

Informe Sistemas Multiagentes.

1. Introducción

El presente proyecto detalla cómo se ha creado un sistema multiagente para la realización de toma de decisiones y realización de consenso en una comunidad. Para ello, se ha utilizado el lenguaje Java de programación con la extensión JADE para la creación de sistemas multiagente.

Se han utilizado 3 métodos de sobreclasificación sobre distintos agentes.

- Electre
- Promethee
- AHP.

El proyecto se encuentra disponible en <https://github.com/alu0100693737/jadeMulticriterio>

2. Información previa.

2.1 Formato Datos de entrada:

a. Datos del problema:

```
# 6 1 //alternativas, atributos 1 si tiene descripcion, 0 datos directos
velMax distDespegue cargaMax precio fiabilidad manejo
1 0 1 0 1 1 //segunda linea max(1) o min(0)
2.0 1.5 2.0 5.5 5 9
2.5 2.7 1.8 6.5 3 5
1.8 2.0 2.1 4.5 7 7
2.2 1.8 2.0 5.0 5 5
```

En la primera línea se indica el número de alternativas existentes y el conjunto de atributos o criterios que se tendrán en cuenta, el último dato muestra si se tiene descripción de cada uno de los criterios; si es así, la siguiente línea detallará esas propiedades.

A continuación se indica con un 0 si se debe minimizar dicho atributo y 1 si debe maximizarse.

Por último y para cada una de las alternativas, se ofrecen los valores para cada uno de los criterios.

b. Importancias Relativas de los agentes:

```
Juan 0.2 0.1 0.1 0.1 0.2 0.3
Pepe 0.2 0.2 0.1 0.2 0.1 0.2
Maria 0.2 0.1 0.2 0.2 0.1 0.2
Luis 0.1 0.2 0.1 0.2 0.1 0.3
```

Para cada uno de los agentes se contempla una importancia relativa para cada criterio. La suma debe ser igual a 1.

2.2 Comportamiento Electre

```
class comportamientoElectre extends OneShotBehaviour {

    @Override
    public void action() {
        // TODO Auto-generated method stub
        calcularMatrizDecisionNormalizada();
        //showSolucionElectre();
        calcularMatrizDecisionPonderada();
        //showSolucionElectre();

        calcularConjuntoConcordanciaDiscordancia();
        //showConjuntosConcordanciaDiscordancia();

        calcularMatrizConcordancia();
        calcularMatrizDiscordancia();

        //showMatricesConcordanciaDiscordancia();

        calcularMatrizDominanciaConcordancia();
        calcularMatrizDominanciaDiscordancia();
        //showMatricesDominancia();

        calcularMatrizDominanciaAgregada();
        //showMatrizDominanciaAgregada();

        calcularPrioridadesFinales();

        ACLMessage msg= new ACLMessage(ACLMessage.INFORM);
        //Enviamos el nombre del agente y las prioridades
        msg.setContent(getName() + "\n" + getImportanciaRelativa().getNombre() + "\n" + getF
        msg.addReceiver(new AID("agenteModerador", AID.ISLOCALNAME));
        send(msg);
    }
}
```

Se calcula:

- La matriz de decisión normalizada
- La matriz de decisión ponderada
- El conjunto de concordancia y discordancia
- La matriz de concordancia y discordancia
- La matriz de dominancia por concordancia y discordancia
- La matriz de dominancia agregada
- Por ultimo transformamos la matriz para ordenarla por prioridades

La información final se envía como mensaje informativo al agente moderador.

NOTA: los valores umbral de concordancia y discordancia son los siguientes:

```
private final float umbralConcordancia = 0.55f;
private final float umbralDiscordancia = 0.71f;
```

2.3 Comportamiento Promethee

```
class comportamientoPromethee extends OneShotBehaviour {  
  
    @Override  
    public void action() {  
        //System.out.println("Comportamiento Promethee");  
        matrizIndicesPreferencia = new ArrayList<ArrayList<Float>>();  
        matrizIndicesPreferencias();  
  
        //System.out.println("Matriz Indices Preferencias");  
        //showMatrizIndicesPreferencia();  
  
        calculosFlujosPositivos();  
        calculosFlujosNegativos();  
  
        calculosFlujoNeto(); //Se calcula el positivo, el negativo y el neto  
  
        ACLMessage msg= new ACLMessage(ACLMessage.INFORM);  
        msg.setContent(getName() + "\n" + getImportanciaRelativa().getNombre() + "\n")  
        msg.addReceiver(new AID("agenteModerador", AID.ISLOCALNAME));  
        send(msg);  
    }  
}
```

En el método Promethee debe calcularse:

- La matriz de índices de preferencia
- Calcular los flujos positivos y negativos
- Calcular el flujo neto que indicará las prioridades finales.

La información final se envía como mensaje informativo al agente moderador.

2.4 AHP (Proceso analítico jerárquico)

```
class comportamientoAHP extends OneShotBehaviour {

    @Override
    public void action() {
        //System.out.println("Comportamiento Proceso Analitico Jerarquico");
        //System.out.println("Se calcula la matriz de comparacion por pares");
        calcularMatrizComparacionPares();
        //showMatrizComparacionPares();
        //System.out.println("Se calcula la matriz normalizada");
        calcularMatrizComparacionesParesNormalizada();
        //showMatrizComparacionParesNormalizada();
        calcularPrioridades();
        //showConjuntoPrioridades();
        calcularMatricesAlternativasSegunPrioridades();
        //showMatricesAlternativasSegunPrioridades();
        calcularMatricesAlternativasSegunPrioridadesNormalizada();
        //showMatricesAlternativasSegunPrioridadesNormalizada();
        calcularConjuntoMatrizPrioridadesAlternativas();
        //showConjuntoMatrizPrioridadAlternativas();
        calcularPrioridadFinal();
        //showPrioridadesFinal();

        ACLMessage msg= new ACLMessage(ACLMessage.INFORM);
        msg.setContent(getName() + "\n" + getImportanciaRelativa().getNombre() + "\n"
        msg.addReceiver(new AID("agenteModerador", AID.ISLOCALNAME));
        send(msg);
    }
}
```

En el método AHP debe:

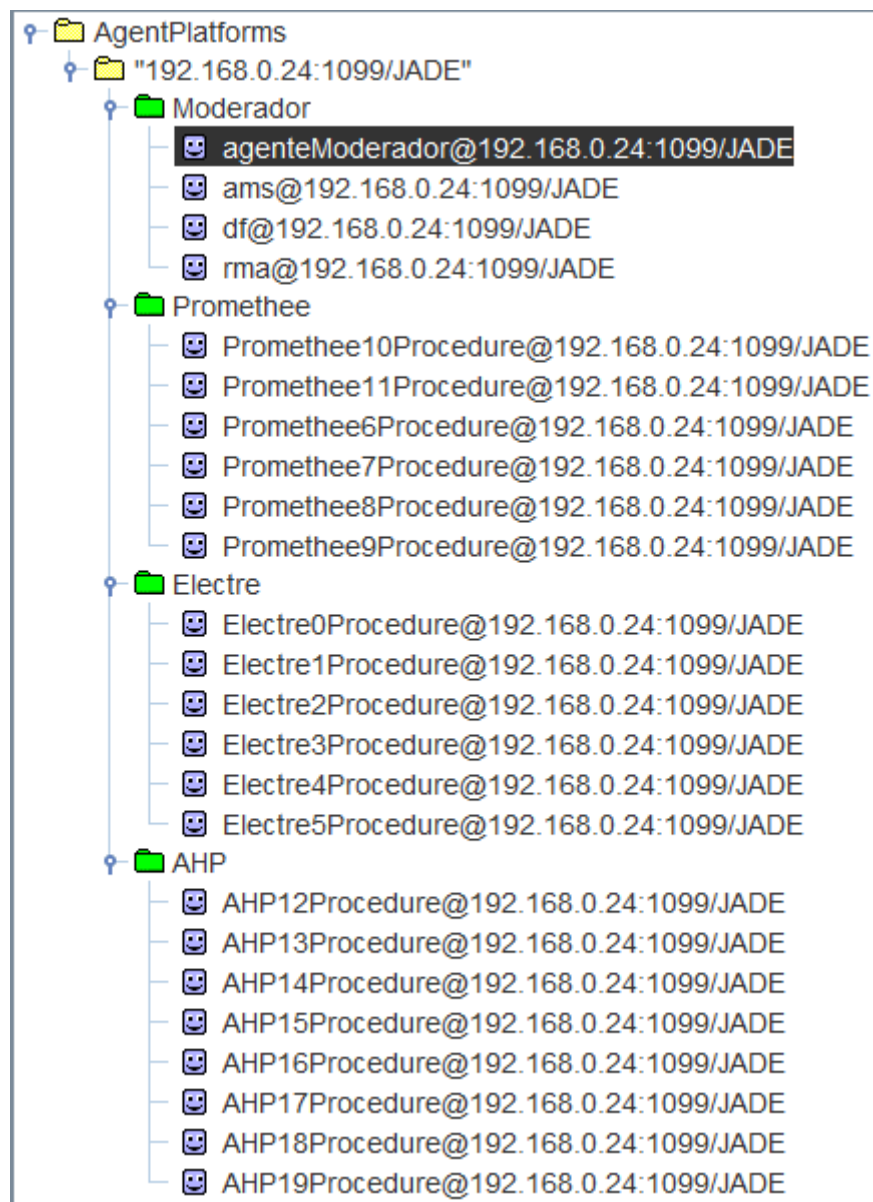
- Calcular la matriz de comparación por pares y normalizarla.
- Calcular el conjunto de prioridades.
- Calcular las matrices de alternativas según la prioridad y normalizarla.
- Calcular el conjunto matriz prioridades alternativa.
- Calcular la prioridad final.

La información final se envía como mensaje informativo al agente moderador.

De esta manera, el proyecto lee un determinado problema y una serie de importancias relativas o prioridades dada por cada uno de los usuarios.

Se crearán de forma proporcional agentes con distintos métodos de sobreclasificación (Electre, Promethee o AHP) y con la importancia relativa individual de un sujeto. Estos agentes, calculan la prioridad final del usuario y se la comunican al moderador.

En la implementación del sistema se ha intentado mantener el mayor orden posible. Cada tipo de agente se mantiene en un Container Jade distinto.



2.5 Moderador

El moderador será el encargado de leer los datos del problema, crear los agentes necesarios y comprobar si existe un consenso entre todos los agentes.

Lectura de ficheros y creación de agentes:

```
lectorFichero = new lectorProblema(FICHERO_PROBLEMA);
lectorFicheroImportancias = new lectorImportanciasRelativas(FICHERO_IMPORTANCIAS_RELATIVAS, getLectorFichero().getM

rt = Runtime.instance();

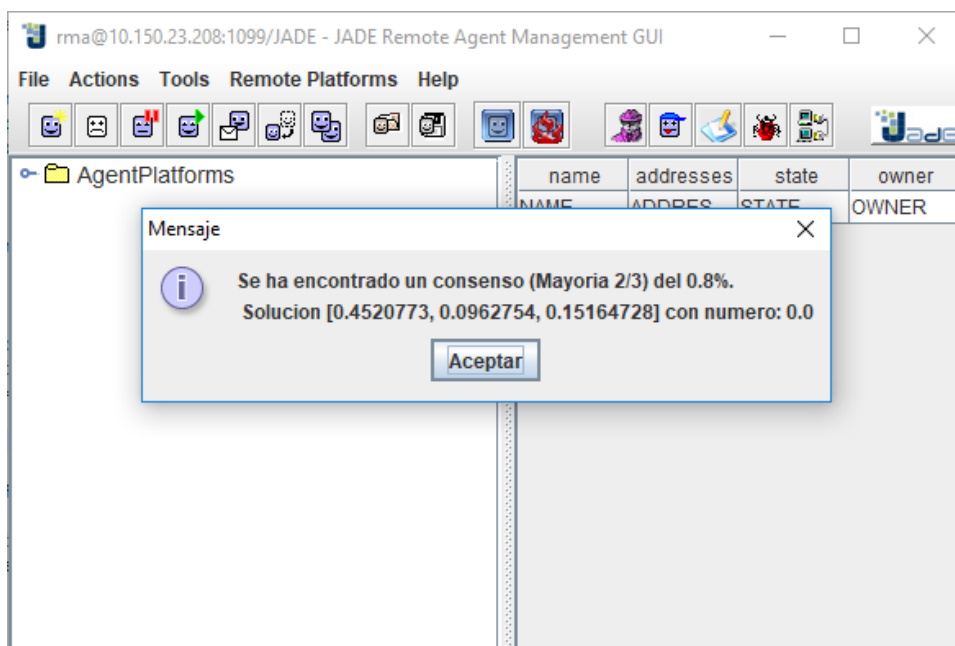
//Propiedades de contenedores, solo nombre
ExtendedProperties propertiesElectred = new ExtendedProperties();
propertiesElectred.setProperty(Profile.CONTAINER_NAME, "Electre");
ExtendedProperties propertiesPromethee = new ExtendedProperties();
propertiesPromethee.setProperty(Profile.CONTAINER_NAME, "Promethee");
ExtendedProperties propertiesAHP = new ExtendedProperties();
propertiesAHP.setProperty(Profile.CONTAINER_NAME, "AHP");

electreContainer = getRunTime().createAgentContainer(new ProfileImpl(propertiesElectred));
PrometheeContainer = getRunTime().createAgentContainer(new ProfileImpl(propertiesPromethee));
ahpContainer = getRunTime().createAgentContainer(new ProfileImpl(propertiesAHP));

primeraVuelta = false;
//System.out.println("Agente moderador, ya se han leído los ficheros, añadimos Electre");

addAgenteTipoElectre(NUMERO_AGENTES/3, 0);
addAgenteTipoPromethee(NUMERO_AGENTES/3, NUMERO_AGENTES/3);
addAgenteTipoAHP(NUMERO_AGENTES - (2 * NUMERO_AGENTES/3 - 1), 2 * NUMERO_AGENTES/3 - 1);
```

A continuación se leerán los resultados dados por cada uno de los agentes y se calculará cuantos agentes están de acuerdo con la solución lograda. Puede comprobarse si el consenso es de tipo 2/3 del conjunto, mayoría absoluta, mayoría 2/3 o absoluta por minoría o no existe consenso.



Si no existiera consenso se aplicará un proceso consistente en:

- Comunicar a los agentes que están en desacuerdo cual es la importancia relativa media de todos los agentes que están de acuerdo.
- Cada agente, modificará sus importancias relativas sin que ello conlleve un cambio en su prioridad entre los atributos. Ejemplo:

Ej: Alternativa usuario num 0, Grupo num 1.

Antes

Nombre: Pepe 0.7 0.2 0.05 0.05

Después

Nombre: Pepe 0.49999997 0.4 0.05 0.05

Esto hace que pueda ser modificada la solución final del agente sin ser alterada su prioridad entre una alternativa u otra. El método utilizado en cada agente para realizar esta operación se llama

```
public void modificarImportanciasRelativas(ArrayList<Float> importanciasGrupo)
```

Hecha las modificaciones de las importancias, se vuelve a calcular las prioridades finales, se comunican al moderador y este vuelve a calcular la alternativa más adecuada según todos los agentes.

NOTA: existe una variable en el método “*modificarImportanciasRelativas*” que indica cuan preciso debe ser el cambio de importancias para que se parezca a la decisión del grupo sin modificar las prioridades. Ese valor **precisión**, si es muy bajo, hará que las importancias se parezcan más, si es alto, la modificación será más estricta. Se puede ver un efecto de la utilización de esta variable en el punto 5.

3. Ejemplo de ejecución Completo.

3 alternativas, 4 criterios, 20 agentes.

Resultados de prioridades de los agentes:

```
Agente Electre0Procedure Juan con los valores: [0.5, 0.16666667, 0.33333334]
Agente Promethee6Procedure Hugo con los valores: [0.20000002, 0.5, -0.70000005]
Agente Promethee7Procedure Samantha con los valores: [0.3, -0.40000004, 0.099999994]
Agente Electre2Procedure Javi con los valores: [0.33333334, 0.5, 0.16666667]
Agente Promethee8Procedure Juana con los valores: [0.60000001, -0.6, -5.9604645E-8]
Agente Electre1Procedure Maria con los valores: [0.33333334, 0.5, 0.16666667]
Agente Promethee10Procedure Santiago con los valores: [0.60000001, -0.6, -5.9604645E-8]
Agente Electre4Procedure Pepe con los valores: [0.33333334, 0.5, 0.16666667]
Agente Electre3Procedure Ivan con los valores: [0.33333334, 0.5, 0.16666667]
Agente Electre5Procedure Sara con los valores: [0.33333334, 0.5, 0.16666667]
Agente AHP13Procedure Pedro con los valores: [0.52752745, 0.17098518, 0.3014873]
Agente Promethee11Procedure Ulises con los valores: [0.29999998, 0.0, -0.29999998]
Agente AHP16Procedure Alfredo con los valores: [0.25680792, 0.5942132, 0.14897878]
Agente Promethee9Procedure Elena con los valores: [0.25, -0.29999995, 0.049999952]
Agente AHP17Procedure Javiera con los valores: [0.29212463, 0.4996575, 0.20821778]
Agente AHP12Procedure Vanesa con los valores: [0.45455495, 0.2628041, 0.282641]
Agente AHP19Procedure Lucia con los valores: [0.44084838, 0.30270997, 0.25644162]
Agente AHP15Procedure Yadira con los valores: [0.52752745, 0.17098518, 0.3014873]
Agente AHP14Procedure Airam con los valores: [0.2856622, 0.5627651, 0.15157263]
Agente AHP18Procedure Susana con los valores: [0.29212463, 0.4996575, 0.20821778]
```

Decisión final calculada por el moderador:

```
Decision final
0,37 0,22 0,11
```

Elementos que están de acuerdo con la solución:

```
[0.5, 0.16666667, 0.33333334]El primer elemento escogido localmente y globalmente es el mismo en 0 [0.37469223, 0.21652225, 0.10878553]
[0.3, -0.40000004, 0.099999994]El primer elemento escogido localmente y globalmente es el mismo en 2 [0.37469223, 0.21652225, 0.10878553]
[0.60000001, -0.6, -5.9604645E-8]El primer elemento escogido localmente y globalmente es el mismo en 4 [0.37469223, 0.21652225, 0.10878553]
[0.60000001, -0.6, -5.9604645E-8]El primer elemento escogido localmente y globalmente es el mismo en 6 [0.37469223, 0.21652225, 0.10878553]
[0.52752745, 0.17098518, 0.3014873]El primer elemento escogido localmente y globalmente es el mismo en 10 [0.37469223, 0.21652225, 0.10878553]
[0.29999998, 0.0, -0.29999998]El primer elemento escogido localmente y globalmente es el mismo en 11 [0.37469223, 0.21652225, 0.10878553]
[0.25, -0.29999995, 0.049999952]El primer elemento escogido localmente y globalmente es el mismo en 13 [0.37469223, 0.21652225, 0.10878553]
[0.45455495, 0.2628041, 0.282641]El primer elemento escogido localmente y globalmente es el mismo en 15 [0.37469223, 0.21652225, 0.10878553]
[0.44084838, 0.30270997, 0.25644162]El primer elemento escogido localmente y globalmente es el mismo en 16 [0.37469223, 0.21652225, 0.10878553]
[0.52752745, 0.17098518, 0.3014873]El primer elemento escogido localmente y globalmente es el mismo en 17 [0.37469223, 0.21652225, 0.10878553]
Valor 10
```

10 elementos están de acuerdo con la solución, 50%. No es suficiente, se realiza la ronda de consenso comunicando a cada agente en desacuerdo la decisión de grupo.

Cada agente en desacuerdo intenta cambiar sus importancias relativas.

Ej:

```
Antes Nombre Pepe [0.7, 0.2, 0.05, 0.05]
Despues Nombre Pepe [0.5, 0.26666668, 0.116666675, 0.116666675]
```


Con las nuevas importancias relativas, se vuelve a calcular las prioridades individuales del agente siguiendo el método de sobreclasificación elegido (Electre, Promethee o AHP).

La información se envía al moderador y se vuelve a calcular si existe consenso o no.

Prioridades finales

Agente Electre0Procedure Juan con los valores: [0.5, 0.16666667, 0.33333334]
 Agente Electre1Procedure Maria con los valores: [0.33333334, 0.5, 0.16666667]
 Agente Electre2Procedure Javi con los valores: [0.33333334, 0.5, 0.16666667]
 Agente Promethee6Procedure Hugo con los valores: [0.26666665, 0.23333332, -0.5]
 Agente Electre3Procedure Ivan con los valores: [0.33333334, 0.5, 0.16666667]
 Agente Electre4Procedure Pepe con los valores: [0.4, 0.4, 0.2]
 Agente Electre5Procedure Sara con los valores: [0.33333334, 0.5, 0.16666667]
 Agente Promethee7Procedure Samantha con los valores: [0.3, -0.40000004, 0.099999994]
 Agente Promethee8Procedure Juana con los valores: [0.60000001, -0.6, -5.9604645E-8]
 Agente Promethee9Procedure Elena con los valores: [0.25, -0.29999995, 0.049999952]
 Agente Promethee10Procedure Santiago con los valores: [0.60000001, -0.6, -5.9604645E-8]
 Agente Promethee11Procedure Ulises con los valores: [0.29999998, 0.0, -0.29999998]
 Agente AHP13Procedure Pedro con los valores: [0.52752745, 0.17098518, 0.3014873]
 Agente AHP14Procedure Airam con los valores: [0.2856622, 0.5627651, 0.15157263]
 Agente AHP15Procedure Yadira con los valores: [0.52752745, 0.17098518, 0.3014873]
 Agente AHP12Procedure Vanesa con los valores: [0.45455495, 0.2628041, 0.282641]
 Agente AHP17Procedure Javiera con los valores: [0.29212463, 0.4996575, 0.20821778]
 Agente AHP19Procedure Lucia con los valores: [0.44084838, 0.30270997, 0.25644162]
 Agente AHP16Procedure Alfredo con los valores: [0.2694404, 0.5458401, 0.18471946]
 Agente AHP18Procedure Susana con los valores: [0.29212463, 0.4996575, 0.20821778]
 Decisión final
 0,38 0,20 0,12

Como puede observarse se han modificado ligeramente los valores de Hugo y Pepe.

[0.5, 0.16666667, 0.33333334]El primer elemento escogido localmente y globalmente es el mismo en 0 [0.38199052, 0.19577023, 0.122239254]
 [0.26666665, 0.23333332, -0.5]El primer elemento escogido localmente y globalmente es el mismo en 3 [0.38199052, 0.19577023, 0.122239254]
 [0.4, 0.4, 0.2]El primer elemento escogido localmente y globalmente es el mismo en 5 [0.38199052, 0.19577023, 0.122239254]
 [0.3, -0.40000004, 0.099999994]El primer elemento escogido localmente y globalmente es el mismo en 7 [0.38199052, 0.19577023, 0.122239254]
 [0.60000001, -0.6, -5.9604645E-8]El primer elemento escogido localmente y globalmente es el mismo en 8 [0.38199052, 0.19577023, 0.122239254]
 [0.25, -0.29999995, 0.049999952]El primer elemento escogido localmente y globalmente es el mismo en 9 [0.38199052, 0.19577023, 0.122239254]
 [0.60000001, -0.6, -5.9604645E-8]El primer elemento escogido localmente y globalmente es el mismo en 10 [0.38199052, 0.19577023, 0.122239254]
 [0.29999998, 0.0, -0.29999998]El primer elemento escogido localmente y globalmente es el mismo en 11 [0.38199052, 0.19577023, 0.122239254]
 [0.52752745, 0.17098518, 0.3014873]El primer elemento escogido localmente y globalmente es el mismo en 12 [0.38199052, 0.19577023, 0.122239254]
 [0.45455495, 0.2628041, 0.282641]El primer elemento escogido localmente y globalmente es el mismo en 15 [0.38199052, 0.19577023, 0.122239254]
 [0.44084838, 0.30270997, 0.25644162]El primer elemento escogido localmente y globalmente es el mismo en 17 [0.38199052, 0.19577023, 0.122239254]
 Valor 12

Mensaje



Se ha encontrado un consenso (Mayoría Absoluta) del 0.6%.

Solucion [0.38199052, 0.19577023, 0.122239254] con numero: 0.0

Aceptar

4. Pequeño Estudio

Se ha querido realizar un pequeño estudio sobre un determinado problema para:

- Analizar que método de sobreclasificación ofrece mejores soluciones.
- Qué diferencias existen si se usa solo 1 de ellos o los 3 de manera proporcional.
- Modificar el valor precisión del método *modificarImportanciasRelativas* para conocer qué elementos en desacuerdo modifica y cuáles no.

Aunque algunas ejecuciones, tras la primera vuelta ya ofrecen solución aceptable, se ha utilizado la segunda vuelta para ver el comportamiento que presentan.

	Solución primera vuelta. (Alternativas)	Solución %	Solución segunda vuelta. (Consenso)	Solución %
Precisión consenso del 0.3				
Método Electre 100%	Decisión final 0,44 0,33 0,24	65%	Decisión final 0,44 0,33 0,24	65%
Método Promethee 100%	Decisión final 0,31 0,04 -0,34	<u>60%</u>	Decisión final 0,31 0,04 -0,34	<u>60%</u>
Metodo AHP 100%	Decisión final 0,36 0,40 0,23	<u>55%</u>	Decisión final 0,36 0,40 0,23	<u>55%</u>
Electre 50% Promethee 50%	Decisión final 0,39 0,15 -0,03	60%	Decisión final 0,39 0,15 -0,03	60%
Electre 50% AHP 50%	Decisión final 0,40 0,37 0,23	45%	Decisión final 0,43 0,32 0,24	75%
Promethee 50% AHP 50%	Decisión final 0,33 0,26 -0,08	45%	Decisión final 0,33 0,26 -0,08	30%
Electre 33% Promethee 33% AHP 33%	Decisión final 0,37 0,22 0,11	50%	Decisión final 0,37 0,22 0,11	50%
Precisión consenso del 0.2				
Método Electre 100%	Decisión final 0,44 0,33 0,24	65%	Decisión final 0,45 0,31 0,24	80%
Método Promethee 100%	Decisión final 0,31 0,04 -0,34	<u>60%</u>	Decisión final 0,32 -0,01 -0,31	70%
Metodo AHP 100%	Decisión final 0,36 0,40 0,23	<u>55%</u>	Decisión final 0,35 0,42 0,24	55%
Electre 50% Promethee 50%	Decisión final 0,39 0,15 -0,03	60%	Decisión final 0,39 0,12 -0,02	70%
Electre 50% AHP 50%	Decisión final 0,40 0,37 0,23	45%	Decisión final 0,41 0,36 0,23	55%
Promethee 50% AHP 50%	Decisión final 0,33 0,26 -0,08	45%	Decisión final 0,33 0,23 -0,06	55%
Electre 33% Promethee 33% AHP 33%	Decisión final 0,37 0,22 0,11	50%	Decisión final 0,38 0,20 0,12	60%
Precisión consenso del 0.1				

Método Electre 100%	Decisión final 0,44 0,33 0,24	65%	Decisión final 0,47 0,28 0,25	100%
Método Promethee 100%	Decisión final 0,31 0,04 -0,34	60%	Decisión final 0,35 -0,08 -0,27	70%
Metodo AHP 100%	Decisión final 0,36 0,40 0,23	55%	Decisión final 0,34 0,42 0,24	55%
Electre 50% Promethee 50%	Decisión final 0,39 0,15 -0,03	60%	Decisión final 0,42 0,08 0,00	90%
Electre 50% AHP 50%	Decisión final 0,40 0,37 0,23	45%	Decisión final 0,44 0,31 0,25	75%
Promethee 50% AHP 50%	Decisión final 0,33 0,26 -0,08	45%	Decisión final 0,37 0,16 -0,02	55%
Electre 33% Promethee 33% AHP 33%	Decisión final 0,37 0,22 0,11	50%	Decisión final 0,41 0,15 0,14	80%
Precisión consenso del 0.05				
Método Electre 100%	Decisión final 0,44 0,33 0,24	65%	Decisión final 0,47 0,28 0,25	100%
Método Promethee 100%	Decisión final 0,31 0,04 -0,34	60%	Decisión final 0,35 -0,12 -0,23	75%
Metodo AHP 100%	Decisión final 0,36 0,40 0,23	55%	Decisión final 0,34 0,43 0,23	55%
Electre 50% Promethee 50%	Decisión final 0,39 0,15 -0,03	60%	Decisión final 0,42 0,06 0,01	95%
Electre 50% AHP 50%	Decisión final 0,40 0,37 0,23	45%	Decisión final 0,44 0,31 0,25	75%
Promethee 50% AHP 50%	Decisión final 0,33 0,26 -0,08	45%	Decisión final 0,37 0,13 0,00	55%
Electre 33% Promethee 33% AHP 33%	Decisión final 0,37 0,22 0,11	50%	Decisión final 0,41 0,15 0,14	80%

5. Problemas encontrados.

Se ha desarrollado todo el proyecto utilizando Java con la extensión JADE. No se ha utilizado la GUI que provee JADE por lo que ha resultado más difícil la implementación pero se ha logrado un proyecto con mayor calidad. La creación de los agentes de forma automática resultó en un inicio bastante complicado.

Se han encontrado problemas en la comunicación entre agentes. Cuando todos los agentes quieren comunicarse con el agente moderador hay que considerar quien realiza la comunicación ya que no necesariamente la comunicación se realiza por orden de creación de agentes. Esto ha hecho que deba tenerse en cuenta el nombre del agente y el nombre personal del sujeto.

El proyecto se encuentra disponible en <https://github.com/alu0100693737/jadeMulticriterio>