

ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA INFORMÁTICA
Grado en Ingeniería Informática

FuzzyTeacher, tutor para el Control Difuso

FuzzyTeacher, tutor for fuzzy control

Realizado por

Alejandro Manuel Espada Aurióles

Tutorizado por

Eva Millán Valleperas

Eduardo Guzmán De los Riscos

Departamento

Lenguajes y Ciencias de la Computación

UNIVERSIDAD DE MÁLAGA

MÁLAGA, Junio 2017

Fecha defensa:

El Secretario del Tribunal

Resumen

FuzzyTeacher es una aplicación web donde un usuario puede aprender los conceptos básicos sobre la lógica difusa. Para ello debe registrarse con su nombre, apellidos, email, usuario y contraseña para poder acceder a toda la información.

La aplicación web muestra información básica sobre la lógica difusa, avanzando poco a poco con más información y ejemplos que favorecen la comprensión del usuario. Podemos diferenciar 6 temas, en los que se empieza una introducción en la que se explica los conjuntos difusos para continuar con los hechos difusos y la decodificación. Finalmente acabamos con una serie de problemas y dos tests que nos permite evaluar tanto los conocimientos teóricos como prácticos.

La aplicación web proporciona una serie de ejercicios simples dentro del temario donde el usuario puede poner a prueba su conocimiento. Además muestra si el resultado es válido o no y dispone de una serie de ayudas por si el usuario se ve atascado en algún ejercicio/problema. Finalmente los test permiten medir el grado de conocimiento alcanzado por el usuario. Para tener dos grados de dificultad, contamos con un test en el que solo podemos sumar puntos y otro en el que una respuesta errónea resta la parte proporcional en la calificación final.

Palabras clave

Lógica difusa, conjuntos difusos, hechos difusos, aplicación web, temario, problemas, test

Summary

FuzzyTeacher is a web-based application where a user can learn fuzzy logic (introductory level). To this end, the user must first introduce some information in his/her account: name, email address, user and password. After the registration process the user will be able to access to this application web with information, tests and exercises about fuzzy control.

The web-based application shows the basic information about fuzzy logic, progressing step by step with advanced information and examples to guide the learning process. We can find six topics, which cover all stages of fuzzy control: fuzzy facts, rules and decoding. It also includes problems and two tests, to measure both theoretical and practical knowledge.

The application web has also exercises (grouped by topic), where the user can assess his/her knowledge. Furthermore, it shows whether the result is right or not, and the user can ask for help if he/she needs it. Finally, the user can take two tests to know if he/she has increased his/her knowledge about fuzzy logic. There are two levels: *easy* (where correct answers score positively and wrong answers are not penalized) and *difficult* (where a wrong answer is penalized by subtracting points from the total grade).

Keywords

Fuzzy logic, fuzzy set, fuzzy facts, application web, blocks, exercises, test

Índice

1. Introducción	9
1.1 FuzzyTeacher.....	9
1.2 Motivación de FuzzyTeacher.....	9
1.3 Objetivos	10
1.4 Fases del trabajo.....	10
1.5 Estructura de la memoria	11
1.6 Aplicación webs relacionadas	11
2. Lógica difusa	15
2.1. Conjuntos difusos.....	15
2.2. Hechos difusos.....	19
2.3. Decodificación	21
2.4. Sistemas de tipo TS	23
3. Especificaciones técnicas.....	25
3.1. Propósito	25
3.2. Ámbito del sistema.....	25
3.3. Perspectiva del producto.....	26
3.4. Funciones del producto	26
3.5. Características de los usuarios	26
3.6. Requisitos futuros.....	27
3.7. Requisitos funcionales	27
3.8. Requisitos no funcionales	28
3.9. Casos de uso	28
4. Diseño de la aplicación.....	31
5. Diseño de la interfaz.....	33
5.1. Login	33
5.2. Añadir cuenta	34
5.3. Home.....	35
5.4. Botones de avance y retroceso	35
5.5. Apuntes	36
5.6. Panel de información.....	37
5.7. Temario	38
5.8. Ejercicios.....	40
5.9. Problemas	41

5.10 Tests	44
6. Tests.....	47
7. Conclusiones y ampliación del proyecto.....	49
8. Referencias bibliográficas.....	51
A Resolución de problemas y tests	53
PROBLEMA 1	53
PROBLEMA 2	55
PROBLEMA 3	56
PROBLEMA 4	57
PROBLEMA 5	58
PROBLEMA 6	59
PROBLEMA 7	60
PROBLEMA 8	62
TEST 1	64
TEST 2	66
B Documento de análisis de requisitos.....	70

1. Introducción

En la introducción vamos a ver en que consiste la aplicación web (FuzzyTeacher), la motivación y los objetivos para realizarla y se nombran algunas aplicaciones web relacionadas con la lógica difusa.

1.1 FuzzyTeacher

FuzzyTeacher es una aplicación web pensada para que los usuarios incrementen y pongan a prueba sus conocimientos sobre la lógica difusa. Para ellos dispone de un temario donde se explican los conceptos básicos para la comprensión de dicha lógica y la realización de problemas. Esto permite adquirir los conocimientos necesarios para la identificación y resolución de estos problemas en la vida cotidiana.

En lo referente a los ejercicios, el usuario dispone de un corrector que indica si la solución es correcta o no, lo cual hace que el usuario sepa si ha adquirido los conocimientos correctamente.

Es una aplicación web gratuita pero para poder tener acceso hay que registrarse, por lo que es posible guardar una relación entre el usuario y los temas leídos o ejercicios realizados, que serán mostrados en el panel de información.

1.2 Motivación de FuzzyTeacher

La motivación para realizar este proyecto es hacer más dinámica la enseñanza de la lógica difusa, donde obviamente hay que presentar un temario para que el usuario pueda adquirir nuevos conocimientos o recordar conocimientos, y una parte más interactiva donde el usuario tiene una serie de retos (ejercicios y test) a superar. Estos retos no solo se plantean sino que el mismo usuario puede comprobar si lo está realizando de forma correcta o no.

De este modo se agiliza el proceso de aprendizaje, además de suponer un incremento en la motivación al ver que se superan dichos retos.

1.3 Objetivos

El objetivo de este proyecto es la realización de una aplicación web lo suficientemente completa en lo que se refiere al temario y en los problemas y tests para poder ser usada tanto en la docencia como de carácter individual por parte de cualquier usuario que quiera hacer uso de ella.

1.4 Fases del trabajo

Para la realización de este proyecto se ha planificado una serie de fases a seguir y el número de horas dedicadas a cada una de las fases:

1. Análisis del proyecto

Esta fase consiste en el estudio de los requisitos, recopilación de documentos teóricos de la lógica difusa y la realización de un documento del análisis de requisitos.

Número de horas: 40

2. Desarrollo de la aplicación

En esta fase se procederá a la separación de tareas para la realización de la aplicación web de forma progresiva, priorizando la funcionalidad de la misma para acabar con la realización de la interfaz gráfica.

Número de horas: 186

3. Testeo y refactorización

Fase final del desarrollo informático en la que se prueba que tanto la funcionalidad como la visualización de las ventanas son correcta y se realiza una refactorización de las partes con el objetivo de mejorar la eficiencia.

Número de horas: 40

4. Documentación

Para finalizar el proyecto se va a realizar una memoria que recoja la información sobre el proyecto con más detalle, para facilitar la comprensión y explicación sobre la aplicación web.

Número de horas: 30

1.5 Estructura de la memoria

La memoria consta de siete bloques. En ellos se habla de las diferentes partes que conciernen al proyecto, un bloque con las referencias bibliográficas y tres apéndices donde podemos encontrar el documento de análisis de requisitos y un solucionario de los problemas.

Los siete bloques diferenciables son:

Bloque 1: Un breve introducción de la memoria, su estructura y objetivos.

Bloque 2: En este bloque se procede a explicar el pilar central del proyecto que es la lógica difusa donde se expondrá el temario de dicho proyecto.

Bloque 3: Bloque de las especificaciones técnicas del proyecto en la que se indicará qué herramientas se han seleccionado para dicho trabajo, exponiendo los motivos para ello hemos usado las recomendaciones de: *Especificación de Requisitos según el estándar de IEEE 830 (s.d.) (2008)*. Asimismo, se mencionarán los datos que se han tenido en cuenta para la toma de las decisiones.

Bloque 4: En este bloque se procede al desarrollo de la aplicación, tanto el *back-end* como el *front-end*, explicaremos las herramientas que se han utilizado y se presentaran los diagramas de clases y los flujos de uso de la aplicación.

Bloque 5: Bloque en el que se hablará sobre la interfaz gráfica del programa. En este bloque comentaremos qué herramientas se han utilizado para la realización de dicha interfaz. También se mostrarán los flujos a partir de cada vista y se enseñarán las interfaces por medio de capturas de pantalla.

Bloque 6: En esta fase se hablará de las decisiones tomadas para la refactorización del código, se comentaran las pruebas realizadas a la aplicación.

Bloque 7: Bloque final en que se exponen las conclusiones obtenidas al finalizar este proyecto y se mencionarán posibles líneas de trabajo futuro y mejoras para la aplicación.

1.6 Aplicación webs relacionadas

Podemos encontrar Aplicación webs de lógica difusa orientadas a la creación de problemas y otras como complementos de otros softwares (en este caso Matlab):

Aplicaciones web para la creación de problemas:

- Fuzzylite (Windows, Mac, Ubuntu, Java/Android)
- Lógica Difusa Práctica (Android)

Aplicaciones web para Matlab

- Fuzzy Logic Designer app
- Neuro-Fuzzy Designer app

Fuzzylite (de Juan Rada-Vilela)

Fuzzylite es una aplicación de escritorio donde se pueden crear problemas de lógica difusa. Esta creación de problemas consiste en definir un universo y una serie de reglas. La aplicación aplicara las reglas adecuadas en función del valor de entrada introducido.

Consta de una representación gráfica de las variables introducidas, facilitando la comprensión de los valores resultantes.

Una carencia de esta aplicación es que una variable puede ser tanto de entrada como de salida si forma parte de un problema mayor (donde se concadenan resultados). Para resolver este problema, tenemos que duplicar dicha variable, por una parte será el valor resultante y por la otra, valor de entrada.

Fuzzylite complementa a nuestra aplicación ya que carece de la parte teórica.

Lógica Difusa Práctica

Esta aplicación para Android trata de mostrar el uso práctico del concepto de Lógica Difusa utilizado en diferentes áreas del conocimiento como son: Inteligencia Artificial, Teoría de Control, entre otros. Consta de un enlace a un artículo de lógica difusa.

La aplicación consiste en el uso del acelerómetro del teléfono para determinar si la posición del mismo es la denominada como "correcta" teniendo 3 posibles valores: Hacia Adelante, No Cambies, Hacia Atrás. El usuario debe girar el teléfono hasta obtener la posición ideal.

Pero a diferencia de nuestra aplicación web, esta aplicación no nos enseña los conceptos básicos de la lógica difusa ni pone a prueba nuestros conocimientos.

Fuzzy Logic Designer app

La aplicación Fuzzy Logic Designer permite diseñar y probar sistemas de inferencia difusos para modelar comportamientos complejos del sistema.

Con esta aplicación, puedes:

- Diseñar sistemas de inferencia difusa de Mamdani y Sugeno.
- Agregar o eliminar variables de entrada y salida.
- Especificar las funciones de pertenencia de las variables.
- Definir reglas difusas if-then.
- Seleccionar las funciones de inferencia difusa para:
 - Operaciones AND
 - Operaciones OR

- Implicación
- Defuzzificación
- Ajustar los valores de entrada y ver los diagramas de inferencia.
- Ver la salida para sistemas de inferencia difusa
- Exportar sistemas de inferencia difusa al espacio de trabajo MATLAB®

Neuro-Fuzzy Designer app

La aplicación Neuro-Fuzzy Designer permite diseñar, entrenar y probar sistemas adaptativos de inferencia neuro-fuzzy (ANFIS) utilizando datos de entrenamiento de entrada / salida.

Con esta aplicación, puedes:

- Ajustar los parámetros de la función de pertenencia de los sistemas de inferencia difuso de tipo Sugeno.
- Generar automáticamente una estructura de sistema de inferencia inicial basada en sus datos de entrenamiento.
- Modificar la estructura del sistema de inferencia.
- Previene el sobreajuste de los datos de entrenamiento utilizando datos de comprobación adicionales.
- Prueba la capacidad de generalización de su sistema de ajuste usando datos de prueba.
- Exporte su sistema de inferencia fuzzy sintonizado al espacio de trabajo de MATLAB®.

Se puede utilizar el Neuro-Fuzzy Designer para entrenar un sistema de inferencia difuso de tipo Sugeno que:

- Tiene una sola salida.
- Utiliza el método del centroide como defuzzificación.
- Todas funciones de pertenencia de salida son del mismo tipo, por ejemplo lineal o constante.
- Tiene cobertura de regla completa sin compartir regla; Es decir, el número de reglas debe coincidir con el número de funciones de pertenencia de salida, y cada regla debe tener un consecuente diferente.
- Tiene peso unitario para cada regla.
- No utiliza funciones de pertenencia personalizadas.

2. Lógica difusa

La lógica difusa deriva de la lógica tradicional, y es más cercana a la forma de pensamiento humano.

La lógica difusa se basa en el hecho de que una afirmación no tiene porqué ser ni completamente cierta o completamente falsa, sino que se verificará en un cierto grado. De este modo, aparece el concepto de conjunto difuso, que no tiene una frontera definida, sino depende de la interpretación que queramos darle.

Ahora vamos a proceder a explicar los conceptos expuestos en la aplicación web, basándonos fundamentalmente de los apuntes de la asignatura Sistemas Inteligentes 2 de Ingeniería Informática en la Universidad de Málaga (Millán E. y Pérez de la Cruz J. L. (2014). *Apuntes de introducción al razonamiento difuso*) y también en los siguientes fuentes:

- Olmo, M. A. (2008). *Tutorial de introducción de lógica borrosa*.
- González, C. (2011). *Lógica difusa*.
- Arredondo, T. (2014). *Introducción a la lógica difusa*.

2.1. Conjuntos difusos

En los conjuntos clásicos se asigna el valor 0 en caso de que el valor no pertenezca al conjunto y 1 si pertenece. En la lógica difusa esta función varía de forma que los elementos del conjunto tomen valores del rango entre 0 y 1. Esta función se llama *función de pertenencia* y el conjunto definido por dicha función *conjunto difuso*.

Hay infinitas funciones que pueden ser usadas como funciones de pertenencia, porque principio la única restricción para usarlas es que los valores que puede tomar la función estén entre 0 y 1. Las más comunes son: función GAMMA, función L, función LAMBDA (triangular) y función PI (trapezoidal). También existen variaciones de estas mismas como son la función S, función Z y función Π , que son la versión cuadrática de las funciones GAMMA, L y PI respectivamente.

Función GAMMA (Γ).

$$\Gamma(x) = \begin{cases} 0 & \text{si } x \leq a \\ \frac{x - a}{m - a} & \text{si } a \leq x \leq m \\ 1 & \text{si } m \leq x \end{cases}$$

Función L. Puede definirse simplemente como $1 - \Gamma(x)$, o sea,

$$L(x) = \begin{cases} 1 & \text{si } x \leq a \\ \frac{m - x}{m - a} & \text{si } a \leq x \leq m \\ 0 & \text{si } m \leq x \end{cases}$$

Función LAMBDA (Λ) o triangular.

$$\Lambda(x) = \begin{cases} 0 & \text{si } x \leq a \\ \frac{x - a}{m - a} & \text{si } a \leq x \leq m \\ \frac{b - x}{b - m} & \text{si } m \leq x \leq b \\ 1 & \text{si } b \leq x \end{cases}$$

Función PI (Π_r) o trapezoidal.

$$\Pi_r(x) = \begin{cases} 0 & \text{si } x \leq a \\ \frac{x - a}{b - a} & \text{si } a \leq x \leq b \\ 1 & \text{si } b \leq a \leq c \\ \frac{d - x}{d - c} & \text{si } c \leq x \leq d \\ 0 & \text{si } d \leq x \end{cases}$$

Función S. Es la versión cuadrática de la Γ definida anteriormente.

$$S(x) = \begin{cases} 0 & \text{si } x \leq a \\ 2\left(\frac{x-a}{c-a}\right)^2 & \text{si } a \leq x \leq \frac{a+c}{2} \\ 1 - 2\left(\frac{x-a}{c-a}\right)^2 & \text{si } \frac{a+c}{2} \leq x \leq c \\ 1 & \text{si } c \leq x \end{cases}$$

Función Z. Es la versión cuadrática de la L definida anteriormente, y se puede definir simplemente por $Z(x) = 1 - S(x)$ o sea

$$Z(x) = \begin{cases} 1 & \text{si } x \leq a \\ 1 - 2\left(\frac{x-a}{c-a}\right)^2 & \text{si } \frac{a+c}{2} \leq x \leq c \\ 2\left(\frac{x-a}{c-a}\right)^2 & \text{si } a \leq x \leq \frac{a+c}{2} \\ 0 & \text{si } c \leq x \end{cases}$$

Función Π . Es la versión cuadrática de la Λ definida anteriormente, suponiendo además que es simétrica respecto al punto b del soporte. Se puede definir como $S(x)$ para $x \leq b$, y $Z(x)$ para $x \geq b$, o sea,

$$\Pi(x) = \begin{cases} S(x) & \text{si } x \leq b - d \\ Z(x) & \text{si } b - d \leq x \end{cases}$$

Otro dato a resaltar cuando hablamos de conjuntos difusos es el *soporte* y el *núcleo*:

Se define como **soporte** de un conjunto difuso A el conjunto nítido de elementos que tienen grado de pertenencia estrictamente mayor que 0:

$$\text{Soporte}(A) = \{x \in U \mid \mu_A(x) > 0\}$$

Se define como **núcleo** de un conjunto difuso A el conjunto nítido de elementos que tienen grado de pertenencia 1:

$$\text{Núcleo}(A) = \{x \in U \mid \mu_A(x) = 1\}$$

Al igual que en la teoría clásica de conjuntos, sobre los conjuntos difusos podemos definir las operaciones de complemento, unión, intersección, etc. Para definir dichas operaciones, se pide que las funciones cumplan unos requisitos mínimos.

En el caso de la unión son:

i1. Concordancia con el caso nítido.

$$u(0, 1) = i(1, 0) = i(1, 1) = 1$$

$$i(0, 0) = 0$$

u2. Conmutatividad. $u(\alpha, \beta) = u(\beta, \alpha)$.

u3. Asociatividad. $u(\alpha, u(\beta, \gamma)) = u(u(\alpha, \beta), \gamma)$.

- u4. Identidad.** $u(\alpha, 0) = \alpha$.
- u5. Monotonía.** si $\alpha \leq \alpha'$ y $\beta \leq \beta'$, entonces $u(\alpha, \beta) \leq u(\alpha', \beta')$.
- u6. Leyes de De Morgan.**
 $u(\alpha, \beta) = c(i(c(\alpha), c(\beta)))$
 $i(\alpha, \beta) = c(u(c(\alpha), c(\beta)))$

Y en el caso de la intersección son:

- i1. Idempotencia.** $A \cap A = A$ o sea
para todo $\alpha, \beta \in [0, 1]$ $i(\alpha, \alpha) = \alpha$.
- i2. Distributividad.** $A \cap (B \cup C) = (A \cap B) \cup (A \cap C)$ o sea
para todo $\alpha, \beta, \gamma \in [0, 1]$ $i(\alpha, u(\beta, \gamma)) = u(i(\alpha, \beta), i(\alpha, \gamma))$.
- u1. Idempotencia.** $A \cup A = A$ o sea
para todo $\alpha, \beta \in [0, 1]$ $u(\alpha, \alpha) = \alpha$.
- u2. Distributividad.** $A \cup (B \cap C) = (A \cup B) \cap (A \cup C)$ o sea
para todo $\alpha, \beta, \gamma \in [0, 1]$ $u(\alpha, i(\beta, \gamma)) = i(u(\alpha, \beta), u(\alpha, \gamma))$.

La teoría de conjuntos difusos nos permite representar hechos difusos. Se entiende por razonamiento difuso el proceso de realizar inferencias a partir de hechos difusos, así como de combinar evidencias y actualizar el grado de las creencias.

Unión

En teoría de conjuntos clásica, se considera que un elemento x pertenece al conjunto unión $A \cup B$ de dos conjuntos A y B si x pertenece a alguno de los conjuntos. En el caso difuso el problema consiste en determinar el grado de pertenencia $\mu_{A \cup B}(x)$ de x al conjunto unión, conocidos los grados de pertenencia $\mu_A(x)$, $\mu_B(x)$ a cada uno de los conjuntos originales. A estas funciones las llamaremos *t-conormas* y las más comunes son:

- T-Conorma del Máximo $u_{max}(A, B) = \max(A, B)$
- T-Conorma de la Suma $u_+(A, B) = A + B - A * B$
- T-Conorma de la Suma Drástica $u_{sup}(A, B) = \begin{cases} A \text{ si } B = 0 \\ B \text{ si } A = 0 \\ 1 \text{ en otro caso} \end{cases}$

Intersección

En teoría de conjuntos clásica, se considera que un elemento x pertenece al conjunto intersección $A \cap B$ de dos conjuntos A y B si y sólo si x pertenece a ambos. En el caso difuso el problema consiste en determinar el grado de pertenencia $\mu_{A \cap B}(x)$ de x al conjunto intersección, conocidos los grados de pertenencia $\mu_A(x)$, $\mu_B(x)$ a cada uno de los conjuntos originales. A estas funciones las llamaremos *t-normas* y las más comunes son:

- T-Norma del Mínimo $i_{min}(A, B) = \min(A, B)$

- T-Norma del Producto $i_+(A, B) = A * B$
- T-Norma del Producto Drástico $i_{inf}(A, B) = \begin{cases} A & \text{si } B = 1 \\ B & \text{si } A = 1 \\ 0 & \text{en otro caso} \end{cases}$

Complemento

Dado un conjunto nítido A, el conjunto A^C complementario de A está formado por los elementos del universo que no pertenecen a A. En el caso difuso, este conjunto vendrá definido por una función de pertenencia μ_{A^C} que se calcula para cada elemento x a partir de su pertenencia μ_A al conjunto A, aplicándole cierta función c.

- Complementario $u_{\bar{A}}(x) = C(u_A(x))$
- Complementario de Yager $c_w(x) = (1 - x^w)^{1/w}$
- Complementario de Sugeno $C_\lambda(x) = \frac{1-x}{1-\lambda x}$

2.2. Hechos difusos

Un *hecho difuso simple* o *proposición difusa simple* es la afirmación de que cierta variable lingüística toma cierto valor, por ejemplo: "la estatura de Pedro es mediana". Una proposición difusa tiene por tanto asociado un conjunto difuso A (que en este caso el valor lingüístico asignado es "mediano") y su correspondiente función de pertenencia μ_A definida sobre los elementos del universo de discurso U.

Para cada valor $u \in U$, el valor de verdad $\mu_P(u)$ de la proposición difusa p dada por "u es A" se identifica con el grado de pertenencia $\mu_A(u)$ de u al conjunto A, con otras palabras, cada valor que está en una proposición tiene un valor que viene dado por la función de pertenencia del valor u en el conjunto A.

Hechos difusos compuestos

Un *hecho difuso compuesto* no es más que la concatenación de dos o más hechos difusos simples a través de una serie de conectivas, con la **excepción** de la negación que solo necesitaríamos un solo hecho difuso simple, ya que solo un hecho difuso simple y la conectiva de la negación formarían un hecho difuso compuesto. Las conectivas son:

NO (Negación): que se representa con el símbolo \neg .

Y (Conjunción): que se representa con el símbolo \wedge .

O (Disyunción): que se representa con el símbolo \vee .

Ahora suponiendo que P es "La velocidad es alta" y Q es "La distancia es pequeña" podemos construir enunciados del tipo:

- $\neg P$ La velocidad **NO** es alta
- $P \wedge Q$ La velocidad es alta **Y** la distancia es pequeña
- $P \vee Q$ La velocidad es alta **O** la distancia es pequeña

Una vez definido los hechos difusos compuestos tenemos que definir para cada conectiva su correspondiente *operador lógico difuso* para poder tratarlos como proposiciones, es decir, que tengan sus propios valores de verdad.

Supongamos que tenemos dos proposiciones difusas, p y q, y tenemos los conjuntos difusos que intervienen en ellas, A y B respectivamente, además de tener sus funciones de pertenencias μ_A y μ_B definidas respectivamente sobre los universo U y V. Entonces podemos definir los operadores lógicos como:

NO

vendrá definida por la función de pertenencia al complemento de A

Y

vendrá definida por la función de pertenencia del tipo intersección, por ejemplo t-norma del mínimo

O

vendrá definida por la función de pertenencia del tipo unión, por ejemplo t-conorma del máximo

Con lo anterior podemos dar un valor $\alpha \in [0, 1]$ a cualquier proposición difusa en la que aparecen las variables u, v si conocemos sus valores nítidos u_1 y v_1 .

Correspondencia entre hechos difusos

Hay casos en los que el valor conocido es un conjunto difuso, por ejemplo, “aproximadamente 100”. En este caso hay que determinar el valor de correspondencia entre “la velocidad es alta” y “la velocidad es aproximadamente 100”. La correspondencia entre la premisa “v es A” y el hecho “v es B” suele definirse como la *posibilidad de A dado B*, $Pos(A|B)$:

$$Pos(A|B) = Pos(B|A) = \max(A \cap B)$$

Con esto podemos dar un valor $\alpha \in [0, 1]$ a cualquier proposición difusa en la que aparecen las variables u, v si conocemos sus valores difusos A' , B' .

Modus Ponens Generalizado

El *modus ponens generalizado* es la forma en la que se realiza la inferencia, a partir de hechos difusos y reglas difusas. El esquema del modus ponens generalizado es el siguiente:

$$\frac{\begin{array}{l} \text{SI } u \text{ es } A \quad \text{Entonces } v \text{ es } B \\ u \text{ es } A' \end{array}}{v \text{ es } B'}$$

Donde tenemos:

A - antecedente de la regla

B - consecuente de la regla

A' - hecho difuso

B' - conjunto a determinar

Primero: determinar el grado de satisfacción del antecedente de la regla (u es A) con el hecho difuso (u es A') mediante el método anteriormente expuesto, y de esta forma se obtiene el valor numérico $z \in [0, 1]$.

Después: A partir de z y del consecuente de la regla (v es B), el conjunto B' se determina por

$$\mu_{B'}(v) = \min(z, \mu_B(v))$$

Esta interpretación de la regla SI u es A ENTONCES v es B es la habitual en los SBC (Sistemas Basados en el Conocimiento) difusos. Esto corresponde a la llamada implicación de **Mandani**.

2.3. Decodificación

En muchos casos el resultado final del proceso de razonamiento difuso debe ser un valor nítido. Se denomina decodificación, nitidificación o defuzzificación al proceso de obtener a partir de un conjunto difuso A un valor nítido a que lo represente adecuadamente.

Los métodos más usados son:

Método de la media de los valores de máxima pertenencia (MOM, Mean of Maximum)

Sea μ_1 el grado de pertenencia máximo que se alcanza en A, y sea X el conjunto de valores u del universo de discurso para los cuales $\mu_A(u) = \mu_1$. Entonces el valor nítido a representativo del conjunto difuso A es el valor medio de X.

Desarrollemos esta definición para varios casos frecuentes:

■ X es un conjunto formado por n puntos aislados, $X = \{x_1, x_2, \dots, x_n\}$. Entonces el valor medio de X será

$$a = \frac{\sum_{1 \leq i \leq n} x_i}{n}$$

■ X es un intervalo, $X = [x_1, x_2]$. Entonces el valor medio de X será

$$a = \frac{x_1 + x_2}{2}$$

■ X es la unión de varios intervalos, $X = [x_{11}, x_{12}] \cup \dots \cup [x_{n1}, x_{n2}]$. Entonces el valor medio de X será

$$a = \frac{\sum_i (x_{i2} - x_{i1})(x_{i2} + x_{i1}) / 2}{\sum_i (x_{i2} - x_{i1})}$$

Método del centroide (COG, Center of Gravity)

Sea A un conjunto difuso con universo de discurso U y soporte S. El valor nítido $a \in S$ que se toma para representar A será la abscisa del centroide de A, o sea,

$$a = \frac{\int_S u \mu_A(u) du}{\int_S \mu_A(u) du}$$

Si el conjunto A es simétrico con respecto a un eje vertical $u = u_1$, es decir, $\mu_A(u_1 - x) = \mu_A(u_1 + x)$, se comprueba fácilmente que el método COG coincide con el método MOM.

El método del máximo más grande (LOM, *Largest of Maximum*)

El valor nítido es el valor más grande, es decir, el último valor más alto del conjunto difuso obtenido.

El método del máximo más pequeño (SOM, *Smallest of Maximum*)

El valor nítido es el valor más pequeño, es decir, el primer valor más alto del conjunto difuso obtenido.

2.4. Sistemas de tipo TS

Los sistemas TS o TSK (Takagi–Sugeno o Takagi–Sugeno–Kang) son también muy empleados para el modelado y control difuso. Las reglas TS son de la forma:

SI u_1 es A_1 Y u_2 es A_2 Y . . . ENTONCES $v = f(u_1, u_2, \dots)$

donde u_1, u_2, \dots, v son variables nítidas. Es decir, a diferencia de las reglas de tipo Mamdani, las reglas de tipo TS tienen un consecuente nítido.

Ejemplo: SI velocidad es alta ENTONCES $f = 2 \cdot \text{velocidad}$

Para cada variable de salida v , siempre habrá al menos dos reglas R_1, R_2 , en cuyo consecuente se asigna un valor a v . En general, sean R_1, \dots, R_n , las reglas que asignan un valor a v . Sean v_1, \dots, v_n los valores que cada regla asigna a la variable v . Sean z_1, \dots, z_n los grados de correspondencia de los valores de las variables de entrada con los antecedentes de cada regla. Entonces, el valor calculado para v es

$$v = \frac{\sum_1^n z_i v_i}{\sum_1^n z_i}$$

es decir, la media de los valores asignados por cada regla ponderados por los grados de correspondencia. Nótese que esta definición es igualmente válida si se consideran varias variables de entrada.

3. Especificaciones técnicas

En este apartado vamos a especificar los requisitos de proyecto que se han obtenido a partir de varias conversaciones con la directora del proyecto. Podemos encontrar una ampliación del documento de requisitos en el anexo B, para el cual se ha seguido la norma IEEE (*Especificación de Requisitos según el estándar de IEEE 830* (s.d.) (2008)).

Otro dato a comentar es que la aplicación está desplegada en Heroku (Anaya, J. (2015). *Instalar un proyecto Django en Heroku* (Consultada en 2017))

A continuación se va a proceder a explicar los aspectos más destacados.

3.1. Propósito

Con FuzzyTeacher se pretende facilitar el aprendizaje de la lógica difusa, tanto como herramienta auxiliar en la docencia presencial como en un posible uso autodidáctico de la aplicación. A continuación exponemos los requisitos y funciones del producto. El documento va dirigido tanto al desarrollador de la aplicación (para no perder el foco de los objetivos a conseguir) y para futuros desarrolladores que quieran trabajar sobre este proyecto.

3.2. Ámbito del sistema

La aplicación tendrá el nombre de FuzzyTeacher, y se pretende usar en el ámbito de la enseñanza. Se trata de una aplicación interactiva, que dispone de una variedad de ejemplos, ejercicios y problemas. Al ser una aplicación web gratuita está al alcance de todos, sin necesidad de instalar ningún software adicional.

La aplicación muestra unos contenidos básicos de lógica difusa con una serie de ejemplos para adquirir conocimientos. Se ha procurado utilizar un lenguaje sencillo y asequible pero mantener cierta rigurosidad. Para poner a prueba los conocimientos adquiridos, se cuenta con ejercicios, problemas y dos test, en los cuales se nos informa si la solución es correcta o no. En algunos casos se nos facilita alguna ayuda en caso de no ser la respuesta correcta.

Tanto los test como los problemas no son dinámicos, es decir, no hay un repertorio de problemas y se muestran aleatoriamente, son siempre los mismos. Para poder incluir nuevos problemas con sus soluciones sería necesario adaptar el código.

3.3. Perspectiva del producto

Esta aplicación se puede considerar una ampliación del tutorial de lógica difusa de Olmo Castillo, M. A. (2008). *Tutorial de Introducción de Lógica Borrosa* para la Universidad Politécnica de Madrid recuperado de http://www.dma.fi.upm.es/recursos/aplicaciones/logica_borrosa/web/tutorial_fuzzy/in_dice.html ya que en ese proyecto el temario es más básico y teórico.

También puede complementar a la aplicación fuzzylite (<http://www.fuzzylite.com>) puesto que mientras que Fuzzylite se centra en la resolución de problemas de control difuso, nuestra aplicación se orienta más hacia el aprendizaje de los conceptos relativos a la lógica difusa y su aplicación en problemas de control.

3.4. Funciones del producto

Como ya hemos comentado, nuestra aplicación está orientada a la enseñanza. Hay un temario que el usuario puede leer con una serie de ejercicios para facilitar y verificar la adquisición de conocimientos. Una vez acabado el temario, dispondremos de problemas que van incrementando la dificultad y dos test para poner a prueba los conocimientos teóricos.

Finalmente los usuarios pueden llevar un control de los temas leídos y de los ejercicios realizados.

3.5. Características de los usuarios

Al ser una aplicación educativa no se requiere que el usuario tenga grandes conocimientos de lógica difusa, pero si es aconsejable tener algunos conocimientos de lógica tradicional (aunque no son imprescindibles para poder usar la aplicación).

El único requisito para poder utilizar la aplicación es tener un acceso web.

3.6. Requisitos futuros

Futuras mejoras que se pueden incluir en la aplicación pueden ser:

- Envío de correo con un código para verificar y finalizar la creación de un nuevo usuario.
- La posibilidad de añadir nuevos problemas y test por parte de usuarios con privilegios.
- Hacer que las preguntas de los test sean dinámicas cada vez que realizamos un test.
- Añadir el rol de alumno/profesor para añadir y realizar test.
- Poder hacer problemas prácticos con datos y reglas para obtener resultados donde introducimos los conjuntos y las reglas y nos muestra un resultado según el valor de entrada.

3.7. Requisitos funcionales

1. **Crear un usuario personal:** El usuario debe crear una cuenta para poder acceder a la aplicación con su nombre de usuario y contraseña. La aplicación debe permitir acceder a cualquier usuario que introduzca unas credenciales almacenadas en el sistema.
2. **Leer temario y visualizar ejemplos:** La aplicación debe permitir a un usuario registrado acceder a todo el temario y ejemplos para su visualización y lectura.
3. **Poner a prueba los conocimientos prácticos:** Un usuario registrado puede poner a prueba sus conocimientos prácticos y para ello dispone:
 - 3.1. **Ejercicios:** Están incluidos en los temarios y donde el usuario debe introducir unos datos o seleccionar la respuesta correcta.
 - 3.2. **Problemas:** Situados en otro apartado donde el usuario debe introducir los valores del conjuntos final y el valor preguntado.En ambos casos el sistema debe indicar si el resultado es correcto o no.
4. **Poner a prueba los conocimientos teóricos:** Apartado final del menú donde el usuario puede realizar dos test indicando la(s) solución(es) y comprobarlo al final. La aplicación debe indicar que respuestas son correctas y cuáles no.
5. **Control del avance:** Panel visible en todo momento y que se actualiza de forma dinámica donde el usuario puede ver los temas abiertos y los problemas y test realizados.

3.8. Requisitos no funcionales

Lenguaje: Para esta aplicación se ha usado Python, Django y HTML5 apoyado por la siguiente documentación:

- *Django documentation* (s.d.) (Consultada en 2017)
- Kirkgaard, M. (2015). *PostgreSQL*. (Consultada en 2017)
- *The Django Book* (s.d.). (Consultada en 2017)
- Venegas, J. M. (2015). *Tutorial del Python virtualenv*.(Consultada en 2017)

Además de usar Pycharm que es una IDE (Integrated Development Environment) usado para la programación, especialmente para el lenguaje Python y soporte el desarrollo web yo Django.

Accesibilidad: La aplicación es accesible desde cualquier navegador web.

Disponibilidad: La aplicación (excepto problemas) estará disponible ininterrumpidamente.

3.9. Casos de uso

A continuación se van a mostrar los tres casos de uso que tiene la aplicación:

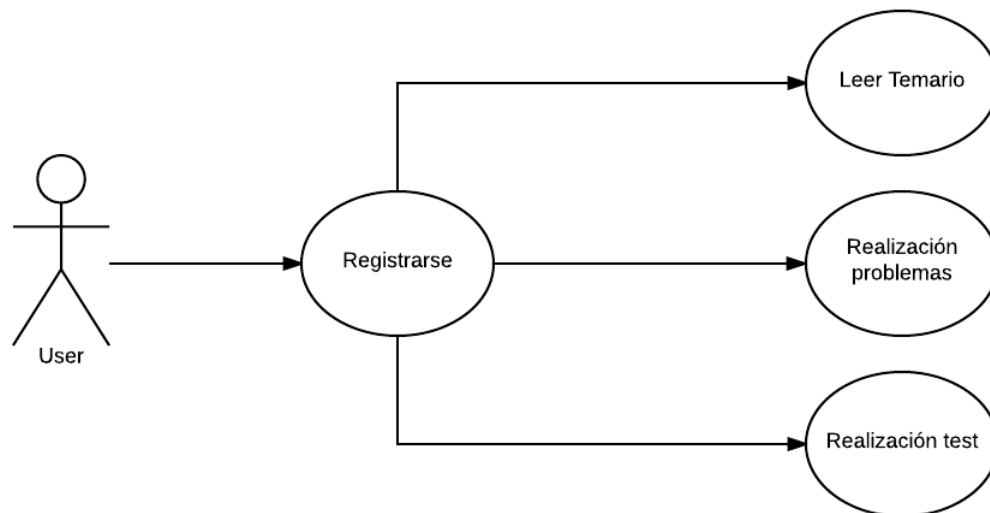


Figura 1. Casos de uso

Lectura de temario

1. El usuario debe registrarse
2. En el menú superior se debe seleccionar el botón de “Apuntes” y nos redirigirá a la introducción, además se mostrará un menú a la izquierda.
3. En el menú de la izquierda se podrá seleccionar el tema deseado o ir avanzando con los botones de la parte inferior de la pantalla.

Resolución de problemas

1. El usuario debe registrarse
2. En el menú superior se debe seleccionar el botón de “Apuntes” que nos redirige a la introducción y debemos seleccionar el apartado de problemas en el menú de la izquierda que nos mostrará los pasos a seguir para la resolución de problemas.
3. En el menú de la izquierda se podrá seleccionar el problema deseado.

Resolución de tests

1. El usuario debe registrarse
2. En el menú superior se debe seleccionar el botón de “Apuntes” que nos redirige a la introducción y debemos seleccionar el apartado de tests en el menú de la izquierda que nos mostrará la explicación de los tests.
3. En el menú de la izquierda o en la ventana actual se podrá seleccionar el test deseado

4. Diseño de la aplicación

A continuación vamos a explicar el diseño de la aplicación y procederemos a explicar porque se han tomado estas decisiones de diseño para más adelante exponer algunas capturas de pantalla de la aplicación.

La aplicación tiene un diseño simple, para resultar muy intuitiva para el usuario que es el que va a trabajar con la herramienta. El usuario puede saltar de una parte del temario a otra por medio de un menú o de unos botones provistos en la parte inferior de cada ventana.

Ahora se va a proceder a explicar las diferentes partes o aspectos destacados de la aplicación:

Login

Es lo primero que nos encontramos y es donde el usuario puede acceder a la aplicación. En caso de no estar registrados el usuario puede acceder a la zona de creación de usuario.

Home

Es la primera zona que nos encontramos nada más registrarnos. Es una zona de presentación, pues solo muestra una imagen, pero como ya estamos registrados podemos acceder al temario a través del menú (en la parte superior).

Apuntes

En esta zona se accede a la aplicación en sí, donde podemos encontrar apuntes, ejemplos, ejercicios y los test, además de poder visualizar por primera vez el panel de datos del usuario.

En esta zona también aparece un nuevo menú donde podremos seleccionar a qué parte del temario queremos ir. Una vez dentro, se desplegará un submenú (en el caso de que lo tenga) para poder movernos dentro de esa parte de los apuntes.

Los apuntes en sí son textos e imágenes introducidos en su mayoría en archivos HTML.

Ejemplos

Los ejemplos están realizados de la misma forma que los apuntes, es decir, son textos, con imágenes en algunos casos, introducidos en archivos HTML.

Imágenes

Las imágenes están depositadas en una parte del proyecto por lo que son estáticas.

Ejercicios

Están formados de forma similar a los ejemplos, pero disponen de unos *inbox* donde el usuario puede poner los resultados y comprobarlo por medio de una comprobación con los resultados que están almacenados en el sistema. Un proceso simple pero que ayuda al usuario en el proceso de aprendizaje.

Test

Funcionan de manera similar a los ejercicios, pero al contrario que ellos (que son corregidos de forma individual, incluso entre apartados) se corrigen todos al mismo tiempo obteniendo como resultado un valor numérico (nota) el cual expresa el grado de conocimiento del usuario

Panel de usuario

En este panel se visualiza la información guardada del usuario, como el nombre, los temas leídos (que se pueden comprobar en el mismo menú, donde mediante marcas se lleva un registro de los temas ya leídos), los ejercicios realizados y los problemas y test realizados. Es un panel sencillo pero que ayuda a llevar un seguimiento al usuario.

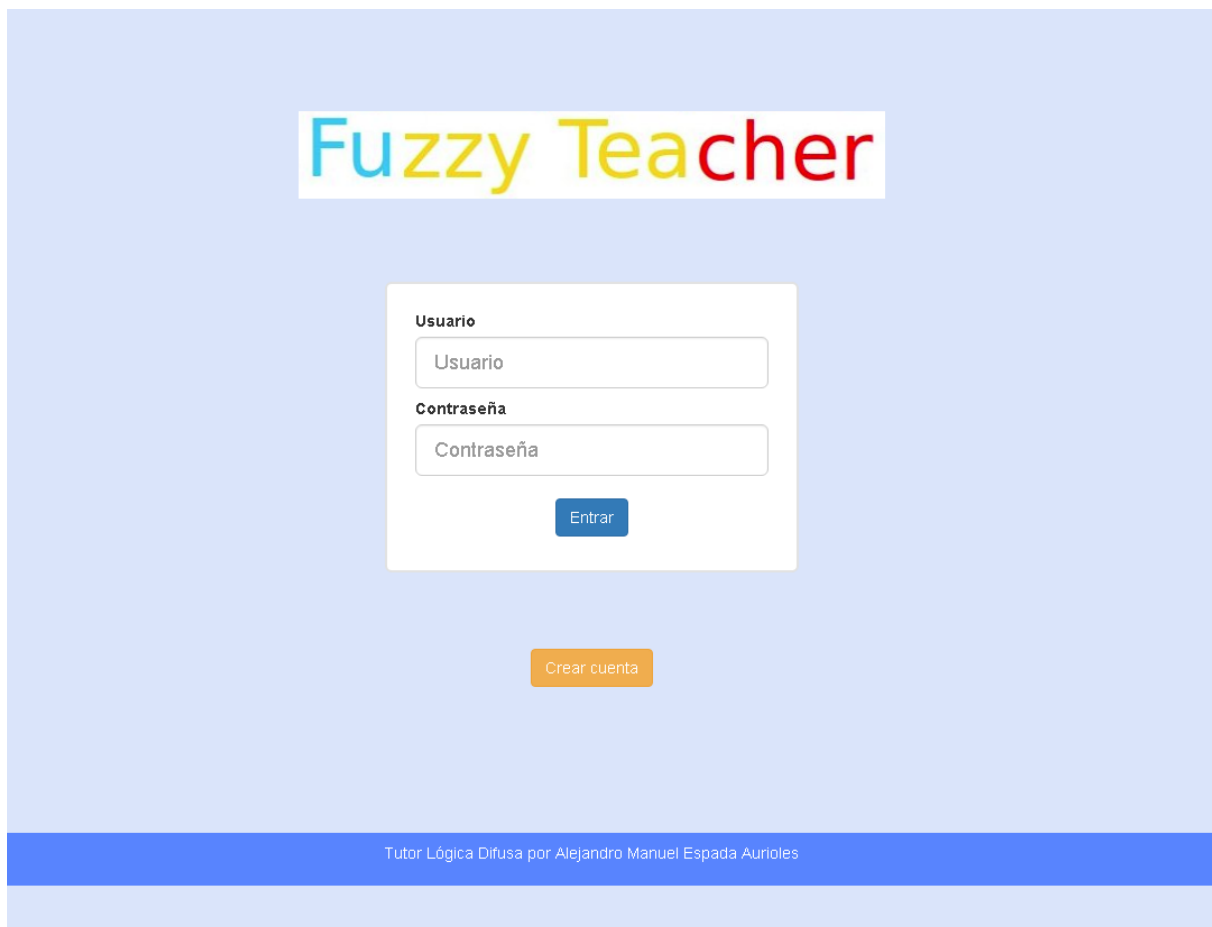
5. Diseño de la interfaz

Ahora vamos a exponer algunas capturas de pantalla de la aplicación para hacer una idea de cómo es la aplicación y cuáles son los apartados a destacar dentro de la interfaz gráfica.

Como dato a destacar en esta parte, toda la interfaz gráfica está realizada con HTML5 y se usa el framework Bootstrap (Bootstrap (s.d.). (Consultada en 2017)) para los botones.

5.1. Login

Lo primero que vamos a mostrar en la ventana de login (Figura 2), en la que como podemos observar solo nos pide un usuario y una contraseña. En el caso de no tener cuenta podemos acceder a la creación de la misma.

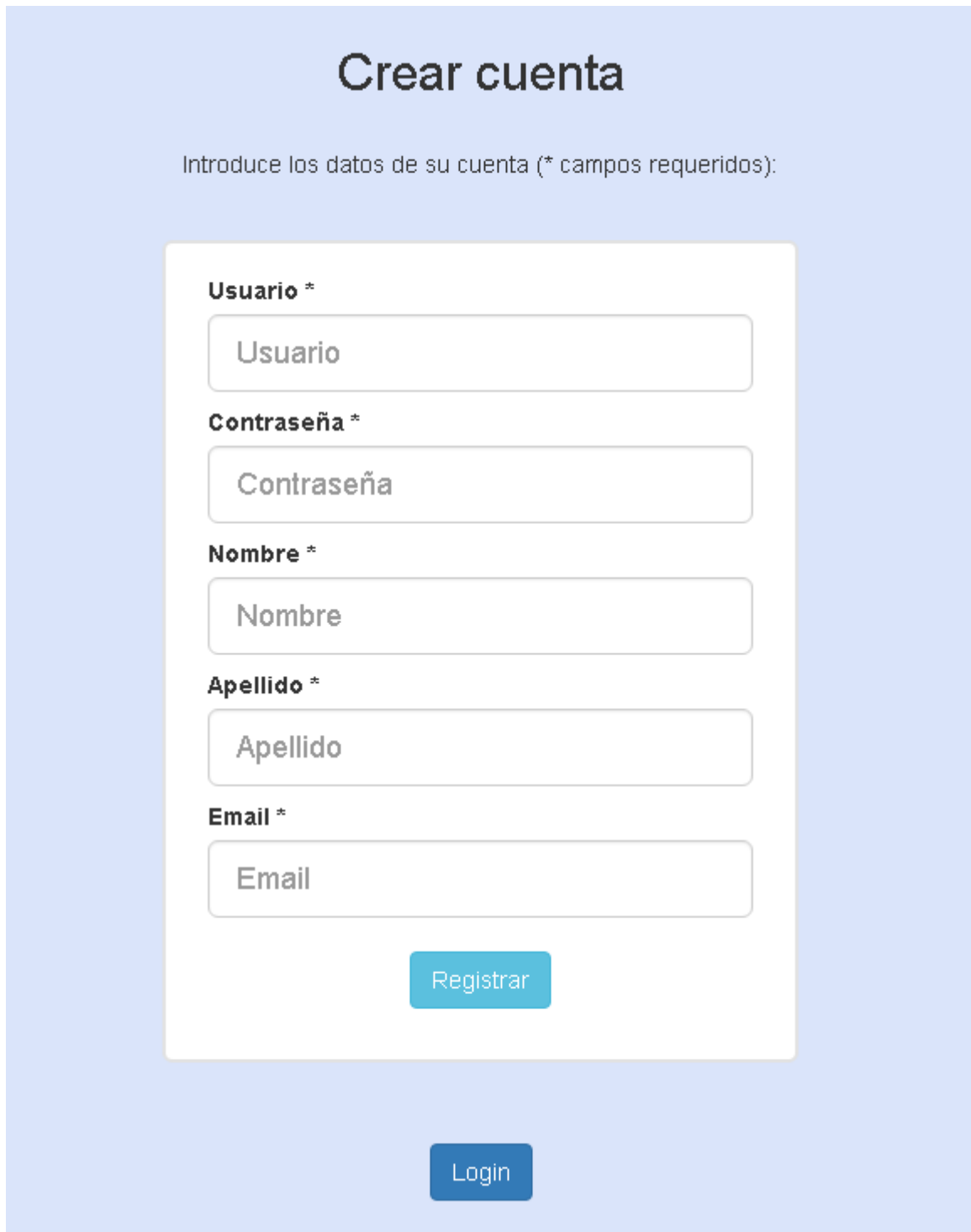


The image shows a login page for 'Fuzzy Teacher'. At the top center, the text 'Fuzzy Teacher' is displayed in a colorful font (blue, yellow, red). Below this, there is a white login form with two input fields: 'Usuario' and 'Contraseña'. A blue 'Entrar' button is positioned below the password field. Below the login form, there is an orange 'Crear cuenta' button. At the bottom of the page, a blue footer bar contains the text 'Tutor Lógica Difusa por Alejandro Manuel Espada Auriolés'.

Figura 2. Ventana de login

5.2. Añadir cuenta

En esta ventana podemos crearnos una cuenta y para ello tenemos que introducir los datos que se muestran en la captura de pantalla. También comprueba que el usuario y el email no estén ya introducidos en el sistema.



Crear cuenta

Introduce los datos de su cuenta (* campos requeridos):

Usuario *

Contraseña *

Nombre *

Apellido *

Email *

Registrar

Login

Figura 3. Formulario para crear una cuenta

5.3. Home

Es la primera vista que nos encontramos cuando accedemos a la aplicación, en la que se muestra una gráfica con el nombre de la aplicación y el nombre y apellidos del usuario, tal y como se ve en la Figura 4

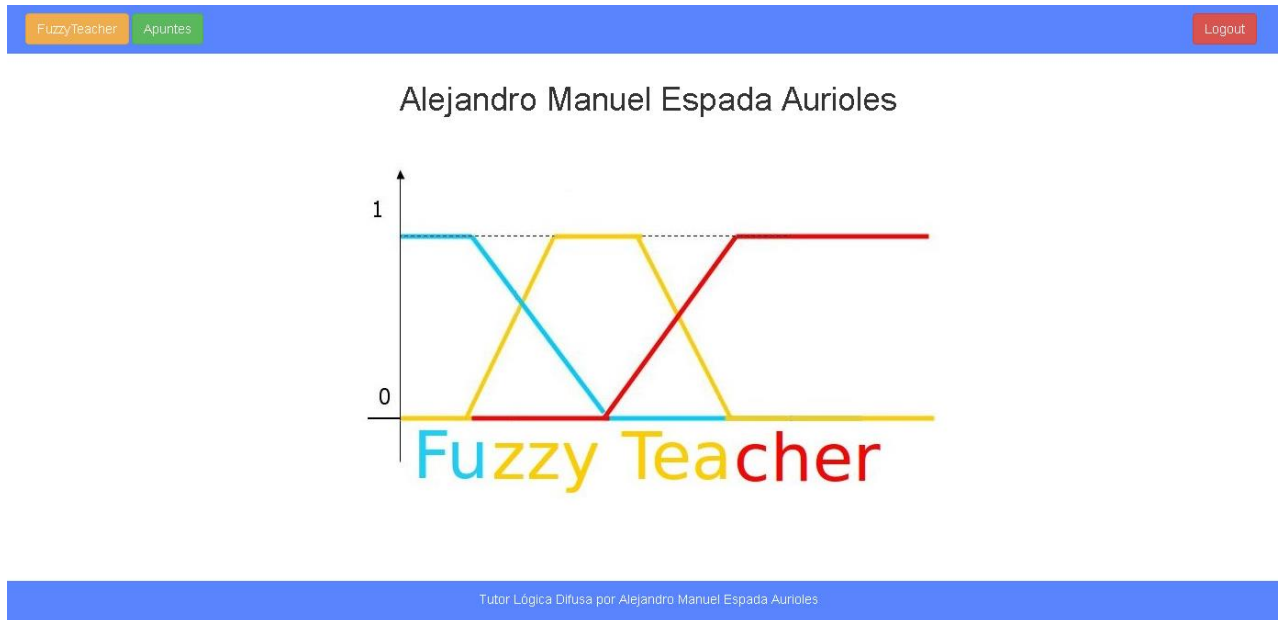


Figura 4. Ventana de entrada a la aplicación

5.4. Botones de avance y retroceso

Estos botones están en todas las ventanas de apuntes y sirven para poder avanzar, retroceder o ir a la introducción del tema en el que estamos. Tanto para avanzar como para retroceder disponemos de dos opciones, la primera es al siguiente o anterior tema o la otra opción es a la siguiente o anterior ventana.

En casos de que la ventana fuera la primera o última del tema, iríamos al tema siguiente o anterior

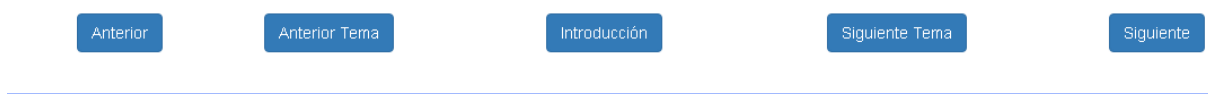


Figura 5. Menú de botones dentro del temario

5.5. Apuntes

Ahora vamos a mostrar el aspecto que tiene la aplicación cuando se muestra el contenido de un temario. Como se puede observar, en la parte superior está el menú que vamos a tener visible en toda la aplicación mientras tengamos iniciada una sesión, en la parte izquierda podemos ver el temario (con un color oscuro los temas y con un color claro los diferentes apartados del tema en el que estamos).

Este submenú aparece y se oculta dependiendo del tema en el que nos encontramos. En la parte izquierda, debajo del temario, nos encontramos con el panel de información que describiremos a continuación.

Finalmente en la parte central/derecha nos encontramos con la información principal, en este caso el temario, pero como veremos más adelante es la misma posición que ocupan los ejercicios, problemas y tests.

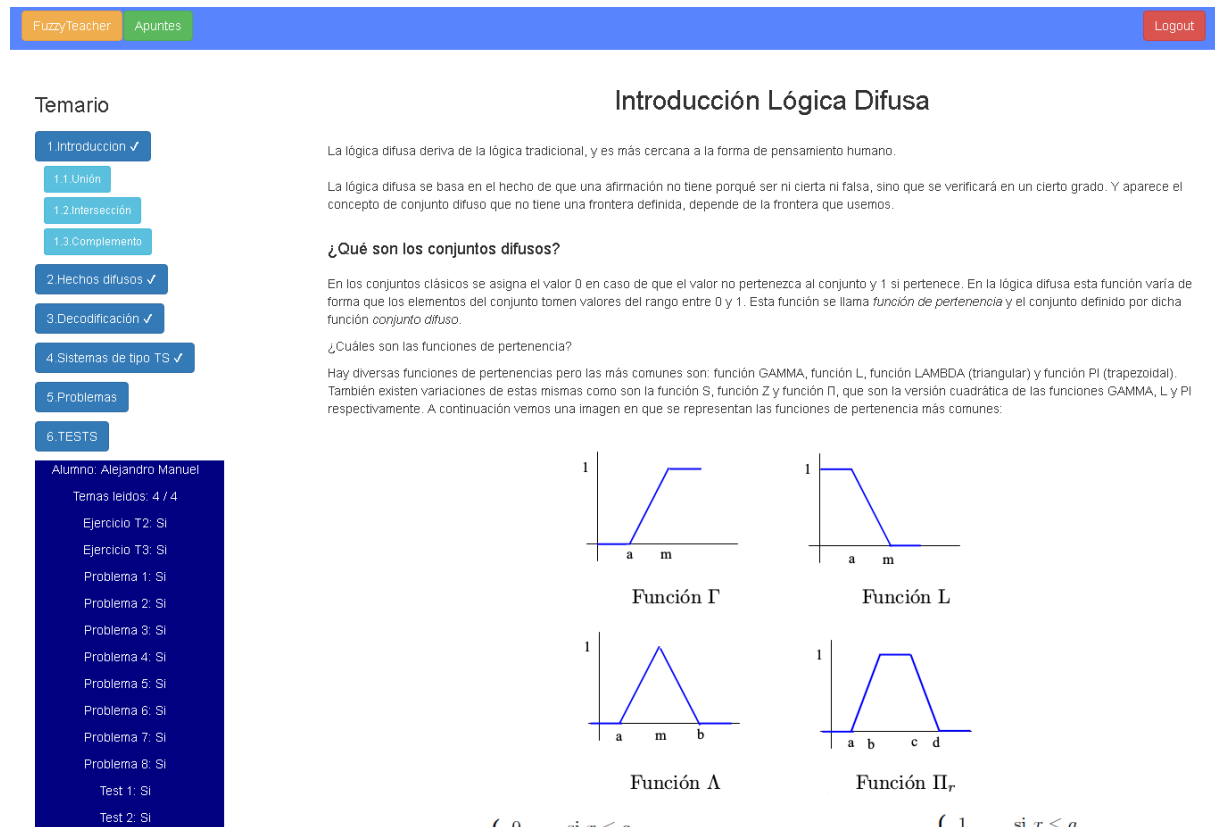


Figura 6. Ejemplo de una ventana en la que se muestra el temario

5.6. Panel de información

Esta parte de la interfaz es la que muestra al usuario sus avances y qué partes de la aplicación le falta por visitar.

Lo primero que nos muestra es nuestro nombre, para continuar con el número de temas leídos (realmente lo que se representa es si el usuario ha entrado o no en ese tema, debido a la dificultad que supone saber si realmente lo ha leído o no).

En esta parte no se muestra que tema es el que se ha leído porque a continuación vamos a ver que en el mismo temario podemos obtener esta información.

Por último, podemos revisar qué ejercicios, problemas y tests hemos realizado ya que en esta parte si me muestra uno a uno si se ha visitado esta página o no. Al igual que pasa con el temario nosotros no podemos saber si el usuario ha realizado o no el ejercicio ya que el sistema no le obliga a probar sus respuestas con el sistema (si el usuario lo desea, puede realizar el ejercicio sin tener que introducir los datos).

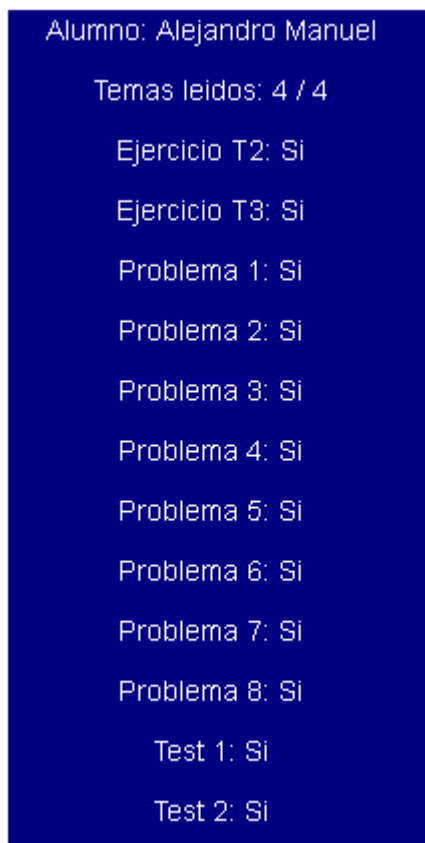


Figura 7. Menú de información del usuario

5.7. Temario

Ahora vamos a ver el menú del temario desglosado (es decir, con los submenús, ya que estos solo se muestran cuando estamos en ese tema mientras que los otros submenús están ocultos).

Temario

- 1. Introducción ✓
 - 1.1. Unión
 - 1.2. Intersección
 - 1.3. Complemento
- 2. Hechos difusos ✓
 - 2.1. Introducción
 - 2.2. Correspondencia entre hechos difusos
 - 2.3. Modus Ponens Generalizado
 - 2.4. Ejercicios
- 3. Decodificación ✓
 - 3.1. Introducción
 - 3.2. Ejercicios
- 4. Sistemas de tipo TS ✓

Figura 8. Primera parte del menú

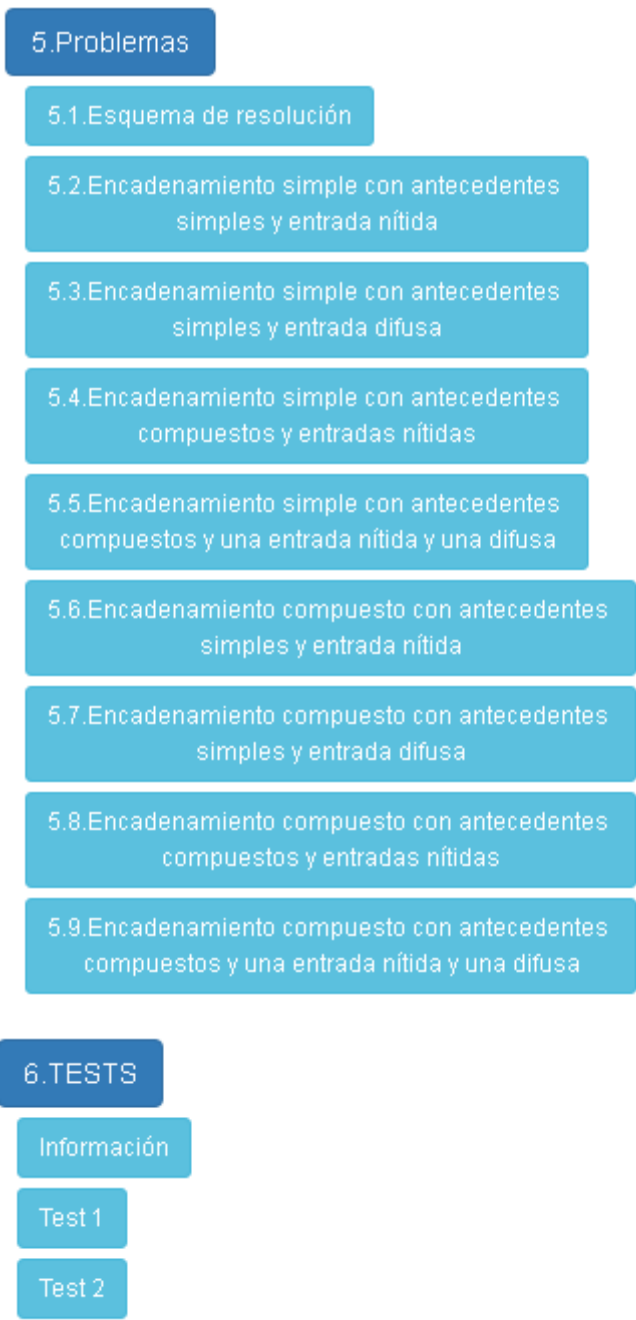


Figura 9. Segunda parte del menú

5.8. Ejercicios

Existen dos tipos de ejercicios: ejercicios tipo test o ejercicios en el que se pide unos datos concretos a rellenar en unas casillas específicas.

Como se puede observar, cada pregunta tiene su propio botón para verificar los resultados, de esta forma el usuario puede comprobar si los datos introducidos son los correctos o no.

Podemos ver ejemplos en las figuras 10, 11 y 12.

Ejercicios Tema 2

Ejercicio 1:

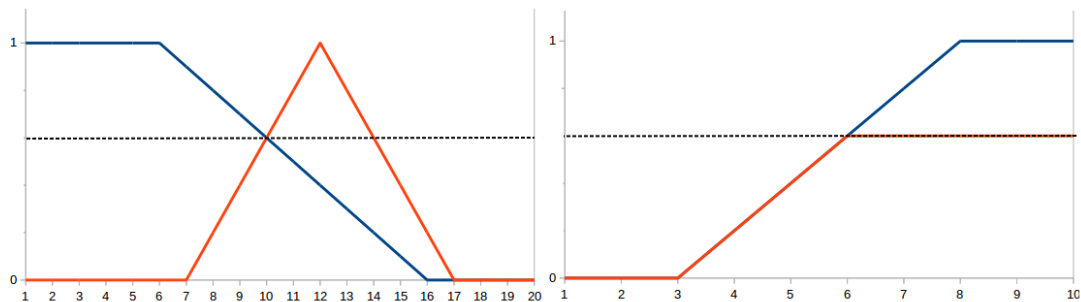
Tenemos los siguientes hechos difusos, indica cual o cuales son hechos difusos compuestos:

- La velocidad es alta y la temperatura es baja
- La velocidad es baja
- La velocidad es media o la temperatura es alta
- La velocidad no es baja
- La velocidad es alta y la temperatura es baja y la humedad no es alta

Comprobar

Ejercicio 2:

Dado el siguiente conjunto (1/1 6/1 16/0 20/0) y la entrada difusa (0/0 7/0 12/1 17/0 20/0) determinar el grado de pertenencia (z), y determinar los puntos característicos de conjunto de salida (0/0 3/0 8/1 10/1) (primer valor, cuando la recta cambia, aquí puede ser más de uno y último valor):



Valor del grado de pertenencia (z):

Eje x

Eje y

Comprobar

Valores característicos del conjunto de salida:

Inicial:

Eje x

Eje y

Cambio de recta:

Eje x

Eje y

Cambio de recta:

Eje x

Eje y

Final:

Eje x

Eje y

Comprobar

Figura 10. Ejemplo de una ventana de ejercicio

Valor del grado de pertenencia (z):

Eje x

Eje y

Comprobar

El valor no es correcto

Figura 11. Ejemplo de respuesta del sistema a una solución errónea

Valor del grado de pertenencia (z):

Eje x

Eje y

Comprobar

El valor es correcto

Figura 12. Ejemplo de respuesta del sistema a una solución correcta

5.9. Problemas

A continuación vamos a ver las ventanas de la parte de los problemas (Figura 13).

Primero nos encontramos con un esquema en el que se nos explican los pasos a seguir y también disponemos de un pequeño ejemplo de dicha resolución

Esquema de los problemas

Paso 1) Para cada una de las entradas, ya se trate de valores nítidos o conjuntos difusos, poner en correspondencia la entrada con los antecedentes de las reglas con objeto de determinar cuál es el conjunto de las reglas aplicables, y determinar el valor z con el cual se debe aplicar cada una de las reglas de dicho conjunto

Paso 2) Aplicar las reglas, obteniendo los conjuntos difusos de salida de las mismas

Paso 3) Unir los conjuntos difusos obtenidos, para obtener el conjunto difuso de salida

Paso 4) Volver al paso 1 utilizando ahora como entrada el conjunto difuso de salida, y repetir los pasos de 1 a 4 hasta que no haya más reglas aplicables

Paso 5) Defuzzificar el conjunto difuso final

Ejemplo

La abuela María prepara sus deliciosas galletas caseras de forma artesanal desde hace más de 40 años. El toque secreto de la receta consiste en hornearlas cuidadosamente de acuerdo a las siguientes reglas:

- (R1) Si las galletas están un poco crudas, entonces la temperatura del horno debe ser alta.
- (R2) Si las galletas están medio hechas, entonces la temperatura del horno debe ser media.
- (R3) Si las galletas están doraditas, entonces la temperatura del horno debe ser baja.

Tras diversas entrevistas con la abuela se han podido establecer los siguientes conjuntos difusos sobre un índice cromático especial (0 = galleta cruda, 10 = galleta chamuscada) y la temperatura del horno:

Índice cromático de las galletas:

un poco crudas:	(1/4, 0.5/6, 0/7)
medio hechas:	(0/3, 1/5, 1/6, 0/8)
doraditas:	(0/5, 1/7)

Temperatura del horno (°C):

baja:	(0/200, 1/210, 1/230, 0/240)
media:	(0/220, 1/240, 1/260, 0/280)
alta:	(0/260, 1/270, 1/290, 0/300)

Supóngase que se interpretan las reglas anteriores como implicaciones de Mamdani. Suponiendo que en cierto momento el índice cromático de las galletas es 6, ¿Cuál será el valor de temperatura aplicado al horno si se utiliza la técnica del primer valor máximo para obtener valores nítidos?

Resolución

1)

El índice cromático 6 pertenece a:

para determinar el valor en caso que no sepamos el valor exacto en el eje Y de la recta utilizaremos la función:

Figura 13. Esquema de resolución de problemas

Después podemos acceder a resolver cada problema. Este esquema se repite para todos los problemas del sistema: descripción, reglas, conjuntos, ayudas (gráficas correspondientes a los conjuntos difusos del ejercicio) y por último las preguntas, con sus opciones de comprobación (una por apartado).

A continuación, vamos a ver que las respuestas (Figuras 15-20) de esta sección son dinámicas y ayudan al usuario a encontrar la solución correcta (dentro de unos límites para no darle la solución directamente).

PROBLEMA 1

Queremos poner en funcionamiento un calefactor, para decidir a la potencia (entre 0 y 100 %) a la que lo ponemos tenemos que tener en cuenta la temperatura que hace (entre 0 °C y 40 °C). Tenemos tres modalidades de potencia del calefactor las cuales son alta, media y baja que dependerá de la temperatura a la que podríamos definir en alta, media y baja también.

Las reglas para decidir la potencia son:

- 1) Si la Temperatura es Baja entonces la Potencia es Alta.
- 2) Si la Temperatura es Media entonces la Potencia es Media.
- 3) Si la Temperatura es Alta entonces la Potencia es Baja.

Los conjuntos son:

Temperatura °C (0-40)

- Baja: [0/1, 10/1, 15/0]
- Media: [10/0, 15/1, 25/1, 30/0]
- Alta: [25/0, 30/1, 40/1]

Potencia % (0-100)

- Baja: [0/1, 20/1, 30/0]
- Media: [20/0, 30/1, 60/1, 70/0]
- Alta: [60/0, 70/1, 100/1]

Si sabemos que la temperatura es de 27 grados, ¿A que potencia tenemos que poner el calefactor?

Ayuda Gráfica Temperatura

Ayuda Gráfica Potencia

Ejemplo introducir datos: [(20/0),(25/0.7),(35/0.3),(40/0.3),(45/0)]

Resultado

Comprobar

¿Cuál es el resultado final, aplicando la defuzzificación MOM?

Resultado

Comprobar

Figura 14. Esquema de ventana de problemas (Problema 1)

Ahora vamos a ver las posibles respuestas del sistema cuando queremos comprobar una solución.

Primero cuando es la pregunta del resultado final:

¿Cuál es el resultado final, aplicando la defuzzificación MOM? **Correcto**

45

Comprobar

Figura 15. Ejemplo de respuesta del sistema a una solución correcta

¿Cuál es el resultado final, aplicando la defuzzificación MOM? **Incorrecto**

455

Comprobar

Figura 16. Ejemplo de respuesta del sistema a una solución errónea

Y por último cuando tenemos que introducir el conjunto resultante:

Ejemplo introducir datos: [(20/0),(25/0.7),(35/0.3),(40/0.3),(45/0)] **Incorrecto, (20/0) no pertenece a la solución o puede estar mal introducido**

[(20/0),(25/0.7),(35/0.3),(40/0.3),(45/0)]

Comprobar

Figura 17. Ejemplo de respuesta del sistema cuando la solución es errónea y el valor erróneo es el primero.

Ejemplo introducir datos: [(20/0),(25/0.7),(35/0.3),(40/0.3),(45/0)] **Incorrecto, (25/0.7) no pertenece a la solución o puede estar mal introducido**

[(0/0.40),(25/0.7),(35/0.3),(40/0.3),(45/0)]

Comprobar

Figura 18. Ejemplo de respuesta del sistema cuando la solución es errónea y el valor erróneo no es el primero.

Ejemplo introducir datos: [(20/0),(25/0.7),(35/0.3),(40/0.3),(45/0)] **Inicio o fin de la cadena de datos mal introducidos**

[(0/0.40),(25/0.7),(35/0.3),(40/0.3),(45/0)]

Comprobar

Figura 19. Ejemplo de respuesta del sistema cuando la solución no tiene el formato correcto.

Ejemplo introducir datos: [(20/0),(25/0.7),(35/0.3),(40/0.3),(45/0)] **Correcto**

[(0/0.40),(24/0.40),(26/0.60),(64/0.60),(70/0)]

Comprobar

Figura 20. Ejemplo de respuesta del sistema cuando la solución es correcta.

Podemos observar que las respuestas del sistema están pensadas para facilitar que el usuario encuentre la solución correcta, ya que en el caso de los problemas puede llegar a costar más trabajo llegar a ella, bien sea por un fallo a la hora de realizar los cálculos o un redondeo mal aplicado.

También podemos observar que cuando se introduce un dato que no es correcto (o bien está mal introducido) siempre nos muestra este dato. Además como va comprobando de izquierda a derecha el usuario puede saber que los datos introducidos hasta ese momento son correctos, lo cual le puede ayudar a progresar en la resolución del problema.

5.10 Tests

Finalmente, vamos a ver las ventanas relacionadas con los tests. En primer lugar, nos encontramos con la introducción (Figura 21) en la que se explica el sistema de puntuación de cada uno de los tests y un enlace a cada uno de ellos

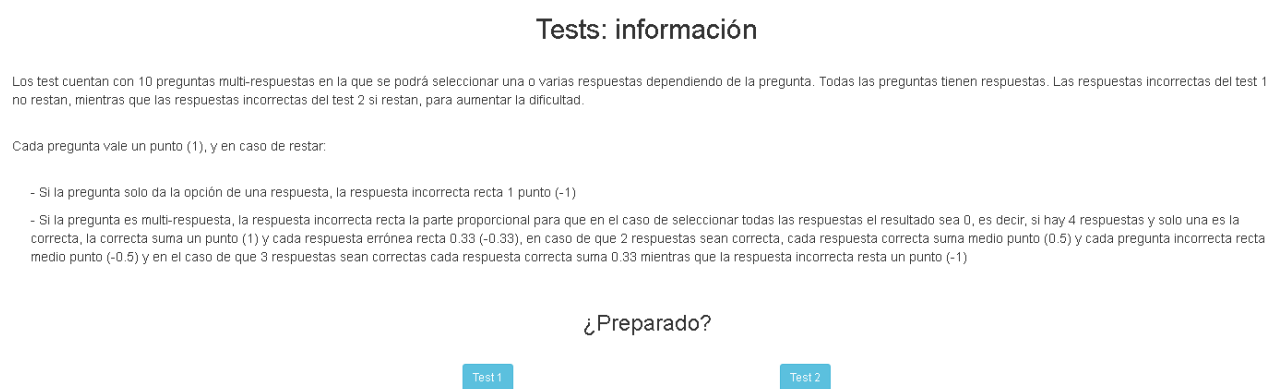


Figura 21. Ventana de explicación de los tests

A continuación se muestra en la figura 22 un ejemplo de uno de los test. Las preguntas que pueden ser tanto de una respuesta o multi-respuesta, y se muestran todas en la misma página para que el usuario pueda tener una visión cómoda de todo el test. Esto facilita el poder cambiar alguna respuesta de manera fácil, sin tener que ir saltando de ventana en ventana.

TEST 1

Las respuestas erróneas de este test NO restan.

1) ¿Cuántas funciones de pertenencia existen?

- a) 4
- b) 7
- c) Infinitas
- d) Ninguna es correcta

2) Las T-conormas más habituales son (Unión)

- a) Mínimo, Producto y Producto Drástico
- b) Máximo, Producto y Producto Drástico
- c) Mínimo, Suma y Suma Drástica
- d) Máximo, Suma y Suma Drástica

3) ¿Cuál es la función del complemento que podríamos definir como $C(A) = 1 - A$?

- a) Complementario
- b) Inverso
- c) Complemento de Sugeno
- d) Complemento de Yager

4) Un hecho difuso compuesto es la concatenación de más de dos hechos difusos simples

- a) Verdadero
- b) Falso

5) El modus ponens generalizado es la forma en la que se realiza la inferencia a partir de hechos difusos y reglas difusas

- a) Verdadero
- b) Falso

6) Las siglas del método de la media de los valores de máxima pertenencia es:

- a) SOM
- b) COG
- c) MOM
- d) LOM

7) Las siglas del método del centroide es:

- a) SOM
- b) COG
- c) MOM
- d) LOM

8) Las reglas del sistema de tipo Takasi-Sugeno (TS) se diferencian de las reglas de tipo Mamdani en que tienen un consecuente nítido

- a) Verdad
- b) Falso

9) El conjunto $\{0/0, 1/0, 2/0, 3/1, 4/1, 5/1\}$ es un conjunto de:

- a) La lógica clásica
- b) La lógica difusa
- c) Ambas lógicas
- d) Ninguna de estas lógicas

10) Se entiende por razonamiento difuso el proceso de realizar inferencias a partir de hechos difusos, así como de combinar evidencias y actualizar el grado de las creencias

- a) Verdad
- b) Falso

Comprobar

Figura 22. Ejemplo de test en el sistema (Test 1).

Por último, podemos ver cómo sería el resultado una vez hemos terminado el test: se muestra la nota que hemos sacado, sobre cuántos puntos se ha valorado dicha puntuación (en el caso del ejemplo 8 puntos de 10). Además de mostrar que preguntas hemos acertado y cuáles no. En esta parte no queda marcada la solución seleccionada, puesto que las respuestas no son dinámicas y además aparecen en la misma posición. Esto hace que el usuario tenga que volver a hacer el test, o en su defecto simular un test donde solo responda a esa pregunta, para poder saber si su respuesta es correcta o no. También está pensado para los casos en los que el usuario duda entre dos, o más, respuestas y al finalizar el test no recuerda cual seleccionó, al volver a realizar el test tendrá que volver a pensar su respuesta..

De este modo se pretende potenciar el carácter didáctico de estos tests:

La calificación es de 8/10

Las respuestas erróneas de este test NO restan.

1) ¿Cuántas funciones de pertenencia existen?

- a) 4
- b) 7
- c) Infinitas
- d) Ninguna es correcta

Se han seleccionado la respuesta correcta.

2) Las T-conormas más habituales son (Unión)

- a) Mínimo, Producto y Producto Drástico
- b) Máximo, Producto y Producto Drástico
- c) Mínimo, Suma y Suma Drástica
- d) Máximo, Suma y Suma Drástica

Se ha seleccionado una respuesta no correcta.

3) ¿Cuál es la función del complemento que podríamos definir como $C(A) = 1-A$?

- a) Complementario
- b) Inverso
- c) Complemento de Sugeno
- d) Complemento de Yager

Se han seleccionado la respuesta correcta.

4) Un hecho difuso compuesto es la concatenación de más de dos hechos difusos simples

- a) Verdadero
- b) Falso

Se ha seleccionado una respuesta no correcta.

Figura 23. Imagen parcial de la resolución de un test.

6. Tests

Ahora vamos a proceder a explicar los test y pruebas realizadas en esta parte del proyecto.

Como la mayor parte de la aplicación se basa en la transmisión de conocimiento y una serie de ejemplos, se ha realizado una revisión con objetivo de comprobar que todo era correcto y que tanto los textos como los ejemplos estaban bien situados. Ya que se trata de texto en lenguaje natural esta revisión no puede automatizarse de forma que se asegure una correcta verificación.

Para la parte de los ejercicios se han realizado pruebas manuales para comprobar que los resultados que daba por validos son los correctos. También se ha comprobado que tanto los textos como las posiciones de los mismos están en las zonas que les corresponde.

También se han realizado varios test en la realización manual de los ejercicios para comprobar que los resultados que se ofrecen al usuario son correctos. Para ello se han realizado los ejercicios en varias ocasiones sin tener acceso a los datos de la realización anterior y con algunos días de diferencia para que la memorización no interfiera en la realización del ejercicio.

Finalmente, también se ha utilizado la realización de ejercicios por parte de otro compañero.

Plan a seguir en el caso de detectar un error

Los pasos realizados en este proceso son los siguientes:

- 1) Identificar el error
- 2) En el caso de ser un texto, buscar la corrección y en el caso de ser un ejercicio práctico, buscar la solución o rehacer el problema para verificar que la solución es correcta.
- 3) Una vez encontrada la solución o los cambios oportunos se procede a realizar los cambios en el código de la aplicación.
- 4) Una vez se han realizado los cambios, se hace un chequeo rápido de la aplicación para comprobar que no ha afectado a otras partes.
- 5) Finalmente se realiza una comprobación más exhaustiva de la zona afectada para verificar que los cambios han resuelto el problema.

7. Conclusiones y ampliación del proyecto

Como conclusiones, creemos que este proyecto puede ayudar tanto a usuarios independientes que quiere empezar o mejorar sus conocimientos.

También puede ser una herramienta para la docencia, ya que se pueden usar a modo de temario.

En cuanto a los posibles aspectos a mejorar en el proyecto. Se podría contemplar una posible ampliación del mismo con el siguiente cambio; almacenar los resultados de los problemas y los enunciados de los mismos en una base de datos, para poder cambiarlos de forma dinámica sin tener que compilar la aplicación de nuevo. Del mismo modo, los textos de los apuntes se podrían cambiar a otros ficheros de traducciones para disponer de la aplicación en diferentes idiomas.

Como ampliación del proyecto podemos encontrar una rama principal que constaría con la elaboración de una parte más práctica donde el usuario pudiese crear sus propios problemas. Para ello, habría que elaborar una base de datos donde podamos guardar los proyectos, variables, universos, reglas, etc.

Los resultados de estos proyectos podrían ser tanto numéricos, para los casos de ser problemas reales en los que tengamos que conocer los resultados concretos tanto como unos resultados más visuales mediante gráficas.

Una vez realizada esta ampliación, otra línea de trabajo futuro que podría resultar bastante beneficiosa para este proyecto sería la realización de unos roles (profesores, alumnos e independientes). De este modo desde el rol profesor se podrían introducir nuevos contenidos, tanto temas como problemas o test, de uso público o restringido a un conjunto particular de estudiantes usuarios de la aplicación. Incluso sería posible implementar sistemas de corrección automática si así se desea.

8. Referencias bibliográficas

- Anaya, J. (2015). *Instalar un proyecto Django en Heroku*. (Consultada en 2017) Recuperado de: <https://pythonbc.com/blog/instalar-un-proyecto-django-en-heroku/>
- Arredondo, T. (2014). *Introducción a la lógica difusa*. Recuperado de: <http://profesores.elo.utfsm.cl/~tarredondo/info/soft-comp/Introduccion%20a%20la%20Logica%20Difusa.pdf>
- Buckley, J. y Eslami, E. (2002) *An Introduction to Fuzzy Logic and Fuzzy Sets (Advances in Intelligent and Soft Computing)* Alemania: Physuca-Verlag
- González, C. (2011). *Lógica difusa*. Recuperado de: http://www.esi.uclm.es/www/cglez/downloads/docencia/2011_Softcomputing/LogicaDifusa.pdf
- Kirkgaard, M. (2015). *PostgreSQL*. (Consultada en 2017) Recuperado de: <https://help.ubuntu.com/community/PostgreSQL>
- Li, H. y Gupta, M. (2013) *Fuzzy Logic and Intelligent Systems*. Editorial: Springer
- Martín del Brio, B. y Sanz, A. (2002) *Redes neuronales y sistemas difusos* (2 ed.): RA-MA Editorial.
- Millán E. y Pérez de la Cruz J. L. (2014). *Apuntes de introducción al razonamiento difuso*. Dpto. de Lenguajes y Ciencias de la Computación. Universidad de Málaga.
- Olmo, M. A. (2008). *Tutorial de introducción de lógica borrosa*. Recuperado de: http://www.dma.fi.upm.es/recursos/aplicaciones/logica_borrosa/web/tutorial_fuzzy/indice.html
- Venegas, J. M. (2015). *Tutorial del Python virtualenv*. (Consultada en 2017) Recuperado de: <http://rukbottoiland.com/blog/tutorial-de-python-virtualenv/>
- Ying H. (2000) *Fuzzy Control and modeling: Analytical foundations and applications*. IEEE Press Series on Biological Engineering.
- *Bootstrap* (s.d.). (Consultada en 2017) Recuperada del sitio web de W3Schools: http://www.w3schools.com/bootstrap/bootstrap_buttons.asp
- *Django documentation* (s.d.) (Consultada en 2017) Recuperado del sitio web de Django: <https://docs.djangoproject.com/en/1.10/>

- *Especificación de Requisitos según el estándar de IEEE 830* (s.d.) (2008) <https://www.fdi.ucm.es/profesor/gmendez/docs/is0809/ieee830.pdf>
- *The Django Book* (s.d.). (Consultada en 2017) Recuperado del sitio web de Django: <http://djangobook.com/>

A Resolución de problemas y tests

PROBLEMA 1

Queremos poner en funcionamiento un calefactor, para decidir a la potencia (entre 0 y 100 %) a la que lo ponemos tenemos que tener en cuenta la temperatura que hace (entre 0 °C y 40 °C). Tenemos tres modalidades de potencia del calefactor las cuales son alta, media y baja que dependerá de la temperatura a la que podríamos definir en alta, media y baja también.

Las reglas para decidir la potencia son:

- 1) Si la Temperatura es Baja entonces la Potencia es Alta.
- 2) Si la Temperatura es Media entonces la Potencia es Media.
- 3) Si la Temperatura es Alta entonces la Potencia es Baja.

Los conjuntos son:

Temperatura °C (0-40)

- Baja: [0/1, 10/1, 15/0]
- Media: [10/0, 15/1, 25/1, 30/0]
- Alta: [25/0, 30/1, 40/1]

Potencia % (0-100)

- Baja: [0/1, 20/1, 30/0]
- Media: [20/0, 30/1, 60/1, 70/0]
- Alta: [60/0, 70/1, 100/1]

Si sabemos que la temperatura es de 27 grados, ¿A que potencia tenemos que poner el calefactor?

¿Cuál es el resultado final, aplicando la defuzzificación MOM?

Solución

Temperatura = 27°

Por lo tanto pertenece al conjunto Media con un valor de 0.6 y al conjunto Alta con un valor de 0.4, entonces los conjuntos de Potencia se cortan en 0.4 para Baja y 0.6 para Media, dada las reglas, para el conjunto Alta el valor máximo es 0.

Entonces tenemos los conjuntos:

- Baja: [0/0.4, 24/0.4, 30/0] - Media: [20/0, 26/0.6, 64/0.6, 70/0] - Alta: [60/0]

Con un conjunto final: **[0/0.4, 24/0.4, 26/0.6, 64/0.6, 70/0]**

$$\text{MOM} = \frac{26+64}{2} = 45$$

PROBLEMA 2

Queremos poner en funcionamiento un calefactor, para decidir a la potencia (entre 0 y 100 %) a la que lo ponemos tenemos que tener en cuenta la temperatura que hace (entre 0 °C y 40 °C). Tenemos tres modalidades de potencia del calefactor las cuales son alta, media y baja que dependerá de la temperatura a la que podríamos definir en alta, media y baja también.

Las reglas para decidir la potencia son:

- 1) Si la Temperatura es Baja entonces la Potencia es Alta.
- 2) Si la Temperatura es Media entonces la Potencia es Media.
- 3) Si la Temperatura es Alta entonces la Potencia es Baja.

Los conjuntos son:

Temperatura °C (0-40)

- Baja: [0/1, 10/1, 15/0]
- Media: [10/0, 15/1, 25/1, 30/0]
- Alta: [25/0, 30/1, 40/1]

Potencia % (0-100)

- Baja: [0/1, 20/1, 30/0]
- Media: [20/0, 30/1, 60/1, 70/0]
- Alta: [60/0, 70/1, 100/1]

Si tenemos el conjunto de entrada: [0/0, 10/0, 15/1, 20/0, 40/0], ¿A que potencia tenemos que poner el calefactor?

¿Cuál es el resultado final, aplicando la defuzzificación LOM?

Solución

Temperatura = [0/0, 10/0, 15/1, 20/0, 40/0]

Por lo tanto pertenece al conjunto Baja con un valor de 0.5 y al conjunto Media con un valor de 1, entonces los conjuntos de Potencia se cortan en 1 para Media y 0.5 para Alta, dada las reglas, para el conjunto Baja el valor máximo es 0.

Entonces tenemos los conjuntos:

- Baja: [0/0] - Media: [20/0, 30/1, 60/1, 70/0] - Alta: [60/0, 70/0.5, 100/0.5]

Con un conjunto final: **[20/0, 30/1, 60/1, 65/0.5, 100/0.5]**

LOM = Último de los máximos = 60

PROBLEMA 3

Queremos cocinar una pizza en nuestro horno, para ellos tenemos que determinar la temperatura del horno y el tiempo que vamos a tener a la pizza en horno. Combinando estas posibilidades, podemos obtener una pizza poco hecha, muy hecha o en su punto. La temperatura del horno puede ser baja o alta, mientras que el tiempo lo podemos clasificar en poco tiempo o mucho.

Las reglas para determinar cómo va a salir nuestra pizza es:

- 1) Si la Temperatura es Baja y el Tiempo es Mucho entonces la Pizza estará En su punto.
- 2) Si la Temperatura es Baja y el Tiempo es Poco entonces la Pizza estará Poco hecha.
- 3) Si la Temperatura es Alta y el Tiempo es Mucho entonces la Pizza estará Muy hecha.
- 4) Si la Temperatura es Alta y el Tiempo es Poco entonces la Pizza estará En su punto.

Los conjuntos son:

Temperatura °C (160-230)

- Baja: [160/1, 180/1, 200/0]

- Alta: [180/0, 200/1, 230/1]

Tiempo min (5-20)

- Poco: [5/1, 10/1, 12/0]

- Mucho: [10/0, 15/1, 20/1]

Pizza (0-10)

- Poco hecha: [0/1, 3/1, 4/0]

- En su punto: [3/0, 4/1, 6/1, 7/0]

- Muy hecha: [6/0, 7/1, 10/1]

Si nuestras entradas son, tiempo 10 min y temperatura 190 °C, ¿Cómo saldrá nuestra pizza? **(T-Norma del Mínimo)**

¿Cuál es el resultado final, aplicando la defuzzificación LOM?

Solución

Tiempo = 10 min Pertenece a Poco con un valor 1 y a Mucho con un valor 0

Temperatura = 190°C Pertenece a Baja con un valor 0.5 y a Alta con un valor 0.5

Entonces los conjuntos de Pizza se cortan en 0.5 ($1 \wedge 0.5$) para Poco hecha y 0.5 ($1 \wedge 0.5$) para En su punto, dada las reglas, para el conjunto Muy hecha el valor máximo es 0.

Entonces tenemos un conjunto final: **[0/0.5, 6.5/0.5, 7/0]**

LOM = Último de los máximos = 6.5

PROBLEMA 4

Queremos cocinar una pizza en nuestro horno, para ellos tenemos que determinar la temperatura del horno y el tiempo que vamos a tener a la pizza en horno. Combinando estas posibilidades, podemos obtener una pizza poco hecha, muy hecha o en su punto. La temperatura del horno puede ser baja o alta, mientras que el tiempo lo podemos clasificar en poco tiempo o mucho.

Las reglas para determinar cómo va a salir nuestra pizza es:

- 1) Si la Temperatura es Baja y el Tiempo es Mucho entonces la Pizza estará En su punto.
- 2) Si la Temperatura es Baja y el Tiempo es Poco entonces la Pizza estará Poco hecha.
- 3) Si la Temperatura es Alta y el Tiempo es Mucho entonces la Pizza estará Muy hecha.
- 4) Si la Temperatura es Alta y el Tiempo es Poco entonces la Pizza estará En su punto.

Los conjuntos son:

Temperatura °C (160-230)

- Baja: [160/1, 180/1, 200/0]
- Alta: [180/0, 200/1, 230/1]

Tiempo min (5-20)

- Poco: [5/1, 10/1, 12/0]
- Mucho: [10/0, 15/1, 20/1]

Pizza (0-10)

- Poco hecha: [0/1, 3/1, 4/0]
- En su punto: [3/0, 4/1, 6/1, 7/0]
- Muy hecha: [6/0, 7/1, 10/1]

Si nuestras entradas son, tiempo con el conjunto de entrada [10/0, 11/1, 12/0] y temperatura 200 °C, ¿Cómo saldrá nuestra pizza? **(T-Norma del Mínimo)**
¿Cuál es el resultado final, aplicando la defuzzificación SOM?

Solución

Tiempo = [10/0, 11/1, 12/0] Por lo que tenemos el conjunto [10/0, 11.66/0.66, 11/0.5, 11.66/0.66, 12/0] Pertenece a Poco con un valor 0.66 y a Mucho con un valor 0.66
Temperatura = 200°C Pertenece a Baja con un valor 0 y a Alta con un valor 1
Entonces los conjuntos de Pizza se cortan en 0.66 ($1 \wedge 0.66$) para Muy hecha y 0.66 ($1 \wedge 0.66$) para En su punto, dada las reglas, para el conjunto Poco hecha el valor máximo es 0.

Entonces tenemos un conjunto final: **[0/0.66, 6.44/0.66, 7/0]**

SOM = Primero de los máximos = 6.5

PROBLEMA 5

Vamos a realizar un ejercicio de levantamiento de peso y para ello emplearemos un porcentaje de nuestra fuerza, dependiendo de este porcentaje nosotros podremos realizar un determinado número de repeticiones.

Las reglas para determinar tanto la fuerza como el número de repeticiones son:

- 1) Si el Peso es Bajo entonces la Fuerza es Poca.
- 2) Si el Peso es Alto entonces la Fuerza es Mucha.
- 3) Si la Fuerza es Poca entonces el nº de Repeticiones es Muchas.
- 4) Si la Fuerza es Mucha entonces el nº de Repeticiones es Pocas.

Los conjuntos son:

Peso Kg (10-100)

- Bajo: [10/1, 30/1, 60/0]
- Alto: [40/0, 70/1, 100/1]

Fuerza % (0-100)

- Poca: [0/1, 40/1, 60/0]
- Mucha: [40/0, 60/1, 100/1]

Repeticiones (0-20)

- Pocas: [0/1, 8/1, 12/0]
- Muchas: [10/0, 14/1, 20/1]

Si tenemos un peso de 50 kg, ¿Cuántas repeticiones haremos? (**T-Conorma del Máximo**)

¿Cuál es el resultado final, aplicando la defuzzificación MOM?

Solución

Peso = 50 kg Pertenece a Poco con un valor 0.66 y a Mucho con un valor 0.5
Entonces los conjuntos de Fuerza se cortan en 0.66 para Poca y 0.5 para Mucha, dada las reglas. -> [0/0.66, 40.44/0.66, 40.5/0.5, 100/0.5] (Se resuelve como si la entrada difusa fuera el conjunto que acabamos de obtener)

Fuerza = [0/0.66, 40.44/0.66, 40.5/0.5, 100/0.5] Pertenece a Poca con un valor 0.66 y a Mucha con un valor 0.5
Entonces los conjuntos de Repeticiones se cortan en 0.66 para Muchas y 0.5 para Pocas, dada las reglas

Entonces tenemos los conjuntos:

- Pocas: [0/0.5, 0/0.5, 12/0]
- Muchas: [10/0, 12.66/0.66, 20/0.66]

Con un conjunto final: **[0/0.5, 10/0.5, 11/0.25, 12.66/0.66, 20/0.66]**

$$\text{MOM} = \frac{12.66 + 20}{2} = 16.33$$

PROBLEMA 6

Vamos a realizar un ejercicio de levantamiento de peso y para ello emplearemos un porcentaje de nuestra fuerza, dependiendo de este porcentaje nosotros podremos realizar un determinado número de repeticiones.

Las reglas para determinar tanto la fuerza como el número de repeticiones son:

- 1) Si el Peso es Bajo entonces la Fuerza es Poca.
- 2) Si el Peso es Alto entonces la Fuerza es Mucha.
- 3) Si la Fuerza es Poca entonces el nº de Repeticiones es Muchas.
- 4) Si la Fuerza es Mucha entonces el nº de Repeticiones es Pocas.

Los conjuntos son:

Peso Kg (10-100)

- Bajo: [10/1, 30/1, 60/0]
- Alto: [40/0, 70/1, 100/1]

Fuerza % (0-100)

- Poca: [0/1, 40/1, 60/0]
- Mucha: [40/0, 60/1, 100/1]

Repeticiones (0-20)

- Pocas: [0/1, 8/1, 12/0]
- Muchas: [10/0, 14/1, 20/1]

Si tenemos un peso con un conjunto de entrada [30/0, 60/1, 70/0], ¿Cuántas repeticiones haremos? **(T-Conorma del Máximo)**
¿Cuál es el resultado final, aplicando la defuzzificación LOM?

Solución

Peso = [30/0, 60/1, 70/0] Por lo que tenemos el conjunto [40/0, 45/0.4, 50/0.33, 62.5/0.75, 70/0] Pertenece a Bajo con un valor 0.4 y a Alto con un valor 0.75
Entonces los conjuntos de Fuerza se cortan en 0.4 para Poca y 0.75 para Mucha, dada las reglas. -> [0/0.4, 52/0.4, 55/0.75, 100/0.75] (Se resuelve como si la entrada difusa fuera el conjunto que acabamos de obtener)

Fuerza = [0/0.4, 52/0.4, 55/0.75, 100/0.75] Pertenece a Poca con un valor 0.4 y a Mucha con un valor 0.75
Entonces los conjuntos de Repeticiones se cortan en 0.4 para Muchas y 0.75 para Pocas, dada las reglas

Entonces tenemos los conjuntos:

- Pocas: [0/0.5, 0/0.5, 12/0]
- Muchas: [10/0, 12.66/0.66, 20/0.66]

Con un conjunto final: **[0/0.75, 9/0.75, 11/0.25, 11.6/0.4, 20/0.4]**

LOM = Último de los máximos = 9

PROBLEMA 7

Para determinar el desgaste de una rueda influyen muchos factores, como la calidad del asfalto y la potencia de frenado, que esta a su vez es determinada por el estado de la rueda y la velocidad a la que circula el vehículo

Las reglas para determinar el desgaste de la rueda son:

- 1) Si la Velocidad del vehículo es Baja y el Estado de la rueda es Bueno entonces la Potencia de frenado es Baja.
- 2) Si la Velocidad del vehículo es Baja y el Estado de la rueda es Regular o Normal entonces la Potencia de frenado es Media.
- 3) Si la Velocidad del vehículo es Media y el Estado de la rueda es Bueno entonces la Potencia de frenado es Baja.
- 4) Si la Velocidad del vehículo es Media y el Estado de la rueda es Regular o Normal entonces la Potencia de frenado es Media.
- 5) Si la Velocidad del vehículo es Media y el Estado de la rueda es Mala entonces la Potencia de frenado es Alta.
- 6) Si la Velocidad del vehículo es Alta y el Estado de la rueda es Bueno o Normal entonces la Potencia de frenado es Media.
- 7) Si la Velocidad del vehículo es Alta y el Estado de la rueda es Regular o Mala entonces la Potencia de frenado es Alta.
- 8) Si la Potencia de frenado es Baja y la Calidad del asfalto es Buena entonces el Desgaste de la rueda es Ninguno.
- 9) Si la Potencia de frenado es Media y la Calidad del asfalto es Buena entonces el Desgaste de la rueda es Poco.
- 10) Si la Potencia de frenado es Alta y la Calidad del asfalto es Buena entonces el Desgaste de la rueda es Normal.
- 11) Si la Potencia de frenado es Baja y la Calidad del asfalto es Mala entonces el Desgaste de la rueda es Poco.
- 12) Si la Potencia de frenado es Media y la Calidad del asfalto es Mala entonces el Desgaste de la rueda es Normal.
- 13) Si la Potencia de frenado es Alta y la Calidad del asfalto es Mala entonces el Desgaste de la rueda es Mucho.



Los conjuntos son:

Velocidad vehículo Km/h (40-120)

- Baja: [40/1, 50/1, 60/0]
- Media: [50/0, 70/1, 90/0]
- Alta: [80/0, 100/1, 120/1]

Estado Rueda % (0-100)

- Mala: [0/1, 20/1, 30/0]
- Regular: [20/0, 30/1, 40/0]
- Normal: [30/0, 50/1, 80/1, 90/0]
- Buena: [80/0, 90/1, 100/1]

Potencia Frenado HP (10-21)

- Baja: [10/1, 13/1, 14/0]
- Media: [13/0, 15/1, 17/0]
- Alta: [16/0, 18/1, 21/1]

Calidad asfalto % (0-100)

- Malo: [0/1, 40/1, 60/0]
- Bueno: [40/0, 60/1, 100/1]

Desgaste rueda % (0-80)

- Ninguna: [0/1, 20/1, 30/0]
- Poca: [20/0, 30/1, 40/0]
- Normal: [30/0, 50/1, 60/1, 70/0]
- Mucha: [60/0, 70/1, 80/1]

Si tenemos de entrada una velocidad de 55 km/h, el estado de la rueda es de 85 % y la calidad del asfalto es 55 %, ¿Cuántas repeticiones haremos?

(T-Norma del Mínimo y T-Conorma del Máximo)

¿Cuál es el resultado final, aplicando la defuzzificación LOM?

Solución

Velocidad = 55 km/h Pertenece a Baja con un valor 0.5 y a Media con un valor 0.25
Estado Rueda = 85% Pertenece a Normal con un valor 0.5 y a Buena con 0.5
Entonces los conjuntos de Potencia de Frenado se cortan en 0.25 ($0.5 \wedge 0.5 \wedge 0.5 \wedge 0.25$) para Baja y 0.25 ($0.5 \wedge 0.5 \wedge 0.5 \wedge 0.25$) para Media, dada las reglas, para el conjunto Alta hecha el valor máximo es 0. -> [10/0.25, 16.6/0.25, 17/0] (Se resuelve como si la entrada difusa fuera el conjunto que acabamos de obtener)

Calidad de Frenado = 55% Pertenece a Bueno con un valor 0.25 y a Malo con 0.75
Potencia de Frenado = [10/0.25, 16.6/0.25, 17/0] Pertenece a Baja con un valor 0.25 y a Media con un valor 0.25

Entonces los conjuntos de Desgaste se cortan en 0.25 ($0.25 \wedge 0.25$) para Ninguno, 0.25 ($0.25 \wedge 0.75 \wedge 0.25 \wedge 0.25$) para Poca y 0.25 ($0.25 \wedge 0.75$) para Normal, dada las reglas, para el conjunto Poca hecha el valor máximo es 0.

Entonces tenemos los conjuntos:

- Ninguna: [0/0.25, 22.5/0.25, 30/0] - Poca: [20/0, 22.5/0.25, 37.5/0.25, 30/0]
- Normal: [30/0, 35/0.25, 62.5/0.25, 70/0]

Con un conjunto final: **[0/0.25, 62.5/0.25, 70/0]**

LOM = Último de los máximos = 62.5

PROBLEMA 8

Para determinar el desgaste de una rueda influyen muchos factores, como la calidad del asfalto y la potencia de frenado, que esta a su vez es determinada por el estado de la rueda y la velocidad a la que circula el vehículo

Las reglas para determinar el desgaste de la rueda son:

- 1) Si la Velocidad del vehículo es Baja y el Estado de la rueda es Bueno entonces la Potencia de frenado es Baja.
- 2) Si la Velocidad del vehículo es Baja y el Estado de la rueda es Regular o Normal entonces la Potencia de frenado es Media.
- 3) Si la Velocidad del vehículo es Media y el Estado de la rueda es Bueno entonces la Potencia de frenado es Baja.
- 4) Si la Velocidad del vehículo es Media y el Estado de la rueda es Regular o Normal entonces la Potencia de frenado es Media.
- 5) Si la Velocidad del vehículo es Media y el Estado de la rueda es Mala entonces la Potencia de frenado es Alta.
- 6) Si la Velocidad del vehículo es Alta y el Estado de la rueda es Bueno o Normal entonces la Potencia de frenado es Media.
- 7) Si la Velocidad del vehículo es Alta y el Estado de la rueda es Regular o Mala entonces la Potencia de frenado es Alta.
- 8) Si la Potencia de frenado es Baja y la Calidad del asfalto es Buena entonces el Desgaste de la rueda es Ninguno.
- 9) Si la Potencia de frenado es Media y la Calidad del asfalto es Buena entonces el Desgaste de la rueda es Poco.
- 10) Si la Potencia de frenado es Alta y la Calidad del asfalto es Buena entonces el Desgaste de la rueda es Normal.
- 11) Si la Potencia de frenado es Baja y la Calidad del asfalto es Mala entonces el Desgaste de la rueda es Poco.
- 12) Si la Potencia de frenado es Media y la Calidad del asfalto es Mala entonces el Desgaste de la rueda es Normal.
- 13) Si la Potencia de frenado es Alta y la Calidad del asfalto es Mala entonces el Desgaste de la rueda es Mucho.



Los conjuntos son:

Velocidad vehículo Km/h (40-120)

- Baja: [40/1, 50/1, 60/0]
- Media: [50/0, 70/1, 90/0]
- Alta: [80/0, 100/1, 120/1]

Estado Rueda % (0-100)

- Mala: [0/1, 20/1, 30/0]
- Regular: [20/0, 30/1, 40/0]
- Normal: [30/0, 50/1, 80/1, 90/0]
- Buena: [80/0, 90/1, 100/1]

Potencia Frenado HP (10-21)

- Baja: [10/1, 13/1, 14/0]
- Media: [13/0, 15/1, 17/0]
- Alta: [16/0, 18/1, 21/1]

Calidad asfalto % (0-100)

- Malo: [0/1, 40/1, 60/0]
- Bueno: [40/0, 60/1, 100/1]

Desgaste rueda % (0-80)

- Ninguna: [0/1, 20/1, 30/0]
- Poca: [20/0, 30/1, 40/0]
- Normal: [30/0, 50/1, 60/1, 70/0]
- Mucha: [60/0, 70/1, 80/1]

Si tenemos de entrada una velocidad con un conjunto [70/0, 80/1, 85/1, 100/0], el estado de la rueda es de 20% y la calidad del asfalto es 42%, ¿Cuántas repeticiones haremos?

(T-Norma del Mínimo y T-Conorma del Máximo)

¿Cuál es el resultado final, aplicando la defuzzificación SOM?

Solución

Velocidad = [70/0, 80/1, 85/1, 100/0] Pertenece a Alta con un valor 0.5 y a Media con un valor 0.66

Estado Rueda = 20% Pertenece a Mala con un valor 1

Entonces los conjuntos de Potencia de Frenado se cortan en 0.5 ($1 \wedge 0.66 \wedge 0.5 \wedge 1$) para Alta, dada las reglas, para el conjunto Baja y Media hecha el valor máximo es 0. -> [16/0, 17/0.5, 21/0.5] (Se resuelve como si la entrada difusa fuera el conjunto que acabamos de obtener)

Calidad de Frenado = 42% Pertenece a Bueno con un valor 0.1 y a Malo con 0.9

Potencia de Frenado = [16/0, 17/0.5, 21/0.5] Pertenece a Alta con un valor 0.5

Entonces los conjuntos de Desgaste se cortan en 0.1 ($0.5 \wedge 0.1$) para Normal, 0.5 ($0.5 \wedge 0.9$) para Mucho, dada las reglas, para el conjunto Poco y Ninguna hecha el valor máximo es 0.

Entonces tenemos los conjuntos:

- Normal: [30/0, 20/0.1, 69/0.1, 70/0] - Mucho: [60/0, 65/0.5, 80/0.5]

Con un conjunto final: **[30/0, 20/0.1, 61/0.1, 65/0.5, 80/0.5]**

SOM = Primero de los máximos = 65

TEST 1

Las respuestas erróneas de este test NO restan.

1) ¿Cuántas funciones de pertenencia existen?

- a) 4
- b) 7
- c) **Infinitas**
- d) Ninguna es correcta

2) Las T-conormas más habituales son (Unión)

- a) Mínimo, Producto y Producto Drástico
- b) Máximo, Producto y Producto Drástico
- c) Mínimo, Suma y Suma Drástica
- d) **Máximo, Suma y Suma Drástica**

3) ¿Cuál es la función del complemento que podríamos definir como $C(A) = 1-A$?

- a) **Complementario**
- b) Inverso
- c) Complemento de Sugeno
- d) Complemento de Yager

4) Un hecho difuso compuesto es la concatenación de más de dos hechos difusos simples

- a) Verdadero
- b) **Falso**

5) El modus ponens generalizado es la forma en la que se realiza la inferencia a partir de hechos difusos y reglas difusas

- a) **Verdadero**
- b) Falso

6) Las siglas del método de la media de los valores de máxima pertenencia es:

- a) SOM
- b) COG
- c) **MOM**
- d) LOM

7) Las siglas del método del centroide es:

- a) SOM
- b) **COG**
- c) MOM
- d) LOM

8) Las reglas del sistema de tipo Takasi-Sugeno (TS) se diferencian de las reglas de tipo Mamdani en que tienen un consecuente nítido

- a) Verdad**
- b) Falso

9) El conjunto $[0/0, 1/0, 2/0, 3/1, 4/1, 5/1]$ es un conjunto de:

- a) La lógica clásica
- b) La lógica difusa
- c) Ambas lógicas**
- d) Ninguna de estas lógicas

10) Se entiende por razonamiento difuso el proceso de realizar inferencias a partir de hechos difusos, así como de combinar evidencias y actualizar el grado de las creencias

- a) Verdad**
- b) Falso

TEST 2

Las respuestas erróneas de este test SI restan.

1) ¿Son las funciones S, Z y PI la versión cuadrática de las funciones GAMMA, L y LAMBDA respectivamente?

- a) Verdad
- b) Falso**

2) ¿Cuáles pertenecen a las T-normas (Intersección)

- a) Máximo
- b) Mínimo**
- c) Producto**
- d) Suma
- e) Producto Drástico**
- f) Suma Drástica
- g) Complementario
- h) Complementario de Sugeno

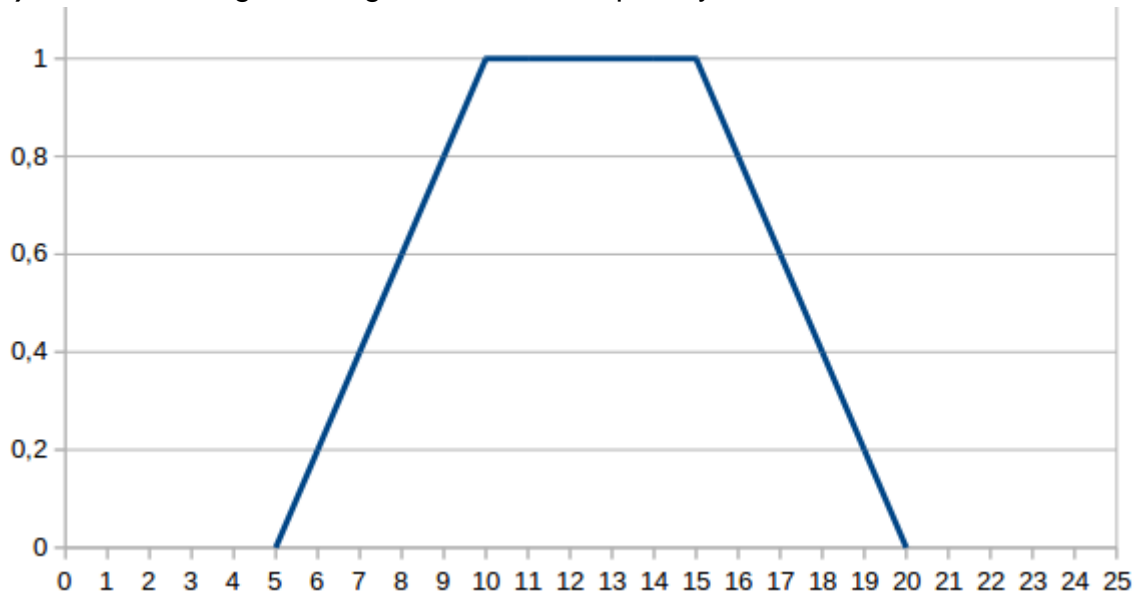
3) ¿Cuál de los siguientes hechos difusos compuesto está mal formado?

- a) $\neg P$
- b) $Q \wedge P$
- c) $\neg Q \vee \neg P$
- d) $P \neg \wedge \neg Q$**
- e) $(Q \vee (P \wedge (R \vee S))) \wedge T$
- f) $((\neg Q \wedge \neg P) \vee ((R \wedge S) \vee \neg T)) \vee \neg U$

4) La posibilidad de A dado B es $POS(A|B)=$

- a) $\max(A \cup B)$
- b) $\min(A \cup B)$
- c) $\max(A \cap B)$**
- d) $\min(A \cap B)$

5) Teniendo la siguiente figura, indicar el soporte y el núcleo



- a) Soporte [0,5] y Núcleo [10,15]
- b) Soporte [5,20] y Núcleo [10,20]
- c) Soporte [5,10] y Núcleo [10,15]
- d) Soporte [10,15] y Núcleo [5,20]
- e) Soporte [5,20] y Núcleo [10,15]**
- f) Soporte [5,25] y Núcleo [5,20]

6) El resultado de un razonamiento difuso puede ser un conjunto

- a) Verdad
- b) Falso**

7) Selecciona las denominaciones del proceso de obtener un valor nítido a partir de un conjunto difuso

- a) Decodificación**
- b) Aproximación
- c) Clarificación
- d) Defuzzificación**
- e) Determinación
- f) Representación
- g) Nitificación**
- h) Valor Representativo

8) Los métodos LOM y SOM corresponden con el menor de los máximos y el mayor de los máximos correspondientemente

- a) Verdad
- b) Falso**

9) En que se diferencian las reglas del sistema de tipo Takasi-Sugeno (TS) de las de tipo Mamdani?

- a) Las de tipo Mamdani tienen un consecuente nítido**
- b) Las de tipo TS tienen un antecedente nítido
- c) Las de tipo Mamdani tienen un antecedente nítido
- d) Las de tipo TS tienen un consecuente nítido

10) La lógica difusa deriva de la lógica tradicional, pero se diferencian en:

- a) La lógica tradicional es más cercana a la forma de pensamiento humano
- b) La lógica difusa se basa en que una afirmación no tiene por qué ser cierta ni falsa**
- c) La a) y la b) son verdaderas
- d) Ninguna de las anteriores

B Documento de análisis de requisitos

1. Introducción

1.1. Propósito

El propósito de este proyecto es crear una aplicación web (llamada FuzzyTeacher) en la que, tanto usuario con conocimiento de lógica difusa como usuario sin dicho conocimiento, puedan ampliar y poner a prueba sus conocimientos. Esta aplicación está pensada para usuarios autodidactas (por el hecho de mostrar el resultado de los problemas al momento) pero puede ser usada en la docencia sin ningún inconveniente.

Este documento está dirigido para todos los desarrolladores que estén interesados en trabajar o ampliar este proyecto.

1.2. Ámbito del Sistema

La aplicación se pretende usar en el ámbito de la enseñanza como ya se ha comentado. Para hacerlo más interactivo, favoreciendo el aprendizaje, se utilizan ejemplos, ejercicios y problemas.

Dicha aplicación está al alcance de todos ya que es gratuita.

La aplicación consta de un temario con una serie de ejemplos. Para poner a prueba los conocimientos contaremos con ejercicios, problemas y dos test. En todos los casos se muestra si la solución es correcta o no.

Tanto los test como los problemas no son dinámicos, es decir, no hay un repertorio de problemas y se muestran aleatoriamente, son siempre los mismos y no se podrán incluir nuevos problemas con sus soluciones.

1.3. Referencias

- Arredondo, T. (2014). *Introducción a la lógica difusa*. Recuperado de: <http://profesores.elo.utfsm.cl/~tarredondo/info/soft-comp/Introduccion%20a%20la%20Logica%20Difusa.pdf>
- González, C. (2011). *Lógica difusa*. Recuperado de: http://www.esi.uclm.es/www/cglez/downloads/docencia/2011_Softcomputing/LogicaDifusa.pdf
- Millán E. y Pérez de la Cruz J. L. (2014). *Apuntes de introducción al razonamiento difusa*. Dpto. de Lenguajes y Ciencias de la Computación. Universidad de Málaga
- Olmo, M. A. (2008). *Tutorial de introducción de lógica borrosa*. Recuperado de: http://www.dma.fi.upm.es/recursos/aplicaciones/logica_borrosa/web/tutorial_fuzzy/indice.html
- *Especificación de Requisitos según el estándar de IEEE 830* (s.d.) (2008) <https://www.fdi.ucm.es/profesor/gmendez/docs/is0809/ieee830.pdf>

1.4. Visión General del Documento

El documento consta de tres bloques diferenciados, uno es la introducción que acabamos de ver y los otros dos tratan de explicar el producto más detalladamente, el primero de estos, de forma general, dando datos de las funciones y características al igual que las restricciones que tiene, mientras que el último bloque se especifica de forma técnica los datos y requisitos de este proyecto.

2. Descripción General

2.1. Funciones del Producto

La aplicación al estar orientada a la enseñanza consta de un temario. Para facilitar la adquisición de conocimientos se dispone de una serie de ejercicios introducidos en el temario. La aplicación consta de dos bloques para poner a prueba los conocimientos adquiridos que son: problemas y tests.

Finalmente los usuarios podrán llevar un control de los temas leídos y de los ejercicios realizados.

2.2. Características de los Usuarios

Al ser una aplicación web el usuario necesita disponer de acceso a internet.

Por otra parte, al ser una aplicación educativa el usuario no requiere de alto conocimiento sobre el temario. Pero si es aconsejable tener un conocimiento básico sobre lógica tradicional.

2.3. Restricciones

La principal restricción de este proyecto es que dado que la aplicación está desplegada en la web se debe disponer de conexión web

2.4. Suposiciones y Dependencias

El lenguaje de programación usado para este proyecto es Python con el framework Django ya que para este tipo de aplicación es muy cómodo y fácil de manejar, además de ser un lenguaje que cada vez es más usado, otro punto a favor de haber elegido este lenguaje en lugar de otro como Java es que es más lento para la web.