, Tutoría	Memoria	-	-

Cuadro 1.1: Temporización del proyecto con tareas semanales

Capítulo 2

El problema de agregación de rankings

El problema de agregación de rankings surge en numerosos contextos en los que es necesario combinar múltiples ordenaciones parciales o totales de un conjunto de elementos en un solo ranking representativo o *ranking* de consenso. Este problema es fundamental en campos como la toma de decisiones multicriterio, la votación, la evaluación de alternativas en ingeniería y la recuperación de información [1, 3, 2].

En este capítulo se presentan las principales medidas empleadas para comparar rankings. Además, se analizan las distintas variantes del problema de agregación, considerando aspectos como el tipo de rankings de entrada (completos, incompletos, con empates), la jerarquía de los evaluadores y las propiedades deseables del ranking resultante.

2.1. Definición

Formalmente, sea $V = \{v_1, v_2, \dots, v_n\}$ un conjunto de n elementos a ser ordenados y sea $\mathcal{R} = \{R_1, R_2, \dots, R_m\}$ un conjunto de m rankings individuales proporcionados por distintos evaluadores o fuentes de información. Cada ranking R_i representa un orden total o parcial sobre los elementos de V . El objetivo es encontrar un ranking agregado R^* que sintetice de forma justa y representativa la información contenida en \mathcal{R} [1, 3].

Dependiendo del contexto, la agregación de rankings puede abordarse mediante distintos criterios de optimización. Entre los más comunes se encuentran minimizar la distancia entre R^* y los rankings individuales en \mathcal{R} —por ejemplo, utilizando métricas como la distancia de Kendall—, maximizar el consenso global entre las ordenaciones o garantizar ciertas propiedades

deseables como la imparcialidad, la neutralidad o la transitividad [5, 2].

Este problema es ampliamente estudiado en teoría de la decisión y en optimización combinatoria, dado que la búsqueda del ranking óptimo puede ser un problema NP-hard ¹ en muchos casos [3].

Una operación clave en el contexto de la agregación de rankings es la comparación entre ordenaciones, para lo cual se recurre al uso de medidas que cuantifican la distancia o similitud entre ellas. A continuación, se explican algunas de las más utilizadas.

2.2. Medidas para la comparación de rankings

Las medidas para la comparación de rankings son utilizadas para evaluar la similitud o discrepancia entre diferentes rankings. Estas medidas resultan esenciales en la comparación de rankings individuales con el ranking de consenso, permitiendo cuantificar la calidad del proceso de agregación y determinar hasta qué punto el ranking agregado representa fielmente las preferencias originales [2].

En la práctica, estas métricas permiten evaluar la coherencia del ranking final con respecto de los originales, lo que resulta fundamental en aplicaciones como la recuperación de información, los sistemas de recomendación y la toma de decisiones automatizada. Según [1], la selección de una medida adecuada depende del tipo de datos, la naturaleza de los rankings (totales o parciales) y los criterios de agregación utilizados.

Existen diversas medidas utilizadas en el análisis de rankings, entre las cuales destacan la distancia de Kendall, el coeficiente de correlación de Spearman (ρ) y el coeficiente WS propuesto por Sałabun y Urbaniak [7], que pondera las diferencias en las posiciones más altas de los rankings.

2.2.1. Distancia de Kendall

La distancia de Kendall es una métrica utilizada para cuantificar la discrepancia entre dos rankings, midiendo el número de pares de elementos cuyo orden relativo difiere entre ambas listas. Es ampliamente utilizada en contextos de agregación de rankings, sistemas de recomendación y búsqueda en la web [5].

¹Un problema es NP-hard si al menos es tan difícil como cualquier problema en la clase NP, lo que implica que no existe un algoritmo polinómico conocido que lo resuelva en el peor caso, a menos que $P = NP$.

Definición

Sean dos rankings τ_1 y τ_2 , la distancia de Kendall (K_d) se define como el número de pares (i, j) tales que el orden de i y j es diferente en τ_1 y τ_2 [8]:

$$K_d(\tau_1, \tau_2) = |\{(i, j) : i < j, [\tau_1(i) \prec \tau_1(j) \wedge \tau_2(i) \succ \tau_2(j)] \vee [\tau_1(i) \succ \tau_1(j) \wedge \tau_2(i) \prec \tau_2(j)]\}|$$

donde $\tau_1(i)$ y $\tau_2(i)$ representan las posiciones del elemento i en los rankings τ_1 y τ_2 , respectivamente.

Esta medida cumple con las propiedades fundamentales de una distancia: es no negativa, se anula si y solo si los rankings son idénticos, es simétrica y satisface la desigualdad triangular. Por ello, K_d puede considerarse formalmente una distancia definida sobre el conjunto de permutaciones [8].

Esta distancia varía entre 0, cuando los rankings son idénticos, y un máximo de $\frac{n(n-1)}{2}$ cuando un ranking es inverso del otro [5].

Distancia de Kendall Normalizada

Para facilitar la interpretación, se utiliza la distancia de Kendall normalizada (K_n), que ajusta el valor de K_d al rango $[0,1]$:

$$K_n = \frac{2K_d}{n(n-1)}$$

En este caso, $K_n = 0$ indica rankings idénticos y $K_n = 1$ significa que los rankings son completamente opuestos [8].

Coefficiente de Correlación de Kendall

El coeficiente de correlación de Kendall (τ) mide la similitud entre dos rankings en un rango de $[-1,1]$. Se define como:

$$\tau = 1 - 2K_n$$

o equivalentemente:

$$\tau = \frac{2(N_s - N_d)}{n(n-1)}$$

donde N_s es el número de pares concordantes y N_d es el número de pares discordantes entre los rankings [7].

Ejemplo

Consideremos dos rankings de tres elementos:

- Ranking 1 (τ_1): $A \succ B \succ C$
- Ranking 2 (τ_2): $C \succ A \succ B$

Evaluamos todos los pares posibles:

Par	Orden en τ_1	Orden en τ_2	Desacuerdo
(A, B)	$A \succ B$	$A \succ B$	No
(A, C)	$A \succ C$	$A \prec C$	Sí
(B, C)	$B \succ C$	$B \prec C$	Sí

Cuadro 2.1: Pares concordantes y discordantes en τ_1 y τ_2 .

Dado que hay dos pares en desacuerdo (A, C) y (B, C), la distancia de Kendall es $K_d = 2$.

La distancia normalizada es:

$$K_n = \frac{2K_d}{n(n-1)} = \frac{2 \times 2}{3 \times 2} = \frac{4}{6} = 0.667$$

Esto indica que aproximadamente el 66.7% de los pares tienen un orden diferente en los dos rankings.

Finalmente, el coeficiente de correlación de Kendall es:

$$\tau = 1 - 2K_n = 1 - 2 \times 0.667 = -0.334$$

lo que sugiere una correlación negativa débil entre los rankings, reflejando cierta discordancia [7].

2.2.2. Coeficiente de correlación de Spearman

El coeficiente de correlación de Spearman, denotado como ρ (rho), es una medida no paramétrica utilizada para evaluar la relación entre dos rankings. A diferencia de la correlación de Pearson², que mide relaciones lineales,

²La correlación de Pearson (r) mide la relación lineal entre dos variables continuas. Asume que ambas variables son normalmente distribuidas, tienen una relación lineal y presentan homoscedasticidad, es decir, una varianza constante en los datos.[9]

Spearman evalúa relaciones monótonas, es decir, si una variable tiende a aumentar o disminuir en función de la otra sin requerir una relación estrictamente lineal [1].

Definición

Dado un conjunto de n elementos con dos rankings τ_1 y τ_2 , el coeficiente de correlación de Spearman se calcula mediante la siguiente fórmula [7]:

$$\rho = 1 - \frac{6 \sum d_i^2}{n(n^2 - 1)}$$

donde:

- d_i es la diferencia entre las posiciones del elemento i en los dos rankings.
- n es el número total de elementos en el ranking.

El coeficiente de Spearman toma valores en el intervalo $[-1, 1]$, donde:

- $\rho = 1$ indica una correlación perfecta positiva.
- $\rho = -1$ indica una correlación perfectamente inversa.
- $\rho = 0$ sugiere ausencia de correlación entre los rankings [1].

Ejemplo

Consideremos los siguientes dos rankings:

- Ranking 1 (τ_1): $A \succ B \succ C$
- Ranking 2 (τ_2): $C \succ A \succ B$

Asignamos valores numéricos a los rankings:

Elemento	Rango en τ_1	Rango en τ_2	Diferencia (d_i)
A	1	2	-1
B	2	3	-1
C	3	1	2

Cuadro 2.2: Tabla con los rangos y diferencias de los elementos en los rankings τ_1 y τ_2 .

Calculamos la sumatoria de las diferencias al cuadrado:

$$\sum d_i^2 = (-1)^2 + (-1)^2 + 2^2 = 1 + 1 + 4 = 6$$

Sustituyendo en la fórmula de Spearman:

$$\rho = 1 - \frac{6 \times 6}{3(3^2 - 1)} = 1 - \frac{36}{3(9 - 1)} = 1 - \frac{36}{24} = 1 - 1.5 = -0.5$$

Este resultado indica una correlación negativa moderada entre los rankings, lo que refleja una discrepancia significativa en el ordenamiento [7].

2.2.3. Coeficiente WS

El coeficiente WS es una métrica diseñada para medir la similitud entre rankings, con una mayor sensibilidad a las diferencias en las primeras posiciones. A diferencia de coeficientes como Spearman o Kendall, que ponderan todas las posiciones por igual, WS prioriza los cambios en los elementos superiores del ranking, lo que lo hace ideal para problemas de toma de decisiones y evaluación de alternativas [7].

Definición

Dado un conjunto de n elementos y dos rankings τ_x y τ_y , el coeficiente WS se define como:

$$WS = 1 - \sum_{i=1}^n \left(2^{-R_{xi}} \cdot \frac{|R_{xi} - R_{yi}|}{\max(|1 - R_{xi}|, |n - R_{xi}|)} \right)$$

donde:

- R_{xi} y R_{yi} representan las posiciones del elemento i en los rankings τ_x y τ_y , respectivamente.
- n es el número total de elementos en el ranking.

Esta fórmula garantiza que WS siempre tome valores en el intervalo $[0, 1]$. Esto se debe a la estructura del término $2^{-R_{xi}}$, que decrece exponencialmente conforme aumenta la posición en el ranking, asegurando que los cambios en los primeros elementos afecten más el resultado final [7].

Ejemplos

A continuación, se presentan ejemplos inspirados en la Tabla 1 de [7], que ilustran cómo responde WS a cambios en los rankings.

Ejemplo 1: Pequeños cambios en posiciones bajas

- Ranking de referencia (τ_x): $A \succ B \succ C \succ D \succ E$
- Ranking comparado (τ_y): $A \succ B \succ C \succ E \succ D$

Aquí, el cambio ocurre en las últimas posiciones (D y E intercambian lugares), lo que tendrá un impacto reducido en WS.

Elemento	Pos. en τ_x	Pos. en τ_y	Término WS
A	1	1	0
B	2	2	0
C	3	3	0
D	4	5	$2^{-4} \cdot \frac{1}{\max(3,1)} = \frac{1}{48}$
E	5	4	$2^{-5} \cdot \frac{1}{\max(4,0)} = \frac{1}{128}$

Cuadro 2.3: Cálculo del coeficiente WS para cambios en posiciones bajas

$$WS = 1 - (0 + 0 + 0 + \frac{1}{48} + \frac{1}{128}) = 1 - 0.0287 = 0.9714$$

Ejemplo 2: Cambio en las primeras posiciones

- Ranking de referencia (τ_x): $A \succ B \succ C \succ D \succ E$
- Ranking comparado (τ_y): $B \succ A \succ C \succ D \succ E$

Aquí, el cambio ocurre en las primeras posiciones (A y B intercambian lugares), lo que tendrá un impacto más significativo en WS.

Elemento	Pos. en τ_x	Pos. en τ_y	Término WS
A	1	2	$2^{-1} \cdot \frac{1}{\max(0,4)} = \frac{1}{8}$
B	2	1	$2^{-2} \cdot \frac{1}{\max(1,3)} = \frac{1}{12}$
C	3	3	0
D	4	4	0
E	5	5	0

Cuadro 2.4: Cálculo del coeficiente WS para cambios en primeras posiciones

$$WS = 1 - (\frac{1}{8} + \frac{1}{12} + 0 + 0 + 0) = 1 - 0.2083 = 0.7917$$

Comparación de resultados:

- En el primer caso, donde los cambios ocurren en posiciones bajas, el WS es **0.9714**, indicando una similitud alta.
- En el segundo caso, donde los cambios ocurren en posiciones altas, el WS baja a **0.7917**, reflejando la importancia de las primeras posiciones.

Resumen

Cada coeficiente tiene un propósito específico:

- **Spearman** (ρ): Evalúa la similitud general del ranking sin priorizar ninguna posición.
- **Kendall** (τ): Considera el número de pares en desacuerdo sin diferenciar posiciones.
- **WS**: Prioriza los primeros elementos del ranking, lo que lo hace útil en problemas de selección y toma de decisiones [7].

Por tanto, podemos concluir que el coeficiente WS es ideal cuando se desea evaluar la similitud de rankings teniendo en cuenta la relevancia de las primeras posiciones. Su capacidad para ponderar cambios en la parte superior del ranking lo convierte en una herramienta valiosa en contextos como la búsqueda en Internet, la evaluación de candidatos y la toma de decisiones multicriterio [7].

2.3. Variantes del Problema de Agregación de Rankings

El problema de la agregación de rankings presenta múltiples variantes según las características de los datos y los contextos en los que se aplica. Estas variantes condicionan los algoritmos y métodos adecuados para obtener un ranking agregado que refleje las preferencias colectivas o la información combinada de distintas fuentes. En esta sección se exploran las variantes más relevantes, profundizando en sus implicancias teóricas y prácticas, así como se muestra un ejemplo de aplicación de las mismas.

Los algoritmos específicos capaces de tratar cada una de estas variantes serán descritos con mayor detalle en el Capítulo 3, donde se explican su funcionamiento, implementación y características técnicas.

2.3.1. Rankings Estrictamente Lineales

La forma más simple del problema de agregación se da cuando todos los rankings son completos y estrictamente lineales, es decir, cada ranking individual ordena todos los elementos sin repeticiones ni empates. Cada elemento ocupa una única posición, y no hay elementos con igualdad de preferencia ni valores ausentes (ejemplo en la Figura 2.1).

En este contexto, si un ranking τ asigna una posición más alta a un elemento e_i que a otro elemento e_j , se denota como $e_i \succ e_j$, lo cual indica que e_i aparece antes que e_j en la ordenación (es decir, tiene mayor preferencia o una posición más baja en términos numéricos).

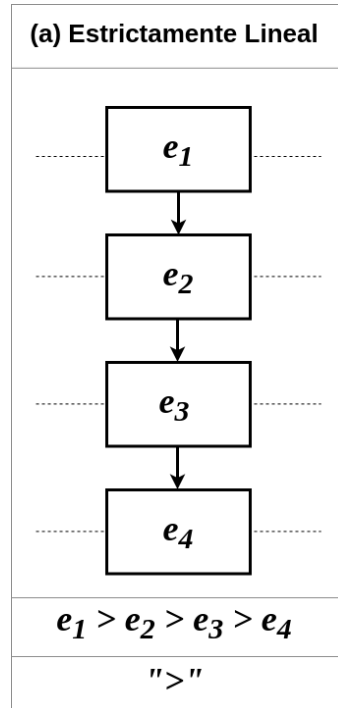


Figura 2.1: Ejemplo de ranking estrictamente lineal, donde cada elemento ocupa una posición única [1].

Aplicación práctica: Supongamos que tres personas participan como jurado en una cata de chocolates, y cada una debe ordenar cinco muestras distintas del 1 al 5 según su preferencia de sabor, sin empates. Por ejemplo, el jurado 1 podría establecer el ranking: $e_3 \succ e_1 \succ e_5 \succ e_2 \succ e_4$, donde e_i representa la i -ésima muestra de chocolate. Este tipo de ordenación estricta es un ejemplo claro de ranking lineal completo: todas las posiciones están asignadas, no hay empates y cada elemento ocupa un lugar único en la clasificación.

2.3.2. Rankings con Empate

En numerosos contextos, los rankings contienen empates, es decir, elementos que ocupan la misma posición debido a valoraciones idénticas o a la imposibilidad de establecer una diferencia clara entre ellos. Este tipo de ranking, denominado *non-strict ranking* o *ranking con indiferencias*, permite representar situaciones realistas en las que dos o más elementos son considerados equivalentes por los evaluadores.

Desde el punto de vista teórico, estos rankings pueden modelarse mediante relaciones de equivalencia (\sim), indicando que dos elementos no son

distinguibles entre sí en términos de preferencia [1]. Algunas técnicas como LCJ (Linear Conjoint of Judgements) o EYA (Extension of the Young Algorithm) están diseñadas para tratar explícitamente estas estructuras, preservando la consistencia del ranking agregado cuando existen empates [1] (ejemplo en la Figura 2.2).

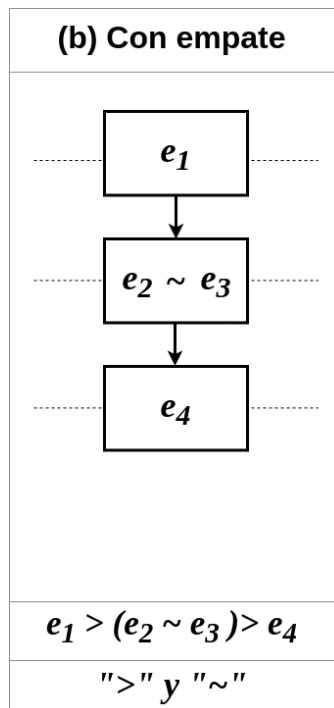


Figura 2.2: Ejemplo de ranking con empates, donde dos elementos comparten la misma posición [1].

Aplicación práctica: Retomando el ejemplo de la cata de chocolates, imaginemos que un jurado no logra distinguir entre dos muestras porque ambas le resultan igualmente agradables. En este caso, su ranking podría ser: $e_3 \prec (e_1 \sim e_5) \prec e_2 \prec e_4$. Esto indica que e_3 es la más preferida, e_1 y e_5 comparten el segundo lugar (empate), seguidos por e_2 y, finalmente, e_4 . Este tipo de ordenación refleja una situación común en evaluaciones sensoriales, donde la subjetividad o la cercanía en calidad de los elementos impide una discriminación clara entre ciertos pares.

2.3.3. Rankings con Ponderación

Cuando los rankings provienen de múltiples fuentes que no tienen la misma fiabilidad o importancia, es fundamental ponderar cada una de ellas para obtener un resultado más representativo. Este enfoque reconoce que

algunas fuentes deben tener más peso que otras, ya sea por su experiencia, conocimiento experto o calidad de información [2] (ejemplo en la Figura 2.3).

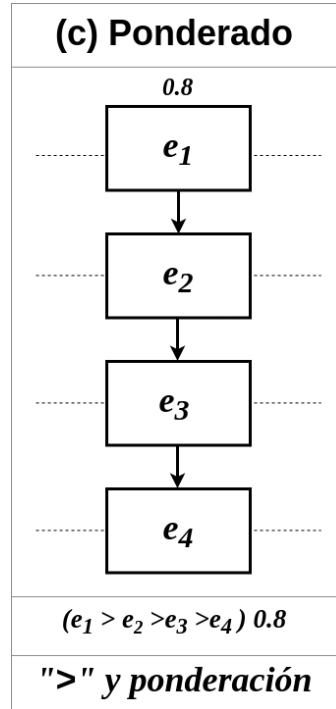


Figura 2.3: Ejemplo de ranking con ponderación.

Aplicación práctica: Plataformas como Netflix, donde las opiniones de usuarios expertos pueden tener más peso que las de usuarios ocasionales. En este caso, los rankings emitidos por usuarios expertos tendrían una ponderación mayor que los emitidos por usuarios ocasionales.

A la luz del análisis anterior, en este trabajo nos centraremos en estas tres variantes específicas del problema de agregación de rankings: *Rankings estrictamente lineales*, *Rankings con empates* y *Agregación con ponderación*, las cuales representan casos fundamentales y recurrentes en aplicaciones reales, y permiten una implementación directa o adaptada de los algoritmos que se han seleccionado para este estudio.

Capítulo 3

Algoritmos de Agregación de Rankings

El problema de agregación de rankings, como se ha visto en los capítulos anteriores, aparece en numerosos contextos donde es necesario consolidar múltiples juicios individuales en una única decisión colectiva. Para ello, se han desarrollado a lo largo del tiempo una gran variedad de algoritmos y enfoques, cada uno con sus propias características, ventajas y limitaciones.

Dada la enorme diversidad de métodos existentes en el ámbito de la agregación de rankings, en este trabajo se ha realizado una selección concreta de algoritmos representativos. La elección no ha sido aleatoria: los algoritmos escogidos —**Borda**, **Copeland**, **Kemeny-Young**, **Schulze**, **Footrule** y los métodos basados en **cadenas de Markov**— han sido seleccionados por su capacidad para abordar las distintas variantes del problema analizadas en el **Capítulo 2**. En la Tabla 3.1, vemos como quedan relacionadas dichas variantes con los algoritmos elegidos. Además, cada uno de ellos pertenece a una familia diferente de enfoques (basados en puntuación, comparaciones por pares, distancias, teoría de grafos o modelos probabilísticos), lo que permite ofrecer una visión amplia y comparativa de las posibles soluciones al problema de agregación.

Algoritmo	Rankings estrictamente lineales	Rankings con empates	Agregación con ponderación
Borda	X	X	X
Copeland	X		X
Kemeny-Young	X		
Schulze	X		
Footrule	X		
Cadenas de Markov	X		

Cuadro 3.1: Algoritmos tratados y su aplicabilidad según la variante de agregación de rankings.

Asimismo, cabe destacar que los algoritmos seleccionados son especial-

mente relevantes en el contexto de rankings estrictamente lineales (es decir, sin empates ni elementos incomparables), una variante que, como se demuestra en la Tabla 3.1, es común a todos los métodos analizados. Por ello, todos los ejemplos de aplicación práctica en este capítulo estarán desarrollados bajo esa variante básica de rankings lineales, lo cual facilita tanto la comparación como la implementación de los algoritmos.

Para ilustrar el funcionamiento de cada uno de los algoritmos, se utilizará el mismo conjunto de rankings individuales a lo largo de todo el capítulo a modo de **ejemplo base**. Este conjunto está formado por tres rankings sobre seis elementos (denotados como A , B , C , D , E y F):

- $\sigma_1: A \succ B \succ C \succ D \succ E \succ F$
- $\sigma_2: B \succ C \succ D \succ A \succ F \succ E$
- $\sigma_3: C \succ D \succ A \succ B \succ E \succ F$

Este conjunto de rankings, completo y estrictamente lineal, será el caso de estudio sobre el que se aplicarán todos los algoritmos. Esto permitirá comparar directamente los resultados obtenidos por cada uno de ellos y observar las diferencias entre los métodos.

En cada sección dedicada a un algoritmo se explicará su funcionamiento, se presentará su formulación (en forma de pseudocódigo), se analizará su complejidad y se discutirán sus ventajas y limitaciones. Además de las particularidades propias de cada método, en todos los casos se valorarán cuatro aspectos comunes: su complejidad computacional, la sencillez de su implementación, la facilidad de interpretación de sus resultados y el cumplimiento (o no) del *criterio de Condorcet*. Este último establece que, si existe un elemento que es preferido frente a cualquier otro en comparaciones pareadas, dicho elemento debería ocupar la primera posición del ranking agregado. Se trata de una propiedad clásica en la teoría de votación y toma de decisiones colectivas, que suele utilizarse como indicador de consistencia mayoritaria [1].

3.1. Método de Borda

Definición y fundamento

El *método de Borda* es uno de los enfoques clásicos para la agregación de rankings, propuesto originalmente por Jean-Charles de Borda en el siglo XVIII. Su funcionamiento se basa en asignar puntuaciones a cada posición dentro de los rankings individuales: cuanto más alta sea la posición de un

elemento, mayor será la puntuación que reciba. Posteriormente, se suman los puntos obtenidos por cada elemento en todos los rankings y se genera el ranking agregado ordenando los elementos por su puntuación total, de mayor a menor.

La descripción de este método y sus propiedades se basa en la presentación realizada en [1]. Formalmente, si hay n elementos y un elemento ocupa la posición i en un ranking individual, se le asigna una puntuación de $n-i$. La motivación es que un elemento que aparece frecuentemente en las primeras posiciones debería considerarse más preferido en el ranking colectivo.

Pseudocódigo y explicación

A continuación, se presenta una versión en pseudocódigo del algoritmo. En este contexto:

- $\mathcal{R} = \{\sigma_1, \sigma_2, \dots, \sigma_k\}$ representa el conjunto de rankings individuales.
- $U = \{u_1, u_2, \dots, u_n\}$ es el conjunto de elementos que aparecen en todos los rankings.
- $S[u_i]$ es un array o diccionario que acumula la puntuación total de cada elemento u_i a lo largo de los rankings.
- $\sigma_j(u_i)$ denota la posición de u_i en el ranking σ_j (siendo 1 la mejor posición).

Algoritmo 1: Método de Borda

Input: $\mathcal{R} = \{\sigma_1, \sigma_2, \dots, \sigma_k\}$ rankings sobre elementos
 $U = \{u_1, \dots, u_n\}$

Output: Ranking agregado σ^*

Inicializar $S[u_i] \leftarrow 0$ para cada $u_i \in U$;

foreach ranking $\sigma_j \in \mathcal{R}$ **do**
 foreach elemento $u_i \in U$ **do**
 $pos \leftarrow \sigma_j(u_i)$
 $S[u_i] \leftarrow S[u_i] + (n - pos)$

Ordenar los elementos de U por $S[u_i]$ en orden decreciente;

return ranking σ^* según el orden obtenido

Este algoritmo se basa en recorrer todos los rankings y acumular las puntuaciones correspondientes para cada elemento según la posición que ocupa en cada ranking. Una vez sumadas todas las puntuaciones, se ordenan los elementos de mayor a menor para obtener el ranking final.

Análisis de complejidad

La eficiencia computacional del método de Borda es una de sus grandes ventajas. El algoritmo requiere recorrer k rankings, cada uno con n elementos, asignando y sumando puntuaciones en cada paso. Esta parte tiene una complejidad de $O(nk)$, ya que se procesan n elementos por cada uno de los k rankings.

Una vez acumuladas las puntuaciones, es necesario ordenar los n elementos según su puntaje total para construir el ranking agregado. Esta operación se realiza utilizando algoritmos de ordenación por comparación como *MergeSort* o *QuickSort*, cuya complejidad en el caso general es $O(n \log n)$.

Por tanto, la complejidad total del algoritmo de Borda es de $O(nk + n \log n)$. Esta eficiencia lo hace especialmente útil en contextos donde se requiere una ejecución rápida y no se dispone de grandes recursos computacionales.

Propiedades, ventajas y desventajas

La descripción de las propiedades, así como las ventajas y limitaciones del método de Borda, se basa principalmente en el análisis de [1] y en revisiones complementarias como [10].

- ✓ **Implementación sencilla:** La lógica del algoritmo es directa y no requiere estructuras complejas, lo que facilita tanto su desarrollo como su interpretación [1].
- ✓ **Refleja una preferencia promedio:** El método premia a los elementos que se mantienen en posiciones relativamente altas a lo largo de los rankings, aunque no sean necesariamente los primeros. Esto proporciona una medida del consenso general [10].
- ✗ **No cumple el criterio de Condorcet:** Puede ocurrir que un elemento gane todas las comparaciones por pares (Condorcet winner) y, aun así, no ocupe el primer lugar en el ranking de Borda. Esta discrepancia ha sido ampliamente documentada en literatura sobre votación social [1].
- ✗ **Vulnerabilidad a manipulación:** Si los rankings individuales pueden ser influenciados estratégicamente (por ejemplo, en votaciones), el método puede verse afectado, ya que pequeños cambios pueden alterar significativamente el resultado [1].
- ✗ **Linealidad en la asignación de puntos:** La puntuación asignada depende únicamente de la posición, sin tener en cuenta la intensidad de

la preferencia o las diferencias relativas entre elementos consecutivos [10].

- ✓ **Complejidad computacional baja:** El cálculo del ranking agregado puede realizarse en tiempo lineal respecto al número de rankings y elementos, lo que lo hace adecuado incluso para conjuntos de datos grandes [1].
- ✓ **Fácil de interpretar:** La mecánica del método es fácilmente entendible, tanto por perfiles técnicos como no técnicos. Se basa en una lógica acumulativa muy intuitiva.

Ejemplo ilustrativo

Consideremos un conjunto de elementos $U = \{A, B, C, D, E, F\}$ y tres rankings individuales estrictamente lineales, es decir, sin empates:

- σ_1 : $A \succ B \succ C \succ D \succ E \succ F$
- σ_2 : $B \succ C \succ D \succ A \succ F \succ E$
- σ_3 : $C \succ D \succ A \succ B \succ E \succ F$

Esto define un conjunto de rankings \mathcal{R} con $k = 3$ rankings y $n = 6$ elementos. Aplicamos el algoritmo de Borda usando un array $S[u_i]$ para acumular los puntos de cada elemento.

A continuación, se muestra la tabla con la posición y la puntuación asignada a cada elemento en cada ranking:

Elemento	σ_1		σ_2		σ_3		Total Pts.
U	Pos.	Pts.	Pos.	Pts.	Pos.	Pts.	$S[u_i]$
A	1	5	4	2	3	3	10
B	2	4	1	5	4	2	11
C	3	3	2	4	1	5	12
D	4	2	3	3	2	4	9
E	5	1	6	0	5	1	2
F	6	0	5	1	6	0	1

Finalmente, se ordenan los elementos según la puntuación total, dando lugar al ranking agregado:

$$C \succ B \succ A \succ D \succ E \succ F$$

Este resultado muestra cómo el método de Borda recompensa a los elementos que, sin necesidad de ocupar siempre la primera posición, mantienen un buen rendimiento global a lo largo de los distintos rankings.

3.2. Método de Copeland

Definición y fundamento

El *método de Copeland* es una técnica de agregación de rankings basada en comparaciones por pares. A diferencia de otros métodos que suman puntuaciones según la posición en los rankings, Copeland se centra en analizar, para cada par de elementos, cuántas veces uno es preferido frente al otro en los rankings individuales. Si un elemento es preferido sobre otro en la mayoría de los rankings, se le considera ganador en esa comparación.

La puntuación final de cada elemento es el número de pares en los que resulta ganador. El ranking agregado se obtiene ordenando los elementos según sus puntuaciones, de mayor a menor. Este enfoque refleja el principio de Condorcet: si existe un elemento que gana todas sus comparaciones directas, será el primero en el ranking resultante.

La descripción del método que se presenta en esta sección está basada principalmente en [1].

Pseudocódigo y explicación

A continuación, se presenta el pseudocódigo para el método de Copeland, junto con la definición de los elementos involucrados:

- $\mathcal{R} = \{\sigma_1, \sigma_2, \dots, \sigma_k\}$ es el conjunto de rankings individuales.
- $U = \{u_1, u_2, \dots, u_n\}$ es el conjunto de elementos.
- $C[u_i]$ es el número de comparaciones por pares que gana el elemento u_i .
- $\sigma_j(u_i)$ representa la posición de u_i en el ranking σ_j (posición 1 es la mejor).

```

Input:  $\mathcal{R} = \{\sigma_1, \dots, \sigma_k\}$  rankings sobre  $U = \{u_1, \dots, u_n\}$ 
Output: Ranking agregado  $\sigma^*$ 
Inicializar  $C[u_i] \leftarrow 0$  para cada  $u_i \in U$ ;
foreach par  $(u_i, u_j)$  tal que  $i \neq j$  do
     $wins \leftarrow 0$ ;
    foreach ranking  $\sigma_r \in \mathcal{R}$  do
        if  $\sigma_r(u_i) < \sigma_r(u_j)$  then
             $wins \leftarrow wins + 1$ ;
    if  $wins > \frac{k}{2}$  then
         $C[u_i] \leftarrow C[u_i] + 1$ ; //  $u_i$  gana a  $u_j$ 

Ordenar los elementos de  $U$  por  $C[u_i]$  en orden decreciente;
return ranking  $\sigma^*$  según el orden obtenido

```

Análisis de complejidad

$$O(n^2k + n \log n) \approx O(n^2k)$$

Propiedades, ventajas y desventajas

✓ **Cumple el criterio de Condorcet:** Si existe un elemento que gana todas las comparaciones por pares, el método de Copeland garantiza que dicho elemento se ubique en la primera posición del ranking agregado [1].

- ✓ **Transparente y neutral:** Todas las comparaciones entre pares se tratan de forma simétrica e independiente del orden en que aparecen en los rankings, lo cual garantiza imparcialidad en la agregación [1, 3].
- ✓ **Interpretación intuitiva:** El funcionamiento del algoritmo es comprensible: gana quien gana más comparaciones uno a uno. Esto facilita su explicación a usuarios no técnicos [1].
- ✗ **Implementación menos directa:** Aunque conceptualmente sencillo, requiere comparar todos los pares de elementos y contabilizar preferencias, lo que implica estructuras adicionales y una programación algo más elaborada que Borda [3].
- ✗ **Complejidad computacional moderadamente alta:** Requiere evaluar todas las comparaciones por pares entre elementos, resultando en una complejidad cuadrática respecto al número de elementos y lineal respecto al número de rankings, es decir, $O(n^2k)$ [3].
- ✗ **Ignora la intensidad de preferencia:** Solo se cuenta si un elemento gana o pierde en una comparación, sin tener en cuenta la magnitud de la diferencia en los rankings individuales [10, 2].
- ✗ **Puede generar empates:** Es habitual que varios elementos obtengan el mismo número de victorias, especialmente cuando el número de rankings es bajo o las preferencias están muy repartidas. Esto exige la aplicación de criterios de desempate adicionales [10, 2].

Ejemplo ilustrativo

Utilizando los mismos rankings lineales y el conjunto de elementos $U = \{A, B, C, D, E, F\}$, se analizan las comparaciones por pares en los tres rankings siguientes:

- $\sigma_1: A \succ B \succ C \succ D \succ E \succ F$
- $\sigma_2: B \succ C \succ D \succ A \succ F \succ E$
- $\sigma_3: C \succ D \succ A \succ B \succ E \succ F$

El siguiente cuadro indica el resultado de cada comparación por pares. Una marca ✓ indica que el elemento de la fila gana frente al de la columna (es decir, es preferido en al menos 2 de los 3 rankings), mientras que ✗ indica lo contrario:

Elemento	A	B	C	D	E	F
A	–	✓	✗	✗	✓	✓
B	✗	–	✓	✓	✓	✓
C	✓	✗	–	✓	✓	✓
D	✓	✗	✗	–	✓	✓
E	✗	✗	✗	✗	–	✓
F	✗	✗	✗	✗	✗	–

El número total de victorias por elemento es:

Elemento	Puntaje Copeland $C[u_i]$
B	4
C	4
A	3
D	3
E	1
F	0

Como resultado, el ranking agregado según Copeland es:

$$B = C \succ A = D \succ E \succ F$$

Este ejemplo muestra que el método de Copeland puede ser útil para capturar de forma sencilla **cuántas veces un elemento es preferido sobre otros en comparaciones directas**. Es decir, se basa en contar cuántos "duelos" gana cada elemento frente al resto, sin importar por cuánto gana ni en qué posición general aparece. En ese sentido, refleja **la fuerza de un elemento en términos de cuántas comparaciones uno a uno gana**, lo cual puede ser útil cuando se quiere priorizar esa información.

Sin embargo, también revela limitaciones importantes. Por ejemplo, genera **empates frecuentes**, como se ve entre los pares $B = C$ y $A = D$, lo cual puede dificultar la construcción de un ranking lineal único. Además, al considerar solo el número de victorias directas, **ignora relaciones indirectas**, como que A gane a B, B a C, y por tanto A debería tener ventaja sobre C, aunque no se hayan comparado directamente. Esto limita su capacidad para capturar patrones más complejos o "transitivos.^{en} los datos. Por tanto, aunque Copeland es un método simple y transparente, puede no ser la mejor opción si se necesita un ranking totalmente ordenado o más matizado.

3.3. Método de Kemeny-Young

Definición y fundamento

El *método de Kemeny-Young* consiste en encontrar un ranking que minimice la suma total de las discrepancias con respecto a los rankings individuales. Para cuantificar dicha discrepancia se utiliza comúnmente la **distancia de Kendall-tau** (2.2.1), que mide el número de pares de elementos que están ordenados de forma diferente en dos rankings.

Formalmente, se busca el ranking σ^* que minimiza la suma de las distancias Kendall-tau con todos los rankings individuales de entrada. Este enfoque se considera óptimo en términos de representación del consenso global, y está respaldado por sólidos fundamentos teóricos en teoría de decisiones y votación social [3, 5].

Pseudocódigo y explicación

Dado un conjunto de rankings $\mathcal{R} = \{\sigma_1, \dots, \sigma_k\}$ sobre un conjunto de elementos $U = \{u_1, \dots, u_n\}$, el objetivo es encontrar el ranking σ que minimice la siguiente función de costo:

$$C(\sigma) = \sum_{j=1}^k d_K(\sigma, \sigma_j)$$

donde d_K representa la distancia de Kendall-tau entre dos rankings.

Algoritmo 3: Método de Kemeny-Young (búsqueda exhaustiva)

Input: $\mathcal{R} = \{\sigma_1, \dots, \sigma_k\}$ rankings sobre $U = \{u_1, \dots, u_n\}$
Output: Ranking agregado σ^*
 Inicializar $C_{\min} \leftarrow \infty$, $\sigma^* \leftarrow \emptyset$;
foreach *perm.* σ *de* U **do**
 $C \leftarrow 0$;
 foreach *ranking* $\sigma_j \in \mathcal{R}$ **do**
 $C \leftarrow C + d_K(\sigma, \sigma_j)$; // suma distancias Kendall-tau
 if $C < C_{\min}$ **then**
 $C_{\min} \leftarrow C$;
 $\sigma^* \leftarrow \sigma$;
return σ^*

La búsqueda se realiza sobre todas las permutaciones posibles del conjunto U , por lo que este procedimiento tiene un coste factorial. Debido a esto, en problemas reales se suelen aplicar algoritmos heurísticos o aproxi-

maciones.

Análisis de complejidad

La principal desventaja del método de Kemeny-Young es su coste computacional. El número de rankings posibles para n elementos es $n!$, y para cada uno se evalúan k distancias Kendall-tau, con un coste de $O(n \log n)$ por comparación. Esto da lugar a una complejidad total de:

$$O(n! \cdot k \cdot n \log n)$$

Este problema ha sido demostrado como NP-hard [5], lo que implica que no se conoce un algoritmo eficiente para resolverlo exactamente en todos los casos. Sin embargo, su precisión a la hora de representar el consenso hace que se utilice en escenarios donde la calidad del resultado es prioritaria frente al tiempo de ejecución.

Propiedades, ventajas y desventajas

El método de Kemeny-Young tiene múltiples propiedades que lo hacen destacar, especialmente desde un punto de vista teórico. A continuación se resumen sus principales características, basadas en [3], [5] y [2].

- ✓ **Alta representatividad del consenso:** El ranking resultante minimiza el número total de desacuerdos con los rankings originales, proporcionando una solución óptima en términos de similitud global [3].
- ✓ **Equidad entre rankings:** Todos los rankings individuales tienen el mismo peso en el resultado final, lo que asegura una agregación justa sin sesgos ni ponderaciones artificiales [2].
- ✓ **Cumple el criterio de Condorcet:** Si existe un ganador de Condorcet, el método de Kemeny-Young lo posicionará como el primero en el ranking agregado [5].
- ✗ **Implementación compleja:** Evaluar todas las permutaciones posibles para encontrar la solución óptima requiere una programación sofisticada, especialmente si se intenta aplicar sin heurísticas [3].
- ✗ **Complejidad computacional muy elevada:** El problema es NP-hard. La necesidad de evaluar factorialmente todas las posibles permutaciones hace que el algoritmo no sea aplicable directamente en conjuntos medianos o grandes sin simplificaciones [5].

- ✗ **Necesidad de heurísticas o algoritmos aproximados:** En la práctica, se suele recurrir a métodos aproximados, lo cual puede alejar la solución obtenida del óptimo teórico [3].
- ✗ **Dificultad de interpretación para usuarios no técnicos:** A pesar de sus fundamentos lógicos sólidos, su formulación matemática y el uso de distancias como Kendall-tau pueden dificultar su comprensión fuera del ámbito especializado [2].
- ✗ **Dependencia de la métrica elegida:** Aunque lo habitual es emplear la distancia de Kendall-tau, el método permite utilizar otras métricas, lo que puede alterar el ranking final y generar inconsistencias si no se elige adecuadamente [2].

Ejemplo ilustrativo

Utilizamos el mismo conjunto de elementos $U = \{A, B, C, D, E, F\}$ y rankings lineales definidos anteriormente:

- $\sigma_1: A \succ B \succ C \succ D \succ E \succ F$
- $\sigma_2: B \succ C \succ D \succ A \succ F \succ E$
- $\sigma_3: C \succ D \succ A \succ B \succ E \succ F$

Sea $\mathcal{R} = \{\sigma_1, \sigma_2, \sigma_3\}$. Para calcular la distancia de Kendall-tau entre rankings, se cuentan los pares de elementos cuyo orden relativo difiere. Por ejemplo:

$$d_K(\sigma_1, \sigma_2) = 4$$

Evalutando todas las permutaciones posibles y sumando sus distancias con los rankings dados, se obtiene el ranking con menor costo total. Para este ejemplo, el ranking que minimiza la suma de distancias Kendall-tau es:

$$\sigma^* = B \succ C \succ D \succ A \succ E \succ F$$

Este ranking es el que tiene menor número de desacuerdos con los rankings originales, lo cual lo convierte en el consenso óptimo según el criterio de Kemeny.

3.4. Método de Schulze

Definición y fundamento

El *método de Schulze* es un algoritmo de agregación de rankings basado en comparaciones por pares y rutas indirectas de preferencia. El enfoque consiste en construir un grafo dirigido donde cada nodo representa un elemento y las aristas están ponderadas según el número de rankings en los que un elemento es preferido a otro. A partir de este grafo, se calcula la *fuerza de la ruta más fuerte* entre cada par de elementos y se establece un orden relativo entre ellos.

El método fue propuesto por Markus Schulze y es ampliamente utilizado en sistemas electorales por su capacidad para resolver de forma robusta ciclos de Condorcet. La relación de preferencia entre dos elementos no se basa solo en comparaciones directas, sino también en rutas alternativas más sólidas. Esta metodología está fundamentada en propiedades axiomáticas que lo hacen especialmente interesante en contextos de votación colectiva [1, 2, 6].

Pseudocódigo y explicación

Dado un conjunto de rankings $\mathcal{R} = \{\sigma_1, \dots, \sigma_k\}$ sobre elementos $U = \{u_1, \dots, u_n\}$:

- $D[u_i, u_j]$ representa el número de rankings en los que u_i es preferido a u_j (matriz de comparaciones directas).
- $P[u_i, u_j]$ representa la fuerza de la ruta más fuerte desde u_i hasta u_j .
- Se establece que $u_i \succ u_j$ si $P[u_i, u_j] > P[u_j, u_i]$.

Algoritmo 4: Método de Schulze**Input:** $\mathcal{R} = \{\sigma_1, \dots, \sigma_k\}$ rankings sobre $U = \{u_1, \dots, u_n\}$ **Output:** Ranking agregado σ^* Inicializar $D[u_i, u_j] \leftarrow 0$ para todo par $i \neq j$;**foreach** ranking $\sigma_r \in \mathcal{R}$ **do** **foreach** $u_i \succ u_j$ en σ_r **do** $D[u_i, u_j] \leftarrow D[u_i, u_j] + 1$;**foreach** $i \neq j$ **do** **if** $D[u_i, u_j] > D[u_j, u_i]$ **then** $P[u_i, u_j] \leftarrow D[u_i, u_j]$; **else** $P[u_i, u_j] \leftarrow 0$ **foreach** $i \neq j \neq k$ **do** **if** $P[u_j, u_k] > 0$ **then** $P[u_i, u_k] \leftarrow \max(P[u_i, u_k], \min(P[u_i, u_j], P[u_j, u_k]))$;Ordenar U según $P[u_i, u_j] > P[u_j, u_i]$;**return** σ^*

El algoritmo se basa en una variante del algoritmo de Floyd-Warshall para encontrar rutas más fuertes entre pares. Esta comparación de rutas hace que el método sea especialmente resistente a manipulaciones y ciclos.

Análisis de complejidad

El cálculo de la matriz de comparaciones directas requiere $O(n^2k)$, ya que se recorren todos los pares en cada ranking. La parte más costosa es la propagación de las rutas más fuertes, que requiere $O(n^3)$ usando el algoritmo de Floyd-Warshall adaptado. Por tanto, la complejidad total es:

$$O(n^2k + n^3)$$

Este coste es asumible en la mayoría de contextos de votación o rankings medianos, pero puede volverse costoso si el número de elementos crece mucho.

Propiedades, ventajas y desventajas

Las propiedades del método Schulze lo hacen especialmente adecuado en escenarios donde se desea consistencia mayoritaria y resistencia a manipulaciones. A continuación se resumen sus características principales, basadas en [1, 2, 6, 3].

- ✓ **Cumple el criterio de Condorcet:** El método de Schulze garantiza que, si existe un candidato de Condorcet, este será el primero en el ranking final [1].
- ✓ **Robustez frente a inconsistencias locales:** La lógica de rutas indirectas permite que, incluso si un ranking individual introduce contradicciones o ciclos, el método identifique la estructura global más coherente [2].
- ✓ **Resistencia a manipulaciones estratégicas:** Estudios recientes posicionan al método de Schulze como uno de los menos vulnerables frente a intentos de manipulación por parte de votantes estratégicos, especialmente frente a métodos basados en puntos o conteos simples [2, 6].
- ✗ **Implementación más compleja:** La necesidad de construir un grafo de comparaciones y calcular rutas fuertes mediante algoritmos tipo Floyd-Warshall implica una programación menos directa que la de métodos como Borda o Copeland [3].
- ✗ **Complejidad computacional superior:** Aunque manejable, el coste computacional del método es considerable, especialmente en conjuntos con muchos elementos, debido al cálculo de rutas para todos los pares posibles [3].
- ✗ **Dificultad de interpretación:** La noción de “rutas más fuertes” no es intuitiva para todos los usuarios. A diferencia de métodos que se basan en conteo de votos o posiciones, la lógica de Schulze requiere un mayor esfuerzo conceptual para ser comprendida plenamente por personas ajenas al ámbito técnico [1].

Ejemplo ilustrativo

Tomamos el conjunto de elementos $U = \{A, B, C, D, E, F\}$ y los siguientes rankings individuales:

- $\sigma_1: A \succ B \succ C \succ D \succ E \succ F$
- $\sigma_2: B \succ C \succ D \succ A \succ F \succ E$
- $\sigma_3: C \succ D \succ A \succ B \succ E \succ F$

Paso 1: Matriz de comparaciones directas $D[u_i, u_j]$ Para cada par de elementos (u_i, u_j) , se cuenta cuántas veces u_i aparece antes que u_j en los rankings. El resultado se muestra en la siguiente matriz:

$D[u_i, u_j]$	A	B	C	D	E	F
A	—	2	1	1	3	3
B	1	—	2	2	3	3
C	2	1	—	3	3	3
D	2	1	0	—	3	3
E	0	0	0	0	—	2
F	0	0	0	0	1	—

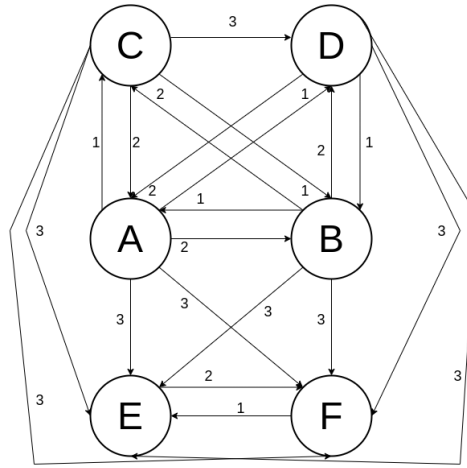


Figura 3.1: Representación completa del grafo dirigido construido a partir de la matriz de comparaciones directas $D[u_i, u_j]$. Cada flecha indica cuántas veces un elemento ha sido preferido sobre otro en los rankings individuales. Este grafo es la base para el cálculo de las rutas más fuertes del método de Schulze.

Paso 2: Cálculo de rutas más fuertes $P[u_i, u_j]$ A partir de la matriz D , se calcula la fuerza de la ruta más fuerte entre cada par (u_i, u_j) . Esta fuerza se define como el valor máximo del mínimo de los pesos a lo largo de todas las rutas posibles entre los nodos¹.

Ejemplo de cálculo de una ruta más fuerte: $P[A, B]$ Para ilustrar cómo se calcula el valor de la matriz $P[u_i, u_j]$, tomamos como ejemplo el caso $P[A, B]$. Según la definición del método de Schulze, debemos encontrar todas las rutas dirigidas posibles que conecten el nodo A con el nodo B y calcular, para cada una, la fuerza de la ruta (mínimo de los pesos del

¹Este tipo de cálculo se basa en una adaptación del algoritmo de Floyd-Warshall, utilizado clásicamente para encontrar caminos más cortos en grafos dirigidos ponderados. En el caso de Schulze, en lugar de sumar pesos, se toma el mínimo de la ruta y luego el máximo entre rutas. Véase [5] para más detalles.

camino). Finalmente, nos quedamos con la ruta cuya fuerza sea máxima entre todas las posibles.

A continuación se muestran todas las rutas válidas desde A hasta B según la matriz D (ver Figura 3.1):

- **Ruta directa:** $A \rightarrow B$
Peso: $D[A, B] = 2$ Fuerza: $\min(2) = \mathbf{2}$
- **Ruta indirecta:** $A \rightarrow C \rightarrow B$ Pesos: $D[A, C] = 1, D[C, B] = 1$ Fuerza: $\min(1, 1) = \mathbf{1}$
- **Ruta indirecta:** $A \rightarrow D \rightarrow B$ Pesos: $D[A, D] = 1, D[D, B] = 1$ Fuerza: $\min(1, 1) = \mathbf{1}$
- **Ruta indirecta:** $A \rightarrow C \rightarrow D \rightarrow B$ Pesos: $D[A, C] = 1, D[C, D] = 3, D[D, B] = 1$ Fuerza: $\min(1, 3, 1) = \mathbf{1}$

De todas estas rutas, la que tiene mayor fuerza es:

Ruta $A \rightarrow B$, con fuerza 2

Por tanto, en la matriz de rutas más fuertes se establece:

$$P[A, B] = 2$$

Este procedimiento es el que debe repetirse para cada par de elementos (u_i, u_j) , evaluando todas las rutas posibles entre ellos, sus respectivas fuerzas, y conservando aquella que maximiza el mínimo. El algoritmo de Floyd-Warshall adaptado permite automatizar este proceso para conjuntos grandes de forma eficiente.

A continuación se muestra la matriz de rutas más fuertes $P[u_i, u_j]$ obtenida a partir de la matriz D anterior:

$P[u_i, u_j]$	A	B	C	D	E	F
A	–	2	2	2	3	3
B	2	–	2	2	3	3
C	2	2	–	3	3	3
D	2	2	2	–	3	3
E	0	0	0	0	–	2
F	0	0	0	0	1	–

Resultado: Una vez aplicados los pasos anteriores y construida la matriz $P[u_i, u_j]$, se procede a comparar cada par de elementos (u_i, u_j) para determinar cuál de los dos presenta una ruta más fuerte hacia el otro. Si $P[u_i, u_j] > P[u_j, u_i]$, se considera que u_i es preferido a u_j . A partir de todas estas comparaciones, se genera una relación de dominancia global entre los elementos del conjunto.

Cuando varios elementos ganan al mismo número de oponentes, como ocurre con A , B y D (que únicamente superan estrictamente a E y F), el método de Schulze no se basa únicamente en el número de victorias, sino que recurre a la comparación directa de las rutas fuertes entre los elementos empatados. Para cada par se evalúan los valores $P[u_i, u_j]$ y $P[u_j, u_i]$, y si estas comparaciones también resultan en empate, se examina la distribución general de las rutas de cada elemento frente al resto del conjunto. De esta forma, el algoritmo puede identificar cuál de ellos tiene una posición estructuralmente más ventajosa en el grafo de dominancia.

En este ejemplo, se obtiene finalmente el siguiente ranking global:

$$C \succ B \succ A \succ D \succ E \succ F$$

Este resultado evidencia cómo el método de Schulze logra integrar tanto comparaciones directas como caminos indirectos en el proceso de agregación, obteniendo una ordenación global que es robusta frente a inconsistencias locales o ciclos de preferencia entre los rankings individuales.

Nota sobre los ciclos de Condorcet: El método de Schulze está especialmente diseñado para afrontar situaciones en las que se presentan *ciclos de Condorcet*, es decir, configuraciones donde A gana a B , B gana a C , y C gana a A . Estos ciclos impiden establecer una preferencia clara mediante comparaciones directas. Sin embargo, al utilizar rutas más fuertes y comparar caminos indirectos, el método consigue establecer una dominancia razonada, proporcionando una ordenación global incluso en contextos donde las preferencias individuales se contradicen mutuamente².

3.5. Método de Footrule

Definición y fundamento

El *método de Footrule* es una técnica de agregación de rankings que busca minimizar la suma de las discrepancias absolutas entre las posiciones

²Para una discusión formal sobre los ciclos de Condorcet y su relación con sistemas de votación, véase [6].

de los elementos en el ranking agregado y en los rankings individuales. Estas discrepancias se cuantifican mediante la **distancia Footrule** (Spearman Footrule), definida como:

$$d_F(\sigma, \sigma_j) = \sum_{i=1}^n |\sigma(i) - \pi_j(i)|$$

donde σ es el ranking agregado y σ_j un ranking individual. El objetivo es encontrar la permutación σ^* que minimice la suma total de estas distancias con respecto a los k rankings dados:

$$\sigma^* = \arg \min_{\sigma} \sum_{j=1}^k d_F(\sigma, \sigma_j)$$

Este método ha sido utilizado en tareas de meta-búsqueda, sistemas de recomendación y combinación de resultados provenientes de múltiples fuentes [5, 4].

Pseudocódigo y explicación

En este trabajo se ha optado por implementar una versión exacta del método de Footrule, formulando el problema como una tarea de *asignación óptima*. Para ello, se construye una matriz de costes C de tamaño $n \times n$, donde $C[i][j]$ representa el coste de asignar el elemento u_i a la posición j del ranking final.

Este coste se calcula como la suma de diferencias absolutas entre la posición j y las posiciones que ocupa u_i en los rankings individuales:

$$C[i][j] = \sum_{r=1}^k |\pi_r(u_i) - j|$$

Una vez construida la matriz de costes, se resuelve el problema de asignación utilizando el **algoritmo húngaro**³ (Kuhn-Munkres), que garantiza encontrar la asignación óptima de elementos a posiciones con coste mínimo total.

³El algoritmo húngaro, propuesto por Harold W. Kuhn, resuelve el problema de asignación mínima en grafos bipartitos. Se basa en una técnica de mejora iterativa que garantiza encontrar una correspondencia óptima entre filas y columnas de una matriz de costes. Para más detalles, véase [11].

Algoritmo 5: Método de Footrule (versión exacta)**Input:** $\mathcal{R} = \{\sigma_1, \dots, \sigma_k\}$ rankings sobre $U = \{u_1, \dots, u_n\}$ **Output:** Ranking agregado σ^* Inicializar matriz $C[i][j] \leftarrow 0$ para $1 \leq i, j \leq n$;**foreach** $u_i \in U$ **do**

	foreach posición $j = 1$ to n do
	$C[i][j] \leftarrow \sum_{r=1}^k \sigma_r(u_i) - j $;

Resolver el problema de asignación mínima sobre C (algoritmo húngaro);**return** permutación σ^* según la asignación óptima

Esta aproximación tiene la ventaja de obtener una solución óptima, manteniendo una ejecución eficiente en problemas de tamaño moderado.

Análisis de complejidad

La complejidad del algoritmo depende de dos fases:

- La construcción de la matriz de costes C tiene una complejidad de $O(n^2k)$, ya que para cada una de las n filas (elementos) y n columnas (posiciones posibles), se suma sobre k rankings.
- La resolución del problema de asignación óptima mediante el algoritmo húngaro tiene una complejidad de $O(n^3)$ en el peor caso.

Por tanto, la complejidad total del método implementado es:

$$O(n^2k + n^3)$$

Este coste, aunque superior al de métodos como Borda o la heurística basada en medias, es perfectamente razonable para conjuntos con un número moderado de elementos (hasta unas pocas centenas). Además, ofrece una solución óptima garantizada para la distancia Footrule.

Propiedades, ventajas y desventajas

A continuación se resumen las principales características del método Footrule, apoyadas en las referencias [5, 4, 2].

- ✓ **Solución óptima exacta:** Este método permite encontrar el ranking agregado que minimiza exactamente la suma de distancias Footrule (diferencias absolutas en las posiciones), utilizando algoritmos como el húngaro para resolver el problema de asignación [5].

- ✓ **Aplicabilidad práctica destacada:** Es especialmente eficaz en contextos como la fusión de resultados en motores de búsqueda y rankings en sistemas de recomendación, donde la precisión posicional es más relevante que otras propiedades teóricas [4].
- ✗ **No cumple el criterio de Condorcet:** Dado que opera únicamente sobre posiciones y no sobre comparaciones por pares, no garantiza que un posible ganador de Condorcet sea situado en la primera posición [5].
- ✗ **Implementación no trivial:** Aunque está bien formalizado, requiere aplicar técnicas de programación lineal combinatoria, como el algoritmo húngaro, lo que implica una implementación más técnica que métodos como Borda [5].
- ✗ **Complejidad computacional considerable:** Su coste es superior al de métodos heurísticos simples. Aunque se puede resolver en tiempo polinómico, la complejidad práctica crece rápidamente con el número de elementos, haciendo necesario optimizar o paralelizar la ejecución en grandes conjuntos de datos [2].
- ✗ **Dificultad de interpretación:** La métrica Footrule es intuitiva desde un punto de vista matemático, pero menos transparente para usuarios no técnicos. No se basa en comparaciones entre elementos, lo que puede dificultar su justificación en contextos donde se esperan argumentos ordinales [4].

Ejemplo ilustrativo

Consideramos el mismo conjunto de elementos $U = \{A, B, C, D, E, F\}$ y los siguientes rankings individuales:

- $\sigma_1: A \succ B \succ C \succ D \succ E \succ F$
- $\sigma_2: B \succ C \succ D \succ A \succ F \succ E$
- $\sigma_3: C \succ D \succ A \succ B \succ E \succ F$

Para aplicar el método de Footrule en su versión exacta, se construye una matriz de costes C , donde cada entrada $C[i][j]$ representa el coste de asignar el elemento u_i a la posición j en el ranking final. Este coste se calcula como la suma de las diferencias absolutas entre la posición j y las posiciones que ocupa u_i en los rankings individuales.

Por ejemplo, el coste de asignar el elemento A a la posición 1 es:

$$|\sigma_1(A) - 1| + |\sigma_2(A) - 1| + |\sigma_3(A) - 1| = |1 - 1| + |4 - 1| + |3 - 1| = 0 + 3 + 2 = 5$$

La matriz de costes completa es la siguiente:

Elemento	Pos 1	Pos 2	Pos 3	Pos 4	Pos 5	Pos 6
A	5	4	3	4	7	10
B	4	3	4	5	8	11
C	3	2	3	6	9	12
D	6	3	2	3	6	9
E	13	10	7	4	1	2
F	12	9	6	3	2	1

A partir de esta matriz, se resuelve un problema de asignación mínima, en el que cada elemento debe ocupar una posición del ranking final, de manera que el coste total global sea mínimo. Para ello, se utiliza el **algoritmo húngaro** (Kuhn-Munkres).

Este algoritmo transforma la matriz de costes mediante restas fila/columna, encuentra ceros que representan asignaciones viables y busca cubrirlos con el mínimo número de líneas hasta construir una solución óptima. Por ejemplo, observando la matriz de costes, puede verse que:

- El coste mínimo para A es en la posición 3 (coste 3).
- Para B, la mejor opción es la posición 2 (coste 3).
- Para C, la mejor opción es la posición 2 (coste 2), pero ya está ocupada por B, así que se opta por la siguiente más baja (posición 1, coste 3), y así sucesivamente hasta completar el emparejamiento óptimo sin repetir posiciones.

La siguiente tabla muestra el resultado del algoritmo húngaro, con la asignación óptima de cada elemento a su posición final, así como el coste asociado:

Elemento	Posición asignada	Coste
C	Pos 1	3
B	Pos 2	3
A	Pos 3	3
D	Pos 4	3
E	Pos 5	1
F	Pos 6	1

El resultado de aplicar el algoritmo completo a esta matriz da lugar al siguiente ranking final:

$$C \succ B \succ A \succ D \succ E \succ F$$

Este ranking es el que minimiza exactamente la suma de distancias Footrule con respecto a los tres rankings individuales.

3.6. Método basado en Cadenas de Markov

Definición y fundamento

El método de agregación basado en *cadenas de Markov* modela la preferencia entre elementos como un proceso estocástico. Cada elemento se representa como un estado en una cadena, y las probabilidades de transición se derivan de la frecuencia relativa con la que un elemento es preferido a otro en los rankings individuales.

Este enfoque fue popularizado en el contexto de fusión de resultados web y algoritmos de ranking como PageRank, y se caracteriza por su capacidad para incorporar información directa e indirecta, así como su tolerancia ante datos incompletos o ruidosos [5, 3].

Pseudocódigo y explicación

La idea central consiste en construir una matriz de transición P donde P_{ij} representa la probabilidad de pasar del elemento u_i al u_j . Estas probabilidades se derivan de las comparaciones por pares en los rankings individuales: si u_i aparece por encima de u_j , se incrementa el contador correspondiente. A partir de la matriz de preferencias se construye la matriz de transición, que posteriormente se utiliza para calcular la distribución estacionaria π , cuyo valor para cada elemento representa su importancia relativa en el ranking agregado, por tanto, la distribución estacionaria de una cadena de Markov es el vector de probabilidad al que converge el sistema tras muchas iteraciones. Se puede calcular mediante métodos iterativos como potencias sucesivas o mediante la resolución del sistema lineal $\pi(P - I) = 0$ con la restricción $\sum_i \pi(i) = 1$.

Las estructuras utilizadas en el algoritmo son:

- $\mathcal{R} = \{\sigma_1, \dots, \sigma_k\}$: conjunto de rankings individuales estrictamente lineales sobre el universo U .
- $U = \{u_1, \dots, u_n\}$: conjunto de elementos a ordenar.

- $M[i][j]$: matriz de comparaciones por pares, donde se acumula cuántas veces el elemento u_i ha sido preferido a u_j .
- $P[i][j]$: matriz de transición probabilística obtenida normalizando las filas de M .
- π : distribución estacionaria de la cadena de Markov definida por P , que se interpreta como el ranking final.

Algoritmo 6: Método basado en Cadenas de Markov

Input: $\mathcal{R} = \{\sigma_1, \dots, \sigma_k\}$ rankings sobre $U = \{u_1, \dots, u_n\}$

Output: Ranking agregado σ^*

Inicializar matriz de preferencias $M[i][j] \leftarrow 0$ para $1 \leq i, j \leq n$;

foreach ranking $\sigma_r \in \mathcal{R}$ **do**

foreach par (u_i, u_j) tal que $u_i \succ u_j$ en σ_r **do**
 $M[i][j] \leftarrow M[i][j] + 1$;

Construir matriz de transición P normalizando cada fila de M ;

Calcular distribución estacionaria π tal que $\pi P = \pi$;

Ordenar U en orden decreciente según los valores de π ;

return σ^*

Este enfoque se basa en la suposición de que los elementos más relevantes son aquellos hacia los que más probablemente se "transiciona.^a lo largo del proceso.

Análisis de complejidad

El método se divide en dos fases principales:

- La construcción de la matriz de transición tiene un coste de $O(n^2k)$, ya que se comparan todos los pares en cada ranking.
- El cálculo de la distribución estacionaria mediante métodos iterativos como potencias sucesivas o algoritmos de eigenvectores tiene coste entre $O(n^2)$ y $O(n^3)$, dependiendo del método usado y la tolerancia deseada.

En conjunto, el coste total se puede considerar como:

$$O(n^2k + n^2)$$

Este coste es asumible incluso para conjuntos grandes, lo que hace que este método sea adecuado para aplicaciones a gran escala, como motores de búsqueda o sistemas de recomendación [2].

Propiedades, ventajas y desventajas

Las principales propiedades de este método se basan en su formulación probabilística y su capacidad para incorporar información implícita a través del grafo de comparaciones. Según [3, 2, 5]:

- ✓ **Robustez frente a inconsistencias y datos faltantes:** Al modelar las preferencias como una cadena de Markov y analizar la distribución estacionaria, el método se muestra menos sensible a rankings individuales anómalos o incompletos [3].
- ✓ **Captura de relaciones indirectas:** Un elemento puede alcanzar una alta posición incluso si no domina en comparaciones directas, gracias a su accesibilidad a través de rutas probabilísticas fuertes en el grafo de preferencias [2].
- ✓ **Buena escalabilidad:** El cálculo de la distribución estacionaria mediante métodos iterativos permite aplicar el algoritmo a conjuntos grandes sin requerir una evaluación exhaustiva de todas las permutaciones posibles [5].
- ✗ **No cumple el criterio de Condorcet:** Aunque puede reflejar popularidad global, no garantiza que un candidato de Condorcet (si existe) sea el primero en el ranking final, ya que la lógica del método no se basa en comparaciones uno a uno [5].
- ✗ **Implementación intermedia:** Aunque no es extremadamente compleja, requiere una formulación matricial cuidadosa, una normalización coherente y técnicas de álgebra lineal para obtener la distribución estacionaria [3].
- ✗ **Sensibilidad a la definición de la matriz P :** Si la matriz de transición está mal normalizada, o si hay pocas comparaciones disponibles entre algunos elementos, se puede inducir un sesgo sistemático en el ranking [2].
- ✗ **Interpretabilidad reducida:** La lógica detrás de la posición final depende de un modelo de absorción probabilística y no de una lógica ordinal directa, lo que puede dificultar su comprensión por parte de perfiles no técnicos [3].

Ejemplo ilustrativo

Consideramos los mismos rankings individuales:

- $\sigma_1: A \succ B \succ C \succ D \succ E \succ F$

- σ_2 : B \succ C \succ D \succ A \succ F \succ E
- σ_3 : C \succ D \succ A \succ B \succ E \succ F

Paso 1: Matriz de preferencias Se cuenta cuántas veces cada elemento u_i es preferido a u_j entre todos los rankings:

$M[i][j]$	A	B	C	D	E	F
A	–	2	1	1	3	3
B	1	–	2	2	3	3
C	2	1	–	3	3	3
D	2	1	0	–	3	3
E	0	0	0	0	–	2
F	0	0	0	0	1	–

Paso 2: Matriz de transición P Cada fila se normaliza dividiendo entre la suma total de la fila:

$$P_{ij} = \frac{M[i][j]}{\sum_j M[i][j]}$$

Esto da lugar a una matriz de probabilidades que define la cadena de Markov.

$P[i][j]$	A	B	C	D	E	F
A	0.0000	0.2000	0.1000	0.1000	0.3000	0.3000
B	0.0909	0.0000	0.1818	0.1818	0.2727	0.2727
C	0.1667	0.0833	0.0000	0.2500	0.2500	0.2500
D	0.2222	0.1111	0.0000	0.0000	0.3333	0.3333
E	0.0000	0.0000	0.0000	0.0000	0.0000	1.0000
F	0.0000	0.0000	0.0000	0.0000	1.0000	0.0000

Paso 3: Distribución estacionaria π Se calcula el vector π que representa la distribución estacionaria de la cadena de Markov definida por la matriz de transición P . El valor $\pi(i)$ indica la relevancia relativa del elemento u_i en el ranking final:

Elemento	Valor de π
E	0.5000000000
F	0.5000000000
D	0.0000000096
A	0.0000000094
B	0.0000000083
C	0.0000000060

Resultado: El ranking final resultante, ordenando los elementos de mayor a menor según sus valores en π , es:

$$E = F \succ D \succ A \succ B \succ C$$

En este ejemplo, la distribución estacionaria concentra toda la probabilidad en los elementos E y F , lo que indica que el proceso de Markov termina absorbiéndose en ese subconjunto.

Esto se debe a la estructura de la matriz de transición: desde A , B , C y D es posible llegar eventualmente a E o F , pero una vez que se entra en ese ciclo cerrado entre ambos elementos, ya no se puede salir. Con el paso de las iteraciones, toda la "probabilidad" se va concentrando en ese ciclo.

El método de Markov no busca quién gana más veces, sino a qué elementos se llega con mayor frecuencia si se camina aleatoriamente a través de las comparaciones, muchas veces. En este caso, E y F están conectados entre sí y nadie puede sacarles de ahí, mientras que el resto de elementos pierden relevancia porque no pueden recibir visitas.

3.7. Comparación de los Algoritmos

La Tabla 3.2 sintetiza los principales algoritmos estudiados en este trabajo, destacando tanto su eficiencia teórica como los escenarios en los que su uso resulta más apropiado. Esta comparación permite orientar la elección del método según las restricciones computacionales y la naturaleza de los datos del problema. Cada método presenta ventajas específicas que lo hacen más adecuado para ciertos tipos de problemas, dependiendo del tamaño de la instancia, la necesidad de precisión, la disponibilidad de los datos y la tolerancia al ruido.

Comparación de resultados sobre el ejemplo base

A continuación se recogen los rankings de consenso generados por cada uno de los algoritmos aplicados al conjunto de rankings individuales utilizado como ejemplo común en este capítulo 3. Esta comparación permite visualizar de forma clara cómo varía el resultado final según el método empleado.

Como se muestra en la tabla 3.3 la mayoría de los algoritmos coinciden en situar a los elementos C , B y A en las primeras posiciones y los elementos E y F en las últimas, lo que sugiere una preferencia colectiva clara. No obstante, se observan diferencias relevantes: el método de Copeland genera empates múltiples, y el algoritmo de Cadenas de Markov ofrece un resultado

Algoritmo	Complejidad teórica	Aplicaciones típicas
Método de Borda	$\mathcal{O}(nk + n \log n)$	Evaluaciones deportivas, sistemas de votación rápidos, motores de búsqueda.
Método de Copeland	$\mathcal{O}(n^2k)$	Votaciones con enfoque Condorcet, torneos, decisiones políticas simples.
Método de Kemeny-Young	$\mathcal{O}(n! \cdot k \cdot n \log n)$ (intratabilidad en general)	Procesos donde se requiere una alta fidelidad al consenso, como votaciones precisas o toma de decisiones crítica.
Método de Schulze	$\mathcal{O}(n^3)$	Elecciones políticas, sistemas robustos de decisión colectiva.
Footrule	$\mathcal{O}(n^2k + n^3)$	Sistemas de recomendación con coste computacional asumible.
Cadenas de Markov	$\mathcal{O}(n^2k + n^2)$	Sistemas de recomendación, rankings con ruido o incompletos, modelos tipo PageRank.

Cuadro 3.2: Comparativa de los algoritmos de agregación de rankings.

Algoritmo	Ranking de consenso obtenido
Borda	$C \succ B \succ A \succ D \succ E \succ F$
Copeland	$B = C \succ A = D \succ E \succ F$
Kemeny-Young	$B \succ C \succ D \succ A \succ E \succ F$
Schulze	$C \succ B \succ A \succ D \succ E \succ F$
Footrule	$C \succ B \succ A \succ D \succ E \succ F$
Cadenas de Markov	$E = F \succ D \succ A \succ B \succ C$

Cuadro 3.3: Rankings de consenso generados por cada algoritmo sobre el conjunto de entrada.

radicalmente distinto, con E y F en la cima. Este contraste evidencia que la elección del algoritmo tiene un impacto directo sobre el ranking final.

Por tanto, se puede concluir que, desde una perspectiva teórica, y considerando el equilibrio entre simplicidad, robustez y expresividad, se puede argumentar que **el método de Borda es adecuado para situaciones sencillas o de baja complejidad estructural**, donde se desea una solución eficiente, interpretable y sin ambigüedades. Para escenarios más exigentes —tales como procesos de toma de decisiones sensibles, aplicaciones con grandes volúmenes de datos o entornos donde se requiere una alta resistencia frente a inconsistencias o intentos de manipulación—, **el algoritmo de Schulze representa una opción más robusta**. Este puede considerarse una *extensión mejorada del método de Copeland*, ya que resuelve dos de sus principales limitaciones: (i) incorpora rutas de victoria indirectas —captu-

rando, por ejemplo, que si A gana a B y B gana a C, entonces A debería tener cierta ventaja sobre C aunque no lo haya vencido directamente—, y (ii) minimiza la incidencia de empates al evaluar la fuerza de las rutas preferenciales. Además, frente al método de Footrule, Schulze presenta ventajas en términos de eficiencia computacional y estabilidad estructural en elecciones de gran escala. Por tanto, la elección del algoritmo debe basarse no sólo en el rendimiento empírico, sino también en las propiedades teóricas que mejor se ajusten a la naturaleza del problema y al tipo de decisiones que se desean respaldar.

Capítulo 4

Herramienta para el análisis y la comparación de algoritmos

Esta herramienta ha sido diseñada con el objetivo de facilitar y ejemplificar el uso de algoritmos de agregación de rankings, como parte del desarrollo de este proyecto. Su código fuente está disponible en un repositorio público de GitHub, que además permite gestionar las versiones del proyecto: <https://github.com/carmenqr/TFG>. La herramienta se presenta como una aplicación web, implementada en **Python** utilizando **Streamlit** para la creación de la interfaz de usuario. Para la ejecución de los algoritmos de agregación y la generación de algunas gráficas de comparación, se emplea el paquete **pyRankMCDA**. Además, se utiliza **SQLite** como sistema de almacenamiento local de los datos en una base de datos.

4.1. Funcionalidad

La herramienta desarrollada proporciona un entorno interactivo que permite:

- Cargar conjuntos de rankings desde archivos Excel, siguiendo un formato predefinido.
- Visualizar los rankings cargados de manera estructurada.
- Ejecutar distintos algoritmos de agregación de rankings, seleccionados en función del tipo de datos (considerando posibles variantes como empates o ponderaciones).

- Almacenar los rankings agregados en una base de datos SQLite para su posterior consulta y análisis.
- Comparar resultados obtenidos mediante diferentes algoritmos de agregación, a través de gráficas y métricas estadísticas.

El sistema está organizado en varias secciones o pestañas, lo que permite una navegación intuitiva para el usuario, incluso sin experiencia técnica previa. A nivel interno, la aplicación sigue una arquitectura modular, en la que las funcionalidades principales se distribuyen en distintos archivos y submódulos. La lógica general y el punto de entrada se encuentran en `main.py`, mientras que la gestión de la base de datos se centraliza en `db.py`, utilizando un archivo SQLite (`rankings.db`) como almacenamiento persistente. Funciones auxiliares y herramientas comunes se encapsulan en `utils.py`. Además, las distintas vistas de la interfaz se agrupan en la carpeta `tabs`, que contiene los módulos `agregaciones.py`, `rankings.py` y `resultados.py`, cada uno correspondiente a una sección específica de la aplicación. Finalmente, el archivo `requirements.txt` documenta las dependencias necesarias para la ejecución del sistema.

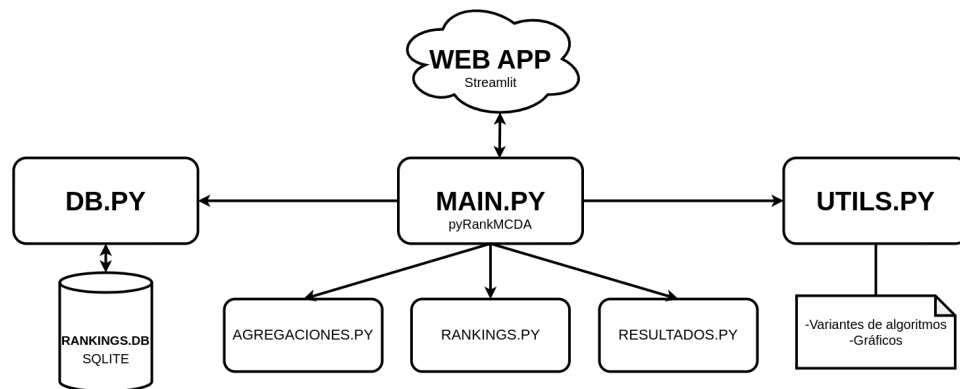


Figura 4.1: Arquitectura de la herramienta de agregación de rankings.

4.1.1. Manual de usuario

La estructura de la aplicación está diseñada de forma muy guiada, permitiendo que cualquier usuario pueda utilizarla sin necesidad de conocimientos técnicos previos. El flujo de trabajo se divide en tres pasos principales, bien diferenciados y accesibles desde la barra lateral de navegación:

1. **Nuevo problema:** permite cargar conjuntos de rankings desde un archivo Excel y gestionar los datos del problema.

2. **Resolver problema:** ofrece distintas técnicas de agregación para obtener un ranking de consenso a partir de los rankings individuales.
3. **Ver resultados:** permite analizar los resultados obtenidos mediante gráficos y métricas de comparación entre rankings.

A continuación se presenta un breve tutorial de uso de la herramienta, mostrando de forma práctica cómo cargar rankings, aplicar algoritmos de agregación y analizar los resultados obtenidos. Para ello, se describen paso a paso las principales funcionalidades disponibles en la aplicación web.

Página de inicio

Al abrir la aplicación, se muestra una pantalla de bienvenida desde la cual el usuario puede iniciar el uso de la herramienta pulsando el botón *Empezar*. Esta vista actúa como punto de entrada al flujo principal de trabajo.

ARPAR: Asistente para Resolución de Problemas de Agregación de Rankings

Esta herramienta está diseñada con el objetivo asistirle en la resolución de problemas de agregación de rankings de forma eficaz, intuitiva y confiable.

Con ella podrá importar conjuntos de rankings desde archivos excel, aplicar distintos métodos de agregación personalizados y visualizar gráficamente los resultados obtenidos.

¿Desea comenzar?

Empezar

Desarrollada por Carmen Quiles Ramírez como parte del Trabajo de Fin de Grado (TFG) - Junio de 2025.

Figura 4.2: Pantalla de bienvenida.

Menú lateral

La aplicación cuenta con un menú lateral desde el cual se accede a las tres etapas principales del proceso de resolución: *1º) Nuevo problema*, *2º) Resolver problema* y *3º) Ver resultados*. Cada una de estas etapas contiene a su vez distintas subsecciones, accesibles mediante pestañas, que guían al usuario de forma estructurada a lo largo del flujo de trabajo.



Figura 4.3: Menú lateral.

Añadir rankings

Dentro de la sección *Nuevo problema*, esta subsección permite cargar un grupo de rankings a partir de un archivo Excel (.xlsx) siguiendo un formato específico. Este formato varía dependiendo del tipo de rankings que queramos añadir:

	0	1	2	3
	Nombre del grupo			
0	PruebaSchuzule	None	None	None
		Nombre de rankings		
1	None	R1	R2	R3
	Nombre de elementos			
2	Elem 1	1	2	2
3	Elem 2	2	1	1
4	Elem 3	3	3	3

Figura 4.4: Formato excel rankings simples

	0	1	2	3
	Nombre del grupo	Tipo de variante		
0	PruebaGrandeEmpates	empates	None	None
			Nombre de rankings	
1	None	R1	R2	R3
	Nombre de elementos			
2	Elem 1	1	4	2
3	Elem 2	2	6	7
4	Elem 3	3	1	5
5	Elem 4	4	3	4

Figura 4.5: Formato excel rankings con empates

	0	1	2	3
	Nombre del grupo	Tipo de variante		
0	PruebaChicaPonderaciones	Ponderaciones	None	None
			Nombre de rankings	
1	None	R1	R2	R3
	Ponderaciones			
2	Ponderación	0.5	0.2	0.3
	Nombre de elementos			
3	Elem 1	1	2	2
4	Elem 2	2	1	1
5	Elem 3	3	3	3

Figura 4.6: Formato excel rankings con ponderaciones

En general, el formato del archivo Excel debe respetar una estructura

específica. La celda $[0, 0]$ debe contener el nombre del grupo de rankings. A continuación, dejando en blanco las celdas $[1, 0]$ y $[0, 1]$, se deben introducir los nombres de los elementos en las filas y los nombres de los rankings en las columnas. La matriz resultante se completa indicando, en cada celda, la posición correspondiente del elemento x en el ranking y .

En el caso de trabajar con variantes, es necesario especificarlo en la celda $[0, 1]$, indicando el tipo de variante (por ejemplo, *empates* o *ponderaciones*). En particular, para las ponderaciones, se debe añadir una fila adicional que contenga el valor de ponderación asociado a cada ranking.

Los pasos a seguir para subir un archivo son los siguientes (véase Figura 4.7): en primer lugar, pulsar el botón *Browse files* y seleccionar el archivo que se desea cargar. A continuación, tras comprobar que el contenido es correcto mediante la previsualización mostrada, se debe pulsar el botón *Guardar* para almacenar los datos en la base de datos.

NUEVO PROBLEMA

[Añadir Rankings](#) [Eliminar Rankings](#) [Ver Rankings](#)

Añadir Rankings desde Excel

Añade el grupo de rankings que contiene su problema

Cargue un archivo Excel (.xlsx)

Drag and drop file here
Límite: 200MB por fila + 10,50

[Prueba Schulze.xlsx](#) 5.0KB

[Browse Files](#) 1

Vista previa del Excel:

	0	1	2	3
0	PruebaSchulze	None	None	None
1	None	R1	R2	R3
2	Elem 1	1	2	2
3	Elem 2	2	1	1
4	Elem 3	3	3	3

[Guardar](#) 2

Figura 4.7: Pantalla de carga de rankings desde archivo Excel.

A partir de este momento, estos rankings quedarán identificados como nuestro problema a resolver.

Eliminar rankings

Dentro de la sección *Nuevo problema*, esta subsección permite eliminar los rankings de un problema previamente almacenado en la base de datos. Para ello, se debe seleccionar el problema que se desea eliminar y, a continuación, pulsar el botón *Eliminar problema*. En caso de que el problema seleccionado contenga soluciones (agregaciones) realizadas, estas también serán eliminadas de la base de datos de manera automática.

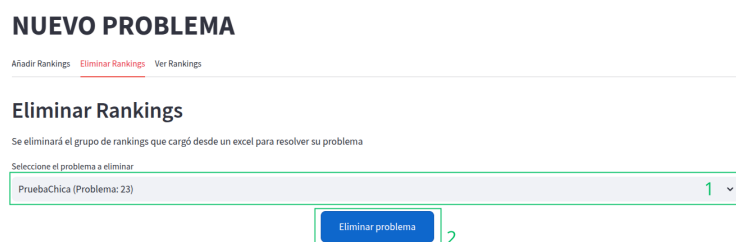


Figura 4.8: Pantalla de eliminación de rankings.

Ver rankings

Dentro de la sección *Nuevo problema*, esta subsección permite visualizar los rankings almacenados en la base de datos de forma estructurada. El formato mostrado es similar al del archivo Excel original, donde las filas representan los elementos y las columnas corresponden a los diferentes rankings. Para su visualización, primero se debe seleccionar el problema a visualizar, tras lo cual se mostrará la tabla correspondiente justo debajo.

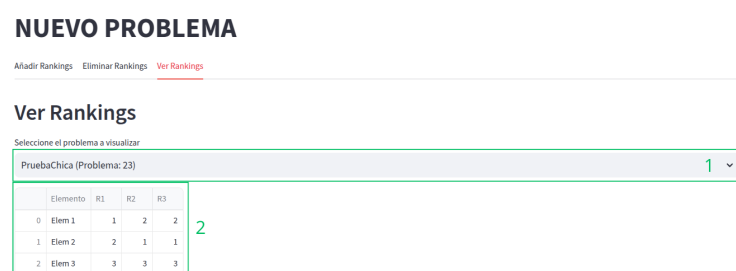


Figura 4.9: Vista de los rankings almacenados.

Resolver

Dentro de la sección *Resolver problema*, esta subsección permite ejecutar algoritmos de agregación sobre los rankings de un problema seleccionado. El usuario puede elegir entre los métodos de Borda, Copeland, Kemeny-Young, Schulze y Footrule. Además, si se detectan ponderaciones o empates en los datos, únicamente se desplegarán las opciones correspondientes: Borda ponderado, Copeland ponderado o Borda con empates, respectivamente.

Los pasos a seguir son los siguientes: en primer lugar, se debe seleccionar el problema a resolver; a continuación, elegir el algoritmo de agregación entre los disponibles; posteriormente, pulsar el botón *Resolver*; y, finalmente, se mostrará una vista previa del resultado obtenido para el ranking de consenso.

RESOLVER PROBLEMA

Resolver Ver Soluciones Eliminar Soluciones

Resolver

Seleccione el problema a resolver

PruebaChica (Problema: 23) 1 ▾

Seleccione el algoritmo de agregación con el que resolver el problema

Borda 2 ▾

Resolver

3

Ranking de consenso

value	
	2
	1
	3

Ranking de consenso guardado correctamente.

Figura 4.10: Resolver un problema seleccionado.

De esta forma queda creada una solución con uno de los algoritmos disponibles para nuestro problema.

Ver soluciones

Esta subsección está dentro de *Resolver problema*, muestra las soluciones de las agregaciones realizadas, permitiendo comparar visualmente los rankings originales con el ranking de consenso obtenido. En primer lugar, se debe seleccionar el problema del que se desea ver la solución; a continuación, se desplegarán las distintas soluciones que se hayan realizado sobre dicho grupo, permitiendo seleccionar entre las disponibles generadas con diferentes algoritmos. Finalmente, se mostrará una tabla donde se visualizan las posiciones de cada ranking individual junto con la posición final del ranking de consenso.

RESOLVER PROBLEMA

Resolver Ver Soluciones Eliminar Soluciones

Ver Soluciones

Seleccione el problema del que ver la solución

PruebaChica (Problema: 23) 1 ▾

Seleccione el algoritmo del que desee ver la solución para el problema seleccionado

Borda (Agregación: 38) 2 ▾

	Elemento	R1	R2	R3	Ranking de consenso
0	Elem 1	1	2	2	2
1	Elem 2	2	1	1	1
2	Elem 3	3	3	3	3

Figura 4.11: Vista previa de las soluciones.

Eliminar soluciones

Como última subsección dentro de *Resolver problema*, encontramos eliminar soluciones, la cual permite eliminar una solución realizada sobre un problema específico. Para ello, se debe seleccionar el problema correspondiente, a continuación elegir la solución que se desea eliminar (identificada por el algoritmo utilizado) y, finalmente, pulsar el botón *Eliminar solución*.

The screenshot shows a web interface titled "RESOLVER PROBLEMA". Below the title are three tabs: "Resolver", "Ver Soluciones", and "Eliminar Soluciones", with the last one being active. The main heading is "Eliminar Soluciones". Below this, there are two dropdown menus. The first is labeled "Seleccione el problema del que eliminar una solución ejecutada" and has "PruebaChica (Problema: 23)" selected. The second is labeled "Seleccione la solución que desea eliminar" and has "Borda (Agregación: 38)" selected. Below the second dropdown is a blue button labeled "Eliminar Solución". A green box highlights the button, and a green number "3" is next to it.

Figura 4.12: Pantalla de eliminación de soluciones.

Ver resultados

En esta sección se ofrece la visualización de resultados de dos maneras diferentes. La primera de ellas, como se muestra en la Figura 4.13, es analizando el resultado del problema general. En este caso, se analizan simultáneamente todas las soluciones (agregaciones) que se han realizado sobre un mismo problema. Para ello, simplemente se debe seleccionar el problema del que se desean comparar las diferentes soluciones.

A continuación, se presentan dos gráficas interactivas que permiten visualizar de forma clara las diferencias entre los algoritmos aplicados a un mismo problema. En la primera, la Figura 4.14, se muestra un mapa de calor en el que se representan las posiciones de los rankings de consenso obtenidos por cada uno de los algoritmos aplicados.

Estas gráficas son completamente interactivas, lo que permite al usuario seleccionar o deseleccionar algoritmos específicos para observar con mayor detalle las distintas soluciones generadas.

Por otro lado, la Figura 4.15 presenta una gráfica de tipo radar, también basada en las posiciones de los rankings de consenso. Esta visualización permite detectar de forma intuitiva el grado de similitud o diferencia entre los resultados: cuanto más se separen las formas en el gráfico, mayor será la divergencia entre los rankings generados por los distintos algoritmos.

VER RESULTADOS

Tipo de resultado

☒ Resultado del problema general 1

☐ Resultado por tipo de solución ejecutada

Seleccione el problema del que ver los resultados

PruebaGrande (Problema: 24) 2

Figura 4.13: Resultados problema general.

Mapa de calor con las posiciones de los rankings de consensos

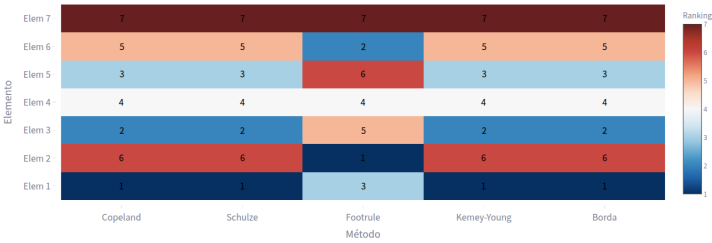


Figura 4.14: Mapa de calor.

Radar para ver la similitud entre los rankings de consenso

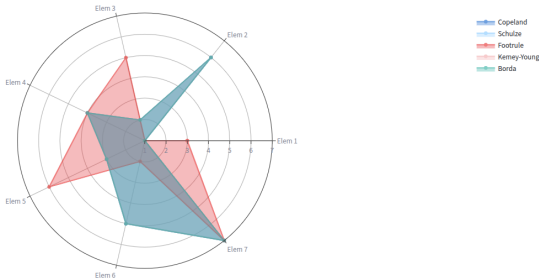


Figura 4.15: Radar de posiciones.

La segunda forma de observar los resultados es según el tipo de solución ejecutada. En este caso, se selecciona el problema y, a continuación, una de las soluciones previamente realizadas sobre dicho grupo. A partir de la solución seleccionada, se analiza el resultado obtenido con el algoritmo específico utilizado.

Para ello, se muestra una tabla (véase Figura 4.17) que recoge las posiciones de cada ranking individual, las posiciones finales del ranking de

consenso y las diversas métricas calculadas, como la distancia de Kendall, el coeficiente de Kendall, el coeficiente de Spearman y el coeficiente WS. Además, se presentan diferentes gráficas comparativas e interactivas que permiten analizar de forma visual las diferencias entre los rankings.

VER RESULTADOS

Tipo de resultado

☐ Resultado del problema general
☒ Resultado por tipo de solución ejecutada ¹

Seleccione el problema del que ver los resultados

PruebaGrande (Problema: 24) 2 ▾

Seleccione una solución ejecutada

Copeland (Agregación: 44) 3 ▾

▼ Ver los resultados de la solución

Figura 4.16: Resultados por solución ejecutada.

Tabla con métricas de distancia

Distancias entre el ranking de consenso y los rankings individuales

	Elemento	R1	R2	R3	Ranking de consenso
0	Elem 1	1	4	2	1
1	Elem 2	2	6	7	6
2	Elem 3	3	1	5	2
3	Elem 4	4	3	4	4
4	Elem 5	5	2	3	3
5	Elem 6	6	5	1	5
6	Elem 7	7	7	6	7
7	Distancia Kendall	5	3	7	None
8	Coeficiente Kendall	0.5238	0.7143	0.3333	None
9	Coeficiente Spearma	0.6071	0.7857	0.5	None
10	Coeficiente WS	0.75	0.7729	0.5888	None

Figura 4.17: Tabla con los datos de la solución.

Entre las diferentes gráficas disponibles para el análisis de resultados, se incluyen varias visualizaciones interactivas que facilitan la interpretación y comparación entre rankings individuales y rankings de consenso para el algoritmo seleccionado en la resolución del problema. En primer lugar, la Figura 4.18 muestra una gráfica de barras que representa las posiciones de los elementos en cada uno de los rankings individuales, junto con el

ranking de consenso, lo que permite identificar visualmente coincidencias o discrepancias entre ellos.

La Figura 4.19 presenta otra gráfica de barras, esta vez centrada en las métricas de comparación. En ella se muestran, para cada ranking individual, los valores obtenidos al compararlo con el ranking de consenso según distintas métricas (Kendall, Spearman, coeficiente WS, etc.).

Además, se genera un mapa de calor por cada métrica calculada (como se ve en la Figura 4.20, por ejemplo), en el que se visualiza la similitud o disimilitud entre todos los rankings entre sí, incluyendo tanto los individuales como el consenso.

Finalmente, se incluye una representación MDS (Multidimensional Scaling), también una por cada métrica utilizada, que proyecta en el plano las distancias relativas entre rankings según dicha métrica. Estas visualizaciones, como se ilustra en la Figura 4.21, permiten interpretar de forma intuitiva cómo se agrupan o separan los rankings en función de sus diferencias.

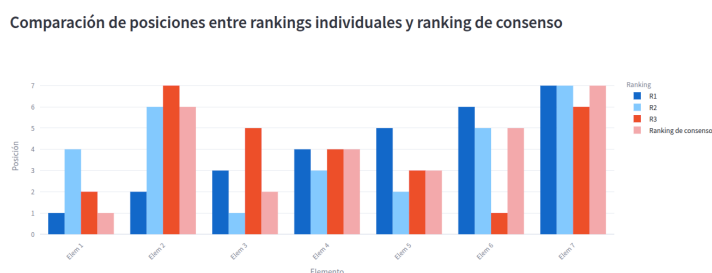


Figura 4.18: Gráfico de barras con las posiciones.

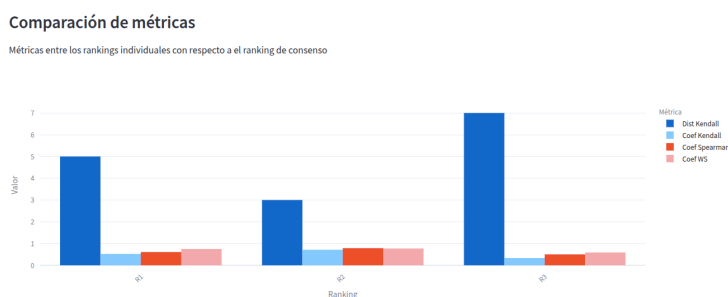


Figura 4.19: Gráfico de barras con las métricas.

Mapas de calor de distancias entre todos los rankings

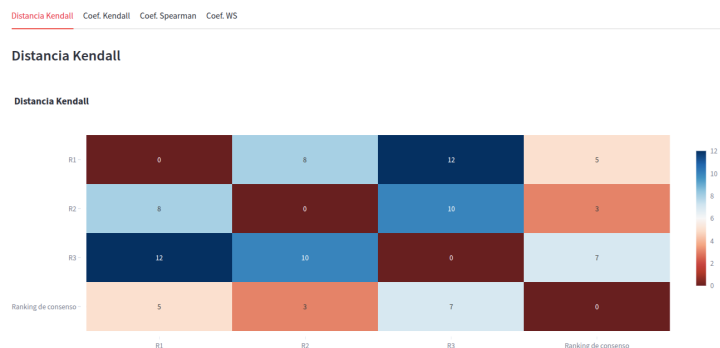


Figura 4.20: Mapas de calor con métricas entre todos los rankings.

Representación MDS de rankings

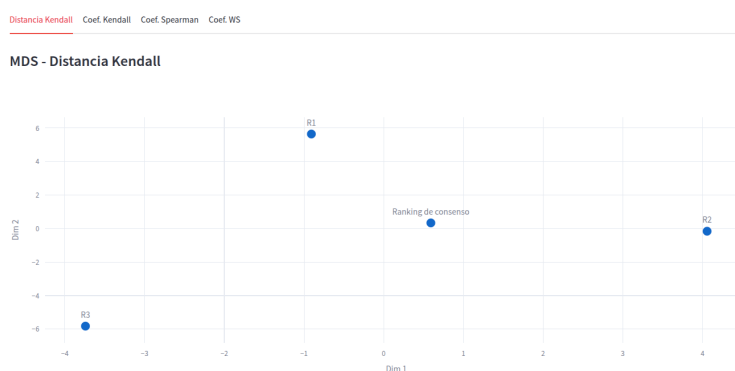


Figura 4.21: MDS entre todos los rankings con las distintas métricas.

4.1.2. Instalación

La herramienta ha sido implementada en **Python**, utilizando el framework **Streamlit** para el desarrollo de la interfaz gráfica. Esto permite que la aplicación pueda ser ejecutada de manera local como una página web interactiva.

Para instalar y ejecutar la herramienta de manera local se deben seguir los siguientes pasos:

1. Clonar el repositorio de GitHub:

```
git clone https://github.com/carmenqr/TFG.git
```

2. Acceder al directorio del proyecto:

```
cd TFG
```

3. Instalar las dependencias necesarias:

```
pip install -r requirements.txt
```

4. Ejecutar la aplicación:

```
streamlit run main.py
```

Una vez realizado esto, el navegador abrirá automáticamente la interfaz de la aplicación en la dirección `localhost:8501`.

4.2. Tecnologías utilizadas

Como se ha explicado previamente, el desarrollo de la herramienta se ha realizado utilizando **Python** junto con diversas librerías especializadas. Además, se han empleado otras tecnologías complementarias, como **Git** para el control de versiones y **SQLite** para el almacenamiento y gestión de los datos.

Python y librerías principales

La herramienta ha sido desarrollada íntegramente en **Python**, debido a su versatilidad, facilidad de uso y su amplio ecosistema de librerías. Dentro de las librerías utilizadas, destaca especialmente **Streamlit**, que permite construir de forma sencilla interfaces web interactivas orientadas a la visualización de datos y ejecución de algoritmos.

Para la implementación de los algoritmos de agregación de rankings se ha utilizado el paquete **pyRankMCDA** [12]. Este paquete proporciona implementaciones clásicas de métodos como Borda, Copeland, Kemeny-Young, Schulze y Footrule. Cabe destacar que algunas variantes específicas de estos algoritmos, como versiones con ponderaciones o soporte de empates, han sido programadas manualmente en el archivo `utils.py`, y no forman parte de las funciones originales de **pyRankMCDA**.

Además, se han empleado librerías adicionales como **Pandas** y **NumPy** para la gestión de datos y operaciones numéricas, **Matplotlib** y **Seaborn**

para la generación de gráficos, y **scikit-learn** para la visualización de distancias mediante técnicas de reducción de dimensionalidad.

Por último, para la creación de gráficos interactivos —como mapas de calor, gráficas de radar o diagramas de barras dinámicos— se ha utilizado la librería **Plotly**, que permite una visualización más flexible y manipulable por parte del usuario desde el navegador.

Git y GitHub

Durante el desarrollo del proyecto se ha utilizado **Git** como sistema de control de versiones, permitiendo gestionar los cambios en el código de manera organizada y eficiente. El repositorio del proyecto se encuentra disponible de manera pública en la plataforma **GitHub** en la siguiente dirección: <https://github.com/carmenqr/TFG>.

Visual Studio Code

El entorno de desarrollo empleado ha sido **Visual Studio Code**, un editor de código ligero, multiplataforma y ampliamente extensible. Gracias a la integración de extensiones como Python, GitLens o Jupyter, se ha facilitado la escritura, prueba y depuración del código durante todo el proyecto.

SQLite y DB Browser for SQLite

Como sistema de almacenamiento de datos se ha utilizado **SQLite**, una base de datos relacional ligera que permite gestionar de manera sencilla los rankings y los resultados de agregación generados por la herramienta. Para la gestión y exploración de la base de datos durante el desarrollo se ha empleado la herramienta gráfica **DB Browser for SQLite**, que facilita la visualización y edición de las tablas de forma intuitiva.

Jira

Para la planificación de tareas, gestión de objetivos y control del tiempo durante el desarrollo del proyecto, se ha utilizado la herramienta **Jira**. Gracias a su funcionalidad basada en tableros Kanban y la división en tareas y subtareas, ha sido posible organizar de forma estructurada el avance del proyecto, garantizando así una correcta distribución del trabajo y el cumplimiento de los plazos establecidos.

Capítulo 5

Caso de estudio

En este capítulo se presenta un caso de estudio real que permite probar el funcionamiento de los algoritmos de agregación de rankings mediante la herramienta desarrollada. El objetivo es aplicar estos algoritmos a un conjunto de rankings contruidos a partir de datos reales, relacionados con el grado de cumplimiento de los Objetivos de Desarrollo Sostenible (ODS) por parte de diferentes países.

Para ello, se parte de una base de datos que recoge, para una selección de veinte países europeos, su nivel de cumplimiento en cada uno de los ODS. A partir de estos valores se generan rankings individuales por criterio, es decir, un ranking por cada ODS, en el que los países se ordenan en función de su grado de cumplimiento. Estos rankings son tratados como entrada del sistema, y se agregan mediante los algoritmos implementados, con el fin de obtener un ranking final de consenso.

El capítulo se organiza de la siguiente forma: en primer lugar, se presenta el problema de decisión abordado y el conjunto de datos disponibles. A continuación, se describe cómo se utiliza la herramienta desarrollada para aplicar los algoritmos de agregación al problema planteado. Posteriormente, se analizan los resultados obtenidos y las diferencias entre los métodos aplicados. Finalmente, se exponen las conclusiones del caso de estudio.

5.1. Planteamiento del problema: Cumplimiento de los Objetivos de Desarrollo Sostenible (ODS)

Los Objetivos de Desarrollo Sostenible (ODS) son una iniciativa impulsada por las Naciones Unidas que forma parte de la Agenda 2030 para el Desarrollo Sostenible. Se trata de un conjunto de 17 objetivos globales que abordan los principales desafíos sociales, económicos y medioambientales del

planeta. Entre ellos se incluyen metas como erradicar la pobreza, garantizar una educación de calidad, promover el crecimiento económico inclusivo o combatir el cambio climático [13].

Para este caso de estudio se utiliza una base de datos que recoge el grado de cumplimiento de los Objetivos de Desarrollo Sostenible (ODS) por parte de veinte países europeos. Aunque la agenda 2030 incluye un total de 17 ODS, el conjunto de datos empleado contiene información únicamente sobre 6 de ellos. Esta selección no solo se debe a la disponibilidad de datos en el fichero, sino que además resulta adecuada para mantener un caso de estudio manejable en extensión y complejidad. Esta información se encuentra estructurada en un fichero Excel compuesto por tres hojas:

- **criteria**: contiene los nombres de los objetivos (criterios), junto con su rango de valores y un campo de “importancia” que podría usarse para ponderar los criterios.
- **alternatives**: recoge la lista de países (alternativas).
- **evaluations**: incluye, para cada país, su puntuación numérica (entre 1 y 100) en cada uno de los ODS.

En el contexto del problema de agregación de rankings, los países se consideran las **alternativas** a ordenar, mientras que los ODS actúan como **criterios de evaluación**. Por tanto, cada ODS genera un **ranking individual**, en el que los países se ordenan según su puntuación en ese objetivo.

Los valores de la hoja **evaluations** representan el grado de cumplimiento de cada objetivo por país, y se encuentran normalizados en una escala numérica de $[1, 100]$. Dado que todos los objetivos deben ser **maximizados** (según se especifica en la hoja **criteria**), el procedimiento para construir los rankings consiste en ordenar los países de mayor a menor puntuación para cada ODS.

Por tanto, los rankings asociados a nuestro problema quedarían de la siguiente manera:

País	SDG1	SDG2	SDG3	SDG4	SDG6	SDG8
BEL	3	18	10	10	13	5
BGR	19	16	20	18	18	18
DEU	8	9	13	15	4	3
DNK	1	4	3	1	3	4
ESP	15	14	5	8	6	15
EST	16	15	16	3	14	14
EUU	12	11	11	13	12	9
FIN	2	17	9	6	7	2
FRA	9	5	7	12	10	12
GBR	13	20	12	7	1	7
GRC	17	6	14	16	8	19
HUN	11	8	18	17	5	11
IRL	5	1	4	2	17	8
ITA	14	12	6	14	15	16
NLD	4	19	2	5	9	1
POL	6	10	17	9	11	10
PRT	10	7	15	11	19	13
ROU	18	3	19	20	20	17
SWE	7	2	1	4	2	6
TUR	20	13	8	19	16	20

Cuadro 5.1: Rankings individuales por ODS para cada país (1 = mejor cumplimiento).

Donde cada columna representa un **criterio**, es decir, un Objetivo de Desarrollo Sostenible. La siguiente tabla recoge la correspondencia entre el código del ODS y su descripción completa:

Código	Objetivo de Desarrollo Sostenible
SDG1	Fin de la pobreza
SDG2	Hambre cero
SDG3	Salud y bienestar
SDG4	Educación de calidad
SDG6	Agua limpia y saneamiento
SDG8	Trabajo decente y crecimiento económico

Cuadro 5.2: Correspondencia entre código y descripción de los ODS utilizados como criterios.

Y cada fila de la Tabla 5.1 representa una **alternativa**, es decir, un país evaluado. La correspondencia entre el código de país (utilizado en los datos)

y su nombre completo se muestra a continuación:

Código	País
BEL	Bélgica
BGR	Bulgaria
DEU	Alemania
DNK	Dinamarca
ESP	España
EST	Estonia
EUU	Unión Europea
FIN	Finlandia
FRA	Francia
GBR	Reino Unido
GRC	Grecia
HUN	Hungría
IRL	Irlanda
ITA	Italia
NLD	Países Bajos
POL	Polonia
PRT	Portugal
ROU	Rumanía
SWE	Suecia
TUR	Turquía

Cuadro 5.3: Correspondencia entre códigos de país y su denominación.

De esta forma, nuestro problema se puede formular como un problema de agregación de rankings, donde cada ODS genera una ordenación distinta de los países. El objetivo es combinar esa información parcial y a veces contradictoria en un único ranking de consenso que permita identificar qué países, en términos generales, están cumpliendo mejor con los Objetivos de Desarrollo Sostenible.

5.2. Resolución del problema

Una vez definidos los rankings individuales por criterio, el siguiente paso consiste en aplicar algoritmos de agregación para obtener un ranking de consenso que combine la información procedente de todos los objetivos. Para ello, se utilizará la herramienta desarrollada específicamente en este trabajo, la cual permite cargar rankings de entrada, seleccionar diferentes métodos de agregación y analizar los resultados obtenidos.

A continuación, se muestra paso a paso cómo resolver el problema de

agregación de rankings utilizando dicha herramienta.

1. Introducir los datos

El primer paso para resolver un problema de agregación de rankings consiste en cargar los datos de entrada, es decir, los rankings individuales generados para cada criterio. Para ello, la herramienta permite importar un archivo Excel con el formato adecuado, que será procesado automáticamente, desde el paso *1º) Nuevo problema* en el menú lateral.

Una vez cargado el archivo, toda la información relacionada con el problema —incluyendo los rankings, los elementos y sus posiciones— se almacena de forma estructurada en una base de datos interna, lo que permite su reutilización y consulta en los pasos posteriores.


Añadir Rankings desde Excel

Añada el grupo de rankings que contiene su problema

Cargue un archivo Excel (.xlsx)

Drag and drop file here
Limit 200MB per file • XLSX

[Browse files](#)

 rankings_ods_paises.xlsx 13.0KB ×

Vista previa del Excel:

	0	1	2	3	4	5	6
0	ODS	None	None	None	None	None	None
1	None	SDG1	SDG2	SDG3	SDG4	SDG6	SDG8
2	BEL	3	18	10	10	13	5
3	BGR	19	16	20	18	18	18
4	DEU	8	9	13	15	4	3
5	DNK	1	4	3	1	3	4
6	ESP	15	14	5	8	6	15
7	EST	16	15	16	3	14	14
8	EUU	12	11	11	13	12	9
9	FIN	2	17	9	6	7	2

[Guardar](#)

Figura 5.1: Introducción de los datos del problema.

2. Resolver problema

Accediendo a la opción *2º) Resolver problema* desde el menú lateral, es posible seleccionar el conjunto de rankings previamente cargado y aplicar uno de los algoritmos de agregación disponibles. La herramienta permite escoger entre distintos métodos clásicos como Borda, Copeland, Kemeny-Young, Schulze o Footrule, entre otros.

En este caso, el problema será resuelto utilizando todas las opciones disponibles con el objetivo de analizar y comparar los resultados obtenidos por cada algoritmo, a excepción del método de *Kemeny-Young*. Tal y como se ha explicado a lo largo del trabajo, este algoritmo presenta una complejidad

computacional elevada y, en problemas con un número considerable de elementos, no existe un procedimiento eficiente que permita obtener la solución exacta en un tiempo razonable.

Figura 5.2: Resolución del problema con distintos algoritmos.

5.3. Análisis de resultados

A través de la tercera pestaña del menú lateral, *3º) Ver resultados*, es posible visualizar de forma gráfica e intuitiva los resultados obtenidos en la resolución del problema. La herramienta ofrece dos modos de análisis, que se describen a continuación.

Resultado del problema general

Este modo permite comparar de manera conjunta los resultados obtenidos al aplicar todos los algoritmos de agregación sobre el mismo conjunto de rankings.

La Figura 5.3 muestra un mapa de calor con las posiciones finales de los rankings de consenso obtenidos mediante los distintos algoritmos de agregación aplicados al problema. Cada fila representa un elemento (en este caso, países), y cada columna corresponde a un método de agregación (Borda, Copeland, Schulze y Footrule).

La intensidad del color refleja la posición asignada a cada elemento en el ranking de consenso generado por cada algoritmo. Los tonos más oscuros hacia el rojo indican posiciones más bajas (es decir, peor puntuación), mientras que los tonos azulados representan posiciones más altas en el ranking (mejor valoración).

Esta visualización permite identificar rápidamente discrepancias o coincidencias entre los algoritmos. Por ejemplo, se pueden observar elementos que mantienen posiciones similares independientemente del método utilizado, así como otros que experimentan variaciones significativas según el algo-

ritmo, lo cual es indicativo de la sensibilidad del resultado final al método de agregación seleccionado.

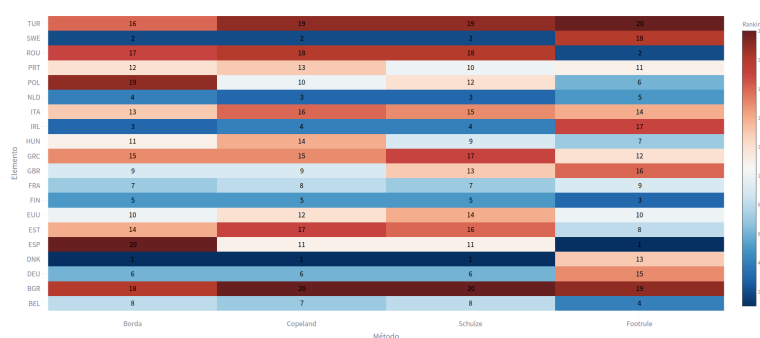


Figura 5.3: Mapa de calor con las posiciones de los rankings de consensos.

La Figura 5.4 muestra una gráfica de radar que representa las posiciones de los elementos en los distintos rankings de consenso generados por los algoritmos aplicados. Cada línea coloreada corresponde a un algoritmo de agregación distinto, y la forma que adopta cada polígono permite visualizar de manera intuitiva cómo se distribuyen las posiciones de los elementos para cada método.

Gracias al trazado sobre el plano polar, es posible identificar visualmente la similitud general entre los rankings de consenso: cuanto más se superponen las áreas de color, más parecidos son los resultados obtenidos por los algoritmos; en cambio, las zonas en las que se observa mayor divergencia indican diferencias significativas en las posiciones asignadas.

Al tratarse de una gráfica interactiva, el usuario puede activar o desactivar los algoritmos desde la leyenda, facilitando así comparaciones específicas de dos en dos o en cualquier combinación deseada. De esta forma, comprobamos que los rankings agregados que mayor similitud muestran son los obtenidos mediante los algoritmos de Copeland y Schulze, como se muestra en la Figura 5.5.

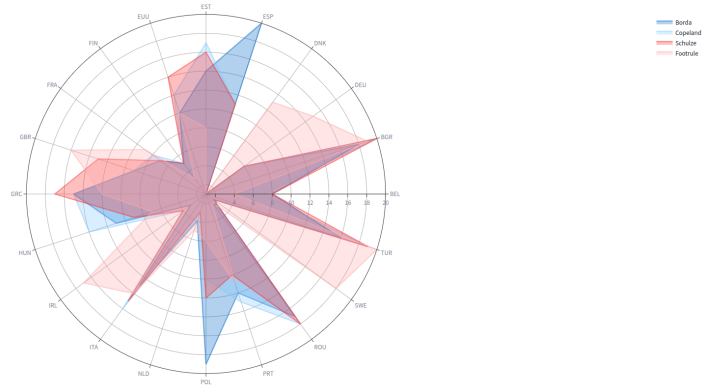


Figura 5.4: Radar para ver la similitud entre los rankings de consenso.

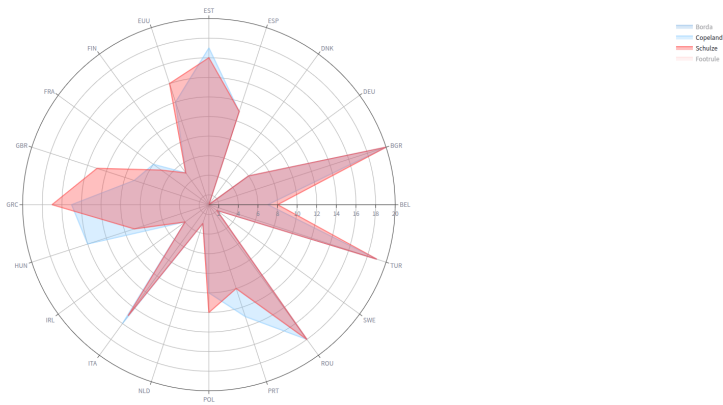


Figura 5.5: Radar comparativo entre los rankings de consenso obtenidos con los métodos de Copeland y Schulze

Resultado por tipo de solución ejecutada

En esta vista, se analiza de forma individual el resultado de cada algoritmo aplicado al problema. Se comparan los rankings individuales originales con el ranking de consenso obtenido mediante el método seleccionado, permitiendo así evaluar el grado de concordancia entre ellos a través de métricas específicas y representaciones visuales detalladas.

Método de Borda

La Tabla 5.4 recoge los resultados de varias métricas utilizadas para comparar los rankings individuales con el ranking de consenso generado

mediante el método de Borda. En ella se observa que las métricas no son homogéneas para todos los rankings, lo que indica diferencias en el grado de similitud que cada ranking individual presenta respecto al consenso.

Por ejemplo, el ranking del objetivo SDG2 muestra una *distancia de Kendall* notablemente más alta (83), junto con los coeficientes de Kendall (0,126) y Spearman (0,203), que son los más bajos del conjunto, lo que evidencia una baja concordancia general con el ranking de consenso. Sin embargo, el coeficiente WS en este caso alcanza un valor relativamente alto (0,813) en comparación con el resto de métricas. Esto se debe a que el WS penaliza menos las diferencias en las posiciones intermedias o bajas, y da más peso a las coincidencias en los primeros puestos. Por tanto, aunque SDG2 difiere bastante en la estructura general, sí coincide en parte con el consenso en los elementos mejor valorados.

En contraste, los rankings de los objetivos SDG1 y SDG8 presentan las distancias más bajas y los valores más altos en todos los coeficientes, destacando especialmente en el coeficiente WS, con valores superiores a 0,89. Como se ha mencionado previamente, este coeficiente da mayor peso a las coincidencias en las primeras posiciones del ranking, lo que sugiere que SDG1 y SDG8 comparten no solo una estructura general parecida, sino también los elementos más relevantes en posiciones altas, en línea con el consenso. En conjunto, SDG1 y SDG8 son claramente los rankings más similares al resultado generado por el método de Borda.

Estos resultados reflejan que, aunque el método de Borda tiende a producir rankings de consenso razonablemente coherentes, existen discrepancias importantes con algunos rankings individuales, lo que justifica la necesidad de aplicar varias métricas de comparación para realizar un análisis más completo.

Cuadro 5.4: Métricas de comparación entre los rankings individuales y el ranking de consenso obtenido mediante el método de Borda.

Métrica	SDG1	SDG2	SDG3	SDG4	SDG6	SDG8
Distancia Kendall	37	83	49.00	51	65	35
Coef. Kendall	0.611	0.126	0.484	0.463	0.316	0.632
Coef. Spearman	0.762	0.203	0.623	0.585	0.442	0.802
Coef. WS	0.899	0.813	0.885	0.887	0.734	0.831

Toda esta información numérica queda representada de forma visual en la Figura 5.6, donde se muestran comparativamente los valores de cada métrica para los distintos rankings individuales respecto al ranking de consenso. Esta representación gráfica facilita la identificación de patrones, permitiendo observar rápidamente qué objetivos presentan mayor o menor concordancia

según cada métrica evaluada.

Comparación de métricas

Métricas entre los rankings individuales con respecto a el ranking de consenso

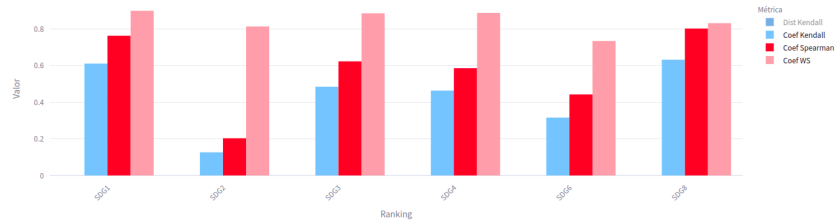


Figura 5.6: Comparación visual de las métricas entre los rankings individuales y el ranking de consenso - método de Borda.

A continuación, se presentan los mapas de calor generados a partir de la matriz de distancias entre los distintos rankings individuales (objetivos SDG) y el ranking de consenso, calculadas según cuatro métricas diferentes: distancia de Kendall, coeficiente de Kendall, coeficiente de Spearman y coeficiente WS. Estas visualizaciones permiten observar el grado de similitud entre todos los rankings de forma comparativa y simultánea.

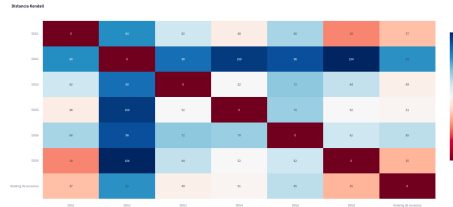


Figura 5.7: Distancia de Kendall



Figura 5.8: Coeficiente de Kendall



Figura 5.9: Coeficiente de Spearman

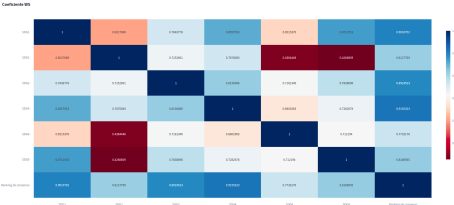


Figura 5.10: Coeficiente WS

Los cuatro mapas de calor ofrecen una visión global de cómo se relacionan entre sí todos los rankings individuales (asociados a los distintos objetivos SDG) y el ranking de consenso generado mediante el método de Borda, utilizando distintas métricas de comparación: distancia de Kendall,

coeficiente de Kendall, coeficiente de Spearman y coeficiente WS.

En el caso de las métricas basadas en coeficientes (Kendall, Spearman y WS), los valores más cercanos a 1 indican una mayor similitud entre rankings. Por el contrario, en la distancia de Kendall, valores más bajos implican rankings más parecidos. Esto permite interpretar de manera directa qué objetivos presentan estructuras de orden similares entre sí, independientemente del ranking de consenso.

En términos generales, se observa que SDG1 y SDG8 tienden a estar más alineados entre sí y con otros rankings, presentando valores elevados en los coeficientes y bajas distancias en sus comparaciones cruzadas. En el otro extremo, SDG2 se mantiene como el ranking más disonante, con correlaciones bajas e incluso negativas y distancias altas respecto al resto, lo que sugiere una lógica de orden muy diferente.

Los objetivos SDG3, SDG4 y SDG6 se sitúan en una posición intermedia: sus comparaciones con el resto muestran una variabilidad moderada, con algunas similitudes destacables y ciertas diferencias en elementos específicos.

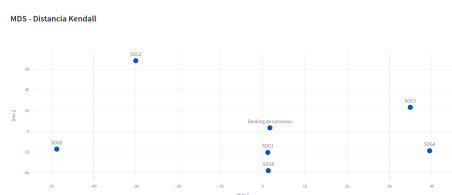


Figura 5.11: Distancia de Kendall

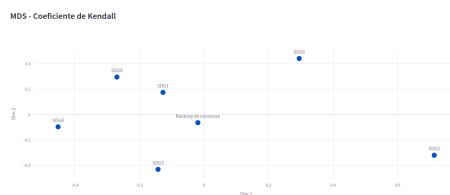


Figura 5.12: Coeficiente de Kendall

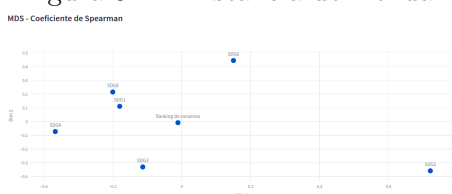


Figura 5.13: Coeficiente de Spearman

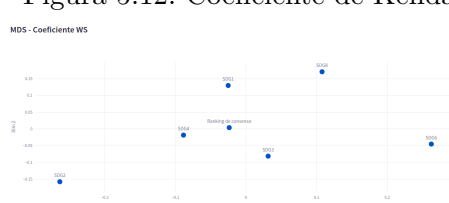


Figura 5.14: Coeficiente WS

Por último, observamos la última de las gráficas proporcionadas: las representaciones *MDS* generadas a partir de las matrices de distancias utilizadas anteriormente. Esta técnica, conocida como *Multi-Dimensional Scaling*, permite proyectar rankings en un plano bidimensional conservando, en la medida de lo posible, las relaciones de distancia reales entre ellos. Así, es posible interpretar visualmente qué rankings están más próximos o alejados entre sí.

A partir de las matrices de distancias empleadas para los mapas de calor,

se han generado estas visualizaciones, donde el ranking de consenso actúa como punto central de referencia. Cuanto más cerca esté un ranking individual de este punto, mayor es su similitud con la solución agregada. Idealmente, los rankings se distribuirían de forma equilibrada alrededor del consenso, como si este fuera el centro de un círculo.

En el caso del método de Borda, se observa una agrupación relativamente cercana entre SDG1, SDG4 y SDG8 alrededor del consenso, lo que refuerza su alineación con el resultado final. Por el contrario, SDG2 aparece sistemáticamente alejado en todas las proyecciones, lo que confirma su disonancia respecto al consenso, ya anticipada en las métricas anteriores. SDG6 y SDG3 también tienden a situarse en posiciones periféricas, indicando una relación más débil con la estructura común.

Método de Copeland

La Tabla 5.5 presenta los resultados de las métricas de comparación entre los rankings individuales y el ranking de consenso obtenido mediante el algoritmo de Copeland. Se observan diferencias notables en la similitud que cada ranking individual presenta respecto al consenso.

El ranking correspondiente al objetivo SDG2 muestra una *distancia de Kendall* elevada (85) y valores bajos en los coeficientes de Kendall (0,105) y Spearman (0,144), lo que indica una baja concordancia general. No obstante, el coeficiente WS alcanza un valor medio (0,776), lo cual sugiere cierta alineación en las primeras posiciones del ranking, ya que esta métrica penaliza menos las diferencias en posiciones bajas y otorga mayor peso a las coincidencias iniciales.

En el otro extremo, los rankings SDG1 y SDG8 presentan distancias de Kendall significativamente más bajas (25 y 27, respectivamente), y obtienen valores elevados en todos los coeficientes. En particular, destacan en el coeficiente WS (0,914 y 0,860), lo que indica una alta similitud en las posiciones más relevantes.

Estos resultados reflejan que el método de Copeland es capaz de generar un ranking de consenso con una estructura consistente respecto a algunos de los rankings individuales, aunque con ciertas discrepancias en otros casos.

Cuadro 5.5: Métricas de comparación entre los rankings individuales y el ranking de consenso obtenido mediante el método de Copeland.

Métrica	SDG1	SDG2	SDG3	SDG4	SDG6	SDG8
Distancia Kendall	25.00	85.00	43.00	39.00	53.00	27.00
Coef. Kendall	0.737	0.105	0.547	0.589	0.442	0.716
Coef. Spearman	0.899	0.144	0.684	0.732	0.594	0.886
Coef. WS	0.914	0.776	0.916	0.853	0.738	0.860

Toda esta información numérica queda representada de forma visual en la Figura 5.15, donde se muestran comparativamente los valores de cada métrica para los distintos rankings individuales respecto al ranking de consenso. Esta representación gráfica facilita la identificación de patrones, permitiendo observar rápidamente qué objetivos presentan mayor o menor concordancia según cada métrica evaluada.

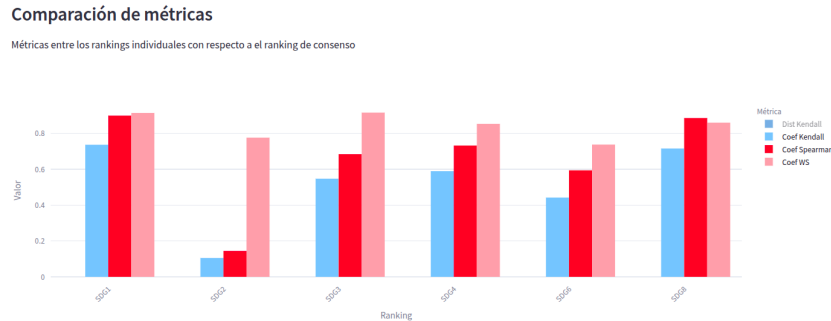


Figura 5.15: Comparación visual de las métricas entre los rankings individuales y el ranking de consenso - método de Copeland.

Los mapas de calor correspondientes al algoritmo de Copeland permiten analizar visualmente el grado de similitud entre todos los rankings individuales generados para los distintos objetivos SDG y el ranking de consenso. Las cuatro métricas utilizadas —distancia de Kendall, coeficiente de Kendall, coeficiente de Spearman y coeficiente WS— permiten captar tanto similitudes globales como coincidencias en posiciones destacadas.

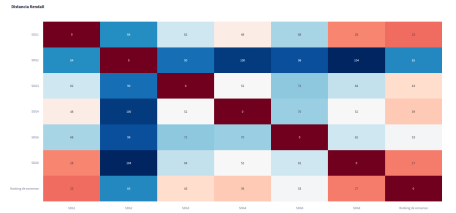


Figura 5.16: Distancia de Kendall



Figura 5.17: Coeficiente de Kendall



Figura 5.18: Coeficiente de Spearman



Figura 5.19: Coeficiente WS

En este caso, se aprecia una mayor homogeneidad general entre rankings si se compara con el método anterior. En particular, los objetivos SDG1, SDG4 y SDG8 presentan valores elevados en la mayoría de los coeficientes, lo que indica que siguen patrones de orden similares y que tienden a coincidir en la priorización de los elementos más relevantes. También las distancias entre estos rankings son relativamente bajas, consolidando su rol como ejes de referencia dentro del conjunto.

SDG2, aunque sigue siendo uno de los rankings más diferentes en comparación con el resto, presenta en este caso algunas correlaciones algo menos extremas, lo que sugiere una ligera mejora en su alineación respecto al consenso y los demás rankings. No obstante, sigue destacando por su baja similitud con prácticamente todos los objetivos.

En cuanto a SDG3 y SDG6, su comportamiento es más variable. Dependiendo de la métrica utilizada, pueden aparecer como moderadamente alineados o con desviaciones notables. En general, su relación con los rankings más consistentes (como SDG1 o SDG8) es menos estable, lo que indica diferencias de criterio o prioridades que no se replican sistemáticamente.

Por último, se observan las representaciones *MDS* generadas para el algoritmo de Copeland, que permiten complementar visualmente la información obtenida en los mapas de calor y métricas.

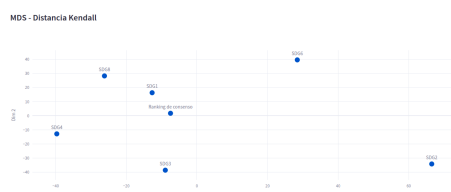


Figura 5.20: Distancia de Kendall

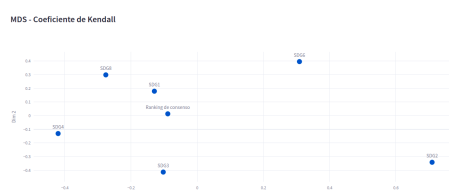


Figura 5.21: Coeficiente de Kendall

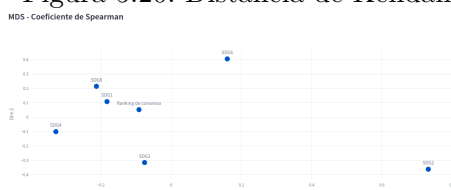


Figura 5.22: Coeficiente de Spearman



Figura 5.23: Coeficiente WS

En este caso, se aprecia una configuración bastante equilibrada, con SDG1, SDG4 y SDG8 nuevamente agrupados cerca del ranking de consenso, lo que indica una buena alineación con la solución final. SDG2 continúa alejándose sistemáticamente en todas las proyecciones, confirmando su comportamiento atípico y más distante. SDG6 y SDG3 vuelven a aparecer más alejados en comparación con los objetivos centrales, aunque su distancia no es tan pronunciada como en otros métodos, lo que sugiere una ligera mejora de su integración en el consenso.

Método de Schulze

La Tabla 5.6 muestra los resultados obtenidos al comparar cada ranking individual con el ranking de consenso generado mediante el método de Schulze. Las métricas reflejan diferencias en el grado de alineación que presentan los rankings individuales con respecto al consenso.

El ranking correspondiente al objetivo SDG2 presenta una *distancia de Kendall* considerable (78), así como valores bajos en los coeficientes de Kendall (0,179) y Spearman (0,233), lo que sugiere una baja similitud global. A pesar de ello, el coeficiente WS alcanza un valor intermedio (0,779), lo que indica cierta coincidencia en las primeras posiciones del ranking.

Por otro lado, el ranking SDG1 presenta una excelente concordancia con el consenso, con la distancia de Kendall más baja (22) y los valores más altos en los coeficientes de Kendall (0,768), Spearman (0,901) y WS (0,905). También destaca SDG8, que obtiene métricas elevadas en todos los casos, especialmente en el coeficiente WS (0,855), evidenciando una estructura muy

similar al ranking de consenso en sus posiciones más relevantes.

Estos resultados refuerzan la utilidad del método de Schulze para generar soluciones consistentes, manteniendo una buena proximidad con ciertos rankings individuales, aunque otros presenten mayores divergencias.

Cuadro 5.6: Métricas de comparación entre los rankings individuales y el ranking de consenso obtenido mediante el método de Schulze.

Métrica	SDG1	SDG2	SDG3	SDG4	SDG6	SDG8
Distancia Kendall	22.00	78.00	48.00	46.00	56.00	32.00
Coef. Kendall	0.768	0.179	0.495	0.516	0.411	0.663
Coef. Spearman	0.901	0.233	0.651	0.683	0.531	0.845
Coef. WS	0.905	0.779	0.918	0.858	0.642	0.855

Toda esta información numérica queda representada de forma visual en la Figura 5.24, donde se muestran comparativamente los valores de cada métrica para los distintos rankings individuales respecto al ranking de consenso. Esta representación gráfica facilita la identificación de patrones, permitiendo observar rápidamente qué objetivos presentan mayor o menor concordancia según cada métrica evaluada.

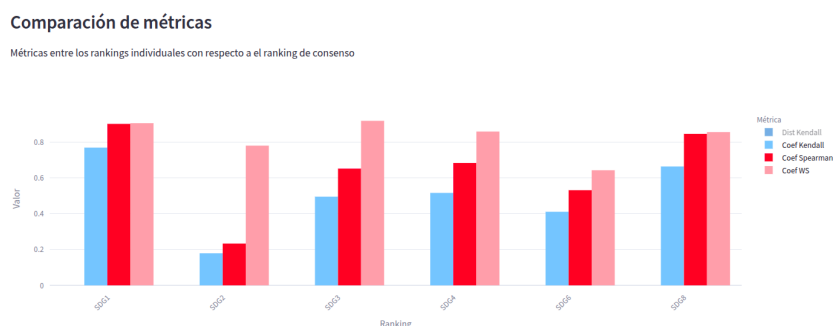


Figura 5.24: Comparación visual de las métricas entre los rankings individuales y el ranking de consenso - método de Schulze.

A partir de los mapas de calor generados, se puede observar cómo se agrupan o se diferencian los distintos objetivos SDG entre sí y respecto al consenso, teniendo en cuenta tanto el orden general de los elementos como las posiciones destacadas.

En este caso, destaca de nuevo el objetivo SDG1, que mantiene una fuerte similitud con varios rankings, en especial SDG8, con el que forma una pareja bastante consistente en términos de correlaciones elevadas y distancias reducidas. Ambos comparten prioridades similares en la mayoría de las

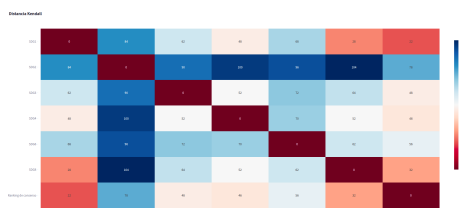


Figura 5.25: Distancia de Kendall



Figura 5.26: Coeficiente de Kendall



Figura 5.27: Coeficiente de Spearman



Figura 5.28: Coeficiente WS

métricas.

Por el contrario, SDG2 sigue presentando los valores más alejados del resto, con coeficientes bajos o incluso negativos y distancias de Kendall elevadas. Esto refuerza su carácter discordante respecto a los patrones comunes de ordenación del conjunto.

SDG4 también muestra un buen nivel de alineación con otros rankings como SDG3 y SDG8, en particular en los coeficientes WS y Spearman, lo que indica cierta coherencia en las prioridades de los elementos más relevantes. SDG3 y SDG6, por su parte, conservan un comportamiento más intermedio, aunque en el caso de SDG6 se perciben algunas mejoras en la alineación respecto a los métodos anteriores.

Por último, se muestran los MDS correspondientes al algoritmo de Schulze.

En estas proyecciones, se mantiene la tendencia observada en algoritmos anteriores: SDG1, SDG4 y SDG8 se ubican relativamente cerca del ranking de consenso, lo que confirma su coherencia con la solución agregada. En cambio, SDG2 vuelve a aparecer alejado del centro en todas las métricas, destacando como el objetivo más discordante. SDG6 y SDG3 se sitúan en posiciones más periféricas, aunque ligeramente más próximas en algunas métricas que en otros métodos, lo que sugiere una mejora parcial de su alineación con el consenso.

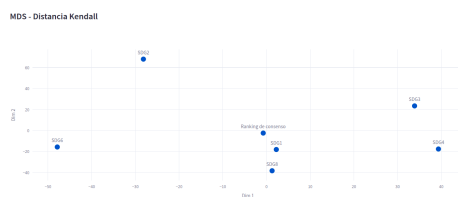


Figura 5.29: Distancia de Kendall



Figura 5.30: Coeficiente de Kendall

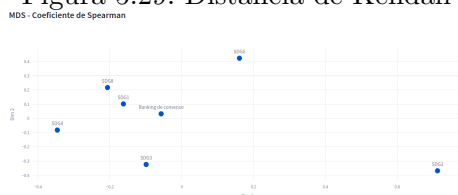


Figura 5.31: Coeficiente de Spearman

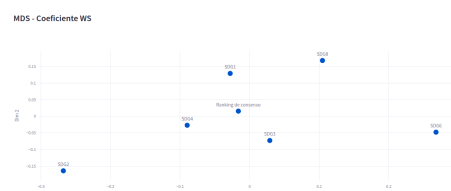


Figura 5.32: Coeficiente WS

Método de Footrule

La Tabla 5.7 recoge las métricas obtenidas al comparar los rankings individuales con el ranking de consenso generado mediante el método de Footrule. En este caso, se observan diferencias significativas respecto a los métodos anteriores, reflejando una menor concordancia general.

Los resultados muestran que todos los rankings individuales presentan *distancias de Kendall* elevadas, destacando especialmente los valores correspondientes a SDG2 (109), SDG3 (107) y SDG6 (95), lo que sugiere una fuerte discrepancia con el ranking de consenso. Asimismo, los coeficientes de Kendall y Spearman para estos objetivos son muy bajos o incluso negativos, indicando una inversión parcial del orden entre rankings individuales y el consenso.

En cuanto al coeficiente WS, los valores también son significativamente más bajos en comparación con los métodos anteriores. Sin embargo, el ranking SDG8 muestra un resultado algo más favorable, con un valor de WS de 0,733, lo que sugiere una coincidencia relativa en las posiciones iniciales, aunque el resto de métricas no son especialmente altas.

En conjunto, los resultados obtenidos con el método de Footrule reflejan una mayor desviación del ranking de consenso respecto a los rankings individuales, lo que podría deberse a su criterio de agregación más centrado en minimizar la distancia absoluta acumulada sin considerar tanto la estructura relativa entre posiciones.

Cuadro 5.7: Métricas de comparación entre los rankings individuales y el ranking de consenso obtenido mediante el método de Footrule.

Métrica	SDG1	SDG2	SDG3	SDG4	SDG6	SDG8
Distancia Kendall	77	109	107	87	95	83
Coef. Kendall	0.189	-0.147	-0.126	0.084	0.000	0.126
Coef. Spearman	0.224	-0.203	-0.131	0.072	-0.002	0.168
Coef. WS	0.625	0.295	0.362	0.372	0.252	0.733

Toda esta información numérica queda representada de forma visual en la Figura 5.33, donde se muestran comparativamente los valores de cada métrica para los distintos rankings individuales respecto al ranking de consenso. Esta representación gráfica facilita la identificación de patrones, permitiendo observar rápidamente qué objetivos presentan mayor o menor concordancia según cada métrica evaluada.



Figura 5.33: Comparación visual de las métricas entre los rankings individuales y el ranking de consenso - método de Footrule.

El método de Footrule muestra un comportamiento visiblemente distinto respecto a los algoritmos anteriores, tanto en los patrones de similitud como en la dispersión de los valores obtenidos en las métricas. Esto se puede ver claramente en los mapas de calor, donde los coeficientes tienden a ser más bajos y las distancias de Kendall mucho más elevadas en general.

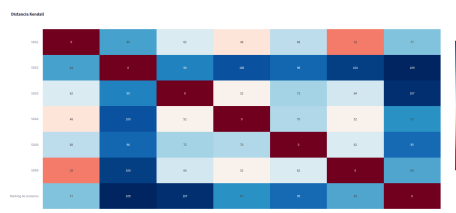


Figura 5.34: Distancia de Kendall



Figura 5.35: Coeficiente de Kendall



Figura 5.36: Coeficiente de Spearman



Figura 5.37: Coeficiente WS

En este caso, la coherencia entre rankings individuales se reduce notablemente. Salvo SDG1 y SDG8, que siguen manteniendo ciertas correlaciones entre ellos y con el consenso, el resto de objetivos muestran relaciones débiles o incluso negativas, en especial SDG2 y SDG3. Estos dos objetivos destacan por presentar valores anómalos en varias métricas: coeficientes de Kendall y Spearman negativos o muy próximos a cero, junto con distancias altas respecto al resto.

También se observa una mayor dispersión global, sin bloques de similitud clara como en otros métodos. La lógica de agregación de Footrule, basada en minimizar la suma de diferencias absolutas entre posiciones, parece introducir mayor sensibilidad a los movimientos en el orden general, lo que hace que los rankings individuales se parezcan menos entre sí y estén más dispersos en cuanto a su orden.

En cuanto al coeficiente WS, aunque suaviza en parte las discrepancias y presenta valores más moderados, sigue evidenciando que la alineación entre rankings individuales es más débil. Este comportamiento sugiere que Footrule puede no ser tan adecuado cuando se busca un consenso estable entre rankings que son muy diferentes entre sí.

Por último, se muestran los MDS correspondientes al algoritmo de Footrule.

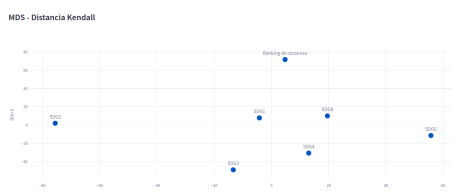


Figura 5.38: Distancia de Kendall

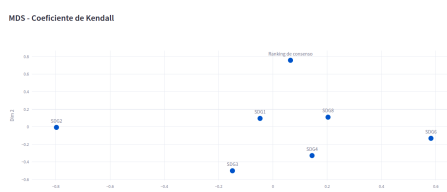


Figura 5.39: Coeficiente de Kendall

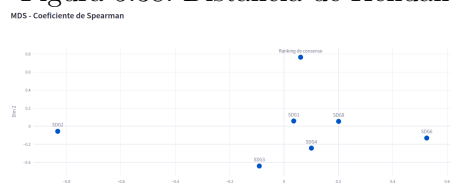


Figura 5.40: Coeficiente de Spearman

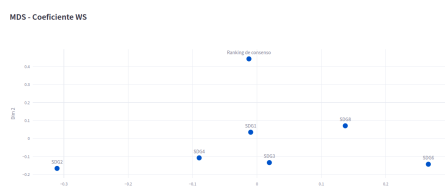


Figura 5.41: Coeficiente WS

En este caso, los rankings individuales aparecen más dispersos y alejados del consenso, lo que refuerza la conclusión de que Footrule genera soluciones menos alineadas globalmente. Aunque SDG1 y SDG8 siguen estando relativamente próximos al consenso, no se forma una estructura tan compacta como en otros métodos. SDG2 vuelve a ser el objetivo más alejado en todas las proyecciones, manteniéndose como el ranking menos concordante.

Además, rankings como SDG3, SDG4 y SDG6 también se sitúan lejos del núcleo, lo que evidencia una mayor divergencia general entre rankings. Este resultado coincide con lo observado en las métricas previas y en los mapas de calor: Footrule tiende a producir consensos que se distancian más del comportamiento medio del conjunto.

5.4. Conclusiones

A lo largo de este caso de estudio se ha evaluado la aplicación práctica de diferentes algoritmos de agregación de rankings sobre un conjunto de datos reales vinculados al cumplimiento de los Objetivos de Desarrollo Sostenible (ODS) 5.2 por parte de países europeos 5.3. Los resultados obtenidos permiten extraer varias conclusiones relevantes tanto sobre el comportamiento de los métodos como sobre las implicaciones de los rankings generados.

En primer lugar, el análisis revela una clara disparidad entre los rankings individuales por objetivo (SDG), siendo **SDG1 (Fin de la pobreza)** y **SDG8 (Trabajo decente y crecimiento económico)** los que presentan una mayor concordancia global con los rankings de consenso. Esto sugiere

que los países que destacan en estos objetivos lo hacen de manera consistente en varios algoritmos, lo que podría reflejar una sólida base estructural en sus políticas socioeconómicas. En cambio, **SDG2 (Hambre cero)** aparece sistemáticamente como el objetivo más disonante, con posiciones muy alejadas del consenso. Esto indica que su comportamiento difiere notablemente del resto de objetivos y que las prioridades de los países en este ámbito no siguen un patrón común. Además, esto también implica que los países que suelen situarse en posiciones altas en otros objetivos no lo hacen necesariamente en SDG2. Es decir, el orden de los países en este objetivo no guarda coherencia con el de los rankings restantes, lo que convierte a SDG2 en el objetivo más inestable a la hora de identificar a los países que mejor lo cumplen.

En cuanto a las **alternativas** —los países—, aquellos que se sitúan con frecuencia en posiciones altas en varios rankings de consenso (como Dinamarca, Países Bajos o Finlandia) pueden considerarse referentes de cumplimiento equilibrado de los ODS analizados. Por otro lado, países como Rumanía, Turquía o Bulgaria tienden a ocupar posiciones más bajas de forma persistente, independientemente del algoritmo utilizado, lo que pone de relieve brechas de desarrollo más acentuadas en ciertos ámbitos.

En lo que respecta a los **métodos de agregación**, el estudio comparativo permite valorar sus características prácticas y sus efectos sobre el resultado:

- **Borda** ha demostrado ser un método estable y razonablemente representativo del conjunto, con resultados equilibrados y patrones de similitud clara entre varios rankings. Su comportamiento es especialmente coherente con los rankings de SDG1, SDG4 y SDG8.
- **Copeland** ha ofrecido también un buen nivel de concordancia general, generando consensos compactos. Tiende a favorecer rankings que mantienen mayor cantidad de coincidencias parciales entre pares de elementos, lo que lo hace adecuado para casos donde se desea minimizar contradicciones en comparaciones directas.
- **Schulze** ha destacado por su capacidad para generar consensos que mantienen la coherencia con varios rankings individuales clave. Aunque es más complejo, ha mostrado un rendimiento intermedio entre Borda y Copeland, con una notable afinidad por SDG1 y SDG8, y cierta sensibilidad a los desalineamientos como en SDG2.
- **Footrule**, por el contrario, ha sido el algoritmo que ha mostrado mayor dispersión y menor alineación con los rankings individuales. Aunque conceptualmente busca minimizar la distancia absoluta total, sus consensos tienden a alejarse más del comportamiento global, lo que lo

convierte en una opción menos robusta para casos con rankings muy heterogéneos.

Desde el punto de vista metodológico, este caso confirma la importancia de utilizar múltiples métricas (Kendall, Spearman, WS) y representaciones visuales (mapas de calor, MDS, gráficas de barras y radar) para evaluar correctamente el grado de alineación entre rankings. No todos los algoritmos producen soluciones igual de interpretables ni con el mismo nivel de coherencia.

Finalmente, cabe destacar que la herramienta desarrollada ha permitido analizar el problema con una perspectiva integral, facilitando tanto el cálculo de rankings como su visualización e interpretación. El sistema ha demostrado ser útil para tomar decisiones basadas en múltiples criterios, y especialmente válido para comparar el comportamiento real de los algoritmos en situaciones complejas y realistas.

Como conclusión final, los rankings de consenso obtenidos para los países analizados han quedado de la siguiente manera según el algoritmo de agregación aplicado:

Cuadro 5.8: Ranking de los países según cada algoritmo de agregación.

Puesto	Borda	Copeland	Schulze	Footrule
1	DNK	DNK	DNK	ESP
2	SWE	SWE	SWE	ROU
3	IRL	NLD	NLD	FIN
4	NLD	IRL	IRL	BEL
5	FIN	FIN	FIN	NLD
6	DEU	DEU	DEU	POL
7	FRA	BEL	FRA	HUN
8	BEL	FRA	BEL	EST
9	GBR	GBR	HUN	FRA
10	EUU	POL	PRT	EUU
11	HUN	ESP	ESP	PRT
12	PRT	EUU	POL	GRC
13	ITA	PRT	GBR	DNK
14	EST	HUN	EUU	ITA
15	GRC	GRC	ITA	DEU
16	TUR	ITA	EST	GBR
17	ROU	EST	GRC	IRL
18	BGR	ROU	ROU	SWE
19	POL	TUR	TUR	BGR
20	ESP	BGR	BGR	TUR

Analizando estas posiciones en conjunto, es posible identificar a los países que, de forma consistente, se sitúan entre los mejores en el cumplimiento equilibrado de los ODS. Destacan especialmente:

- **Dinamarca (DNK), Suecia (SWE) y Finlandia (FIN)**: se mantienen de forma clara en el top 5 en al menos tres de los cuatro métodos, lo que refleja una aplicación sólida, coherente y sostenida de políticas relacionadas con los ODS.
- **Países Bajos (NLD) e Irlanda (IRL)** también muestran una tendencia a estar entre los cinco primeros puestos, reflejando una alineación equilibrada en la mayoría de los criterios evaluados.

Por el contrario, los países con un desempeño más bajo de forma consistente han sido:

- **Bulgaria (BGR) y Turquía (TUR)**: aparecen sistemáticamente en los últimos puestos, lo que indica una falta de alineación transversal con los objetivos evaluados.
- **Rumanía (ROU)** y en menor medida **Grecia (GRC)** también ocupan posiciones bajas en la mayoría de los métodos, lo que evidencia que sus fortalezas se concentran en objetivos puntuales pero no son generalizables al conjunto.

Una observación especialmente llamativa es el comportamiento del algoritmo **Footrule**, cuyas posiciones difieren significativamente del resto, sobre todo en los extremos del ranking. Por ejemplo, **España (ESP)** —última según el algoritmo de Borda— aparece en primera posición con Footrule. Del mismo modo, **Suecia (SWE)**, que ocupa el segundo puesto en tres de los algoritmos, cae hasta el antepenúltimo lugar en este caso. Esta variabilidad tan acusada no se reproduce en la parte media del ranking, donde las diferencias son menores y más estables.

Estas desviaciones pueden entenderse a partir del funcionamiento interno del algoritmo **Footrule** (descritas en la Sección 3.5), que se basa en minimizar la *suma total de distancias absolutas* entre las posiciones asignadas en el ranking de consenso y las posiciones ocupadas por cada elemento en los rankings individuales.

En particular, el método tiende a premiar a aquellos elementos que, sin ocupar siempre las primeras posiciones, se mantienen relativamente cerca de ellas de forma consistente. A su vez, puede elevar en el ranking a alternativas que presentan *resultados extremos* —por ejemplo, muy buenas posiciones en algunos rankings y muy malas en otros— siempre que su colocación final minimice la discrepancia global respecto a todas las entradas. Esto explica, por

ejemplo, por qué **España (ESP)** asciende del último lugar (según Borda) y de posiciones intermedias (según Copeland y Schuzule) al primer lugar con Footrule: aunque su rendimiento sea bajo en varios objetivos, presenta posiciones altas en otros que compensan ese desfase. Del mismo modo, **Suecia (SWE)**, a pesar de ocupar lugares altos en general, puede verse penalizada si pequeñas diferencias acumuladas generan un mayor coste absoluto total en comparación con otras asignaciones.

Estas diferencias refuerzan la importancia de seleccionar cuidadosamente el método de agregación más apropiado, en función del objetivo analítico: Footrule resulta útil cuando se busca una solución que minimice las diferencias absolutas entre la posición final de cada elemento y sus posiciones en los rankings individuales. En cambio, métodos como Borda, Copeland o Schulze tienden a resaltar aquellas alternativas con un rendimiento más consistente o estable a lo largo de los distintos criterios, sin verse tan influenciados por extremos aislados.

Este análisis final pone de manifiesto que, a pesar de las diferencias entre los algoritmos, existe un patrón común que permite detectar un grupo de países con rendimiento elevado y otro con carencias recurrentes. Así, aplicar diferentes métodos de agregación ha resultado útil para contrastar enfoques y confirmar que, en un caso de estudio de tamaño moderado y con datos estructurados, los resultados tienden a converger en torno a ciertas posiciones clave. Esta coincidencia entre métodos aporta robustez al análisis y refuerza la fiabilidad de las conclusiones extraídas, al permitir identificar consensos entre diferentes rankings sin depender exclusivamente de un único método de agregación.

Capítulo 6

Conclusiones y trabajos futuros

Este Trabajo de Fin de Grado ha tenido como propósito general el estudio e implementación de algoritmos para la agregación de rankings, abordado a través de cuatro objetivos principales:

- Estudiar las variantes del problema de agregación de rankings.
- Analizar un conjunto de algoritmos de agregación de rankings.
- Diseñar una herramienta para implementar dichos algoritmos y facilitar la resolución de problemas de agregación de rankings.
- Aplicar dicha herramienta en un caso de estudio.

Tras el desarrollo del proyecto, puede afirmarse que todos estos objetivos han sido cumplidos satisfactoriamente. Se ha llevado a cabo un análisis detallado del problema, clasificando sus diferentes variantes y evaluando diversas métricas de comparación. También se han implementado varios algoritmos clásicos y modernos, y se ha diseñado una herramienta funcional para probarlos y compararlos entre sí. Finalmente, se ha aplicado dicha herramienta en un caso práctico relacionado con los Objetivos de Desarrollo Sostenible, lo cual ha permitido estudiar el comportamiento real de los métodos desarrollados.

Desde un punto de vista personal, la realización de este trabajo ha supuesto una experiencia muy enriquecedora. A lo largo del proceso he aprendido a planificarme de manera más eficiente, a adaptarme a los tiempos marcados y a tomar decisiones sobre el enfoque del trabajo. Contar con el apoyo de un tutor me ha permitido orientar mejor el estudio y profundizar en los aspectos realmente relevantes.

Además, he disfrutado tanto del estudio teórico del problema como del desarrollo de la herramienta. Implementar los algoritmos, verlos en funcionamiento y poder analizar sus resultados de forma práctica ha sido una parte especialmente gratificante del proyecto. Superar las dificultades técnicas que han ido surgiendo durante el desarrollo ha contribuido también a mi crecimiento personal y académico. En definitiva, este trabajo me ha permitido aprender sobre un tema que me ha entusiasmado y llevar a cabo un proyecto completo, desde la investigación hasta la implementación y análisis de resultados.

Trabajos futuros

Aunque el proyecto se encuentra en un estado funcional y cumple con los objetivos planteados, como posibles líneas de continuación y mejora, se proponen las siguientes ideas para futuros desarrollos.

Una dirección lógica sería continuar el desarrollo de la herramienta con el objetivo de convertirla en un sistema más completo, no solo centrado en la agregación, sino también en la **gestión de rankings individuales**. Esto incluiría funcionalidades para la **creación y edición de rankings por parte de distintos usuarios**, la **definición de criterios personalizados** según el contexto, y la **visualización comparativa** entre rankings originales y resultados agregados.

Asimismo, podría añadirse la posibilidad de almacenar y organizar conjuntos de rankings según distintas categorías o escenarios, facilitando su uso en estudios de grupo, procesos de toma de decisiones o aplicaciones donde intervienen múltiples fuentes de ordenación.

Estas extensiones permitirían avanzar hacia una herramienta más general para la gestión de rankings y sus agregaciones, con potencial para ser aplicada en contextos académicos, colaborativos o profesionales.

Bibliografía

- [1] Fiorenzo Franceschini, Domenico A. Maisano, and Luca Mastrogiacomo. Rankings and decisions in engineering. <https://link.springer.com/book/10.1007/978-3-030-89865-6>, 2020.
- [2] Siyi Wang, Qi Deng, Shiwei Feng, Hong Zhang, and Chao Liang. A survey on rank aggregation. <https://dl.acm.org/doi/10.24963/ijcai.2024/915>, 2024.
- [3] Javier Alcaraz, Mercedes Landete, and Juan F. Monge. Rank aggregation: Models and algorithms. https://ideas.repec.org/h/spr/sprchp/978-3-030-96935-6_5.html, 2020.
- [4] F. Ricci, L. Rokach, and B. Shapira. Recommender systems handbook. <https://www.springer.com/gp/book/9780387858203>, 2011.
- [5] Cynthia Dwork, Ravi Kumar, Moni Naor, and D. Sivakumar. Rank aggregation methods for the web. <https://dl.acm.org/doi/pdf/10.1145/371920.372165>, 2001.
- [6] Alon Altman and Moshe Tennenholtz. Axiomatic foundations for ranking systems. https://www.eecs.harvard.edu/cs286r/courses/fall09/papers/altman_jair.pdf, 2008.
- [7] Wojciech Sałabun and Karol Urbaniak. A new coefficient of rankings similarity in decision-making problems. https://doi.org/10.1007/978-3-030-50417-5_47, 2020.
- [8] Wikipedia contributors. Kendall tau distance. https://en.wikipedia.org/wiki/Kendall_tau_distance, 2024.
- [9] Intellectus Consulting. Correlation (pearson, kendall, spearman). <https://www.statisticssolutions.com/free-resources/directory-of-statistical-analyses/correlation-pearson-kendall-spearman/>, 2024.

- [10] Nir Ailon. An active learning algorithm for ranking from pairwise preferences with an almost optimal query complexity. <https://www.jmlr.org/papers/volume13/ailon12a/ailon12a.pdf>, 2012.
- [11] Harold W. Kuhn. The hungarian method for the assignment problem. <https://doi.org/10.1007/BF01580087>, 1955.
- [12] Valdecy Alves. pyrankmcda: Multi-criteria decision analysis in python. <https://github.com/Valdecy/pyRankMCDA>, 2020.
- [13] Naciones Unidas. Objetivos de desarrollo sostenible (ods). <https://www.un.org/sustainabledevelopment/es/>, 2015. Consulta: junio 2025.
- [14] OpenAI. Chatgpt (versión gpt-4). <https://chat.openai.com>, 2025.

