TRABAJO PRÁCTICO 4
INFORME LÓGICA DIFUSA

INTRODUCCIÓN A LA INTELIGENCIA ARTIFICIAL

REALIZADO POR

Bini Valentina
Zimmermann Sebastián

Universidad Nacional de Rosario
Facultad de Ciencias Exactas, Ingeniería y Agrimensura

19/06/2020

1. Enunciado

1.1. Introducción

Un equipo docente desea realizar un sistema que agilice la evaluación de los parciales de alumnos de una materia. La idea es que le docente solo tenga que dar un valor entre 0 y 10 de cada criterio de evaluación y luego, a partir de eso el sistema recomiende una condición para el parcial.

Caso de estudio: Exámenes parciales de la materia Programación 2.

En una primera etapa dados los siguientes criterios, la **Claridad (C)** y la **Modularización (M)**, y se busca analizar la **Calidad de Código (CC)** del trabajo realizado.

En la segunda etapa, dada la **Calidad de Código** obtenida del primer paso y los siguientes criterios de evaluación **Cumplimiento de Tarea (CT)** y **Eficiencia (E)**, se busca recomendar la **Condición Parcial (CP)** del alumne.

El equipo docente decidió los siguientes criterios.

1.2. Reglas

1. Si la modularización es *buena* y el código es *claro*, la CC es *alta*.
2. Si la modularización es *mala* y el código es *confuso*, la CC es *baja*.
3. Si la modularización es *mala* y el código es *claro*, la CC es *media*.
4. Si la modularización es *buena* y el código es *confuso*, la CC es *media*.

5. Si la tarea esta *incumplida*, la condición parcial es *libre*.
6. Si la tarea está *cumplida*, la calidad del código *media* o *alta* y el código es *eficiente* la condición parcial es *promovido*.
7. Si la tarea está *cumplida*, la calidad del código *baja* y el código es *eficiente* la condición parcial es *regular*.
8. Si la tarea está *cumplida*, la calidad del código *alta* y el código es *ineficiente* la condición parcial es *regular*.
9. Si la tarea está *cumplida*, la calidad del código *baja* o *media* y el código es *ineficiente* la condición parcial es *libre*.

1.3. Rangos

- Rango de M [0,10]:
  - Mala: Función trapezoidal que de 0 a 3 vale 1, de 7 a 10 vale 0.
  - Buena: Función trapezoidal que de 0 a 3 vale 0 y de 7 a 10 vale 1.
- Rango de C [0,10]:
  - Confuso: Función trapezoidal que de 0 a 3 vale 1, de 7 a 10 vale 0.
  - Claro: Función trapezoidal que de 0 a 3 vale 0 y de 7 a 10 vale 1.
- Rango de CT [0,10]:
  - Incumplida: Función trapezoidal que de 0 a 4 vale 1, de 6 a 10 vale 0.
  - Cumplida: Función trapezoidal que de 0 a 4 vale 0 y de 6 a 10 vale 1.
- Rango de CC [0,10]:
  - Baja: Función trapezoidal que de 0 a 3 vale 1, de 5 a 10 vale 0.
  - Media: Función trapezoidal que de 0 a 2 vale 0, de 5 a 6 vale 1 y de 9 a 10 vale 0.
  - Alta: Función trapezoidal que de 0 a 5 vale 0 y de 8 a 10 vale 1.
- Rango de E [0,10]:
  - Ineficiente: Función trapezoidal que de 0 a 3 vale 1, de 8 a 10 vale 0.
  - Eficiente: Función trapezoidal que de 0 a 3 vale 0 y de 8 a 10 vale 1.
- Rango de CP [0,10]:
  - Libre: Función trapezoidal que de 0 a 4 vale 1, de 6 a 10 vale 0.
  - Regular: Función trapezoidal que de 0 a 4 vale 0, de 6 a 7 vale 1 y de 9 a 10 vale 0.
  - Promovido: Función trapezoidal que de 0 a 6 vale 0 y de 8 a 10 vale 1.

1.4. Consignas

a) Distinguir las variables lingüísticas de entrada y salida para cada etapa.
b) Analizar la calidad de código (CC) para un caso con una modularización de 4 y una claridad de 8.
OBS: defuzzificar con el punto medio del máximo.
c) Tomando el valor *crisp* de la calidad de código (CC) del item b, y un valor de cumplimiento de tarea (CT) de 8 y una Eficiencia (E) de 9, obtener la condición parcial (CP) y la nota defuzzificada del examen. Defuzzificar con el método que considere más apropiado y justificar.
d) Indique qué reglas se dispararon en los items anteriores.
e) Realizar el item c) y d) con una Eficiencia (E) de 4.

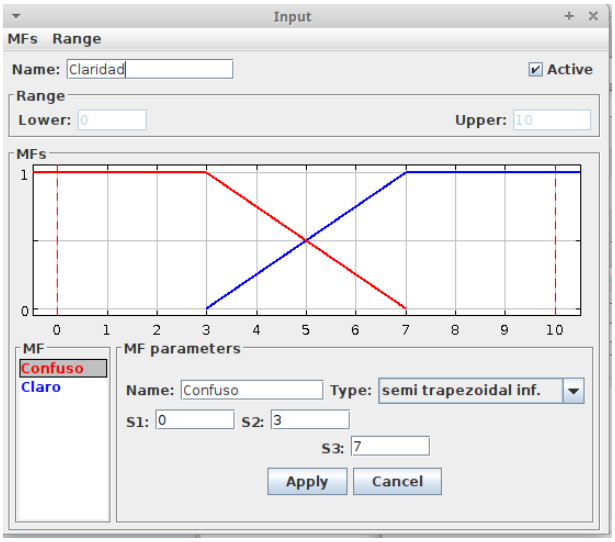
2. Resolución

2.a. Variables Lingüísticas

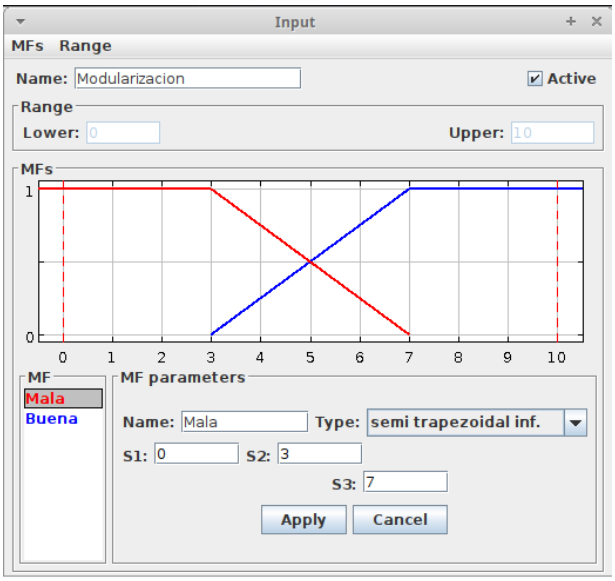
2.a.I. Primera Etapa

Variables de entrada:

- Claridad

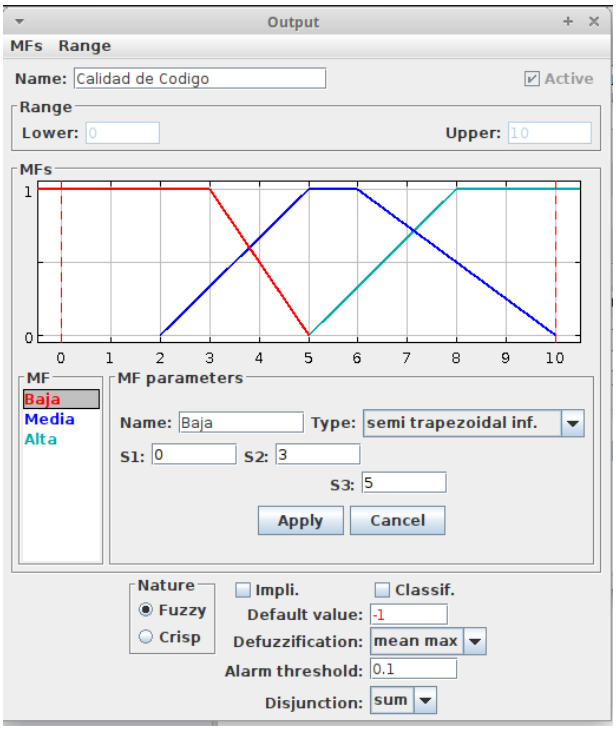


- Modularización



Variables de salida:

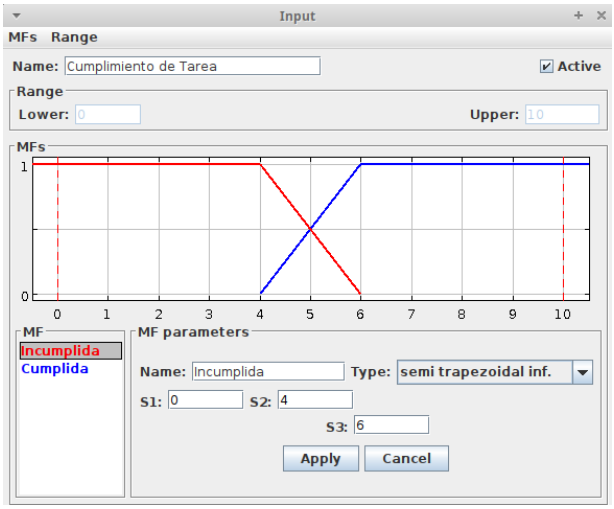
- Calidad de Código



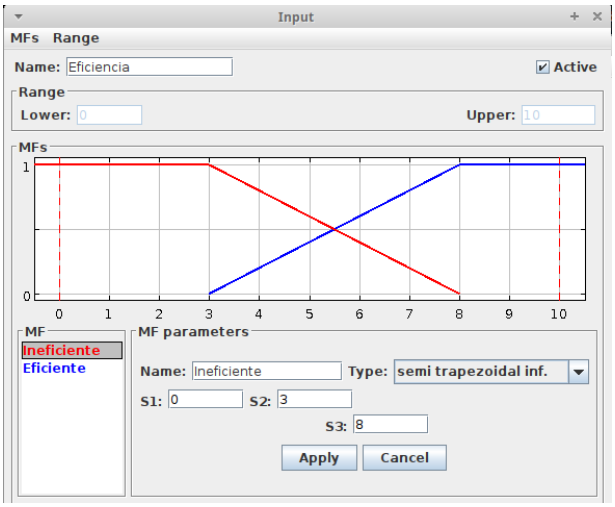
2.a.II. Segunda Etapa

Variables de entrada:

- Calidad de Código (resultado de la primera etapa)
- Cumplimiento de Tarea

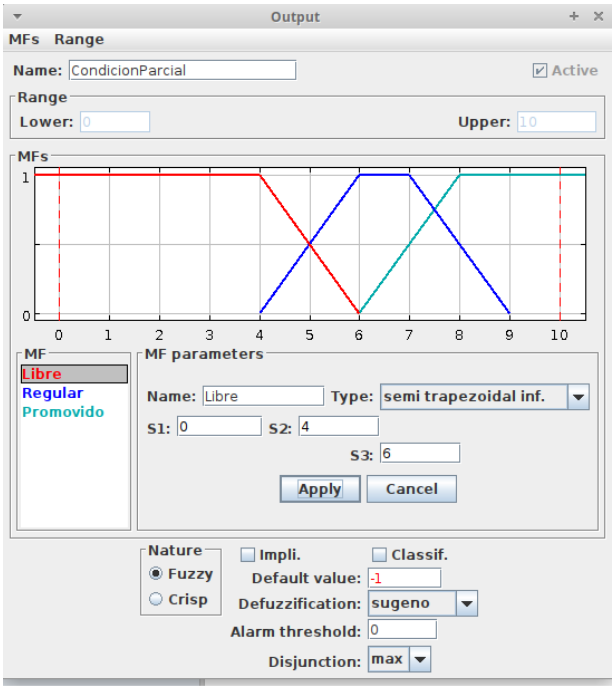


- Eficiencia



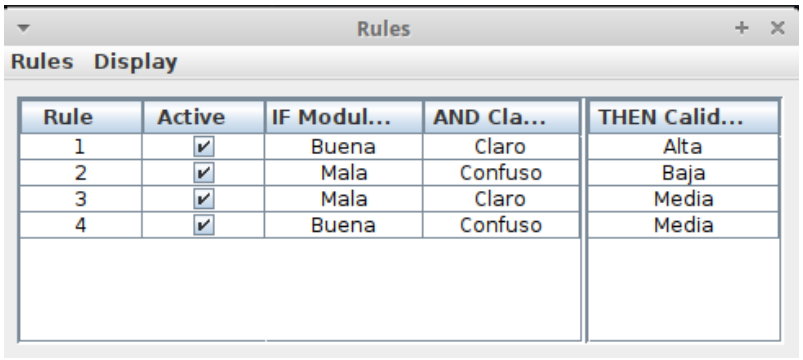
Variables de salida:

- Condición Parcial



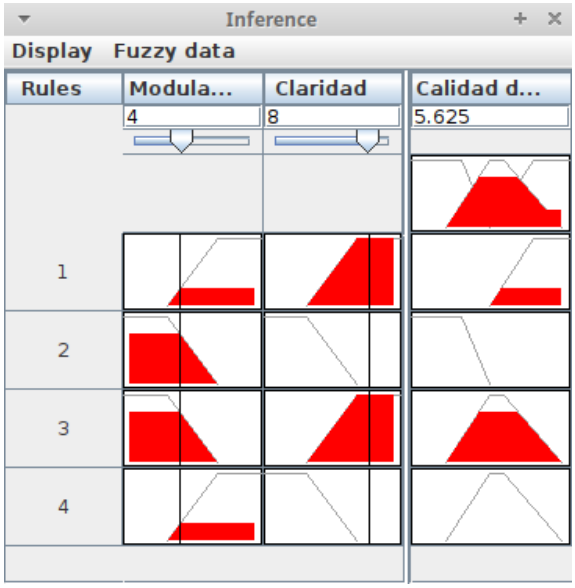
2.b. Análisis de la Calidad de Código

Luego de analizar las reglas del enunciado, hicimos un pasaje a FisPro que resultó en lo siguiente:



Reglas de la primera etapa

A continuación realizamos la inferencia con los valores indicados en el item b.



Inferencia de la primera etapa

Los valores de pertenencia resultantes fueron aproximadamente:

- Modularización:
  - Mala: 0.8
  - Buena: 0.2
- Claridad:
  - Claro: 1
  - Confuso: 0

Con estos valores las reglas que se dispararon fueron:

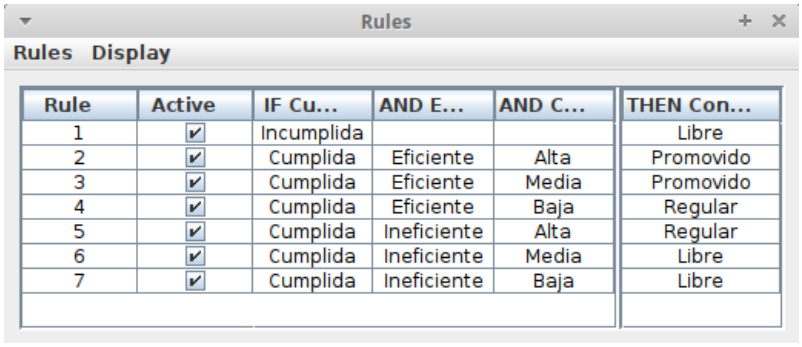
1. $min(0.2,\ 1) = 0.2$
2. $min(0.8,\ 0) = 0$
3. $min(0.8,\ 1) = 0.8$
4. $min(0.2,\ 0) = 0$

Defuzzificando con el promedio de los máximos de FisPro el resultado fue: **5.625**.

2.c. Análisis de la Condición Parcial

Tomando las reglas del enunciado, tuvimos que realizar un cambio respecto a las reglas número 6 y 9. Particionamos a cada una en dos sub-reglas, que representan las opciones del *or* en la variable difusa *Calidad de Código*. Esto dio por resultado las siguientes reglas:

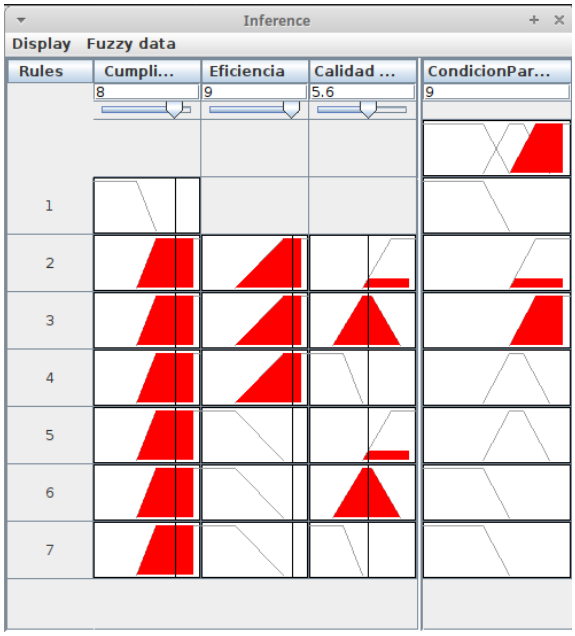
Observación: Los números de reglas fueron modificados por FisPro. De aquí en adelante utilizaremos estas reglas como referencia.



Rule	Active	IF Cu...	AND E...	AND C...	THEN Con...
1	✔	Incumplida			Libre
2	✔	Cumplida	Eficiente	Alta	Promovido
3	✔	Cumplida	Eficiente	Media	Promovido
4	✔	Cumplida	Eficiente	Baja	Regular
5	✔	Cumplida	Ineficiente	Alta	Regular
6	✔	Cumplida	Ineficiente	Media	Libre
7	✔	Cumplida	Ineficiente	Baja	Libre

Reglas de la segunda etapa

Luego de esto, realizamos la inferencia con los valores indicados del item c.



Inferencia de la segunda etapa, item c

Los valores de pertenencia resultantes fueron aproximadamente:

- Cumplimiento de Tarea:
  - Incumplida: 0
  - Cumplida: 1
- Eficiencia:
  - Ineficiente: 0
  - Eficiente: 1
- Calidad de Código:
  - Baja: 0
  - Media: 1
  - Alta: 0.1

Con estos valores las reglas que se dispararon fueron:

1. 0
2. $min(1,\ 1,\ 0.1) = 0.1$
3. $min(1,\ 1,\ 1) = 1$
4. $min(1,\ 1,\ 0) = 0$
5. $min(1,\ 0,\ 0.1) = 0$
6. $min(1,\ 0,\ 1) = 0$
7. $min(1,\ 0,\ 0) = 0$

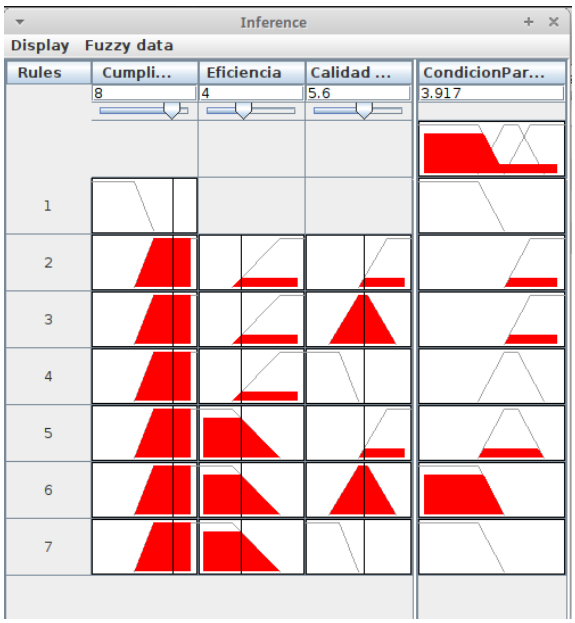
Para este caso, nos parece que el promedio de los máximos representa un resultado coherente con lo que se espera recomendar para esas notas. Por lo tanto, utilizamos este método para defuzzificar, dando como resultado un valor crisp de **9**.

2.d. Disparado de reglas

El disparado de reglas fue resuelto en cada item, así como lo resolveremos en el próximo.

2.e. Analisis de Condición Parcial con Eficiencia 4

Para esta sección las reglas son las mismas que el item C, así que solo nos restaba realizar la inferencia.



Inferencia de la segunda etapa, item e

Los valores de pertenencia resultantes fueron aproximadamente:

- Cumplimiento de Tarea:
  - Incumplida: 0
  - Cumplida: 1
- Eficiencia:
  - Ineficiente: 0.9
  - Eficiente: 0.1
- Calidad de Código:
  - Baja: 0
  - Media: 1
  - Alta: 0.1

Con estos valores las reglas que se dispararon fueron:

1. 0
2. $min(1,\ 0.1,\ 0.1) = 0.1$
3. $min(1,\ 0.1,\ 1) = 0.1$
4. $min(1,\ 0.1,\ 0) = 0$
5. $min(1,\ 0.9,\ 0.1) = 0.1$
6. $min(1,\ 0.9,\ 1) = 0.9$
7. $min(1,\ 0.9,\ 0) = 0$

Para esta situación, nos parece que el promedio de los máximos no refleja un resultado coherente con lo que pretendíamos alcanzar, ya que si bien los valores de cada nota no son tan altos, el resultado refleja mucho peso en la condición libre, que no es consistente con los valores buscados.

Por esto decidimos defuzzificar con **mayor de los máximos**, dando entonces con los resultados obtenidos, el valor crisp **4.25**.

3. Conclusiones

3.1. Introducción al problema

La evaluación de resultados de parciales es una tarea que a nosotres, como docentes, siempre nos es dificultosa. Resulta importante reflejar el aprendizaje y el avance de conocimientos del estudiante, lo cual no siempre resulta tan sencillo ya que en muchos momentos nos vemos obligades a expresar dicho aprendizaje mediante un número.

Asumiendo que dichos exámenes realmente representan dicho aprendizaje, ¿cómo transformamos este examen en una condición final? ¿Este resultado resulta siempre representativo? ¿Resulta necesario encontrar un límite numérico que represente la diferencia entre una condición y otra?

El año pasado, ambos dos trabajamos como auxiliares de segunda de la cátedra Programación 2. Durante este tiempo, tuvimos que corregir múltiples parciales y luego parcialmente decidir una condición final para cada alumne. Cuando en clases dimos la idea de conjuntos difusos, nos pareció que el concepto de “alumne libre”, “alumne regular” o “alumne promovide” no siempre es algo tan fácil de determinar como un promedio de sus parciales.

Con esto en mente, decidimos implementar un sistema de lógica difusa para determinar condiciones finales de alumnes de programación 2 al final de la materia.

3.2. Diseño del sistema

Nuestra primera aproximación al sistema tenía como objetivo, dados dos parciales y sus respectivas correcciones, llegar a una condición final de la materia. Si bien éramos conscientes de que el alcance del trabajo no sería algo sumamente complejo, nos encontramos en la necesidad de analizar y defuzzificar cada parcial por separado, para luego trabajar en la condición.

Esto resultaba sumamente complejo, dado que necesitaríamos 3 etapas y en un principio generamos alrededor de 17 reglas. Por esta razón, decidimos reducir el trabajo al análisis de condiciones de un parcial.

Una vez que hicimos que el sistema solamente abarque parciales, la resolución resultó mucho más sencilla y concluimos en un número de reglas más acorde a lo esperado para el trabajo.

3.3. Pasaje a FisPro

El pasaje del problema a FisPro resultó bastante sencillo, sin embargo tuvimos que realizar algunos pequeños cambios:

- Tuvimos que realizar un archivo .fis para cada etapa por separado.
- Para representar los sistemas con un *or*, tuvimos que separarles en reglas distintas.
- Las opciones para defuzzificar no eran muchas, así que cuando ninguna era consistente con el resultado esperado tuvimos que analizar los gráficos a “ojo”. Esto también resultó necesario para analizar los valores con los que se dispararon las reglas.

3.4. Discusión de resultados

Respecto a la idea inicial de lo que pretendíamos generar como sistema, el resultado obtenido resuelve un problema mucho menor, y al vernos en la necesidad de defuzzificar numéricamente, si bien resulta mucho más preciso que hacer un promedio, termina resultando en un valor numérico.

Con la idea de que el sistema fuera aplicable a múltiples cátedras, sería necesario considerar los siguientes factores:

- En nuestro caso y materia, sería necesario reformular las reglas, ya que en algunos casos el resultado no era el esperado inicialmente.
- Si se espera que la defuzzificación genere una nota numérica, sacar 10 en todos los valores y sacar 9 da el mismo resultado de la defuzzificación. Una posible solución sería generar funciones más triangulares que trapezoidales, haciendo que la defuzzificación sea más extrema en cada valor.
- En caso de realizarse en otra cátedra, los criterios deberían modificarse según decisiones de la cátedra o particularidades de la materia. Esto sería importante para mejorar el sistema y llegar a mejores resultados.
- Una mejora posible sería incoporar la categorías intermedias, como por ejemplo la categoría *desaprobado* para diferenciarla de la categoría *libre*. En el item e, nuestra defuzzificación le otorgaba el carácter *libre* al resultado del examen. Esto permitiría representar mejor los valores intermedios generados entre etapa y etapa.