



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

PROGRAMA DE MAESTRÍA Y DOCTORADO EN
INGENIERÍA

FACULTAD DE INGENIERÍA

DECISIONES EN GRUPO CON PREFERENCIAS DE
SEGUNDO ORDEN

T E S I S

QUE PARA OPTAR POR EL GRADO DE:

DOCTOR EN INGENIERÍA

SISTEMAS- INVESTIGACIÓN DE OPERACIONES

P R E S E N T A :

LAURA PLAZOLA ZAMORA



TUTOR:

DR. SERVIO TULIO GUILLÉN BURGUETE

2007



Universidad Nacional
Autónoma de México



UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

DERECHOS RESERVADOS ©
PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

Todo el material contenido en esta tesis esta protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.

JURADO ASIGNADO:

Presidente: Dr. José Jesús Acosta Flores

Secretario: Dr. Ricardo Aceves García

1^{er}. Vocal: Dr. Servio Tulio Guillén Burguete

1^{er}. Suplente: Dr. Ramón Espinosa Armenta

2^{do}. Suplente: Dr. Roger Díaz de Cosío

Lugar o lugares donde se realizó la tesis:

Instituto de Ingeniería, UNAM.

TUTOR DE TESIS:
DR. SERVIO TULIO GUILLÉN BURGUETE



FIRMA

Para
Memo, Xavi y Javier

AGRADECIMIENTOS:

Quisiera ser más elocuente para poder expresar mi gratitud a todas las personas que de alguna manera han contribuido en la conclusión de mis estudios doctorales, pero a veces las palabras no son suficientes, espero mis actitudes sirvan como complemento para que sepan lo importantes que han sido en este proceso.

Estoy profundamente agradecida con mi director de tesis el Dr. Servio Tulio Guillén Burguete, por la propuesta de un tema tan fascinante como la toma de decisiones en grupo, por sus consejos, por la confianza y por todo el apoyo y comprensión que me ha brindado. ¡Muchas gracias Servio!

Agradezco también a los miembros del Jurado Doctoral, los Doctores José de Jesús Acosta Flores, Ricardo Aceves García, Roger Díaz de Cosío, por sus comentarios y sugerencias, y especialmente al Dr. Ramón Espinosa Armenta por sus valiosas y siempre acertadas observaciones y recomendaciones.

También deseo reconocer el beneficio obtenido por parte del CONACYT, así como del Instituto de Ingeniería de la UNAM, por aceptarme como becaria y por el apoyo recibido en el proceso de titulación.

Al Jefe del Departamento de Métodos Cuantitativos del CUCEA, Dr. Rubén Chavarín, por su total apoyo, interés y comprensión. A Luz González y Javier Dueñas por atender mis peticiones siempre amablemente y con una sonrisa, y por la inmensa cantidad de oficios que han tenido que hacer para la guardería de mi hijo. Al personal del STAUDG siempre dispuesto a ayudar con los múltiples trámites de la licencia.

A mis compañeros en el doctorado Hérica, Mayra y Norma, gracias por estar siempre dispuestas a ayudarme. También quiero expresar mi gratitud a mis entrañables amigos Benito Sánchez y Martha Torres, quienes han tenido que hacer mil trámites por mí y aguantar la lata que les doy, siempre me han ayudado solicita e incondicionalmente... a todos gracias por todos los favores que me han hecho, incluso a su pesar... A Leonor, Arturo, Paty, y a Judith, gracias por el hospedaje, por escuchar siempre mis patoaventuras y por recibirme como a una hermana.... Gerardo, gracias por la amena plática y los riquísimos tlacoyos... A Abel Castorena y a mi hermana Malena, por el financiamiento de algunos de mis viajes. Gracias por ser mi soporte durante las distintas etapas de este trabajo.

A mis compañeros del CUCEA, Sara Marín, Salvador Sandoval, Filadelfo León, Semeí Coronado, Víctor Gualajara, por su colaboración permanente y su actitud de apoyo. A la esposa de Semeí, Eileen Daspro, y al Dr. Héctor Cortés por la revisión del artículo en inglés, al Dr. Willy Cortéz por su ayuda con las citas bibliográficas. A todos gracias por ser mis amigos.

A mis padres, Goyita y Rey, por ser el mejor ejemplo de tenacidad y responsabilidad que he tenido; gracias a ustedes y a mi hermano Reynaldo por ayudarme a cuidar a mis niños, sin su apoyo hubiera sido prácticamente imposible poder terminar con la tesis. Gracias por impulsarme para seguir adelante siempre.

A mis hermosos muñequitos, Memo y Xavi, gracias por sus sonrisas, sus abrazos y sus travesuras, gracias por enseñarme el sentido de la vida. A Javier, mi esposo, por ser tan tolerante y comprensivo conmigo, por ponerme los pies en la tierra y por mostrarme día a día cuánto me amas. Gracias a los tres por motivarme a ser una mejor persona, por soportar mis ausencias, mi mal humor y mis locuras, los amo.

Laura Plazola Zamora

Tabla de contenido

Resumen

- **Introduccion**

- **El Estado del arte**
 - Articulos panoramicos**
 - Restricción del dominio de la Constitucion**
 - Relajamiento de las condiciones de racionalidad colectiva de la constitucion**
 - Comparaciones interpersonales**
 - Informacion adicional sobre las valoraciones individuales**

- **Planteamiento del problema**
 - Objetivos generales y especificos**
 - Justificacion**
 - Hipotesis**
 - Estrategia de investigacion**

- **Marco teorico**
 - Preferencia estricta e indiferencia**
 - Preferencia debil**
 - Funcion de valor ordinal**
 - Funcion de valor aditiva**
 - Medicion de diferencias de valor**
 - Conjuntos finitos de desigualdades lineales homogeneas**

- **Algunos metodos de votacion**
 - El metodo de agregacion de Borda**
 - El metodo de Condorcet**

Voto aprobatorio

Voto por veto

- **El teorema de imposibilidad de Arrow**
- **Constitucion de segundo orden**
- **Asignacion de una funcion de diferencia de valor**
- **Ejemplo intuitivo**
 - Bosquejo del metodo**
 - Software asociado**
 - Ejemplo de aplicacion**
- **Conclusiones**
- **Referencias bibliograficas**

RESUMEN

El problema de decisiones en grupo consiste en obtener la preferencia del grupal, sobre un conjunto finito de alternativas dado, a partir de las preferencias de los individuos que conforman el grupo, a través de un proceso llamado *constitución*.

En este trabajo se propone una constitución que considere un criterio de equidad entre los individuos, y que consiste en que todos ellos tengan la misma influencia o “número de votos” en el ordenamiento del grupo. Para esto se requiere que la información preferencial de los individuos incluya no solamente un ordenamiento de las alternativas, sino también datos sobre la fuerza de sus preferencias. Para medir la fuerza de preferencia de los integrantes de un grupo sobre un conjunto finito se utiliza el concepto de “*preferencias de segundo orden*”, que consiste en que el individuo ordene los posibles ordenamientos de las alternativas.

Además, si la fuerza de preferencia de cada miembro del grupo se puede modelar con una función aditiva de diferencia de valor, entonces la influencia de cada individuo en la decisión del grupo es aproximadamente proporcional a la diferencia de valor entre la mejor y la peor alternativa. El problema de encontrar una función de diferencia de valor que represente una preferencia conocida sobre los posibles ordenamientos del conjunto de alternativas, se como un problema de programación lineal. Se ha desarrollado un programa de cómputo en Delphi para implementar el procedimiento.

ABSTRACT

Group decision making consists in obtaining the group preference on a given finite set of alternatives, starting from individual's preferences through a process called *constitution*.

This study incorporates a criterion of equity among individuals, in which everybody influences the group ranking to the same degree. In order to achieve this, it is necessary that preferential information of the individuals not only includes a ranking of the alternatives, but also data on the strength of their preferences. Preferential information on the influence the group members have on the collective decision can be obtained by means of the concept of second order preferences, in which the individuals order the possible rankings of the alternatives.

If the preference strength of each group member can be modeled with an additive value difference function, then the influence of each individual in the decision of the group is approximately proportional to the value difference between the best and the worst alternative, which suggests that the Constitution must provide the adjustments necessary for this difference to be the same for each group member.

The problem of finding a value difference function representing a known preference of the possible rankings of A is solved as a linear programming problem. In order to implement this procedure it has been developed a computer program in Delphi.

INTRODUCCIÓN

Con frecuencia un grupo de personas tiene la responsabilidad de tomar decisiones, por ejemplo un comité directivo, un equipo de Administración, un cuerpo legislativo o la sociedad misma. Aunque los miembros del grupo tengan intereses en común, podrá haber diferencias de opinión y de preferencias entre ellos, lo que hace que la toma de decisiones en grupo sea un proceso complejo.

El problema de decisiones en grupo consiste en obtener la preferencia grupal, sobre un conjunto finito de alternativas dado, a partir de las preferencias de los individuos que conforman el grupo a través de un proceso llamado *constitución*. Formalmente, dado un conjunto finito de alternativas A , cuya cardinalidad es al menos 3, y un conjunto de n individuos, $n \geq 2$, que tienen preferencias sobre dicho conjunto de alternativas y se representan por medio de relaciones binarias R_i , completas y transitivas; decimos que una *constitución* es una función que asigna a cada posible configuración de preferencias individuales un ordenamiento de preferencias grupal. Esto es, si consideramos todos los posibles ordenamientos de preferencia \mathfrak{R} sobre A y el conjunto \mathfrak{R}^n de todas las n -adas (R_1, R_2, \dots, R_n) de órdenes de preferencia en \mathfrak{R} , llamado conjunto de *perfiles de preferencia individual*, entonces una constitución $f : \mathfrak{R}^n \rightarrow \mathfrak{R}$ asocia a cada perfil individual admisible un ordenamiento factible, llamado *orden de preferencia grupal* (D'Aspremont, 1985).

Aunque el estudio formal de las decisiones en grupo inició hace más de dos siglos con C. De Borda y el Marqués de Condorcet, la teoría adoptó su forma moderna con el teorema de imposibilidad de Arrow (1963) quien propuso 5 axiomas intuitivamente satisfactorios que debería cumplir todo procedimiento de agregación de preferencias individuales (Blair y Pollack, 1983):

1. La regla de agregación debe estar definida para todo posible perfil¹ de preferencias. (Condición de universalidad)

¹ Si consideramos todos los posibles ordenes débiles \mathfrak{R} sobre A y el conjunto \mathfrak{R}^n de todas las n -adas (R_1, R_2, \dots, R_n) de ordenes débiles en \mathfrak{R} , llamado conjunto de *perfiles de preferencia individual*, entonces una constitución $f : \mathfrak{R}^n \rightarrow \mathfrak{R}$ asocia a cada perfil individual un ordenamiento factible, llamado *orden de preferencia grupal*.

2. El orden grupal debe ser completo y transitivo, (condición de racionalidad) el contradominio de la Constitución es el conjunto \mathfrak{R} de preferencias completas y transitivas.
3. La regla debe satisfacer el principio de Pareto (condición de unanimidad). Para todo perfil (R_1, R_2, \dots, R_n) y para cualquier par de alternativas $x, y \in A$ se tiene, que si cada individuo prefiere estrictamente la alternativa x sobre la alternativa y , entonces el grupo prefiere estrictamente la alternativa x sobre la alternativa y .
4. El juicio sobre cada par de alternativas, debe ser independiente de las alternativas restantes. (condición de independencia de alternativas irrelevantes). El orden grupal entre dos alternativas sólo debe depender de cómo los individuos ordenan a esas dos alternativas y no debe depender de la manera en que los individuos ordenan al resto de las alternativas.
5. Ningún individuo puede constituirse en un dictador, en el sentido de que sus preferencias se reflejen en el orden social (condición de equidad).

El teorema de Arrow puede enunciarse como sigue:

No existe una constitución que permita que el orden de preferencia grupal sea definido a partir de los órdenes de preferencias individuales de manera tal que se satisfagan las condiciones 1, 2, 3 4 y 5 simultáneamente.

El teorema afirma que para cada constitución posible existe al menos un conjunto de preferencias individuales tal que la construcción del orden grupal viola al menos uno de los axiomas. Cada posible constitución tiene el potencial de ser injusto o irracional. (French, 1988). La imposibilidad podría traducirse como la no existencia de un procedimiento de agregación de preferencias individuales que sea indiscutiblemente democrático.

El resultado parece poco alentador, sin embargo, para tratar de evitar la imposibilidad, se han explorado varias maneras de proceder:

1. Restringir el dominio de la constitución
2. Relajar las condiciones de racionalidad de la constitución
3. Obtener más datos acerca de las valoraciones individuales por parte de los miembros del grupo

En algunos casos se han sugerido restricciones de los perfiles de preferencias individuales admisibles y en algunos otros se sustituye la propiedad de transitividad, por alguna propiedad más débil, como casi-transitividad² o aciclicidad³. El primer camino es de interés solamente para casos muy particulares y aunque en determinadas circunstancias es posible escapar de la imposibilidad, todavía no existe un procedimiento adecuado para establecer reglas de valoración grupal eficientes, racionales y democráticas (Villar, 1988). En el segundo caso se encontró que no había constituciones que fueran satisfactorias, ya que toda función de decisión grupal casi transitiva implicaba la existencia de una oligarquía⁴; además, bajo ciertas condiciones adicionales, la constitución resultaba dictatorial. (Ver Mas-Colell y Sonnenschein, 1972)

El tercer camino ha llevado a la conclusión de que una regla de agregación grupal que garantice en todas las situaciones que las decisiones puedan ser racionales y democráticas requiere que se tome en cuenta no únicamente las preferencias de los miembros del grupo, sino también la fuerza de esas preferencias, dado que en la formulación de Arrow, los ordenamientos individuales y el ordenamiento grupal no consideran la fuerza de preferencia de los individuos que conforman la sociedad. De hecho, el axioma de la Independencia de la alternativa irrelevante prohíbe cualquier idea de intensidad de preferencia y comparación interpersonal (Vincke et al, 1992). El propio Arrow (1963 pp. 59) concluye que:

Si excluimos la posibilidad de hacer comparaciones interpersonales de utilidad, entonces los únicos métodos para pasar de los gustos individuales a las preferencias sociales, que sean satisfactorios y que estén definidos para un amplio campo de conjuntos de ordenamientos individuales, serán impuestos o dictatoriales.

² La relación binaria R (ó \succeq), completa y transitiva, está formada por una parte simétrica, \sim (I), y una parte asimétrica, \succ (P). La casi transitividad supone que la parte simétrica de la relación R no es transitiva, pero su parte simétrica si lo es. En otras palabras, para todo $a, b, c \in A$, aPb y $bPc \Rightarrow aPc$.

³ Para todo $x_1, \dots, x_j \in A$, $\left[\left\{ x_1Px_2 \wedge x_2Px_3 \wedge \dots \wedge x_{j-1}Px_j \right\} \rightarrow x_1Rx_j \right]$

⁴ Un subconjunto de individuos con capacidad de decisión sobre los demás, cada uno con poder de veto.

Este último camino se ha topado con otra imposibilidad, la de hacer comparaciones de la fuerza la preferencia entre individuos distintos, del tipo “el individuo A prefiere la alternativa a sobre la alternativa b con mayor fuerza de la que el individuo B prefiere la alternativa c sobre la alternativa d ”. Desafortunadamente, hasta ahora, no se ha desarrollado un método objetivo para hacer comparaciones interpersonales y no hay manera de satisfacer las condiciones de equidad y racionalidad a menos que se realicen esas comparaciones.

La información preferencial más completa requiere de comparaciones interpersonales de la fuerza de preferencia. Pues para que el grupo prefiera una alternativa sobre otra es necesario que cada miembro del grupo pueda comparar las diferencias del tipo $v_i(a) - v_i(b)$, que se interpretan como la fuerza de la preferencia de la alternativa a sobre la alternativa b , donde $v_i(\cdot)$ es la función de evaluación del individuo i , definida sobre A . Cada función de evaluación representa una preferencia sobre el producto cartesiano A^2 , de tal forma que

$$v_i(a) - v_i(b) > v_i(c) - v_i(d)$$

indica que la preferencia de a sobre b es más fuerte que la preferencia de c sobre d . Una función con esta característica se conoce como *función de diferencia de valor*. Así, la función de evaluación del grupo sobre el conjunto de alternativas A , v_g está dada por la función:

$$v_g(a) = \varphi(v_1(a), v_2(a), \dots, v_n(a))$$

De acuerdo con French (1988) se puede mostrar que si se permite a cada miembro del grupo establecer su función de diferencia de valor para las alternativas, éstas se pueden agregar en una función de valor grupal de manera que sea consistente con respecto a los axiomas de Arrow. Este resultado sugiere que si los miembros del grupo “votan” considerando sus fuerzas de preferencias para las distintas alternativas, por medio de funciones de diferencia de valor, entonces existen procedimientos de agregación que están de acuerdo con los axiomas de Arrow. Aunque implícitamente se asume que es posible comparar la fuerza de preferencia de una persona con la de otra.

Un camino poco explorado ha sido el de agregar información preferencial que no resulte de la comparación de la fuerza de preferencia entre los individuos del grupo,

sino de tomar en cuenta en forma equitativa las preferencias de los elementos del grupo, estableciendo cuál es la aportación de cada individuo al orden grupal, tratando de que todos tengan la misma influencia sobre la decisión final. Esta falta de información ha motivado la búsqueda de nuevos métodos de agregación que incorporen magnitudes que reflejen las influencias de los individuos en la decisión colectiva.

El principal interés de este proyecto de investigación es proponer una constitución que considere un criterio de equidad entre los individuos, y que consiste en que todos ellos tengan la misma influencia o “número de votos” en el ordenamiento del grupo. Para esto se requiere que la información preferencial de los individuos incluya no solamente un ordenamiento de las alternativas, sino también datos sobre la fuerza de sus preferencias. Para medir la fuerza de preferencia de los integrantes de un grupo sobre un conjunto finito usaremos el concepto de “*preferencias de segundo orden*”, que consiste en que el individuo ordene los posibles ordenamientos de las alternativas.

Este trabajo se organiza de la siguiente manera, en el capítulo 1 se presenta el estado del arte, iniciando con las referencias de artículos que abordan una panorámica general del problema de decisiones en grupo, y continuamos con la literatura disponible organizada en cuanto a las diversas maneras de proceder para escapar de la imposibilidad, a decir:

1. Restricciones del dominio de la constitución
2. Relajación de las condiciones de racionalidad colectiva de la constitución
3. Información adicional acerca de las valoraciones individuales por parte de los miembros del grupo.

También se incluye la literatura que reporta los aspectos teóricos de las comparaciones interpersonales de la fuerza de preferencia.

En el capítulo 2 se plantea en forma general la problemática que se enfrenta, se definen los objetivos de la investigación, se da la justificación, la hipótesis y la estrategia de investigación.

El capítulo 3 trata sobre los principales conceptos y teoremas acerca de las relaciones binarias y órdenes, las relaciones de preferencia estricta, débil y de indiferencia, las

funciones de valor ordinal, funciones aditivas y la medición de diferencias de valor, también se muestra que si se permite a cada miembro del grupo establecer su función de diferencia de valor para las alternativas, éstas se pueden agregar en una función de valor grupal de manera que sea consistente con respecto a los axiomas de Arrow (French, 1988).

El capítulo 4 se refiere a algunos métodos de votación como el método de agregación de Borda, un método ordinal para decisiones en grupo que esencialmente consiste en sumar los rangos obtenidos por una alternativa con respecto a cada uno de los individuos del grupo. También se expone el método de comparación por pares de Condorcet, desarrollado en 1785, y simula una serie de elecciones entre cada alternativa y las demás. Este método se propuso cuando al encontrarse con que la votación por mayoría simple puede dar como resultado configuraciones de preferencia grupal no transitivas. El método del voto aprobatorio (Brams y Fishburn, 1978) en el que se puede votar solo una vez por cada alternativa, pero se puede votar por tantas alternativas como se quiera. Y por último el método del voto por veto (Mueller (1978), Barberá, (1984)), en el que se establece un orden en el que los individuos pueden ejercer su veto.

En el capítulo 5 se presenta el teorema de imposibilidad de Arrow y su demostración basada en French, 1988.

En el capítulo 6, se define una *constitución de segundo orden* como aquella que toma en cuenta la preferencia de cada miembro i del grupo sobre el conjunto de los órdenes débiles sobre A , interpretados como posibles resultados P_g de la elección de grupo. Y se establecen sus propiedades, se muestra además que la constitución de segundo orden satisface los axiomas de Arrow.

El capítulo 7 trata sobre la asignación de una función de diferencia de valor (acorde con una función de diferencia de valor sobre el conjunto de alternativas) que represente una preferencia conocida sobre el conjunto de los ordenamientos de las alternativas. El problema se resuelve como un problema de programación lineal (Krantz, 1971).

El capítulo 8 ilustra el procedimiento de construcción de una función de valor para los individuos en un grupo con un ejemplo intuitivo, también se describe el programa de cómputo que se desarrolló para implementar el procedimiento y se resuelve un

problema aplicando la metodología y el software de decisiones en grupo con preferencias de segundo orden.

Existen dos apéndices, en el primero se ilustra que se satisface el axioma de la independencia de la alternativa irrelevante y en el segundo se presenta el código fuente de programa desarrollado.

1. EL ESTADO DEL ARTE

Existe una amplia literatura generada por el resultado de imposibilidad de Arrow. Parte de ésta ha tratado de proponer conjuntos de propiedades alternativos al propuesto por Arrow, manteniendo el mismo marco conceptual determinado por el axioma de racionalidad. En algunos casos se han sugerido restricciones de las configuraciones de preferencias individuales admisibles y en algunos otros se sustituye la propiedad de transitividad, por alguna propiedad más débil, como casi- transitividad o aciclicidad.

Algunos autores han investigado sobre las implicaciones de los supuestos concernientes a la medibilidad y la comparabilidad interpersonal de la utilidad. Por ejemplo, Roberts (1980), Blackorby, Ch, *et al* (1984), D'Aspremont y Gevers (2001), entre otros, y han mostrado que la introducción de tales conceptos hacen posible encontrar procedimientos de agregación "racionales", es decir, que satisfacen condiciones similares a las de Arrow, pero cada uno de ellos asume implícitamente una unidad común de preferencia.

Un camino poco explorado ha sido el de agregar información preferencial que no resulte de la comparación de la fuerza de preferencia entre los individuos del grupo, sino de tomar en cuenta en forma equitativa las preferencias de los elementos del grupo.

En este capítulo se presenta el estado del arte, iniciando con las referencias de artículos que abordan una panorámica general del problema de decisiones en grupo, y continuamos con la literatura disponible organizada en cuanto a las diversas maneras de proceder para escapar de la imposibilidad, a decir:

1. Restricciones del dominio de la constitución

Relajación de las condiciones de racionalidad colectiva de la constitución

Información adicional acerca de las valoraciones individuales por parte de los miembros del grupo

También se incluye la literatura que reporta los aspectos teóricos de las comparaciones interpersonales de la fuerza de preferencia.

1.1 Artículos panorámicos:

Barberá, S (1984) destaca diversos grupos de problemas de la teoría de la Elección social, esencialmente sobre los temas de agregación, comportamientos estratégicos, elección probabilística y distribución de derechos, y explica el interés de cada uno de ellos proporcionando ejemplos tipo de los resultados obtenidos.

Blair, D. y R. Pollack (1983) hacen un análisis de los axiomas de Arrow y las posibles salidas a la imposibilidad, concluyendo que la racionalidad colectiva, la decisividad y la igualdad en la fuerza del voto, se encuentran en conflicto irreconciliable, no obstante, el análisis ha permitido entender mas a fondo los sistemas de votación existentes y sugieren que quizá se puedan encontrar otros mejores.

Villar, A. (1988), ofrece una visión general de la Teoría de la elección social, haciendo una revisión de la literatura relacionada con el análisis de las distintas posibilidades de diseño de reglas de elección social. Se divide el análisis en tres bloques, el primero, trata sobre el Teorema de Arrow y los intentos de escape a la imposibilidad, manteniendo el mismo esquema de ordinalidad y no comparabilidad interpersonal. El segundo, trata de los resultados de posibilidad que surgen cuando se permiten comparaciones interpersonales de utilidad y diferentes grados de medibilidad de las preferencias individuales y por último se analizan las dificultades que surgen al construir reglas de elección social, relacionados principalmente con las limitaciones derivadas de utilizar preferencias individuales como único criterio de evaluación.

Plata-Pérez, L (1996) presenta un amplio panorama de resultados y problemas abiertos en la Teoría de la elección social, así como su relación con la teoría de juegos.

Taylor, L (1998). Revisa algunos trabajos de Sen (1997) y hace algunas reflexiones acerca de la teoría de la elección social relacionadas con la metodología, la formalización y su pertinencia. Cuestiona la pertinencia de la teoría como programa de investigación y como instrumento analítico para explicar y solucionar problemas de desigualdad e inequidad.

Müller, A. (ND). Presenta una demostración diagramática del teorema de Arrow y se desarrollan algunas reflexiones acerca del alcance de las conclusiones el teorema, en cuanto a la existencia de uno o varios dictadores, la coincidencia entre dos situaciones

alternativas, una donde las preferencias se construyen a partir de un individuo en particular, y otra donde el procedimiento para definir la preferencia social arroja como resultado preferencias coincidentes con las de algún individuo en particular.

Dardanoni, V. (2001). Presenta una prueba sencilla del Teorema de Imposibilidad de Arrow iniciando con el caso de tres individuos sobre tres alternativas.

1.2 Restricción del dominio de la constitución.

Black, D. (1958), propone que las preferencias sean unimodales (single peaked preferences). Si se representa la función de preferencias como función de una variable que ordene las opciones de menor a mayor, las preferencias unimodales son aquellas en las que todo máximo local es un máximo global, lo cual restringe el requisito de universalidad a distribuciones unimodales de preferencias cuando todos los individuos evalúan para sí cada una de las alternativas de acuerdo con un solo criterio y en cada una de las elecciones por pares votan por la alternativa mas próxima a la situación considerada óptima por el individuo.

Maskin (1995), da una prueba sencilla del Teorema de Imposibilidad de Arrow y establece un nuevo resultado para la regla de la mayoría, prueba que las reglas de decisión colectiva que satisfacen anonimidad¹, neutralidad, propiedad de Pareto e independencia de la alternativa irrelevante, conducen a que la regla de la mayoría es transitiva sobre todas las clases de dominio de preferencias, mientras el número de individuos sea impar.

Massó, J. (1996), se enfoca en situaciones donde los individuos tienen preferencias unimodales, generalizadas a varias dimensiones y en este contexto investiga el conjunto de funciones de decisión social no manipulables, donde se supone que declarar la verdad es una estrategia dominante para todos los individuos. Esta clase de funciones es de interés porque permite dejar de considerar los aspectos estratégicos del proceso de decisión social.

¹ Esta condición dice que el resultado de la votación no varía ante permutaciones de los individuos, todos juegan un papel simétrico en la votación.

Schwartz, T. (2001), muestran que los supuestos del teorema de Arrow concluyen que la intransitividad social debe afectar algún perfil de preferencias individuales, éste no debe ser un ciclo, pero los demás pueden tener empates. Si se agrega un límite de empates, entonces obtiene un ciclo caótico e inestabilidad. Esta situación no depende más que de los perfiles.

1.3 Relajamiento de las condiciones de racionalidad colectiva de la constitución.

Mas Colell, A. y H. Sonnenschein (1972), demuestran varios teoremas para funciones de decisión social con relaciones de preferencia social casitrantitivas. Estos teoremas prueban que es imposible encontrar reglas de agregación social que satisfagan al mismo tiempo la independencia de la alternativa irrelevante, la condición de Pareto y la coexistencia de un dictador débil (o condición de respuesta positiva de May), además de que la introducción de la respuesta positiva genera, de nuevo, reglas dictatoriales.

Peris, J. E. y C. Sánchez (2001), Analizan la representación explícita de una agenda fija de correspondencias de elección social bajo diferentes condiciones de racionalidad y establecen una relación entre el conjunto de elección social y los conjuntos maximales individuales, que describen explícitamente una correspondencia de elección social que satisface estas condiciones de racionalidad. Analizan la existencia de dictadores, oligarquías e individuos con poder de veto sobre el conjunto de elección social en el marco de una agenda fija.

1.4 Comparaciones interpersonales

Los procedimientos de agregación de preferencias individuales que intentan reflejar la fuerza de preferencia se enfrentan con el problema de las comparaciones interpersonales, en el sentido de establecer una unidad de medición común para todos los miembros del grupo y/o una base de referencia (Luce, R. y H. Raiffa, 1957). Los supuestos de información se formalizan típicamente especificando el tipo de escala en la que la fuerza de la preferencia es medible intra e interpersonalmente, esto es, especificando la clase de transformaciones a las que pueden estar sujetos los perfiles individuales sin cambiar su contenido informativo (Khmelnitskaya, A. Y J. Weymark, 2000). Por lo que el problema de obtener información adicional sobre la fuerza de preferencia continúa vigente.

Existen varios trabajos que clasifican todas las posibles formas de hacer comparaciones, es decir, los diferentes axiomas de invarianza, y han mostrado que la introducción de tales conceptos hace posible encontrar procedimientos de agregación “racionales”, que satisfacen condiciones similares a las de Arrow, pero cada uno de ellos asume implícitamente una unidad común de preferencia. Para resultados detallados véase D’Aspremont y Gevers (1977), Roberts (1980), Blackorby et al (1984) y Sen (1979). D’Aspremont y Gevers (2001), entre otros.

Harsanyi, J. (1955). Propone un método conocido como “utilitarista” donde cada individuo establece la utilidad que obtendría con cada alternativa. Para cada alternativa se considera la suma de los niveles de utilidad sobre los individuos y se elige la alternativa con el mayor nivel de utilidad agregado. Utiliza comparaciones interpersonales entre las diferencias de niveles.

Sen (1977, 1979, 1984) hace una taxonomía de diferentes supuestos de medibilidad y comparabilidad. En Sen (1979), las interpretaciones de las comparaciones interpersonales pueden clasificarse en dos tipos, descriptivas y normativas. Dentro de las descriptivas, se distinguen tres: conductista, comparación introspectiva, y elección introspectiva. Cada uno de los enfoques descriptivos tiene sus ventajas y desventajas, aunque estas interpretaciones son más bien de interés ético y de relevancia moral. En cuanto a las interpretaciones normativas, cabe señalar que éstas tomarán diferentes formas de acuerdo con la regla de agregación elegida. No existe una única interpretación normativa de las comparaciones interpersonales, cada regla de agregación implica sobre la interpretación normativa una clase diferente de restricciones.

Roberts (1980), considera la derivación de ordenamientos sociales no impuestos y no neutrales cuando el bienestar individual satisface varios supuestos de comparabilidad y medibilidad. Modifica los axiomas de Arrow para admitir la influencia de diferentes tipos de información, principalmente la condición de independencia de la alternativa irrelevante y el principio de Pareto

Blackorby, Ch, *et al* (1984), presentan un modelo y muestran que bajo un conjunto de supuestos, llamados “Welfarism”, todos los ordenamientos asociados con una regla dada pueden representarse por un ordenamiento sencillo del espacio de n -adas de utilidad. Introducen la taxonomía de comparabilidad y medibilidad de utilidades.

Plata-Pérez, L. (1999). Presenta algunos avances en el problema de la clasificación de ordenamientos de bienestar social al tratar con el tipo de comparaciones interpersonales permitidas. Da una caracterización de los axiomas de invarianza que admiten comparaciones interpersonales parciales. Cada axioma de comparación interpersonal está asociado con un par de particiones del conjunto de individuos: clases que admiten comparaciones de orígenes y clases que admiten comparaciones de unidades.

Khmelnitskaya, A. y J. Weymark (2000), suponen que los tipos de comparaciones de utilidad que pueden hacerse difieren en los distintos subgrupos de la población. Como dentro de cada subgrupo la utilidad puede ser ordinal o cardinalmente medible, los niveles y diferencias de utilidad pueden o no ser interpersonalmente comparables dentro de un subgrupo. Además, no es posible hacer comparaciones interpersonales entre subgrupos. Con esta información muestran que cualquier orden de bienestar social continuo que satisface el principio débil de Pareto sólo depende de las utilidades de uno de los subgrupos. El tipo de escala de este subgrupo dictatorial determina la clase de órdenes de bienestar social consistentes con estos supuestos de información preferencial.

1.5 Información adicional sobre las valoraciones individuales.

Un camino poco explorado ha sido el de agregar información preferencial que no resulte de la comparación de la fuerza de preferencia entre los individuos del grupo, sino de tomar en cuenta en forma equitativa las preferencias de los elementos del grupo, estableciendo cuál es la aportación de cada individuo al orden grupal, tratando de que todos tengan la misma influencia sobre la decisión final. Esta falta de información ha motivado la búsqueda de nuevos métodos de agregación que incorporen magnitudes que reflejen las influencias de los individuos en la decisión colectiva.

En Sen (1974, 1979) y en Van der Veen (1981) se sugieren algunas interpretaciones alternativas de comparación interpersonal basadas en ordenamientos de orden superior, donde es posible definir un ordenamiento no directamente sobre el espacio de alternativas, sino sobre los ordenamientos de las alternativas, estos meta-ordenamientos pueden aplicarse al problema de encontrar una medida significativa de la utilidad cardinal.

Arora, N. y G.M. Allenby (1999) desarrollan un modelo jerárquico Bayesiano de toma de decisiones en grupo que proporciona los estimados individuales de influencia en el nivel de atributos del producto. El modelo propuesto relaciona las medidas de influencia para identificar a los individuos con mayor influencia en la decisión.

Jabeur, K y J.Martel (2002), proponen un método para cuantificar, para cada par de acciones, la importancia relativa de los miembros del grupo. El método se basa primero en el método de Simos revisado y el método de DeGroot y después en la explotación de las estructuras de preferencia individuales.

En relación con el criterio de equidad, en Guillén et al (2002) y Guillén (2003) se sugiere que si las fuerzas de preferencia de cada miembro del grupo se pueden modelar con una función aditiva de diferencia de valor, entonces la influencia de cada individuo en la decisión del grupo es aproximadamente proporcional a la diferencia de valor entre la mejor y la peor alternativa, lo cual implica que la constitución debe hacer los ajustes para que esta diferencia sea la misma para todos los miembros del grupo.

2. PLANTEAMIENTO DEL PROBLEMA.

En la toma de decisiones en grupo, al agregar preferencias individuales con información limitada, como en el caso de la agregación de Arrow, se ha visto condenada a fallar, precisamente por que no se permite el uso de información cardinal que considere la fuerza de preferencia de los individuos. Por otro lado, no existe un método de agregación de preferencias individuales que refleje la igualdad de la fuerza del voto entre los miembros del grupo. La dificultad en el proceso de agregación radica en cómo medir la fuerza de preferencia de los miembros del grupo, ya que los procedimientos existentes para medir fuerzas de preferencia requieren que se cumpla una condición o axioma de solubilidad, que en la práctica implica que el conjunto de alternativas sea infinito.

El trabajo de Guillén *et al* (2002) y Guillén (2003) sugiere la posibilidad de diseñar un procedimiento de medición de la fuerza de preferencia sobre un conjunto finito empleando el concepto de “preferencias de segundo orden”, consistente en que el individuo ordene los posibles ordenamientos de las alternativas. Esta sugerencia está lejos de llevarse a la práctica por varias razones. Una es que el procedimiento debe estar ligado a la constitución y no hay aún la teoría para hacer esta liga; otra es que el problema es extremadamente combinatorio, por lo que se requiere elaborar programas de cómputo que apoyen al individuo en esas mediciones.

2.1 Objetivos generales y específicos

1. Desarrollar un procedimiento de agregación de preferencias individuales, para la toma de decisiones en grupo, que utilice información preferencial adicional a los ordenamientos de preferencia individual, basado en el concepto de preferencias de segundo orden.
 - 1.1. Diseñar la Constitución de Segundo orden
 - 1.2. Establecer las propiedades de la Constitución

2. Diseñar un mecanismo que permita medir la fuerza de la preferencia de los miembros del grupo, utilizando el concepto de preferencias de segundo orden.

- 2.1. Definir el procedimiento de asignación de una función de diferencia de valor
- 2.2. Elaborar el programa de cómputo que facilite la medición de la fuerza de preferencia ya que el problema es extremadamente combinatorio.

2.2 Justificación

Toda sociedad o agrupación de personas, tiene que tomar decisiones colectivas y diseñar procedimientos de agregación de preferencias, por imperfectos que estos sean. Sin embargo, el teorema de Imposibilidad de Arrow ajusta sólidamente en un programa que trate de hacer mas sistemático el análisis de la agregación social (Sen, 1995).

Es fácilmente observable que en muchas de las decisiones de Estado o en un contexto político, se emplean nociones agregadas del “bien común” o de “voluntad popular” y obviamente se requiere hacer una revisión a la luz de los axiomas de Arrow. Las decisiones colectivas de interés práctico deben basarse en fundamentos sólidos. La formulación de Arrow hace explícito ese fundamento y su enfoque tiene una importancia esencial para la elaboración de políticas y procedimientos de orden económico, político o social.

Aunado a esto, pese a la gran cantidad de literatura generada por el resultado de imposibilidad de Arrow, donde se ha tratado de proponer conjuntos de propiedades alternativos al propuesto por Arrow, se **mantiene el mismo marco conceptual** determinado por la ordinalidad de la información y el axioma de racionalidad. La investigación doctoral que se propone se interesa por explorar la posibilidad, en la cual se considere un criterio de equidad entre los individuos, y que todos ellos tengan la misma influencia o “número de votos” en el ordenamiento del grupo. Para esto se requiere que la información preferencial de los individuos incluya no solamente un ordenamiento de las alternativas, sino también datos sobre la fuerza de sus preferencias, ampliando así la cantidad de información sobre las preferencias de los individuos; por lo que es necesario diseñar un procedimiento para modelar esta fuerza de preferencia cuando el conjunto de alternativas es finito.

2.3 Hipótesis

Si las fuerzas de preferencia de cada miembro del grupo se pueden modelar con una función aditiva de diferencia de valor, entonces la influencia de cada individuo en la decisión del grupo será aproximadamente proporcional a la diferencia de valor entre la mejor y la peor alternativa, lo cual sugiere que la constitución debe hacer los ajustes para que esta diferencia sea la misma para todos los miembros del grupo. La información preferencial acerca de la influencia que tienen los miembros del grupo sobre la decisión colectiva se podrá obtener a partir del concepto de preferencias de segundo orden consistente en que el individuo ordene los posibles ordenamientos de las alternativas

2.4 Estrategia de investigación

Para poder introducir información preferencial más allá de los ordenamientos individuales, en un procedimiento de agregación social de preferencias es necesario comparar la fuerza de preferencia de un individuo con la de otro. Después de una revisión de la literatura disponible que incluya desarrollos teóricos concernientes a las comparaciones interpersonales de la fuerza de preferencia y condiciones de medibilidad y comparabilidad, se estimarán las funciones de diferencia de valor.

Dado que el problema de medición de la fuerza de preferencia sobre un conjunto finito empleando el concepto de “preferencias de segundo orden es extremadamente combinatorio, se elaborará el software necesario que facilite las mediciones. Se generarán rutinas con datos reales para corroborar los supuestos. Se contempla realizar el programa de cómputo en Delphi 7 ya que es una potente herramienta de desarrollo de programas que permite la creación de aplicaciones para Windows.

La corroboración de los supuestos se hará en dos fases: evaluaciones individuales e integradas. El propósito de las evaluaciones individuales es identificar las contribuciones en cuanto a la medición de la fuerza de preferencia de cada individuo, mientras que la evaluación integrada es para identificar las interacciones y complicaciones de las comparaciones de las influencias de los individuos en la decisión del grupo tratando que estas sean iguales.

3. MARCO TEÓRICO

3.1. Relaciones binarias y órdenes

Una *relación binaria* R sobre un conjunto A , es un subconjunto de $A \times A$, tal que,

$$R = \{(a, b) \mid a, b \in A, \} \subseteq A \times A$$

A es el conjunto de alternativas u opciones factibles sobre las que se toma la decisión. Si el par ordenado (a, b) pertenece a R , se pueden utilizar indistintamente las siguientes notaciones: $(a, b) \in R$ o aRb . R incluye las relaciones de preferencia estricta P e indiferencia I . De este modo, $x P y$ significa que x es preferido a y , $x I y$ significa que x e y son indiferentes.

R puede satisfacer las siguientes propiedades:

- 1.- Transitividad, $a R b$ y $b R c \Rightarrow a R c$, para todo $a, b, c \in A$;
- 2.- Simetría, $a R b \Rightarrow b R a$, para todo $a, b \in A$;
- 3.- Asimetría, $a R b \Rightarrow b \not R a$, para todo $a, b \in A$;
- 4.- Antisimetría, $a R b$ y $b R a \Rightarrow a = b$ para todo $a, b \in A$;
- 5.- Reflexividad, $a R a$ para todo $a \in A$;
- 6.- Irreflexividad, $a \not R a \forall a \in A$;
- 7.- Transitividad negativa, $a \not R b, b \not R c \Rightarrow a \not R c \forall a, b, c \in A$;
- 8.- Completitud, para todo $a, b \in A$, $a R b$ ó $b R a$, de forma equivalente, $a \not R b$ y $b \not R a$;
- 9.- Casi-transitividad, para todo $a, b, c \in A$, $a P b$ y $b P c \Rightarrow a P c$;
- 10.- Aciclicidad, Para todo $x_1, \dots, x_j \in A$, $\left[\{x_1 P x_2 \wedge x_2 P x_3 \wedge \dots \wedge x_{j-1} P x_j\} \rightarrow x_1 R x_j \right]$

Cualquier relación R que sea reflexiva, transitiva y completa se conoce como una *relación de orden*. En particular, una relación asimétrica y transitiva, es un *orden parcial estricto* (algunos autores se refieren a éste simplemente como orden estricto). Una relación completa y transitiva es un *orden débil*. Un orden débil antisimétrico, se conoce como *orden lineal* u *orden simple*. Una relación reflexiva, simétrica y transitiva, se llama *relación de equivalencia*. Una relación reflexiva y transitiva, se conoce como *cuasiorden* (o preorden parcial). Una relación reflexiva, transitiva y antisimétrica, es un

² $a \not R b$ indica la negación de la relación, o lo que es lo mismo $a R^c b$ o $(a, b) \in R^c$.

orden parcial. Una relación asimétrica y negativa transitiva, se conoce como un *orden débil estricto*. Una relación transitiva, asimétrica y completa es un *orden fuerte*.

3.2. Preferencia estricta e indiferencia

Escribimos aPb ó $a \succ b$, para representar que un individuo prefiere estrictamente la opción a sobre la opción b . La preferencia estricta de un individuo racional debe cumplir las propiedades de transitividad y asimetría, por lo que la relación de preferencia es un orden estricto.

De la misma forma, para la relación de indiferencia usamos la notación aIb o $a \sim b$, para dar a entender que el individuo es indiferente entre las opciones a y b . La relación de indiferencia debe ser reflexiva, simétrica y transitiva o sea una relación de equivalencia.

3.3. Preferencia débil

Escribimos $a \sqsupseteq b$ para representar que el individuo prefiere débilmente la opción a sobre la opción b ($a \sqsupseteq b$ si y solo si a es preferida o indiferente a b). La relación \sqsupseteq es la unión de las relaciones \succ y \sim y se interpreta como que la opción a es al menos tan buena como la opción b .

Para poder suponer la racionalidad del individuo \sqsupseteq debe cumplir un mínimo de condiciones: completitud, transitividad, consistencia de la indiferencia y la preferencia débil y consistencia de la preferencia estricta y la preferencia débil.

A continuación se establecen formalmente las condiciones:

- Completitud.- \sqsupseteq es completa: para todo $a, b \in A$, $a \sqsupseteq b$ o $b \sqsupseteq a$ o ambos se cumplen.
- Transitividad.- \sqsupseteq es transitiva: para todo $a, b, c \in A$, si $a \sqsupseteq b$ y $b \sqsupseteq c$, entonces $a \sqsupseteq c$.
- Consistencia de la indiferencia y la preferencia débil.- Para cualquier par de opciones $a, b \in A$, $a \sim b \leftrightarrow (a \sqsupseteq b \text{ y } b \sqsupseteq a)$.
- Consistencia de la preferencia estricta y la preferencia débil.- Para cualquier par de opciones $a, b \in A$, $a \succ b \leftrightarrow b \not\underset{\sim}{\sqsupseteq} a$.

De esta manera, la completitud y la transitividad hacen de la relación \sqsupseteq un orden débil.

3.4. Función de valor ordinal.

Sea A el conjunto de alternativas y \square la relación de preferencia débil del individuo sobre A . Entonces $v(a)$ es una función de valor ordinal que representa estas preferencias si $v(\cdot)$ es una función valuada en los reales sobre A tal que

$$v(a) \geq v(b) \leftrightarrow a \square b \quad (1)$$

Cuando (1) se cumple decimos que $v(\cdot)$ es acorde o representa a la relación \square sobre A .

Al usar $v(\cdot)$ solamente utilizamos el orden de la recta real en esta representación, por lo que no tiene significado alguno atribuir alguna interpretación al valor de $v(\cdot)$ sobre un conjunto de objetos. De la misma forma, tampoco podemos atribuir significado a las diferencias de valor. Por ejemplo

$$(v(a) - v(c)) > (v(b) - v(a))$$

No tiene significado el decir que un incremento sobre el valor de a sobre c es mayor o menor que el de b sobre a .

Para que exista una representación de función de valor ordinal es necesario que \square sea un orden débil. La preferencia estricta y la indiferencia tienen la siguiente representación:

$$a \succ b \Leftrightarrow b \not\succeq a \quad (\text{por la condición de consistencia de la preferencia estricta y débil}),$$

$$\Leftrightarrow v(b) \not\geq v(a) \quad (\text{por la definición de función de valor ordinal})$$

$$\Leftrightarrow v(a) > v(b)$$

$$a \sim b \Leftrightarrow a \square b \text{ y } b \square a \quad (\text{por la condición de consistencia de la indiferencia y la preferencia débil})$$

$$\Leftrightarrow v(a) \geq v(b) \text{ y } v(b) \geq v(a) \Leftrightarrow v(a) = v(b).$$

Si el conjunto A es finito entonces las condiciones de racionalidad de la preferencia débil son necesarias y suficientes para que exista la representación de la función de valor ordinal.

Teorema.

Para un conjunto finito de objetos $A = \{a_1, a_2, \dots, a_n\}$ con un orden débil \square que cumple las condiciones de completitud, transitividad, consistencia de la indiferencia, y consistencia de la preferencia estricta, siempre se puede construir una función de valor ordinal acorde.

Demostración. (French, 1988)

Sea $v(a_1)$ igual al número de objetos $a_j \succ A$ tal que $a_1 \square a_j$. Se desea mostrar que $v(a_i) \geq v(a_j) \leftrightarrow a_i \square a_j$

a) $a_i \square a_j \Rightarrow v(a_i) \geq v(a_j)$. Sea a_k cualquier objeto tal que $a_j \square a_k$. Si $a_i \square a_j$, entonces por transitividad, $a_i \square a_k$. Por tanto, $a_i \square a_j \Rightarrow v(a_i) \geq v(a_j)$.

b) $a_i \succ a_j \Rightarrow v(a_i) > v(a_j)$. Ahora $a_i \succ a_j \Rightarrow a_i \square a_j \Rightarrow v(a_i) \geq v(a_j)$. Pero $a_i \succ a_j \Rightarrow a_j \not\simeq a_i$. Se desea mostrar que $v(a_i) \geq v(a_j) \Rightarrow a_i \square a_j$.

Supongamos que esto no es verdad. Entonces existe un a_k y un a_l tal que $v(a_k) \geq v(a_l)$ y que $a_k \not\simeq a_l$. Pero,

$$a_k \not\simeq a_l \Rightarrow a_l \succ a_k \Rightarrow v(a_l) > v(a_k) \Rightarrow v(a_k) \geq v(a_l),$$

lo cual contradice nuestra suposición. Entonces a_k y a_l no existen y se ha demostrado el resultado requerido.

Una función $\Phi: \mathfrak{R} \rightarrow \mathfrak{R}$ es estrictamente creciente si $\forall x_1, x_2 :$

$$\Phi(x_1) > \Phi(x_2) \Leftrightarrow x_1 > x_2$$

En otras palabras una función estrictamente creciente tiene un gradiente estrictamente positivo en cualquier lugar. De manera equivalente, $\forall x_1, x_2 : \Phi(x_1) \geq \Phi(x_2) \Leftrightarrow x_1 \geq x_2$.

Teorema.

Sea \square un orden débil que satisface los axiomas de racionalidad. Entonces $v(\cdot)$ y $w(\cdot)$ son funciones de valor ordinal acordes con \square si y sólo si existe una función estrictamente creciente Φ , tal que:

$$w(a) = \Phi[v(a)], \forall a \in A$$

Demostración (French, 1988)

Supongamos que $v(\cdot)$ es una función de valor ordinal acorde con \square y con que Φ es una función estrictamente creciente. Entonces

$a \sqsupseteq b \leftrightarrow v(a) \geq v(b)$, ya que v es una función de valor ordinal,
 $\leftrightarrow \Phi[v(a)] \geq \Phi[v(b)]$,
 $\leftrightarrow w[a] \geq w[b]$.

Por tanto, $w(\cdot)$ es una función de valor ordinal que también es acorde con \sqsupseteq .

3.5. Función de valor aditiva.

Una estructura de preferencia es aditiva, si existe una función de valor que refleja que la estructura de preferencia puede expresarse como:

$$v(x, y) = v_x(x) + v_y(y)$$

Un procedimiento de agregación de preferencias es *aditivo* si esta función tiene la forma

$$v(a) = \sum_{i \in I} v_i(a)$$

La función aditiva de valor v no es única, pues se puede demostrar que (para dos o más criterios) cualquier otra función aditiva sobre A

$$w(a) = \sum_{i \in I} w_i(a)$$

es equivalente a v si y sólo si se relacionan por una transformación afín positiva, es decir, que existen $\alpha, \beta_i, \alpha > 0$, tales que para todo $i \in I$ se cumple $w_i(a) = \alpha v_i(a) + \beta_i$.

3.6. Medición de diferencias de valor.

3.6.1. Teoría de la Medición

a. Estructuras relacionales

Un sistema relacional es una $(p+q+1)$ -ada $\mathcal{U} = (A, R_1, R_2, \dots, R_p, \circ_1, \circ_2, \dots, \circ_q)$, donde A es un conjunto, R_1, R_2, \dots, R_p son relaciones sobre A (no necesariamente binarias) y $\circ_1, \circ_2, \dots, \circ_q$ son operaciones binarias sobre A , por ejemplo $\langle A, \succ, \circ \rangle$. Un sistema se conoce como sistema relacional numérico, si el conjunto A es el conjunto de los números reales, por ejemplo $\mathfrak{B} = \langle \mathfrak{R}, \succ, + \rangle$. Un mapeo f de un sistema relacional \mathcal{U} a otro \mathfrak{B} que preserve todas las relaciones y operaciones se llama Homomorfismo

La asignación numérica Φ es un *homomorfismo* en el sentido que envía A en \mathfrak{R} , \succ en $>$ y \circ en $+$ de tal forma que $>$ preserva las propiedades de \succ y $+$ preserva las propiedades de \circ .

Si $\langle A, R_1, R_2, \dots, R_m \rangle$ es una estructura relacional empírica y $\langle \mathfrak{R}, S_1, S_2, \dots, S_m \rangle$ es una estructura relacional numérica, una función Φ sobre A valuada en los reales es un homomorfismo si mapea cada R_i en S_i , $i=1, 2, \dots, m$. En general podemos tener n conjuntos A_1, A_2, \dots, A_n , m relaciones R_1, R_2, \dots, R_m sobre $A_1 \times A_2 \times \dots \times A_n$, y un homomorfismo valuado en vectores Φ cuyos componentes consisten de n funciones valuadas en los reales $\Phi_1, \Phi_2, \dots, \Phi_n$ con Φ_j definida sobre A_j , tal que Φ toma cada R_i en la relación S_i sobre \mathfrak{R}^n .

Un teorema de representación asegura que si una estructura relacional dada satisface ciertos axiomas, entonces puede construirse un homomorfismo en una cierta estructura relacional numérica.

La medición puede considerarse como la construcción de homomorfismos (escalas) de estructuras relacionales empíricas en estructuras relacionales numéricas de utilidad.

Las escalas donde las transformaciones permisibles son transformaciones $\Phi \rightarrow f(\Phi)$ donde f es una función de variable real estrictamente creciente valuada en los reales, se llama *escala ordinal*, solo el orden se preserva bajo estas transformaciones.

b. El problema de representación y unicidad.

El primer problema básico de la medición es el *problema de representación*: dado un sistema relacional numérico en particular \mathfrak{B} , encontrar condiciones necesarias y suficientes sobre un sistema relacional observado \mathcal{U} para la existencia de un homomorfismo de \mathcal{U} a \mathfrak{B} .

Si tenemos un homomorfismo, se dice que el homomorfismo da una representación y la triada $(\mathcal{U}, \mathfrak{B}, f)$ se llama *escala*.

El segundo problema básico de la teoría de la medición es el problema de unicidad. Un teorema de unicidad nos dice qué tipo de escala es f , también pone limitantes sobre las manipulaciones matemáticas que pueden desarrollarse sobre los números que surgen como valores de la escala.

Decimos que un enunciado que involucre escalas numéricas es significativo si y sólo si sus valores de verdad permanecen sin cambios si toda escala (U, B, f) implicada se cambia por otra escala aceptable $(\mathcal{U}, \mathcal{B}, g)$, esto es, permanecen sin cambios bajo todas las transformaciones admisibles de las escalas involucradas. La significancia puede estudiarse analizando las transformaciones admisibles de la escala.

3.6.2 Tipos de escala

La idea de definir el tipo de escala por medio de las clases de transformaciones admisibles se debe a S.S Stevens (1959).

El ejemplo más simple de escala es aquella donde la única transformación admisible es $\Phi(x)=x$. Con esta escala solo hay una manera de medir las cosas, esta escala se llama escala absoluta.

Supongamos que las transformaciones admisibles son todas las funciones $\Phi: f(A) \rightarrow f(B)$ de la forma $\Phi(x) = \alpha x, \alpha > 0$. Una escala con este tipo de transformaciones se llama Escala de Razón.

Cuando las clases de transformaciones admisibles son todas las funciones $\Phi: f(A) \rightarrow f(B)$ de la forma $\Phi(x) = \alpha x + \beta, \alpha > 0$. Φ se llama Transformación lineal positiva y su correspondiente escala se llama escala de intervalo.

Si la escala solo preserva el orden, las transformaciones admisibles son funciones monótonas crecientes $\Phi(x)$, que satisfacen la condición

$$x \geq y \Leftrightarrow \Phi(x) \geq \Phi(y),$$

o de forma equivalente

$$x > y \Leftrightarrow \Phi(x) > \Phi(y).$$

Estas escalas se llaman escalas ordinales. Las preferencias de los individuos son un ejemplo de escalas ordinales.

En algunas escalas, las transformaciones admisibles son todas las funciones Φ uno-uno. Esta escala se llama escala *Nominal*. La siguiente tabla muestra algunos ejemplos de los tipos de escala.

Tipo de escala	Transformaciones admisibles	Ejemplos
Absoluta	$\Phi(x) = x$, identidad	Conteo
Razón	$\Phi(x) = \alpha x, \alpha > 0$.	Masa Temperatura en la escala Kelvin Intervalos de tiempo Brillo Volumen (sonido)
Intervalo	$\Phi(x) = \alpha x + \beta, \alpha > 0$.	Temperatura (Fahrenheit, Centígrados, etc.) Tiempo (calendarios) Test de inteligencia
Ordinal	$x \geq y \Leftrightarrow \Phi(x) \geq \Phi(y)$	Preferencias Dureza Calidad del aire
Nominal	Cualquier Φ uno-uno	RFC CURP Números en los uniformes de jugadores

Tabla 1. Tipos de escala

Los tipos de escala enlistados en la tabla, van de la “mas fuerte” a la “más débil”, en el sentido de que la escala absoluta y la escala de razón contienen mucha más información que las escalas ordinal o nominal.

En general, las escalas ordinales pueden usarse para hacer comparaciones de la forma

$$f(a) > f(b)$$

Las escalas de intervalo pueden usarse para hacer comparaciones de diferencias

$$f(a) - f(b) > f(c) - f(d).$$

Y las escalas de razón para hacer comparaciones cuantitativas como

$$f(a) = kf(b), y$$

$$\frac{f(a)}{f(b)} = \lambda$$

3.6.3 Modelos de diferencia de valor

En las decisiones en grupo aparece en forma implícita o explícita algo que puede interpretarse como “fuerza de la preferencia”. Este tipo de relación binaria está definida no sobre el conjunto de alternativas sino sobre el conjunto de pares de alternativas y la denotamos por D . Sea la expresión $(a \leftarrow b)$ que representa el intercambio de b para recibir a . Aquí se debe hacer la distinción entre la preferencia sobre objetos \square y la preferencia sobre intercambios D . Así $(a \leftarrow b) D (c \leftarrow d)$ significa que intercambiar b por a es preferido que intercambiar d por c .

El propósito es encontrar una función de valor $v(\cdot)$ tal que

$$a \square b \leftrightarrow v(a) \geq v(b) \quad \text{y} \quad (a)$$

$$(a \leftarrow b) D (c \leftarrow d) \leftrightarrow v(a) - v(b) \geq v(c) - v(d). \quad (b)$$

Esta función se llama función de diferencia de valor o función de valor medible.

Desde luego que **(a)** requiere que una función de diferencia de valor sea una función de valor ordinal. Bajo ciertas condiciones, que a continuación enumeramos, entre las que se pide que D sea un orden débil, esta relación se puede representar por una función de *diferencia de valor* v sobre el conjunto de alternativas.

- Axioma 6.6.3.1. (orden débil):
 - \square cumple las condiciones de completitud, transitividad, consistencia de la preferencia estricta, preferencia débil e indiferencia sobre el conjunto de objetos.
 - D cumple las mismas condiciones sobre el conjunto de intercambios.
- Axioma 6.6.3.2 (consistencia de \square y D): para todo $a \in A$

$$a \square b \Leftrightarrow (a \leftarrow b) D (c \leftarrow c), \quad \forall c \in A.$$

- Axioma 6.6.3.3: para todo $a, b, c, d \in A$, se cumple

$$(a \leftarrow b) D (c \leftarrow d) \Leftrightarrow (d \leftarrow c) D (b \leftarrow a)$$

- Axioma 6.6.3.4: $\forall a, b, c, d, e, f \in A$

$$\left. \begin{array}{l} (a \leftarrow b) \quad D \quad (d \leftarrow e) \\ (b \leftarrow c) \quad D \quad (e \leftarrow f) \end{array} \right\} \Rightarrow (a \leftarrow c)D(d \leftarrow f)$$

Estos cuatro axiomas son necesarios pero no suficientes para construir una función de diferencia de valor, para esto se requiere considerar la siguiente condición de solubilidad:

- Axioma 6.6.3.5 (solubilidad):
 - $\forall b, c, d \in A, \exists a \in A$ tal que $(a \leftarrow b) \sim_D (c \leftarrow d)$
 - $\forall b, c \in A, \exists a \in A$ tal que $(b \leftarrow a) \sim_D (a \leftarrow c)$

En la construcción de la función de diferencia de valor se asume que dado cualquier a existe un entero n tal que $a_{(n+1)} \sqsupset a \sqsupset a_n$. De forma equivalente se supone que no existe un $a \in A$ tal que $a \sqsupset a_n, \forall n$. Este supuesto se conoce como el supuesto *arquimedeano*, y es una condición necesaria para la existencia de una función de diferencia de valor, sirve para establecer una unidad de preferencia definida por $(a_1 \leftarrow a_0)$: se construye una secuencia $a_0, a_1, a_2 \dots$ tal que $(a_n \leftarrow a_0)$ es “valioso n unidades”. Entonces se pide que para cualquier objeto a existe un entero N tal que $(a_N \leftarrow a_0)D(a \leftarrow a_0)$, lo que implica que $a_N \succ a$. De forma equivalente $(v(a) - v(a_0))$ no puede ser infinitamente mas grande que $(v(a_1) - v(a_0))$.

La secuencia $a_0, a_1, a_2 \dots$ se conoce como *secuencia estándar*. La propiedad que define a las secuencias estándar es que están igualmente espaciadas en algún sentido. Aquí $(a_n \leftarrow a_{(n-1)}) \sim_D (a_1 \leftarrow a_0)$ para todo n . Una secuencia estándar *estrictamente acotada* se define $\{a_n | b \succ a_n : (a_n \leftarrow a_{n-1}) \approx_D (a_1 \leftarrow a_0)\}$ para algún b fijo.

- Axioma 6.6.3.6 (arquimedeano): Toda secuencia estándar estrictamente acotada es finita.

Teorema.

Si se cumplen los axiomas 6.6.3.1- 6.6.3.6, entonces existe una función de diferencia de valor que representa a $\{a_n|b \succ a_n : (a_n \leftarrow a_{n-1}) \approx_D (a_1 \leftarrow a_0)\}$ y a D en el sentido de (a) y (b). Además estos axiomas son necesarios para tal representación.

Los métodos para construir con todo rigor una función de diferencia de valor requieren que la fuerza de la preferencia sobre el conjunto de alternativas constituya una estructura de medición de diferencias (French, 1988, §3.7; Krantz, *et al*, 1971, Cap 4; Roberts, 1979, §3.3), la cual incluye una condición de solubilidad que en los problemas prácticos de toma de decisiones en grupo es imposible de cumplir, ya que esta demanda que el conjunto de alternativas sea lo suficientemente rico en objetos. Se puede demostrar que la función de diferencia de valor v no es única, y que cualquier otra función w sobre A es equivalente a v si y sólo si ellas están relacionadas por una transformación afín positiva, por lo que la escala correspondiente es de intervalo.

Si se permite a cada miembro del grupo establecer su función de diferencia de valor para las alternativas, éstas se pueden agregar en una función de valor grupal de manera que sea consistente con respecto a los axiomas de Arrow.

Teorema (French, 1988)

Sean $v_1(\cdot), v_2(\cdot), \dots, v_n(\cdot)$ funciones de diferencia de valor de los N miembros del grupo y $v_g(a) = \varphi(v_1(a), v_2(a), \dots, v_n(a))$ una función diferenciable n-dimensional, tal que al menos dos derivadas parciales son positivas en todo su dominio y ninguna es negativa en ningún lado, entonces si se define la preferencia grupal \square_g por

$$a \square_g b \Leftrightarrow v_g(v_1(a), v_2(a), \dots, v_n(a)) \geq v_g(v_1(b), v_2(b), \dots, v_n(b))$$

resulta un procedimiento de agregación que satisface lo siguiente:

- i) Orden débil.- \square_g es un orden débil
- ii) No trivialidad.- \square_g está definida para cualquier número de alternativas
- iii) Dominio Universal.- \square_g está definida para cualesquiera funciones de diferencia de valor $v_1(\cdot), v_2(\cdot), \dots, v_n(\cdot)$ dadas por los miembros del grupo.
- iv) Relevancia binaria.- El orden de cualquier par de alternativas (reales) no depende de cuáles de las otras alternativas son factibles.

- v) Principio de Pareto.- Si para todo individuo, $v_i(a) > v_i(b)$, entonces $a \succ_g b$.
- vi) No dictadura.- No hay individuo tal que, para todos los pares de alternativas a, b , siempre que él prefiera a sobre b , el grupo cumpla $a \succ_g b$, sin importar lo que los otros miembros prefieran.

DEMOSTRACIÓN

- i) \square_g es un orden débil ya que se basa en el orden numérico de los valores de la función $v_g(v_1(a), v_2(a), \dots, v_n(a))$ asignado a las alternativas.
- ii) Trivial
- iii) Trivial
- iv) Para cualquier par de alternativas a, b , los números $v_g(v_1(a), v_2(a), \dots, v_n(a))$ y $v_g(v_1(b), v_2(b), \dots, v_n(b))$ no dependen de ninguna otra alternativa.
- v) Como todas las derivadas parciales son no negativas, $v(\dots, \dots)$ es no decreciente en todos sus argumentos. Además, como por lo menos dos derivadas parciales son positivas, es estrictamente creciente en algunos de sus argumentos. Así, $v_i(a) > v_i(b) \quad \forall i \Rightarrow v_g(v_1(a), v_2(a), \dots, v_n(a)) > v_g(v_1(b), v_2(b), \dots, v_n(b))$.
- vi) Suponga que para un par de alternativas, a, b , el individuo 1 tiene una ligera preferencia por a dada por $v_1(a) = v_1(b) + \delta$, donde δ es pequeña pero positiva. Se mostrará que para δ lo suficientemente pequeña, el grupo no necesita cumplir $a \square_g b$. Así el individuo 1 no es un dictador en el sentido establecido. Un argumento similar muestra que ningún otro individuo puede ser un dictador. Como al menos dos derivadas parciales son positivas, existe algún individuo $k \neq 1$ tal que $\frac{\partial v_g}{\partial v_k} > 0$. Suponga que k prefiere b sobre a : $v_k(b) = v_k(a) + \Delta$. Además, suponga que todos los otros miembros del grupo son indiferentes entre a y b : $v_i(a) = v_i(b), \forall i \neq 1 \text{ ó } k$. Entonces

$$v_g(v_1(b), v_2(b), \dots, v_n(b)) - v_g(v_1(a), v_2(a), \dots, v_n(a)) =$$

$$\frac{\partial v_g}{\partial v_1}(-\delta) + \frac{\partial v_g}{\partial v_k}\Delta + \text{términos de segundo orden} > 0$$

Para δ suficientemente pequeña y $\Delta = \left[1 + \frac{\partial v_g / \partial v_1}{\partial v_g / \partial v_k} \right] \delta$.

De esta manera, la preferencia marginal de 1 por a , puede ser rechazada por la preferencia de k por b .

Este teorema sugiere que si los miembros del grupo “votan” considerando la fuerza de sus preferencias sobre las distintas alternativas, por medio de funciones de diferencia de valor, entonces existen procedimientos de agregación que están de acuerdo con los axiomas de Arrow.

3.7. Conjuntos finitos de desigualdades lineales homogéneas.

El problema general de este apartado es encontrar solución a familias de desigualdades de la forma

$$\begin{aligned} \sum_{j=1}^n \alpha_{ij} x_j &> 0, \quad i = 1, 2, \dots, m' \\ \sum_{j=1}^n \beta_{ij} x_j &= 0, \quad i = 1, 2, \dots, m'' \end{aligned} \quad (1)$$

Tal familia de desigualdades y ecuaciones surge en contextos de medición cuando los x_1, x_2, \dots, x_n son valores desconocidos de cantidades a ser medidas y cada desigualdad del sistema de ecuaciones (1) está motivada, vía un modelo de medición lineal, por una observación correspondiente de orden o igualdad entre dos objetos. Cualquier solución al sistema (1) da una escala de medición compatible con las observaciones dadas de orden e igualdad. Los coeficientes α_{ij}, β_{ij} son enteros.

3.7.1. Explicación intuitiva del criterio de solución.

Se analizará el sistema (1) utilizando el método de vectores. La solución deseada del sistema (1) es un vector, y los coeficientes de cualquier desigualdad o ecuación del sistema también forman un vector $(\alpha_{i1}, \dots, \alpha_{in})$. Por conveniencia se representa al vector (x_1, x_2, \dots, x_n) con x , $(\alpha_{i1}, \dots, \alpha_{in})$ por α_i y $(\beta_{i1}, \dots, \beta_{in})$ por β_i .

El ángulo entre dos vectores x, y es el ángulo cuyo vértice es el origen $0 = (0, \dots, 0)$ y cuyos lados son las líneas desde el cero a través de cada vector. El producto escalar de dos vectores $x = (x_1, x_2, \dots, x_n)$ y $y = (y_1, y_2, \dots, y_n)$ simplemente es la suma de los productos de sus respectivas componentes, ie. $\sum_{i=1}^n x_i y_i$.

Se puede verificar que el ángulo entre x e y es agudo, obtuso o recto dependiendo de si su producto escalar es, respectivamente, positivo, negativo o cero. En términos de estos conceptos, el sistema puede reestablecerse como:

$$\begin{aligned} &x \text{ forma un ángulo agudo con cada } \alpha_i, i=1, 2, \dots, m' \\ &x \text{ forma un ángulo recto con cada } \beta_i, i=1, 2, \dots, m'' \end{aligned} \quad (1')$$

Suponga que $n=3$, $m''=2$ y β_1, β_2 son diferentes de cero y forman un ángulo no cero entre ellos. Los 3 puntos $0, \beta_1$ y β_2 yacen en un plano único, y si x resuelve (1'), entonces la línea desde cero hacia x debe estar en ángulo recto a ese plano. Así, la dirección de x a 0 está completamente determinada excepto por la elección del lado del plano sobre el que está x . Si elegimos x sobre un lado del plano, entonces hace un ángulo agudo con cualquier otro punto sobre el mismo lado, un ángulo recto con algún punto en el plano, y un ángulo obtuso con algún punto sobre el lado opuesto. Por tanto, es obvio que, en este caso, una solución al sistema (1') puede encontrarse si y sólo si todos los puntos α_i quedan sobre el mismo lado del plano determinado por $0, \beta_1$ y β_2 .

Mantengamos $n=3$ y consideremos los valores de m'' distintos de 2. Si $m'' \geq 3$, hay dos posibilidades. Si todos los β_i quedan en un solo plano (determinado por dos de ellos con 0), entonces tomamos x sobre una línea perpendicular a este plano en 0 , y la situación es como antes. De otra manera, no quedan en un solo plano, y así no puede encontrarse ninguna x que forme un ángulo recto con todas las β_i , por tanto el único vector x que satisface las ecuaciones es 0 , y este no satisface las desigualdades estrictas. De esta manera el sistema (1') no tiene solución (a menos que no haya desigualdades).

Si $m''=1$, entonces 0 y B , pueden ser parte de un número infinito de planos diferentes. Si todas las α_i quedan sobre el mismo lado de sólo uno de esos planos, entonces una x perpendicular a ese plano resuelve el sistema (1').

El caso extremo es $m''=0$, en el cual no hay igualdades, por el momento es el caso que nos ocupa. Esto significa que dos objetos no se pueden aceptar como exactamente iguales. Aquí es suficiente que haya algún plano a través del cero tal que todas las α_i queden sobre el mismo lado de él; no hay β_i que restrinja la posición del plano.

El criterio de solubilidad del sistema (1') si $m''=0$ es el siguiente: 0 debe ser exterior al poliedro generado por los α_i . Una manera de elegir una solución es elegir el punto sobre el poliedro que esté "más cercano" a 0. Esta es una solución x . A través de este punto podemos pasar un plano tangente al poliedro y el plano paralelo a través de 0 tiene al poliedro a un lado de él. La línea de 0 a x es perpendicular a ambos planos.

El análisis para $m''>0$ es similar. Si el plano a través de 0 se restringe a contener a β_1 ($m''=1$), entonces debemos pedir que el α -poliedro no interseque la línea a través de 0 y β_1 . Si el plano está determinado por β_1 y β_2 , entonces el α -poliedro no debe intersecar este plano.

Para $n>3$, el análisis es análogo al que se ha dado. La única diferencia es que hay más posibilidades para el número de β_i independientes, distintas de 0 1 ó 2. Si hay $n-1$ vectores igualdad β_i , entonces ellos determinan un hiperplano $n-1$ dimensional, al que la solución x debe ser perpendicular. Todos los α_i deben quedar sobre el mismo lado de este hiperplano. En general cualquier número de β_i independientes, que no excedan $n-1$, generan un subespacio, y este subespacio restringe parcialmente la posición para un hiperplano con el poliedro de vectores desigualdad que quedan sobre un lado de él. El teorema general es el siguiente:

Teorema .

El sistema (1') tiene una solución si sólo si el poliedro generado por los vectores α_i no interseca el subespacio generado por los vectores β_i (cuando $m''=0$, el subespacio en cuestión se toma igual a 0). En otras palabras el sistema

$$\begin{aligned} \alpha_i \cdot x &> 0, i = 1, \dots, m \\ \beta_i \cdot x &= 0, i = 1, \dots, m \end{aligned}$$

tiene solución si la cubierta convexa de $\alpha_1, \dots, \alpha_m$ no interseca al subespacio generado por $0, \beta_1, \dots, \beta_m$.

La demostración del teorema es constructiva y se basa en el siguiente lema

Lema .

Sea $\alpha_i = (\alpha_{i1}, \dots, \alpha_{in})$ y suponga que $\alpha_1, \dots, \alpha_m$ son linealmente independientes: entonces para cualesquiera número t_1, \dots, t_m las ecuaciones

$$\alpha_i \cdot x = t_i, i = 1, \dots, m$$

tienen una solución $x = (x_1, \dots, x_n)$.

Esto es, siempre es posible resolver m ecuaciones $\alpha_i \cdot x = t_i$, con n incógnitas $x = (x_1, \dots, x_n)$, dado que las m expresiones lineales $\sum \alpha_{ij} x_j$ son independientes una de la otra y así, en particular, $m \leq n$.

La demostración del teorema se basa en la existencia de una demostración constructiva del lema. Para construir una solución $x = (x_1, \dots, x_n)$, se usa la primera ecuación para expresar una de las x_i en términos de las otras $n-1$, sustituyendo la expresión para x_i en las otras $m-1$ ecuaciones, reduciendo el problema a $m-1$ ecuaciones con $m-1$ incógnitas. Este proceso continúa hasta que no haya ecuaciones, o hasta que uno de los resultados de la sustitución resulte en una contradicción, en cuyo caso, el proceso se detiene y puede mostrarse que el α_i original no era independiente. (ver Krantz, 1971).

4. ALGUNOS MÉTODOS DE VOTACIÓN

4.1. El método de agregación de Borda.

En 1781 el Caballero Jean-Charles de Borda (1733-1799) formuló el sistema de votación de ordenamiento preferencial para comités formados por varias personas, conocido como *conteo de borda*. La Academia Francesa de Ciencias utilizó el método de Borda para elegir a sus miembros durante dos décadas y todavía se utiliza en nuestros días en instituciones educativas, competencias y en política.

La idea central del método consiste en sumar los rangos obtenidos por una alternativa con respecto a cada uno de los individuos del grupo. En lo sucesivo supondremos, por tanto, que tratamos con n alternativas y que elegimos n enteros $k_1 > k_2 > k_3 > \dots > k_n \geq 0$, que se llaman *coeficientes de Borda*. Para cada individuo j , las alternativas se clasifican según un preorden (o cuasiorden) completo, y se denomina r_{ij} al rango de la alternativa i para el preorden asociado al individuo j . Las alternativas forman un camino de preferencias del tipo: $a_{i1} \succ a_{i2} \succ a_{i3} \approx a_{i4} \succ \dots \succ a_{im-1} \succ a_{im}$ donde \succ designa la preferencia estricta y \approx la indiferencia. Llamamos rg a la función que asocia k_1 con a_1 , k_2 con a_2 , y así sucesivamente cuando solo hay preferencias estrictas. Si hay empates (indiferencias), se atribuye a cada una de las alternativas empatadas la media de los coeficientes que habrían obtenido si no lo estuvieran (rango promedio de Kendall).

Una vez definida para cada individuo j la función rg , se calcula $b(a_i) = \sum_j (rg_j(a_i))$, que es la función de agregación de preferencias. Es decir, que la alternativa clasificada como primera será aquella que obtenga más puntos, siguiéndola las demás en orden decreciente de puntos. En caso de igualdad de puntos, las alternativas se consideran indiferentes en el preorden grupal.

Ejemplo: Consideremos cuatro alternativas a , b , c y d , y tres individuos C_1 , C_2 y C_3 , que presentan los siguientes perfiles de preferencia

individuo 1: $a \succ_1 b \succ_1 c \succ_1 d$

individuo 2: $a \succ_2 b \succ_2 d \succ_2 c$.

individuo 3: $b \succ_3 c \succ_3 d \succ_3 a$.

cada individuo asigna una evaluación, es decir los llamados los *coeficientes canónicos de Borda* $4 > 3 > 2 > 1$ para las cuatro alternativas:

	C ₁	C ₂	C ₃	<i>b</i> (.)
<i>a</i>	4	4	1	9
<i>b</i>	3	3	4	10
<i>c</i>	2	1	3	6
<i>d</i>	1	2	2	5

Tabla 1. Coeficientes de Borda

De esta manera, el orden grupal es: $b \succ_g a \succ_g c \succ_g d$.

El método de Borda es un método ordinal que no satisface el axioma de la alternativa irrelevante.

4.2. El método de Condorcet.

El método de Condorcet se llama así por el Marqués de Condorcet (1743-1794). Su obra más importante fue *Essay on the Application of Analysis to the Probability of Majority Decisions* (1785). Es conocido por la Paradoja de Condorcet que establece que la votación por mayoría simple puede dar como resultado configuraciones de preferencia grupal no transitivas y propuso un método de comparación por pares en 1785.

El método de Condorcet de comparación por pares simula una serie de elecciones entre cada alternativa y las demás. Si hay una alternativa que gane todas las comparaciones por pares, se llama Ganador de Condorcet, y es la que se elige por este método. Por ejemplo, supongamos que una sociedad va elegir entre tres candidatos {D, C, I} y para esto ordena sus preferencias de la siguiente manera:

36 votantes opinan que $D \succ C \succ I$

15 votantes: $C \succ D \succ I$

15 votantes: $C \succ I \succ D$

34 votantes: $I \succ C \succ D$

la comparación por pares da como resultado:

(D, C): D obtiene 36 votos mientras que C obtiene 64, gana C

(D, I): D obtiene 51 votos e I sólo 49, gana D

(C, I): C tiene 66 votos e I solo 34, gana C

Así, el orden grupal es $C \succ D \succ I$.

La existencia de un ganador de Condorcet es muy posible, aunque, también es posible que exista un ciclo en la regla de la mayoría, esta situación se conoce como la PARADOJA DEL VOTO: sean tres individuos con tres alternativas $\{a, b, c\}$, los cuales cumplen las siguientes preferencias estrictas:

individuo 1: $a \succ_1 b \succ_1 c$.

individuo 2: $b \succ_2 c \succ_2 a$.

individuo 3: $c \succ_3 a \succ_3 b$.

Utilizando \succ_g para indicar la preferencia estricta para el grupo, la regla de la mayoría simple conduce a lo siguiente:

$a \succ_g b$ (2 de 3 prefieren a sobre b)

$b \succ_g c$ (2 de 3 prefieren b sobre c)

$c \succ_g a$ (2 de 3 prefieren c sobre a)

Así tendríamos que si $a \succ_g b$, y $b \succ_g c$, debería ocurrir que $a \succ_g c$, y la configuración de preferencias nos muestra que $c \succ_g a$, por tanto la preferencia del grupo resulta intransitiva, a , b y c forman un ciclo.

Otro problema es que si se sigue un orden determinado en la votación, el resultado final depende del orden en que se hace la votación, digamos que primero se “vota” a a contra b y resulta ganadora a y al enfrentar a contra c , se llega a la conclusión de que la ganadora global es c ; si empezamos comparando b contra c , gana b y ésta contra a , ahora gana a ; además no hay ninguna razón para decidirse por un orden en particular.

4.3. Voto aprobatorio.

En el voto aprobatorio (Brams y Fishburn, 1978) se puede votar solo una vez por cada alternativa, pero se puede votar por tantas alternativas como se quiera. Esto es equivalente a decir que cada individuo puede “aprobar” o “desaprobar” cada opción votando o no votando por ella. La alternativa ganadora será la que tenga más votos. Por ejemplo, si A , B , C , D , y E están participando en una elección, se puede votar sólo por A , o por A y B , o por A , B y C , incluso por las cinco alternativas si así se desea, pero esto equivale a no votar, ya que los votos afectan a los candidatos de la misma forma.

El voto aprobatorio es probablemente una mejora de la pluralidad por una sencilla razón, la pluralidad fuerza a los individuos a elegir entre votar por una alternativa que se percibe viable o por una alternativa con la que se está fuertemente de acuerdo, mientras que el voto aprobatorio permite hacer ambas cosas. Este método satisface el axioma de independencia de la alternativa irrelevante, en gran parte porque es un método cardinal y votar por una alternativa no prohíbe votar por otra.

4.4. Voto por veto.

Se establece un orden en el que los individuos pueden ejercer su veto, Cada individuo va eliminando sucesivamente la alternativa que menos desea entre las no eliminadas hasta su turno, se procede así hasta que queda una sola alternativa, la elegida. Para que el método tenga sentido debe haber una alternativa más que el número de individuos. (Mueller (1978), Barberá, (1984))

5. EL TEOREMA DE IMPOSIBILIDAD DE ARROW.

Kenneth Arrow (1963) plantea el problema de elección social axiomáticamente como el de elegir una constitución, es decir, una regla que asigne un ordenamiento de preferencias grupal a cada una de las posibles configuraciones de preferencias individuales. Arrow propone 5 axiomas intuitivamente satisfactorios que debería cumplir todo procedimiento de agregación de preferencias individuales.

Antes de enunciar el Teorema de Arrow y los axiomas propuestos por él mismo, es necesario introducir algunas definiciones para la demostración del teorema.

En lo subsecuente A denota al conjunto de alternativas, cuya cardinalidad es al menos 3, y $n \geq 2$ es el número de individuos en el grupo.

Un subconjunto V de individuos es *decisivo* para a sobre b si, siempre que $a \succ_i b$ para todo $i \in V$ y $b \succ_{i'} a$ para todos los $i' \notin V$, el grupo cumple $a \succ_g b$. En otras palabras, bajo la constitución que construye \succ_g a partir de las \succ_i individuales, el subconjunto V es decisivo para a sobre b si, cuando sean unánimes en su preferencia de a sobre b , tienen el poder de forzar esta preferencia como la preferencia del grupo como un todo, a pesar de las preferencias estrictas de los miembros del grupo que no están en V para b sobre a . Un subconjunto V es un *subgrupo decisivo minimal* si V es decisivo para cualquier a sobre cualquier b y ningún subgrupo propio de V es decisivo para cualquier c sobre cualquier d . Note que un subgrupo decisivo minimal siempre existe bajo los axiomas de Arrow.

El primer axioma pide que la constitución sea de carácter universal, esto es, que sea posible agregar todas las configuraciones de preferencias de los miembros del grupo.

- Axioma 1. \succ_g está definida para cualquier $\succ_i, i=1, \dots, n$.

Tratando de evitar la paradoja del voto Arrow impone las condiciones de completitud y transitividad de la preferencia grupal

- Axioma 2. $\succsim_1, \succsim_2, \dots, \succsim_n$ y \succsim_g son ordenes débiles, es decir, $\succsim_1, \succsim_2, \dots, \succsim_n$ y \succsim_g , son completos y transitivos.

El siguiente axioma condiciona el respeto a las decisiones por unanimidad. Es decir, si todos y cada uno de los individuos del grupo piensan que la alternativa a es mejor que la alternativa b , la preferencia grupal debe respetar que a es mejor que b .

- Axioma 3. Si cada individuo cumple $a \succ_i b$, entonces el grupo cumple $a \succ_g b$.

Esta condición se conoce como el principio de Pareto y nos asegura que el grupo completo es decisivo para todo a sobre todo b .

La siguiente condición impone que el orden grupal entre dos alternativas sólo debe depender de cómo los individuos ordenan a esas dos alternativas y debe ser independiente de cómo los individuos ordenan al resto de las alternativas. Esta condición es la que más exige y en la que en realidad radica toda la fuerza del resultado (Plata, 1996).

- Axioma 4. Sean $\succsim_1, \succsim_2, \dots, \succsim_n$ un conjunto de órdenes de preferencia individual sobre A . Sean $\succsim'_1, \succsim'_2, \dots, \succsim'_n$ otro conjunto de órdenes de preferencia individual sobre el conjunto de alternativas A' . Supongamos que las alternativas a y b cumplen:

$$\{a, b\} \subset A \cap A'$$

Supongamos además que $\succsim_1, \succsim_2, \dots, \succsim_n$ y $\succsim'_1, \succsim'_2, \dots, \succsim'_n$ son idénticas sobre $\{a, b\}$:
 $\forall i,$

$$a \succsim_i b \leftrightarrow a \succ'_i b.$$

$$b \succsim_i a \leftrightarrow b \succ'_i a.$$

Entonces la constitución nos debe llevar a la misma preferencia grupal entre a y b :

$$a \succsim_g b \leftrightarrow a \succ'_g b \quad \text{y}$$

$$b \succsim_g a \leftrightarrow b \succ'_g a.$$

La última condición establece el carácter no dictatorial de la constitución, esto es, que el orden grupal refleje el gusto de todos los individuos y no los de un solo individuo en particular quien podría imponer su preferencia individual como preferencia grupal.

- Axioma 5. No existe un individuo $i \in N$ tal que para cualquier par $(a, b) \in A$ se tenga que $a \succ_i b$ implique $a \succ_g b$.

Arrow demuestra en su famoso teorema de Imposibilidad que no existe constitución alguna que pueda satisfacer simultáneamente los cinco axiomas, esto es, los axiomas son mutuamente inconsistentes.

TEOREMA DE IMPOSIBILIDAD DE ARROW.

No existe una constitución que permita que la preferencia grupal \succ_g sea definida a partir de las preferencias individuales $\succ_1, \succ_2, \dots, \succ_n$ de una manera consistente con los axiomas 1,2,3,4 y 5.

DEMOSTRACIÓN¹.

La idea de la demostración es analizar los conjuntos de individuos decisivos. La condición de Pareto nos dice que el conjunto de todos los individuos es decisivo, mientras que la condición de independencia de la alternativa irrelevante y la transitividad de la preferencia grupal nos ayudan a ver que si dividimos un conjunto decisivo en dos, alguna de las partes continúa siendo decisiva. En un grupo finito podemos hacer N particiones sucesivamente hasta llegar a un conjunto con un solo individuo que resulta ser el dictador.

Supongamos que V es un subconjunto mínimo decisivo para a sobre b . Sea j un individuo específico en V , W es el conjunto de los individuos restantes en V y U el conjunto de aquellos individuos que no están en V . Considere los siguientes órdenes de preferencia para j, W, V y U :

$$V \begin{cases} j : a \succ b \succ c \\ W : c \succ a \succ b \end{cases} \quad U : b \succ c \succ a$$

Como V es decisivo para a sobre b y $a \succ_i b$, entonces debemos tener que $a \succ_g b$. Como W es un subconjunto propio de V entonces no puede ser decisivo para ningún par de alternativas. De esta manera no podríamos tener $c \succ_g b$, porque entonces W sería decisivo para c sobre b . Así, $b \succ_g c$. Como \succ_g es un orden débil, entonces $a \succ_g b, b \succ_g c \rightarrow a \succ_g c$. Pero sólo j cumple que $a \succ_j c$, el resto de los integrantes de V cumplen que

¹ La prueba que se presenta está basada en el texto de French (1988). Existen otras versiones similares a la demostración original de Arrow en Sen (1979,1984) y en Mas-Collel (1995)

$c \succ_i a$. Así j es decisiva para a sobre c , lo que contradice la naturaleza minimal decisiva de V . La única manera de evitar esta contradicción sin violar explícitamente alguno de los axiomas de Arrow es aceptando que V está conformado por una sola persona.

Ahora mostraremos que j , quien es decisivo para un a en particular sobre un b en particular, es un dictador, es decir, que es decisivo para algún c sobre algún d . Primero mostramos que j es decisivo para algún a sobre algún d . Considere la siguiente situación; U comprende a los elementos del grupo diferentes de j

$$\begin{aligned} j &: a \succ b \succ d \\ U &: b \succ d \succ a, \\ & d \neq b \end{aligned}$$

j es decisivo para a sobre b ; así $a \succ_g b$. Todos cumplen que $b \succ_i d$, entonces por el principio de Pareto, $b \succ_g d$. Como \succ_g es transitiva $a \succ_g d$. Pero sólo j cumple que $a \succ_j d$; todos lo demás cumplen $d \succ_i a$. De esta manera j es decisivo para a sobre cualquier d . A continuación se muestra que j es decisivo para cualquier $c \neq a$ sobre cualquier $d \neq a$. Considere la siguiente situación:

$$\begin{aligned} j &: c \succ a \succ d \\ U &: d \succ c \succ a, \\ & d \neq a \end{aligned}$$

$$\text{Principio de Pareto} \quad \rightarrow c \succ_g a.$$

$$j \text{ es decisivo para } a \text{ sobre } b \rightarrow a \succ_g d$$

$$\succ_g \text{ transitiva} \quad \rightarrow c \succ_g d.$$

Pero sólo j cumple que $c \succ_j d$; todos los demás cumplen que $d \succ_i c$. Por tanto j es decisivo para c sobre d .

Finalmente, se mostrará que j es decisivo para cualquier c sobre a . Considere:

$$\begin{aligned} j &: c \succ d \succ a \\ U &: d \succ a \succ c, \\ & d \neq a \end{aligned}$$

$$j \text{ es decisivo para } c \text{ sobre } d \rightarrow c \succ_g d.$$

$$\text{Principio de Pareto} \quad \rightarrow d \succ_g a.$$

$$\succ_g \text{ transitiva} \quad \rightarrow c \succ_g a.$$

Pero sólo j cumple $c \succ_j a$; todos los demás cumplen que $a \succ_i c$. Por tanto j es decisivo para cualquier c sobre a . Por tanto j es un dictador, contradiciendo el axioma 5. De esta manera, los cinco axiomas son mutuamente inconsistentes. Ninguna Constitución puede satisfacerlos simultáneamente.

Analizando la independencia de las condiciones, es fácil ver que la regla de la mayoría satisface todos los axiomas excepto el axioma 2 de transitividad del orden grupal. Esta condición no se deduce de las otras y es una prueba de su independencia con respecto a las condiciones de No dictadura, Pareto e independencia de la alternativa irrelevante. El conteo de Borda satisface todas las condiciones excepto la independencia de la alternativa irrelevante, por tal razón, esta condición es independiente de las demás. El teorema afirma implícitamente que cualquier regla que satisfaga los axiomas 2, 3 y 4 es dictatorial por lo que el axioma de no dictadura también es independiente de ellas. Por último la independencia de la condición de Pareto, se prueba con la regla que siempre asigna un orden constante a cualquier perfil que se le presente.

También están implícitas ciertas condiciones que son importantes. La estructura informacional de la condición de independencia de la alternativa irrelevante y de la regla de agregación de preferencias es ordinal y no comparable. Es decir, sólo se utiliza la información que se obtiene de los órdenes individuales para obtener el orden grupal. De acuerdo con Sen (1976) no podemos suponer que los ordenamientos de preferencias reflejan los intereses, el bienestar o que describen el comportamiento de elección de los individuos. Se requiere de una estructura informacional más elaborada que permita comparar y hacer juicios sobre los ordenamientos de las alternativas y así poder tener una idea de la cardinalidad de sus preferencias.

6. CONSTITUCIÓN DE SEGUNDO ORDEN.

Sen (1974,1976, 1979) y Van der Veen (1981) sugieren algunas interpretaciones alternativas de comparabilidad interpersonal basadas en ordenamientos de “orden superior”, donde es posible definir un ordenamiento no directamente sobre el espacio de alternativas A , sino sobre los ordenamientos de las alternativas, estos meta-ordenamientos pueden aplicarse al problema de encontrar una medida significativa de la utilidad cardinal. Así, la información adicional se podrá obtener a partir del concepto de preferencias de segundo orden consistente en que el individuo ordene los posibles ordenamientos de las alternativas, es decir, que se tome en cuenta la preferencia de cada miembro del grupo sobre el conjunto de los ordenamientos sobre A , interpretados como posibles resultados de aplicar la regla de la constitución.

Nuestra investigación se centra en una constitución de función aditiva en la que cada individuo $i \in I$ del grupo expresa su función de evaluación en un subconjunto cerrado y acotado de números reales, Y , con más de un elemento, común para todos los miembros del grupo, $v_i: A \rightarrow Y$; la preferencia del grupo P_g está dada por

$$(a,b) \in P_g \Leftrightarrow v_g(a) \geq v_g(b), \quad \forall a,b \in A, \quad (1)$$

donde v_g está dada por $v_g(a) = \sum_{i \in I} v_i(a)$.

Se define una *constitución de segundo orden* como aquella que toma en cuenta la preferencia de cada miembro i del grupo sobre el conjunto de los órdenes débiles sobre A , interpretados como posibles resultados P_g de la elección de grupo. Para representar estas preferencias adoptamos un conjunto de referencia, común para todos los miembros del grupo, denotado por $O(A)$, formado por todos los ordenamientos posibles del conjunto A en orden decreciente en preferencia, cada uno de los cuales tiene la forma $a_1 \succeq a_2 \succeq \dots \succeq a_m$. De esta manera, cada miembro i del grupo tiene asociada una relación binaria \succeq_i sobre $O(A)$, respecto de la cual P_i será estrictamente el mejor elemento de $O(A)$. si es que P_i no contiene indiferencias, y si P_i contiene alguna indiferencia, entonces P_i está representado por un subconjunto de $O(A)$, que por simplicidad se denota también por $P_i \subseteq O(A)$, cuyos elementos coinciden con P_i en las preferencias estrictas en P_i , y que difieren entre sí respecto de las indiferencias en P_i , justamente para que la cerradura transitiva de dicho subconjunto sea idéntica al orden débil P_i . En este

último caso el miembro i es indiferente entre los elementos de $O(A)$ contenidos en $P_i \subseteq O(A)$ y él prefiere estrictamente los elementos que se encuentran en $P_i \subseteq O(A)$ sobre los otros elementos de $O(A)$. Más precisamente, la relación \succeq_i determina totalmente a P_i , como elemento o subconjunto de $O(A)$, según el caso, a través de

a) caso $P_i \in O(A)$: $P_i \in O(A)$ es el único elemento que cumple $P_i \succeq_i o', \forall o' \in O(A)$

b) caso $P_i \subseteq O(A)$: $P_i = \{o: o \succeq_i o', \forall o' \in O(A)\}$

P_i como orden débil sobre A se llama *primera opción* del individuo i sobre $O(A)$. Este orden débil es en el primer caso el propio elemento $P_i \in O(A)$ y en el segundo es la cerradura transitiva del subconjunto $P_i = \{o: o \succeq_i o', \forall o' \in O(A)\} \subseteq O(A)$. Por tanto el conjunto de preferencias sobre $O(A)$, $\{\succeq_1, \succeq_2, \dots, \succeq_n\}$, denominado *perfil de preferencias de segundo orden (del grupo)* sobre A , determina unívocamente el perfil de preferencias de primer orden sobre A , $\{P_1, P_2, \dots, P_n\}$.

Una constitución es de *segundo orden* si la información preferencial que ella procesa de cada miembro i del grupo determina unívocamente un orden débil sobre $O(A)$, \succeq_i , de manera que las respectivas preferencias $\succeq_1, \succeq_2, \dots, \succeq_n$ sobre $O(A)$ y la elección del grupo P_g cumplen la siguiente condición paretiana, la cual expresa de manera formal que en la elección de grupo se toman en cuenta las preferencias $\succeq_1, \succeq_2, \dots, \succeq_n$:

Optimalidad de Pareto de P_g respecto del perfil de preferencias de segundo orden: Si la elección del grupo $P_g \in O(A)$, o en su caso cualquier elemento de $P_g \subseteq O(A)$, se sustituye por otro elemento de $O(A)$, entonces al menos un miembro i del grupo pierde estrictamente en el sentido de su relación \succeq_i sobre $O(A)$.

Se dice que una constitución de segundo orden *cumple las condiciones de Arrow* si el correspondiente perfil de primer orden $\{P_1, P_2, \dots, P_n\}$ y la elección del grupo P_g cumplen las condiciones 1, 2, 3, 4, 5.

La preferencia del grupo sobre A , dada por (1) se puede expresar como

$$(a,b) \in P_g \Leftrightarrow \sum_{i \in I(a,b)} (v_i(a) - v_i(b)) \geq \sum_{i \in I(b,a)} (v_i(b) - v_i(a)) \quad \forall a,b \in A, \quad (2)$$

donde $I(a,b) \subseteq I$ es el conjunto de los miembros del grupo que prefieren a sobre b ,

$$I(a,b) = \{i: v_i(a) > v_i(b), a,b \in A\}, \quad \forall a,b \in A. \quad (3)$$

La expresión (2) admite una interpretación similar a la regla de la mayoría simple, solo que ahora en vez de comparar la cardinalidad de los conjuntos $I(a,b)$ e $I(b,a)$ se comparan las respectivas sumas de la fuerza de la preferencia: para los individuos en la clase $I(a,b)$ se suman las diferencias $v_i(a) - v_i(b)$ y para la clase $I(b,a)$ se suman las diferencias $v_i(b) - v_i(a)$, la clase cuya suma es mayor impone la preferencia del grupo entre a y b . Entonces, cada individuo que prefiere a sobre b contribuye no con un voto sino con el número real $v_i(a) - v_i(b)$, que podría llamarse *magnitud del voto* del individuo i para que el grupo elija que a es preferible a b , (es decir, para que $(a,b) \in P_g$). En otras palabras, la preferencia del grupo para cada par de alternativas a, b , sobre el conjunto de elección $\{(a,b), (b,a)\}$, está dada por la función de valor ordinal con dos argumentos, $w_g(\cdot, \cdot)$,

$$w_g(a,b) = \sum_{i \in I(a,b)} (v_i(a) - v_i(b)), \quad w_g(b,a) = \sum_{i \in I(b,a)} (v_i(b) - v_i(a)) \quad \forall a,b \in A.$$

Una constitución tipo A contiene de manera implícita un sistema de votaciones, que además de los conjuntos de elección de la forma $\{(a,b), (b,a)\}$ antes considerados, incluye también el conjunto de elección $O(A)$: se puede mostrar que la magnitud del voto $w_i(o)$ del individuo i para que el elemento $o \in O(A)$ sea seleccionado como ordenamiento del grupo, es igual a la suma de las magnitudes de los votos que el individuo i asigna a cada una de las parejas ordenadas que pertenecen a dicho elemento, $(a,b) \in o$, es decir,

$$w_i(o) = \sum_{(a,b) \in o \cap P_i} (v_i(a) - v_i(b)) \quad \forall o \in O(A), \quad (4)$$

donde P_i es el orden débil sobre A asociado a la función ordinal de valor v_i , dado por

$$(a,b) \in P_i \Leftrightarrow v_i(a) \geq v_i(b), \quad \forall a,b \in A. \quad (5)$$

De acuerdo con lo anterior, en una constitución tipo A cada función de evaluación v_i determina una preferencia P_i sobre A dada por (5) y una preferencia \succeq_i sobre $O(A)$ dada por

$$o \succeq_i o' \Leftrightarrow \sum_{(a,b) \in o \cap P_i} (v_i(a) - v_i(b)) \geq \sum_{(a,b) \in o' \cap P_i} (v_i(a) - v_i(b)), \quad \forall o, o' \in O(A), \quad (6)$$

o de forma equivalente,

$$o \succeq_i o' \Leftrightarrow w_i(o) \geq w_i(o'); \quad (7)$$

Se dice entonces que $\succeq_i (P_i)$ es la *preferencia revelada* o *representada* por v_i sobre $O(A)$ (sobre A , respectivamente).

Por (7) una constitución de función aditiva se puede interpretar como un sistema de votaciones sobre el conjunto de órdenes débiles sobre A , en el que por (4) el voto de cada función de diferencia de valor v_i , a favor de un determinado orden débil sobre A , es igual a la suma de las fuerzas de preferencia para aquellos pares de alternativas cuyo orden según v_i y según el orden débil considerado coinciden.

Una relación binaria sobre $O(A)$, \succeq_i , es *acorde* con una función de diferencia de valor sobre A si existe una función real sobre A , v_i , tal que (6), en cuyo caso se dice que \succeq_i es *acorde* con la función (de diferencia de valor) v_i y que v_i es una función (de diferencia de valor sobre A) que *representa* a \succeq_i . Aquí vale la pena hacer la observación de que “función de diferencia de valor” se refiere a cualquier función que representa la preferencia de un individuo sobre $O(A)$, mientras que “función de evaluación” se refiere a una función de diferencia de valor que está escalada de acuerdo con la constitución del grupo.

Antes del siguiente teorema, que le da sentido a las definiciones anteriores, conviene hacer explícitas propiedades de la preferencia revelada sobre $O(A)$.

Lema 1. En una constitución tipo A cada preferencia \succeq_i sobre $O(A)$ cumple:

- a) \succeq_i es un orden débil sobre $O(A)$ cuya función de valor ordinal es w_i ,
- b) La primera opción de \succeq_i sobre $O(A)$ es el orden débil P_i sobre A dado por (5), y

$$w_i(P_i) = \sum_{(a,b) \in P_i} (v_i(a) - v_i(b)); \quad (8)$$

c) \succeq_i es invariante bajo transformaciones afines de v_i .

Demostración. a) Evidente de (6). b) De (4) resulta directamente (8); P_i hace máxima $w_i(\cdot)$, y por tanto es la primera opción de \succeq_i sobre $O(A)$, pues la sumatoria en (8) incluye todas las parejas (a,b) en que $v_i(a) > v_i(b)$, c) Inmediato. \square

Teorema 1. En una constitución tipo A :

a) El orden débil del grupo, P_g , es la cerradura transitiva de los elementos en $O(A)$ que hacen máxima la función sobre $O(A)$, w_g , dada por

$$w_g(o) = \sum_{i \in I} w_i(o), \quad \forall o \in O(A), \quad (9)$$

donde las w_i están dadas por (4), cumpliéndose

$$w_g(o) = \sum_{(a,b) \in o} \sum_{i \in I(a,b)} (v_i(a) - v_i(b)), \quad \forall o \in O(A); \quad (10)$$

b) Se cumple la condición de *Optimalidad de Pareto*: la elección del grupo P_g es un óptimo de Pareto respecto de las preferencias reveladas \succeq_i sobre $O(A)$.

c) Se cumplen las condiciones de *Arrow*.

d) Si las funciones de evaluación de todos los miembros del grupo sufren transformaciones afines positivas, entonces para que las desigualdades

$$v_i(a) - v_i(b) \geq v_j(c) - v_j(d) \quad \forall i, j \in I, a, b, c, d \in A, \quad (11)$$

el ordenamiento P_g del grupo y la relación de preferencia \succeq_g del grupo sobre $O(A)$ no cambien se requiere que dichas transformaciones afines sean idénticas (las mismas constantes multiplicativa y aditiva).

Demostración. a) Sustituyendo (4) en (9) se obtiene directamente (10):

$$w_g(o) = \sum_{i \in I} \sum_{(a,b) \in o \cap P_i} (v_i(a) - v_i(b)) = \sum_{(a,b) \in o} \sum_{i \in I(a,b)} (v_i(a) - v_i(b)).$$

Por (2) el valor de $w_g(\cdot)$ sobre $O(A)$ dado por (10) es máximo para un cierto elemento $o \in O(A)$ si y sólo si $(a,b) \in o$ siempre que $v_g(a) > v_g(b)$; o es igual a $P_g \in O(A)$ o es un elemento de $P_g \subseteq O(A)$; por tanto P_g es el menor orden débil que contiene a todos los elementos de $O(A)$ que hacen máxima la función $w_g(\cdot)$.

b) Es suficiente mostrar por contradicción que para cada $o \in O(A)$ distinto de P_g (o no contenido en P_g), hay al menos un miembro i del grupo tal que $P_g \succ_i o$. Sea el caso $P_g \in O(A)$ y sea $o \in O(A)$ tal que $o \succ_i P_g$ para toda i , donde $o \succ_i P_g$ para alguna i . Entonces, por (6),

$$\sum_{(a,b) \in o \cap P_i} (v_i(a) - v_i(b)) \geq \sum_{(a,b) \in P_g \cap P_i} (v_i(a) - v_i(b)) \quad \forall i \in I,$$

siendo estricta alguna de las desigualdades. Sumando ordenadamente para ambos lados de esta desigualdad para todo $i \in I$,

$$\sum_{i \in I} \sum_{(a,b) \in o \cap P_i} (v_i(a) - v_i(b)) > \sum_{i \in I} \sum_{(a,b) \in P_g \cap P_i} (v_i(a) - v_i(b))$$

es decir,

$$\sum_{(a,b) \in o} \sum_{i \in I(a,b)} (v_i(a) - v_i(b)) > \sum_{(a,b) \in P_g} \sum_{i \in I(a,b)} (v_i(a) - v_i(b)).$$

Por (10) esta última desigualdad equivale a $w_g(o) > w_g(P_g)$, lo cual contradice a); el caso $P_g \subseteq O(A)$ se demuestra de manera similar sustituyendo P_g por $o' \in P_g$. c) Ver Keeney (1976) y French (1988). d) Se demuestra directamente haciendo las sustituciones correspondientes. \square

Entonces, por el teorema 1 las constituciones tipo A cumplen la condición paretiana. Por tanto, en condiciones de ignorancia sobre las preferencias de los otros individuos, a cada miembro del grupo le conviene manifestar la función de evaluación que mejor represente su verdadera preferencia sobre $O(A)$. Para esto, un individuo que cumpla la siguiente condición de independencia puede obtener una función de evaluación más refinada si en vez de restringirse al conjunto A obtiene su función de diferencia de valor sobre un conjunto B más amplio, que además de A contiene alternativas no factibles: La restricción de esta función al conjunto A sería la que el individuo manifestaría al grupo.

Condición de Independencia de las funciones de diferencia de valor respecto de las alternativas irrelevantes: Si la función de diferencia de valor v_i representa la preferencia del individuo i sobre $O(B)$, con $B \supseteq A$, entonces para todo subconjunto $B' \subseteq B$ la función v_i restringida a B' representa la preferencia del individuo i sobre $O(B')$.

Por contener la relación \succ_i menos información preferencial que la función v_i , es natural tratar de determinar primero la preferencia \succ_i sobre $O(A)$, mediante una función de diferencia de valor, y después seleccionar la función de evaluación v_i que se declarará al

grupo dentro del conjunto de las funciones de evaluación compatibles con \succeq_i , $V(\succeq_i)$, dado por

$$V(\succeq_i) = \{v_i: A \rightarrow [0, M]: \min_{a \in A} \{v_i(a)\} = 0, \max_{a \in A} \{v_i(a)\} = M, \text{ se cumple (6)}\}, \quad (12)$$

Las funciones de evaluación en este conjunto son equivalentes en el sentido que representan igualmente la preferencia \succeq_i sobre $O(A)$, no existiendo una transformación afín positiva que relacione dos de ellas.

7. ASIGNACIÓN DE UNA FUNCIÓN DE DIFERENCIA DE VALOR

Por el *Teorema 1* del capítulo 6, a cada miembro de un grupo, sujeto a una constitución tipo A , en condiciones de ignorancia sobre las preferencias de los demás miembros, le conviene expresar una función de diferencia de valor v sobre A (aquí se omiten el subíndice de los miembros del grupo) que represente lo mejor posible su preferencia sobre $O(A)$, \succeq . Por tanto, se cumple la relación

$$o \succeq o' \Leftrightarrow \sum_{(a,b) \in o-P} (v(b) - v(a)) \leq \sum_{(a,b) \in o'-P} (v(b) - v(a)) \quad \forall o, o' \in O(A), \quad (13)$$

donde P es el orden débil sobre A correspondiente a su preferencia sobre A , dado por

$$(a,b) \in P \Leftrightarrow v(a) \geq v(b), \quad \forall a, b \in A \quad (14)$$

y $(a,b) \in o-P$ denota las parejas de alternativas que están en o pero no en P . Por tanto los términos en las dos sumatorias en (13) son todos no negativos, cada uno de los cuales corresponde a una pareja en o con sentido opuesto a P .

La relación (13) es equivalente a la relación (6) cuando se elimina el subíndice i , debido a que el lado derecho de la igualdad

$$\sum_{(a,b) \in o \cap P} (v(a) - v(b)) + \sum_{(a,b) \in o-P} (v(b) - v(a)) = \sum_{(a,b) \in P} (v(a) - v(b)).$$

es una constante que no depende de $o \in O(A)$ (es igual a la magnitud del voto $w(P)$ para la primera opción P respecto de \succeq ; ver (8)). La forma (13) es más apropiada que la (6) para el problema individual de determinar una función de diferencia de valor. Aquí el ordenamiento de los elementos de $O(A)$ no se da en términos de la magnitud del voto, como en (6), sino en términos de “la magnitud del voto en contra” o del *costo* $c(o)$, dado por

$$c(o) = \sum_{(a,b) \in o-P} (v(b) - v(a)) \quad \forall o \in O(A), \quad (15)$$

por lo que (13) se puede escribir como

$$o \succeq o' \Leftrightarrow c(o) \leq c(o') \quad \forall o, o' \in O(A), \quad (16)$$

Sea $s = (s_m, s_{m-1}, \dots, s_1) \in O(A)$ una primera opción de \succeq sobre $O(A)$, la cual se toma como referencia y se denomina *semilla*. Por tanto $s = P$ si P no contiene indiferencias sobre A

y en caso contrario $s \in P$. Sea $K_j(o)$ el número de veces que el par de alternativas consecutivas en s , (s_{j+1}, s_j) aparecen directa o indirectamente en orden opuesto en o ,

$$K_j(o) = |\{(s_j, s_{j+1}): (s_j, s_{j+1}) \rightarrow (a, b), (a, b) \in o - P\}| \quad j = 1, 2, \dots, m-1, \quad \forall o \in O(A), \quad (17)$$

siendo $|\cdot|$ la función cardinalidad y \rightarrow la relación de “inclusión” dada por

$$(s_j, s_{j+1}) \rightarrow (s_\alpha, s_\beta) \Leftrightarrow \alpha \leq j, \beta \geq j+1, \quad (18)$$

Sea $C \subseteq O(A)$ el conjunto de todos los ordenamientos $o_j \in O(A)$ que se obtienen de intercambiar en s dos elementos consecutivos, s_{j+1}, s_j ,

$$o = o_j \Leftrightarrow K_k(o_j) = \delta_{jk} \quad j = 1, 2, \dots, m-1 \quad (19)$$

donde δ_{jk} es la delta de Kronecker. Por (15) y (19)

$$c(o_j) = v(s_{j+1}) - v(s_j) \quad j = 1, 2, \dots, m-1 \quad (20)$$

$$c(o) = \sum_{j=1}^{m-1} K_j(o) c(o_j) \quad \forall o \in O(A), \quad (21)$$

y por (16)

$$o \succ o' \Leftrightarrow \sum_{j=1}^{m-1} (K_j(o') - K_j(o)) c(o_j) \geq 0 \quad \forall o, o' \in O(A), \quad (22)$$

El problema de encontrar una función de diferencia de valor que represente una preferencia conocida sobre $O(A)$, la cual es acorde con una función de diferencia de valor sobre A , por (22) se puede resolver como un problema de programación lineal (Krantz, 1971) en donde cada igualdad debida a una indiferencia se usa para eliminar una incógnita $c(o_j)$, y cada desigualdad en (28) corresponde a un elemento $o \in O(A)$ y su inmediato sucesor $o' \in O(A)$, resultando un problema lineal de la forma

max μ

$$\text{tal que } \mu \leq \sum_k A_{ik} x_k, \quad i = 1, \dots, m' \quad (23)$$

donde cada x_k corresponde con una $c(o_j)$. La correspondiente función de diferencia de valor se determina dando un valor arbitrario a $v(s_1)$, por ejemplo θ , y por (23) los demás valores de la función están dados por $v(s_{j+1}) = v(s_j) + c(o_j)$.

Puesto que se supone que \succ es acorde con una función de diferencia de valor, en la práctica es más cómodo determinar simultáneamente dicha preferencia y una función de diferencia de valor v sobre $O(A)$ que la represente.

8. EJEMPLO INTUITIVO

Consideremos el siguiente problema para ilustrar el procedimiento de construcción de una función de valor para los individuos en un grupo. Sean tres individuos que van a decidir sobre un conjunto de tres alternativas a, b, c , los cuales cumplen las siguientes preferencias estrictas:

individuo 1: $a \succ_1 b \succ_1 c$.

individuo 2: $b \succ_2 c \succ_2 a$.

individuo 3: $c \succ_3 a \succ_3 b$.

Utilizando \succ_g para indicar la preferencia estricta para el grupo, la regla de la mayoría simple conduce a lo siguiente:

$a \succ_g b$ (2 de 3 prefieren a sobre b)

$b \succ_g c$ (2 de 3 prefieren b sobre c)

$c \succ_g a$ (2 de 3 prefieren c sobre a)

la preferencia del grupo resulta intransitiva, a, b y c forman un ciclo_j. Nótese que la única información disponible son los ordenamientos de las alternativas de cada individuo, es decir sólo tenemos información de primer orden.

Si aplicamos el concepto de preferencias de segundo orden, veremos que el problema de la intransitividad desaparece, ya que los individuos eligen sobre el conjunto de ordenamientos. Supongamos que a cada miembro del grupo establece un orden débil como su primera opción, que en este caso es el que se señala como o_1 para cada individuo, los órdenes débiles marcados como o_2, o_3, o_4, o_5 , y o_6 son las opciones subsecuentes que resultan de sacrificar sus preferencias “menos fuertes”. Todos los ordenamientos posibles del conjunto de alternativas representan las preferencias de segundo orden. Los órdenes débiles para los tres individuos son:

	Individuo 1	Individuo 2	Individuo 3
o_1	$a \succ_1 b \succ_1 c$	$b \succ_2 c \succ_2 a$	$c \succ_3 a \succ_3 b$
o_2	$a \succ_1 c \succ_1 b$	$c \succ_2 b \succ_2 a$	$c \succ_3 b \succ_3 a$
o_3	$b \succ_1 a \succ_1 c$	$b \succ_2 a \succ_2 c$	$a \succ_3 c \succ_3 b$
o_4	$c \succ_1 a \succ_1 b$	$c \succ_2 a \succ_2 b$	$b \succ_3 c \succ_3 a$
o_5	$b \succ_1 c \succ_1 a$	$a \succ_2 b \succ_2 c$	$a \succ_3 b \succ_3 c$
o_6	$c \succ_1 b \succ_1 a$	$a \succ_2 c \succ_2 b$	$b \succ_3 a \succ_3 c$

Ahora, surge el problema de encontrar una función de diferencia de valor que represente las preferencias de los individuos sobre $O(A)$. El procedimiento que determina una función de valor consistente con el ordenamiento, se muestra a continuación.

Cada individuo establece su orden de preferencia sobre las alternativas, establecemos cierta escala de referencia para construir la función de valor acorde con sus preferencias, digamos de 0 a 10. Para esto necesitamos de cierta valoración a priori de las alternativas; si se le asigna el 0 a la alternativa menos preferida y 10 a la alternativa mas preferida y conocemos su orden de preferencia: $c \succ a \succ b$, entonces $v(b)=0, v(c)=10$, y resta encontrar $v(a)$.

8.1. Bosquejo del método:

1. Generar las permutaciones $O(A)$ a partir del ordenamiento inicial, que representan las preferencias de segundo orden.

2. Calcular las magnitudes del voto en contra para cada permutación con la expresión

$$c(o) = \sum_{(a,b) \in o-P} (v(b) - v(a)) \quad .$$

3. Calcular las diferencias entre las magnitudes del voto consecutivas. Aquí resultan varias expresiones algebraicas que formaran los lados derechos de las restricciones del programa lineal a resolver.

4. Resolver el programa lineal que resulta, para encontrar las valoraciones intermedias del individuo que garantizan que las diferencias entre las magnitudes del voto en contra sean iguales en su mayoría o que se maximice la mínima diferencia entre magnitudes del voto consecutivas.

5. Sustituir el valor encontrado en las magnitudes del voto en contra

6. Paso de Agregación: Una vez calculadas las magnitudes del voto en contra de cada individuo, se procede a agregar la información del grupo con la expresión

$$w_g(o) = \sum_{i \in I} w_i(o) = \sum_{(a,b) \in o} \sum_{i \in I(a,b)} (v_i(a) - v_i(b)), \forall o \in O(A);$$

ya que las w_i están dadas por $w_i(o) = \sum_{(a,b) \in o \cap P_i} (v_i(a) - v_i(b))$

El ordenamiento “ganador” será aquel con la magnitud del voto en contra más pequeña, o bien, con la magnitud del voto a favor más grande.

Regresando a nuestro ejemplo, la siguiente tabla muestra las preferencias de segundo orden y sus correspondientes votos en contra.

	Preferencias	Votos en contra
Primer orden	$c \succ a \succ b$,	0
Segundo orden	$c \succ b \succ a$	x
	$a \succ c \succ b$	$10-x$
	$b \succ c \succ a$	$10+x$
	$a \succ b \succ c$	$20-x$
	$b \succ a \succ c$	20

Tabla 1. Preferencias de segundo orden y votos en contra

La forma en que se calculan los votos en contra se explica a continuación: Sabemos que $v(b)=0$, $v(c)=10$, y escribimos $v(a)=x$. Ahora se compara la preferencia de segundo orden con el ordenamiento inicial que sería nuestro dato de entrada, así como ciertas valoraciones a priori que servirían de referencia para ordenar las magnitudes del voto en forma creciente, por ejemplo, sabemos que el ordenamiento inicial es cab , su magnitud del voto en contra es cero porque el individuo no está haciendo ningún sacrificio con respecto a su ordenamiento inicial, entonces no está "perdiendo" nada, pero sigamos con cba si lo comparamos con respecto al orden inicial está haciendo un intercambio entre a y b , entonces

$$c(o) = \sum_{(a,b) \in o-P} (v(b) - v(a)) = v(a) - v(b) = x - 0$$

De la misma forma, para acb , se hizo un intercambio entre a y c , entonces

$$c(o) = v(c) - v(a) = 10 - x$$

Y así sucesivamente para cada preferencia de segundo orden. Una vez calculadas todos los votos en contra, se procede a calcular las diferencias entre evaluaciones consecutivas:

$c(o_2) - c(o_1)$	$x - 0 = x$
$c(o_3) - c(o_2)$	$(10 - x) - x = -2x + 10$
$c(o_4) - c(o_3)$	$(10 + x) - (10 - x) = 2x$
$c(o_5) - c(o_4)$	$(20 - x) - (10 + x) = -2x + 10$
$c(o_6) - c(o_5)$	$20 - (-x + 20) = x$

Tabla 2. Diferencias entre evaluaciones consecutivas

Entonces tenemos 3 expresiones algebraicas distintas: $\{x, -2x + 10, 2x\}$ que forman los lados derechos del programa lineal:

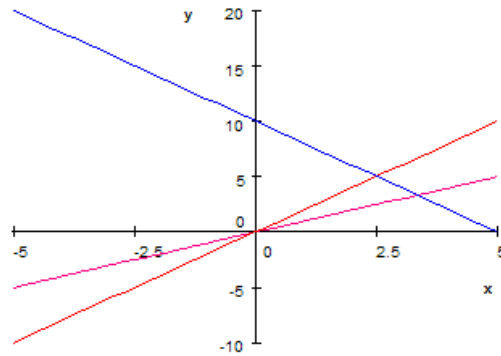
$$\begin{aligned} & \text{Max } \mu \\ \text{s.a } & \mu \leq x \\ & \mu \leq -2x + 10 \\ & \mu \leq 2x \\ & x, \mu \geq 0 \end{aligned}$$

donde $\mu = \min\{x, [-2x + 10], [2x]\}$. A partir de este procedimiento que “minimiza errores” conoceremos, únicamente con información ordinal, las valoraciones “verdaderas” de cada individuo.

Con las transformaciones necesarias el programa queda:

$$\begin{aligned} & \text{Max } \mu \\ \text{s.a } & \mu - x \leq 0 \\ & \mu + 2x \leq 10 \\ & \mu - 2x \leq 0 \\ & x, \mu \geq 0 \end{aligned}$$

al resolverlo con un software especializado tenemos que la solución óptima es: $\mu = 3.33$ $x = 3.33$. Gráficamente podemos ver que tenemos tres líneas rectas cuyas intersecciones son los puntos $x = 0$, $x = (10)/3$ y $x = (10)/4$



Al evaluar las diferencias en estos puntos tenemos que:

Diferencias consecutivas	$x = 0$	$x = \frac{10}{3}$	$x = \frac{10}{4}$
x	0	3.33	2.5
$-2x + 10$	10	3.33	5
$2x$	0	6.66	5
$-2x + 10$	10	3.33	5
x	0	3.33	2.5

Tabla 3. Evaluación de las diferencias consecutivas.

El vértice que hace máxima la mínima diferencia es en $x = 3.33$, entonces podemos decir que $v(a) = 3.33$.

El procedimiento es el mismo para todos los individuos y posteriormente se agrega la información en la preferencia grupal.

8.2. Software asociado.

El programa de cómputo para resolver el problema de decisiones en grupo con preferencias de segundo orden está diseñado en Delphi 7, que es una potente herramienta de desarrollo de programas que permite la creación de aplicaciones para Windows. Delphi dispone del Object Pascal, un lenguaje de programación muy poderoso que está sin duda a la altura del C++ y que incluso lo supera en algunos aspectos. Se eligió esta herramienta ya que las aplicaciones pueden colocarse de forma

muy sencilla en la pantalla según el principio de módulos y porque permite crear el desarrollo del programa en forma visual en la pantalla y a través de la programación convencional. En este momento el software cuenta con 6 opciones en el menú:

- Datos.- Aquí se ingresan el número de opciones, las distintas alternativas y los nombres de los individuos que conforman el grupo. El programa está diseñado para funcionar correctamente con un número indefinido de individuos pero sólo hasta cinco alternativas debido a lo extremadamente combinatorio que resulta un problema con mayor número de alternativas.

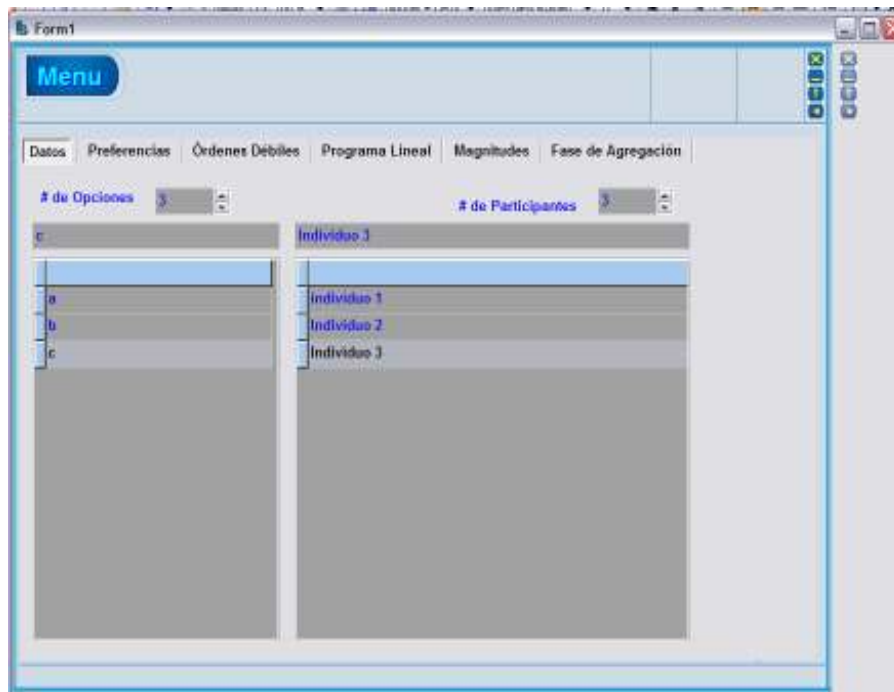


Figura 1. Datos

- Preferencias.- En esta pantalla se selecciona al participante y éste tiene que ordenar las alternativas de acuerdo con sus preferencias, ubicando en el botón **Mas** su alternativa más preferida y en el botón **Menos**, la alternativa menos preferida, y en la barra que inicia en 0 y termina en 10 tiene que ubicar las alternativas intermedias de acuerdo con sus intensidades de preferencia. Esta herramienta nos da una valoración de las alternativas aún cuando el individuo no da expresamente un valor numérico, sino que sólo se le pide información ordinal.

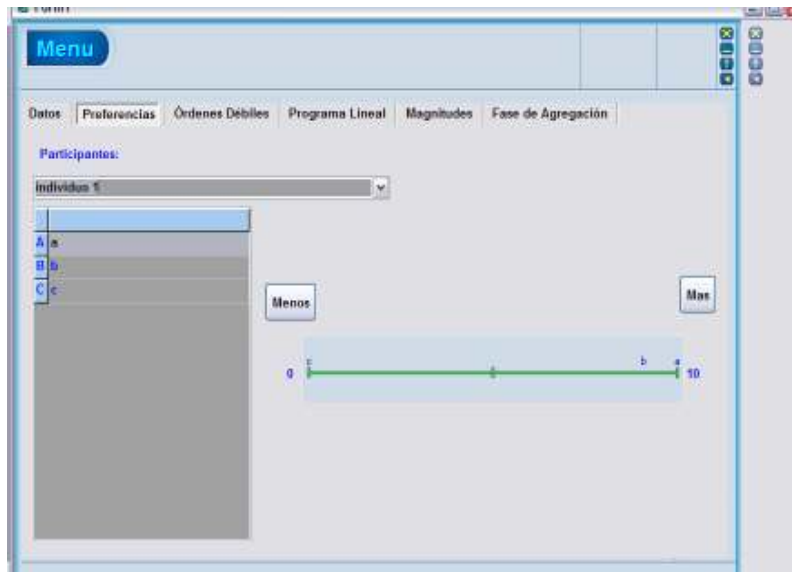


Figura 2. Preferencias

- Ordenes Débiles.- En esta opción aparecen las evaluaciones de los ordenamientos donde se utiliza la información preferencial. La evaluación a priori se refiere a los valores numéricos asociados al orden que dispuso el individuo en la barra de la pantalla anterior. La columna de evaluación presenta las expresiones algebraicas de lo que serán las magnitudes del voto una vez que se haya encontrado la función de valor del individuo.

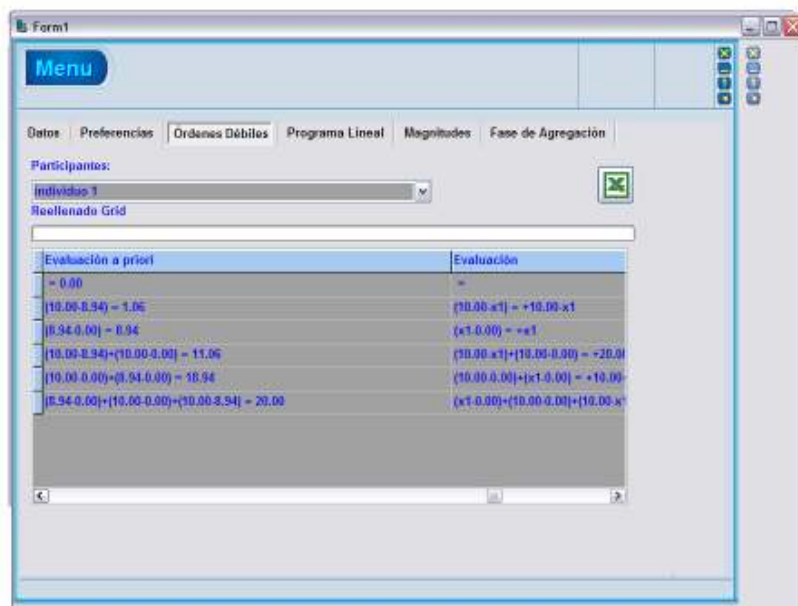


Figura 3. Órdenes débiles

También se muestran las diferencias entre evaluaciones consecutivas que formaran los lados derechos de las restricciones del programa lineal

The screenshot shows a software window titled 'Form1' with a 'Menu' button. Below the menu are tabs for 'Datos', 'Preferencias', 'Ordenas Débiles', 'Programa Lineal', 'Magnitudes', and 'Fase de Agregación'. The 'Programa Lineal' tab is active. Under 'Participantes', there is a dropdown menu showing 'Individuo 1'. Below that is a 'Reellenado Grid' field. The main area contains a table with the following data:

	Dif. entre Opciones	Evaluación a priori	Evaluación
a b c =		= 0.00	=
b a c +10.00·x1		(10.00-8.94) = 1.06	(10.00)·x1
a c b +2x1-10.00		(8.94-0.00) = 8.94	(x1-0.00)
b c a +20.00-2x1		(10.00-8.94)+(10.00-0.00) = 11.06	(10.00)·x1
c a b -10.00+2x1		(10.00-0.00)+(8.94-0.00) = 18.94	(10.00)·x1
c b a +10.00-1x1		(8.94-0.00)+(10.00-0.00)+(10.00-8.94) = 20.00	(x1-0.00)

Figura 4. Diferencias entre evaluaciones

- Programa lineal.- En la pantalla anterior se desplegaron las diferencias entre las evaluaciones consecutivas, en esta pantalla se identifican las expresiones algebraicas diferentes que resultan de las diferencias entre evaluaciones consecutivas y se construyen las restricciones del programa lineal y agrega las variables de holgura para poner el problema en forma estándar y generar la tabla simplex. Es en esta opción donde se resuelve el programa lineal que dará como resultado el valor de la alternativa intermedia. Aquí el programa identifica si la solución inicial básica es factible o no factible, dependiendo del caso se utiliza el simplex primal o el simplex dual para resolver el problema.

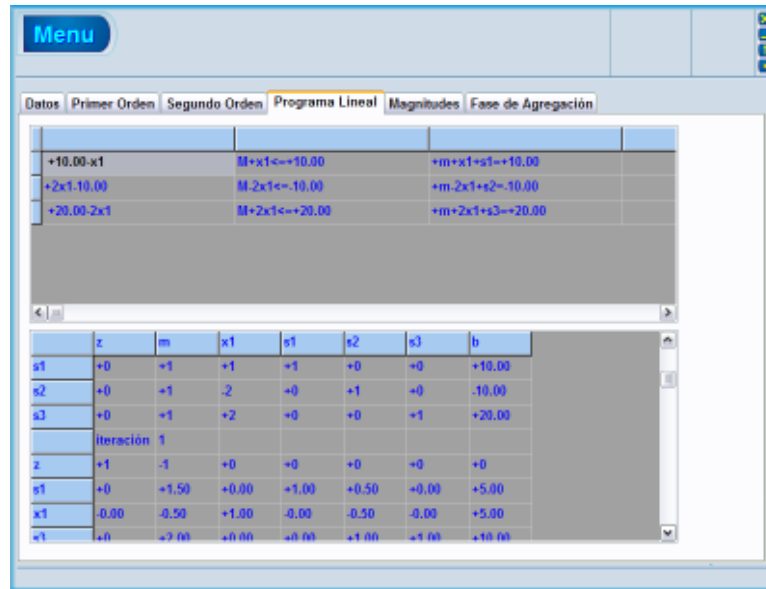


Figura 5. Programa lineal

- Magnitudes.- Una vez que se obtiene la solución del problema lineal, específicamente el valor de x_i , se calculan las magnitudes del voto, utilizando las expresiones que se generaron en la etapa de evaluación. También se sustituye el valor de x_i en las diferencias de las evaluaciones consecutivas, ésta información podrá utilizarse para otro tipo de análisis.

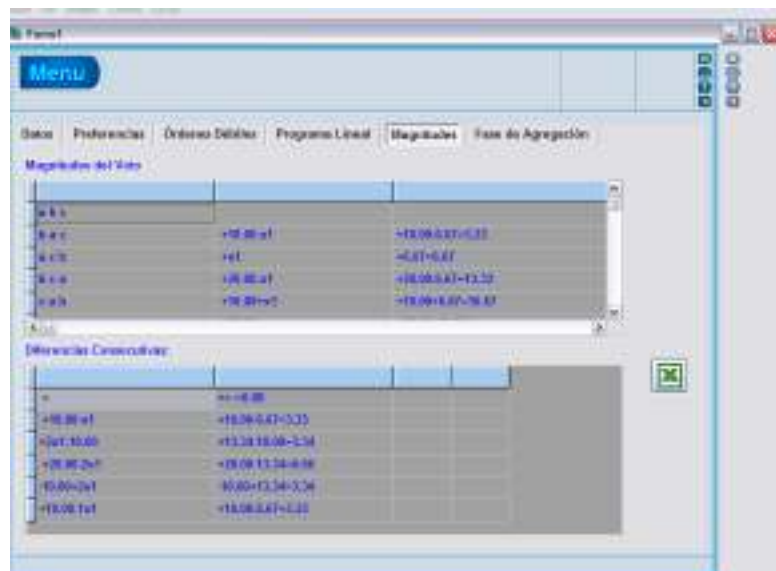


Figura 6. Magnitudes

- Fase de Agregación.- En esta pantalla se concentra la información de todos los individuos que participan en la decisión y, para cada ordenamiento se calcula la magnitud del voto grupal, el ordenamiento “ganador” será aquel con la menor

magnitud del voto en contra. También se muestra una tabla donde se muestran las funciones de valor de cada individuo para cada alternativa y el orden social resultante.

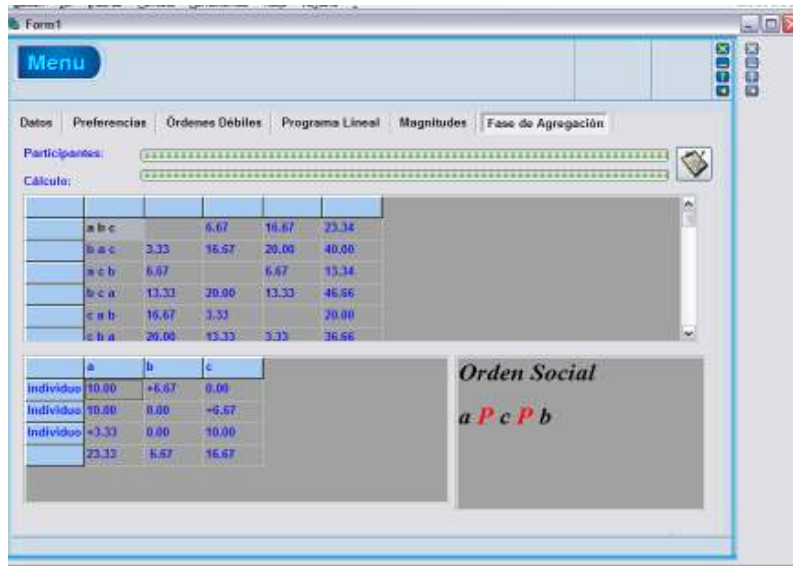


Figura 7. Fase de agregación

8.3 Ejemplo de aplicación.

El Departamento de Métodos Cuantitativos¹ del Centro Universitario de Ciencias Económico Administrativas de la Universidad de Guadalajara, tiene un presupuesto anual aproximado de 370,000, de los cuales se utilizan aproximadamente 75,000 para apoyo a los profesores ponentes en eventos, el presupuesto no es fijo y cambia cada año dependiendo de las necesidades del Departamento. Adscritos al departamento existen 27 Profesores de tiempo completo, de los cuales aproximadamente 8 profesores son los que compiten por una parte del presupuesto para apoyo a ponentes. Los encargados de aprobar o desaprobar una solicitud de apoyo son los miembros del Colegio Departamental, integrado por el Jefe del Departamento, los Presidentes de Academia (4 academias: Matemáticas, Optimización, Estadística y Economía Matemática y Econometría) y el director del Centro de Investigación en Teoría Económica (CITEC). El problema que se presenta es que el dinero que se utiliza se agota antes del mes de junio y sólo se ven beneficiados un par de profesores; así los profesores que tienen aceptada alguna ponencia en algún evento programado a finales de año, no cuentan con el apoyo, simplemente porque se terminó. El Colegio departamental sigue ciertos lineamientos para otorgar los apoyos, pero son tan generales que basta con llegar primero para ser beneficiado. Para ayudar al Colegio Departamental y sobre todo al Jefe de Departamento en la asignación de estos recursos, le propusimos el uso de la metodología de Decisiones en Grupo con Preferencias de Segundo Orden, para obtener un ordenamiento de los profesores solicitantes y de esta manera asignar el presupuesto entre los profesores que cumplan con los requisitos que se solicitan (el Colegio departamental estableció requisitos adicionales, para filtrar las solicitudes) y no de acuerdo no únicamente con la fecha de entrada de la solicitud; para, de esta modo, garantizar la asistencia a eventos de importancia académica de la mayoría de los profesores (el presupuesto es finito y escaso) durante todo el año y no sólo para un par de profesores al inicio del año. Así, una vez que los integrantes del Colegio Departamental aplican sus filtros, ordenan a los solicitantes de acuerdo con sus criterios, para este problema después del filtro pasaron cinco solicitudes. A continuación mostramos las pantallas con los resultados que arroja PRESEO.

Como se mencionó son 6 individuos los que toman la decisión: El jefe del Departamento, El Director del CITEC, el Pdte. de la Academia de Matemáticas, el Pdte.

¹ El ejemplo es ficticio y se utilizó para mostrar al Colegio departamental el uso de la metodología.

Academia de Estadística, el Pdte. Academia de Optimización y el Pdte. Academia de Economía Matemática quienes toman la decisión sobre un conjunto de 5 alternativas: {P1, P2, P3, P4, P5}.

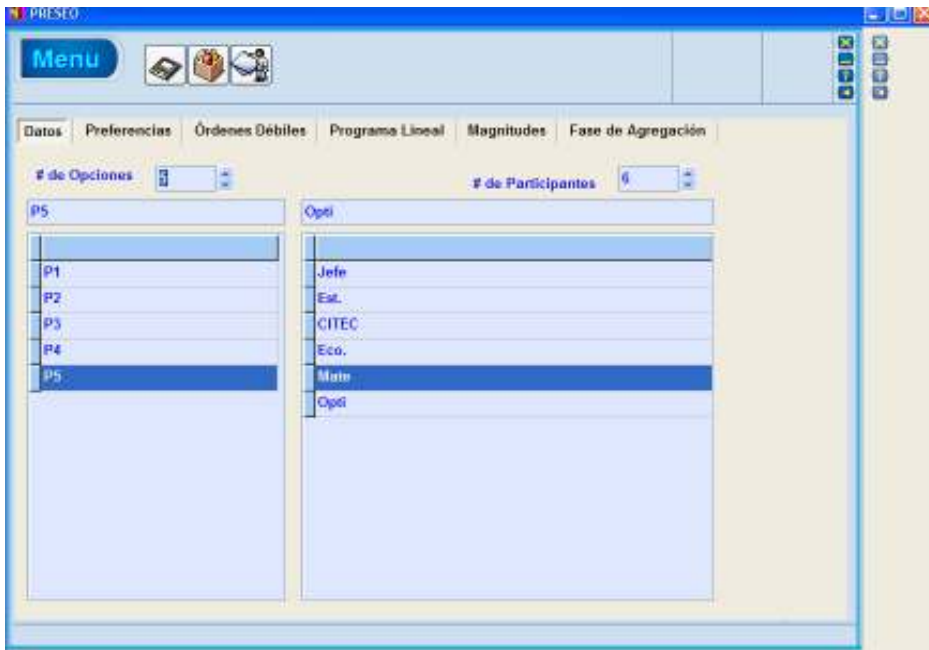


Figura 8. Ejemplo con 6 participantes y 5 alternativas

A continuación se muestran las preferencias de cada uno de los individuos y sus correspondientes fuerzas de preferencia:

- Jefe: $P2 \succ P4 \succ P3 \succ P5 \succ P1$

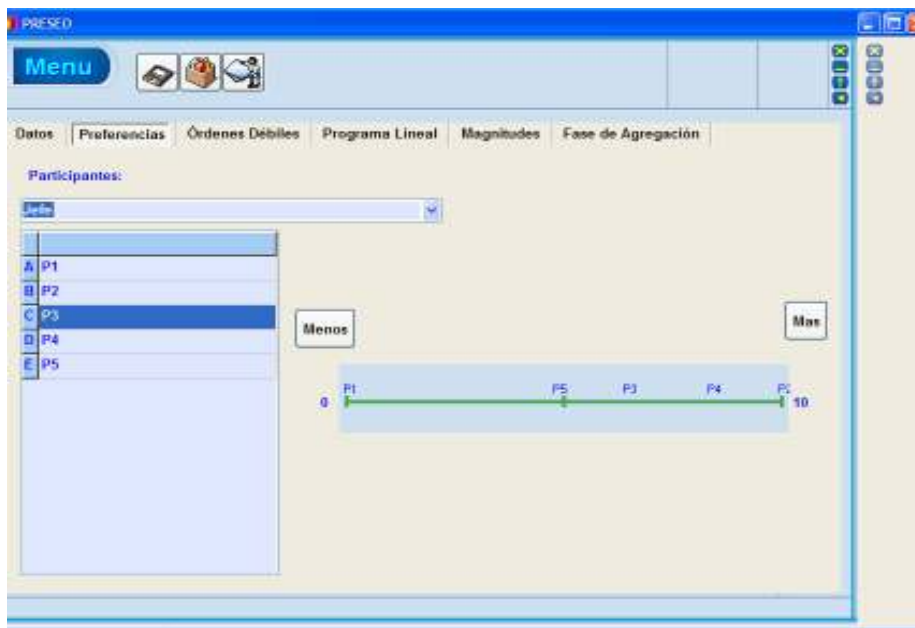


Figura 9. Preferencias del Jefe de Departamento

- Estadística: $P1 \succ P3 \succ P4 \succ P5 \succ P2$

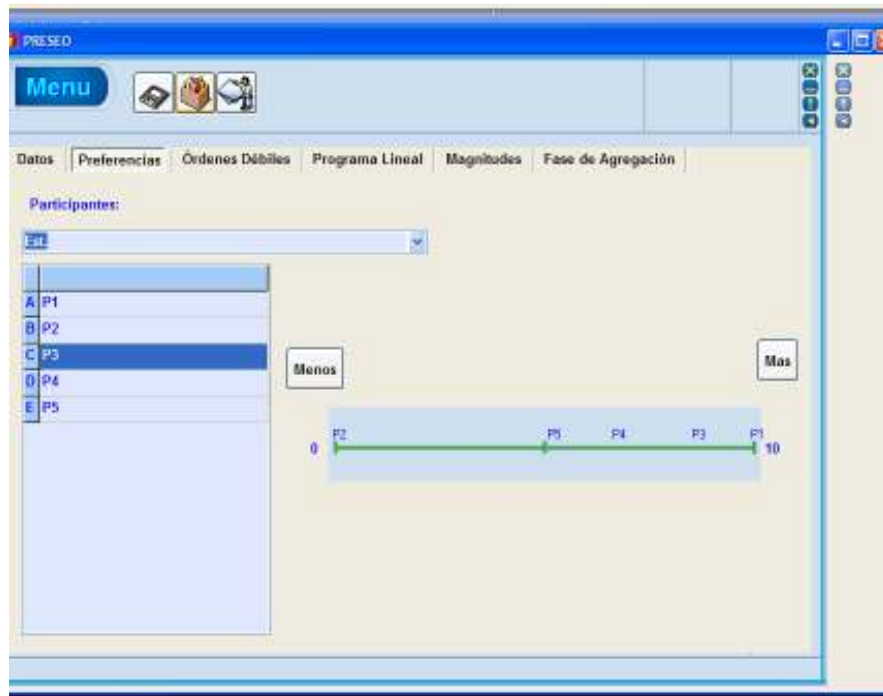


Figura 10. Preferencias del Pdte. de la Academia de Estadística

- CITEC: $P4 \succ P2 \succ P1 \succ P3 \succ P5$

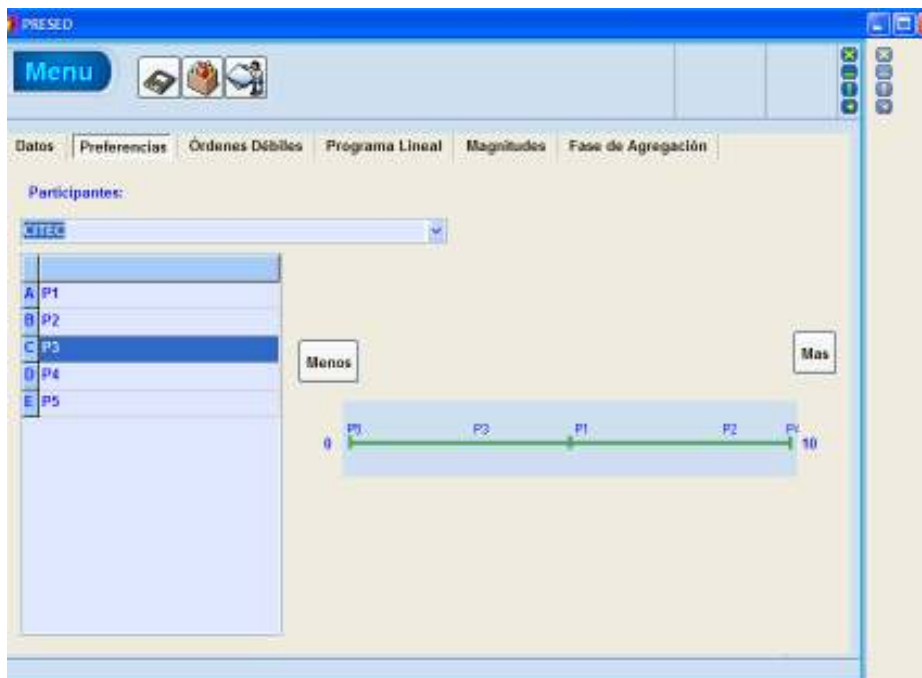


Figura 11. Preferencias del Coord. del CITEC

- Economía: $P2 \succ P5 \succ P4 \succ P3 \succ P1$

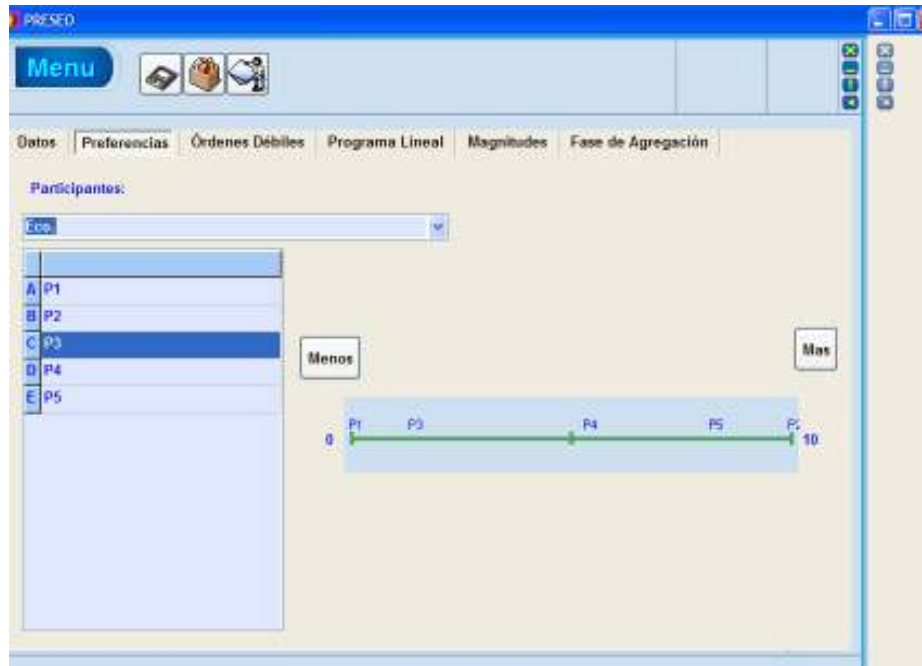


Figura 12. Preferencias del Pdte. de la Academia de Economía Matemática y Econometría

- Matemáticas: $P5 \succ P1 \succ P2 \succ P4 \succ P3$

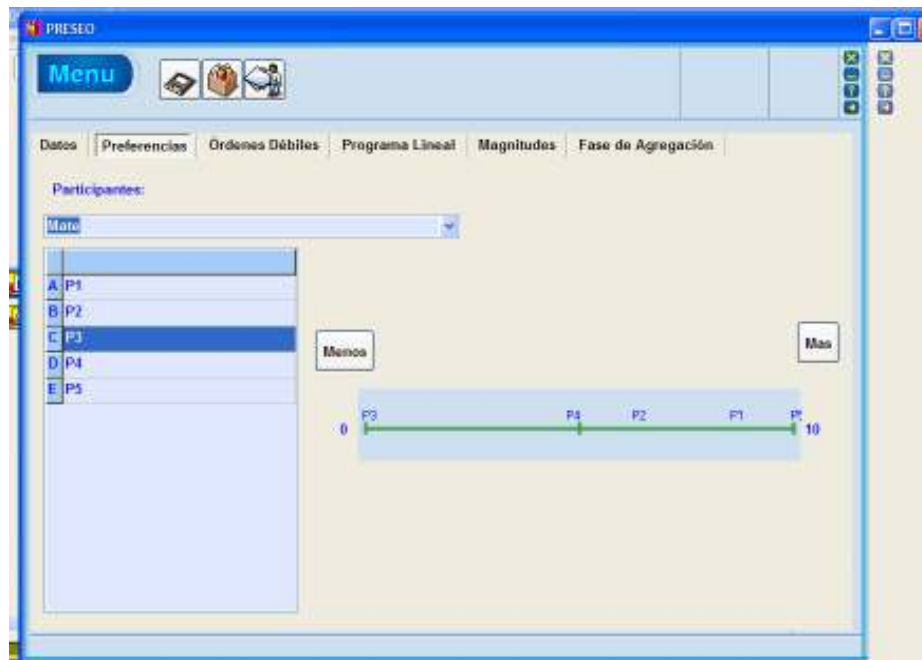


Figura 13. Preferencias del Pdte. de la Academia de Matemáticas Generales

- Optimización: P2 > P1 > P5 > P4 > P3

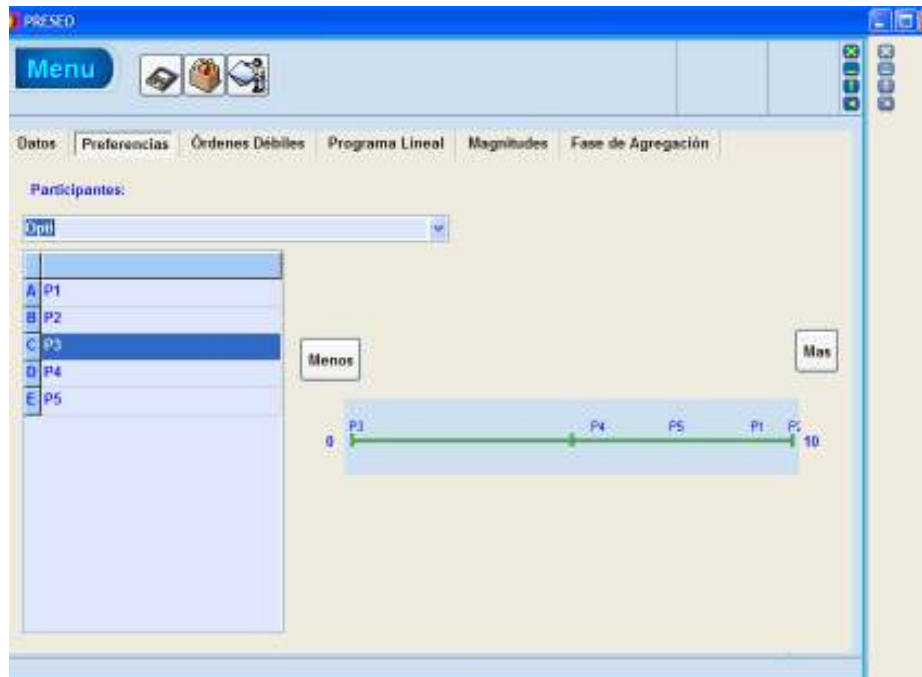


Figura 14. Preferencias del Pdte. de la Academia de Optimización

Después de calcular las diferencias entre las evaluaciones de los ordenamientos y de resolver el programa lineal de cada individuo, se obtienen las funciones de valor de éstos, así como las magnitudes del voto para cada alternativa, esta información se muestra en la siguiente pantalla:



Figura 15. Magnitudes del voto, funciones de valor y Orden grupal

De esta manera el orden que se obtiene es $P2 \succ P4 \succ P5 \succ P1 \succ P3$, por lo que para la asignación de apoyos será tomado en cuenta en primer lugar la solicitud del profesor 2, y posteriormente las demás solicitudes de acuerdo con el orden obtenido.

Nota acerca de la alternativa irrelevante: Ver APENDICE I.

9. CONCLUSIONES

Este trabajo resuelve el problema de la comparación interpersonal de la preferencia de manera indirecta: Se basa en que todos los miembros deben tener la misma influencia en la elección del grupo, y en que las magnitudes M_i reflejan de alguna manera estas influencias, como se muestra en las propiedades de la constitución tipo A. De acuerdo con esto, suponemos que una argumentación similar al conocido principio de la razón insuficiente de Laplace, elaborado en el contexto de la asignación de probabilidades *a priori*, podría convencer en muchas situaciones a los miembros del grupo de que estos valores debieran ser todos iguales, a menos que se encontraran razones suficientes para hacer diferencias entre las M_i (ver ec.12), sin discriminar miembros del grupo ni alternativas. Tales razones sirven para ajustar las M_i mediante un factor de corrección de escala invariable frente a cambios arbitrarios en la designación de las alternativas y la designación de los miembros del grupo.

De acuerdo con Sen (1974, 1979) el uso de ordenamientos de orden superior en el problema de elección social puede aplicarse al problema de encontrar una medida significativa de la utilidad cardinal. Se ha mostrado que una Constitución que toma en cuenta la información preferencial adicional a los ordenamientos de preferencia individual, que no resulta de la comparación de la fuerza de preferencia entre los individuos del grupo, sino que toma en cuenta en forma equitativa las preferencias de los elementos del grupo, y establece cuál es la aportación de cada individuo al orden grupal, tratando de que todos tengan la misma influencia sobre la decisión, y donde cada miembro del grupo establece su función de diferencia de valor para las alternativas, éstas se pueden agregar en una función de valor grupal de manera consistente con los axiomas de Arrow.

En resumen se concluye que:

1. Se ha diseñado un nuevo procedimiento para la toma de decisiones en grupo
2. Se ha resuelto el problema de decisiones en grupo utilizando información ordinal e información acerca de la fuerza de la preferencia de los individuos usando el concepto de preferencias de segundo orden. Como se mostró en el Teorema 1, la constitución cumple con las condiciones *de Optimalidad de Pareto* y las condiciones *de Arrow*.

3. Se ha desarrollado un algoritmo capaz de asignar una función de valor sobre un conjunto finito para cada individuo en el problema de decisiones en grupo con preferencias de segundo orden, este mecanismo nos permite medir la fuerza de preferencia de los miembros del grupo.
4. Se ha diseñado un programa de cómputo en Delphi 7, al que se ha llamado PRESEO, en el que se implementa la metodología de preferencias de segundo orden y el algoritmo desarrollado para realizar las mediciones de la fuerza de las preferencias de los individuos, para asignar la función de valor. Ya que el problema es extremadamente combinatorio, el uso del software facilita el proceso.

9.1. Contribuciones.

El problema de encontrar una preferencia \succeq sobre $O(A)$ acorde con una función de diferencia de valor es formalmente equivalente al problema de asignar una medida de probabilidad sobre conjuntos finitos (Fishburn *et al*, 1989; Roberts, §8.5.4; Scott, 1964) y ambos problemas son equivalentes al problema de asignación de constantes escaladas k_i , de aquí la relevancia de la investigación.

Las principales contribuciones de la investigación son:

- Propuesta de una nueva forma de hacer comparaciones interpersonales de la intensidad de preferencia
- Un procedimiento para medir la fuerza de preferencia para conjuntos finitos.
- Diseño del software que facilita las mediciones.

9.2. Investigación futura

Por el momento el software desarrollado sólo resuelve problemas de hasta 5 alternativas, se pretende ampliar el número de alternativas consideradas. En este trabajo sólo se han considerado problemas en donde no se permite la indiferencia, debido sobre todo a la implementación computacional del procedimiento que las incluye. También es necesario diseñar una versión en red de PRESEO para permitir resolver problemas de decisión en grupo vía web.

Por otro lado el procedimiento desarrollado pudiera usarse en el contexto de la Teoría de la Decisión multicriterio, en donde cada individuo representaría a un criterio o punto de vista, y no necesariamente se tendrían que utilizar funciones de diferencia de valor.

APÉNDICE 1.

Para ilustrar que se satisface uno de los axiomas más polémicos en la formulación de Arrow supongamos que la alternativa P5 no puede participar del apoyo debido a que pidió una licencia por estar realizando sus estudios de posgrado, entonces ahora resolvemos el problema con sólo 4 alternativas y verificaremos que el orden grupal no cambia.

Ahora el conjunto de alternativas esta formado por P1, P2, P3 y P4

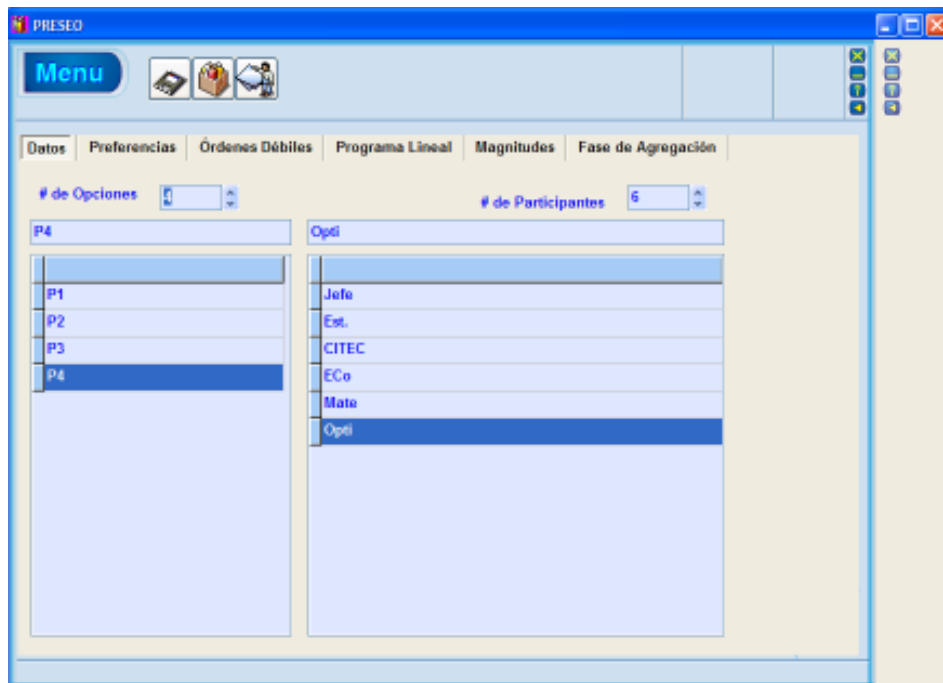


Figura 1. A.I. Datos

Los individuos mantienen el orden de sus preferencias:

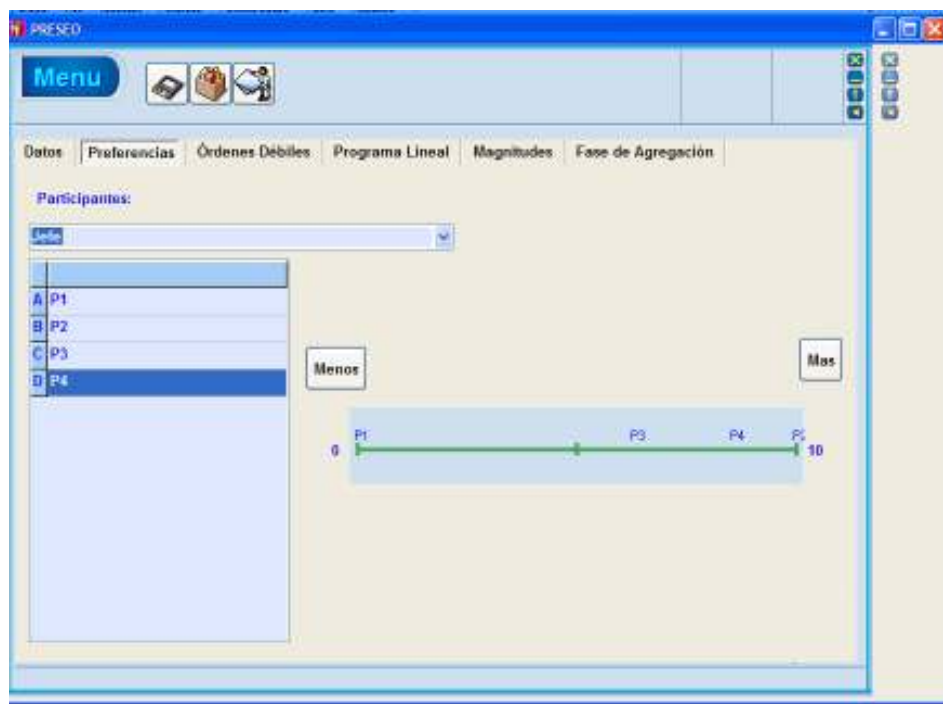


Figura 2. A. I. Preferencias jefe

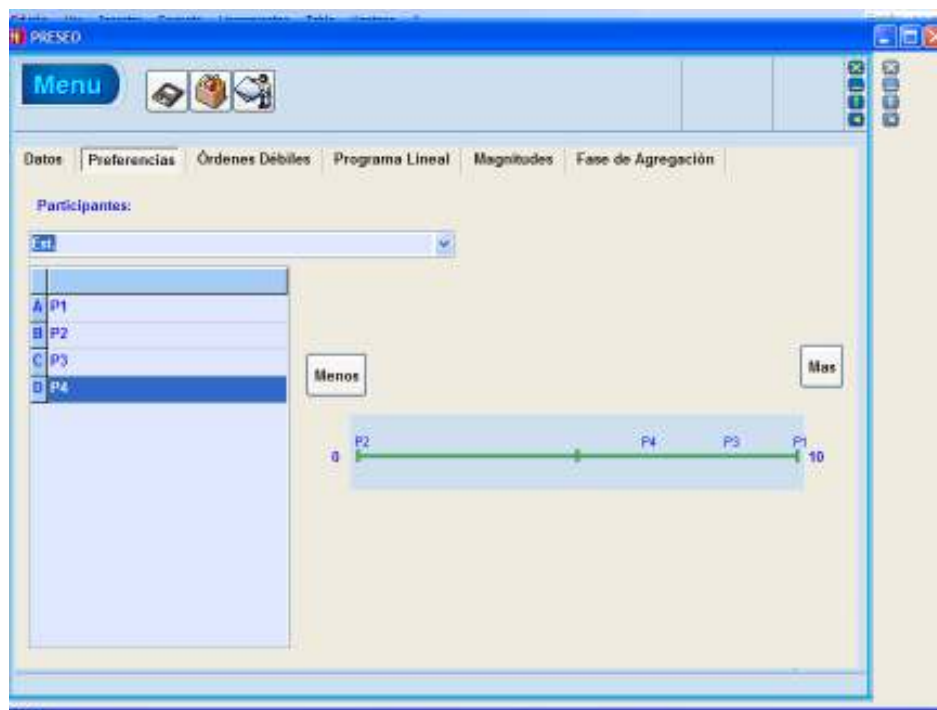


Figura 3. A.I. Preferencias Estadística

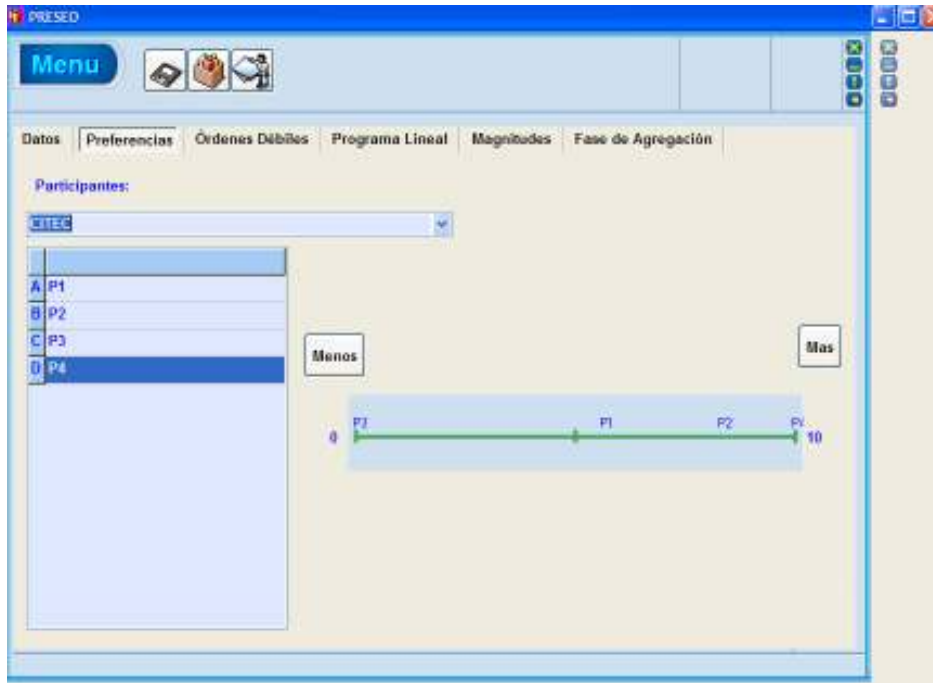


Figura 4. A.I. Preferencias CITEC

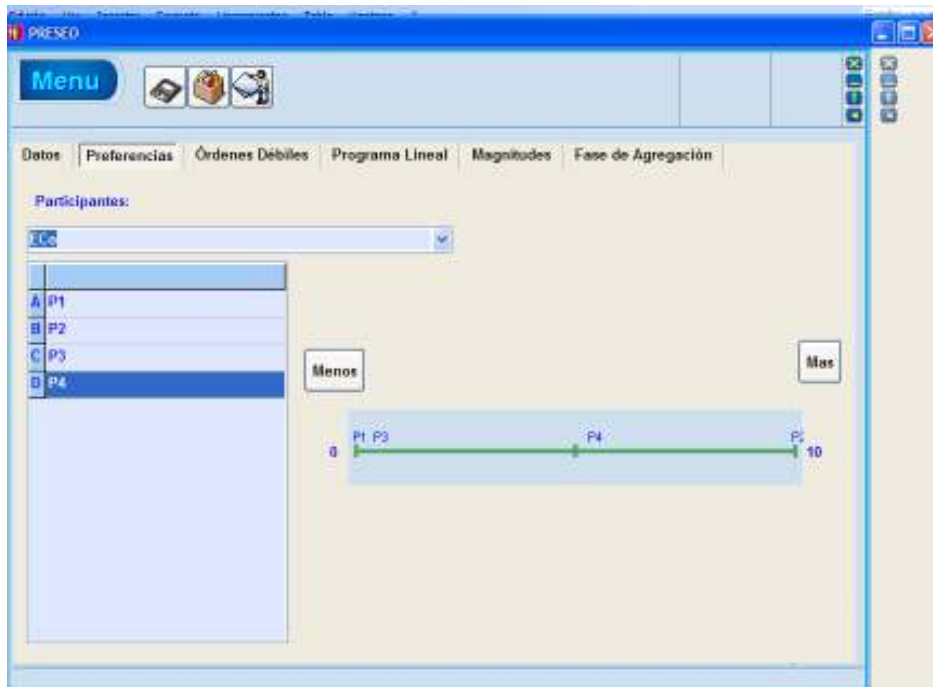


Figura 5. A.I. Preferencias Economía Matemática

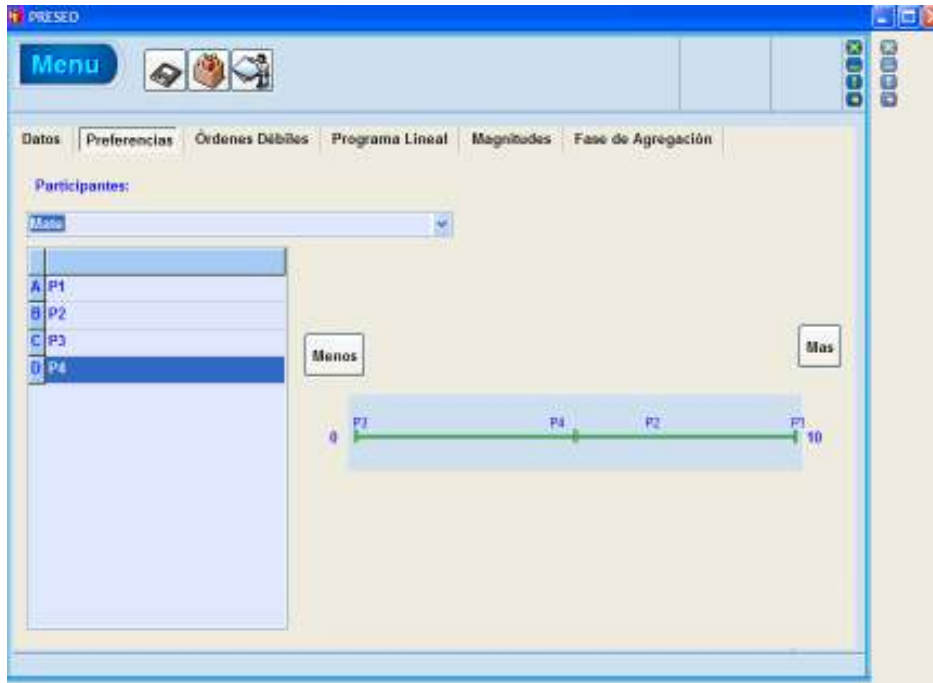


Figura 6. A.I. Preferencias Matemáticas

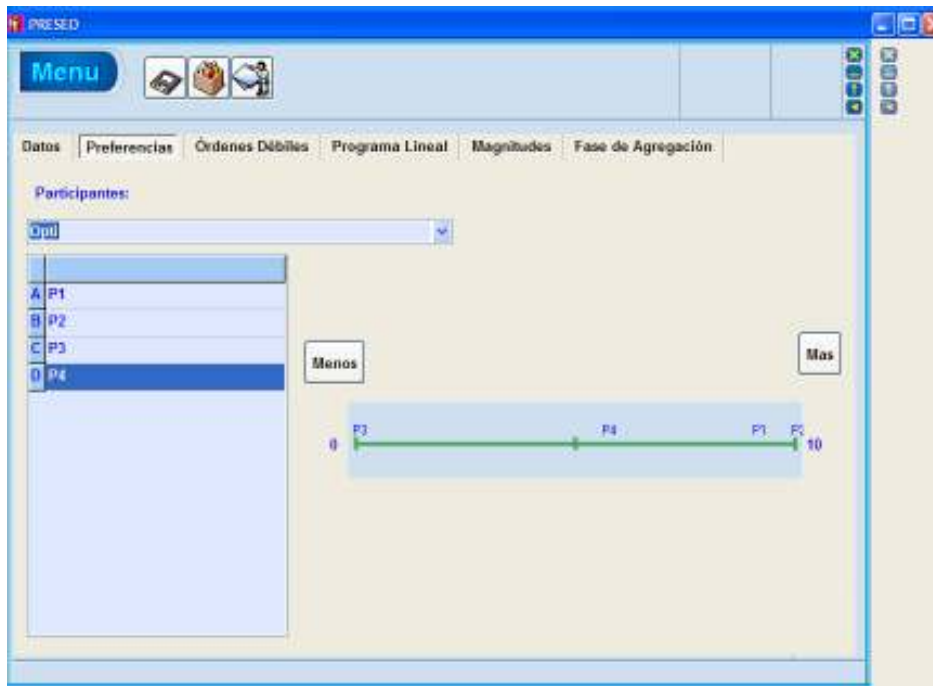


Figura 7. A. I. Preferencias Optimización

Y el orden grupal se muestra a continuación:

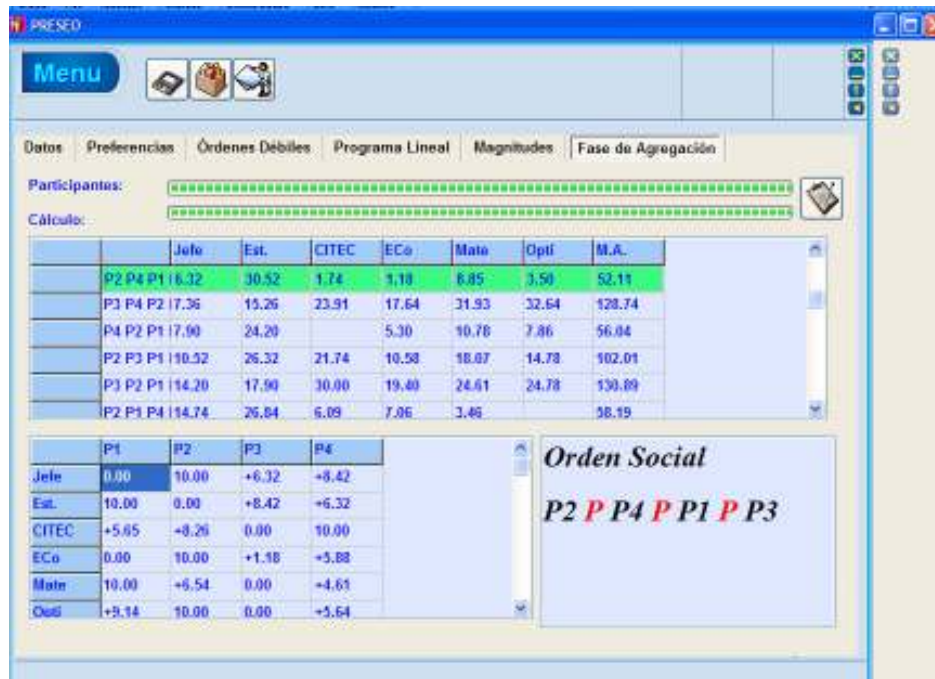


Figura 8. A.I. Orden grupal

Se hizo una comparación con respecto a las funciones de valor obtenidas para cada alternativa en ambos problemas y podría argumentarse que éstas fueron afectadas por el “pulso” del individuo al ubicar las alternativas en la barra de preferencias.

	P1	P2	P3	P4	P5	P1	P2	P3	P4
Jefe	0	10	6.34	8.24	4.78	0	10	6.32	8.42
Est.	10	0	8.56	6.53	5.16	10	0	8.42	6.32
CITEC	5.14	8.44	2.95	10	0	5.65	8.26	0	10
Eco.	0	10	1.45	5.33	8.10	0	10	1.18	5.88
Mate	8.47	6.26	0	4.78	10	10	6.54	0	4.61
Opti	9.09	10	0	5.45	7.27	9.14	10	0	5.64

Tabla 1. A.I. Comparación de las funciones de valor

Aún así, el grupo cumple el axioma de la alternativa irrelevante cuando todos los miembros del grupo la cumplen porque la función de valor no depende de normalizaciones; de hecho éstas pueden sufrir transformaciones afines.

Para evitar estas imprecisiones usaremos las funciones de valor del ejemplo con 5 alternativas en el ejemplo de 4 alternativas y verificamos que las preferencias entre ordenamientos son las mismas. Los resultados para el individuo 1 (Jefe) se muestran a continuación. Sólo se anexan los cálculos de un solo individuo ya que para el problema de 5 alternativas tenemos $5! = 120$ ordenamientos para cada individuo.

Las funciones de valor obtenidas para los 6 individuos en el problema de 5 alternativas son las siguientes

alternativa	P1	P2	P3	P4	P5
individuo					
Jefe	0	10	6.34	8.24	4.78
Est	10	0	8.56	6.53	5.16
CITEC	5.14	8.44	2.95	10	0
Eco	0	10	1.45	5.33	8.10
Mate	8.47	6.26	0	4.78	10
Opti	9.09	10	0	5.45	7.27

Tabla 2. Funciones de valor para el problema de 5 alternativas

Ahora considerando estas funciones de valor, se calculan las preferencias de segundo orden para el problema con cuatro alternativas, por ejemplo para el individuo 1 (Jefe): en la tabla se muestran las preferencias entre ordenamientos de acuerdo con la evaluación obtenida, el ordenamiento con la menor evaluación es el más preferido, ya que estamos considerando las magnitudes del voto en contra.

<u>Jefe</u>	<u>evaluación $w(.)$</u>
2431	0
4231	1.76
2341	1.9
4321	5.42
3241	5.56
2413	6.34
3421	7.32

<u>Jefe</u>	<u>evaluación $w(.)$</u>
4213	8.1
2314	10.14
3214	13.7
2143	14.58
4312	15.42
2134	16.48
3412	17.32
4123	18.1
4132	21.76
3124	23.8
1243	24.58
3142	25.56
1423	26.34
1234	26.48
1432	30
1324	30.14
1342	31.9

Tabla 3. Preferencias de segundo orden para el individuo 1

Las preferencias entre ordenamientos coinciden con las del problema con 4 alternativas calculando sus propias funciones de valor.

<u>Jefe</u>	<u>evaluación $w(.)$</u>
2431	0
4231	1.58
2341	2.10
4321	5.26
3241	5.78
2413	6.32
3421	7.36

<u>Jefe</u>	<u>evaluación $w(.)$</u>
4213	7.90
2314	10.52
3214	14.20
2143	14.74
4312	15.26
2134	16.84
3412	17.36
4123	17.90
4132	21.58
3124	24.20
1243	24.74
3142	25.78
1423	26.32
1234	26.84
1432	30
1324	30.52
1342	32.10

Tabla 4. Preferencias entre ordenamientos para el problema con 4 alternativas

De esta manera, no importa las normalizaciones de la escala que usemos, ni el valor numérico asignado, siempre se conserva el mismo orden de preferencia entre ordenamientos. Lo mismo ocurre con el resto de los individuos del grupo.

Las tablas siguientes muestran las preferencias entre ordenamientos calculadas por PRESEO y son exactamente las mismas que muestran las tablas de arriba.

- Problema con 5 alternativas, para el individuo 1 (Jefe):

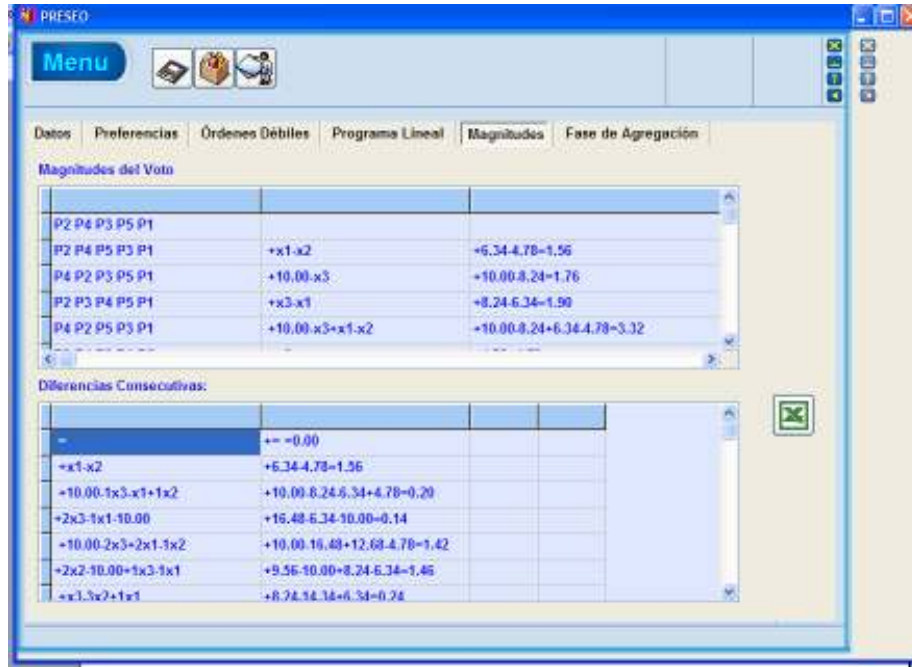


Figura 9. Magnitudes del voto para el individuo 1

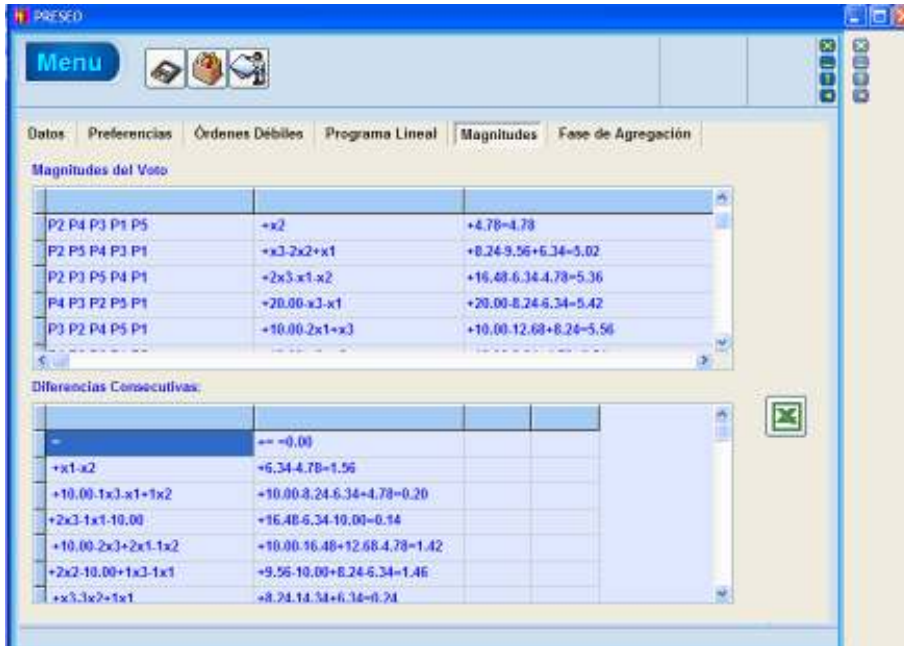


Figura 24a. Magnitudes del voto para el individuo 1

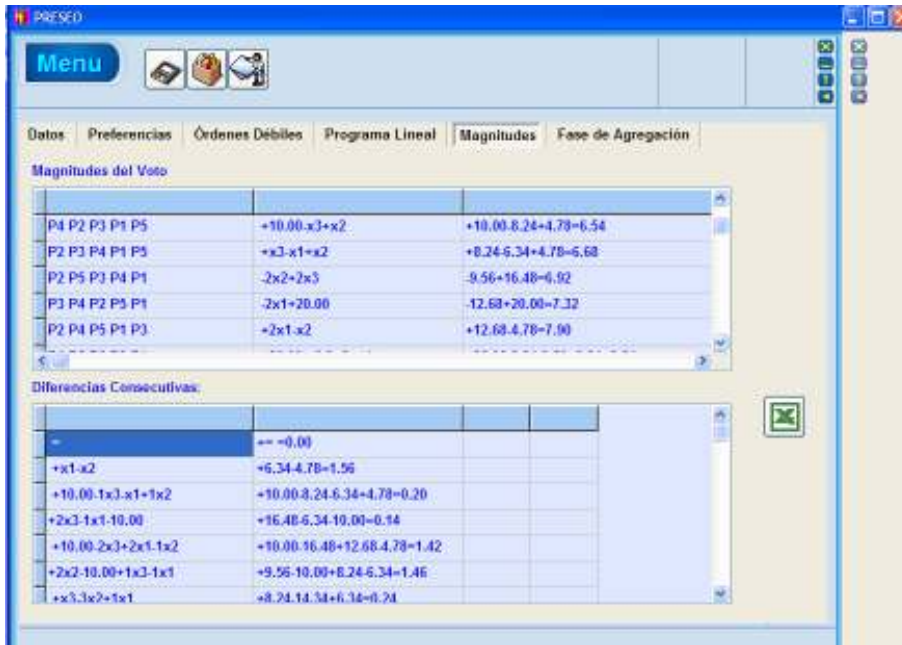


Figura 24b. Magnitudes del voto para el individuo 1.

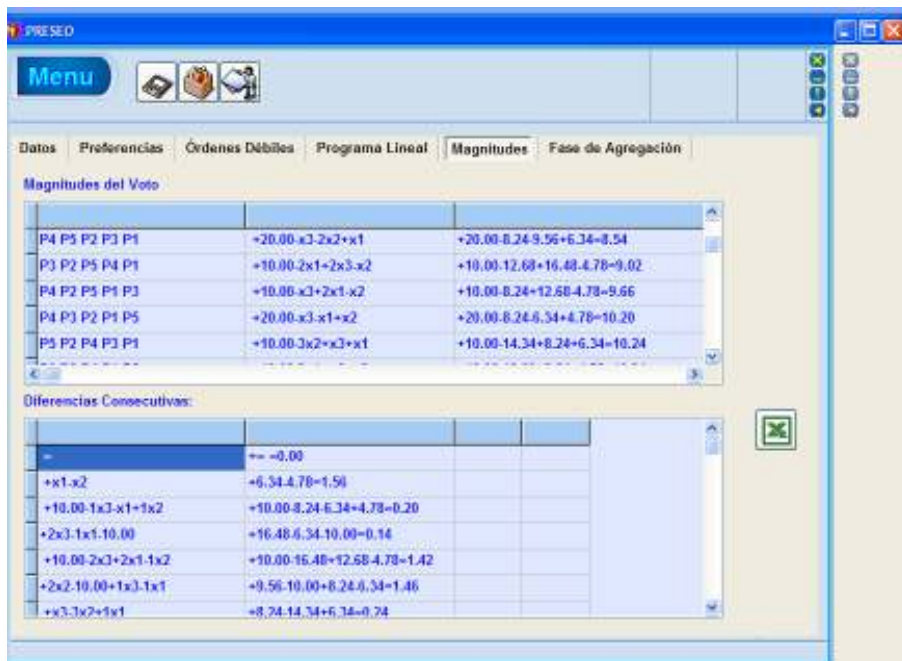


Figura 24c. Magnitudes del voto para el individuo 1

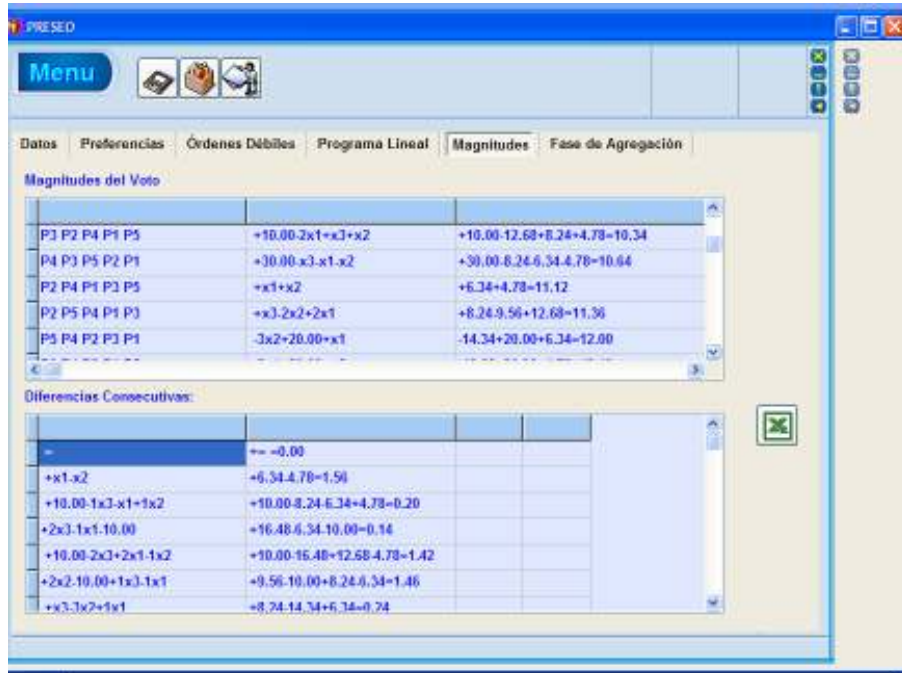


Figura 24d. Magnitudes del voto para el individuo 1.

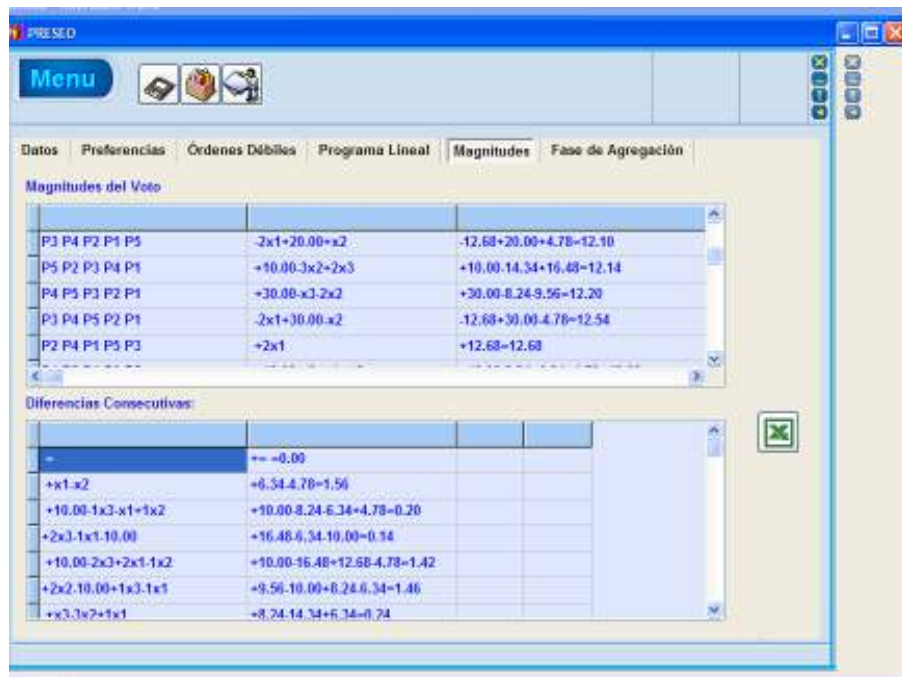


Figura 24e. Magnitudes del voto para el individuo 1.

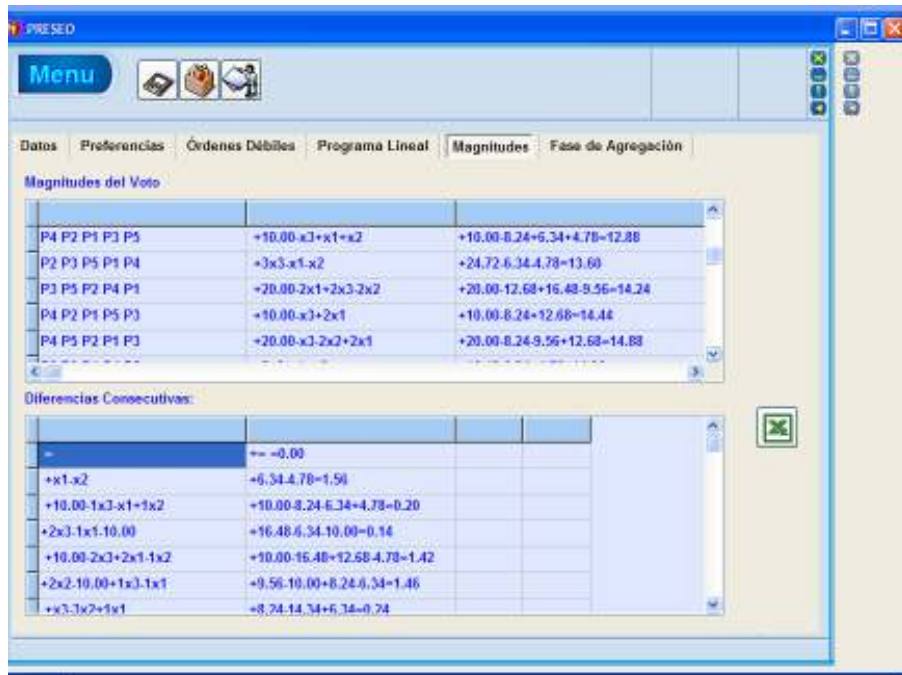


Figura 24f. Magnitudes del voto para el individuo 1

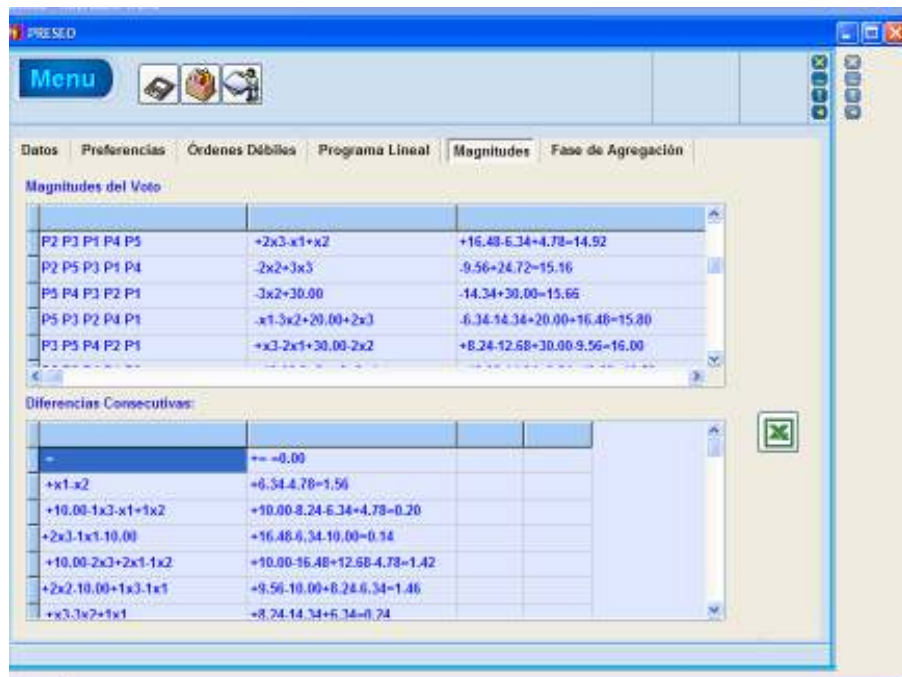


Figura 24g. Magnitudes del voto para el individuo 1.

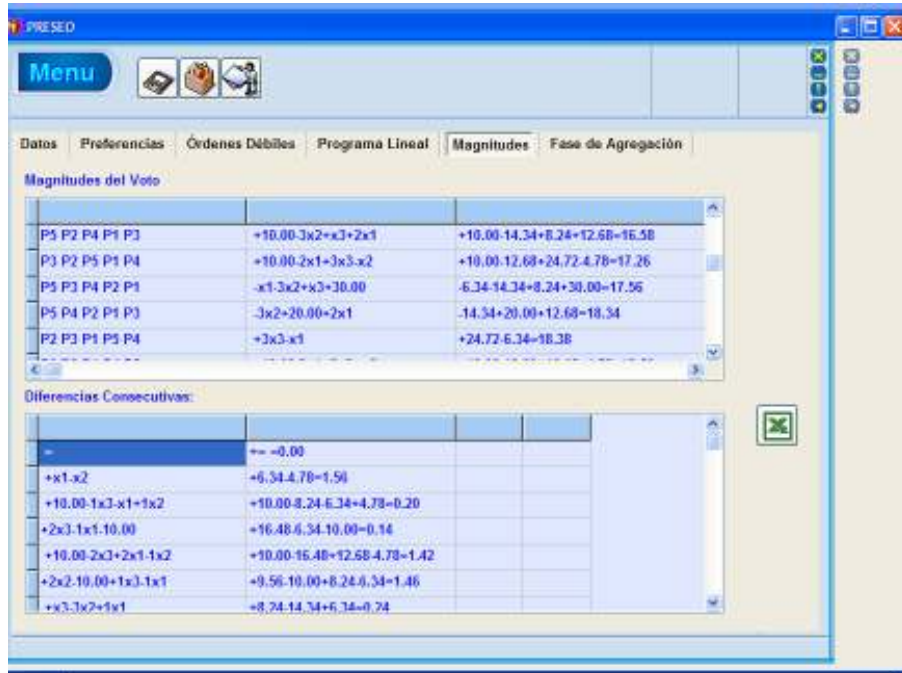


Figura 24h. Magnitudes del voto para el individuo 1.

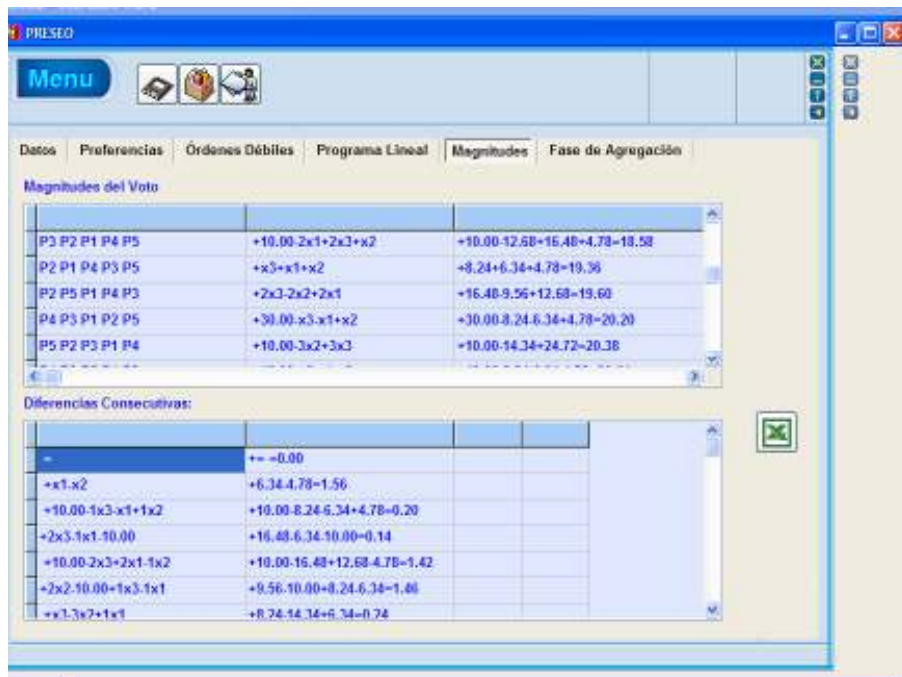


Figura 24i. Magnitudes del voto para el individuo 1.

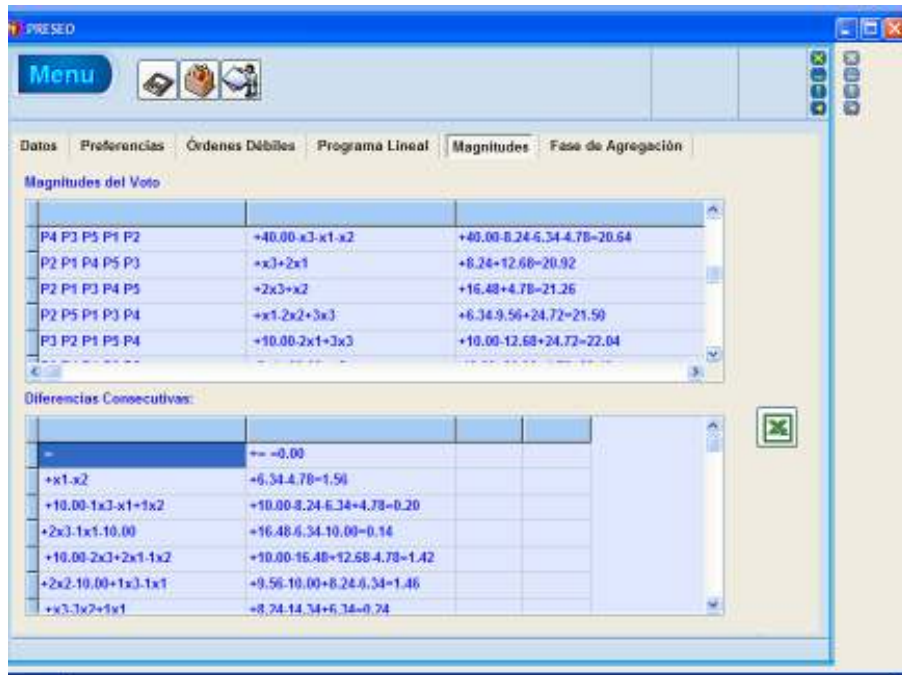


Figura 24j. Magnitudes del voto para el individuo1.

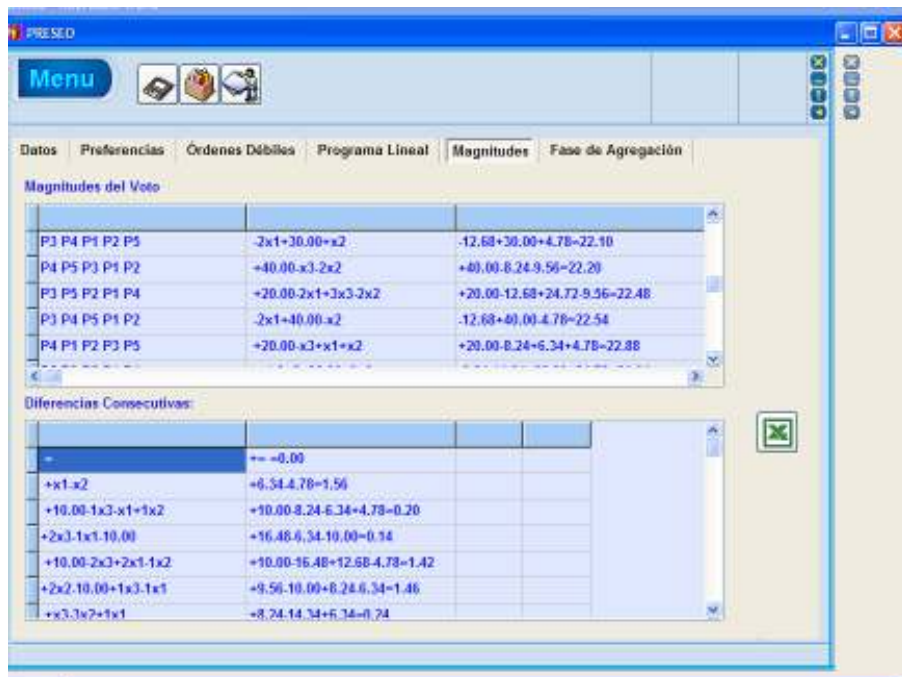


Figura 24k. Magnitudes del voto para el individuo 1.

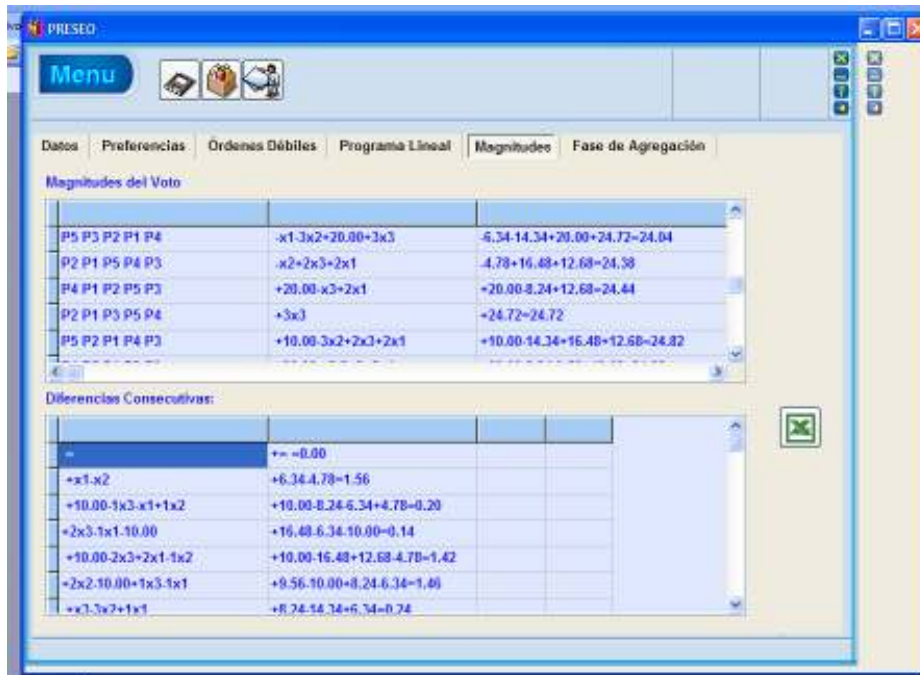


Figura 24l. Magnitudes del voto para el individuo 1.

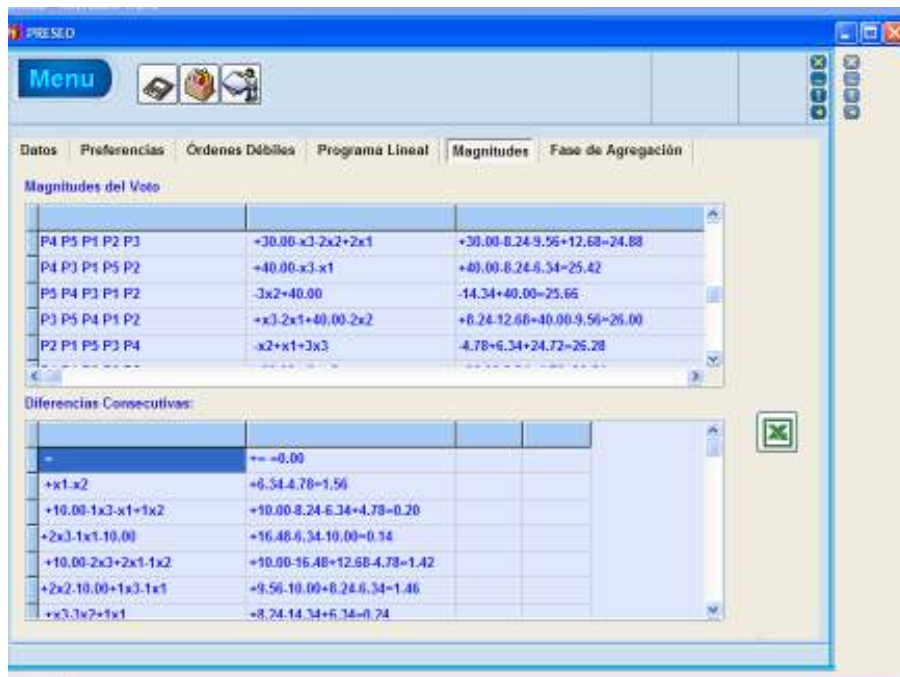


Figura 24m. Magnitudes del voto para el individuo 1.

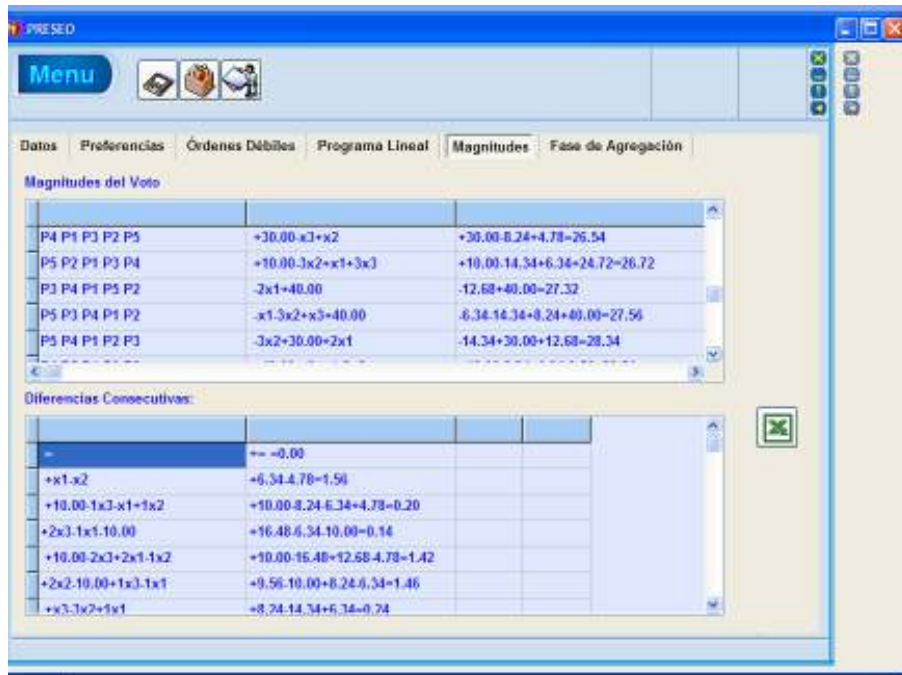


Figura 24n. Magnitudes del voto para el individuo 1.

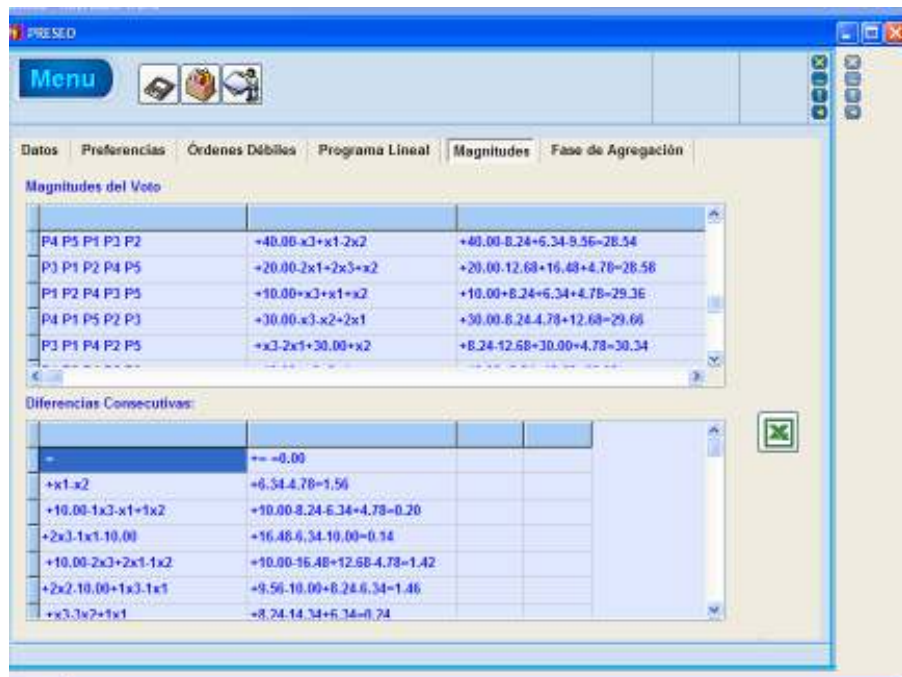


Figura 24o. Magnitudes del voto para el individuo 1.

PRESED

Menu

Datos Preferencias Órdenes Dóbles Programa Lineal **Magnitudes** Fase de Agregación

Magnitudes del Voto

P1 P2 P4 P5 P3	$+10.00 \cdot x3 + 2x1$	$+10.00 + 8.24 + 12.68 - 30.92$
P1 P4 P2 P3 P5	$+20.00 \cdot x1 + x2$	$+20.00 + 6.34 + 4.78 - 31.12$
P1 P2 P3 P4 P5	$+10.00 + 2x3 + x2$	$+10.00 + 16.48 + 4.78 - 31.26$
P4 P1 P3 P5 P2	$+40.00 \cdot x3$	$+40.00 \cdot 8.24 - 31.76$
P5 P4 P1 P3 P2	$-3x2 + x1 + 40.00$	$-14.34 + 6.34 + 40.00 - 32.00$

Diferencias Consecutivas:

=	$= -0.00$	
$+x1 \cdot x2$	$+6.34 \cdot 4.78 = 1.50$	
$+10.00 \cdot 1x3 \cdot x1 + 1x2$	$+10.00 \cdot 8.24 \cdot 6.34 + 4.78 = 0.20$	
$+2x3 \cdot 1x1 \cdot 10.00$	$+16.48 \cdot 6.34 \cdot 10.00 = 0.34$	
$+10.00 \cdot 2x3 + 2x1 \cdot 1x2$	$+10.00 \cdot 16.48 + 12.68 \cdot 4.78 = 1.42$	
$+2x2 \cdot 10.00 + 1x3 \cdot 1x1$	$+9.56 \cdot 10.00 + 8.24 \cdot 6.34 = 1.46$	
$+x3 \cdot 3x2 + 1x1$	$+8.24 \cdot 14.34 + 6.34 = 0.74$	

Figura 24p. Magnitudes del voto para el individuo 1.

PRESED

Menu

Datos Preferencias Órdenes Dóbles Programa Lineal **Magnitudes** Fase de Agregación

Magnitudes del Voto

P3 P1 P2 P5 P4	$+20.00 \cdot 2x1 + 3x3$	$+20.00 \cdot 12.68 + 24.72 = 32.04$
P3 P5 P1 P2 P4	$+30.00 \cdot 2x1 + 3x3 \cdot 2x2$	$+30.00 \cdot 12.68 + 24.72 \cdot 9.56 = 32.48$
P1 P4 P2 P5 P3	$+20.00 \cdot 2x1$	$+20.00 + 12.68 = 32.68$
P4 P1 P5 P3 P2	$+40.00 \cdot x3 \cdot x2 + x1$	$+40.00 \cdot 8.24 \cdot 4.78 + 6.34 = 33.32$
P5 P3 P1 P2 P4	$x1 \cdot 3x2 + 30.00 + 3x3$	$6.34 \cdot 14.34 + 30.00 + 24.72 = 34.04$

Diferencias Consecutivas:

=	$= -0.00$	
$+x1 \cdot x2$	$+6.34 \cdot 4.78 = 1.50$	
$+10.00 \cdot 1x3 \cdot x1 + 1x2$	$+10.00 \cdot 8.24 \cdot 6.34 + 4.78 = 0.20$	
$+2x3 \cdot 1x1 \cdot 10.00$	$+16.48 \cdot 6.34 \cdot 10.00 = 0.34$	
$+10.00 \cdot 2x3 + 2x1 \cdot 1x2$	$+10.00 \cdot 16.48 + 12.68 \cdot 4.78 = 1.42$	
$+2x2 \cdot 10.00 + 1x3 \cdot 1x1$	$+9.56 \cdot 10.00 + 8.24 \cdot 6.34 = 1.46$	
$+x3 \cdot 3x2 + 1x1$	$+8.24 \cdot 14.34 + 6.34 = 0.74$	

Figura 24q. Magnitudes del voto para el individuo 1.

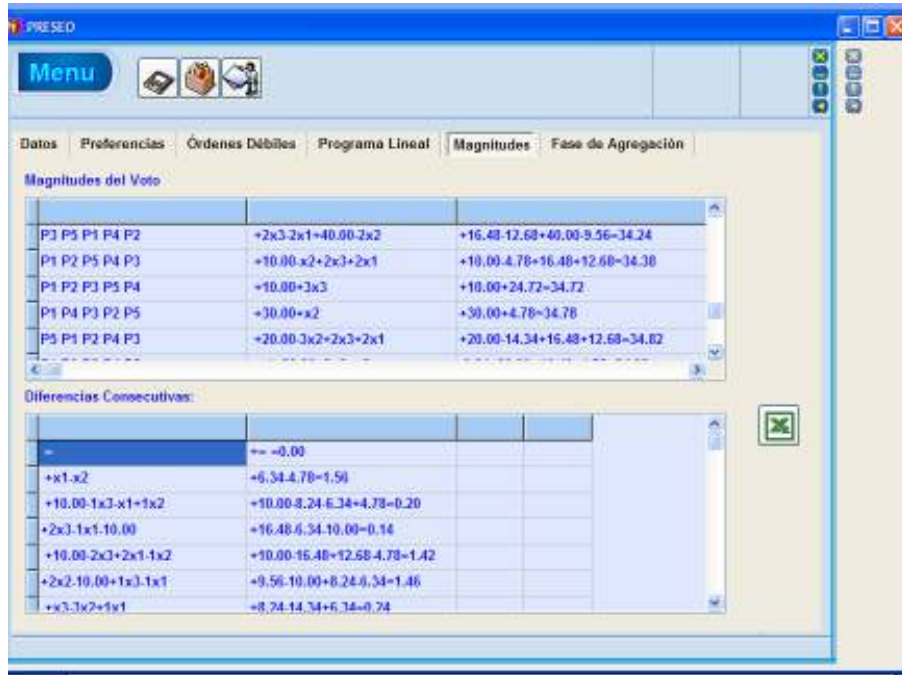


Figura 24r. Magnitudes del voto para el individuo 1.

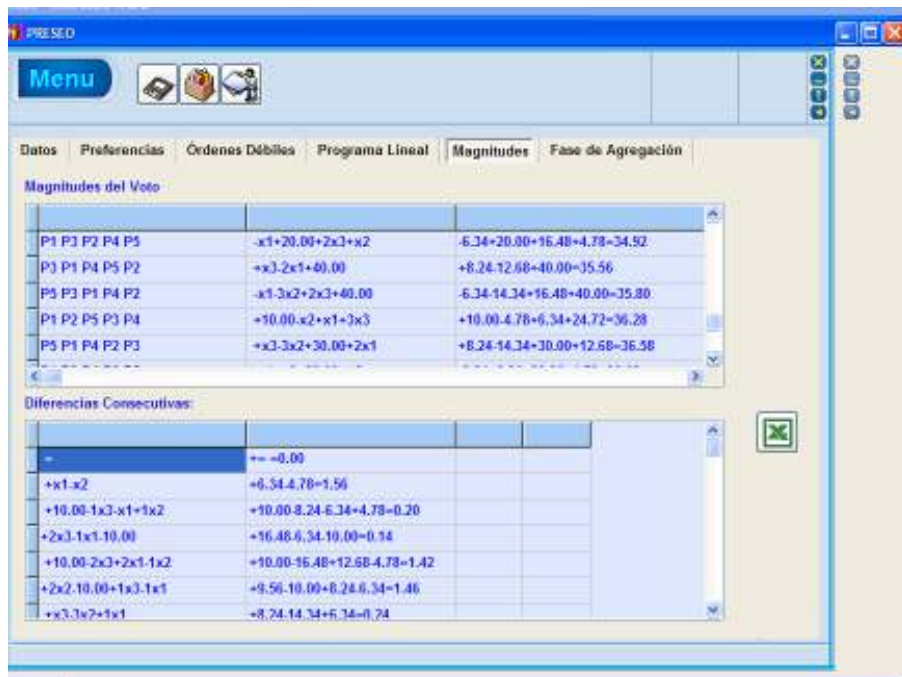


Figura 24s. Magnitudes del voto para el individuo 1.

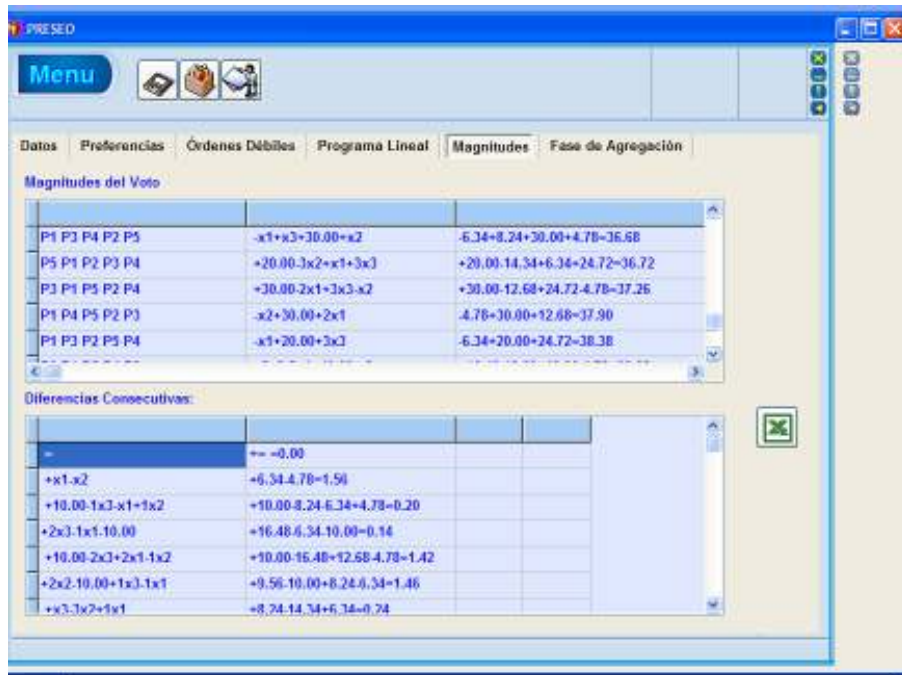


Figura 24t. Magnitudes del voto para el individuo 1.

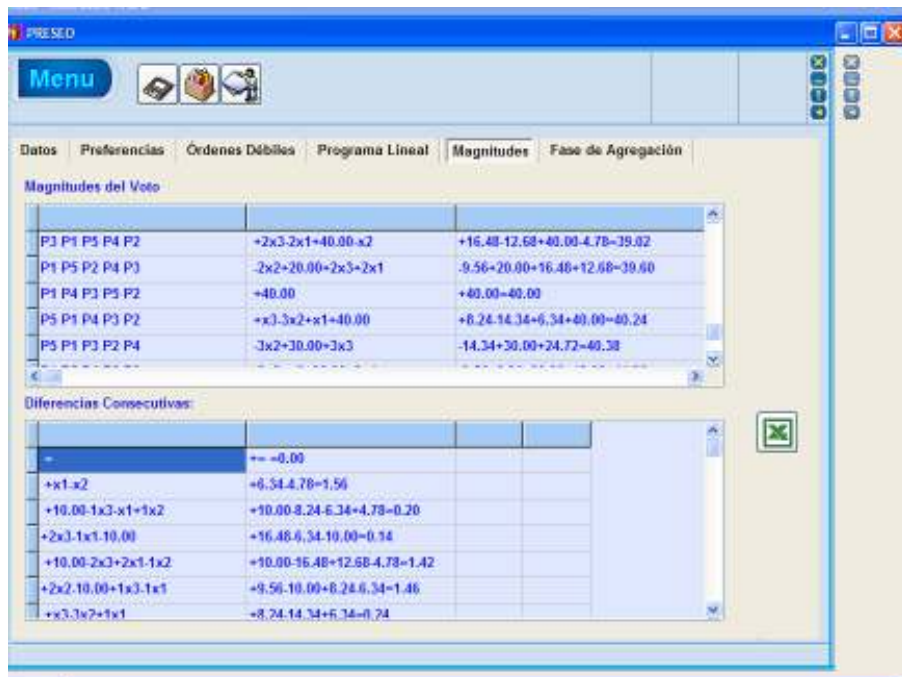


Figura 24u. Magnitudes del voto para el individuo 1.

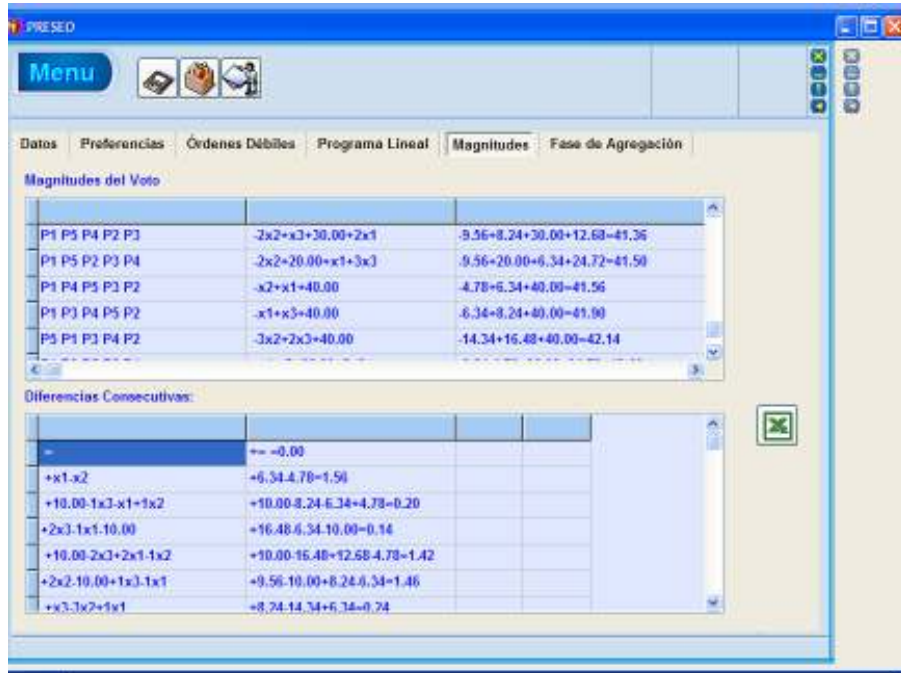


Figura 24v. Magnitudes del voto para el individuo 1.

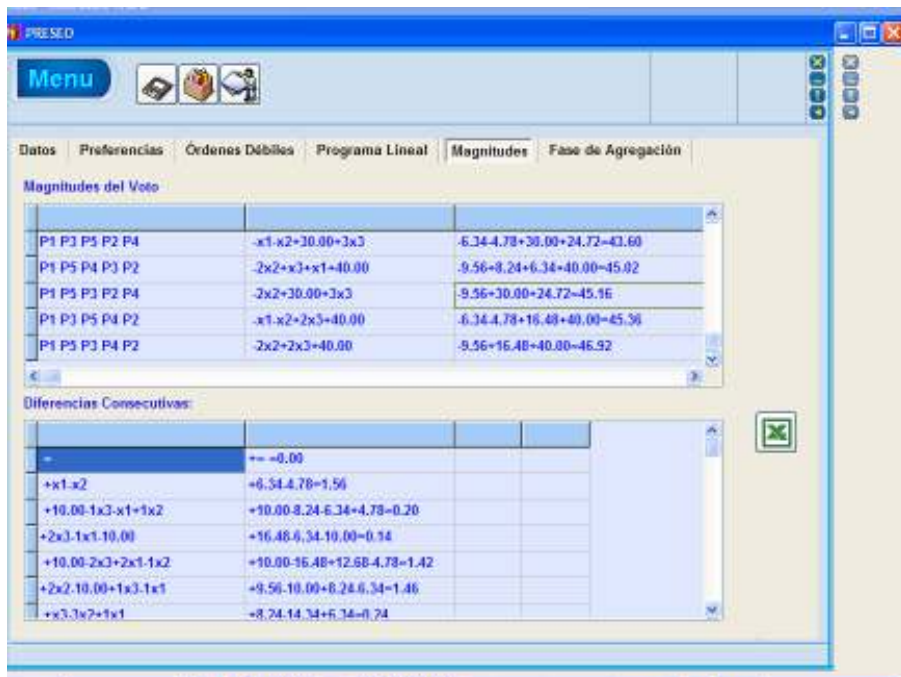


Figura 24w Magnitudes del voto para el individuo 1.

- Problema con 4 alternativas para el individuo 1 (Jefe):

The screenshot shows the PRESED software interface. The 'Magnitudes del Voto' table is as follows:

Alternativas	Expresión	Valor
P2 P4 P3 P1		
P4 P2 P3 P1	$+10.00 \cdot x1$	$+10.00 \cdot 8.42 = 1.58$
P2 P3 P4 P1	$+x1 \cdot x2$	$+8.42 \cdot 6.32 = 2.10$
P4 P3 P2 P1	$+20.00 \cdot x1 \cdot x2$	$+20.00 \cdot 8.42 \cdot 6.32 = 5.26$
P3 P2 P4 P1	$+10.00 \cdot 2x2 + x1$	$+10.00 \cdot 12.64 + 8.42 = 5.78$

The 'Diferencias Consecutivas' table is as follows:

Expresión	Valor
=	== -0.00
$+10.00 \cdot x1$	$+10.00 \cdot 8.42 = 1.58$
$+2x1 \cdot 1x2 \cdot 10.00$	$+16.84 \cdot 6.32 \cdot 10.00 = 0.52$
$+20.00 \cdot 2x1$	$+20.00 \cdot 16.84 = 3.16$
$-10.00 \cdot 1x2 + 2x1$	$-10.00 \cdot 6.32 + 16.84 = 0.52$
$+3x2 \cdot 10.00 \cdot 1x1$	$+18.96 \cdot 10.00 \cdot 8.42 = 0.54$
$-3x2 + 20.00$	$-18.96 + 20.00 = 1.04$

Figura 10. Problema con 4 alternativas. Magnitudes del voto para el individuo 1.

The screenshot shows the PRESED software interface. The 'Magnitudes del Voto' table is as follows:

Alternativas	Expresión	Valor
P2 P4 P1 P3	$+x2$	$+6.32 = 6.32$
P3 P4 P2 P1	$-2x2 + 20.00$	$12.64 + 20.00 = 7.36$
P4 P2 P1 P3	$+10.00 \cdot x1 + x2$	$+10.00 \cdot 8.42 + 6.32 = 7.90$
P2 P3 P1 P4	$+2x1 \cdot x2$	$+16.84 \cdot 6.32 = 10.52$
P3 P2 P1 P4	$+10.00 \cdot 2x2 + 2x1$	$+10.00 \cdot 12.64 + 16.84 = 14.20$

The 'Diferencias Consecutivas' table is as follows:

Expresión	Valor
=	== -0.00
$+10.00 \cdot x1$	$+10.00 \cdot 8.42 = 1.58$
$+2x1 \cdot 1x2 \cdot 10.00$	$+16.84 \cdot 6.32 \cdot 10.00 = 0.52$
$+20.00 \cdot 2x1$	$+20.00 \cdot 16.84 = 3.16$
$-10.00 \cdot 1x2 + 2x1$	$-10.00 \cdot 6.32 + 16.84 = 0.52$
$+3x2 \cdot 10.00 \cdot 1x1$	$+18.96 \cdot 10.00 \cdot 8.42 = 0.54$
$-3x2 + 20.00$	$-18.96 + 20.00 = 1.04$

Figura 25a. Problema con 4 alternativas. Magnitudes del voto para el individuo 1.

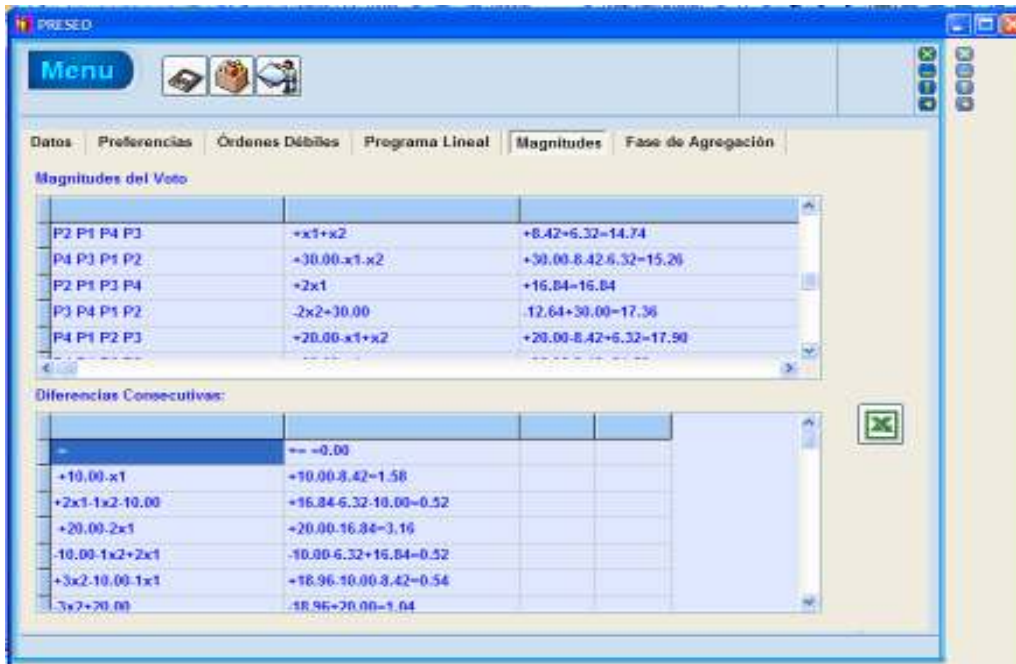


Figura 25b. Problema con 4 alternativas. Magnitudes del voto para el individuo 1.

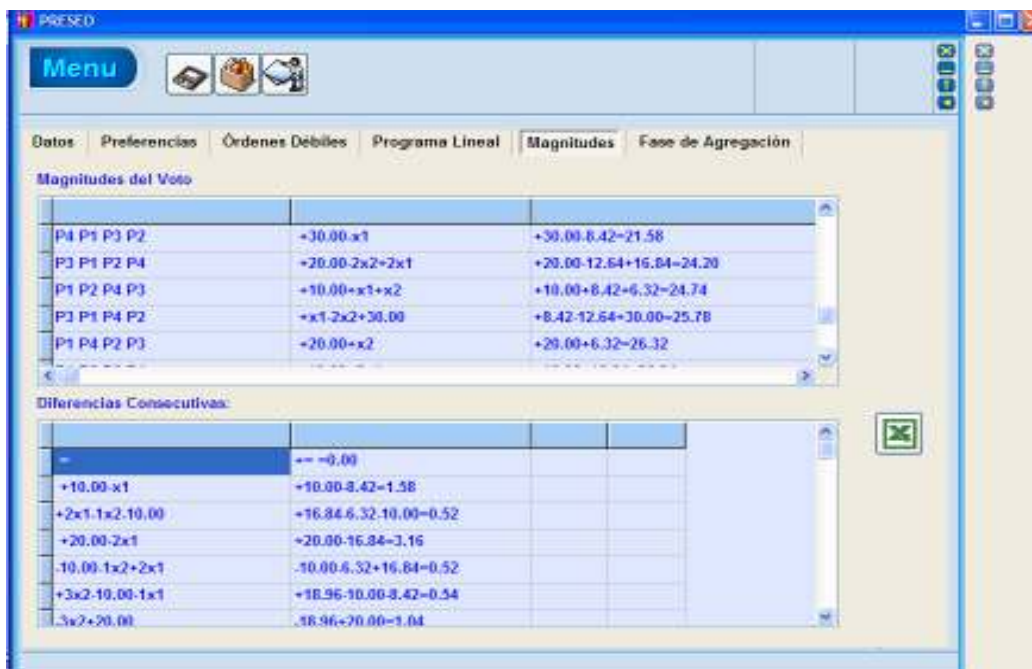


Figura 25c. Problema con 4 alternativas. Magnitudes del voto para el individuo 1.

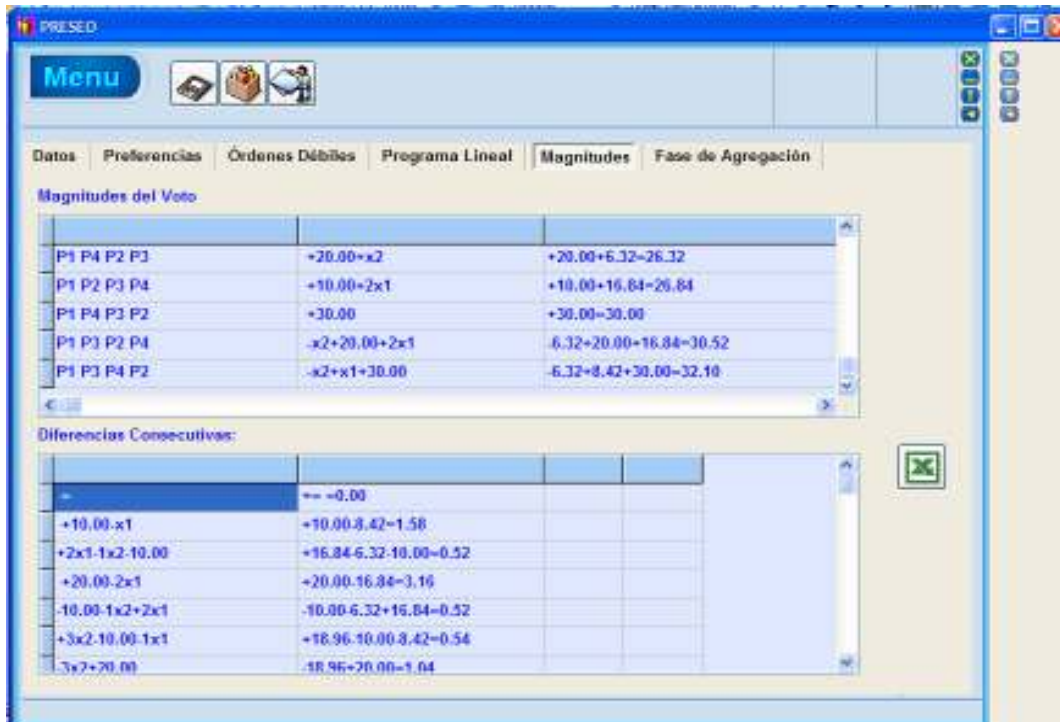


Figura 25d. Problema con 4 alternativas. Magnitudes del voto para el individuo 1.

APENDICE II. Código Fuente.

unit principal;

interface

uses

Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
Dialogs, ComCtrls, XPMan, StdCtrls, ExtCtrls, Grids, Buttons;

type

Tregistro = record

opc1 : Integer;

opc2 : Integer;

opc3 : Integer;

opc4 : Integer;

opc5 : Integer;

End;

TForm1 = class(TForm)

Image1: TImage;

Image2: TImage;

Image3: TImage;

Image4: TImage;

Image5: TImage;

Image6: TImage;

Image7: TImage;

Image8: TImage;

Image9: TImage;

Label1: TLabel;

Label2: TLabel;

Label3: TLabel;

PageControl1: TPageControl;

XPManifest1: TXPManifest;

TabSheet1: TTabSheet;

TabSheet2: TTabSheet;

TabSheet3: TTabSheet;

TabSheet4: TTabSheet;

Label4: TLabel;

Edit1: TEdit;

UpDown1: TUpDown;

StringGrid1: TStringGrid;

Label5: TLabel;

Edit2: TEdit;

UpDown2: TUpDown;

Edit3: TEdit;

Edit4: TEdit;

StringGrid2: TStringGrid;

ComboBox1: TComboBox;

Label6: TLabel;

StringGrid3: TStringGrid;

```

Image10: TImage;
Label9: TLabel;
ComboBox2: TComboBox;
ProgressBar1: TProgressBar;
StringGrid4: TStringGrid;
SpeedButton1: TSpeedButton;
Label11: TLabel;
Label12: TLabel;
Label13: TLabel;
SpeedButton2: TSpeedButton;
SpeedButton3: TSpeedButton;
SaveDialog1: TSaveDialog;
StringGrid5: TStringGrid;
StringGrid6: TStringGrid;
TabSheet5: TTabSheet;
TabSheet6: TTabSheet;
StringGrid7: TStringGrid;
StringGrid8: TStringGrid;
Label7: TLabel;
Label8: TLabel;
StringGrid9: TStringGrid;
SpeedButton4: TSpeedButton;
SpeedButton5: TSpeedButton;
SpeedButton6: TSpeedButton;
ProgressBar2: TProgressBar;
ProgressBar3: TProgressBar;
Label10: TLabel;
Label14: TLabel;
StringGrid10: TStringGrid;
RichEdit1: TRichEdit;
SpeedButton7: TSpeedButton;
OpenDialog1: TOpenDialog;
SpeedButton8: TSpeedButton;
SpeedButton9: TSpeedButton;
SpeedButton10: TSpeedButton;
procedure TabSheet5Show(Sender: TObject);

procedure FormCreate(Sender: TObject);
procedure StringGrid1Click(Sender: TObject);
procedure UpDown1Click(Sender: TObject; Button: TUDBtnType);
procedure Edit3Change(Sender: TObject);
procedure UpDown2Click(Sender: TObject; Button: TUDBtnType);
procedure StringGrid2Click(Sender: TObject);
procedure Edit4Change(Sender: TObject);
procedure ComboBox1Change(Sender: TObject);
procedure TabSheet2Show(Sender: TObject);
Procedure DibujaOPciones();
procedure Edit3KeyPress(Sender: TObject; var Key: Char);
procedure StringGrid3Click(Sender: TObject);
procedure Image10Click(Sender: TObject);

```



```

procedure Image10MouseMove(Sender: TObject; Shift: TShiftState; X,
  Y: Integer);
procedure Edit4KeyPress(Sender: TObject; var Key: Char);
procedure TabSheet3Show(Sender: TObject);
procedure ComboBox2Change(Sender: TObject);
Procedure ordenatemp();
Procedure ordenatemp_inv();
Procedure Carga();
Procedure rellenagrid();
Procedure Ini_inicio_uno();
Procedure elimina_duplicados();
procedure TabSheet4Show(Sender: TObject);
procedure SpeedButton1Click(Sender: TObject);
Procedure calcula_orden();
Procedure ordena_valores();
procedure SpeedButton2Click(Sender: TObject);
procedure SpeedButton3Click(Sender: TObject);
Procedure Genera_Diferencias();
Procedure grabar_tabla();
Procedure ajustar_Columnas();
Procedure ordena_Primer();
Procedure Ini_Elementos();
Procedure En_Elementos(cual, igual:String; lin:Integer);
Function Suma_Valores():string;
Procedure ordena_ecua(col:Integer);
Procedure busca_ecua();
Procedure Calcula_M();
Function letras_sino(o:string):boolean;
Procedure Genera_Simplex();
Function dame_numero(q:string):String;
Procedure Calcula_Simplex();
Procedure Simplex_Dual();
Procedure Simplex_Primal(ini:integer);
Procedure encuatra_Valores();
Function sustituye(ecua:String):string;
Function Sustitulle_x(lin:Integer):string;
Procedure Calcula_magnitud();
procedure SpeedButton4Click(Sender: TObject);
Function Verifica_1(o:String):String;
procedure SpeedButton5Click(Sender: TObject);
Function checaletra(letra, cual:String):Boolean;
procedure SpeedButton6Click(Sender: TObject);
Procedure Inicia_tablas();
Procedure genrea_Simplex();
Procedure Recopila_tabla(caul:Integer);
function dame_total(texto :string ):string;
Procedure OTotales();
Procedure carga_valor_final( t:Integer; t2,t3:string );
Procedure captura_Valores( pp:integer );
procedure SpeedButton7Click(Sender: TObject);

```

```

procedure SpeedButton8Click(Sender: TObject);
procedure SpeedButton9Click(Sender: TObject);
procedure SpeedButton10Click(Sender: TObject);
procedure Reiniciagrid();
procedure StringGrid9DrawCell(Sender: TObject; ACol, ARow: Integer;
  Rect: TRect; State: TGridDrawState);
private
  Ubicado:Array[1..100,1..50]of Integer;
  inicio, temp:array[1..1000,1..100] of Integer;
  inicio_uno:array[1..100] of integer;
  inicio_dos:array[1..100] of integer;
  cobinaciones:array[1..41000,1..101] of Integer;
  totales:Array[1..41000] of real;
  x1:Array[1..1000,1..10] of String;
  simplex:Array[1..1000,1..100] of string;
  valores_x: Array[1..2,1..8] of String;
  factorial :Integer;
  lin_tab:Integer;
  opcx : Integer;
  mapa : Tbitmap;
  iteracion, cuan_participante:integer;
  Funciones_valor1:Array[1..8 ] of integer;
  Funciones_valor2:Array[1..8,1..3] of String;
  LP:integer;
  { Private declarations }
public
  { Public declarations }
end;

var
  Form1: TForm1;
  registro : tregistro;
  arch_datos : File of Tregistro;

implementation

{$R *.dfm}

procedure TForm1.FormCreate(Sender: TObject);
Var
  a,b:Integer;
begin

  factorial := 7;

  stringgrid1.ColWidths[0] := 10;
  Stringgrid1.ColWidths[1] := 220;

  stringgrid2.ColWidths[0] := 10;
  Stringgrid2.ColWidths[1] := 370;

```

```

stringgrid3.ColWidths[0] := 15;
Stringgrid3.ColWidths[1] := 220;

for a:= 1 to 100 do begin
  for b := 1 to 100 do begin
    ubicado[a,b] := 0;
  end;
end;

mapa := Tbitmap.Create;
mapa.Assign(image10.Picture.Bitmap);

end;

procedure TForm1.StringGrid1Click(Sender: TObject);
begin
  Edit3.Text := Stringgrid1.Cells[1, stringgrid1.Row];
  edit3.SetFocus;
end;

procedure TForm1.UpDown1Click(Sender: TObject; Button: TUDBtnType);
Var
  a:Integer;
begin
  a := Updown1.Position;
  stringgrid1.RowCount := a + 1;
  factorial := 1;
  For a:= 2 to updown1.Position do factorial := factorial * a;

end;

procedure TForm1.Edit3Change(Sender: TObject);
begin

  stringgrid1.Cells[1,stringgrid1.Row] := Edit3.Text;
end;

procedure TForm1.UpDown2Click(Sender: TObject; Button: TUDBtnType);
begin
  stringgrid2.RowCount := updown2.Position + 1;
end;

procedure TForm1.StringGrid2Click(Sender: TObject);
begin

  edit4.Text := Stringgrid2.Cells[1,stringgrid2.Row];
  edit4.SetFocus;

end;

```

```

procedure TForm1.Edit4Change(Sender: TObject);
begin

    stringgrid2.Cells[1,stringgrid2.Row] := edit4.Text;
end;

procedure TForm1.ComboBox1Change(Sender: TObject);
begin

    If combobox1.Text <> 'Seleccione el Usuario' then begin

        DibujaOPciones();
    end;

end;

procedure TForm1.TabSheet2Show(Sender: TObject);
Var
    a:Integer;
begin

    combobox1.Items.Clear;
    For a:= 1 to stringgrid2.RowCount - 1 do begin

        Combobox1.Items.Add(stringgrid2.Cells[1,a]);

    end;

    stringgrid3.RowCount := stringgrid1.RowCount;
    For a:= 1 to stringgrid1.RowCount - 1 do begin
        stringgrid3.Cells[1,a] := stringgrid1.Cells[1,a];
        stringgrid3.Cells[0,a] := Chr( 65 + (a-1));
    end;

end;

Procedure TForm1.DibujaOPciones();
Var
    a, aum :Integer;
begin

    image10.Picture.Bitmap.Assign(mapa);
    for a := 1 to 100 do begin
        If a < Stringgrid1.RowCount then begin
            aum := 0;
            If ubicado[combobox1.ItemIndex, a] = 0 then aum := +5;
            If ubicado[combobox1.ItemIndex, a] = 417 then aum := -7;
            image10.Picture.Bitmap.Canvas.Brush.Color := Rgb(207,220,232);
            image10.Picture.Bitmap.Canvas.Font.Color := Rgb(0,0,255);
        end;
    end;
end;

```

```

    image10.Picture.Bitmap.Canvas.TextOut(ubicado[combobox1.ItemIndex, a] + aum,
20, stringgrid3.Cells[1,a]);
    end;
    end;
end;

```

```

procedure TForm1.Edit3KeyPress(Sender: TObject; var Key: Char);
begin

```

```

    If key = #13 then begin

```

```

        If stringgrid1.Row < stringgrid1.RowCount - 1 then
            Stringgrid1.Row := Stringgrid1.Row + 1;

```

```

        edit3.Text := stringgrid1.Cells[1, Stringgrid1.Row];

```

```

    end;

```

```

end;

```

```

procedure TForm1.StringGrid3Click(Sender: TObject);
begin

```

```

    If combobox1.Text <> 'Seleccione el Usuario' then begin

```

```

        stringgrid3.Cursor := crdrag;
        Image10.Cursor := Crdrag;
    end;

```

```

end;

```

```

procedure TForm1.Image10Click(Sender: TObject);
begin

```

```

    If image10.Cursor = crdrag then begin
        If combobox1.ItemIndex <> -1 then begin
            ubicado[combobox1.ItemIndex, stringgrid3.Row] := opcx;
            DibujaOPciones();
            image10.Cursor := Crdefault;
            stringgrid3.Cursor := crdefault;

```

```

        end;
    end;
end;

```

```

procedure TForm1.Image10MouseMove(Sender: TObject; Shift: TShiftState; X,
    Y: Integer);

```

```

begin
    opcx := x;
end;

```

```

procedure TForm1.Edit4KeyPress(Sender: TObject; var Key: Char);
begin

    If key = #13 then begin

        If stringgrid2.Row < stringgrid2.RowCount - 1 then
            Stringgrid2.Row := Stringgrid2.Row + 1;

        edit4.Text := stringgrid2.Cells[1, Stringgrid2.Row];

    end;

end;

procedure TForm1.TabSheet3Show(Sender: TObject);
Var
    a:Integer;
begin

    combobox2.Items.Clear;
    For a:= 1 to stringgrid2.RowCount - 1 do begin

        Combobox2.Items.Add(stringgrid2.Cells[1,a]);

    end;

    stringgrid4.ColWidths[0]:= 10;
    stringgrid4.ColCount := stringgrid1.RowCount + 3;
    stringgrid4.ColWidths[stringgrid1.RowCount ]:= 500;
    stringgrid4.ColWidths[stringgrid1.RowCount + 1 ]:= 500;
    stringgrid4.ColWidths[stringgrid1.RowCount + 2 ]:= 500;

end;

procedure TForm1.ComboBox2Change(Sender: TObject);
begin
    cuan_participante:= combobox2.ItemIndex;
    Inicia_tablas();
    calcula_orden();
    ordena_valores();
    rellenagrid();
    Genera_Diferencias();

    stringgrid4.Cells[stringgrid4.ColCount-3, 0]:= 'Dif. entre Opciopnes';
    stringgrid4.Cells[stringgrid4.ColCount-2, 0]:= 'Evaluación a priori';
    stringgrid4.Cells[stringgrid4.ColCount-1, 0]:= 'Evaluación';

    ajustar_Columnas();

end;

```

```

Procedure TForm1.Inicia_tablas();
Var
  a,b:Integer;
begin

  For a := 1 to factorial do
    For b := 1 to 101 do
      combinaciones[a,b] := 0;

  For a:= 1 to 8 do begin
    Funciones_valor1[a]:= 1000;
    Funciones_valor2[a,1]:= "";
    Funciones_valor2[a,2]:= "";
    Funciones_valor2[a,3]:= "";
  end;

  For a:= 0 to stringgrid4.ColCount-1 do begin
    For b := 0 to stringgrid4.RowCount - 1 do begin
      stringgrid4.Cells[a,b] := "";
    end;
  end;

  lin_tab:= 1;
  For a := 1 to stringgrid1.RowCount - 1 do begin
    inicio_uno[a] := a;
    inicio_dos[a] := (stringgrid1.RowCount )-a;
  end;
  ordenatemp();
  inicio := temp;
  For a:= 1 to 1000 do begin
    If inicio[a,1]<>0 then begin
      Ini_inicio_uno();
      For b := 1 to stringgrid1.RowCount - 1 do begin
        inicio_uno[b] := inicio[a,b];
      end;
      Ordenatemp_inv();
      Carga();
    end;
  end;
end;

```

```

Procedure TForm1.ordenatemp();
Var
  a,b,c,d,e:Integer;
begin

  c := 1;
  For a := 1 to stringgrid1.RowCount-1 do begin
    for b:= 1 to stringgrid1.RowCount-1 do begin

```

```

If b= 1 then begin
  for d := 1 to stringgrid1.RowCount-1 do begin
    temp[c,d] := inicio_uno[d];
  end;
  c := c + 1;
end else begin
  e := inicio_uno[b];
  inicio_uno[b] := inicio_uno[b-1];
  inicio_uno[b-1] := e;

  e := inicio_dos[b];
  inicio_dos[b] := inicio_dos[b-1];
  inicio_dos[b-1] := e;
  for d := 1 to stringgrid1.RowCount - 1 do begin
    temp[c,d] := inicio_uno[d];
    temp[c + 1,d] := inicio_dos[d];

    end;
  c:= c + 2;
end;
end;
end;
end;
end;

```

Procedure TForm1.ordenatemp_inv();

Var

a,b,c,d,e:Integer;

begin

c := 1;

For a := 1 to stringgrid1.RowCount - 1 do begin

for b:= stringgrid1.RowCount - 1 downto 1 do begin

If b = stringgrid1.RowCount - 1 then begin

for d := 1 to stringgrid1.RowCount - 1 do begin

temp[c,d] := inicio_uno[d];

end;

c := c + 1;

end else begin

e := inicio_uno[b];

inicio_uno[b] := inicio_uno[b + 1];

inicio_uno[b + 1] := e;

for d := 1 to stringgrid1.RowCount - 1 do begin

temp[c,d] := inicio_uno[d];

end;

c := c + 1;

end;

end;

end;

end;


```

Procedure TForm1.Carga();
Var
  a,b,c,d:Integer;
  Sino,t1,t2:String;
begin

  For a := 1 to 1000 do begin
    If temp[a,1]<>0 then begin
      sino := 'N';
      For c := 1 to lin_tab - 1 do begin

        t1 := "";
        For d := 1 to stringgrid1.RowCount - 1 do
          t1 := t1 + Inttostr(temp[a,d]) + '*';

        t2 := "";
        For d := 1 to stringgrid1.RowCount - 1 do
          t2 := t2 + Inttostr(cobinaciones[c,d]) + '*';

        If t1 = t2 then sino := 'S';

      end;
      If Sino = 'N' then begin
        For b := 1 to stringgrid1.RowCount - 1 do begin
          cobinaciones[lin_tab,b] := temp[a,b];
        end;
        lin_tab := lin_tab + 1;
      end;
    end;
  end;
end;

```

```

Procedure TForm1.rellenagrid();
Var
  a,b,c,b2, elm, a2, cual_dato:Integer;
  total, tot1,tot2:Real;
  tem1,tem2:String;
  valores:array[1..8] of real;
  elementos:array[1..1000,1..2] of String;
  elementos2:Array[1..1000] of Integer;

```

```

Function letra(v:Real):String;
Var
  aa: Integer;
begin

```

```

  for aa:= 1 to 8 do begin

```

```

If valores[aa] = 0 then begin
    valores[aa] := v;
    letra:='x' + inttostr(aa);
    exit;
end;
If Valores[aa] = v then begin
    letra:= 'x' + inttostr(aa);
    exit;
end;

end;

end;

Procedure Organiza_ecua(que,como:String);
var
    aa : Integer;
begin

    For aa := 1 to 1000 do begin

        If elementos[aa, 1] = " then begin
            elementos[aa,1] := que;
            elementos[aa,2] := como;
            exit;
        end;

    end;

end;

function Calcula_ecua():String;
Var
    aa, bb,cc,vv : Integer;
    dd,signo:string;
    si_x, ttt, temporal :string;
    valor1, valor2:Real;
begin

    for aa := 1 to 1000 do elementos2[aa] := 0;

    for aa := 1 to 1000 do begin

        If elementos[aa,1] <> " then begin
            If elementos[aa,2]='+' then elementos2[aa]:= 1 else elementos2[aa]:= -1;
            for bb := aa + 1 to 1000 do begin

                If elementos[bb,1]<>" then begin
                    If elementos[aa,1] = elementos[bb,1] then begin
                        if elementos[bb,2] = '+' then begin

```

```

    elementos2[aa]:= elementos2[aa] + 1;
end else begin
    elementos2[aa]:= elementos2[aa] - 1;
end;
elementos[bb,1] := "";
end;
end;

end;
end;

end;

for aa := 1 to 1000 do begin

If elementos2[aa] <> 0 then begin
If elementos2[aa] < 0 then signo := '-' else signo := '+';
If (elementos2[aa] = 1) or (elementos2[aa] = -1) then begin
    ttt := elementos[aa,1];
    si_x := "";
    For cc := 1 to length(ttt) do begin
        If ttt[cc] = 'x' then si_x := 'S';
    end;
    If si_x = 'S' then begin
        dd := dd + signo + elementos[aa,1];

    end else begin
        val( elementos[aa,1], valor1, vv );
        If valor1 > 0 then begin
            dd := dd + signo +elementos[aa,1];

        end;
    end;
end else begin
    ttt := elementos[aa,1];
    si_x := "";
    For cc := 1 to length(ttt) do begin
        If ttt[cc] = 'x' then si_x := 'S';
    end;
    If si_x = 'S' then begin
        If elementos2[aa]<0 then
            dd := dd + inttostr(elementos2[aa])+elementos[aa,1]
        else
            dd := dd + signo +inttostr(elementos2[aa])+elementos[aa,1];

    end else begin
        val( elementos[aa,1], valor1, vv );
        If valor1 > 0 then begin
            valor2 := valor1 * elementos2[aa];
            str( valor2:4:2, temporal);

```

```

        dd := dd + signo + temporal;
    end;

    end;
end;
end;

Calcula_ecua := trimleft(dd);

end;

begin

c := 1;
for a:= 1 to 8 do valores[a] := 0;
for a:= 1 to 1000 do begin
    elementos[a, 1] := "";
end;
cual_dato := 1;
label11.Caption := 'Reellenado Grid';
label11.Repaint;
For a:= 1 to factorial do begin
    progressbar1.Position := a;
    If cobinaciones[a,1]<>0 then begin

        for a2:= 1 to 1000 do elementos[a2, 1] := "";

        stringgrid4.RowCount := c + 1;
        stringgrid1.Cells[stringgrid1.ColCount, c] := "";
        total := 0;
        for b := 1 to stringgrid1.RowCount - 1 do begin
            // Cambio de Letra
            stringgrid4.Cells[b,c] := stringgrid1.Cells[1,cobinaciones[a,b]];
            { crear formula }
            For b2 := b to stringgrid1.RowCount - 1 do begin
                If cobinaciones[a,b+50] < cobinaciones[a,b2 + 50] then begin
                    tot1 := ((cobinaciones[a,b2+50])* 10 / 417);
                    tot2 := ((cobinaciones[a,b+50])* 10 / 417);

                    str(tot1:3:2,tem1);
                    str(tot2:3:2,tem2);
                    If stringgrid4.Cells[stringgrid4.ColCount-2, c] <> " then
                        stringgrid4.Cells[stringgrid4.ColCount-2, c] :=
stringgrid4.Cells[stringgrid4.ColCount-2, c] + '+';

                    stringgrid4.Cells[stringgrid4.ColCount-2, c] :=
stringgrid4.Cells[stringgrid4.ColCount-2, c] +

```

```

        ('+tem1 + '-' + tem2 + ');

    If (tot1 <> 10) and (tot1 <> 0) then tem1 := letra(tot1);
    If (tot2 <> 10) and (tot2 <> 0) then tem2 := letra(tot2);

    // buscar valores
    carga_valor_final( cobinaciones[a,b2+50], tem1,
stringgrid1.Cells[1,cobinaciones[a,b2]]);
    carga_valor_final( cobinaciones[a,b+50], tem2,
stringgrid1.Cells[1,cobinaciones[a,b]]);

    Organiza_ecua(tem1,'+');
    Organiza_ecua(tem2,'-');

    If stringgrid4.Cells[stringgrid4.ColCount-1, c] <> " then
        stringgrid4.Cells[stringgrid4.ColCount-1, c] :=
stringgrid4.Cells[stringgrid4.ColCount-1, c] + '+';

        stringgrid4.Cells[stringgrid4.ColCount-1, c] :=
stringgrid4.Cells[stringgrid4.ColCount-1, c] +
        ('+tem1 + '-' + tem2 + ');

    end;
end;
end;
Str(totales[a]:3:2,tem1);
stringgrid4.Cells[stringgrid4.ColCount-2, c] :=
stringgrid4.Cells[stringgrid4.ColCount-2, c] + '=' + tem1;

stringgrid4.Cells[stringgrid4.ColCount-1, c] :=
stringgrid4.Cells[stringgrid4.ColCount-1, c] +'='+ Calcula_ecua();

    c := c + 1;
    cual_dato := cual_dato + 1;
end;
end;

end;

Procedure TForm1.Ini_inicio_uno();
Var
    a:Integer;
begin
    For a := 1 to 100 do inicio_Uno[a] := 0;
end;

```

```

Procedure TForm1.elimina_duplicados();
Var
  a,b,c:Integer;
  t1,t2:String;
begin
  label11.Caption := 'Eliminado Duplicados';
  label11.Repaint;

  For a := 1 to 41000 do begin
    t1 := "";
    progressbar1.Position := a;
    If cobinaciones[a,1]<> 0 then begin

      For c := 1 to stringgrid1.RowCount - 1 do
        t1 := t1 + Inttostr(cobinaciones[a,c]) + '*';

      For b:= a + 1 to 41000 do begin
        If cobinaciones[b,1]<> 0 then begin
          t2 := "";

          For c := 1 to stringgrid1.RowCount - 1 do
            t2 := t2 + Inttostr(cobinaciones[b,c])+'*';

          If t1 = t2 then begin
            cobinaciones[b,1]:=0;
          end;
        end;
      end;
    end;
  end;

end;

procedure TForm1.TabSheet4Show(Sender: TObject);
begin
  stringgrid5.ColWidths[0]:= 10;
  stringgrid5.ColWidths[1]:= 200;
  stringgrid5.ColWidths[2]:= 200;
  stringgrid5.ColWidths[3]:= 200;

  busca_ecua();
  Calcula_M();
  Reiniciagrid();
  Genera_Simplex();
  Calcula_Simplex();

end;

procedure TForm1.SpeedButton1Click(Sender: TObject);
Var

```

```

a,b:Integer;
Archivo:Textfile;
t:String;
begin

If savedialog1.Execute then begin
  assignfile(archivo,savedialog1.filename+'.csv');
  Rewrite(archivo);

  For b := 1 to stringgrid4.RowCount - 1 do begin
    t := "";
    for a := 1 to stringgrid4.ColCount - 1 do begin
      t := t + ""+stringgrid4.cells[a,b]+""+'';
    end;
    Writeln(Archivo,t);
  end;
  Closefile(archivo);
end;

end;

Procedure TForm1.calcula_orden();
Var
  a,b,b2:Integer;
  Total:Real;
begin
  label11.Caption := 'Reorganizando tabla';
  label11.Repaint;
  For a:= 1 to factorial do begin
    progressbar1.Position := a;
    If cobinaciones[a,1] <> 0 then begin
      For b := 1 to stringgrid1.RowCount - 1 do begin
        cobinaciones[a,b+50] := Ubicado[cuan_participante,cobinaciones[a,b]];
      end;
      total := 0;
      For b := 1 to stringgrid1.RowCount - 1 do begin
        for b2:= b to stringgrid1.RowCount - 1 do begin
          If (cobinaciones[a,b+50] < cobinaciones[a,b2+50]) then begin
            Total := total + ((cobinaciones[a,b2+50] - cobinaciones[a,b+50]) * 10 / 417);;
          end;
        end;
      end;
      Totales[a] := total;
    end;
  end;
end;

end;

```

```

Procedure TForm1.ordena_valores();
Var
  a,b,c:Integer;
  temp:Array[1..101] of integer;
  tem:Real;
begin
  label11.caption := 'Organiza Valores';
  label11.repaint;

  For a:= 1 to factorial do begin
    progressbar1.Position := a;
    label11.Repaint;
    If combinaciones[a,1] <> 0 then begin
      for b := a + 1 to factorial do begin
        If totales[a] > totales[b] then begin

          For c := 1to 101 do begin
            temp[c] := combinaciones[a,c];
          end;

          For c := 1to 101 do begin
            combinaciones[a,c] := combinaciones[b,c];
          end;

          For c := 1to 101 do begin
            combinaciones[b,c] := temp[c];
          end;

          tem := totales[a];
          totales[a] := totales[b];
          totales[b]:= tem;

          end;
        end;
      end;
    end;

  end;

end;

procedure TForm1.SpeedButton2Click(Sender: TObject);
begin
  ubicado[combobox1.ItemIndex, stringgrid3.Row] := 0;
  DibujaOPciones();
  image10.Cursor := Crdefault;
  stringgrid3.Cursor := crdefault;

end;

```



```

procedure TForm1.SpeedButton3Click(Sender: TObject);
begin
    ubicado[combobox1.ItemIndex, stringgrid3.Row] := 417;
    DibujaOPciones();
    image10.Cursor := Crdefault;
    stringgrid3.Cursor := crdefault;

end;

Procedure TForm1.Genera_Diferencias();
Var
    ff,rr:Integer;
    si_x, opc, vale:String;
begin
    For ff := stringgrid4.RowCount - 1 downto 3 do begin
        {en este punto va la limpia de elementos}
        Ini_Elementos();

        opc := stringgrid4.Cells[stringgrid4.ColCount-1, ff];

        En_Elementos( opc,'S',1);

        opc := stringgrid4.Cells[stringgrid4.ColCount-1, ff - 1];

        En_Elementos( opc,'S',2);

        stringgrid4.Cells[stringgrid4.ColCount-3, ff] := Suma_Valores();

    end;

    stringgrid4.Cells[stringgrid4.ColCount-3, 1] :=
stringgrid4.Cells[stringgrid4.ColCount-1, 1];
    si_x := "";
    vale := " ";
    opc := stringgrid4.Cells[stringgrid4.ColCount-1, 2];

    for rr:= 1 to length(opc) do begin
        If si_x = 'S' then begin
            vale := vale + opc[rr];
        end;
    end;
end;

```

```

    If opc[rr] = '=' then begin
        si_x := 'S';
    end;

end;

stringgrid4.Cells[stringgrid4.ColCount-3, 2] := vale;

end;

Procedure TForm1.grabar_tabla();
Var
    aa:integer;
    archivo:Textfile;
begin

assignfile(archivo, 'c:\er.csv');
rewrite(archivo);
For aa := 0 to 1000 do begin

    If cobinaciones[aa,1]<>0 then begin
        Writeln(archivo, stringgrid1.Cells[1,cobinaciones[aa,1]]+',' +
            stringgrid1.Cells[1,cobinaciones[aa,2]]+',' +
            stringgrid1.Cells[1,cobinaciones[aa,3]]);
    end;

end;

closefile(archivo);

end;

Procedure TForm1.ajusticar_Columnas();
var
    aa,bb, ll :Integer;
    texto:String;
begin

    For aa := 1 to stringgrid4.ColCount - 1 do begin

        ll := 0;
        for bb := 1 to stringgrid4.RowCount - 1 do begin

            texto := stringgrid4.Cells[aa,bb];
            If ll < length(texto) then begin
                ll := length(texto)
            end;
        end;
    end;
end;

```

```

end;

stringgrid4.ColWidths[aa] := ll * 10;

end;

end;

Procedure TForm1.ordena_Primer();
Var
  aa,bb,cc:Integer;
begin

  bb := 1;
  For aa := 1 to 41000 do begin

    If cobinaciones[aa,1] <> 0 then begin

      for cc := 1 to stringgrid1.RowCount - 1 do
        cobinaciones[bb,cc] := cobinaciones[aa,cc];
        bb := bb + 1;

      end;

    end;

  end;

end;

end;

Procedure TForm1.Ini_Elementos();
Var
  a,b:Integer;
Begin

  For a := 1 to 1000 do begin
    For b := 1 to 10 do Begin
      x1[a,b] := "";

    end;
  end;

End;

Procedure TForm1.En_Elementos(cual,igual:String;lin:Integer);
Var
  rr, ww,zz:Integer;
  si_x, vale, opc :String;

Begin
  If igual = 'S' then begin

```

```

si_x := "";
vale := "";
opc := cual;
for rr:= 1 to length(cual) do begin

    If si_x = 'S' then begin
        vale := vale + cual[rr];
    end;
    If opc[rr] = '=' then begin
        si_x := 'S';
    end;
end;
end else begin

    vale := cual;

end;

ww := 0;
zz := 1;
opc := "";
for rr := 1 to length(vale) do begin
    If (vale[rr] = '+') or (vale[rr] = '-') then begin

        ww := ww + 1;

        If ww = 2 then begin
            If (opc <> '-') and (opc <> '+') then begin
                x1[zz,lin] := verifica_1(opc);
                If (vale[rr] <> ')') and (vale[rr] <> '(') then
                    opc := vale[rr]
                else
                    opc := "";
                zz := zz + 1;
            end;
            ww := 1;

        end else begin

            If (vale[rr] <> ')') and (vale[rr] <> '(') then
                opc := opc + vale[rr];

        end;

    end else begin
        If (vale[rr] <> ')') and (vale[rr] <> '(') then
            opc := opc + vale[rr];

    end;

end;
end;

```

```
x1[zz,lin] := Verifica_1(opc);
```

```
End;
```

```
Function TForm1.Verifica_1(o:String):String;  
begin
```

```
  If o[1]='x' then
```

```
    o := '+1' + o
```

```
  else
```

```
    If o[2]='x' then
```

```
      If (o[1]='-') or (o[1]='+') then
```

```
        o := o[1] + '1' + copy(o,2,length(o)- 1);
```

```
  Verifica_1 := o;
```

```
end;
```

```
Function TForm1.Suma_Valores():String;
```

```
Var
```

```
  ff,rr,ww,zz, lugarx, lugarx2, checar:Integer;
```

```
  si_x, SI_X2,opc, opc2, val2, vale:String;
```

```
  valor1,valor2,valor:Real;
```

```
Begin
```

```
  For rr := 1 to 1000 do begin
```

```
    If x1[rr,1] = '-' then x1[rr,1] := ";
```

```
    If x1[rr,1] = '+' then x1[rr,1] := ";
```

```
    If (x1[rr,1] <> ") then begin
```

```
      si_x := ";
```

```
      opc := x1[rr,1];
```

```
      opc := trimleft(opc);
```

```
      For zz:= 1 to length(opc) do begin
```

```
        If opc[zz] = 'x' then begin
```

```
          si_x := 'S';
```

```
          lugarx := zz;
```

```
        end;
```

```
      end;
```

```
    for ww:= 1 to 1000 do begin
```

```

If x1[ww,2] = '-' then x1[ww,2] := "";
If x1[ww,2] = '+' then x1[ww,2] := "";

if x1[ww,2] <> " then begin
    si_x2 := "";
    opc := x1[ww,2];
    opc := trimleft(opc);
    For zz:= 1 to length(opc) do begin

        If opc[zz] = 'x' then begin
            si_x2 := 'S';
            lugarx2 := zz;
        end;

    end;

end;

If (si_x <> 'S') and (si_x2 <> 'S') then begin

    opc:= trimleft(x1[rr,1]);
    opc2:= trimleft(x1[ww,2]);

    val( opc, valor1, checar);
    val( opc2, valor2, checar);
    valor := valor1 - valor2;
    str(valor:4:2,vale);
    IF valor = 0 then begin
        x1[rr,1]:= "";
        x1[ww,2]:= "";
        break;
    end else begin
        If valor>0 then vale := '+'+vale;
        x1[rr,1]:= vale;
    end;
    x1[ww,2]:= "";
end else begin
    If (si_x = 'S') and (si_x2 = 'S') then begin
        opc:= trimleft(x1[rr,1]);
        opc2:= trimleft(x1[ww,2]);
        If (opc[lugarx] = 'x') and (opc2[lugarx2] = 'x') and
        (opc[lugarx+1] = opc2[lugarx2+1] ) then begin
            vale := copy( opc, 1, lugarx-1);
            val2 := copy( opc2, 1, lugarx2-1);

            If length(vale) = 1 then vale := vale +'1';
            If length(val2) = 1 then val2 := val2 +'1';
            If (strtoint( vale ) - strtoint( val2 )) >=0 then begin
                If (strtoint( vale ) - strtoint( val2 )) = 0 then begin
                    x1[rr,1] := "";
                    x1[ww,2] := "";
                end;
            end;
        end;
    end;
end;

```

```

                break;
            end else begin
                x1[rr,1] := '+' + inttostr(strtoint( vale ) - strtoint(
val2 )) + copy(opc,lugarx,length(opc));
            end;
            end else begin
                x1[rr,1] := inttostr(strtoint( vale ) - strtoint( val2 )) +
copy(opc,lugarx,length(opc));
            end;
            x1[ww,2] := "";
        end;
    end;
end;
end;
end;
end;
end;
end;
end;

```

```

opc := "";
for rr := 1 to 1000 do begin
    If x1[rr,1] <> "" then begin
        opc := opc + x1[rr,1];
    end;
end;
for rr := 1 to 1000 do begin

    if x1[rr,2] <> "" then begin
        opc2 := x1[rr,2];
        {IF (opc2[1] <> '0') and (opc2[2] <> '0') then begin}
        opc2 := trimleft(opc2);
        If opc2[1] = '-' then begin
            opc := opc + '+' + copy(opc2, 2, length(x1[rr,2]));
        end else begin
            opc := opc + '-' + copy(opc2, 2, length(x1[rr,2]));
        end;
    end;
end;
{end;}
end;

Suma_Valores := opc;

```

end;

Procedure TForm1.ordena_ecua(col:Integer);

Var

a,b:Integer;

```

    tem:String;
begin
    for a := 1 to 1000 do begin
        If x1[a,col]=" Then break;
        for b := a + 1 to 1000 do begin
            If x1[b,col] = " then break;
            If x1[a,col] > x1[b,col] then begin
                tem := x1[a,col];
                x1[a,col] := x1[b,col];
                x1[b,col] := tem;
            end;
        end;
    end;
end;

Procedure TForm1.busca_ecua();
Var
    a, b,c,d, temp:Integer;
    sino:String;
    opc1,opc2:string;
    v1:real;
begin

    Stringgrid5.RowCount := 2;
    stringgrid5.Cells[1,1] := stringgrid4.Cells[stringgrid4.ColCount - 3, 2];

    For a := 3 to stringgrid4.RowCount - 1 do begin

        sino := 'N';
        For b := 1 to stringgrid5.RowCount - 1 do begin
            Ini_Elementos();
            opc1 := Stringgrid4.Cells[stringgrid4.ColCount - 3, a];
            opc1 := trimleft(opc1);
            If opc1 = " then opc1 := '0';
            If (opc1[1]<>'+') and (opc1[1]<>'-' ) then
                opc1 := '+' + opc1;
            En_Elementos( opc1, 'N' , 1);
            opc1 := Stringgrid5.Cells[1, b];
            opc1 := trimleft(opc1);
            If opc1 = " then opc1 := '0';
            If (opc1[1]<>'+') and (opc1[1]<>'-' ) then
                opc1 := '+' + opc1;
            En_Elementos( opc1, 'N' ,2);
            Ordena_ecua(1);
            ordena_ecua(2);
            opc1:="";
            opc2:="";
        end;
    end;
end;

```



```

    For c := 1 to 1000 do begin
        If x1[c,1] <> " then begin
            Val(x1[c,1], v1, temp);
            If v1 <> 0 then
                opc1 := opc1 + x1[c,1];
            end;
        end;
    end;

    For c := 1 to 1000 do begin
        If x1[c,2] <> " then begin
            Val(x1[c,2], v1, temp);
            If v1 <> 0 then
                opc2 := opc2 + x1[c,2];
            end;
        end;
        opc1 := trimleft(opc1);
        opc2 := trimleft(opc2);

        If opc1 = opc2 then begin
            sino := 'S';
            break;
        end;
    end;

    end;

    If sino = 'N' then begin
        Stringgrid5.RowCount := Stringgrid5.RowCount + 1;
        stringgrid5.Cells[1, Stringgrid5.RowCount - 1] :=
stringgrid4.Cells[stringgrid4.ColCount - 3, a];
    end;

    end;

end;

Procedure TForm1.Calcula_M();
Var
    a,b, c:Integer;
    operacion, operacion2,opc1:string;
begin
    For a := 1 to 1000 do begin
        For b := 1 to 100 do begin
            simplex[a,b] := "";
        end;
    end;

    For a := 1 to Stringgrid5.rowcount -1 do begin
        Ini_Elementos();
        opc1 := Stringgrid5.Cells[1, a];
    end;
end;

```

```

opc1 := trimleft(opc1);
If opc1 = " then opc1 := '0';
If (opc1[1]<>'+') and (opc1[1]<>'-' ) then
    opc1 := '+' + opc1;
En_Elementos( opc1, 'N' , 1);
operacion := "";
operacion2 := "";
c := 1;
For b := 1 to 1000 do begin
    If x1[b,1]<>" then begin
        opc1 := x1[b,1];
        opc1 := trimleft(opc1);
        If letras_sino(opc1) then begin
            If (opc1[1] = '+') then begin
                opc1 := '-' + copy(opc1,2,length(opc1))
            end else begin
                opc1 := '+' + copy(opc1,2,length(opc1));
            end;
            simplex[c,a] := opc1;
            c := c + 1;
            operacion := operacion + opc1
        end else begin
            operacion2 := operacion2 + opc1;
        end;
    end;
end;
If operacion2 = " then operacion2 := '+0.00';
stringgrid5.Cells[2,a]:= 'M' + operacion + '<=' + operacion2;
simplex[c,a] := '+s'+inttostr(a);
c := c + 1;
simplex[c,a] := operacion2;
end;

```

end;

Function TForm1.letras_sino(o:string):boolean;

Var

a:Integer;

sino:boolean;

begin

sino :=False;

For a := 1 to length(o) do begin

 If o[a]='x' then

 sino := True;

 end;

letras_sino := sino;

end;

```

Procedure TForm1.Genera_Simplex();
Var
  a,b,c:Integer;
  operacion, opc,ss:String;
begin
  c:= 0;
  stringgrid6.RowCount := stringgrid5.RowCount + 1;
  stringgrid6.ColCount := stringgrid5.RowCount + 2 + ((stringgrid1.RowCount - 1) -
2 )+1;
  Stringgrid6.Cells[1,0] := 'z';
  Stringgrid6.Cells[2,0] := 'm';
  b := 0;
  For a := 1 to (stringgrid1.RowCount - 1) -2 do begin
    Stringgrid6.Cells[2+a,0] := 'x'+ inttostr(a);
    b := b +1;
  end;
  For a := 1 to (stringgrid5.RowCount - 1) do begin
    Stringgrid6.Cells[2 + b + a,0] := 's'+ inttostr(a);
    Stringgrid6.Cells[2 + b + a,b] := '1';
  end;

  Stringgrid6.Cells[Stringgrid6.ColCount - 1,0] := 'b';
  Stringgrid6.Cells[0,1] := 'z';

  For a := 1 to (stringgrid5.RowCount - 1) do begin
    Stringgrid6.Cells[0,1+a] := 's'+ inttostr(a);
  end;

  For b := 1 to stringgrid5.RowCount-1 do begin
    operacion := "";
    opc := "";
    Stringgrid6.Cells[2,b+1]:='+1';
    Stringgrid6.Cells[1,b+1]:='+0';
    ss := 'N';

    For c := 3 to stringgrid6.ColCount - 1 do begin
      Stringgrid6.Cells[c , b+1] := '+0.00';
    end;

    For a := 1 to 1000 do begin
      If simplex[a,b] = " then begin

        break;
      end;

      operacion := operacion + opc;
      opc := simplex[a,b];
    end;
  end;
end;

```

```

        For c := 3 to stringgrid6.ColCount - 1 do begin
            If checaletra( stringgrid6.Cells[ c,0 ], opc ) then begin
                Stringgrid6.Cells[c , b+1] := dame_numero(opc);
            end;
        end;

        end;
        Stringgrid5.Cells[3,b]:= '+m' + operacion+'='+opc;
    end;
    Stringgrid6.Cells[1,1]:='+1';
    Stringgrid6.Cells[2,1]:='-1';
    For a := 3 to stringgrid6.ColCount - 1 do
        Stringgrid6.Cells[a,1]:='+0.00';

end;

Function TForm1.checaletra(letra, cual :String):Boolean;
Var
    l,s:String;
    a:Integer;
begin
    For a := 1 to length(cual) do begin
        If (cual[a]='s') or (cual[a]='x') then begin
            s:= 'S';
        end;
        If s='S' then l := l + cual[a];
    end;
    If l = " then l:='b';
    If letra = l then checaletra := True else checaletra := False;

end;

Function TForm1.dame_numero(q:string):String;
Var
    a:Integer;
    num:string;
begin
    num := q[1];
    For a := 2 to Length(q) do begin
        If (q[a] = 'x') or (q[a] = 's') then
            If a = 2 then
                num := num + '1'
            else
                num := num + copy(q,2,a-2);
        end;
    end;
    If num = q[1] then

```

```

        dame_numero := q
    else
        dame_numero := num;
end;

Procedure TForm1.Calcula_Simplex();
Var
    a, tem:Integer;
    v1:real;
    sino:String;
begin
    sino := 'N';
    For a:= 1 to stringgrid6.RowCount - 1 do begin
        Val( stringgrid6.Cells[stringgrid6.colCount - 1,a], v1, tem );
        If v1 < 0 then
            sino := 'S';
        end;
        If sino = 'S' then
            Simplex_Dual()
        else
            Simplex_Primal(1);
        end;
    end;

Procedure TForm1.Simplex_Dual();
Var
    sino, sv1:String;
    ini,ini2, a, b, Linea, linea2, columna,
    temp, maxlin, suma :Integer;
    v1, v2, v3, v4, sss :Real;
begin
    {calculo del simplexdaul}
    ini := 1;
    iteracion := 0;
    Repeat
        V1 := 0;
        maxlin := stringgrid6.RowCount -1;
        For a := ini to maxlin do begin
            Val( stringgrid6.Cells[stringgrid6.ColCount -1, a ], v2, temp);
            If v1 > v2 then begin
                v1 := v2;
                linea := a;
            end;
        end;
        end;
        V3 := 1000000;
        For a := 1 to Stringgrid6.ColCount - 2 do begin
            Val( (stringgrid6.Cells[a,ini]), v1, temp);
            Val( (stringgrid6.Cells[a,linea]), v2, temp);
            If v2<0 then
                If v3 > abs(v1 / v2) then begin

```

```

        v3 := abs(v1 / v2);
        columna := a;
        sss := v2;

        end;
    end;
    If v3 = 1000000 then stringgrid6.Cells[ stringgrid6.ColCount-1, linea ] := '0';
    If v3 <> 1000000 then begin

        stringgrid6.RowCount := stringgrid6.RowCount + 1;
        ini2 := stringgrid6.RowCount ;

        iteracion := iteracion + 1;
        Stringgrid6.Cells[1,ini2-1] := 'iteración';
        Stringgrid6.Cells[2, ini2-1] := intostr(iteracion);

        For a := ini to linea - 1 do begin
            stringgrid6.RowCount := stringgrid6.RowCount + 1;
            For b := 0 to stringgrid6.ColCount - 1 do begin
                stringgrid6.Cells[b,stringgrid6.RowCount - 1]
:=stringgrid6.Cells[b,a];
                end;
            end;

            linea2 := stringgrid6.RowCount;
            stringgrid6.RowCount := stringgrid6.RowCount + 1;
            val( Stringgrid6.Cells[columna, linea], v1, temp);
            stringgrid6.Cells[0,stringgrid6.RowCount - 1] := stringgrid6.Cells[columna,
0];

            For a := 1 to stringgrid6.ColCount - 1 do begin
                Val( Stringgrid6.Cells[a,linea], v2, temp);
                v3 := v2/v1;
                Str( v3:10:10, sv1);
                sv1:= trimleft(sv1);
                If (sv1[1] = '+') or (sv1[1] = '-') then begin
                    stringgrid6.Cells[a,stringgrid6.RowCount - 1] := sv1;
                end else begin
                    stringgrid6.Cells[a,stringgrid6.RowCount - 1] := '+' + sv1;
                end;
            end;

            For a := linea + 1 to maxlin do begin
                stringgrid6.RowCount := stringgrid6.RowCount + 1;
                For b := 0 to stringgrid6.ColCount - 1 do begin
                    stringgrid6.Cells[b,stringgrid6.RowCount - 1]
:=stringgrid6.Cells[b,a];
                    end;
                end;

            For a:= ini2 to stringgrid6.RowCount -1 do begin

```

```

Val(stringgrid6.Cells[columna,a], v1, temp );
If a <> linea2 then
If v1 < 0 then begin
Val( stringgrid6.Cells[columna,a], v2, temp);
v2 := v2 * -1;
For b := 2 to Stringgrid6.ColCount - 1 do begin
Val( stringgrid6.Cells[b,a], v3, temp );
Val( stringgrid6.Cells[b,linea2], v1, temp );
v4 := v1*v2+v3;
Str( v4:10:10, sv1);
sv1:= trimleft(sv1);
If (sv1[1] = '+') or (sv1[1] = '-') then begin
stringgrid6.Cells[b,a] := sv1;
end else begin
stringgrid6.Cells[b,a] := '+' + sv1;
end;

end;
end;
end;
end;

sino := 'N';
For a:= ini2 to stringgrid6.RowCount - 1 do begin
Val( stringgrid6.Cells[stringgrid6.colCount - 1,a], v1, temp );
If v1 < 0 then
sino := 'S';
end;

ini := ini2;

Until (sino = 'N') or (v3 = 1000000);

sino := 'N';

For a := 2 to stringgrid6.ColCount - 2 do begin
Val(stringgrid6.Cells[a,ini2], v1, temp);
If v1 < 0 then sino := 'S';
end;

If sino = 'S' then Simplex_Primal(ini2);

end;

Procedure TForm1.Simplex_Primal(ini:integer);
Var
sino, sv1:String;
ini2, a, b, Linea, linea2, columna,

```

```

temp, maxlin, suma :Integer;
v1, v2, v3, v4 :Real;
begin
{calculo del simplex primal}
repeat

V1 := 0;
maxlin := stringgrid6.RowCount - 1;
For a := 2 to Stringgrid6.ColCount - 2 do begin
Val( stringgrid6.Cells[ a, ini ], v2, temp);
If v1 > v2 then begin
v1 := v2;
columna := a;
end;
end;
V3 := 1000000;
For a := ini to maxlin do begin
Val( (stringgrid6.Cells[ Stringgrid6.colcount-1, a ]), v1, temp);
Val( (stringgrid6.Cells[columna,a]), v2, temp);
If v2>0 then
If v3 > (v1 / v2) then begin
v3 := v1 / v2;
Linea := a;
end;
end;

stringgrid6.RowCount := stringgrid6.RowCount + 1;

ini2 := stringgrid6.RowCount ;

iteracion := iteracion + 1;
Stringgrid6.Cells[1,ini2-1] := 'iteración';
Stringgrid6.Cells[2, ini2-1] := inttostr(iteracion);

For a := ini to linea - 1 do begin
stringgrid6.RowCount := stringgrid6.RowCount + 1;
For b := 0 to stringgrid6.ColCount - 1 do begin
stringgrid6.Cells[b,stringgrid6.RowCount - 1]
:=stringgrid6.Cells[b,a];
end;
end;

linea2 := stringgrid6.RowCount;
stringgrid6.RowCount := stringgrid6.RowCount + 1;
val( Stringgrid6.Cells[columna, linea], v1, temp);
stringgrid6.Cells[0,stringgrid6.RowCount - 1] := stringgrid6.Cells[columna,
0];

For a := 1 to stringgrid6.ColCount - 1 do begin

```



```

Val( Stringgrid6.Cells[a,linea], v2, temp);
v3 := v2/v1;
Str( v3:10:2, sv1);
sv1:= trimleft(sv1);
If (sv1[1] = '+') or (sv1[1] = '-') then begin
    stringgrid6.Cells[a,stringgrid6.RowCount -1] := sv1;
end else begin
    stringgrid6.Cells[a,stringgrid6.RowCount -1] := '+' + sv1;
end;
end;

For a := linea + 1 to maxlin do begin
    stringgrid6.RowCount := stringgrid6.RowCount + 1;
    For b := 0 to stringgrid6.ColCount - 1 do begin
        stringgrid6.Cells[b,stringgrid6.RowCount -1]
:=stringgrid6.Cells[b,a];
    end;
end;

For a:= ini2 to stringgrid6.RowCount -1 do begin
    Val(stringgrid6.Cells[columna,a], v1, temp );
    If a<>linea2 then
        If v1 <> 0 then begin
            Val( stringgrid6.Cells[columna,a], v2, temp);
            v2 := v2 * -1;
            For b := 2 to Stringgrid6.ColCount - 1 do begin
                Val( stringgrid6.Cells[b,a], v3, temp );
                Val( stringgrid6.Cells[b,linea2], v1, temp );
                v4 := v1*v2+v3;
                Str( v4:10:2, sv1);
                sv1:= trimleft(sv1);
                If (sv1[1] = '+') or (sv1[1] = '-') then begin
                    stringgrid6.Cells[b,a] := sv1;
                end else begin
                    stringgrid6.Cells[b,a] := '+' + sv1;
                end;
            end;
        end;
    end;

    sino := 'S';

    For a := 2 to stringgrid6.ColCount - 2 do begin
        Val(stringgrid6.Cells[a,ini2], v1, temp);
        If v1 < 0 then sino :='N';
    end;
    ini := ini2;

```

```

Until sino = 'S'

end;

procedure TForm1.TabSheet5Show(Sender: TObject);
begin
    genrea_Simplex();
end;

Procedure TForm1.genrea_Simplex();
Var
    a, b:Integer;
    texto, texto2:String;
begin

    stringgrid7.RowCount := stringgrid4.RowCount;
    stringgrid8.RowCount := stringgrid4.RowCount;
    stringgrid7.ColWidths[0] := 10;
    stringgrid8.ColWidths[0] := 10;
    stringgrid7.ColWidths[1] := 200;
    stringgrid7.ColWidths[2] := 200;
    stringgrid8.ColWidths[1] := 200;
    stringgrid8.ColWidths[2] := 200;
    stringgrid8.ColWidths[3] := 300;
    encuatra_Valores();
    For a:= 1 to stringgrid4.RowCount - 1 do begin
        texto := "";
        For b := 1 to stringgrid1.RowCount - 1 do begin
            texto := texto + stringgrid4.Cells[b,a] + ' ';
        end;
        stringgrid8.Cells[1,a] := texto;
        texto2 := Stringgrid4.Cells[stringgrid4.ColCount-1,a];
        for b:= 1 to length(texto2) do begin
            If texto2[b]='=' then begin
                texto2 := copy(texto2, b+1, length(texto2));
                Break;
            end;
        end;
        Stringgrid8.Cells[2,a] := texto2;
        Stringgrid8.Cells[3,a] := sustituye(texto2);
    end;

    For a:= 1 to stringgrid4.RowCount - 1 do begin
        stringgrid7.Cells[1,a] := Stringgrid4.Cells[stringgrid4.colcount - 3, a];
        stringgrid7.Cells[2,a] := sustituye( Stringgrid4.Cells[stringgrid4.colcount - 3,
a]);
    end;
end;

```

```

    Calcula_magnitud();

end;

Procedure TForm1.encuatra_Valores();
Var
    a, b, c :Integer;
    tem, ss:String;
begin
    {encuatra los valores de las x}
    b := 1;
    For a:= stringgrid6.RowCount - 1 downto 1 do begin
        tem := Stringgrid6.Cells[0,a];
        tem := trimleft(tem);
        if tem = "" then break;
        If letras_sino(tem) then begin
            ss := 'N';
            For c := 1 to 8 do begin
                If valores_x[1,c] = tem then begin
                    valores_x[1,c] := tem;
                    valores_x[2,c] := stringgrid6.Cells[stringgrid6.colcount-1,a];
                    ss:= 'S';
                end;
            end;
            If ss = 'N' then begin
                valores_x[1,b] := tem;
                valores_x[2,b] := stringgrid6.Cells[stringgrid6.colcount-1,a];
                b := b + 1;
            end;
        end;
    end;
end;

Function TForm1.Sustitulle_x(lin:Integer):string;
Var
    a, gg:Integer;
    tem:String;
    cc:string;
    v1,v3, v2, v4:real;

Function busca_Valor(ecu:String):String;
Var
    aa, tt:Integer;
    signo : String;
    elem, res, cv1:String;
begin
    res := "";
    cv1 := "";

```

```

signo := "";
for aa := 1 to length(ecu) do begin
  If ecu[aa]='x' then begin
    if aa = 1 then begin
      elem := copy(ecu,aa,length(ecu)-aa+1)
    end else begin
      elem := copy(ecu,aa ,length(ecu)-aa+1);
      signo :=ecu[aa-1];
    end;
    break;
  end else begin
    cv1 := cv1 + ecu[aa];
  end;
end;
For aa := 1 to 8 do begin

  If valores_x[1,aa] = elem then begin
    res := valores_x[2,aa];
  end;
end;

If res = " then res := '0.00';
val( res, v1, tt);

If signo= '-' then v1 := v1 * -1;

if (cv1='-') or (cv1='+') then cv1 :='';

If cv1 = " then begin
  v2:=1;
end else begin
  val(cv1,v2,tt);
end;

v3 := v1*v2;
str( v3:10:2, elem);
If v3<0 then elem := trimleft(elem) else elem := '+'trimleft(elem);
busca_valor := elem;

end;

begin
cc := "";
v4 := 0;
for a := 1 to 1000 do begin
  tem := x1[a,lin];

```

```

        If tem = " then break;
        IF letras_sino(tem) then begin
            tem := busca_Valor(tem);
            x1[a,lin] := tem;
        end;
        cc := cc + tem;
        val( x1[a,lin], v2, gg);
        v4 := v4 + v2;
    end;
    str(v4:12:2,tem);
    tem := trimleft(tem);
    Sustitulle_x := cc + '=' + tem;

end;

Function TForm1.sustituye(ecua:String):String;
begin

    { calcula la ecuacion }
    ecua := trimleft(ecua);
    If ecua <> " then begin

        If not (ecua[1] = '-') then
            If not (ecua[1] = '+') then
                ecua := '+' + ecua;
        Ini_Elementos();
        En_Elementos( ecua,'N',1);
        sustituye := Sustitulle_x(1);
    end;

end;

end;

Procedure TForm1.Calcula_magnitud();
begin

    {For a := 1 to stringgrid}

end;

procedure TForm1.SpeedButton4Click(Sender: TObject);
Var
    a,b:Integer;
    Archivo:Textfile;
    t:String;

begin

```

```

If savedialog1.Execute then begin
  assignfile(archivo,savedialog1.filename+'.csv');
  Rewrite(archivo);

  For b := 1 to stringgrid5.RowCount - 1 do begin
    t := "";
    for a := 1 to stringgrid5.ColCount - 1 do begin
      t := t + ""+stringgrid5.cells[a,b]+""+',';
    end;
    Writeln(Archivo,t);
  end;
  Closefile(archivo);
end;

end;

procedure TForm1.SpeedButton5Click(Sender: TObject);
Var
  a,b:Integer;
  Archivo:Textfile;
  t:String;

begin
  If savedialog1.Execute then begin
    assignfile(archivo,savedialog1.filename+'.csv');
    Rewrite(archivo);

    For b := 0 to stringgrid6.RowCount - 1 do begin
      t := "";
      for a := 0 to stringgrid6.ColCount - 1 do begin
        t := t + ""+stringgrid6.cells[a,b]+""+',';
      end;
      Writeln(Archivo,t);
    end;
    Closefile(archivo);
  end;

end;

procedure TForm1.SpeedButton6Click(Sender: TObject);
Var
  a:Integer;

begin
  Progressbar3.Max := stringgrid2.RowCount - 1;
  stringgrid9.ColCount := stringgrid2.RowCount + 2;

  For a := 1 to stringgrid9.ColCount - 2 do begin
    Stringgrid9.Cells[a+1,0] := Stringgrid2.Cells[1,a];
  end;
end;

```

```

end;

stringgrid9.Cells[stringgrid9.ColCount - 1, 0] := 'M.A.';

Stringgrid10.ColCount := Stringgrid1.RowCount;
stringgrid10.RowCount := Stringgrid2.RowCount+1;

For a := 1 to Stringgrid1.RowCount - 1 do begin
    stringgrid10.Cells[a,0] := stringgrid1.Cells[1,a];
end;

For a := 1 to Stringgrid2.RowCount - 1 do begin
    stringgrid10.Cells[0, a] := stringgrid2.Cells[1,a];
end;

combobox2.Items.Clear;
For a:= 1 to stringgrid2.RowCount - 1 do begin

    Combobox2.Items.Add(stringgrid2.Cells[1,a]);

end;

stringgrid4.ColWidths[0]:= 10;
stringgrid4.ColCount := stringgrid1.RowCount + 3;
stringgrid4.ColWidths[stringgrid1.RowCount ]:= 500;
stringgrid4.ColWidths[stringgrid1.RowCount + 1 ]:= 500;
stringgrid4.ColWidths[stringgrid1.RowCount + 2 ]:= 500;
combobox2.ItemIndex := 0;
combobox2.Text := combobox2.Items[0];

For a:= 1 to stringgrid2.RowCount - 1 do begin

    Progressbar3.Position := a;
    progressbar3.Repaint;

    cuan_participante:= a - 1;

    Inicia_tablas();
    calcula_orden();
    ordena_valores();
    rellenagrid();

    Genera_Diferencias();

    progressbar2.Position := 1;
    Progressbar2.Repaint;
    busca_ecua();
    Progressbar2.Position := 2;

```

```

Progressbar2.Repaint;

Calcula_M();
Progressbar2.Position := 3;
Progressbar2.Repaint;
Reiniciagrid();
Genera_Simplex();
Progressbar2.Position := 4;
Progressbar2.Repaint;

Calcula_Simplex();
Progressbar2.Position := 5;
Progressbar2.Repaint;

genrea_Simplex();
Progressbar2.Position := 6;
Progressbar2.Repaint;

captura_Valores(a);
Recopila_tabla(a+1);
Progressbar2.Position := 7;
Progressbar2.Repaint;

end;
Ototales();
end;

Procedure TForm1.Recopila_tabla(caul:Integer);
Var
    a,b,t:integer;
Begin
    // Genera
    stringgrid9.RowCount := stringgrid8.RowCount;
    For a := 1 to stringgrid8.RowCount - 1 do begin
        For b:= 1 to stringgrid9.RowCount - 1 do begin
            If (stringgrid8.Cells[1,a]= stringgrid9.Cells[1,b]) or
                (stringgrid9.Cells[1,b]= "")then begin
                stringgrid9.Cells[caul,b] := dame_total(stringgrid8.Cells[3,a]);
                stringgrid9.Cells[1,b] := stringgrid8.Cells[1,a];
                break;
            end;
        end;
    end;

end;

end;

end;

```



```

Procedure TForm1.OTotales();
Var
  a,b,t:integer;
  tot,tem,tem2, tt2 :real;
  d,t1,t2:string;
  orden:Array[1..5,1..2] of string;
Begin

  tem := 0;
  tt2 := 1000000;
  For a := 1 to stringgrid9.RowCount - 1 do begin
    tot := 0;
    For b:= 2 to stringgrid9.ColCount - 2 do begin

      val( stringgrid9.Cells[b, a], tem, t);
      tot := tot + tem;
    end;

    If tt2 > tot then begin
      lp := a;
      tt2 := tot;
    end;

    str(tot:5:2,d);
    stringgrid9.Cells[stringgrid9.ColCount-1,a] := d;
  end;

  tem := 0;
  For a := 1 to stringgrid10.ColCount - 1 do begin
    tot := 0;
    For b:= 1 to stringgrid10.RowCount - 2 do begin
      val( stringgrid10.Cells[ a, b ], tem, t);
      tot := tot + tem;
    end;
    str(tot:5:2,d);
    stringgrid10.Cells[a, stringgrid10.RowCount-1] := d;
    orden[a,1]:= d;
    orden[a,2]:= stringgrid10.Cells[a,0 ];
  end;

  for a := 1 to 5 do begin
    for b:= a + 1 to 5 do begin
      Val( orden[a,1], tem, t);
      Val( orden[b,1], tem2, t);
      If tem < tem2 then begin
        t1 := orden[a,1];
        t2 := orden[a,2];
        orden[a,1] := orden[b,1];

```

```

        orden[a,2] := orden[b,2];
        orden[b,1] := t1;
        orden[b,2] := t2;
    end;
end;
end;
Richedit1.Lines.Clear;
richedit1.Font.Size := 12;
richedit1.Lines.Add('Orden Social');
richedit1.Lines.Add("");
Richedit1.SelLength := 0;
For a:= 1 to 5 do begin
    If orden[a,2] <> " then begin
        If a > 1 then begin
            If orden[a,1] = orden[a-1,1] then begin
                Richedit1.SelAttributes.Color := Rgb(255,0,0);
                Richedit1.SelAttributes.Size := 20;
                Richedit1.SelText := ' I ';
            end else begin
                Richedit1.SelAttributes.Color := Rgb(255,0,0);
                Richedit1.SelAttributes.Size := 20;
                Richedit1.SelText := ' P ';
            end;
            Richedit1.SelLength := 0;

            end;
            Richedit1.SelAttributes.Color := Rgb(0,0,0);
            Richedit1.SelAttributes.Size := 20;
            Richedit1.SelText := orden[a,2];
            Richedit1.SelLength := 0;
        end;
    end;
end;
end;

```

```

function TForm1.dame_total(texto :string ):string;
Var
    a:Integer;
    opc,opc2 :String;
begin
    opc :='N';
    opc := "";
    For a := 1 to length(texto) do begin

        If opc = 'S' then begin
            opc2 := opc2 + texto[a];
        end;
    end;
end;

```

```

        If texto[a]='=' then begin
            opc := 'S';
        end;
    end;

    dame_total := opc2;

end;

Procedure TForm1.carga_valor_final(t:integer;t2,t3:string);
Var
    a:Integer;
begin
    For a := 1 to 10 do begin
        IF Funciones_valor1[a] = t then break;
        If Funciones_valor1[a] = 1000 then begin
            Funciones_valor1[a] := T;
            Funciones_valor2[a,1] := T2;
            Funciones_valor2[a,2] := T3;
            Funciones_valor2[a,3] := T2;
            Break;
        end;
    end;
end;
Procedure TForm1.captura_Valores(pp:Integer);
Var
    a, b :Integer;
begin
    {no es}
    For a:= 1 to 8 do begin
        For b := 1 to 8 do begin
            If valores_x[1,a] <> " then
                If valores_x[1,a] = Funciones_valor2[b,1] then begin
                    Funciones_valor2[b,3]:= valores_x[2,a];
                end;
            end;
        end;
    end;
    For b:= 1 to Stringgrid10.ColCount - 1 do begin
        For a:= 1 to 8 do begin
            If stringgrid10.Cells[b,0 ] = Funciones_valor2[a,2] then begin
                stringgrid10.Cells[b,pp] := Funciones_valor2[a,3];
            end;
        end;
    end;
end;
end;

```

```

procedure TForm1.SpeedButton7Click(Sender: TObject);
Var
  a,b:Integer;
  Archivo:Textfile;
  t:String;

begin
If savedialog1.Execute then begin
  assignfile(archivo,savedialog1.filename+'.csv');
  Rewrite(archivo);

  For b := 0 to stringgrid7.RowCount - 1 do begin
    t := "";
    for a := 0 to stringgrid7.ColCount - 1 do begin
      t := t + ""'+stringgrid7.cells[a,b]+'""+';';
    end;
    Writeln(Archivo,t);
  end;
  Closefile(archivo);
end;

end;

procedure TForm1.SpeedButton8Click(Sender: TObject);
Var
  archivo : Textfile;
  a:Integer;
begin
  If savedialog1.Execute then begin

    Assignfile(archivo, savedialog1.FileName+'.pro');
    Rewrite(archivo);
    Writeln( Archivo, 'Archivo de proyecto' );
    Closefile( Archivo );

    Assignfile(archivo, savedialog1.FileName+' - opciones'+'.opc');
    Rewrite( Archivo );

    For a := 1 to stringgrid1.RowCount - 1 do begin

      Writeln( Archivo, stringgrid1.Cells[1,a] );

    end;
    Closefile(Archivo);

    Assignfile(archivo, savedialog1.FileName+' - Parti'+'.opc');
    Rewrite( Archivo );

```

```

For a := 1 to stringgrid2.RowCount - 1 do begin
    Writeln( Archivo, stringgrid2.Cells[1,a] );

end;
Closefile(Archivo);

Assignfile(arch_datos, savedialog1.FileName+' - datos+'.opc');
Rewrite( arch_datos );
For a := 0 to stringgrid2.RowCount - 1 do begin
    registro.opc1 := ubicado[a,1];
    registro.opc2 := ubicado[a,2];
    registro.opc3 := ubicado[a,3];
    registro.opc4 := ubicado[a,4];
    registro.opc5 := ubicado[a,5];
    Write(arch_datos, registro);
end;
closefile(arch_datos);

end;

end;

procedure TForm1.SpeedButton9Click(Sender: TObject);
Var
    archivo : Textfile;
    a:Integer;
    Linea:string;
begin
    If opendialog1.Execute then begin
        Assignfile(archivo, opendialog1.FileName);
        Reset(archivo);
        readln(Archivo, linea);
        If linea = 'Archivo de proyecto' then begin
            Closefile(archivo);
            Assignfile(archivo, copy( opendialog1.FileName,1,pos(' ',
opendialog1.FileName)-1)+'
                ' - opciones'+'.opc');
            Reset(archivo);
            a := 1;

            Repeat

                Readln(archivo, linea);
                stringgrid1.RowCount := a +1;
                stringgrid1.Cells[1,a] := linea;
                a:= a+1;

            Until eof(archivo);

```

```

edit1.Text := inttostr(a-1);
Closefile(archivo);

Assignfile(archivo, copy( opendialog1.FileName,1,pos('.',
opendialog1.FileName)-1)+
' - Parti'+'.opc');
Reset(archivo);
Reset(archivo);
a := 1;

Repeat

    Readln(archivo, linea);
    stringgrid2.RowCount := a +1;
    stringgrid2.Cells[1,a] := linea;
    a:= a+1;

Until eof(archivo);
edit2.Text := inttostr(a-1);
CloseFile(archivo);

Assignfile(arch_datos, copy( opendialog1.FileName,1,pos('.',
opendialog1.FileName)-1)+
' - datos'+'.opc');
reset( arch_datos );
a := 0;
Repeat
    Read(Arch_datos, registro);
    ubicado[a,1] := registro.opc1 ;
    ubicado[a,2] := registro.opc2 ;
    ubicado[a,3] := registro.opc3 ;
    ubicado[a,4] := registro.opc4 ;
    ubicado[a,5] := registro.opc5 ;
    a:= a+1;

Until eof( arch_datos);
closefile(arch_datos);

end;
end;

end;

procedure TForm1.SpeedButton10Click(Sender: TObject);
Var
    a,b:Integer;
begin

    Updown1.Position := 3;
    stringgrid1.RowCount := 4;

```

```

For a:= 1 to stringgrid1.RowCount - 1 do begin
    Stringgrid1.Cells[1,a] := "";

end;
Updown2.Position := 2;
stringgrid2.RowCount := 3;
For a:= 1 to stringgrid2.RowCount - 1 do begin
    Stringgrid2.Cells[1,a] := "";
end;

For a := 1 to 100 do begin
    For b := 1 to 100 do begin
        ubicado[a,b] := 0;
    end;
end;

end;

procedure TForm1.Reiniciagrid();
Var
    a,b:Integer;
begin

    For a:= 1 to stringgrid6.RowCount - 1 do begin
        For b := 0 to Stringgrid6.ColCount -1 do begin
            Stringgrid6.Cells[b,a] := "";

            end;
        end;

end;

procedure TForm1.StringGrid9DrawCell(Sender: TObject; ACol, ARow: Integer;
    Rect: TRect; State: TGridDrawState);
begin

    If (acol >0 ) and (aRow>0 ) then begin
        If arow = lp then begin

            stringgrid9.Canvas.Brush.Color := Rgb( 62, 249, 128);
            Stringgrid9.Canvas.Rectangle(rect);
            Stringgrid9.Canvas.TextOut(rect.Left + 3 ,rect.Top + 4,
stringgrid9.Cells[acol,arow]);
            end;

        end;

end;

end;

end.

```

REFERENCIAS BIBLIOGRÁFICAS.

Arora, N. y G.M. Allenby (1999). Measuring the Influence of Individual Preference Structures in Group Decision Making. *Journal of Marketing Research* 36:176-487.

Arrow, K. (1963). *Social Choice and Individual Values*. 2nd. Ed. Wiley, New York.

Barberá, S. (1984). Teoría de la elección Social: algunas líneas de desarrollo. *Hacienda Pública Española* 91, 221-243.

Basu, K., P. Pattanaik, K. Suzumuran (1995). Choice, Welfare, and Development. A Festschrift in honour of Amartya K. Sen.

Blair, D. y R. Pollack (1983). Decisiones Racionales Colectivas. *Investigación y Ciencia*

Blackorby, Ch. D. Donaldson y J. Weymark (1984). Social Choice with interpersonal utility comparisons: A diagrammatic Introduction. *International Economic Review*. Vol. 25. Núm 2: 327-356.

Dardanoni, V. (2001). A pedagogical proof of Arrow's Impossibility Theorem. *Social Choice and Welfare* 18:107-112.

D'Aspremont, C. (1985). Axioms for Social Welfare Orderings. En Hurwicz, L., D. Schmeidler y H. Sonnenschein (Eds), *Social Goals and Social organization, Essays in memory of Elisha Pazner*. Cambridge University Press.

D'Aspremont, C. y L. Gevers (1977). Equity and Informational Basis of Collective Choice. *Review of Economic Studies*, 44: 199-209.

D'Aspremont, C. y L. Gevers (2001). Social welfare functionals and interpersonal Comparability. CORE discussion paper 2001/ 40. Université Catholique de Louvain, Bélgica.

French, S. (1988) Decision theory: an introduction to the mathematics of rationality. John

Wiley & Sons.

Guillén S T, G. Mazcorro , H. Sánchez (2002). A method for constructing cardinal value functions on finite sets, *Multiple Criteria Decision Making Winter Conference*, Semmering, Austria.

Guillén S T, (2003) Second order preferences for assigning value difference functions on finite sets, comunicación personal .

Harsanyi, J. (1955). Cardinal Welfare, Individualistic Ethics, and Interpersonal Comparisons of Utility. *Journal of Political Economy* 63:309-321.

Khmelnitskaya, A. y J. Weymark (2000). Social Choice with independent subgroup utility scales. *Social Choice and Welfare* 17:739-748.

Krantz, D. H.; Luce, R. D.; Suppes, P. and Tversky, A. (1971) *Foundations of Measurement*. Vol. 1. Academic Press.

Luce, R. y H. Raiffa (1957). *Games and Decisions*. John Wiley & Sons.

Mas-Colell, A. y H. Sonnenschein (1972). General Possibility Theorems for Group Decisions. *Review of Economic Studies*, 39:185-192.

Mas- Coller, A, M. Whinston y J. Green. (1995). *Microeconomic Theory*. Oxford University Press.

Maskin, E. (1978). A Theorem on Utilitarianism. *Review of Economic Studies*. Vol. 93-96.

Maskin, E. (1995) . Majority Rule, Social Welfare Functions, and Game Forms. In Basu, K., P. Pattanaik, K. Suzumuran (1995). *Choice, Welfare, and Development: A Festschrift in honour of Amartya K. Sen*. Clarendon press, Oxford & N. York. Pp. 100-109.

Massó, J. (1996). La teoría de la elección social: métodos de votación no manipulables. Mimeo, Universidad Autónoma de Barcelona.

Müller, A. (ND). El teorema de la Imposibilidad de Arrow: Presentación diagramática y breves comentarios. Nota de Investigación, proyecto UBACyT TE-07. Universidad Autónoma de Barcelona.

Peris, J.E. y C. Sanchez (2001). Characterization of social choice sets in terms of individuals' maximal sets: The fixed agenda framework. *Social choice and Welfare* 18: 113-127.

Plata-Pérez, L. (1996). Un panorama de resultados y problemas abiertos en la Teoría de la Elección Social. Documento de trabajo. Departamento de Economía, Instituto Tecnológico Autónomo de México.

Plata-Pérez, L. (1999). Partial Comparability and Social Welfare Orderings. Documento de trabajo. Departamento de Economía, Instituto Tecnológico Autónomo de México.

Roberts, K. (1980). Interpersonal comparability and social choice Theory. *Review of Economics Studies* 47:421-439.

Schwartz, T. (2001). From Arrow to cycles, instability, and chaos by untying alternatives. *Social Choice and Welfare* 18: 1-22.

Sen, A. (1974). Choice, Orderings and Morality. En *Choice, Welfare and Measurement*. (1997). Harvard University Press. 74-83

Sen, A. (1976). Rational Fools: A critique of the Behavioural Foundations of Economic Theory. En José Casas Pardo Ed. *Amartya Kumar Sen. Nueva Economía del Bienestar escritos seleccionados*. Universitat de Valencia (1995), 83-102.

Sen, A (1977). On Weights and measures: Informational constraints in Social Welfare Analysis. *Econometrica* 45:1539-1572.

Sen, A. (1979). Interpersonal comparisons of Welfare. En Choice, Welfare and Measurement.(1997). Harvard University Press. 264-281

Sen, A. (1984). Social Choice Theory. En K.Arrow y M. Intriligator, eds., Handbook of Mathematical Economics III. North Holland.

Stevens, S. (1959). Measurement, Psychophysics, and Utility. En Churchman and Ratoosh (eds). Measurement: Definitions and theories. Wiley, NY., pp18-63.

Taylor, L. (1998) La teoría de la elección social y el mundo en que vivimos. Cuadernos de Economía, vol. 17, núm. 29: 235-248

Villar, A. (1988). La lógica de la elección social: Una revisión de los resultados básicos. Investigaciones Económicas (segunda época). Vol. XII, núm. 1: 3-44.

Van der Veen, R. (1981). Meta-rankings and collective optimality. *Social Science Information*, 20, 2: 345-374.

Vincke, Ph., M Gassner, B. Roy (1992). *Multicriteria Decision Aid*. John Wiley & Sons.