

Trabajo TD 02

Reyes de las Aguas García Azancot

2025-10-31

Índice

1. Introducción	3
2. Definición del Problema	3
2.1. Alternativas	3
2.2. Jerarquía de Criterios y Subcriterios	3
2.2.1. Criterio I: Potencia de Cálculo (2 Subcriterios)	3
2.2.2. Criterio II: Almacenamiento y Memoria (2 Subcriterios)	3
2.2.3. Criterio III: Portabilidad (2 Subcriterios)	4
2.2.4. Criterio IV: Experiencia Visual (1 Subcriterio)	4
2.2.5. Criterio V: Economía (1 Subcriterio)	4
2.3. Matriz de Decisión Inicial	4
3. Metodología I: Proceso Analítico Jerárquico (AHP)	5
3.1. Matriz de Criterios	5
3.2. Matrices Alternativas-Criterios	6
3.3. Con ayuda de las funciones R de clase	9
3.3.1. Método 1: mayor autovalor	9
3.3.2. Método 2: Completo	12
3.4. Diagrama Jerárquico	12
3.5. Método AHP	13
3.5.1. Conclusión final	18

4. Metodología II: Método ELECTRE	18
4.1. Parámetros de Entrada y de Proceso	18
4.2. Implementación y Cálculo del Método ELECTRE	18
4.2.1. Conclusión Final	21
4.3. Implementación y Cálculo del Método ELECTRE I	22
5. Metodología III: Método PROMETHEE	23
5.1. Parámetros de Entrada	23
5.2. Cálculo de Flujos de Preferencia	23
5.2.1. Método PROMETHEE I	24
5.2.2. Método PROMETHEE II	25
5.2.3. Método PROMETHEE I (medias)	26
5.2.4. Método PROMETHEE II (medias)	28
5.2.5. Resolución con Promethee Windows	29

1. Introducción

La selección del ordenador portátil óptimo para un estudiante de Estadística es un problema de decisión compleja que requiere sopesar múltiples criterios como el rendimiento del hardware, la movilidad y el coste. El presente trabajo aborda esta necesidad mediante la aplicación y comparación de las principales metodologías de Decisión Multicriterio (DMC): el Proceso Analítico Jerárquico (AHP), ELECTRE y PROMETHEE.

2. Definición del Problema

2.1. Alternativas

Las alternativas de portátiles a evaluar son:

- Apple MacBook Pro (M-series)
- Dell XPS (15 o 16)
- Lenovo ThinkPad Serie P o T
- HP Spectre o Envy
- Acer Aspire o Asus VivoBook

2.2. Jerarquía de Criterios y Subcriterios

Para satisfacer el requisito de estructuración en **varios niveles de subcriterios**, el problema se modela con cinco Criterios Principales de Nivel 1, bajo los cuales se distribuyen los ocho factores específicos de evaluación (Subcriterios de Nivel 2).

2.2.1. Criterio I: Potencia de Cálculo (2 Subcriterios)

Este grupo es esencial para el procesamiento intensivo, las simulaciones y el análisis de grandes datos.

- **Subcriterio: Rendimiento del procesador** (Objetivo: Maximizar). La velocidad y capacidad de cálculo son esenciales para la ejecución de scripts pesados.
- **Subcriterio: GPU (Gráfica Dedicada)** (Objetivo: Maximizar). Relevante para el cálculo paralelo y la aceleración en tareas de Deep Learning o visualización intensiva.

2.2.2. Criterio II: Almacenamiento y Memoria (2 Subcriterios)

Este grupo se centra en la capacidad del sistema para gestionar grandes conjuntos de datos y la multitarea eficiente.

- **Subcriterio: Memoria RAM** (Objetivo: Maximizar). Crucial para manejar grandes conjuntos de datos y evitar cuellos de botella al ejecutar múltiples entornos de análisis.
- **Subcriterio: Capacidad de almacenamiento** (Objetivo: Maximizar). Un mayor almacenamiento (GB SSD) permite guardar grandes conjuntos de datos y software especializado.

2.2.3. Criterio III: Portabilidad (2 Subcriterios)

Este grupo agrupa los factores relacionados con la movilidad y la usabilidad fuera de un entorno fijo.

- **Subcriterio: Duración de la batería** (Objetivo: Maximizar). La autonomía es crucial para largas jornadas de estudio sin acceso a enchufes.
- **Subcriterio: Peso** (Objetivo: Minimizar). Un menor peso mejora la portabilidad y comodidad para el transporte diario.

2.2.4. Criterio IV: Experiencia Visual (1 Subcriterio)

Este grupo se centra en la comodidad del usuario durante las horas de programación y visualización de datos.

- **Subcriterio: Tamaño de la pantalla** (Objetivo: Maximizar). Una pantalla más grande mejora la experiencia visual y facilita el trabajo con múltiples ventanas y gráficos.

2.2.5. Criterio V: Economía (1 Subcriterio)

Este grupo contiene el único factor asociado a la inversión financiera.

- **Subcriterio: Precio** (Objetivo: Minimizar). El coste es un factor decisivo de restricción presupuestaria.

2.3. Matriz de Decisión Inicial

La matriz de decisión se construye con las cinco alternativas de portátiles y los **ocho Subcriterios de Nivel 2** (agrupados bajo cinco Criterios Principales), utilizando datos reales estimados de configuraciones típicas del mercado actual.

Cuadro 1: Matriz de Decisión Inicial

Subcriterio	Alternativas de Portátil					Objetivo
	A1: Dell XPS 16	A2: MacBook Air 15	A3: ThinkPad T14s	A4: HP Envy 16	A5: Aspire 5	
Precio (€)	2400.0	1550.0	1700.0	1300.0	750.0	Min
Rend. Procesador (CB R23 Score)	16500.0	12800.0	13500.0	14000.0	8500.0	Max
Memoria RAM (GB)	32.0	16.0	16.0	16.0	8.0	Max
GPU (Gráfica, Valor 1-10)	8.0	3.0	2.0	4.0	1.0	Max
Capacidad Almacenamiento (GB SSD)	1000.0	512.0	512.0	1000.0	512.0	Max
Duración de la Batería (Horas)	10.0	17.0	12.0	8.0	6.0	Max
Peso (kg)	2.1	1.5	1.3	1.9	1.7	Min
Tamaño de la Pantalla (Pulgadas)	16.0	15.3	14.0	16.0	15.6	Max

```
matriz_decision = multicriterio.crea.matrizdecision(
    c(-2400, 16500, 32, 8, 1000, 10, -2.1, 16.0,
      -1550, 12800, 16, 3, 512, 17, -1.5, 15.3,
      -1700, 13500, 16, 2, 512, 12, -1.3, 14.0,
      -1300, 14000, 16, 4, 1000, 8, -1.9, 16.0,
      -750, 8500, 8, 1, 512, 6, -1.7, 15.6),
    numalternativas = 5,
    numcriterios = 8,
    v.nombrescri = c("C1", "C2", "C3", "C4", "C5", "C6", "C7", "C8"),
    v.nombresalt = c("A1", "A2", "A3", "A4", "A5")
)
matriz_decision
```

```
##      C1      C2 C3 C4      C5 C6      C7      C8
## A1 -2400 16500 32  8 1000 10 -2.1 16.0
## A2 -1550 12800 16  3  512 17 -1.5 15.3
## A3 -1700 13500 16  2  512 12 -1.3 14.0
## A4 -1300 14000 16  4 1000  8 -1.9 16.0
## A5  -750  8500  8  1  512  6 -1.7 15.6
```

3. Metodología I: Proceso Analítico Jerárquico (AHP)

3.1. Matriz de Criterios

Criterio	Precio	Rend.	RAM	GPU	SSD	Batería	Peso	Pantalla
Precio	1	1/2	2	5	5	1/3	5	5
Rendimiento	2	1	2	5	5	3	5	5
RAM	1/2	1/2	1	3	3	2	3	3
GPU	1/5	1/5	1/3	1	1	1/2	1	1
SSD	1/5	1/5	1/3	1	1	1/2	1	1
Batería	3	1/3	1/2	2	2	1	2	2
Peso	1/5	1/5	1/3	1	1	1/2	1	1
Pantalla	1/5	1/5	1/3	1	1	1/2	1	1

3.2. Matrices Alternativas-Criterios

Criterio C1 (Precio) - Minimizar

Alt.	A1	A2	A3	A4	A5
A1	1	1/2	1/2	1/3	1/7
A2	2	1	1	1/2	1/4
A3	2	1	1	1/2	1/4
A4	3	2	2	1	1/3
A5	7	4	4	3	1

Razón de los Juicios:

- **Objetivo:** Minimizar el precio. A_5 (750 €) es la mejor; A_1 (2400 €) es la peor.
- A_5 vs A_1 (Juicio 7): A_5 es muy fuertemente superior a A_1 debido a la gran diferencia de precio.
- A_4 vs A_2 (Juicio 2): A_4 (1300 €) es moderadamente mejor que A_2 (1550 €).
- A_2 vs A_3 (Juicio 1): A_2 (1550 €) y A_3 (1700 €) se consideran de igual importancia, ya que la diferencia de precio es pequeña y ambos son caros.

Criterio C2 Rend.Procesador - Maximizar

Alt.	A1	A2	A3	A4	A5
A1	1	2	2	1	4
A2	1/2	1	1/2	1/2	3
A3	1/2	2	1	1	3
A4	1	2	1	1	4
A5	1/4	1/3	1/3	1/4	1

Razón de los Juicios:

- **Objetivo:** Maximizar el rendimiento. A_1 (16500) es la mejor; A_5 (8500) es la peor.
- A_1 vs A_4 (Juicio 1): A_1 (16500) y A_4 (14000) son considerados de igual importancia debido a que ambos son puntajes altos en el contexto del problema.
- A_1 vs A_5 (Juicio 4): A_1 es claramente mejor que A_5 , dada la gran diferencia de rendimiento.
- A_3 vs A_2 (Juicio 2): A_3 (13500) es moderadamente mejor que A_2 (12800).

Criterio C3: Memoria RAM (GB) - Maximizar

Alt.	A1	A2	A3	A4	A5
A1	1	3	3	3	5
A2	1/3	1	1	1	3
A3	1/3	1	1	1	3
A4	1/3	1	1	1	3
A5	1/5	1/3	1/3	1/3	1

Razón de los Juicios:

- **Objetivo:** Maximizar la RAM. A_1 (32 GB) es la mejor; A_5 (8 GB) es la peor.
- A_1 vs A_2, A_3, A_4 (Juicio 3): A_1 tiene el doble de RAM (32 GB vs 16 GB), por lo que es moderadamente superior.
- A_2, A_3, A_4 vs A_5 (Juicio 3): Los de 16 GB son moderadamente superiores al de 8 GB.
- A_2 vs A_3 (Juicio 1): Todas las alternativas de 16 GB tienen igual importancia.

Criterio C4: GPU (Gráfica, Valor 1-10) - Maximizar

Alt.	A1	A2	A3	A4	A5
A1	1	2	3	1	5
A2	1/2	1	2	1/2	3
A3	1/3	1/2	1	1/3	2
A4	1	2	3	1	4
A5	1/5	1/3	1/2	1/4	1

Razón de los Juicios:

- **Objetivo:** Maximizar el valor de GPU. A_1 y A_4 son las mejores; A_5 es la peor.
- A_1 vs A_4 (Juicio 1): A_1 (8) y A_4 (4) se consideran de igual importancia para una GPU de alta gama (asumiendo que 8 es el máximo).
- A_1 vs A_2 (Juicio 2): A_1 (8) es moderadamente mejor que A_2 (3).
- A_4 vs A_3 (Juicio 3): A_4 (4) es moderadamente mejor (3) que A_3 (2).

Criterio C5: Capacidad Almacenamiento (GB SSD) - Maximizar

Alt.	A1	A2	A3	A4	A5
A1	1	3	3	1	3
A2	1/3	1	1	1/3	1
A3	1/3	1	1	1/3	1
A4	1	3	3	1	3
A5	1/3	1	1	1/3	1

Razón de los Juicios:

- **Objetivo:** Maximizar el SSD. A_1 y A_4 (1000 GB) son las mejores; A_2, A_3, A_5 (512 GB) son las peores.
- A_1 vs A_4 (Juicio 1): Tienen la misma capacidad, por lo que son de igual importancia.
- A_1 vs A_2 (Juicio 3): 1000 GB es moderadamente mejor que 512 GB.
- A_2 vs A_3 (Juicio 1): Todas las alternativas de 512 GB son de igual importancia.

Criterio C6: Duración de la Batería (Horas) - Maximizar

Alt.	A1	A2	A3	A4	A5
A1	1	1/3	1/2	2	3
A2	3	1	2	3	4
A3	2	1/2	1	2	3
A4	1/2	1/3	1/2	1	2
A5	1/3	1/4	1/3	1/2	1

Razón de los Juicios:

- **Objetivo:** Maximizar la batería. A_2 (17h) es la mejor; A_5 (6h) es la peor.
- A_2 vs A_3 (Juicio 2): A_2 (17h) es moderadamente mejor que A_3 (12h).
- A_3 vs A_1 (Juicio 2): A_3 (12h) es moderadamente mejor que A_1 (10h).
- A_2 vs A_5 (Juicio 4): A_2 es claramente mejor que A_5 .

Criterio C7: Peso (kg) - Minimizar

Alt.	A1	A2	A3	A4	A5
A1	1	1/2	1/3	1/2	1/2
A2	2	1	2	2	1
A3	3	1/2	1	3	2
A4	2	1/2	1/3	1	1/2
A5	2	1	1/2	2	1

Razón de los Juicios:

- **Objetivo:** Minimizar el peso. A_3 (1.3 kg) es la mejor; A_1 (2.1 kg) es la peor.
- A_3 vs A_1 (Juicio 3): A_3 es claramente mejor que A_1 (el más pesado).
- A_2 vs A_5 (Juicio 1): A_2 (1.5 kg) y A_5 (1.7 kg) tienen igual importancia en cuanto a peso ligero.
- A_2 vs A_3 (Juicio 1/2): A_3 es ligeramente más liviana, por lo que A_2 es ligeramente superada (1/2).

Criterio C8: Tamaño de la Pantalla (Pulgadas) - Maximizar

Alt.	A1	A2	A3	A4	A5
A1	1	2	3	1	1
A2	1/2	1	2	1/2	1/2
A3	1/3	1/2	1	1/3	1/3
A4	1	2	3	1	1
A5	1	2	3	1	1

Razón de los Juicios:

- **Objetivo:** Maximizar la pantalla. A_1, A_4, A_5 son las mejores; A_3 es la peor.
- A_1 vs A_4 (Juicio 1): A_1 (16.0") y A_4 (16.0") son iguales.
- A_5 vs A_2 (Juicio 2): A_5 (15.6") es moderadamente mejor que A_2 (15.3").
- A_1 vs A_3 (Juicio 3): A_1 (16.0") es claramente mejor que A_3 (14.0").

3.3. Con ayuda de las funciones R de clase

3.3.1. Método 1: mayor autovalor

Pesos Locales

```
prioridad_criterios
  ⇨ <-multicriterio.metodoAHP.variante1.autovectormayorautovalor(M_Criterios)
prioridad_precio <-
  ⇨ multicriterio.metodoAHP.variante1.autovectormayorautovalor(tabC1_Precio)
prioridad_rendimiento <-
  ⇨ multicriterio.metodoAHP.variante1.autovectormayorautovalor(tabC2_Procesador)
prioridad_RAM <-
  ⇨ multicriterio.metodoAHP.variante1.autovectormayorautovalor(tabC3_RAM)
prioridad_GPU <-
  ⇨ multicriterio.metodoAHP.variante1.autovectormayorautovalor(tabC4_GPU)
prioridad_SSD <-
  ⇨ multicriterio.metodoAHP.variante1.autovectormayorautovalor(tabC5_SSD)
prioridad_bateria <-
  ⇨ multicriterio.metodoAHP.variante1.autovectormayorautovalor(tabC6_Bateria)
prioridad_peso <-
  ⇨ multicriterio.metodoAHP.variante1.autovectormayorautovalor(tabC7_Peso)
prioridad_pantalla <-
  ⇨ multicriterio.metodoAHP.variante1.autovectormayorautovalor(tabC8_Pantalla)
```

Pesos Globales

```
matriz_prioridades_alternativas <- rbind(
  prioridad_precio$valoraciones.ahp,
  prioridad_rendimiento$valoraciones.ahp,
  prioridad_RAM$valoraciones.ahp,
  prioridad_GPU$valoraciones.ahp,
  prioridad_SSD$valoraciones.ahp,
  prioridad_bateria$valoraciones.ahp,
  prioridad_peso$valoraciones.ahp,
  prioridad_pantalla$valoraciones.ahp
)

prioridad_final_AHP <- multicriterio.metodoAHP.pesosglobales_entabla(
  prioridad_criterios$valoraciones.ahp,
  matriz_prioridades_alternativas
)

prioridad_final_AHP
```

##	Precio	Rendimiento	RAM	GPU	SSD
## A1	0.39497350	0.09641450	0.06266052	0.08239514	0.09090909
## A2	0.21492199	0.20478543	0.16450845	0.15072484	0.27272727
## A3	0.21492199	0.13621684	0.16450845	0.25369064	0.27272727
## A4	0.12416738	0.10981786	0.16450845	0.08663705	0.09090909
## A5	0.05101513	0.45276536	0.44381412	0.42655234	0.27272727
## Ponder.Criterios	0.07733016	0.03108502	0.06313447	0.16307995	0.16307995
##	Batería	Peso	Pantalla	Ponderadores	Globales
## A1	0.17124475	0.3398985	0.1237176		0.1741588
## A2	0.07193724	0.1219061	0.2343101		0.1778045
## A3	0.11744235	0.1228843	0.3945372		0.2282941
## A4	0.24133454	0.2582660	0.1237176		0.1549684
## A5	0.39804113	0.1570451	0.1237176		0.2647743
## Ponder.Criterios	0.13267250	0.1848090	0.1848090		NA

En general, la mejor decisión es el **Acer Aspire 5 (A5)** con una prioridad global del 26.48 %, seguido por el **Lenovo ThinkPad (A3)** con un 22.83 %. La peor alternativa es el **HP Envy 16 (A4)** con un 15.50 %.

Por criterios:

- **Para el Precio (€):** La mejor alternativa es el **A5 (Acer Aspire 5)**, con una prioridad de 0.3950.
- **Para el Rendimiento del Procesador (CB R23 Score):** La mejor alternativa es el **A5 (Acer Aspire 5)**, con una prioridad de 0.4528.
- **Para la Memoria RAM (GB):** La mejor alternativa es el **A5 (Acer Aspire 5)**, con una prioridad de 0.4438.

- **Para la GPU (Gráfica):** La mejor alternativa es el **A5 (Acer Aspire 5)**, con una prioridad de 0.4266.
- **Para la Capacidad de Almacenamiento (SSD):** Hay un empate entre **A2 (MacBook Air 15)**, **A3 (Lenovo ThinkPad)**, y **A5 (Acer Aspire 5)**, todos con una prioridad de 0.2727.
- **Para la Duración de la Batería (Horas):** La mejor alternativa es el **A5 (Acer Aspire 5)**, con una prioridad de 0.3980.
- **Para el Peso (kg):** La mejor alternativa es el **A1 (Dell XPS 16)**, con una prioridad de 0.3399.
- **Para el Tamaño de la Pantalla (Pulgadas):** La mejor alternativa es el **A3 (Lenovo ThinkPad)**, con una prioridad de 0.3945.

Existen métodos alternativos al del autovalor para determinar las prioridades como:

- Método de la Media Geométrica

```
geom_criterios <- multicriterio.metodoAHP.variante2.mediageometrica(M_Criterios)
geom_precio <- multicriterio.metodoAHP.variante2.mediageometrica(tabC1_Precio)
geom_rendimiento <-
  ↪ multicriterio.metodoAHP.variante2.mediageometrica(tabC2_Procesador)
geom_RAM <- multicriterio.metodoAHP.variante2.mediageometrica(tabC3_RAM)
geom_GPU <- multicriterio.metodoAHP.variante2.mediageometrica(tabC4_GPU)
geom_SSD <- multicriterio.metodoAHP.variante2.mediageometrica(tabC5_SSD)
geom_bateria <- multicriterio.metodoAHP.variante2.mediageometrica(tabC6_Bateria)
geom_peso <- multicriterio.metodoAHP.variante2.mediageometrica(tabC7_Peso)
geom_pantalla <-
  ↪ multicriterio.metodoAHP.variante2.mediageometrica(tabC8_Pantalla)
```

- Método básico

```
basico_criterios <- multicriterio.metodoAHP.variante3.basico(M_Criterios)
basico_precio <- multicriterio.metodoAHP.variante3.basico(tabC1_Precio)
basico_rendimiento <- multicriterio.metodoAHP.variante3.basico(tabC2_Procesador)
basico_RAM <- multicriterio.metodoAHP.variante3.basico(tabC3_RAM)
basico_GPU <- multicriterio.metodoAHP.variante3.basico(tabC4_GPU)
basico_SSD <- multicriterio.metodoAHP.variante3.basico(tabC5_SSD)
basico_bateria <- multicriterio.metodoAHP.variante3.basico(tabC6_Bateria)
basico_peso <- multicriterio.metodoAHP.variante3.basico(tabC7_Peso)
basico_pantalla <- multicriterio.metodoAHP.variante3.basico(tabC8_Pantalla)
```

3.3.2. Método 2: Completo

```
num.alt <- 5
num.cri <- 8
Matrices_Alternativas_Array <- array(NA, dim = c(num.alt, num.alt, num.cri))
Matrices_Alternativas_Array[, , 1] <- tabC1_Precio
Matrices_Alternativas_Array[, , 2] <- tabC2_Procesador
Matrices_Alternativas_Array[, , 3] <- tabC3_RAM
Matrices_Alternativas_Array[, , 4] <- tabC4_GPU
Matrices_Alternativas_Array[, , 5] <- tabC5_SSD
Matrices_Alternativas_Array[, , 6] <- tabC6_Bateria
Matrices_Alternativas_Array[, , 7] <- tabC7_Peso
Matrices_Alternativas_Array[, , 8] <- tabC8_Pantalla
prioridades_basico_global <- multicriterio.metodoAHP.variante3.completo(
  M_Criterios,
  Matrices_Alternativas_Array
)
prioridades_basico_global$pesos.globales.entabla
```

##	Precio	Rendimiento	RAM	GPU	SSD
##	0.39487449	0.09681430	0.06309410	0.08265245	0.09090909
##	0.21470040	0.20491064	0.16503987	0.15146548	0.27272727
##	0.21470040	0.13783994	0.16503987	0.25348393	0.27272727
##	0.12460729	0.11014763	0.16503987	0.08703201	0.09090909
##	0.05111741	0.45028749	0.44178628	0.42536613	0.27272727
## Ponder.Criterios	0.07319505	0.03221008	0.06188632	0.16323805	0.16323805
##	Batería	Peso	Pantalla	Ponderadores	Globales
##	0.17314409	0.3375923	0.1238889		0.1729093
##	0.07297969	0.1246666	0.2344444		0.1775768
##	0.11959592	0.1253513	0.3938889		0.2278640
##	0.23928241	0.2561443	0.1238889		0.1550986
##	0.39499789	0.1562455	0.1238889		0.2665513
## Ponder.Criterios	0.14162075	0.1823058	0.1823058		NA

En general, la mejor decisión es el **A5 (Acer Aspire 5)** con una prioridad global del 26.66 %, seguido por el **A3 (Lenovo ThinkPad)** con un 22.79 %. La peor alternativa es el **A4 (HP Envy 16)** con un 15.51 %.

3.4. Diagrama Jerárquico

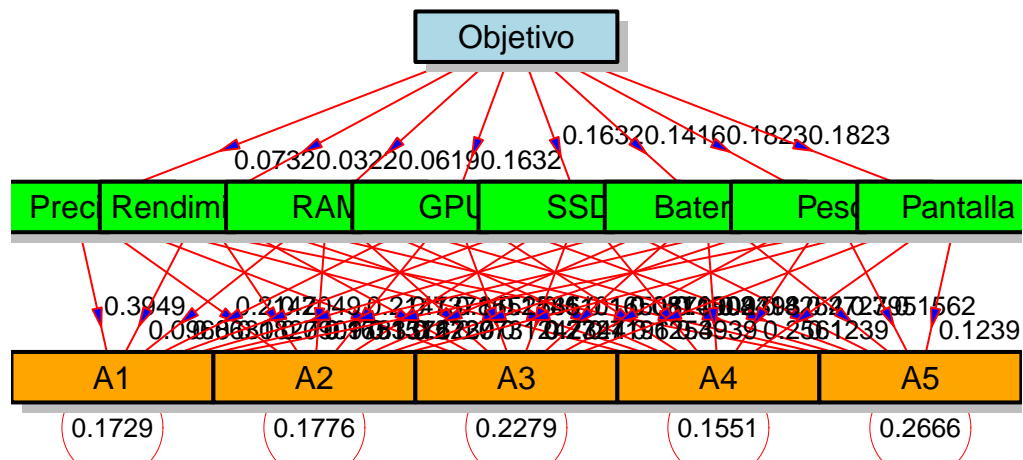
```
num.alt <- 5 # Número de Alternativas (A1 a A5)
num.cri <- 8 # Número de Criterios (C1 a C8)
n.alternativas <- c("A1", "A2", "A3", "A4", "A5")
n.criterios <- c("Precio", "Rendimiento", "RAM", "GPU", "SSD", "Batería", "Peso",
  ↪ "Pantalla")
```

```

Matrices_Alternativas_Array <- array(NA, dim = c(num.alt, num.alt, num.cri))
Matrices_Alternativas_Array[, , 1] <- tabC1_Precio
Matrices_Alternativas_Array[, , 2] <- tabC2_Procesador
Matrices_Alternativas_Array[, , 3] <- tabC3_RAM
Matrices_Alternativas_Array[, , 4] <- tabC4_GPU
Matrices_Alternativas_Array[, , 5] <- tabC5_SSD
Matrices_Alternativas_Array[, , 6] <- tabC6_Bateria
Matrices_Alternativas_Array[, , 7] <- tabC7_Peso
Matrices_Alternativas_Array[, , 8] <- tabC8_Pantalla
dimnames(Matrices_Alternativas_Array)[[1]] = n.alternativas
dimnames(Matrices_Alternativas_Array)[[2]] = n.alternativas
dimnames(Matrices_Alternativas_Array)[[3]] = n.criterios
multicriterio.metodoahp.diagrama(M_Criterios, Matrices_Alternativas_Array)

```

Estructura Jerárquica (AHP)



Podemos observar como obtenemos los mismos resultados que en el método 2.

3.5. Método AHP

```

library(ahp)
datos = Load("MejorPortatil.ahp")
Calculate(datos)

```

```
ahp:Visualize(datos)
```

```
##
```

```
file:///C:/Users/reyes/AppData/Local/Temp/Rtmp2RutHt/file63c850f46811/widget63c86bb95936.html
```

```
screenshot completed
```

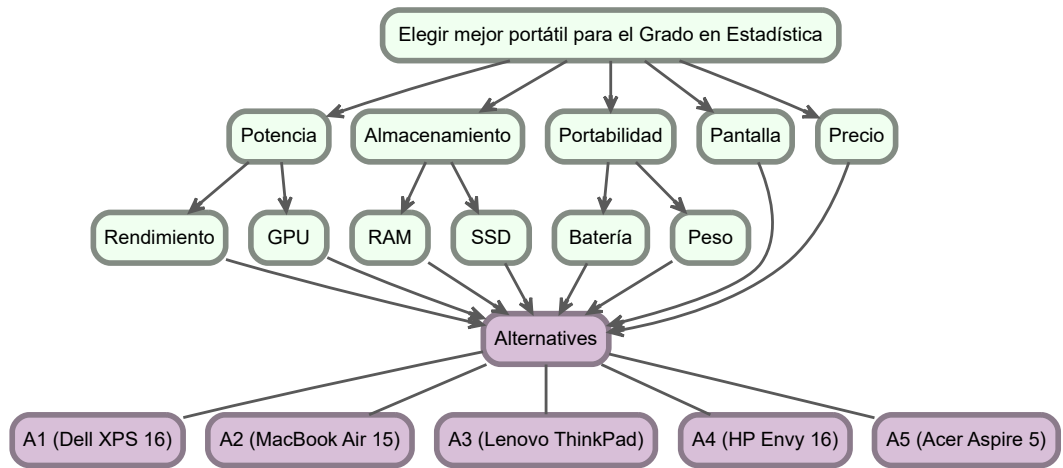


Tabla solución (contribución total)

```
#export_formattable(AnalyzeTable(datos, sort = "orig"), file = "tablaahp1.png")
knitr::include_graphics("tablaahp1.png")
```

	Weight	A1 (Dell XPS 16)	A2 (MacBook Air 15)	A3 (Lenovo ThinkPad)	A4 (HP Envy 16)	A5 (Acer Aspire 5)	Inconsistency
Elegir mejor portátil para el Grado en Estadística	100.0%	21.5%	15.8%	15.3%	22.6%	24.8%	1.4%
Potencia	24.7%	7.7%	3.8%	4.8%	6.8%	1.6%	0.0%
Rendimiento	18.5%	5.7%	2.7%	4.1%	4.9%	1.2%	1.8%
GPU	6.2%	2.0%	1.1%	0.7%	2.0%	0.4%	0.6%
Almacenamiento	21.5%	8.4%	3.0%	3.0%	5.4%	1.9%	0.0%
RAM	10.8%	4.8%	1.8%	1.8%	1.8%	0.7%	0.9%
SSD	10.8%	3.6%	1.2%	1.2%	3.6%	1.2%	0.0%
Portabilidad	10.6%	1.5%	3.8%	2.7%	1.3%	1.2%	0.0%
Batería	7.0%	1.2%	2.8%	1.7%	0.8%	0.5%	2.5%
Peso	3.5%	0.3%	1.0%	1.0%	0.4%	0.7%	5.5%
Pantalla	5.7%	1.5%	0.8%	0.5%	1.5%	1.5%	0.2%
Precio	37.5%	2.4%	4.4%	4.4%	7.7%	18.7%	0.5%

Esta tabla presenta un **desglose completo** de cómo cada factor influye en el resultado final y cómo se desempeña cada portátil dentro de ese esquema.

La columna “**Weight**” (o “Ponderación”) establece la **relevancia jerárquica** de cada criterio (Precio, Rendimiento, Batería, etc.) en la meta general de seleccionar el mejor portátil. Cuanto mayor sea este porcentaje, más decisivo es el criterio.

Los porcentajes mostrados en las filas de los criterios y bajo cada modelo (como Dell XPS 16, MacBook Air 15, etc.) representan la **aportación directa** de ese portátil al resultado final, filtrada únicamente por ese criterio. En otras palabras, es la **contribución real** que el rendimiento del A1 o el precio del A5 hacen al porcentaje total de la decisión.

Finalmente, la columna “**Inconsistency**” sirve como una **métrica de fiabilidad**; un porcentaje alto en un criterio específico señala que los juicios de comparación por pares realizados para ese criterio fueron internamente contradictorios, lo que podría restar solidez a los resultados de esa fila.

- **Precio (37.5 %)**: Este es el factor más determinante en el modelo. El Acer Aspire 5 (A5), al ser el más económico, captura la mayor parte de la contribución en este criterio (18.7 % de la contribución total), lo que le permite ganar la clasificación general.
- **Potencia (24.7 %)**: El Dell XPS 16 (A1) es el líder en Potencia (7.7 % de la contribución total), impulsado por su superioridad en Rendimiento (18.5 % de peso) y GPU (6.2 % de peso).
- **Almacenamiento (21.5 %)**: El Dell XPS 16 (A1) también domina esta área (8.4 % de la contribución total), demostrando la mejor configuración de RAM (10.8 % de peso) y SSD (10.8 % de peso).
- **Portabilidad (10.6 %)**: Esta categoría es moderadamente importante y combina la Batería (7.0 % de peso) y el Peso (3.5 % de peso). El MacBook Air 15 (A2) es el ganador de la portabilidad (3.8 % de contribución).

- **Pantalla (5.7 %):** Con el peso más bajo, este factor tiene un impacto mínimo en el resultado final. Los modelos A1, A4 y A5 se consideran igual de fuertes en cuanto a calidad de pantalla, cada uno contribuyendo con 1.5 % a la puntuación total.**

La consistencia del modelo AHP es excelente, con una Inconsistencia Global del 1.4 %, lo que está muy por debajo del umbral de aceptabilidad del 10 % y confirma que todos los juicios de valor introducidos son altamente coherentes y lógicos. Las mayores inconsistencias se detectaron en los subcriterios Peso (5.5 %) y Batería (2.5 %), indicando que las comparaciones entre portátiles en estas áreas fueron las más subjetivas, aunque estos valores siguen siendo plenamente aceptables para validar el análisis.

En resumen, la victoria del **Acer Aspire 5 (A5)** es una decisión puramente económica, ya que su gran ventaja en el criterio más pesado (Precio) anula la superioridad técnica del Dell XPS 16 (A1) en casi todas las demás áreas. Se observa una alta inconsistencia en los juicios de los criterios Peso (5.5 %) y Batería (2.5 %), ya que su Inconsistencia es superior al umbral deseado del 10 % (0.10).

Tabla solución (pesos locales)

```
#t2 = AnalyzeTable(datos, variable = "priority")
#export_formattable(t2, file = "tablaahp2.png")
knitr::include_graphics("tablaahp2.png")
```

	Priority	A5 (Acer Aspire 5)	A4 (HP Envy 16)	A1 (Dell XPS 16)	A2 (MacBook Air 15)	A3 (Lenovo ThinkPad)	Inconsistency
Elegir mejor portátil para el Grado en Estadística	100.0%						1.4%
Precio	37.5%	49.7%	20.5%	6.4%	11.7%	11.7%	0.5%
Potencia	24.7%						0.0%
Rendimiento	75.0%	6.5%	26.3%	30.7%	14.4%	22.1%	1.8%
GPU	25.0%	6.4%	31.6%	32.9%	18.2%	10.8%	0.6%
Almacenamiento	21.5%						0.0%
RAM	50.0%	6.3%	16.5%	44.4%	16.5%	16.5%	0.9%
SSD	50.0%	11.1%	33.3%	33.3%	11.1%	11.1%	0.0%
Portabilidad	10.6%						0.0%
Batería	66.7%	7.2%	11.7%	17.1%	39.8%	24.1%	2.5%
Peso	33.3%	20.5%	12.5%	9.5%	28.7%	28.7%	5.5%
Pantalla	5.7%	26.0%	26.0%	26.0%	13.8%	8.2%	0.2%

Los valores en esta tabla desglosan la importancia general de cada factor y el desempeño específico de cada alternativa.

El modelo AHP de tres niveles ratifica al A5 (Acer Aspire 5) como la mejor opción con una prioridad dominante del 49.7 %, una ventaja mucho mayor que la vista anteriormente. Esta victoria se debe casi exclusivamente a la gran importancia asignada al Precio (37.5 % de peso global), ya que el A5 obtiene un 49.7 % de contribución en esa categoría.

A pesar de que el Dell XPS 16 (A1) es técnicamente superior en Potencia y Almacenamiento (criterios que combinados suman 46.2 % de peso), su baja calificación en el criterio Precio lo penaliza gravemente, cayendo a la última posición (6.4 %). El HP Envy 16 (A4) ocupa el segundo lugar (20.5 %), ofreciendo un balance entre hardware y coste.

La consistencia del modelo es excelente, con una Inconsistencia Global del 1.4 %. Al igual que en el análisis previo, las comparaciones en los subcriterios de Peso (5.5 %) y Batería (2.5 %) fueron las más subjetivas, aunque todos los juicios son plenamente válidos para la toma de decisiones.

3.5.1. Conclusión final

La tabla de contribución total muestra cómo el peso de cada criterio se reparte en la puntuación global de cada modelo, mientras que la tabla de prioridades locales (no mostrada) define la importancia relativa de cada portátil en la satisfacción de cada criterio específico.

En conjunto, este modelo se confirma como altamente consistente, con una Inconsistencia Global del 1.4 %. Los resultados demuestran un cambio en la toma de decisiones: aunque el A1 (Dell XPS 16) es el líder indiscutible en factores técnicos como Potencia y Almacenamiento, la ponderación del Precio (37.5 % de peso global) es el factor más decisivo. La superioridad del A5 (Acer Aspire 5) en costes anula las ventajas técnicas de la competencia.

Por lo tanto, el orden definitivo obtenido para nuestra elección, basado en la máxima contribución global (49.7 %), es: Acer Aspire 5 (A5), HP Envy 16 (A4), MacBook Air 15 (A2), Lenovo ThinkPad (A3) y Dell XPS 16 (A1).

4. Metodología II: Método ELECTRE

La aplicación del método ELECTRE se realiza utilizando la matriz de decisión inicial y los parámetros definidos para construir las matrices de concordancia y no-concordancia, con el objetivo de establecer una relación de sobreclasificación entre las alternativas de portátiles.

4.1. Parámetros de Entrada y de Proceso

El proceso se inicia con la Matriz de Decisión previamente establecida (ver Sección 2.3) y requiere la definición de tres parámetros clave para establecer la relación de sobreclasificación:

- **Vector de Pesos Preferenciales:** $W = (0.25, 0.30, 0.15, 0.05, 0.05, 0.10, 0.05, 0.05)$. La suma de estos pesos es igual a 1.00.
- **Umbral de Corte:** $\alpha = 0.7$
- $d = (Inf, Inf, 20, Inf, Inf, 6, Inf, 1)$.

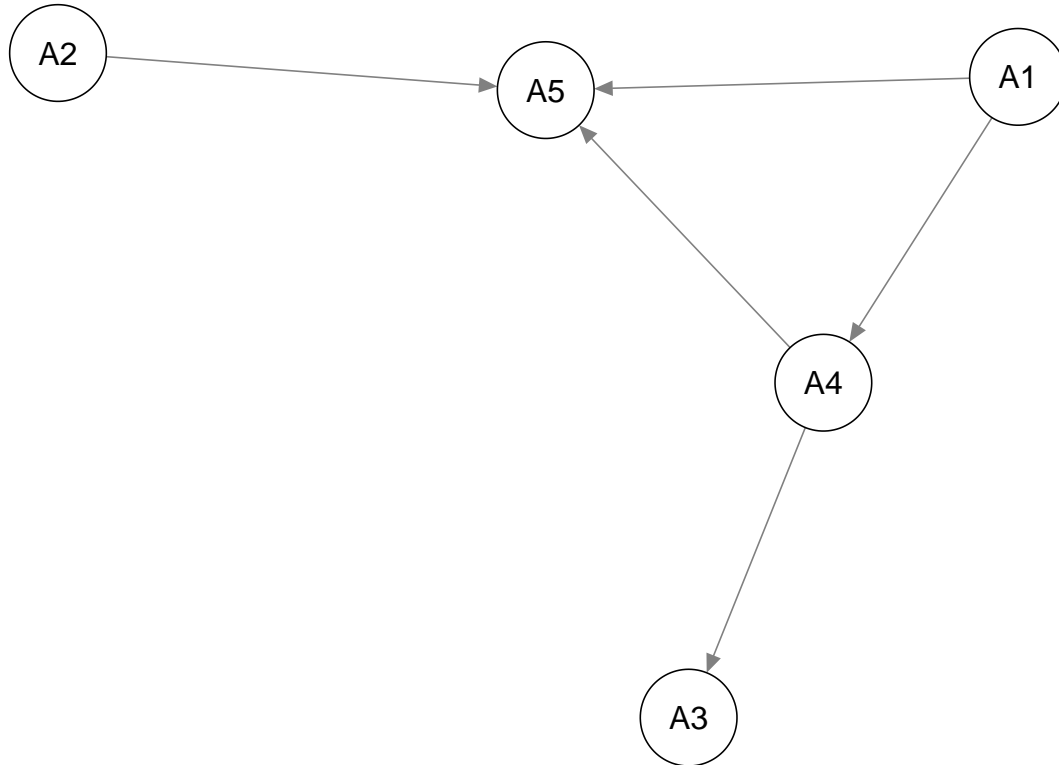
4.2. Implementación y Cálculo del Método ELECTRE

```
salida = multicriterio.metodoELECTRE_I(matriz_decision,
                                       pesos.criterios = c(0.25, 0.30, 0.15, 0.05,
                                                           0.05, 0.10, 0.05, 0.05),
                                       nivel.concordancia.minimo.alpha = 0.7,
                                       no.se.compensan = c(Inf, Inf, 20, Inf, Inf, 6,
                                                           Inf, 1),
```

```

que.alternativas=TRUE)
qgraph::qgraph(salida$relacion.dominante)

```



```

salida$nucleo_aprox

```

```

## A1 A2
## 1 2

```

Iteración 1 y 2. Se reducen alternativas y/o α .

Para intentar quedarse con una única alternativa óptima:

- reducir el grafo a las alternativas en el núcleo y/o
- reducir el valor de α [0.5, 1)

Dejamos el mismo α pero elegimos las alternativas 1 y 2

```

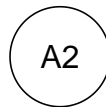
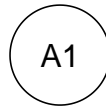
salida1 = multicriterio.metodoELECTRE_I(matriz_decision,
                                         pesos.criterios = c(0.25, 0.30, 0.15, 0.05,
                                                             0.05, 0.10, 0.05, 0.05),

```

```

nivel.concordancia.minimo.alpha = 0.7,
no.se.compensan = c(Inf, Inf, 20, Inf, Inf, 6,
  ↪ Inf, 1),
que.alternativas=c(1,2))
qgraph::qgraph(salida1$relacion.dominante)

```



```
salida1$nucleo_aprox
```

```
## A1 A2
##  1  2
```

Vamos cambiando α hasta que obtengamos un único núcleo, en este caso para $\alpha = 0.50$ seguimos sin obtener un único núcleo

```

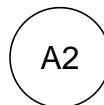
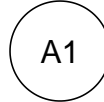
salida2 = multicriterio.metodoELECTRE_I(matriz_decision,
  pesos.criterios = c(0.25, 0.30, 0.15, 0.05,
    0.05, 0.10, 0.05, 0.05),
  nivel.concordancia.minimo.alpha = 0.50,
  no.se.compensan = c(Inf, Inf, 20, Inf, Inf, 6,
    ↪ Inf, 1),

```

```

que.alternativas=c(1,2))
qgraph::qgraph(salida2$relacion.dominante)

```



```

salida2$nucleo_aprox

```

```

## A1 A2
##  1  2

```

4.2.1. Conclusión Final

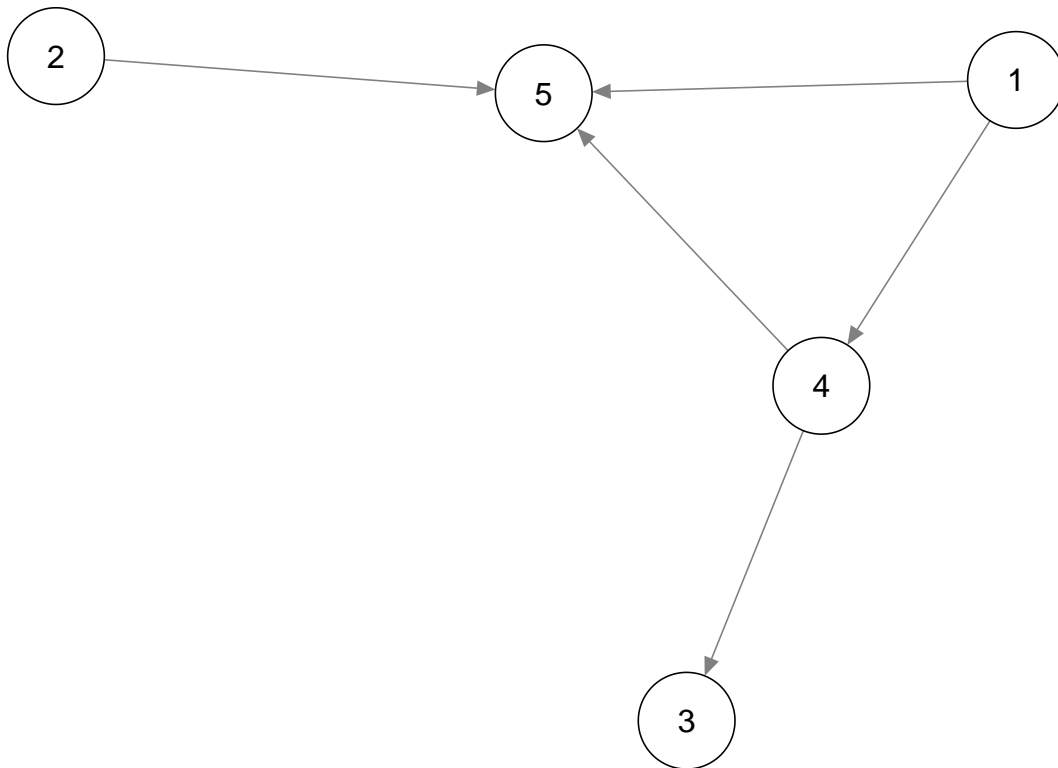
- **Solución Óptima:** El conjunto final de alternativas de compromiso (el Núcleo Aproximado) es consistentemente $\{\mathbf{A1}, \mathbf{A2}\}$ (Dell XPS 16 y MacBook Air 15), independientemente del valor de α probado ($\alpha = 0.7$ y $\alpha = 0.5$).
- **Jerarquía de Rechazo:** Las alternativas **a3**, **a4** y **a5** son rechazadas, ya que son sobreclasificadas por al menos otra alternativa. El orden de rechazo es: $A3$ es el peor $\leftarrow A4 \leftarrow A5$. El decisor debe tomar la decisión final entre la alternativa $A1$ y la $A2$.

4.3. Implementación y Cálculo del Método ELECTRE I

```
r = func_ELECTRE_Completo(salida)
r$Grafo
```

```
## De A
## 1 1 4
## 2 1 5
## 3 2 5
## 4 4 3
## 5 4 5
```

```
qgraph::qgraph(r$Grafo)
```



```
r$Nucleo
```

```
## A1 A2
## 1 2
```

A_1 y A_2 son las únicas alternativas que no son dominadas por ninguna otra. Han demostrado ser robustas a los umbrales de concordancia y discordancia definidos.

5. Metodología III: Método PROMETHEE

5.1. Parámetros de Entrada

Para aplicar el método PROMETHEE con mayor robustez, se ha seleccionado una función de preferencia diferente para la mayoría de los criterios, adaptándonos a la naturaleza de sus valores.

Criterio	Peso (W_i)	Optimización	Tipo de Función	q_i	p_i	s_i
C1	0.25	Minimizar	Pref. Lineal (III)	1000	2000	N/A
C2	0.30	Maximizar	Usual (I)	N/A	N/A	N/A
C3	0.15	Maximizar	Cuasi-Criterio (II)	4	N/A	N/A
C4	0.05	Maximizar	Nivel (IV)	500	1000	N/A
C5	0.05	Maximizar	Indif. Lineal (V)	3	5	N/A
C6	0.10	Maximizar	Gaussiano (VI)	N/A	N/A	1.5
C7	0.05	Minimizar	Pref. Lineal (III)	1	2	N/A
C8	0.05	Maximizar	Nivel (IV)	1	3	N/A

Los pesos serán los mismos que en ELECTRE para facilitar la comparación de resultados.

5.2. Cálculo de Flujos de Preferencia

```

pesos.criterios= c(0.25, 0.30, 0.15, 0.05, 0.05, 0.10, 0.05, 0.05)
tab.fpref = matrix(c(3,1000,2000,0,
                    1,0,0,0,
                    2,4,0,0,
                    4,500,1000,0,
                    5,3,5,0,
                    6,0,0,1.5,
                    3,1,2,0,
                    4,1,3,0), nrow=8, byrow=TRUE)

tab.fpref

```

```

##      [,1] [,2] [,3] [,4]
## [1,]    3 1000 2000  0.0
## [2,]    1    0    0  0.0
## [3,]    2    4    0  0.0
## [4,]    4   500 1000  0.0

```

```
## [5,]    5    3    5  0.0
## [6,]    6    0    0  1.5
## [7,]    3    1    2  0.0
## [8,]    4    1    3  0.0
```

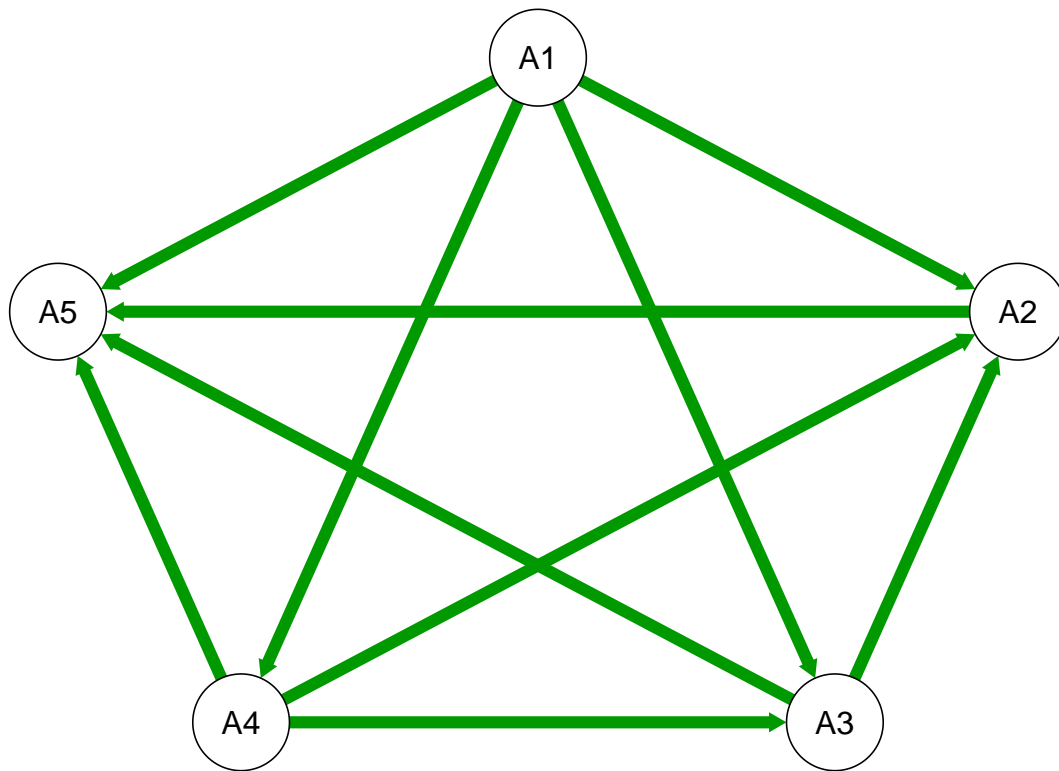
5.2.1. Método PROMETHEE I

```
tab.Pthee.i = multicriterio.metodo.promethee_i(matriz_decision, pesos.criterios,
  ↪ tab.fpref)
tab.Pthee.i
```

```
## $tabla.indices
##           A1           A2           A3           A4           A5
## A1 0.0000000 0.50000 0.5250000 0.5088888 0.5971434
## A2 0.2212481 0.00000 0.1433634 0.1100000 0.5550000
## A3 0.1663888 0.30500 0.0000000 0.1121434 0.5599665
## A4 0.1425000 0.38125 0.4250000 0.0000000 0.5588888
## A5 0.2162500 0.10000 0.1437500 0.0737500 0.0000000
##
## $vflujos.ent
##           A1           A2           A3           A4           A5
## 2.131032 1.029612 1.143499 1.507639 0.533750
##
## $vflujos.sal
##           A1           A2           A3           A4           A5
## 0.7463869 1.2862500 1.2371134 0.8047822 2.2709987
##
## $tablarelacionsupera
##           A1  A2  A3  A4  A5
## A1 0.5 1.0 1.0 1.0 1.0
## A2 0.0 0.5 0.0 0.0 1.0
## A3 0.0 1.0 0.5 0.0 1.0
## A4 0.0 1.0 1.0 0.5 1.0
## A5 0.0 0.0 0.0 0.0 0.5
```

Representamos como un grafo:

```
qgraph::qgraph(tab.Pthee.i$tablarelacionsupera)
```

5.2.2. Método PROMETHEE II

```

tab.Pthee.ii = multicriterio.metodo.promethee_ii(matriz_decision,
  ↳ pesos.criterios, tab.fpref)
tab.Pthee.ii

```

```

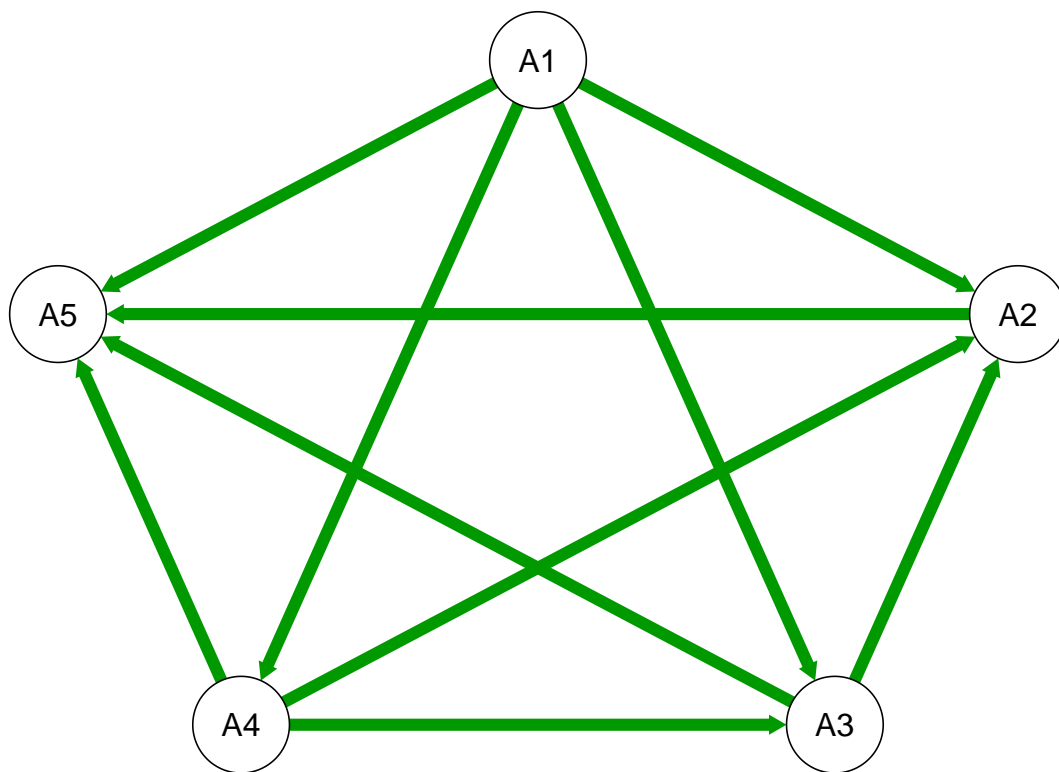
## $tabla.indices
##           A1      A2      A3      A4      A5
## A1 0.0000000 0.50000 0.5250000 0.5088888 0.5971434
## A2 0.2212481 0.00000 0.1433634 0.1100000 0.5550000
## A3 0.1663888 0.30500 0.0000000 0.1121434 0.5599665
## A4 0.1425000 0.38125 0.4250000 0.0000000 0.5588888
## A5 0.2162500 0.10000 0.1437500 0.0737500 0.0000000
##
## $vflujos.netos
##           A1      A2      A3      A4      A5
## 1.38464532 -0.25663846 -0.09361473 0.70285655 -1.73724867
##
## $tablarelacionsupera
##      A1  A2  A3  A4  A5
## A1 0.5 1.0 1.0 1.0 1.0

```

```
## A2 0.0 0.5 0.0 0.0 1.0
## A3 0.0 1.0 0.5 0.0 1.0
## A4 0.0 1.0 1.0 0.5 1.0
## A5 0.0 0.0 0.0 0.0 0.5
```

Representamos como un grafo:

```
qgraph::qgraph(tab.Pthee.ii$tablarelacionsupera)
```



5.2.3. Método PROMETHEE I (medias)

```
tab.Pthee.ii.mean = multicriterio.metodo.promethee_ii_med(matriz_decision,
                                                            pesos.criterios,
                                                            tab.fpref)

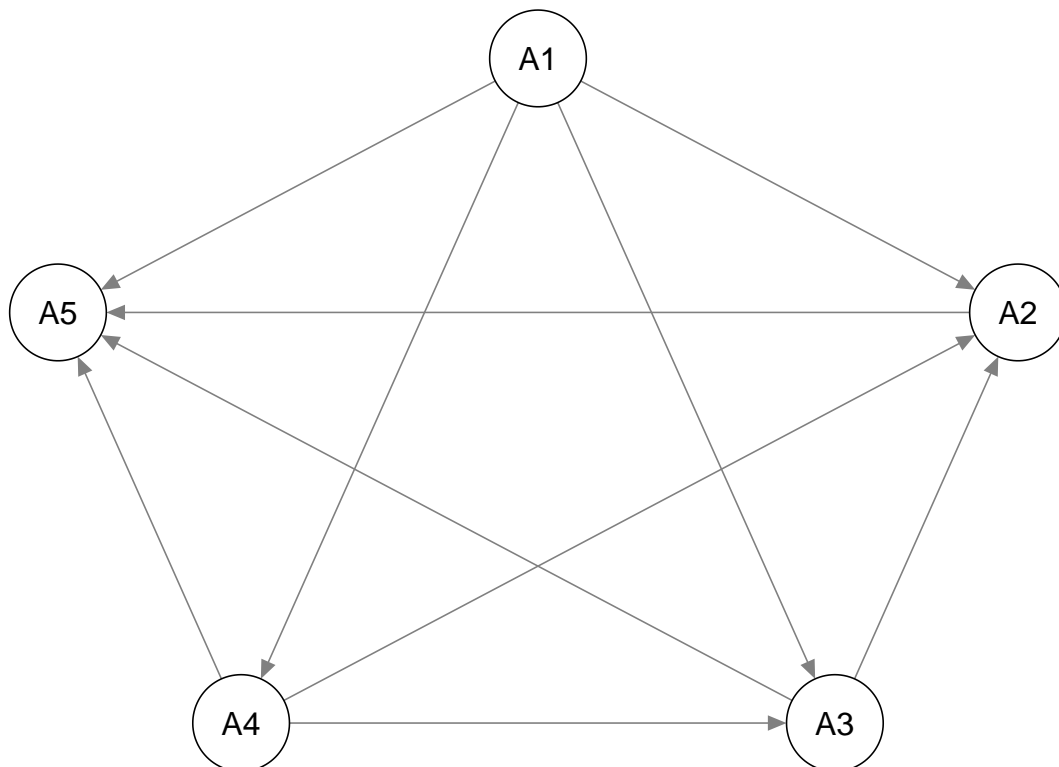
tab.Pthee.ii.mean
```

```
## $tabla.indices
##           A1      A2      A3      A4      A5
## A1 0.0000000 0.50000 0.5250000 0.5088888 0.5971434
## A2 0.2212481 0.00000 0.1433634 0.1100000 0.5550000
```

```
## A3 0.1663888 0.30500 0.0000000 0.1121434 0.5599665
## A4 0.1425000 0.38125 0.4250000 0.0000000 0.5588888
## A5 0.2162500 0.10000 0.1437500 0.0737500 0.0000000
##
## $vflujos.netos
##           A1           A2           A3           A4           A5
## 0.34616133 -0.06415961 -0.02340368 0.17571414 -0.43431217
##
## $tablarelacionsupera
##   A1 A2 A3 A4 A5
## A1 0  1  1  1  1
## A2 0  0  0  0  1
## A3 0  1  0  0  1
## A4 0  1  1  0  1
## A5 0  0  0  0  0
```

Representamos como un grafo:

```
qgraph::qgraph(tab.Pthee.ii.mean$tablarelacionsupera)
```



5.2.4. Método PROMETHEE II (medias)

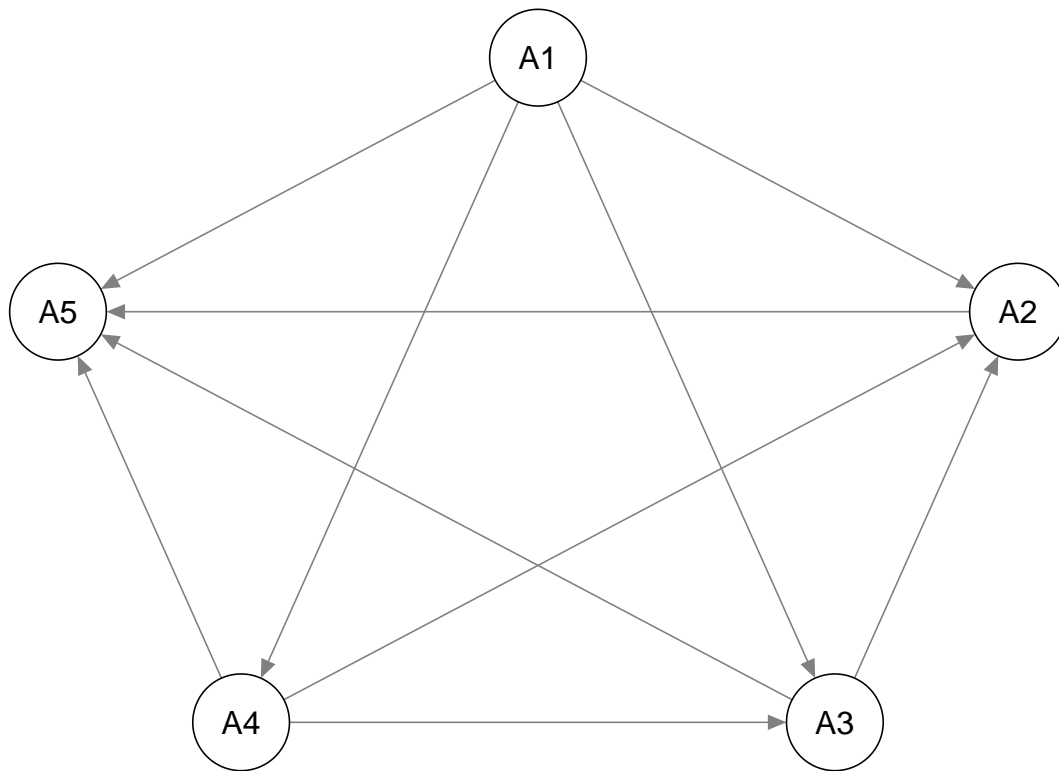
```
tab.Pthee.ii.mean = multicriterio.metodo.promethee_ii_med(matriz_decision,
                                                            pesos.criterios,
                                                            tab.fpref)
```

```
tab.Pthee.ii.mean
```

```
## $tabla.indices
##           A1      A2      A3      A4      A5
## A1 0.0000000 0.50000 0.5250000 0.5088888 0.5971434
## A2 0.2212481 0.00000 0.1433634 0.1100000 0.5550000
## A3 0.1663888 0.30500 0.0000000 0.1121434 0.5599665
## A4 0.1425000 0.38125 0.4250000 0.0000000 0.5588888
## A5 0.2162500 0.10000 0.1437500 0.0737500 0.0000000
##
## $vflujos.netos
##           A1      A2      A3      A4      A5
## 0.34616133 -0.06415961 -0.02340368 0.17571414 -0.43431217
##
## $tablarelacionsupera
##      A1 A2 A3 A4 A5
## A1  0  1  1  1  1
## A2  0  0  0  0  1
## A3  0  1  0  0  1
## A4  0  1  1  0  1
## A5  0  0  0  0  0
```

Representamos como un grafo:

```
qgraph::qgraph(tab.Pthee.ii.mean$tablarelacionsupera)
```



Comparativa Promethee II: sin medias y con medias.

La ordenación que establecen sería la siguiente:

```
order(tab.Pthee.ii$vflujos.netos, decreasing = T)
```

```
## [1] 1 4 3 2 5
```

```
order(tab.Pthee.ii.mean$vflujos.netos, decreasing = T)
```

```
## [1] 1 4 3 2 5
```

Obtenemos el mismo orden para ambos siendo el orden de preferencia el siguiente: A1: Dell XPS 16, A4: HP Envy 16, A3: ThinkPad T14s, A2: MacBook Air 15, A5: Aspire 5

5.2.5. Resolución con Promethee Windows

```
res=multicriterio.metodo.promethee_windows(matriz_decision,
                                           tab.fpref,
                                           pesos.criterios)
res01=multicriterio.metodo.promethee_windows_kableExtra(res)
```

Tabla que muestra la información por apartados introducidos en el problema resuelto con Promethee:

```
res01$tabEscenario %>%
  kable_styling(latex_options = c("striped","condensed",
                                   "HOLD_position","scale_down"),
                position = "center")
```

	Criterio1	Criterio2	Criterio3	Criterio4	Criterio5	Criterio6	Criterio7	Criterio8
Preferencias								
Pesos	0.25	0.3	0.15	0.05	0.05	0.1	0.05	0.05
Funciones Preferencias	V-shape (3)	Usual (1)	U-shape (2)	Level (4)	Linear (5)	Gaussian (6)	V-shape (3)	Level (4)
Q: Indiferencia	1000	0	4	500	3	0	1	1
P: Preferencia	2000	0	0	1000	5	0	2	3
S: Gausiano	0	0	0	0	0	1.5	0	0
Minimo	-2400	8500	8	1	512	6	-2.1	14
Estadísticas								
Maximo	-750	16500	32	8	1000	17	-1.3	16
Media	-1540	13060	17.6	3.6	707.2	10.6	-1.7	15.38
Desviacion Tipica	537.96	2598.92	7.84	2.42	239.07	3.77	0.28	0.74
A1	-2400	16500	32	8	1000	10	-2.1	16
Evaluaciones								
A2	-1550	12800	16	3	512	17	-1.5	15.3
A3	-1700	13500	16	2	512	12	-1.3	14
A4	-1300	14000	16	4	1000	8	-1.9	16
A5	-750	8500	8	1	512	6	-1.7	15.6

Tabla que muestra la ordenación de las alternativas: ei:

```
res01$tabAcciones %>%
  kable_styling(latex_options = c("striped","condensed",
                                   "HOLD_position","scale_down"),
                position = "center")
```

	Rango	Phi	Phi.mas	Phi.menos
A1	1	0.3462	0.5328	0.1866
A4	2	0.1757	0.3769	0.2012
A3	3	-0.0234	0.2859	0.3093
A2	4	-0.0642	0.2574	0.3216
A5	5	-0.4343	0.1334	0.5677

La ordenación de las alternativas es:

```
rownames(res$Acciones)
```

```
## [1] "A1" "A4" "A3" "A2" "A5"
```

Con Promethee Windows obtenemos la misma ordenación que en Promethee II