

# trabajo02\_Martín\_Fernández

Marta Martín Fernández

2025-10-28

Presentación (Presenta Gráficos; Motiva Problema);

ENLACES: <https://modelocurriculum.net/blog/trabajos-perfectos-para-compaginar-con-los-estudios>

TEMA : ELECCIÓN DEL MEJOR TIPO DE EMPLEO PARA COMPAGINAR CON LOS ESTUDIOS UNIVERSITARIOS

He elegido este tema porque es un problema que afecta a muchos estudiantes universitarios que buscan equilibrar sus estudios con un empleo que les permita cubrir sus gastos y adquirir experiencia laboral relevante. La elección del tipo de empleo adecuado puede influir significativamente en el rendimiento académico y en la calidad de vida del estudiante. Además, este tema es relevante en el contexto actual, donde muchos estudiantes enfrentan desafíos financieros y buscan oportunidades laborales que se adapten a sus horarios y necesidades. Al analizar las diferentes opciones de empleo, se pueden considerar factores como la flexibilidad horaria, la remuneración, la relevancia para la carrera académica y el ambiente laboral. Este análisis puede ayudar a los estudiantes a tomar decisiones informadas sobre qué tipo de empleo es más adecuado para ellos, permitiéndoles maximizar tanto su desarrollo académico como profesional.

Es un tema que me motiva debido a que ha sido una decisión que he tenido que tomar en el pasado, ya que necesitaba un empleo que pudiera compaginar con mi estudio en la universidad, y es una decisión que tienen que tomar mucho de los estudiantes universitarios año tras año.

## Definición del problema

El problema consiste en seleccionar el mejor tipo de empleo para un estudiante universitario que busca equilibrar sus estudios con un trabajo que le permita cubrir sus gastos y adquirir experiencia laboral relevante. Se deben considerar varios criterios y alternativas para tomar una decisión informada mediante el uso de las técnicas de toma de decisiones multicriterio (AHP, ELECTRE, PROMETHEE).

## Alternativas (Opciones de empleo)

Vamos a considerar 4 tipos de empleo comunes entre los estudiantes universitarios:

A1: EMPLEO EN HOSTELERÍA/SERVICIOS.

DESCRIPCIÓN: Camarero, repartidor o trabajador en un bar. Horarios fijos, a menudo suelen ser los fines de semana.

A2: PROFESOR PARTICULAR/TUTOR.

DESCRIPCIÓN: Dar clases particulares a otros estudiantes. Horarios flexibles, generalmente por las tardes o fines de semana cuando al estudiante le venga mejor por su organización.

A3: RRPP/PROMOTOR DE EVENTOS.

DESCRIPCIÓN: Trabajar en la promoción de eventos, productos o servicios, además de captar gente para el local en el que trabaja. Suelen ser trabajos con horarios nocturnos y fines de semana.

A4: DEPENDIENTE/A.

DESCRIPCIÓN: Es uno de los empleos más solicitados entre los estudiantes. Muchas tiendas de ropa, calzado y complementos ofrecen puestos de trabajo a media jornada, ideal para jóvenes estudiantes apasionados por la moda.

A5: CUIDADOR DE NIÑOS.

DESCRIPCIÓN: Fácil empleo para los estudiantes con horario flexible, que puede adaptarse a la organización del estudiante.

## Criterios y Subcriterios

Hemos definido cuatro criterios principales y un total de 8 subcriterios para evaluar las alternativas de empleo:

C1. Remuneración e impacto financiero

- Subcriterios:

C1.1. Salario por hora. Tenemos que maximizar, queremos ganar el mayor dinero posible. ESCALA: Valor en euros por hora.

C1.2. Beneficios adicionales. Tenemos que maximizar, ya que si ganamos algo más aparte del salario normal es beneficioso para nosotros. ESCALA: Escala del 1 (bajo) al 5 (alto) para reflejar descuentos, propinas...

OBJETIVO: Maximizar la remuneración total y los beneficios financieros asociados al empleo, ya que vamos a tener más dinero para ahorrar y cubrir nuestras necesidades.

C2. Impacto académico

- Subcriterios:

C2.1. Cercanía al hogar. Tenemos que maximizar, para que este trabajo no suponga desplazarnos y perder más tiempo, ya que siendo estudiantes no es algo que nos sobre. ESCALA: Escala del 1 (lejos) al 10 (muy cerca/teletrabajo).

C2.2. Carga mental/estrés. Tenemos que minimizar, para que el trabajo no sea una carga en nuestra vida. ESCALA: Escala del 1 (muy bajo) al 10 (muy alto).

OBJETIVO: Minimizar, ya que buscamos que el impacto negativo del empleo en el rendimiento académico sea mínimo, y que el trabajo no suponga un estrés adicional en nuestra vida que pueda llegar a interferir con los estudios. Queremos sobre todo el bienestar del estudiante.

C3. Relevancia para la Carrera Académica

- Subcriterios:

C3.1. Oportunidades de desarrollo profesional. Tenemos que maximizar, para que nos beneficie todo lo posible. ESCALA: Escala del 1 (bajo) al 10 (alto).

C3.2. Relación con los estudios. Tenemos que maximizar, para que también nos ayude en nuestra formación. ESCALA: Escala del 1 (bajo) al 10 (alto).

OBJETIVO: Maximizar, ya que buscamos que el empleo tenga una relación directa con la carrera académica del estudiante, proporcionando experiencia relevante y oportunidades de desarrollo profesional que puedan beneficiar su futuro laboral.

C4. Desarrollo personal

- Subcriterios:

C4.1. Satisfacción personal. Tenemos que maximizar, para que el trabajo nos haga sentir bien en nuestro día a día y cuando vayamos. ESCALA: Escala del 1 (bajo) al 10 (alto).

C.4.2. Relaciones con compañeros y jefes. Tenemos que maximizar, ya que es un aspecto importante siempre en un trabajo. Se trabaja mejor con personas con las que te lleves bien y tengas buena relación.  
ESCALA: Escala del 1 (bajo) al 10 (alto).

OBJETIVO: Maximizar, ya que buscamos que el empleo contribuya al crecimiento personal del estudiante, proporcionando un ambiente de trabajo positivo y oportunidades para desarrollar habilidades interpersonales y profesionales.

AHPconR (Estudio Inconsistencia; Matrices de comparación 2 a 2 están bien, no usar: 1.5,...);

AHPconPaquete (Estudio Inconsistencia; Comentarios a tablas; Matrices de comparación 2 a 2 están bien);

Electre (Comenta el paso a paso seguido; Qué pesos distintos a AHP; Un solo paso; Muchas salidas innecesarias);

## MÉTODO ELECTRE

Tenemos que maximizar los subcriterios C1.1, C1.2, C2.1, C3.1, C3.2, C4.1 y C4.2, y solo tenemos que minimizar el subcriterio C2.2.

Los pesos asignados a cada uno de estos subcriterios son:  $W=(0.25, 0.10, 0.15, 0.15, 0.05, 0.05, 0.15, 0.10)$ .

La matriz de decisión viene recogida en la siguiente tabla:

Table 1: Matriz de Decisión (Método ELECTRE)

	Salario	Beneficios	Cercanía	Estres	Desarrollo	Relacion_Estudio	Satisfaccion	Relaciones
Hostelería	8	2	3	7	4	1	6	8
Tutor	15	1	8	2	8	9	9	3
RRPP	10	5	5	8	5	2	7	10
Dependiente	9	1	7	6	5	3	5	7
Cuidador	12	3	7	5	3	5	8	5

Vamos a trabajar con un valor del umbral de concordancia (alpha) igual a 0.65, y un vector de umbrales de discordancia (d) igual a (6,4,5,4,6,4,5,5).

Ahora procedemos a aplicar el método ELECTRE para ordenar las alternativas.

```
source("teoriadecision_funciones_multicriterio.R")
source("teoriadecision_funciones_multicriterio_utiles.R")

metodo_electre = multicriterio.crea.matrizdecision(c(8,2,3,7,4,1,6,8,
15,1,8,2,8,9,9,3,
10,5,5,8,5,2,7,10,
9,1,7,6,5,3,5,7,
12,3,7,5,3,5,8,5),
numalternativas = 5,
numcriterios = 8)

metodo_electre

##      C1  C2  C3  C4  C5  C6  C7  C8
## a1   8   2   3   7   4   1   6   8
## a2  15   1   8   2   8   9   9   3
## a3   10   5   5   8   5   2   7  10
## a4    9   1   7   6   5   3   5   7
## a5   12   3   7   5   3   5   8   5
```

```

salida_electre= multicriterio.metodoELECTRE_I(metodo_electre, pesos.criterios=
c(0.25, 0.10 , 0.15, 0.15, 0.05, 0.05, 0.15 , 0.10), nivel.concordancia.minimo.alpha=0.65, no.se.compensa

salida_electre

## $datos
##          C1   C2   C3   C4   C5   C6   C7   C8
## a1      8.00 2.00 3.00 7.00 4.00 1.00 6.00 8.0
## a2     15.00 1.00 8.00 2.00 8.00 9.00 9.00 3.0
## a3     10.00 5.00 5.00 8.00 5.00 2.00 7.00 10.0
## a4      9.00 1.00 7.00 6.00 5.00 3.00 5.00 7.0
## a5     12.00 3.00 7.00 5.00 3.00 5.00 8.00 5.0
## pesos.criterios 0.25 0.1 0.15 0.15 0.05 0.05 0.15 0.1
## no.se.compensan 6.00 4.0 5.00 4.00 6.00 4.00 5.00 5.0
##
## $alpha
## [1] 0.65
##
## $Imas
## , , C1
##
##      a1   a2   a3   a4   a5
## a1 FALSE FALSE FALSE FALSE FALSE
## a2 TRUE FALSE TRUE  TRUE  TRUE
## a3 TRUE FALSE FALSE TRUE FALSE
## a4 TRUE FALSE FALSE FALSE FALSE
## a5 TRUE FALSE  TRUE  TRUE FALSE
##
## , , C2
##
##      a1   a2   a3   a4   a5
## a1 FALSE  TRUE FALSE  TRUE FALSE
## a2 FALSE FALSE FALSE FALSE FALSE
## a3 TRUE  TRUE FALSE  TRUE  TRUE
## a4 FALSE FALSE FALSE FALSE FALSE
## a5 TRUE  TRUE FALSE  TRUE FALSE
##
## , , C3
##
##      a1   a2   a3   a4   a5
## a1 FALSE FALSE FALSE FALSE FALSE
## a2 TRUE FALSE  TRUE  TRUE  TRUE
## a3 TRUE FALSE FALSE FALSE FALSE
## a4 TRUE FALSE  TRUE FALSE FALSE
## a5 TRUE FALSE  TRUE FALSE FALSE
##
## , , C4
##
##      a1   a2   a3   a4   a5
## a1 FALSE  TRUE FALSE  TRUE  TRUE
## a2 FALSE FALSE FALSE FALSE FALSE
## a3 TRUE  TRUE FALSE  TRUE  TRUE
## a4 FALSE  TRUE FALSE FALSE  TRUE
## a5 FALSE  TRUE FALSE FALSE FALSE

```

```

##
## , , C5
##
##      a1   a2   a3   a4   a5
## a1 FALSE FALSE FALSE FALSE  TRUE
## a2  TRUE FALSE  TRUE  TRUE  TRUE
## a3  TRUE FALSE FALSE FALSE  TRUE
## a4  TRUE FALSE FALSE FALSE  TRUE
## a5 FALSE FALSE FALSE FALSE FALSE
##
## , , C6
##
##      a1   a2   a3   a4   a5
## a1 FALSE FALSE FALSE FALSE FALSE
## a2  TRUE FALSE  TRUE  TRUE  TRUE
## a3  TRUE FALSE FALSE FALSE FALSE
## a4  TRUE FALSE  TRUE FALSE FALSE
## a5  TRUE FALSE  TRUE  TRUE FALSE
##
## , , C7
##
##      a1   a2   a3   a4   a5
## a1 FALSE FALSE FALSE  TRUE FALSE
## a2  TRUE FALSE  TRUE  TRUE  TRUE
## a3  TRUE FALSE FALSE  TRUE FALSE
## a4 FALSE FALSE FALSE FALSE FALSE
## a5  TRUE FALSE  TRUE  TRUE FALSE
##
## , , C8
##
##      a1   a2   a3   a4   a5
## a1 FALSE  TRUE FALSE  TRUE  TRUE
## a2 FALSE FALSE FALSE FALSE FALSE
## a3  TRUE  TRUE FALSE  TRUE  TRUE
## a4 FALSE  TRUE FALSE FALSE  TRUE
## a5 FALSE  TRUE FALSE FALSE FALSE
##
##
## $Iigual
## , , C1
##
##      a1   a2   a3   a4   a5
## a1  TRUE FALSE FALSE FALSE FALSE
## a2 FALSE  TRUE FALSE FALSE FALSE
## a3 FALSE FALSE  TRUE FALSE FALSE
## a4 FALSE FALSE FALSE  TRUE FALSE
## a5 FALSE FALSE FALSE FALSE  TRUE
##
## , , C2
##
##      a1   a2   a3   a4   a5
## a1  TRUE FALSE FALSE FALSE FALSE
## a2 FALSE  TRUE FALSE  TRUE FALSE
## a3 FALSE FALSE  TRUE FALSE FALSE

```

```

## a4 FALSE TRUE FALSE TRUE FALSE
## a5 FALSE FALSE FALSE FALSE TRUE
##
## , , C3
##
##      a1   a2   a3   a4   a5
## a1  TRUE FALSE FALSE FALSE FALSE
## a2 FALSE  TRUE FALSE FALSE FALSE
## a3 FALSE FALSE  TRUE FALSE FALSE
## a4 FALSE FALSE FALSE  TRUE  TRUE
## a5 FALSE FALSE FALSE  TRUE  TRUE
##
## , , C4
##
##      a1   a2   a3   a4   a5
## a1  TRUE FALSE FALSE FALSE FALSE
## a2 FALSE  TRUE FALSE FALSE FALSE
## a3 FALSE FALSE  TRUE FALSE FALSE
## a4 FALSE FALSE FALSE  TRUE FALSE
## a5 FALSE FALSE FALSE FALSE  TRUE
##
## , , C5
##
##      a1   a2   a3   a4   a5
## a1  TRUE FALSE FALSE FALSE FALSE
## a2 FALSE  TRUE FALSE FALSE FALSE
## a3 FALSE FALSE  TRUE  TRUE FALSE
## a4 FALSE FALSE  TRUE  TRUE FALSE
## a5 FALSE FALSE FALSE FALSE  TRUE
##
## , , C6
##
##      a1   a2   a3   a4   a5
## a1  TRUE FALSE FALSE FALSE FALSE
## a2 FALSE  TRUE FALSE FALSE FALSE
## a3 FALSE FALSE  TRUE FALSE FALSE
## a4 FALSE FALSE FALSE  TRUE FALSE
## a5 FALSE FALSE FALSE FALSE  TRUE
##
## , , C7
##
##      a1   a2   a3   a4   a5
## a1  TRUE FALSE FALSE FALSE FALSE
## a2 FALSE  TRUE FALSE FALSE FALSE
## a3 FALSE FALSE  TRUE FALSE FALSE
## a4 FALSE FALSE FALSE  TRUE FALSE
## a5 FALSE FALSE FALSE FALSE  TRUE
##
## , , C8
##
##      a1   a2   a3   a4   a5
## a1  TRUE FALSE FALSE FALSE FALSE
## a2 FALSE  TRUE FALSE FALSE FALSE
## a3 FALSE FALSE  TRUE FALSE FALSE

```

```

## a4 FALSE FALSE FALSE  TRUE FALSE
## a5 FALSE FALSE FALSE FALSE  TRUE
##
## $Imenos
## , , C1
##
##      a1    a2    a3    a4    a5
## a1 FALSE  TRUE  TRUE  TRUE  TRUE
## a2 FALSE FALSE FALSE FALSE FALSE
## a3 FALSE  TRUE FALSE FALSE  TRUE
## a4 FALSE  TRUE  TRUE FALSE  TRUE
## a5 FALSE  TRUE FALSE FALSE FALSE
##
## , , C2
##
##      a1    a2    a3    a4    a5
## a1 FALSE FALSE  TRUE FALSE  TRUE
## a2  TRUE FALSE  TRUE FALSE  TRUE
## a3 FALSE FALSE FALSE FALSE FALSE
## a4  TRUE FALSE  TRUE FALSE  TRUE
## a5 FALSE FALSE  TRUE FALSE FALSE
##
## , , C3
##
##      a1    a2    a3    a4    a5
## a1 FALSE  TRUE  TRUE  TRUE  TRUE
## a2 FALSE FALSE FALSE FALSE FALSE
## a3 FALSE  TRUE FALSE  TRUE  TRUE
## a4 FALSE  TRUE FALSE FALSE FALSE
## a5 FALSE  TRUE FALSE FALSE FALSE
##
## , , C4
##
##      a1    a2    a3    a4    a5
## a1 FALSE FALSE  TRUE FALSE FALSE
## a2  TRUE FALSE  TRUE  TRUE  TRUE
## a3 FALSE FALSE FALSE FALSE FALSE
## a4  TRUE FALSE  TRUE FALSE FALSE
## a5  TRUE FALSE  TRUE  TRUE FALSE
##
## , , C5
##
##      a1    a2    a3    a4    a5
## a1 FALSE  TRUE  TRUE  TRUE FALSE
## a2 FALSE FALSE FALSE FALSE FALSE
## a3 FALSE  TRUE FALSE FALSE FALSE
## a4 FALSE  TRUE FALSE FALSE FALSE
## a5  TRUE  TRUE  TRUE  TRUE FALSE
##
## , , C6
##
##      a1    a2    a3    a4    a5
## a1 FALSE  TRUE  TRUE  TRUE  TRUE

```

```

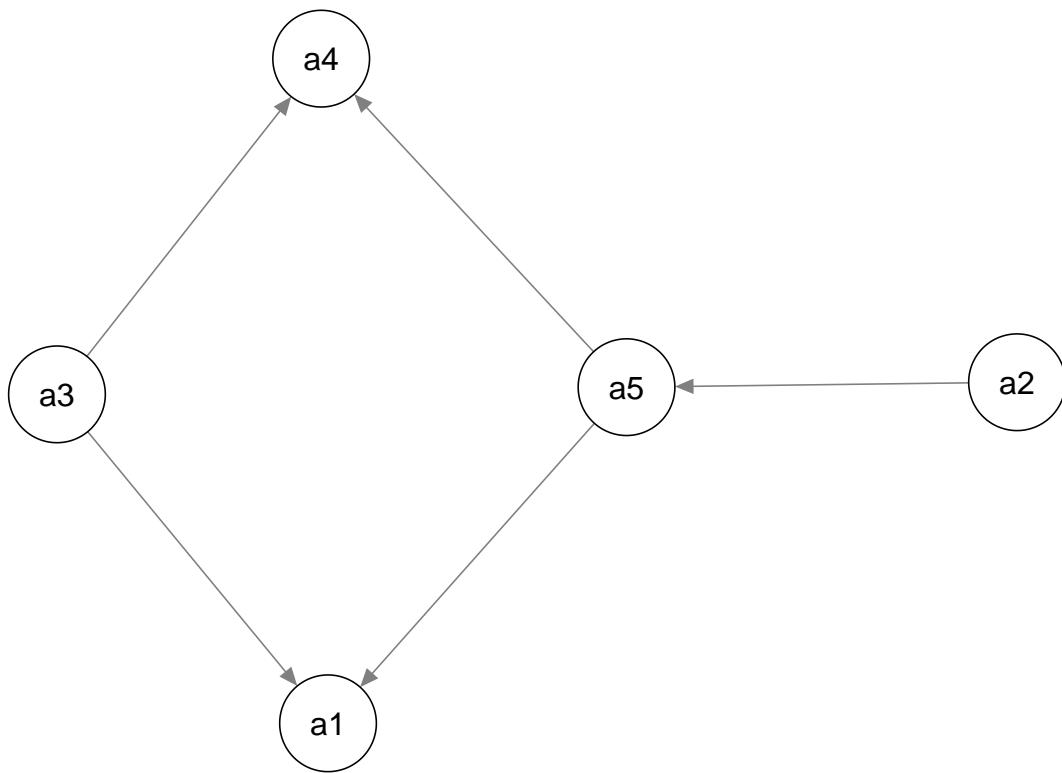
## a2 FALSE FALSE FALSE FALSE FALSE
## a3 FALSE TRUE FALSE TRUE TRUE
## a4 FALSE TRUE FALSE FALSE TRUE
## a5 FALSE TRUE FALSE FALSE FALSE
##
## , , C7
##
##      a1    a2    a3    a4    a5
## a1 FALSE TRUE TRUE FALSE TRUE
## a2 FALSE FALSE FALSE FALSE FALSE
## a3 FALSE TRUE FALSE FALSE TRUE
## a4 TRUE TRUE TRUE FALSE TRUE
## a5 FALSE TRUE FALSE FALSE FALSE
##
## , , C8
##
##      a1    a2    a3    a4    a5
## a1 FALSE FALSE TRUE FALSE FALSE
## a2 TRUE FALSE TRUE TRUE TRUE
## a3 FALSE FALSE FALSE FALSE FALSE
## a4 TRUE FALSE TRUE FALSE FALSE
## a5 TRUE FALSE TRUE TRUE FALSE
##
## $ind.concordancia
##      a1    a2    a3    a4    a5
## a1 1.00 0.35 0.00 0.50 0.30
## a2 0.65 1.00 0.65 0.75 0.65
## a3 1.00 0.35 1.00 0.80 0.40
## a4 0.50 0.35 0.25 1.00 0.45
## a5 0.70 0.35 0.60 0.70 1.00
##
## $ind.concordancia.gorro
##      a1        a2        a3        a4        a5
## a1      NaN 0.5384615 0.0000000 1.000000 0.4285714
## a2 1.857143      NaN 1.8571429 2.600000 1.8571429
## a3      Inf 0.5384615      NaN 3.750000 0.6666667
## a4 1.000000 0.3846154 0.2666667      NaN 0.5454545
## a5 2.333333 0.5384615 1.5000000 1.833333      NaN
##
## $test.concordancia
##      a1    a2    a3    a4    a5
## a1 FALSE FALSE FALSE FALSE FALSE
## a2 TRUE FALSE TRUE TRUE TRUE
## a3 TRUE FALSE FALSE TRUE FALSE
## a4 FALSE FALSE FALSE FALSE FALSE
## a5 TRUE FALSE FALSE TRUE FALSE
##
## $test.discordancia
##      a1    a2    a3    a4    a5
## a1 TRUE FALSE TRUE TRUE FALSE
## a2 FALSE TRUE FALSE FALSE TRUE
## a3 TRUE FALSE TRUE TRUE TRUE
## a4 TRUE FALSE FALSE TRUE TRUE

```

```

## a5  TRUE FALSE FALSE  TRUE  TRUE
##
## $relacion.dominante
##      a1   a2   a3   a4   a5
## a1 FALSE FALSE FALSE FALSE FALSE
## a2 FALSE FALSE FALSE FALSE  TRUE
## a3  TRUE FALSE FALSE  TRUE FALSE
## a4 FALSE FALSE FALSE FALSE FALSE
## a5  TRUE FALSE FALSE  TRUE FALSE
##
## $nucleo_aprox
## a2 a3
## 2 3
qgraph::qgraph(salida_electre$relacion.dominante)

```



En este grafo de dominancia se representa visualmente la relación de superación obtenida al aplicar los test de concordancia y discordancia. Los resultados obtenidos son:

A2 domina a a5. A5 domina a las alternativas A1 y A4. A3 domina a las alternativas A4 y A1.

```
salida_electre$nucleo_aprox
```

```

## a2 a3
## 2 3

```

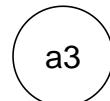
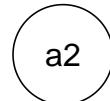
Este es el núcleo aproximado, que es el conjunto de soluciones que no son dominadas por otras alternativas. El resultado es A2 y A3. Esto nos indica que las alternativas “Profesor particular/Tutor” y “RRPP/Promotor de eventos” son las más adecuadas, ya que ninguna de las otras opciones logró superarlas tras la aplicación de los subcriterios y umbrales.

Para intentar quedarnos con una única alternativa óptima, vamos a reducir el grafo a las alternativas en el

núcleo aproximado. Aplicamos el método ELECTRE de nuevo para ver si una alternativa domina a la otra en este subconjunto.

```
salida_electre_b= multicriterio.metodoELECTRE_I(metodo_electre, pesos.criterios=
c(0.25, 0.10 , 0.15, 0.15, 0.05, 0.05, 0.15 , 0.10), nivel.concordancia.minimo.alpha=0.65, no.se.compensa)

qgraph::qgraph(salida_electre_b$relacion.dominante)
```



El grafo nos indica que no hay dominancia de una alternativa sobre la otra, ya que no aparece ninguna flecha.

```
salida_electre_b$nucleo_aprox
```

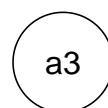
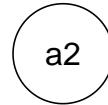
```
## a2 a3
## 1 2
```

Después de reducir el análisis al subconjunto A2,A3, seguimos teniendo que el núcleo aproximado final sigue siendo A2,A3. Los números 1 y 2 solo indican el orden de estas alternativas en el subconjunto.

Ahora vamos a probar a reducir el umbral alpha para ver si alguna alternativa domina a la otra e intentar resolver la incomparabilidad. Si ajustamos los umbrales podemos intentar forzar esta dominancia para seleccionar una única alternativa óptima.

```
salida_electre_c= multicriterio.metodoELECTRE_I(metodo_electre, pesos.criterios=
c(0.25, 0.10 , 0.15, 0.15, 0.05, 0.05, 0.15 , 0.10), nivel.concordancia.minimo.alpha=0.10, no.se.compensa)

qgraph::qgraph(salida_electre_c$relacion.dominante)
```



Probando a reducir el valor de alpha numerosas veces hasta llegar a un límite extremadamente permisivo (0.10), sigue sin aparecer dominancia en el grafo. Esto indica que necesitamos modificar también los valores del vector de discordancia.

```
salida_electre_d= multicriterio.metodoELECTRE_I(metodo_electre, pesos.criterios=
c(0.25, 0.10 , 0.15, 0.15, 0.05, 0.05, 0.15 , 0.10), nivel.concordancia.minimo.alpha=0.65, no.se.compensa)

qgraph::qgraph(salida_electre_d$relacion.dominante)
```



Tras probar una serie de valores para el vector de discordancia, se obtiene que A2 domina a A3 con  $d=(6,5,5,7,6,4,5,8)$ , que es el vector de discordancia mínimamente ajustado. Tras observar esta solución, podemos decir que la alternativa A2 domina a la alternativa A3, por lo que la mejor opción es “Profesor particular/Tutor”.

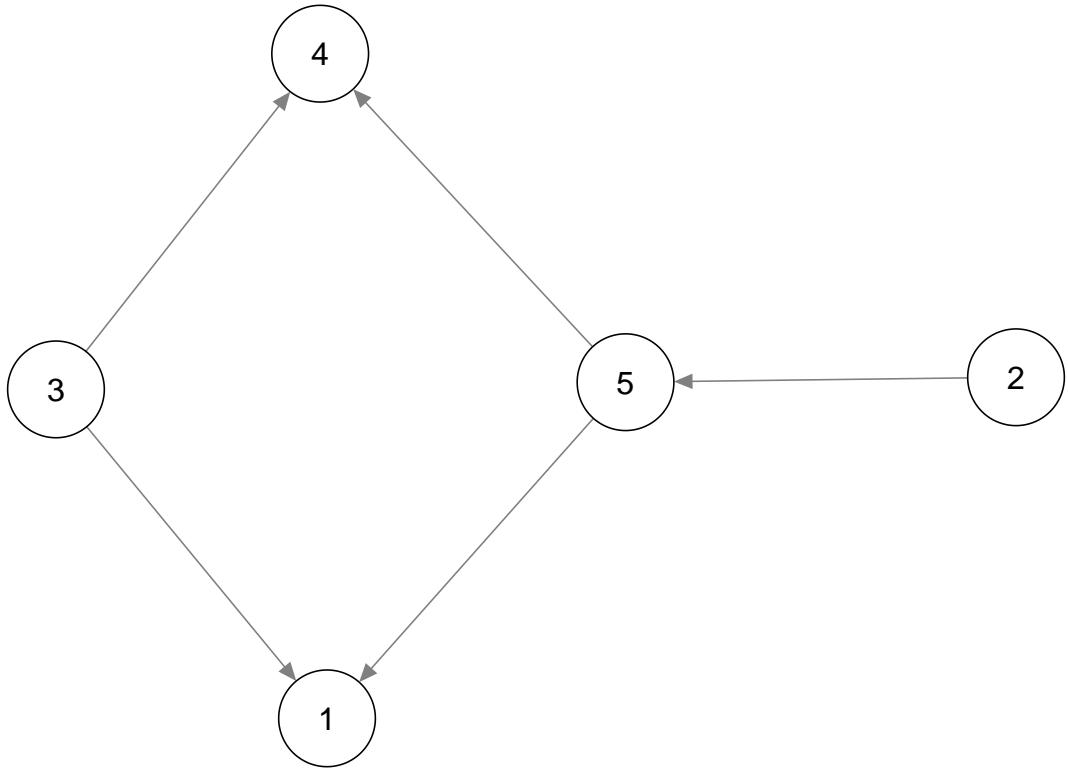
## CÁLCULOS EN EL MÉTODO ELECTRE I

```

e1 = func_ELECTRE_Completo(salida_electre)
e1$Grafo

## De A
## 1 2 5
## 2 3 1
## 3 3 4
## 4 5 1
## 5 5 4
qgraph::qgraph(e1$Grafo)

```



e1\$Nucleo

```
## a2 a3
## 2 3
```

Con el método ELECTRE I nos salen los mismos resultados que con el método ELECTRE anterior. El núcleo sigue estando compuesto por las alternativas 2 y 3, y siguen existiendo las mismas dominancias entre las alternativas, como vemos con el código e1\$Grafo.

Promethee (Hace Promethee R y Windows; Qué Pesos distintos a AHP; Explicaciones); Conclusiones

## MÉTODO PROMETHEE

Tenemos que maximizar los subcriterios C1.1, C1.2, C2.1, C3.1, C3.2, C4.1 y C4.2, y solo tenemos que minimizar el subcriterio C2.2. Para crear la matriz para este método, multiplicamos la columna del subcriterio 2.2 por -1.

```
source("teoriadecision_funciones_multicriterio.R")
source("teoriadecision_funciones_multicriterio_utiles.R")

tabdecision_promethee = multicriterio.crea.matrizdecision(c(8,2,3,-7,4,1,6,8,15,1,8,-2,8,9,9,3,10,5,5,-1))

tabdecision_promethee
```

	Salario	Beneficios	Cercanía	Estrés	Desarrollo	Relación_Estudios
## Hostelería	8	2	3	-7	4	1
## Tutor	15	1	8	-2	8	9
## RRPP	10	5	5	-8	5	2
## Dependiente	9	1	7	-6	5	3
## Cuidador	12	3	7	-5	3	5
##	Satisfacción	Relaciones				

```

## Hostelería      6      8
## Tutor           9      3
## RRPP            7     10
## Dependiente    5      7
## Cuidador        8      5

```

Vamos a asignar los siguientes pesos a los criterios:  $W=(0.20, 0.15, 0.20, 0.15, 0.10, 0.05, 0.10, 0.05)$ .

A continuación se define la matriz de funciones de preferencia denominada tab.fpref en el código R, esencial para aplicar este método PROMETHEE. Esta matriz indica el tipo de función, el umbral de indiferencia ( $q$ ), el umbral de preferencia ( $p$ ) y s.

```
pesos.criterios=c(0.20, 0.15, 0.20, 0.15, 0.10, 0.05, 0.10, 0.05)
```

```
tab.fpref = matrix(c(2,3,1,0,1,0,1,0,3,0,5,0,5,2,5,0,1,0,1,0,4,3,6,0,6,0,1,2,3,0,4,0), ncol=4, byrow=TRUE)
```

```
tab.fpref
```

```

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

```

## REPRESENTACIÓN GRÁFICA DE LAS FUNCIONES DE PREFERENCIA

```

fpref.criterio_usual_di <- function(di) {
  # di = vaj - vah
  if (di <= 0) {
    res = 0
  } else {
    res = 1
  }
  return(res)
}

fpref.cuasi_criterio_di <- function(di, qi) {
  # di = vaj - vah
  if (di <= qi) {
    res = 0
  } else {
    res = 1
  }
  return(res)
}

fpref.criterio_preflineal_di <- function(di, pi) {
  # di = vaj - vah
  if (di <= 0) {
    res = 0
  } else if (di > pi) {

```

```

        res = 1
    } else {
        res = di / pi
    }
    return(res)
}

fpref.criterio_nivel_di <- function(di, qi, pi) {
    # di = vaj - vah
    if (di <= qi) {
        res = 0
    } else if (di > pi) {
        res = 1
    } else {
        res = 0.5
    }
    return(res)
}

fpref.criterio_preflineal_indif_di <- function(di, qi, pi) {
    # di = vaj - vah
    if (di <= qi) {
        res = 0
    } else if (di > pi) {
        res = 1
    } else {
        res = (di - qi) / (pi - qi)
    }
    return(res)
}

fpref.criterio_gaussiano_di <- function(di, qi, pi, si) {
    # di = vaj - vah.
    if (di <= 0) {
        res = 0
    } else {
        res = 1 - exp(-(di^2) / (2 * si^2))
    }
    return(res)
}

x = seq(-10,10,length.out=100)

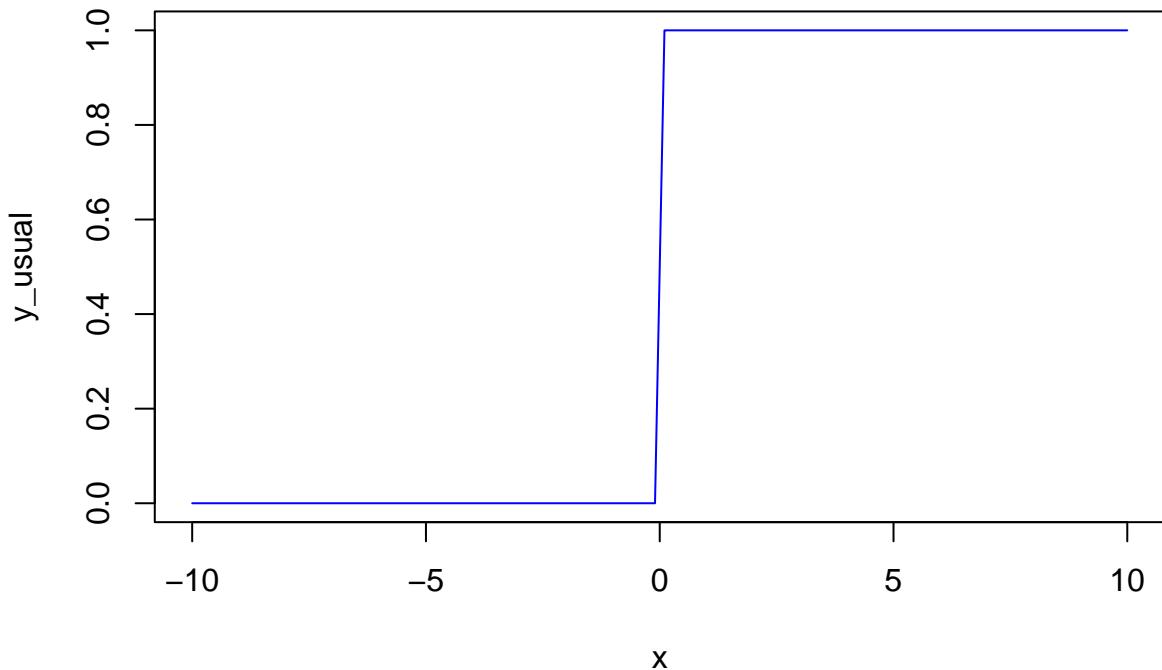
# GRÁFICA DEL CRITERIO USUAL

y_usual = sapply(x,fpref.criterio_usual_di)

plot(x,y_usual,type="l",col="blue",main="Criterio usual")

```

### Criterio usual

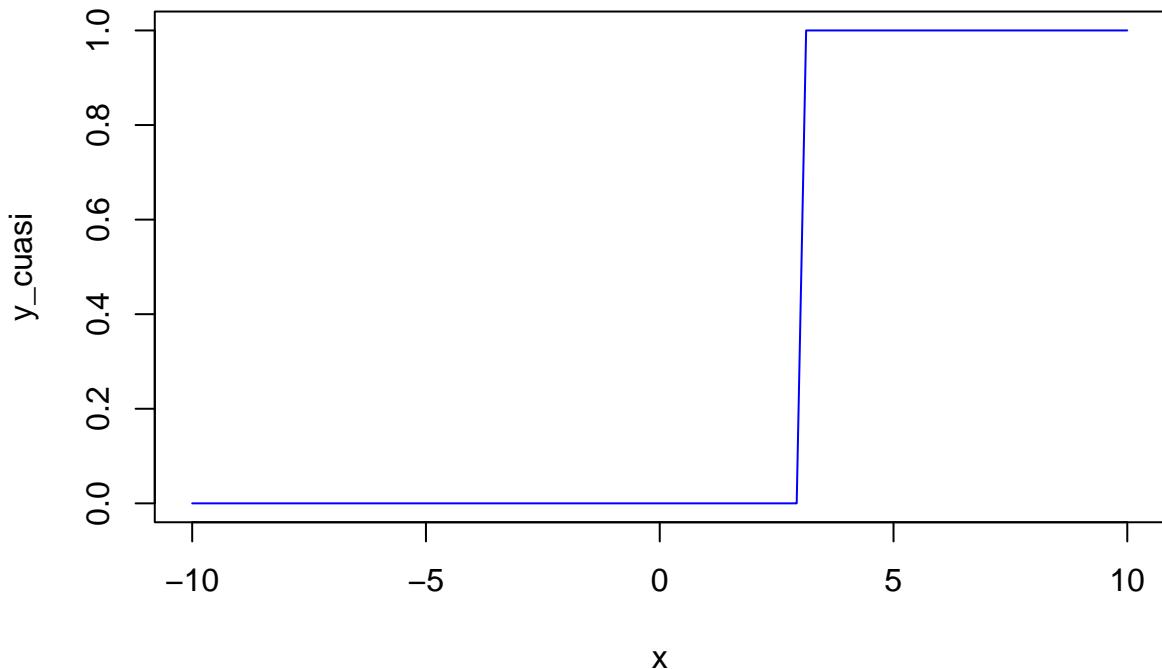


```
# GRÁFICA DEL CRITERIO CUASI CON Q=3
```

```
y_cuasi <- sapply(x, function(xx) fpref.cuasi_criterio_di(xx, qi = 3))

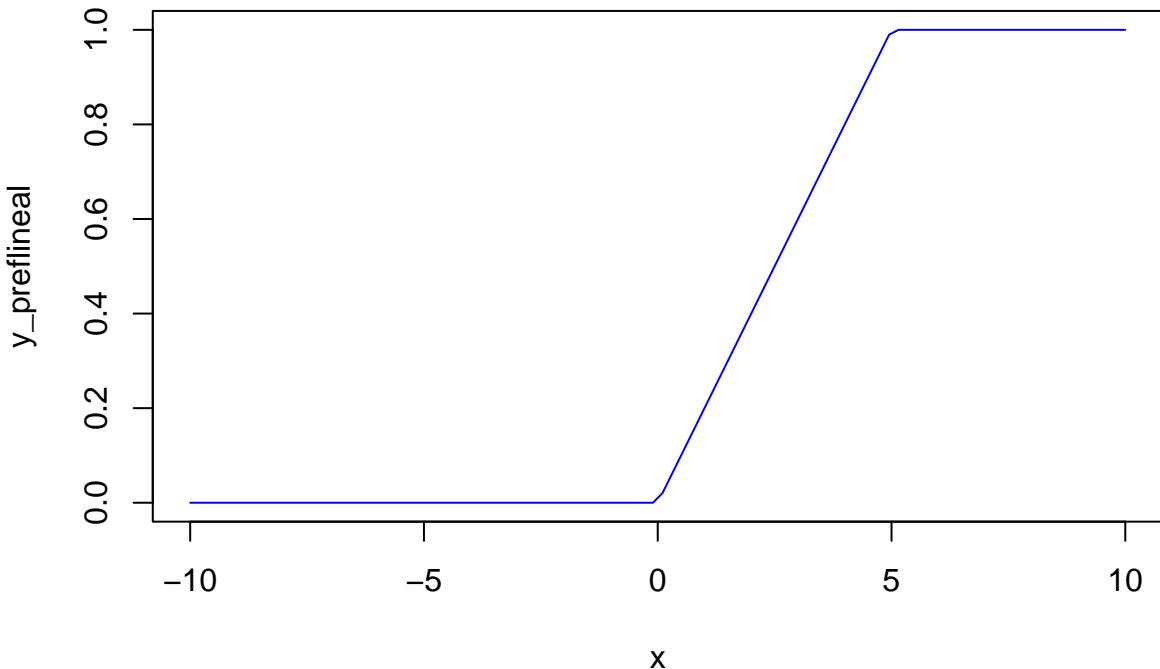
plot(x,y_cuasi,type="l",col="blue",main="Criterio cuasi con Q=3")
```

### Criterio cuasi con Q=3



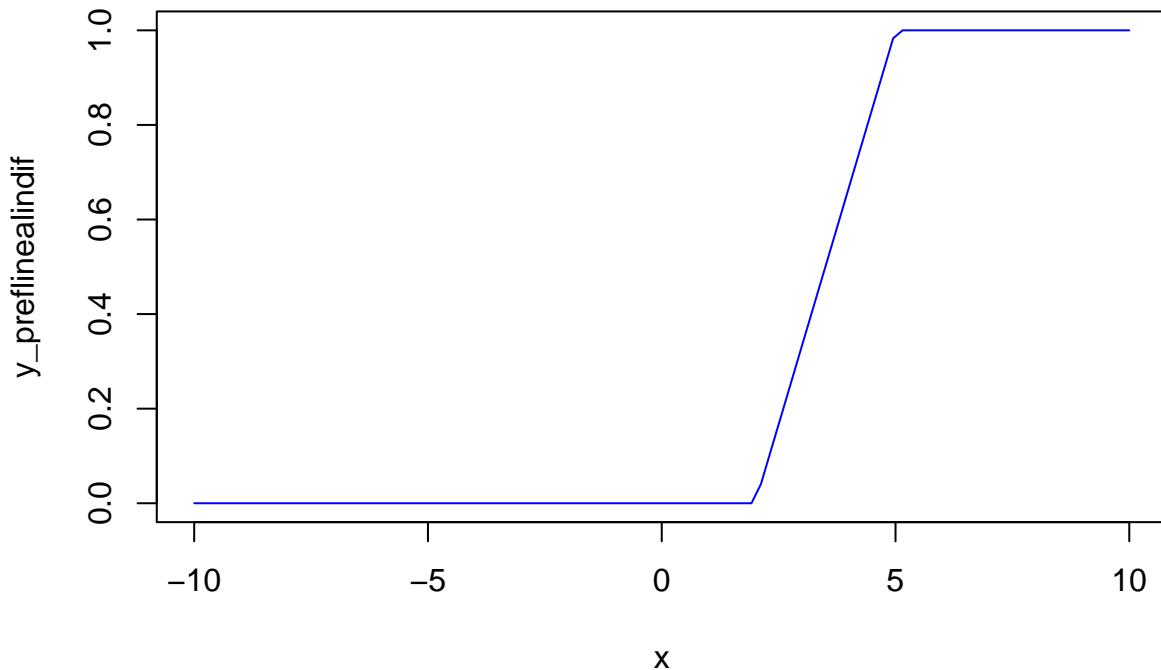
```
# Usamos q=3 porque consideramos que una diferencia en el salario de 1 o 2 euros no es suficiente, pero
# GRÁFICA DEL CRITERIO DE PREFERENCIA LINEAL CON P=5
y_preflineal <- sapply(x, function(xx) fpref.criterio_preflineal_di(xx, pi = 5))
plot(x,y_preflineal,type="l",col="blue",main="Criterio preferencia lineal con P=5")
```

**Criterio preferencia lineal con P=5**



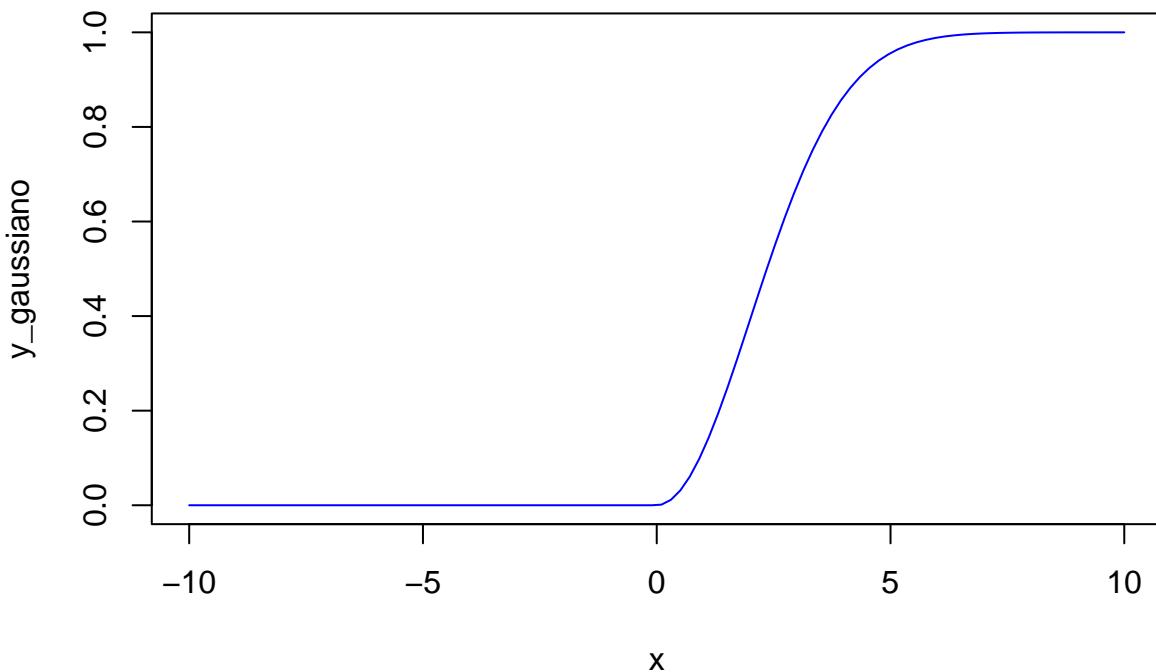
```
# 5 es la diferencia necesaria para considerar que un empleo es preferible a otro en términos de cercanía
# GRÁFICA DEL CRITERIO DE PREFERENCIA LINEAL INDIFERENTE CON Q=2,P=5
y_preflinealindif <- sapply(x, function(xx) fpref.criterio_preflineal_indif_di(xx, qi=2, pi = 5))
plot(x,y_preflinealindif,type="l",col="blue",main="Criterio preferencia lineal indiferente con Q=2,P=5")
```

## Criterio preferencia lineal indiferente con Q=2,P=5



```
# Una diferencia de hasta 2 puntos de carga mental/estrés en un empleo se considera indiferente. Una di  
# GRÁFICA DEL CRITERIO GAUSSIANO  
y_gaussiano <- sapply(x, function(xx) fpref.criterio_gaussiano_di(xx, qi= 0, pi = 1, si=2))  
plot(x,y_gaussiano,type="l",col="blue",main="Criterio gaussiano")
```

## Criterio gaussiano



### APLICACIÓN DEL MÉTODO PROMETHEE I

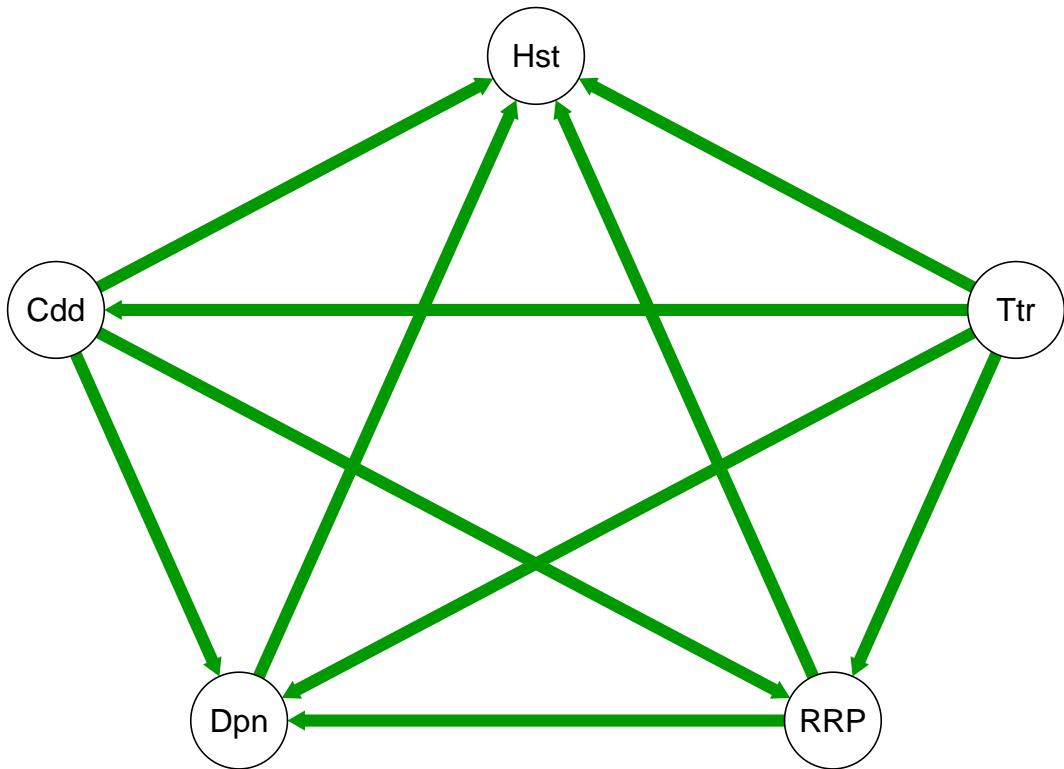
```
tab.promethee_i = multicriterio.metodo.promethee_i(tabdecision_promethee, pesos.criterios,tab.fpref)

tab.promethee_i

## $tabla.indices
##           Hostelería Tutor         RRPP Dependiente Cuidador
## Hostelería  0.0000000 0.200 0.0000000  0.1742503 0.1375000
## Tutor       0.7675348 0.000 0.6593469  0.5764665 0.2267503
## RRPP        0.3667503 0.200 0.0000000  0.2268469 0.3000000
## Dependiente 0.2600000 0.050 0.0800000  0.0000000 0.1250000
## Cuidador    0.5743469 0.175 0.1417503  0.2175348 0.0000000
##
## $vflujos.ent
##   Hostelería      Tutor         RRPP Dependiente Cuidador
##   0.5117503  2.2300985  1.0935972  0.5150000  1.1086320
##
## $vflujos.sal
##   Hostelería      Tutor         RRPP Dependiente Cuidador
##   1.9686320  0.6250000  0.8810972  1.1950985  0.7892503
##
## $tablarelacionsuper
##           Hostelería Tutor RRPP Dependiente Cuidador
## Hostelería      0.5  0.0  0.0      0.0  0.0
## Tutor          1.0  0.5  1.0      1.0  1.0
## RRPP           1.0  0.0  0.5      1.0  0.0
## Dependiente    1.0  0.0  0.0      0.5  0.0
## Cuidador        1.0  0.0  1.0      1.0  0.5
```

```
qgraph::qgraph(tab.promethee_i$tablarelacionsuper)
```

```
## Warning in abbreviate(colnames(input), 3): abreviatura utilizada con caracteres
## no ASCII
```



```
tab.promethee_i$tablarelacionsuper
```

	Hostelería	Tutor	RRPP	Dependiente	Cuidador
Hostelería	0.5	0.0	0.0	0.0	0.0
Tutor	1.0	0.5	1.0	1.0	1.0
RRPP	1.0	0.0	0.5	1.0	0.0
Dependiente	1.0	0.0	0.0	0.5	0.0
Cuidador	1.0	0.0	1.0	1.0	0.5

En el grafo se representa esta tabla de la relación que hay entre las alternativas. Donde aparezca un 1, significará que la alternativa de esa fila supera a la alternativa de esta columna. Por ejemplo, de la fila de “RRPP” sacamos que supera a “Hostelería” y “Dependiente”, por tanto, en la fila de “Hostelería” y “Dependiente” debería de aparecer un 0 en la columna de “RRPP”, como vemos que ocurre. Así vamos conociendo estas relaciones.

En el grafo, al igual que ocurre en la tabla anterior, vemos que todas las alternativas son superadas por al menos otra alternativa, menos “Profesor particular/Tutor”, que no es superada por ninguna. El método Promethee I nos dice que la mejor alternativa es esta.

## APLICACIÓN DEL MÉTODO PROMETHEE II

```
tab.promethee_ii = multicriterio.metodo.promethee_ii(tabdecision_promethee, pesos.criterios, tab.fpref)

tab.promethee_ii

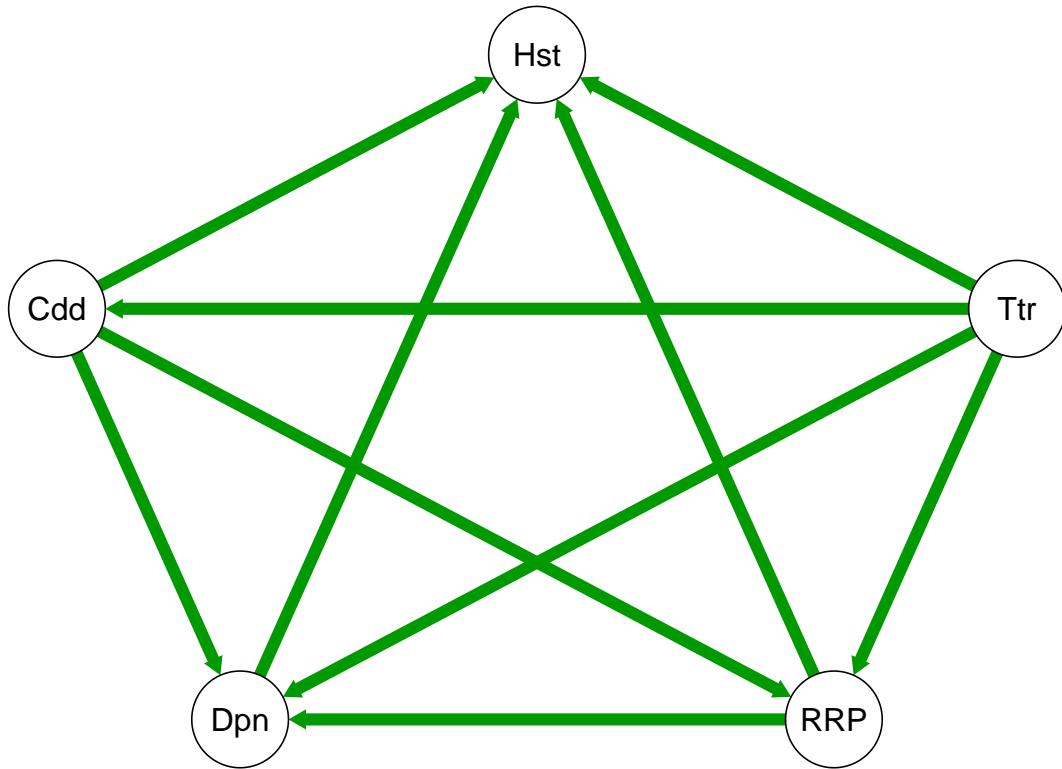
## $tabla.indices
```

```

##          Hostelería Tutor      RRPP Dependiente Cuidador
## Hostelería  0.0000000 0.200 0.0000000  0.1742503 0.1375000
## Tutor       0.7675348 0.000 0.6593469  0.5764665 0.2267503
## RRPP        0.3667503 0.200 0.0000000  0.2268469 0.3000000
## Dependiente 0.2600000 0.050 0.0800000  0.0000000 0.1250000
## Cuidador    0.5743469 0.175 0.1417503  0.2175348 0.0000000
##
## $vflujos.netos
##   Hostelería      Tutor      RRPP Dependiente Cuidador
## -1.4568817  1.6050985  0.2125000 -0.6800985  0.3193817
##
## $tablarelacionsuper
##          Hostelería Tutor RRPP Dependiente Cuidador
## Hostelería      0.5  0.0  0.0      0.0  0.0
## Tutor          1.0  0.5  1.0      1.0  1.0
## RRPP           1.0  0.0  0.5      1.0  0.0
## Dependiente    1.0  0.0  0.0      0.5  0.0
## Cuidador       1.0  0.0  1.0      1.0  0.5
qgraph::qgraph(tab.promethee_ii$tablarelacionsuper)

## Warning in abbreviate(colnames(input), 3): abreviatura utilizada con caracteres
## no ASCII

```



## ORDENACIÓN FINAL DE LAS ALTERNATIVAS POR EL MÉTODO DE PROMETHEE II

```
order(tab.promethee_ii$vflujos.netos, decreasing = TRUE)
```

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

La mejor alternativa vuelve a ser la de “Profesor particular/Tutor”. Después le sigue “Cuidador de niños”,

“RRPP/Promotor de eventos”, “Dependiente”, y por último como peor opción, “Empleo en hostelería/Servicios”.

## MÉTODO PROMETHEE WINDOWS

He utilizado el software PROMETHEE Windows para corroborar los resultados obtenidos con R. He introducido los mismos datos de la matriz de decisión, los pesos de los criterios y las funciones de preferencia definidas anteriormente.

```

source("teoriadecision_funciones_multicriterio.R")
source("teoriadecision_funciones_multicriterio_utiles.R")
source("teoriadecision_funciones_multicriterio_diagram.R")

## Cargando paquete requerido: shape
tabdecision_promethee = multicriterio.crea.matrizdecision(c(8,2,3,-7,4,1,6,8,15,1,8,-2,8,9,9,3,10,5,5,-

pesos.criterios01=c(0.20, 0.15 , 0.20, 0.15, 0.10, 0.05, 0.10 , 0.05)

tab.fpref01 = matrix(c(2,3,1,0,1,0,1,0,3,0,5,0,5,2,5,0,1,0,1,0,4,3,6,0,6,0,1,2,3,0,4,0), ncol=4, byrow=1)

res = multicriterio.metodo.promethee_windows(tabdecision_promethee, tab.fpref01, pesos.criterios01)

res

## $Escenario
##                               Criterio1    Criterio2    Criterio3    Criterio4
## Pesos                      "0.2"      "0.15"      "0.2"       "0.15"
## Funciones Preferencias "U-shape (2)" "Usual (1)" "V-shape (3)" "Linear (5)"
## Q: Indiferencia            "3"        "0"         "0"         "2"
## P: Preferencia             "1"        "1"         "5"         "5"
## S: Gausiano                "0"        "0"         "0"         "0"
## Minimo                     "8"        "1"         "3"         "-8"
## Maximo                     "15"       "5"         "8"         "-2"
## Media                       "10.8"     "2.4"       "6"         "-5.6"
## Desviacion Tipica          "2.48"     "1.5"       "1.79"      "2.06"
## Hostelería                 "8"        "2"         "3"         "-7"
## Tutor                       "15"       "1"         "8"         "-2"
## RRPP                        "10"       "5"         "5"         "-8"
## Dependiente                "9"        "1"         "7"         "-6"
## Cuidador                    "12"       "3"         "7"         "-5"
##                               Criterio5    Criterio6    Criterio7    Criterio8
## Pesos                      "0.1"      "0.05"      "0.1"       "0.05"
## Funciones Preferencias "Usual (1)" "Level (4)" "Gaussian (6)" "V-shape (3)"
## Q: Indiferencia            "0"        "3"         "0"         "0"
## P: Preferencia             "1"        "6"         "1"         "4"
## S: Gausiano                "0"        "0"         "2"         "0"
## Minimo                     "3"        "1"         "5"         "3"
## Maximo                     "8"        "9"         "9"         "10"
## Media                       "5"        "4"         "7"         "6.6"
## Desviacion Tipica          "1.67"     "2.83"     "1.41"      "2.42"
## Hostelería                 "4"        "1"         "6"         "8"
## Tutor                       "8"        "9"         "9"         "3"
## RRPP                        "5"        "2"         "7"         "10"
## Dependiente                "5"        "3"         "5"         "7"
## Cuidador                    "3"        "5"         "8"         "5"

```

```

## $Acciones
## Rango      Phi Phi.mas Phi.menos
## Tutor      1  0.4013  0.5575   0.1562
## Cuidador   2  0.0798  0.2772   0.1973
## RRPP       3  0.0531  0.2734   0.2203
## Dependiente 4 -0.1700 0.1288   0.2988
## Hostelería 5 -0.3642 0.1279   0.4922

```

Fijándonos en los valores de Phi, tenemos el orden de preferencia de las alternativas. “Tutor” es la alternativa con el valor más alto, un 0.4013, seguida de “Cuidador”, “RRPP” y “Dependiente”. Por último, la peor opción es la de “Hostelería”, que tiene un valor de Phi de -0.3642.

Datos introducidos en el problema resuelto con Promethee:

```

res02 = multicriterio.metodo.promethee_windows_kableExtra(res)

res02$tabEscenario

```

	Criterio1	Criterio2	Criterio3	Criterio4	Criterio5	Criterio6	Criterio7	Criterio8
<b>Preferencias</b>								
Pesos	0.2	0.15	0.2	0.15	0.1	0.05	0.1	0.05
Funciones Preferencias	U-shape (2)	Usual (1)	V-shape (3)	Linear (5)	Usual (1)	Level (4)	Gaussian (6)	Value (V)
Q: Indiferencia	3	0	0	2	0	3	0	0
P: Preferencia	1	1	5	5	1	6	1	4
S: Gausiano	0	0	0	0	0	0	2	0
Mínimo	8	1	3	-8	3	1	5	3
<b>Estadísticas</b>								
Maximo	15	5	8	-2	8	9	9	10
Media	10.8	2.4	6	-5.6	5	4	7	6.2
Desviación Tipica	2.48	1.5	1.79	2.06	1.67	2.83	1.41	2.2
Hostelería	8	2	3	-7	4	1	6	8
<b>Evaluaciones</b>								
Tutor	15	1	8	-2	8	9	9	3
RRPP	10	5	5	-8	5	2	7	10
Dependiente	9	1	7	-6	5	3	5	7
Cuidador	12	3	7	-5	3	5	8	5

Ordenación de las alternativas:

```

res02$tabAcciones

```

Rango	Phi	Phi.mas	Phi.menos
Tutor	1  0.4013	0.5575	0.1562
Cuidador	2  0.0798	0.2772	0.1973
RRPP	3  0.0531	0.2734	0.2203
Dependiente	4 -0.1700	0.1288	0.2988
Hostelería	5 -0.3642	0.1279	0.4922

En esta tabla volvemos a observar los valores de Phi obtenidos para cada alternativa, confirmando que “Tutor” es la mejor opción.

```

rownames(res$Acciones)

```

```

## [1] "Tutor"      "Cuidador"    "RRPP"        "Dependiente" "Hostelería"

```

(Justifica posibles diferencias entre métodos)