

Capítulo II - Metodologia para simulação da Lógica Fuzzy através do SCILAB SciFLTEditor

Fábio Sampaio⁴
Matheus Cuenca⁵
Ruan Rodrigues⁶
Prof. Esp. Wesley Candido da Silva⁷

RESUMO

Para classificar uma comida, é usual utilizar-se de adjetivos como “péssimo, digestível, deliciosa, boa ou muito boa”; de maneira geral para comunicar a impressão sobre as coisas, mas em sistemas inteligentes, precisamos transformar esses conceitos em uma escala, para que se possa aplicar funções de pertinência, geralmente em números reais, e com isso, pode-se perder informações importantes, que estão embutidas nesses adjetivos. Este artigo tem como premissa principal, contribuir para comunidade acadêmica e egressos com conhecimentos matemáticos, científicos e tecnológicos aplicados à engenharia, fazendo uso do de uma metodologia para simulação da lógica *Fuzzy* através do software “*Scilab*” e sua extensão “*sciFLT*”, utilizando como exemplo a definição do valor de uma gorjeta, tendo como base a nota atribuída pelo cliente em relação a qualidade da comida e a qualidade dos serviços prestados.

Palavras-chave: Lógica Fuzzy. Scilab. sciFLT.

⁴ Graduando – Engenharia Elétrica, da Faculdade de Tecnologia do Senai Londrina.

⁵ Graduando – Engenharia Elétrica, da Faculdade de Tecnologia do Senai Londrina.

⁶ Graduando – Engenharia Elétrica, da Faculdade de Tecnologia do Senai Londrina.

⁷ Docente da Faculdade da Indústria SENAI Londrina. E-mail: wesley.candido@sistemafiep.org.br

ABSTRACT

To classify a food, it is very common to use adjectives like “awful, edible, delicious”, this is used in general to communicate our impression of things, but in intelligent systems, we need to transform these concepts into numbers, to apply functions of classification. pertinence, usually in real numbers, and with that, we end up losing important information, which is embedded in these adjectives. This article's main premise is to contribute to the academic community and graduates mathematical, scientific and technological knowledge applied to engineering, making use of a methodology for simulating Fuzzy logic through the software "Scilab" and its extension "sciFLT", using as an example the definition of the value of a tip, based on the grade given by the customer in relation to the quality of the food and the quality of the services provided.

Keywords: Logic Fuzzy. Scilab. sciFLT.

1. INTRODUÇÃO

A lógica Difusa ou *Fuzzy*; proposta em 1965 por **Lofti Zadeh** a partir da Teoria dos Conjuntos Difusos, a Lógica Difusa preenche uma lacuna entre a comunicação humana e os sistemas computacionais.

Com esta lógica pode-se através das funções de pertinência, conseguimos mensurar em valores a esses adjetivos. Sendo assim, podemos modelar o nosso sistema através dos adjetivos, também conhecida como difusa ou nebulosa. tem por objetivo modelar modos de raciocínio aproximados ao invés de precisos, trazendo a inteligência humana para a arquitetura computacional.

Na lógica computacional (binário) as proposições são unicamente “verdadeiras” ou “falsas”, diferente da lógica Difusa onde as proposições podem ter valores intermediários entre “verdadeiro” e “falso”, ou seja, a veracidade dessas pode assumir

qualquer valor entre 0 e 1, considerando que as sentenças passam a ter um grau de pertinência.

A lógica *Fuzzy* por sua vez, visa modelar modos de raciocínio imprecisos, tendo os casos precisos como situações limite. No mundo real, os problemas muitas vezes não conseguem ser representados pela lógica clássica, tendo em vista que os critérios de pertinência são apenas “pertence” ou “não pertence”, ou seja, um elemento não pode pertencer parcialmente a um conjunto, da mesma forma que um conjunto não pode estar parcialmente contido em outro.

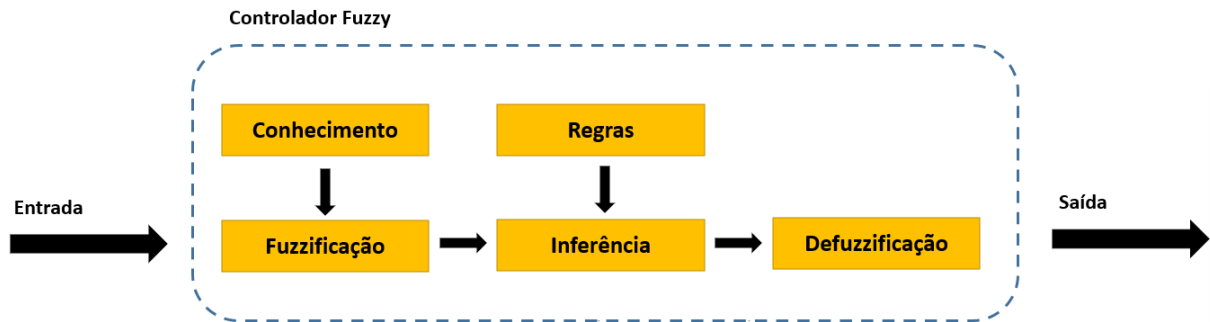
O grau de pertinência na lógica *Fuzzy* possibilita uma transição gradual da falsidade para a verdade, nos permitindo representar valores imprecisos. Muitas vezes é utilizado uma discretização dos possíveis valores para um domínio lógico de múltiplos valores, por exemplo: [0 - 0,5 - 1] para valores que indiquem “Falso”, “Talvez Verdadeiro” e “Verdadeiro” respectivamente, dessa forma a lógica Difusa nos ajuda a modelar modos de raciocínios imprecisos, tendo os casos precisos como situações limites.

“As pessoas funcionam de modo “vago”, ao invés de no modo “verdadeiro ou falso”. - Conceito de Charles Sanders Peirce em 1870.

2. METODOLOGIA

A fuzzificação está relacionada ao conhecimento do seu processo, nesta etapa, se faz necessário a conversão dos valores de entrada (sensores, grandezas físicas, condições de um processo) em variáveis linguísticas. Passando para a próxima etapa, de inferência, onde relaciona as regras de decisão com as nossas entradas e saídas da nossa lógica. A *Defuzzificação* é o processo para conversão dos resultados obtidos em resultados nítidos. A figura 1, mostra um fluxograma da lógica *Fuzzy*.

Figura 1 – Logica *Fuzzy*



Fonte: Dos Autores.

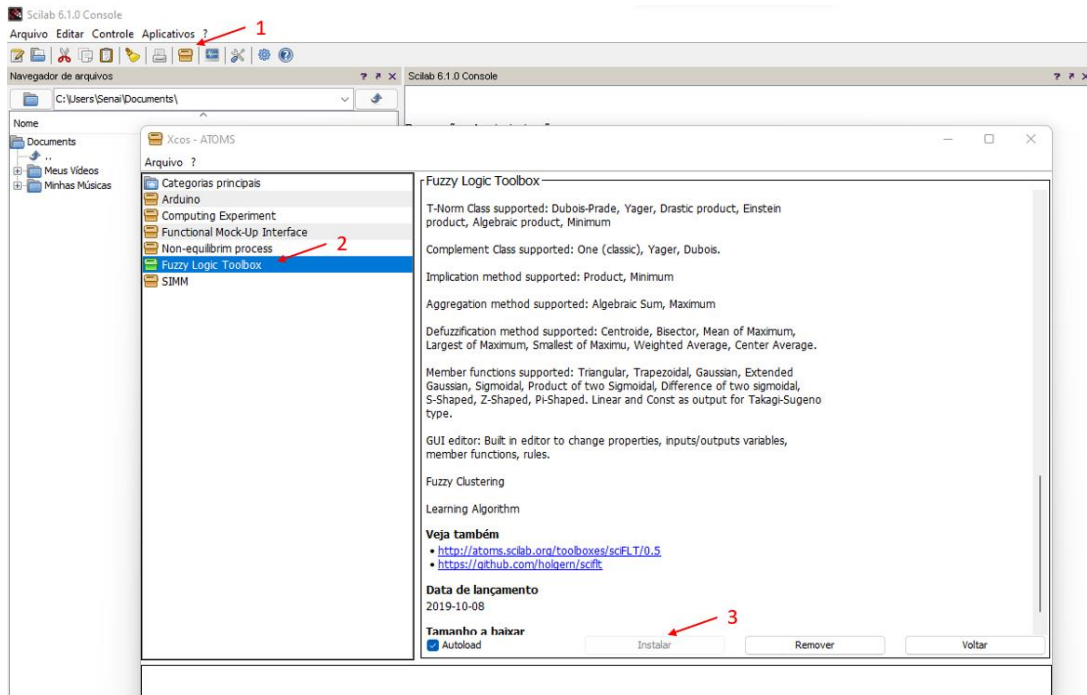
Para explanação do algoritmo envolvendo a lógica *Fuzzy*, será usado um problema no qual consiste em definir o valor da gorjeta em um restaurante com base na qualidade do serviço prestado e da comida, para o processamento do algoritmo será utilizado a ferramenta computacional *Scilab*⁸; que por sua vez é um software gratuito, de código aberto e multiplataforma orientado à computação científica. O *Scilab* possibilita a manipulação de matrizes de maneira simples e direta, permitindo a resolução de problemas difíceis aplicando métodos numéricos.

Após a instalação do software, se faz necessário configurar a ferramenta computacional para trabalhar com a lógica *Fuzzy*, para isso uma extensão do programa deverá ser acrescentada.

Na figura 2, são indicados três passos, é referente ao acesso do gerenciador de módulos - *ATOMS*, após o acesso ao gerenciador de módulos, localizar a extensão *Fuzzy Logic Toolbox* e por fim instalar a extensão *SciFLTEditor*.

Figura 2 – Instalação da extensão *SciFLTEditor*

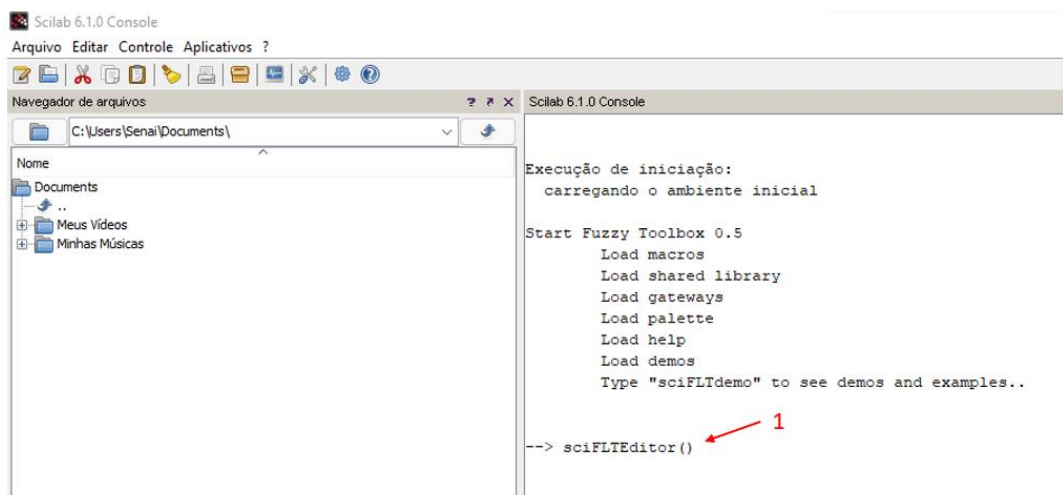
⁸ Disponível em: <https://www.scilab.org/>



Fonte: Dos Autores.

Com a extensão instalada, é necessário iniciar o *SciFLTEditor*, para isso no console de script do *Scilab*, executar o comando conforme indicado na figura 3, dessa forma será habilitada a extensão *SciFLTEditor*.

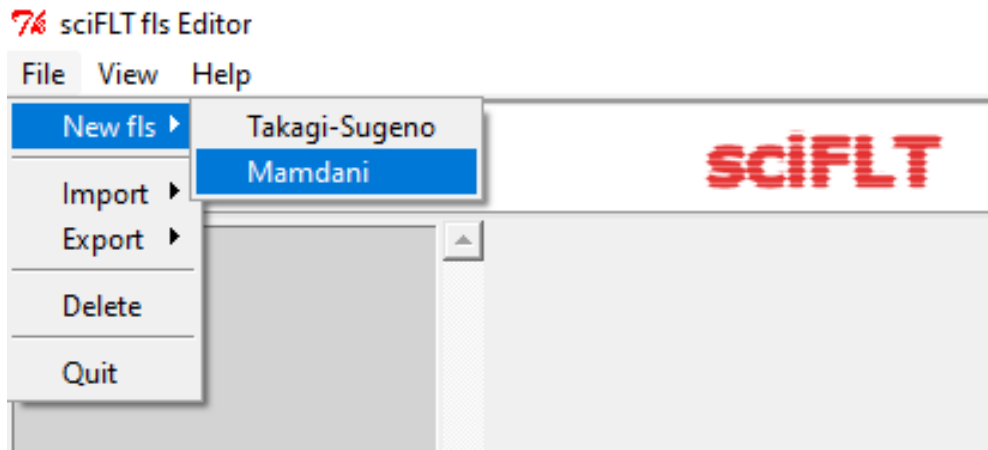
Figura 3 – Abrindo a extensão *SciFLTEditor*



Fonte: Dos Autores.

Com a extensão *SciFLTEditor* aberta, é necessário iniciar um novo projeto (*new fls*) conforme indicado na figura 4.

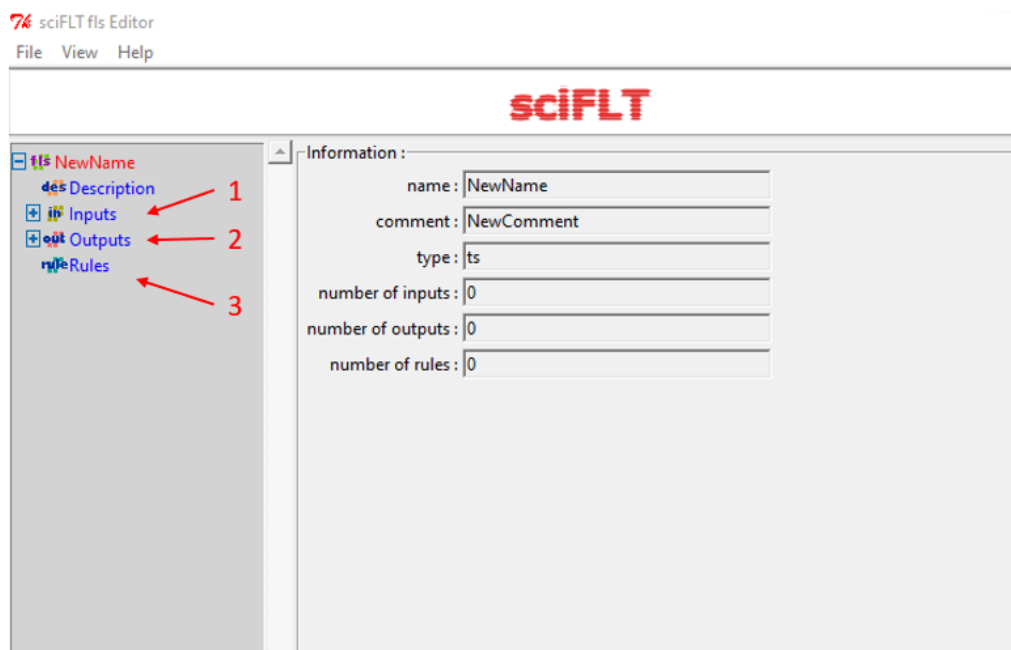
Figura 4 – Novo projeto



Fonte: Dos Autores.

Com o projeto aberto, se faz necessário definir os parâmetros de entradas, saídas, regras de inferência e funções de pertinência, indicado na figura 5.

Figura 5 – Informações do projeto



Fonte: Dos Autores.

2.1. Entradas

Uma das etapas da lógica *Fuzzy* e o processo de Fuzzificação, que consiste transformar as entradas do sistema em uma variável linguística, correlacionando as entradas a um conjunto de valores, a figura 6 indica o passo necessário para adicionar variáveis de entradas.

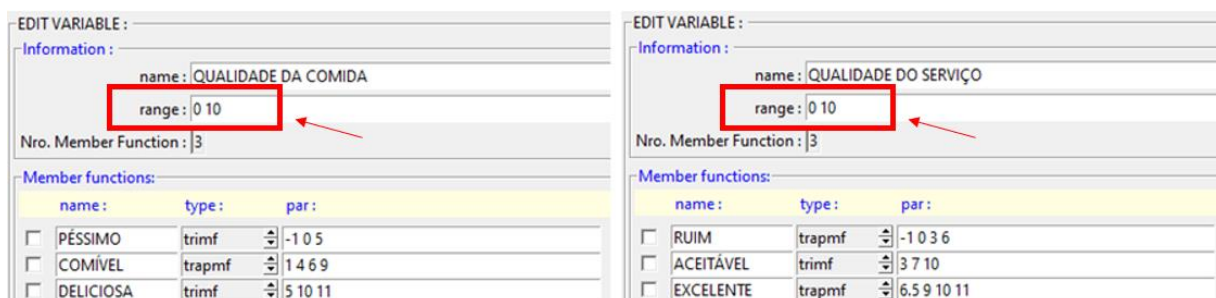
Figura 6 – Adicionando entradas



Fonte: Dos Autores.

Para cada entrada é necessário configurar o parâmetro do range de atuação, composto por um intervalo de valores nítidos, dessa forma podemos mensurar as entradas “*qualidade de comida*” e “*qualidade do serviço*” em um range de 0 a 10.

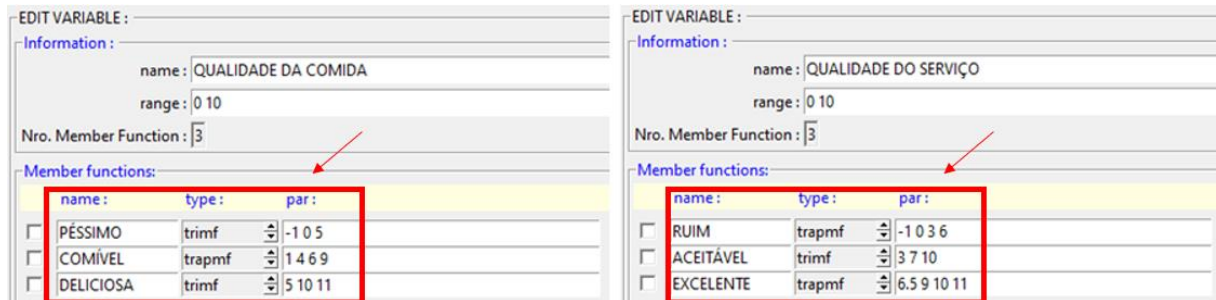
Figura 7 – Intervalo de valores nítidos



Fonte: Dos Autores.

A partir de uma base de conhecimento do problema proposto, para cada entrada criada é atribuído um subconjunto, denominado *conjunto difuso*, tendo como parâmetro, o range de atuação e a função de pertinência, apresentado na figura 8.

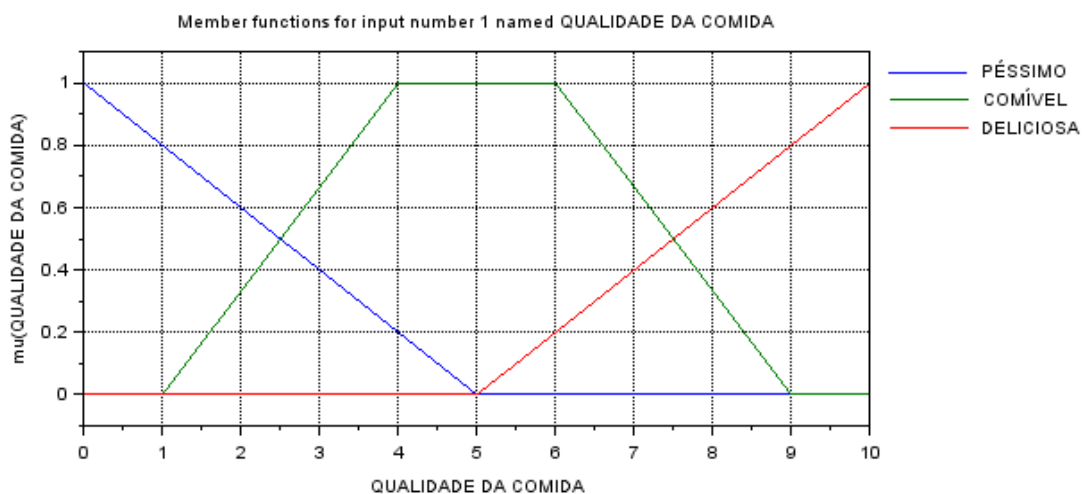
Figura 8 – Intervalo de valores conjunto difuso



Fonte: Dos Autores.

É possível observar que na entrada “*qualidade de comida*” foi criado três subconjuntos denominado “péssimo”, “comível” e “deliciosa”, sendo atribuído a cada conjunto difuso uma função de pertinência *trimf* (função triangular) e *trapmf* (função trapezoidal). Para cada função de pertinência é determinado um conjunto de valores no qual mensuramos seu grau de contribuição; e quantidade de pontos a serem inseridos no subconjunto, por exemplo uma função triangular são necessários três pontos enquanto a trapezoidal contempla quatro pontos.

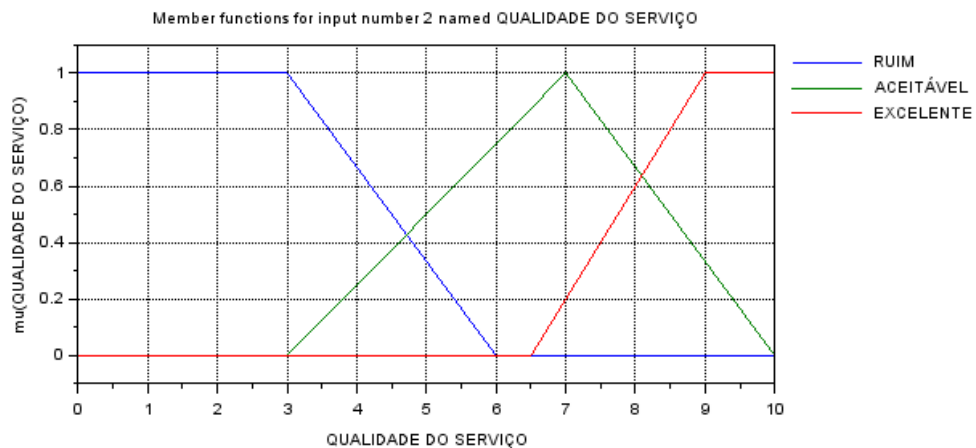
Figura 9 – Conjunto difuso “qualidade da comida”



Fonte: Dos Autores.

No eixo x do gráfico da figura 9 apresenta o range [0 a 10] da entrada “qualidade da comida”, já no eixo y temos o grau de pertinência da função, que compreende o range de [0 a 1], O mesmo contexto é aplicado para a entrada “qualidade de serviço” conforme a figura 10.

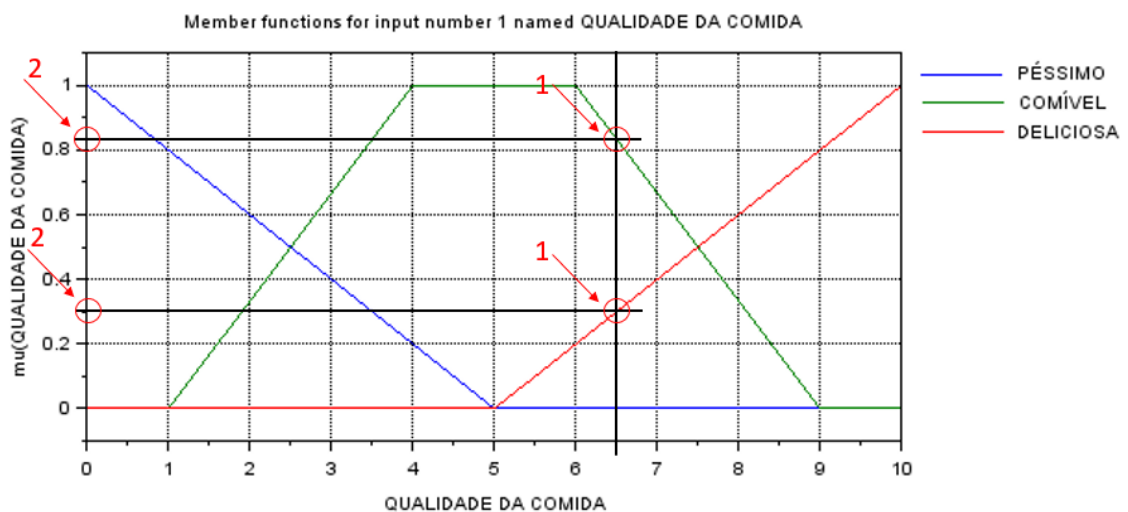
Figura 10 – Conjunto difuso “qualidade do serviço”



Fonte: Dos Autores.

Para elucidarmos o contexto do conjunto difuso, usamos a entrada “qualidade de comida” como exemplo; foi traçada uma reta no instante 6,5 no qual cruzou as funções de pertinência trapezoidal “comível” e a triangular “deliciosa”, rebatendo o cruzamento para o eixo y é possível notar o grau de pertinência que cada função terá, conforme apresentado na figura 11.

Figura 11 – Instante 6,5 do conjunto difuso “qualidade da comida”



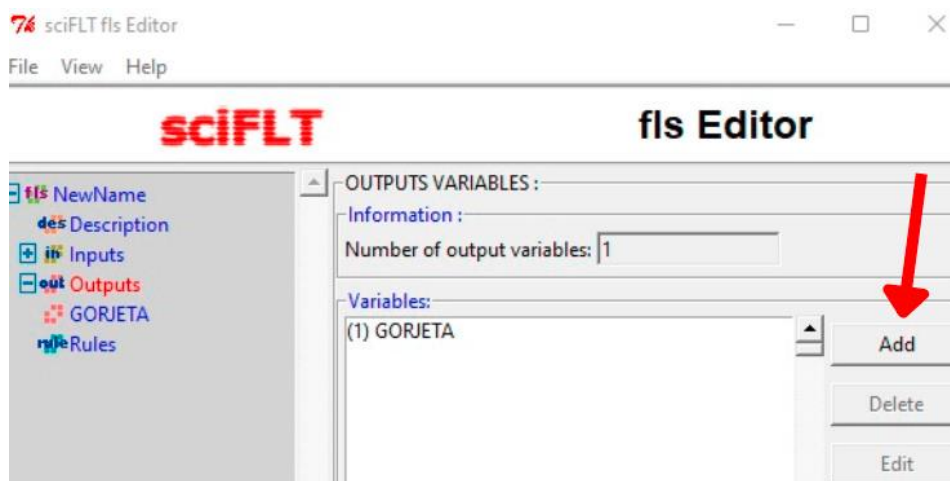
Fonte: Dos Autores.

Dessa forma observamos que para uma nota de 6,5 temos uma contribuição dos conjuntos difusos “*comível*” e “*deliciosa*”, sendo que o grau de pertinência de “*comível*” é em torno de 85%, enquanto “*deliciosa*” é 30%, os graus de pertinência obtidos irão refletir na minha saída para o processo de *Defuzzificação*.

2.2. Saída

Com as entradas configuradas no *SciFLTEditor*, é necessário definirmos uma saída, para isso inicialmente devemos adicionar *outputs variables*, de acordo com a figura 12.

Figura 12 – Adicionando variáveis de saída no SciFLTEditor



Fonte: Dos Autores

A variável linguística “*gorjeta*” foi criada, o conjunto do valor de saída abrange [0 a 25], no qual se refere ao valor monetário da gorjeta. Para os subconjuntos difuso da saída foi determinado como “*alta*”, “*média*” e “*baixa*”, com a função de pertinência do tipo triangular *trimf*.

Figura 13 – Parâmetros da variável de saída do SciFLTEditor

EDIT VARIABLE :

Information :

name : GORJETA

range : 0 25

Nro. Member Function : 3

Member functions:

name :	type :	par :
<input type="checkbox"/> BAIXA	trimf	-1 0 12.5
<input type="checkbox"/> MÉDIA	trimf	0 12.5 25
<input type="checkbox"/> ALTA	trimf	12.5 25 26

Variável linguística da saída.

Range da saída.

Função de pertinência triangular.

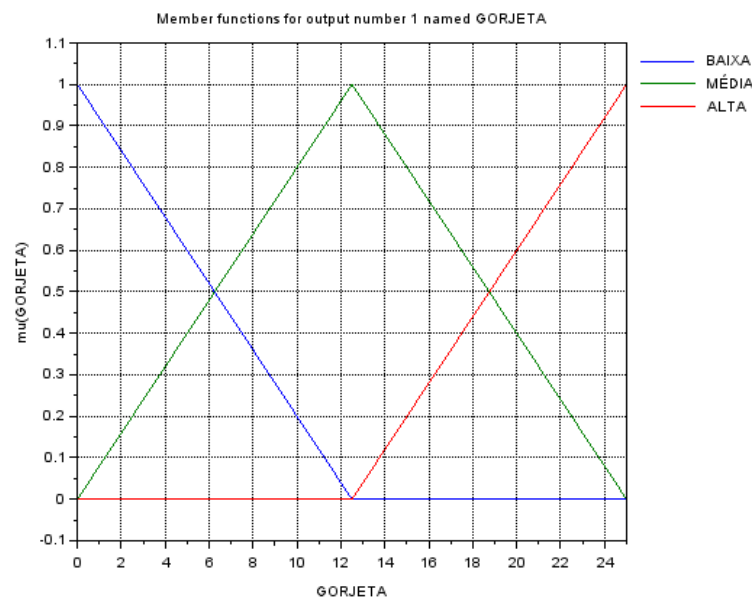
Subconjunto difuso.

Variável linguística do subconjunto difuso.

Fonte: Dos Autores.

No gráfico da figura 14 temos plotado a variável de saída, conforme parâmetros apresentados na figura 13, no processo de *Defuzzificação*, a saída irá obedecer a estratégia denominada *Centroide*⁹.

Figura 14 – Gráfico da variável de saída do SciFLTEditor



⁹ Centroide é o ponto associado a uma forma geométrica no qual representa o centro da massa.

Fonte: Dos Autores.

2.3. Regras de decisão

Para correlacionar as entradas com a saída, regras são necessárias. A figura 14 nos indica dois passos, o primeiro deles é necessário para definir as regras de decisão, e por fim, adicionamos a regra para que seja aplicada na lógica *Fuzzy*.

Figura 14 – Gráfico da variável de saída do SciFLTEditor



Fonte: Dos Autores.

As regras estão relacionadas a algo específico do seu problema. Se uma determinada ação for correspondida, a saída estará de acordo com a sua regra criada para ela. Com base na nossa análise do problema, definimos as seguintes regras de decisão:

Regra 1: **SE** o serviço for excelente **E** a comida estava deliciosa **ENTÃO** a gorjeta deve ser alta.

Regra 2: **SE** o serviço for aceitável **ENTÃO** a gorjeta deve ser média.

Regra 3: **SE** o serviço for ruim **OU** a comida estava comível **ENTÃO** a gorjeta deve ser baixa.

Regra 4: **SE** o serviço for excelente **E** a comida estava péssima **ENTÃO** a gorjeta deve ser média.

O modelo pode contemplar a quantidade de regras necessária para se aproximar da base de conhecimento existente do problema. A figura 15 representa na extensão *SciFLTEditor*, as regras de decisão citadas acima.

Figura 15 – Regras da Lógica Fuzzy

```

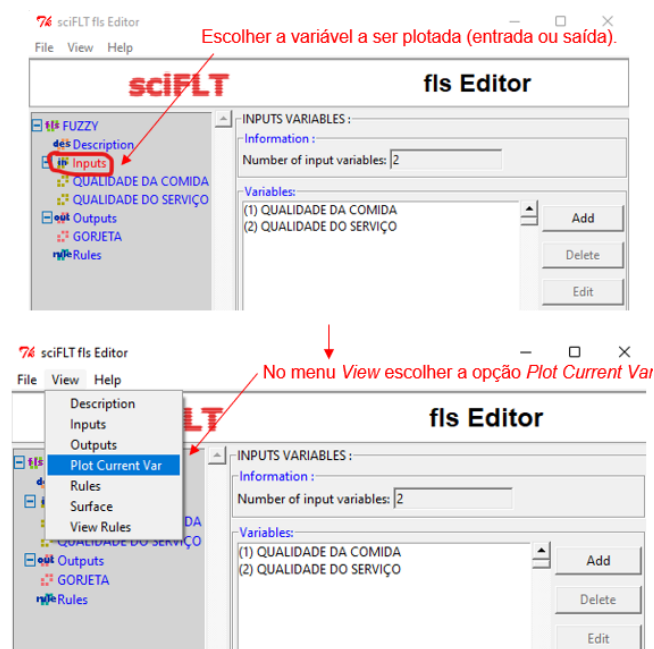
RULE EDITOR
R1: IF {QUALIDADE DA COMIDA IS DELICIOSA} AND {QUALIDADE DO SERVIÇO IS EXCELENTE} THEN {GORJETA IS ALTA} weighth=1.0
R2: IF {QUALIDADE DO SERVIÇO IS ACEITÁVEL} THEN {GORJETA IS MÉDIA} weighth=1.0
R3: IF {QUALIDADE DA COMIDA IS COMÍVEL} OR {QUALIDADE DO SERVIÇO IS RUIM } THEN {GORJETA IS BAIXA} weighth=1.0
R4: IF {QUALIDADE DA COMIDA IS PÉSSIMO} AND {QUALIDADE DO SERVIÇO IS EXCELENTE} THEN {GORJETA IS MÉDIA} weighth=1.0

```

Fonte: Dos Autores.

Ao longo do artigo foram apresentados diversos gráficos que representam as variáveis de entrada e saída. Para fazer essa plotagem, se faz necessário selecionar a variável e em seguida acessar a ferramenta *Plot Current Var*, a figura 16 mostra os passos necessários.

Figura 16 – Plotagem de gráficos

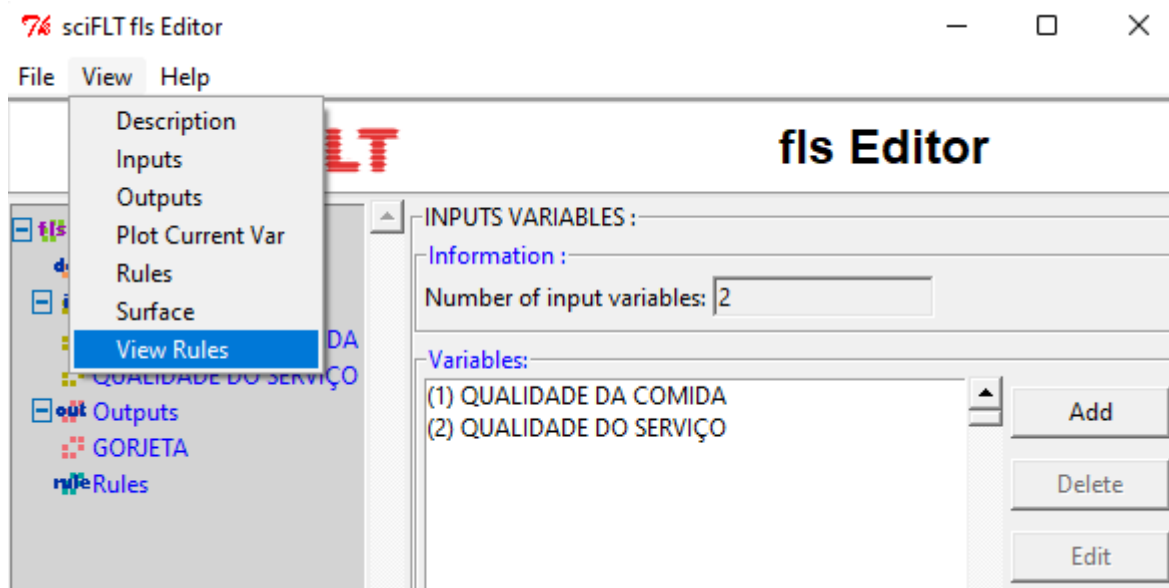


Fonte: Dos Autores.

3. Resultados

Com a definição dos parâmetros das entradas, saídas e regras; o próximo passo é analisar a saída, para isso deve ser acessada a ferramenta *View Rules*, conforme figura 17, para essa simulação, se faz necessário estar selecionado a variável “*Inputs*”.

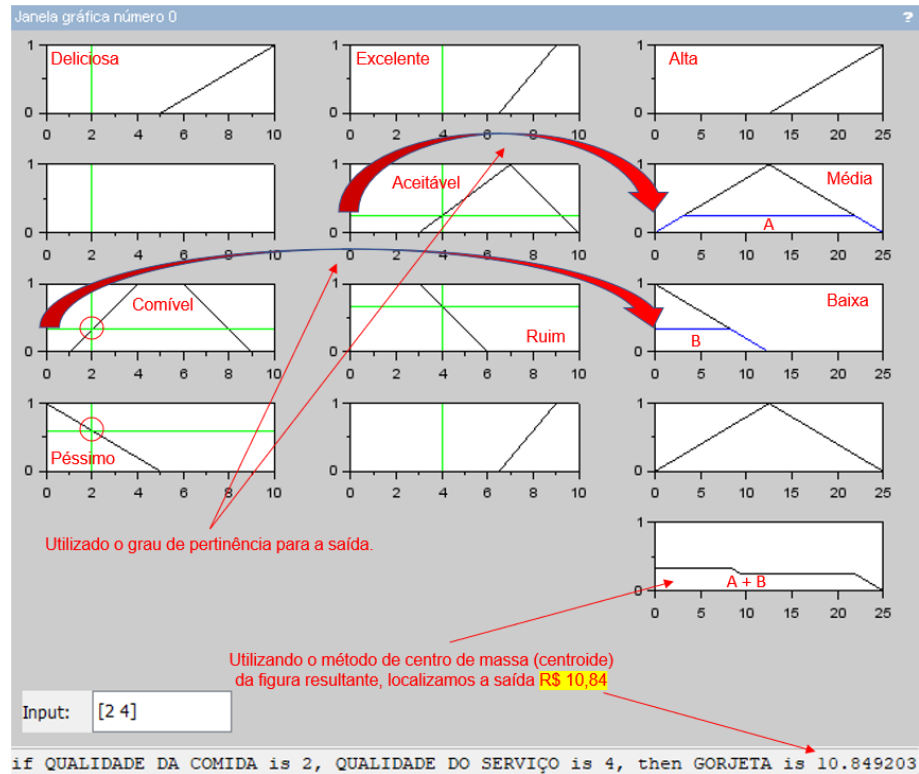
Figura 17 – Plotagem da resposta



Fonte: Dos Autores.

Ao acessar a ferramenta *View Rules*, temos o a resposta apresentado na figura 18, foi atribuído as entradas os valores das notas, sendo 2 para a *qualidade da comida* e 4 para a *qualidade do serviço*, a figura 18 mostra todas as relações de entrada e saída, de acordo com a lógica *fuzzy*.

Figura 18 – Relações de entrada e saída



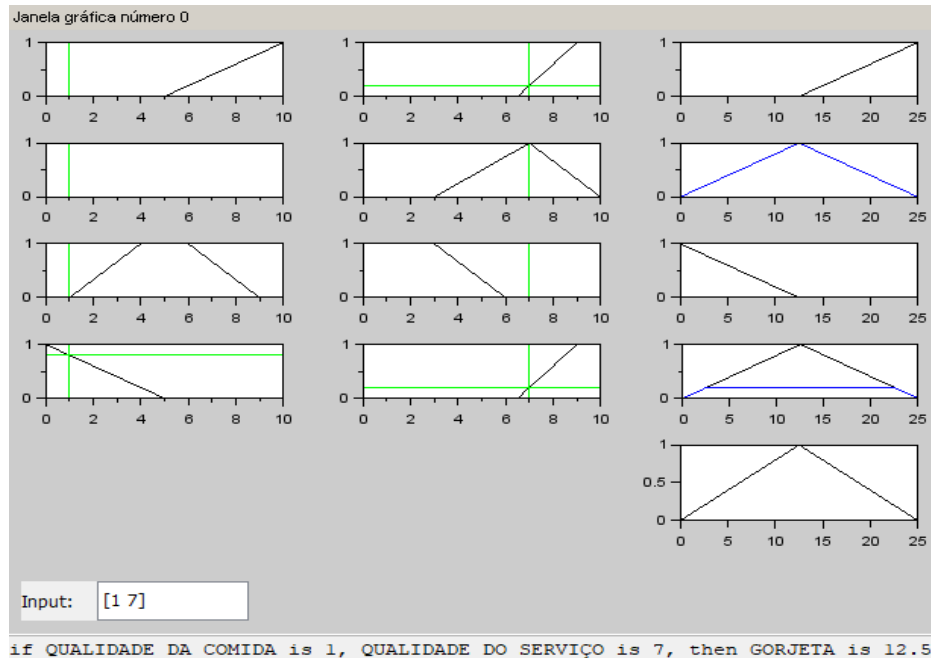
Fonte: Dos Autores.

Conforme apresentado na figura 18, a qualidade da comida está sendo avaliada em duas variáveis diferentes (Péssimo e Comível), toda via, a saída analisa os graus de pertinências das variáveis e se comporta de acordo com o grau de pertinência mais influente, o mesmo se aplica para a qualidade do serviço.

Com base nessas informações, a gorjeta se mostra proporcional a relação das entradas, e faz uso do método de centro de massa para definir o valor final, conforme apresentado na figura, sendo assim, temos um valor atribuído na saída de R\$ 10,84.

Para uma resposta em que a qualidade da comida recebe uma nota 1 e a qualidade do serviço uma nota 7. Nesta situação, existe a aplicação de umas das regras feitas. A figura 19 mostra todas as relações de entradas e saída.

Figura 19 – Relações de entrada e saída



Fonte: Dos Autores.

Mesmo com os graus de pertinências existentes, a aplicação se baseou em uma das regras, na qual diz que se a qualidade do serviço for “aceitável”, a gorjeta deverá ser média, tendo seu valor aplicado em R\$ 12,50

Foram realizadas diversas simulações no software, a fim de validar as possibilidades do cotidiano. A tabela 1 contempla os valores das entradas e os respectivos resultado da saída, sendo possível observar a coerência entre os valores.

Tabela 1 – Simulações.

Qualidade da comida	Qualidade do serviço	Gorjeta (R\$)
1	1	4,08
4	4	8,39
3	6	10,68
3	7	10,77
6	8	10,38
8	7	12,12
5	6	10,22
9	10	20,76
10	8	14,2

Fonte: Dos Autores.

4. CONCLUSÃO

A Lógica *Fuzzy* permite aos sistemas computacionais inteligentes “raciocinar” considerando aspectos inerentes à incerteza e aos processos realísticos e torná-lo mais “humano”. A principal vantagem de utilizar a Lógica *Fuzzy* deve-se a sua capacidade de lidar com incertezas, raciocínio aproximado e termos vagos, o que não é possível de se fazer com as lógicas clássicas.

O raciocínio humano envolve todos esses elementos tratados pela Lógica *Fuzzy*, por isso ela tem uma grande importância no contexto da Inteligência Artificial, que procura representar o raciocínio e o conhecimento humano da forma mais realística possível.

No problema de classificação de Comida usado como exemplo, após fazer as simulações utilizando a extensão SCIFLT do software *SCILAB* e analisando os gráficos obtidos, ficou fácil saber qual seria o valor adequado da gorjeta a ser paga levando em consideração a qualidade do serviço e da comida. É avaliado para trabalhos futuros a utilização da lógica fuzzy para controle de processos industriais.

5. REFERÊNCIAS BIBLIOGRÁFICAS

BENEVIDES, Mário. **Introdução Motivação e Conjuntos Fuzzy**.

Disponível em: <https://www.cos.ufri.br/~mario/logica/logicaFuzzy.pdf>.

Acesso em: 28/01/2021.

TEDESCO, Patricia. GERMANO, Vasconcelos. **Logica Difusa Fuzzy**.

Disponível em: <https://www.cin.ufpe.br/~if684/EC/aulas/Aula-logica-Fuzzy-SI.pdf>

Acesso em: 30/03/2021.

TANSCHHEIT, Ricardo. **Sistema Fuzzy**.

Disponível em: <http://www.inf.ufsc.br/~mauro.roisenberg/ine5377/Cursos-ICA/LN-Sistemas%20Fuzzy.pdf>

Acesso em: 25/03/2021.

RIGNEL, Diego G. De Sousa; CHENCI, Gabriel Pupin; LUCAS, Carlos Alberto.

Uma Introdução A Lógica Fuzzy.

Disponível em: http://www.logicafuzzy.com.br/wp-content/uploads/2013/04/uma_introducao_a_logica_fuzzy.pdf

Acesso em: 19/04/2021

GOMIDE, Fernando A. Campos; GUDWIN, Ricardo Ribeiro. **Modelagem, Controle, Sistemas E Lógica Fuzzy.**

Disponível em: <https://www.sba.org.br/revista/volumes/v4n3/v4n3a02.pdf>

Acesso em: 19/04/2021.

TEDESCO, Patrícia; VASCONCELOS, Germano. **Lógica Difusa (Fuzzy).**

Disponível em: <https://www.cin.ufpe.br/~if684/EC/aulas/Aula-logica-fuzzy-SI.pdf>

Acesso em: 19/04/2021.