

PROGRAMA DE PÓS-GRADUAÇÃO EM SISTEMAS E PROCESSOS
INDUSTRIAIS – MESTRADO
ÁREA DE CONCENTRAÇÃO EM CONTROLE
E OTIMIZAÇÃO DE PROCESSOS INDUSTRIAIS

Ederson Luis Posselt

INFUZZY - FERRAMENTA PARA DESENVOLVIMENTO DE APLICAÇÕES DE
SISTEMAS DIFUSOS

Santa Cruz do Sul, Abril de 2011

Ederson Luis Posselt

INFUZZY - FERRAMENTA PARA DESENVOLVIMENTO DE APLICAÇÕES DE
SISTEMAS DIFUSOS

Dissertação apresentada ao Programa de Pós-Graduação em Sistemas e Processos Industriais da Universidade de Santa Cruz do Sul – UNISC como requisito parcial para a obtenção do título de Mestre em Sistemas e Processos Industriais.

Orientadores: Prof^ª. Dr^ª Rejane Frozza.

Prof. Dr. Rolf Fredi Molz.

Santa Cruz do Sul, Abril de 2011.

“Imagination is more important than knowledge.

Knowledge is limited.

Imagination encircles the world.”

Albert Einstein.

*“A mente que se abre a uma nova idéia
jamais voltará ao seu tamanho original.”*

Albert Einstein.

AGRADECIMENTO

Agradeço imensamente a minha esposa Kéli por estar comigo ao longo dessa etapa, me dando apoio e incentivo nos momentos de fraqueza, tornando possível esse dia. Dedico essa vitória a você minha prenda amada.

Agradeço a todos os meus familiares, Flávia, Eduarda, Jonas S., Sérgio, Zuleica, Jaqueline, Jonas B., e em especial a minha mãe Glaci e minha avó Similda que entenderam minhas ausências. Por acreditarem na concretização deste sonho, compartilho, agora, o mérito desta conquista.

A todos os meus colegas do PPGSPI 2009, mais do que colegas de classe, mais do que parceiros no estudo, irmãos que fizeram parte de um cotidiano.

A meus colegas de trabalho que me acompanharam e me incentivaram nessa jornada, abnegando muitas vezes de seu tempo para me ajudar. A toda equipe da empresa em que trabalho, Impley Tecnologia Eletrônica, que muitas vezes entenderam as minhas necessidades e me apoiaram com tempo e recursos.

Enfim, muito obrigado ao corpo docente e secretariado do Programa de Pós-Graduação em Sistemas e Processos Industriais - PPGSPI (Mestrado), que compartilharam de seus conhecimentos, especialmente minha orientadora, professora, Rejane Frozza e ao meu co-orientador, professor, Rolf Fredi Molz que me acompanharam no decorrer desse projeto, contribuindo para o sucesso do mesmo.

Agradeço à CAPES (Coordenação de Aperfeiçoamento de Pessoal de Nível Superior) pelo apoio financeiro, que permitiu que essa etapa fosse alcançada.

E a todos aqueles que contribuíram na busca deste ideal, meus mais profundos e sinceros agradecimentos.

RESUMO

A lógica difusa é uma excelente técnica para a solução de problemas que envolvam o uso de dados imprecisos, porém a sua implementação e o seu entendimento não são fatos triviais. A carência de uma ferramenta de aplicação de sistemas difusos no mercado que contemple os seguintes requisitos mínimos: acesso gratuito, interação gráfica, fácil aprendizado, simulação e um ambiente para integração com outros sistemas motivou esse projeto. As ferramentas mais completas de sistemas difusos, existentes atualmente no mercado, são de domínio de empresas privadas e estão disponíveis apenas em versões de demonstração ou compra. Existem algumas implementações gratuitas e de código aberto, porém essas ferramentas são na sua grande maioria incompletas, sem ambientes gráficos, de difícil aprendizado, com pouca ou nenhuma documentação. Após uma revisão bibliográfica sobre a lógica difusa, ergonomia e usabilidade de *software*, foi desenvolvida a ferramenta InFuzzy, uma ferramenta gráfica para modelagem de aplicações de sistemas difusos, dotada de recursos de simulação e integração com outros *softwares* através do protocolo UDP/IP. Para avaliação e validação foram modeladas aplicações na ferramenta InFuzzy e essas comparadas com os resultados obtidos no Matlab^{®1} e artigos.

Palavras-Chave: Lógica difusa, ferramenta de modelagem de sistemas difusos, ferramenta InFuzzy.

¹ Marca registrada

ABSTRACT

Fuzzy logic is an excellent technique for the solution of problems involving the use of imprecise data, but its implementation and its understanding are not trivial facts. The deficiency of an enforcement tool of fuzzy systems on the market that includes the following minimum requirements: free access, interaction, graphical, easy to learn, and a simulation environment for integration with others systems prompted this project. The most comprehensive tools for fuzzy systems, currently on the market, are the domain of private companies and are only available in trial versions or purchase. There are some implementations free and open source, but these tools are mostly incomplete, with no graphical environments, difficult to learn, with little or no documentation. After a literature review on fuzzy logic, ergonomics and usability of *software*, was developed InFuzzy Tool, a graphical tool for modeling of fuzzy systems, with its simulation capabilities and integration with other *software* via UDP / IP. For evaluation and validation of the tool, were modeled applications in InFuzzy and compared with results obtained in Matlab[®] and articles.

Keywords: Fuzzy logic, modeling tool of fuzzy systems, tool InFuzzy.

LISTA DE FIGURAS

Figura 1- Breve histórico da evolução que antecede a lógica difusa.....	19
Figura 2 - Diferença entre lógica clássica e lógica difusa (<i>fuzzy Logic</i>)	20
Figura 3 - Exemplo do uso de conjuntos difusos.....	23
Figura 4 – Variável linguística e termos linguísticos.	28
Figura 5 - Função triangular.	29
Figura 6 - Função de pertinência trapezoidal.	30
Figura 7 - Função de pertinência gaussiana.	31
Figura 8 - Função de pertinência em forma de sino (<i>Cauchy</i>).	31
Figura 9 - Função de pertinência em conjuntos discretos.	32
Figura 10 - Centro da gravidade.	34
Figura 11 - Primeiro dos máximos.	36
Figura 12 - Centro dos máximos.	36
Figura 13 – Método de defuzificação último dos máximos.	38
Figura 14 - Método de defuzificação por altura.	39
Figura 15 - Método de defuzificação do centro da soma.	40
Figura 16 - Sistema de controle difuso baseado no modelo Mamdani.....	43
Figura 17 - Diagrama de inferência Mamdani completo.....	44
Figura 18 - Sistema de inferência Sugeno completo.	46
Figura 19 - Tela do <i>Software Fuzzy Logic ToolBox™</i>	50
Figura 20 - Tela do <i>Software Mathematica®</i>	52
Figura 21 - Tela do <i>Software FIDE™ (Fuzzy Inference Development Environment)</i>	53
Figura 22 - Tela do <i>Software TILShell</i>	54
Figura 23 - Tela do <i>Software FuzzyTECH®</i>	56
Figura 24 - Tela do <i>Software RockOn</i>	57
Figura 25 - Tela do <i>Software SciLab</i>	59
Figura 26 – UNFUZZY Sistema de lógica difusa desenvolvido por Velasco em 1997.....	60
Figura 27 - Organização da ferramenta XFuzzy.	61
Figura 28 - Tela principal do <i>software XFuzzy</i>	62
Figura 29 - Relação de aprendizado focado em usuários iniciantes e usuários avançados.	66

Figura 30 - Tela do <i>software</i> Ergolist.....	73
Figura 31- Diagrama de caso de uso	79
Figura 32- Modelagem da estrutura dos dados.....	80
Figura 33- Fluxograma InFuzzy.....	81
Figura 34- Resultado da aplicação da lista de checagem	85
Figura 35- Resultado aplicação <i>checklist</i> , itens aplicáveis.....	85
Figura 36- InFuzzy - Ferramenta de desenvolvimento de sistemas difusos.....	86
Figura 37- Menu de seleção de novo projeto	89
Figura 38- Janela de novo projeto.	89
Figura 39- Tela de controle sobre o projeto difuso.	90
Figura 40- Árvore da estrutura do projeto difuso modelado.	92
Figura 41- Tela de edição de variáveis de entrada e saída.	95
Figura 42- Tela de definição das regras.....	97
Figura 43- Tela de personalização do projeto difuso.	99
Figura 44- Tela de adição e configuração de textos do projeto.....	100
Figura 45- Tela de simulação do sistema difuso.	102
Figura 46- Tela de simulação etapa defuzificação.	106
Figura 47- Gráfico do histórico de saídas.....	107
Figura 48- <i>Log</i> de comunicação UDP.	108
Figura 49- Estrutura de comunicação UDP.....	119
Figura 50- Modelagem do sistema difuso no Matlab [®] e no InFuzzy.....	124
Figura 51- Variável de entrada "Serviço" e os termos linguísticos.....	125
Figura 52- Variável de entrada "Comida" e os termos linguísticos	126
Figura 53- Variável de saída "Gorjeta" e termos linguísticos	127
Figura 54- Conjunto de regras do projeto Gorjeta.....	128
Figura 55- Configuração dos parâmetros do projeto Gorjeta.....	128
Figura 56- Etapa de simulação do projeto Gorjeta.....	129
Figura 57- Projeto gorjeta, lista de simulações	130
Figura 58- Erro apresentado pelo <i>software</i> Matlab [®] no resultado final.....	131
Figura 59- Erro apresentado pelo <i>software</i> Matlab [®] no processo de defuzificação.	132
Figura 60- Instalador do sistema InFuzzy etapa 1	133

Figura 61- Instalador do sistema InFuzzy etapa 2.....	134
Figura 62- Instalador do sistema InFuzzy etapa 3.....	135
Figura 63- Instalador do sistema InFuzzy etapa 4.....	135
Figura 64- Instalador do sistema InFuzzy etapa 5.....	136
Figura 65- Instalador menu iniciar	137
Figura 66- Instalação da ferramenta InFuzzy no Linux	138
Figura 67- Iniciar a ferramenta InFuzzy através de linha de comando.	138
Figura 68- Iniciar a ferramenta InFuzzy através do menu do Gnome.....	139
Figura 69- Implementação do projeto energia limpa na ferramenta InFuzzy	145
Figura 70- Projeto difuso - Controle de ventilador.....	149
Figura 71- Projeto controle de ventilador implementado na ferramenta InFuzzy.....	150
Figura 72- Projeto difuso controle ventilador: Gráfico de saída da variável "Velocidade"	153
Figura 73- Tela de simulação da ferramenta InFuzzy	189
Figura 74- Ferramenta InFuzzy UDP Tool.....	191
Figura 75- Ferramenta InFuzzy UDP Tool com os valores preenchidos	192
Figura 76- Registro do envio de mensagens da ferramenta InFuzzy UDP Tool.....	193

LISTA DE TABELAS

Tabela 1- Aplicação da lógica difusa	20
Tabela 2 - Tabela para exemplificar o processo de avaliação <i>walkthrough</i>	68
Tabela 3 - Exemplo de avaliação de heurística.	70
Tabela 4 - Exemplo de avaliação GOMS	71
Tabela 5- Tabela resumida da aplicação do método <i>checklist</i> baseado em questões do Ergolist .	83
Tabela 6- Tabela com os opcionais de configuração do sistema difuso.....	87
Tabela 7- Tabela com informações sobre a tela de modelagem de projetos difusos.....	91
Tabela 8- Descrição do processo de analítico das etapas do controlador difuso.....	103
Tabela 9- Descrição textual do sistema difuso implementado.	105
Tabela 10- Arquivo do sistema: Seção projeto.....	110
Tabela 11- Arquivo do sistema: Seção definição das funções	110
Tabela 12- Arquivo do sistema: Seção janela	111
Tabela 13- Arquivo do sistema: Definição de variáveis de entrada.....	112
Tabela 14- Arquivo do sistema: Termo linguístico de entrada	112
Tabela 15- Arquivo do sistema: Definição de variáveis de saída.....	113
Tabela 16- Arquivo do sistema: Termo linguístico de saída	114
Tabela 17- Arquivo do sistema: Bloco de regra.....	114
Tabela 18- Arquivo do sistema: Conexão de variáveis e blocos de regras	115
Tabela 19- Arquivo do sistema: Regras	116
Tabela 20- Arquivo do sistema: Mensagens de texto.....	117
Tabela 21- Arquivo do sistema: Simulação.....	117
Tabela 22- Arquivo do sistema: Opções do projeto	118
Tabela 23- Protocolo de comunicação	120
Tabela 24- Retornos de erros de comunicação.....	122
Tabela 25- Projeto definição salário: Etapa definição variáveis	141
Tabela 26- Projeto definição salário: Etapa definição das regras.....	142
Tabela 27-Projeto definição salário: Etapa definição dos operadores.....	143
Tabela 28-Projeto definição salário: Etapa de simulação.....	143

Tabela 29- Projeto Energia Limpa: Etapa definição variáveis	145
Tabela 30- Projeto Energia Limpa: Etapa definição regras.....	146
Tabela 31- Projeto Energia Limpa: Etapa definição dos parâmetros	147
Tabela 32- Projeto Energia Limpa: Etapa de simulação	148
Tabela 33- Projeto difuso controle ventilador: Definição das variáveis	150
Tabela 34- Projeto difuso controle ventilador: Definição das regras	151
Tabela 35- Projeto difuso controle ventilador: Definição dos parâmetros.....	151
Tabela 36- Projeto difuso controle ventilador: <i>Debug</i> da comunicação	152
Tabela 37- Índice das ferramentas analisadas	163
Tabela 38 - Comparativo das principais características de uma ferramenta difusa.....	163
Tabela 39 - Questões de validação de ergonomia extraídas do ErgoList.....	166
Tabela 40- Diagnóstico automático de rolamentos baseado baseado em lógica fuzzy	181
Tabela 41- Controle fuzzy para a melhoria do perfil de tensão em sistema de distribuição de energia elétrica.....	182
Tabela 42- Implementação de um sistema de controle inteligente utilizando a lógica fuzzy	183
Tabela 43- Desenvolvimento de um algoritmo de controle para sistemas de elevadores, com base na lógica fuzzy.....	184
Tabela 44- <i>Optimal Control And Fuzzy Control Application In Nonlinear Vehicular Suspension Models With Magnetorheological Damper</i>	185
Tabela 45- Aplicação de sistemas fuzzy e neuro-fuzzy para a predição da temperatura retal de frangos de corte	186
Tabela 46- Sistema especialista em lógica fuzzy para o controle, gerenciamento e manutenção da qualidade.....	187

LISTA DE ABREVIATURAS

GOMS	<i>Goals / Operators / Methods / Selection Rules</i>
UNISC	Universidade de Santa Cruz do Sul
IHC	Interface Homem Computador
μ	Função de Pertinência
Min	Função Mínima
Max	Função Máxima
T	<i>T-Norm</i> , Norma-T ou Norma triangular
S	<i>T-Conorm</i> , Conorma-T ou <i>S-norm</i>
OU, OR	Operador de União
AND, E	Operador de Intersecção
NOT, Não ou ~	Operador de Negação
IA	Inteligência Artificial
UDP	<i>User Datagram Protocol</i>
IP	<i>Internet Protocol</i>
PID	Proporcional + Integral + Derivativo
IMSE-CNM	Instituto de Microeletrônica de Sevilha – Centro Nacional de Microeletrônica
IDE	<i>Integrated Development Environment</i>
NMEA	<i>National Marine Electronics Association</i>
MDI	<i>Multiple Document Interface</i>
TM	<i>Trademark</i>
®	Marca Registrada Comercial
DLL	<i>Dynamic-Link Library</i>

SUMÁRIO

INTRODUÇÃO.....	16
2 FUNDAMENTAÇÃO TEÓRICA	18
2.1 Lógica difusa	18
2.1.1 Teoria dos conjuntos difusos	23
2.1.2 Operações sobre conjuntos difusos.....	24
2.1.3 Variável linguística.....	27
2.1.4 Fuzificação.....	28
2.1.5 Defuzificação.....	32
2.1.5.1 Defuzificação centro de gravidade	33
2.1.5.2 Defuzificação primeiro dos máximos.....	34
2.1.5.3 Defuzificação centro dos máximos	36
2.1.5.4 Defuzificação último dos máximos	37
2.1.5.4 Defuzificação pelo método da altura	38
2.1.5.5 Defuzificação centro da soma	40
2.1.6 Sistema de controle difuso.....	41
2.1.6.1 Sistema de controle difuso baseado no modelo Mamdani	43
2.1.6.2 Sistema de controle difuso baseado no modelo Takagi-Sugeno	45
2.2 Ferramentas de lógica difusa existentes no mercado.....	47
2.2.1 FuzzyF – <i>Fuzzy Logic Framework</i>	47
2.2.2 Fuzzy Logic ToolBox™	49
2.2.3 Mathematica® <i>Fuzzy Logic</i>	50
2.2.4 FIDE™	52
2.2.5 TILShell 3.0.....	53
2.2.6 FuzzyTECH®	55
2.2.7 RockOn Fuzzy Tool	56
2.2.8 SciLab Fuzzy Tool	58
2.2.9 UNFUZZY: Sistema de lógica difusa	59
2.2.10 XFuzzy	61
2.2.11 Comparativo entre as ferramentas analisadas.....	63

2.3 Ergonomia e usabilidade de interfaces	64
2.3.1 Metáforas de interfaces.....	65
2.3.2 Usabilidade de interfaces.....	65
2.3.3 Avaliação de interfaces.....	67
2.3.3.1 <i>Cognitive walkthrough</i>	67
2.3.3.2 Avaliação heurística	70
2.3.3.3 GOMS.....	71
2.3.3.4 <i>CheckList</i>	72
2.3.3.5 Definição da técnica a ser utilizada	74
3 METODOLOGIA.....	75
3.1 Métodos de pesquisa.....	75
3.2 Síntese dos procedimentos metodológicos	76
4 MODELAGEM E DESENVOLVIMENTO DA FERRAMENTA INFUZZY	78
4.1 Modelagem da ferramenta InFuzzy	78
4.2 Aplicação das características de ergonomia e usabilidade de <i>software</i>	82
4.3 Funcionalidades da ferramenta InFuzzy.....	86
4.4 Ferramenta InFuzzy	88
4.4.1 Modelagem de projetos difusos.....	90
4.4.2 Tela de edição de variáveis de entrada e saída	93
4.4.3 Tela de edição de regras	95
4.4.4 Tela de configuração de preferências	97
4.4.5 Tela de adição de texto no projeto.....	99
4.4.6 Tela de simulação	101
4.4.6.1 Descrição das etapas de simulação.....	103
4.4.6.2 Descrição do sistema difuso	105
4.4.6.3 Gráfico de defuzificação.....	106
4.4.6.4 Histórico de saídas.....	107
4.4.6.5 <i>Debug</i> Comunicação	108
4.5 Estrutura do arquivo gerado pelo sistema InFuzzy	109
4.6 Protocolo de comunicação UDP.....	119
4.7 Homologação da ferramenta InFuzzy.....	123

4.7.1 Inconformidades encontradas no <i>software</i> Matlab®	130
4.8 Processo de instalação da ferramenta InFuzzy	132
4.9 Projetos implementados na ferramenta InFuzzy	140
4.9.1 Projeto difuso: Definição salário	141
4.9.2 Projeto difuso: Uso da lógica <i>fuzzy</i> para a otimização de sistemas energéticos baseados em energia limpa	144
4.9.3 Projeto difuso: Controle da velocidade de um ventilador aplicado em <i>hardware</i>	148
4.10 Diferenciais da ferramenta InFuzzy em relação as demais ferramentas apresentadas	154
CONCLUSÃO.....	156
REFERÊNCIAS	160
ANEXO A: Tabela comparativa de sistemas difusos.....	163
ANEXO B: Lista de checagem de ergonomia ErgoList.....	166
ANEXO C: Projetos Fuzzy implementados	181
ANEXO D: InFuzzy UDP Tool	189

INTRODUÇÃO

A lógica difusa (*fuzzy logic*) tem como principal característica a capacidade de operar sobre dados imprecisos, mostrando-se mais adequada para tratar modos de raciocínio aproximados ao invés de exatos. A aplicação desta técnica pode retornar resultados satisfatórios para diversas aplicações, e isso se faz visível através de várias publicações realizadas sobre o tema em congressos, trabalho de conclusão de curso, dissertações e teses, seja como fundamentação ou estudo de caso. No Anexo C é possível conferir trabalhos desenvolvidos com lógica difusa aplicada nas mais diversas situações.

Atualmente, mesmo com os benefícios da utilização da lógica difusa, essa técnica é pouco disseminada nas empresas de *software* como forma de auxílio em processos. Esse fato se deve em grande parte à questão de sua modelagem e implementação que requer tempo, profissionais qualificados e conhecimento sobre o assunto.

Para a execução desse projeto, foi realizado um estudo sobre as principais ferramentas de sistema difusos existentes no mercado. A carência de um *software* gratuito, com interação gráfica e de fácil manuseio, capaz de realizar simulações e integração com outros sistemas, motivou este trabalho. Os *softwares* mais completos são de domínio de empresas privadas e estão disponíveis apenas em versões de demonstração ou compra. Existem alguns programas gratuitos e de código aberto, porém, essas ferramentas são na sua grande maioria incompletas, sem ambientes gráficos, de difícil aprendizado, com pouca ou sem documentação.

Desta forma, justifica-se a importância de estudar e desenvolver um *software* que atenda aos requisitos propostos, que sirva de apoio ao meio acadêmico e como solução no meio industrial. A disponibilidade de uma ferramenta de sistemas difusos gratuita incentiva o seu uso por professores e profissionais da área de Inteligência Artificial (IA), fazendo com que alunos tenham a possibilidade de visualizar as vantagens da lógica difusa na prática e as empresas possam ver a utilidade de seu uso em processos próprios.

O objetivo principal deste trabalho foi modelar e implementar uma ferramenta para desenvolvimento de aplicações de sistemas difusos, incentivando a solução de problemas com esta técnica, facilitando a interação com o usuário. Para desenvolver a aplicação, fez-se necessária uma pesquisa dos conceitos da lógica difusa, análise das ferramentas e sistemas já existentes e modelagem das funcionalidades propostas. Durante o desenvolvimento do *software*, realizou-se uma validação das suas funcionalidades. Essa validação ocorreu através da comparação entre resultados da ferramenta proposta e de uma ferramenta consagrada no mercado – o Matlab[®]. Outra forma de validar a ferramenta foi através da implementação de modelos propostos em artigos, trabalhos de conclusão de curso, dissertações e teses, a fim de provar que a ferramenta possui os recursos necessários para o desenvolvimento de aplicações.

A organização do texto está dividida em capítulos que contemplam a evolução do trabalho. O capítulo 2 trata da fundamentação teórica envolvendo a lógica difusa, ferramentas existentes no mercado e ergonomia/usabilidade de *software*. O capítulo 3 traz a metodologia aplicada neste trabalho. O capítulo 4 apresenta as características e as funcionalidades implementadas, homologação, implementação de projetos difusos da ferramenta InFuzzy. No último capítulo são apresentadas as conclusões e os trabalhos futuros.

2 FUNDAMENTAÇÃO TEÓRICA

Este capítulo tem por objetivo descrever conceitos relacionados ao tema do trabalho, sendo necessário o conhecimento de duas áreas. A primeira área é a lógica difusa, sendo essa técnica o motivo principal do projeto, juntamente com uma análise das ferramentas já existentes no mercado, para determinar as características da ferramenta a ser desenvolvida. A segunda área refere-se ao estudo sobre conceitos e técnicas de ergonomia de *software*.

As próximas seções conduzem aos principais conceitos necessários para o desenvolvimento do trabalho, iniciando com uma breve introdução à lógica difusa. Na sequência, aborda-se os conjuntos difusos e suas operações, métodos de fuzificação, métodos de defuzificação e sistemas de inferência. Um estudo realizado sobre as principais ferramentas de sistemas difusos é apresentado, com uma comparação entre elas. Por último, ocorre uma explanação sobre usabilidade e ergonomia de *software*, item indispensável para quem desenvolve e projeta sistemas.

2.1 Lógica difusa

Conforme Kohagura (2007), é possível traçar um histórico que antecede a lógica difusa. A Figura 1 apresenta os fatos históricos em sequência cronológica, iniciando pela lógica aristotélica formulada por Aristóteles entre 348 e 322 A.C. que definia uma informação como verdadeira ou falsa, não podendo uma informação ser verdadeira e falsa. Em 1847, Boole atribui valores numéricos para as afirmações, 1 (um) para afirmações verdadeiras e 0 (zero) para afirmações falsas, originando a álgebra booleana. Em 1903, através do paradoxo de Russell, Bertrand demonstra que nem todos os problemas podem ser resolvidos com lógica bivalente. Por volta do ano de 1930, Jan Lukasiewicz defende que uma informação pode ser verdadeira e falsa ao mesmo tempo, desde que existam vários níveis de veracidade. Em 1965, Lotfi A. Zadeh publica o trabalho sobre conjuntos *fuzzy*, baseado na lógica multinível.

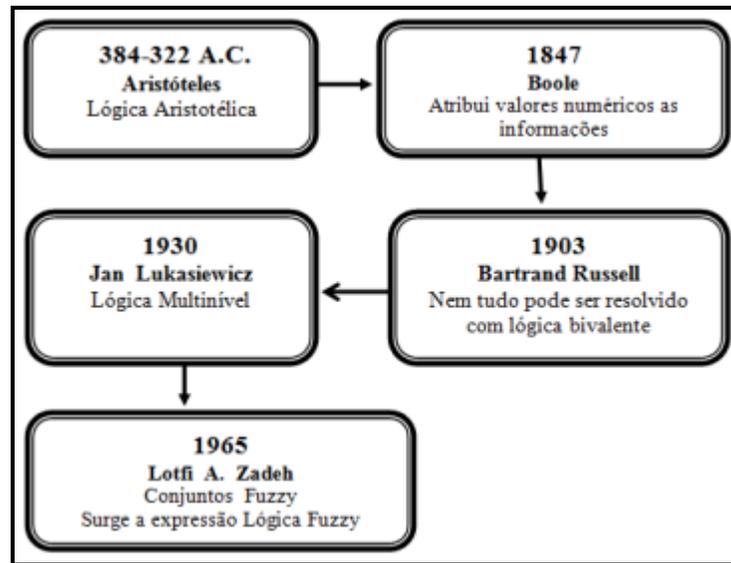


Figura 1- Breve histórico da evolução que antecede a lógica difusa

Fonte: do autor

Lotfi Zadeh, professor de engenharia elétrica e ciência da computação pela Universidade de Berkeley na Califórnia, desenvolveu em 1965 a teoria da lógica difusa (ZADEH, 2009). Zadeh observou que muitas das regras utilizadas pelas pessoas para fazer inferência não podiam ser explicadas por elas. Um exemplo disso seria dizer: “Aquele homem deve ter 40 anos”. Explicar de maneira formal as regras utilizadas nesta observação não é uma tarefa fácil. Assim, baseado nessas observações Zadeh criou a lógica difusa, inicialmente criticada, porém acabou sendo utilizada por engenheiros e cientistas da computação (COSTA et al., 2009). Maiores informações sobre Zadeh podem ser encontradas na sua página pessoal, que contém informações sobre publicações com possibilidade de *download* (ZADEH, 2009).

A lógica difusa é uma técnica que consiste em mapear dados de entrada em dados de saída. Existem algumas técnicas que realizam essa implementação como: sistemas especialistas, redes neurais, entre outras (FUJIMOTO, 2005). Em uma citação, Lotfi Zadeh menciona: “*In almost every case you can build the same product without fuzzy logic, but fuzzy is faster and cheaper.*” (MATHWORKS, 2009).

Tratar informações subjetivas e imprecisas ou até mesmo ambíguas é algo natural para a mente humana, porém computadores foram construídos e preparados para trabalharem com

informações precisas e bem estruturadas, não podendo reconhecer informações do tipo: “Está um pouco quente hoje”; “Está quase pronto em sua forma inicial”. Tendo em vista esses conceitos, a lógica difusa surgiu para que essas informações pudessem ser tratadas.

Na Figura 2 é possível observar a diferença entre a lógica clássica e a lógica difusa. A lógica clássica não abre espaço para valores intermediários, enquanto na lógica difusa é possível analisar os valores intermediários. Na lógica clássica, a pessoa é alta ou baixa, enquanto que na lógica difusa é utilizado o grau de pertinência para informar o grau de veracidade para a informação alta, por exemplo. Uma pessoa pode ser alta com grau de pertinência 0, o que indica que se trata de uma pessoa baixa, ou pode ser alta com grau de veracidade 0,5, o que indica uma pessoa de estatura média. Outro exemplo seria: Na lógica clássica, o dia pode estar frio ou quente, enquanto que na lógica *fuzzy* pode haver várias determinações como: frio, muito frio, pouco frio, morno, quente, muito quente, pouco quente entre outras.

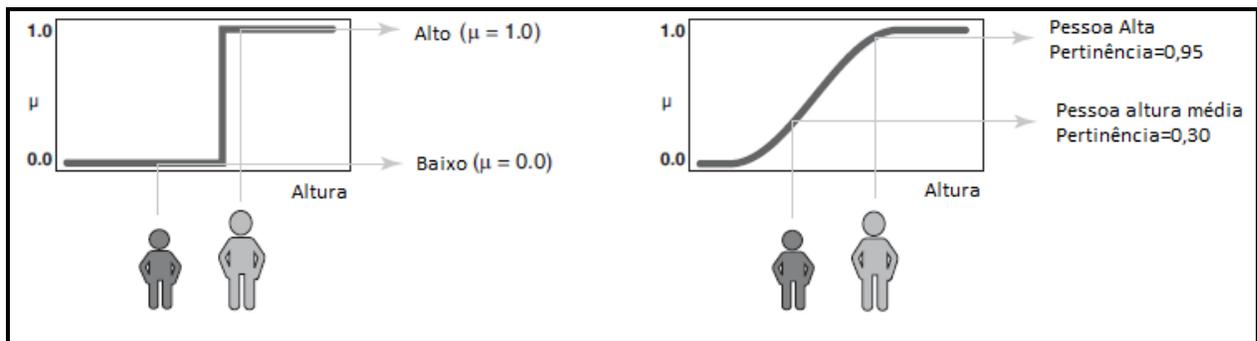


Figura 2 - Diferença entre lógica clássica e lógica difusa (*fuzzy Logic*)

Fonte: Adaptado do Manual Matlab® (MATHWORKS, 2009).

A Tabela 1 apresenta uma lista de produtos que aplicam a lógica difusa. Através desta tabela é possível perceber a diversidade de aplicações desta técnica, que parte de aplicações simples até complexas.

Tabela 1- Aplicação da lógica difusa

Produto	Empresa	Função
Condicionador de Ar	Hitachi, Matsushita, Mitsubishi, Sarp	Controle de temperatura

Freios “ <i>Anti-lock</i> ”	Nissan	Controla o freio em situações perigosas, baseado na velocidade e aceleração do carro.
Motor de carro	NOK/Nissan	Controla a injeção e ignição, baseado na regulação do acelerador, temperatura da água de refrigeração, concentração de oxigênio, RPM, volume de combustível, entre outras.
Transmissão Automática	Honda, Nissan, Subaru	Seleciona a relação de engrenagens de transmissão, baseado na carga do motor, estilo do motorista e condições da estrada.
Máquina Copiadora	Canon	Ajusta a voltagem do tambor, baseado na densidade da imagem, temperatura e umidade.
Controle de Velocidade de Cruzeiro	Isuzu, Nissan, Mitsubishi	Ajusta a regulagem do acelerador para manter velocidade de cruzeiro, baseado na velocidade e aceleração do carro.
Máquina de Lavar Louça	Matsushita	Ajusta o ciclo de limpeza e enxágue, e a estratégia de lavagem, baseado no número de louças, no tipo e na quantidade de comida encrostada na louça.
Secadora	Matsushita	Ajusta a estratégia e tempo de secagem, baseado no tamanho da carga e tipo de tecido.
Controle de Elevador	Fujitec, Mitsubishi Electric, Toshiba	Reduz o tempo de espera, baseado no tráfego de passageiro.
Sistema de Gerenciamento de Saúde	Omron	Avalia e acompanha a saúde e disposição de empregados de uma empresa.
Forno de Microondas	Hitachi, Sanyo, Sarp, Toshiba	Ajusta a estratégia e potência de cozimento.
Palmtop Computer	Sony	Reconhece caracteres “Kanji” escritos à mão.
Refrigerador	Sharp	Ajusta o tempo de degelo e resfriamento.
Rice Cooker	Matsushita, Sony	Ajusta o método e o tempo de cozimento, baseado no volume do arroz, temperatura e vapor.

Câmara Fixa	Canon, Minolta	Acha o objeto em qualquer lugar do quadro e ajusta o autofoco.
Televisão	Goldstar, Hitachi, Samsung, Sony	Ajusta a cor e textura da tela para cada quadro, e estabiliza o volume baseado na localização do espectador.
Tostadeira	Sony	Ajusta o aquecimento e o tempo de tostagem para cada tipo de pão.
Aspirador de Pó	Hitachi, Matsushita, Toshiba	Ajusta a estratégia do motor de sucção, baseado na quantidade de poeira e tipo do piso.
Câmara de Vídeo	Canon, Sony	Ajusta autofoco e iluminação.
Câmara de Vídeo	Panasonic	Ajusta autofoco e cancela a distorção por tremor de imagem
Máquina de Lavar Roupa	Daewoo, Goldstar, Hitachi, Matsushita, Samsung, Sony, Sharp	Ajusta a estratégia de lavagem, baseado no nível sujeira, tipo de tecido, na quantidade de roupa, e nível d'água.
Sistema de diagnóstico de Golf	Maruman Golf	Escolha do clube de golfe baseada no físico e tacada dos jogadores.
Controle de fabricação	Omron	Listas de tarefas e estratégias das linhas de montagens.
Umidificador	Casio	Ajusta a umidade contida de acordo com as condições da sala.
Controle de moinho de ferro	Nippon Steel	Combina as entradas de conjuntos de tempo e temperatura.

Fonte: Material SOUZA (2010), KOHAGURA (2007)

Para um melhor entendimento da lógica difusa, deve-se realizar uma detalhada explanação sobre alguns conceitos importantes dessa teoria. Um deles é a teoria dos conjuntos difusos, parte fundamental para a criação de controladores difusos.

2.1.1 Teoria dos conjuntos difusos

Os conjuntos difusos foram desenvolvidos para determinar o quanto um elemento pertence ou não a um determinado conjunto, com o uso de graus de pertinência (μ), que são valores reais no intervalo $[0;1]$. Assim, é possível determinar os elementos que pertencem a um grupo e seu respectivo grau de pertinência.

A representação dos conjuntos difusos depende da natureza e das dimensões do universo a ser representado. Quando for necessário representar conjuntos que contenham um universo pequeno, a melhor solução é a analítica, que tem o objetivo de apresentar todos os elementos do universo separados por vírgula, sendo que cada elemento é composto pelo grau de pertinência e o valor do termo. Quando o universo a ser representado contiver um grande número de elementos, ele deve ser representado pelo gráfico de sua função de pertinência, chamado de diagrama de Hassi-Euler (H-E) (REZENDE, 2003).

A Figura 3 apresenta um exemplo de determinação de conjuntos difusos. É possível perceber a representação do universo dos elementos no item 1 da figura, sendo esse universo composto por idades. No item 2 da figura são definidos três conjuntos difusos, sendo eles jovens, adultos e velhos. Já no item 3 da figura são representados os graus de pertinência aos conjuntos em relação aos elementos do universo. É possível reparar que aos trinta anos um indivíduo tem, por exemplo, um grau de pertinência de 0,5 em relação ao conjunto jovem; 0,4 em relação ao adulto e 0,1 em relação ao velho.

1)	U = {5, 10, 20, 30, 40, 50, 60, 70}			
2)	Conjuntos difusos: jovens, adultos, velhos			
3)	Idades	Jovens	Adultos	Velhos (μ)
	5	1,0	0,0	0,0
	10	1,0	0,0	0,0
	20	0,8	0,2	0,0
	30	0,5	0,4	0,1
	40	0,2	0,4	0,4
	50	0,1	0,3	0,6
	60	0,0	0,2	0,8
	70	0,0	0,0	1,0

Figura 3 - Exemplo do uso de conjuntos difusos

Fonte: Material de aula (FROZZA, 2007)

Os conjuntos difusos podem ser discretos ou contínuos. Se os dados do conjunto Jovem da Figura 3 forem discretos, deverão ser representados pela união de todos os elementos do universo juntamente com seus respectivos graus de pertinência. O sinal “+” é utilizado para representar a união dos elementos e cada elemento é explicado por “X / pertinência de X”.

$$\text{Jovens} = \{5/1,0 + 10/1,0 + 20/0,8 + 30/0,5 + 40/0,2 + 50/0,1 + 60/0,0 + 70/0,0\}$$

Conjuntos contínuos são representados por funções ou conjunto de funções de pertinência. Utilizando novamente o conjunto Jovem da Figura 3, tem-se o seguinte conjunto contínuo triangular para a exemplificação. As variáveis A, B e X pertencem ao eixo das abscissas, onde A e B são os dois primeiros pontos do triângulo.

$$\text{Jovem}(X) = \{1 \text{ para } X < 10; X-A / B-A \text{ para } 10 < X \leq 50; 0 \text{ para } X > 50\}$$

Na próxima seção, serão apresentadas as principais operações sobre conjuntos difusos. Essas operações são necessárias para o desenvolvimento do trabalho.

2.1.2 Operações sobre conjuntos difusos

Para a aplicação da lógica difusa, é necessária a utilização de operações sobre conjuntos difusos, sendo essas operações divididas em dois grupos: Norma-T (Norma Triangular ou *T-norm*) e Conorma-T (Conorma Triangular ou *T-conorm*). As operações são aplicadas sobre o grau de pertinência (μ) dos elementos. Para exemplificar as operações, serão utilizados os conjuntos A, B e C descritos a seguir. As operações que serão apresentadas são equivalentes para conjuntos discretos e contínuos (CALDEIRA et al, 2007):

$$A(X) = \{0/0,0 + 1/0,5 + 2/0,3 + 3/0,5 + 4/0,8 + 5/0,6\}$$

$$B(X) = \{0/0,5 + 1/0,3 + 2/0,3 + 3/0,2 + 4/0,7 + 5/0,3\}$$

$$C(X) = \{0/1,0 + 1/1,0 + 2/0,5 + 3/0,2 + 4/0,1 + 5/0,0\}$$

- a) **Norma-T (Mínimo - T_{min}):** para cada par ordenado a pertinência será a menor pertinência entre A e B.

$$T_{\min}(A,B) = \{0/0,0 + 1/0,3 + 2/0,3 + 3/0,2 + 4/0,7 + 5/0,3\}$$

- b) **Norma-T (Produto Algébrico - T_{pa}):** para cada elemento o valor da pertinência será igual ao produto da pertinência de A por B.

$$T_{pa}(A,B) = \{0/0,0 + 1/0,15 + 2/0,09 + 3/0,1 + 4/0,56 + 5/0,18\}$$

- c) **Norma-T (Produto Limitado - T_{pl}):** para cada elemento o valor da pertinência será a soma da pertinência de A e B. Essa operação possui apenas uma restrição: caso a soma das pertinências seja maior que 1, o valor da pertinência será 1.

$$T_{pl}(A,B) = \{0/0,5 + 1/0,8 + 2/0,6 + 3/0,7 + 4/1,0 + 5/0,9\}$$

- d) **Norma-T (Produto Drástico - T_{pd}):** essa operação é baseada em três regras para cada elemento: pertinência de A se pertinência de B = 1, pertinência de B se pertinência de A=1 e por último, pertinência 0 se pertinência de A e B for menor que 1.

$$T_{pd}(B, C) = \{0/0,5 + 1/0,3 + 2/0,0 + 3/0,0 + 4/0,0 + 5/0,0\}$$

- e) **Conorma-T (Máximo - S_{max}):** Essa operação é o contrário da Norma-T mínimo. Para cada elemento, a pertinência será a maior entre as pertinências de A e B.

$$S_{\max}(A, B) = \{0/0,5 + 1/0,5 + 2/0,3 + 3/0,5 + 4/0,8 + 5/0,6\}$$

- f) **Conorma-T (Soma Algébrica - S_{sa}):** Essa operação consiste na seguinte operação matemática para cada elemento: soma da pertinência de A e B subtraindo-se o produto da pertinência de A e B.

$$S_{sa}(A, B) = \{0/0,5 + 1/0,65 + 2/0,51 + 3/0,6 + 4/0,94 + 5/0,72\}$$

- g) Conorma-T (Soma Limitada - Ssl):** para cada elemento será realizado o mínimo entre 1 e a soma da pertinência de A e B. O resultado dessa operação é igual ao resultado da Norma-T do produto limitado.

$$Ssl(A,B) = \{0/0,5 + 1/0,8 + 2/0,6 + 3/0,7 + 4/1,0 + 5/0,9\}$$

- h) Intersecção padrão:** a intersecção de dois conjuntos difusos é dada pela função de mínimo: $\mu = \min(\mu_A, \mu_B)$. Segue exemplo:

$$A(X) = \{0/0,0 + 1/0,5 + 2/0,3\}$$

$$B(X) = \{0/0,5 + 1/0,3 + 2/0,3\}$$

$$A \text{ AND } B = \{0/0,0 + 1/0,3 + 2/0,3\}$$

- i) União padrão:** a união de dois conjuntos difusos é dada pela função de máximo: $\mu = \max(\mu_A, \mu_B)$. Como modelo de exemplificação cita-se:

$$A(X) = \{0/0,0 + 1/0,5 + 2/0,3\}$$

$$B(X) = \{0/0,5 + 1/0,3 + 2/0,3\}$$

$$A \text{ OR } B = \{0/0,5 + 1/0,5 + 2/0,3\}$$

- j) Complemento padrão:** o complemento de um conjunto difuso é expresso pela seguinte função: $\mu = 1 - \mu_A$. Segue exemplo.

$$A(X) = \{0/0,0 + 1/0,5 + 2/0,3\}$$

$$A^{\sim}(X) = \{0/1,0 + 1/0,5 + 2/0,7\}$$

- k) Concentração:** a concentração de um conjunto difuso é representada pela função: $\mu = (\mu_A)^2$. Linguisticamente a concentração está ligada ao termo “muito” e tem como consequência a diminuição da fuzificação. Na seqüência ilustra-se um exemplo.

$$A(X) = \{3/0,7 + 5/0,8 + 8/0,3\}$$

$$A^2(X) = \{3/0,49 + 5/0,64 + 8/0,09\}$$

- l) **Dilatação:** a dilatação de um conjunto difuso é representada pela função: $\mu = (\mu A)^{1/2}$. Linguisticamente a dilatação está associada ao termo “mais ou menos” e tem como consequência o aumento da fuzificação. Segue exemplo para exemplificar.

$$A(X) = \{0/0,7 + 1/0,4 + 2/0,6\}$$

$$A^{1/2}(X) = \{0/0,84 + 1/0,2 + 2/0,77\}$$

Após o estudo dos conjuntos difusos e as principais operações sobre eles, a próxima etapa na fundamentação da lógica difusa são as variáveis linguísticas e os termos linguísticos.

2.1.3 Variável linguística

Variável linguística é a entidade utilizada para representar um conceito ou uma variável de um determinado problema. Essa variável aceita somente valores linguísticos, tais como: frio, pouco frio, muito frio, grande, muito grande (REZENDE, 2003). Analisando-se a Figura 3, pode-se reconhecer a idade como sendo a variável linguística e os seus valores representados pelos conjuntos difusos: jovem, adulto e velho.

O processo de fuzificação é responsável por atribuir a cada conjunto difuso os respectivos valores de pertinência. Na Figura 4 é possível observar um exemplo da atribuição dos valores de pertinência aos valores referentes à variável linguística velocidade. Observando-se a Figura 4, pode-se notar que a velocidade de 30 km/h tem grau de pertinência 0,5 tanto para o valor devagar quanto para médio, em contrapartida pode-se reparar que a velocidade de 60 km/h tem grau máximo de pertinência em relação ao valor médio e nulo em relação aos valores devagar e rápido.

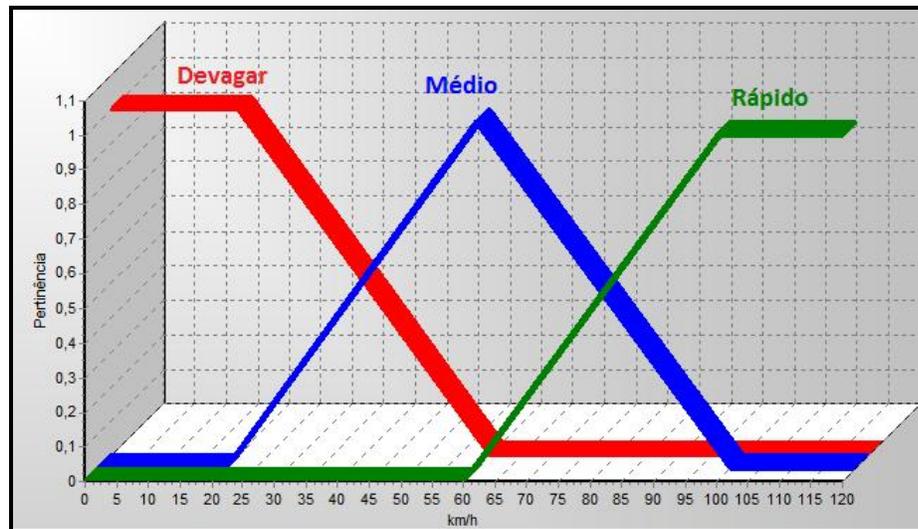


Figura 4 – Variável linguística e termos linguísticos.

Fonte: do autor, obtida através da ferramenta InFuzzy.

Com o conhecimento das variáveis linguísticas adquirido, a próxima etapa a ser estudada engloba processos de fuzificação (primeira etapa de um controlador difuso).

2.1.4 Fuzificação

O primeiro passo realizado no controlador difuso é a fuzificação, esse processo deve ser realizado para cada valor de entrada e consiste no mapeamento dos dados para conjuntos difusos relevantes. Os valores de entrada são valores precisos provenientes geralmente de medições ou observações. O processo de fuzificação também é responsável pela ativação das regras relevantes para uma determinada situação (TANSCHKEIT, 2003).

A fuzificação é realizada através de funções de pertinência. A seguir, serão apresentadas as funções de pertinência mais utilizadas e conhecidas (CALDEIRA et al., 2007).

- a) **Função de pertinência triangular:** essa é a função mais utilizada juntamente com a função do trapézio. Para a criação de um triângulo, são necessários três pontos em x (a, b, c). Os valores de a, b e c devem obedecer a seguinte regra: $a < b < c$. Para ser um conjunto normal, deve existir um valor de x com pertinência 1. Assim, o valor de y

para a e c será 0 e para b será 1. Para calcular o valor de pertinência da função triangular deve-se utilizar a função (eq. 1) a seguir:

$$f(x; a, b, c) \begin{cases} 0, & x < a \\ (x - a) / (b - a), & a \leq x < b \\ (c - x) / (c - b), & b \leq x < c \\ 0, & c \leq x \\ \text{Restrição:} & a < b < c \end{cases}$$

...(1)

A Figura 5 apresenta o gráfico de função triangular. Observando esse gráfico, é possível perceber a pertinência para as entradas de X . Se a função triangular representasse um conjunto difuso de “Velocidade Média”, então teríamos a confiança de 100% para a velocidade de 50 Km/h, e possibilidade de veracidade de 0% para 20 e 80 Km/h.

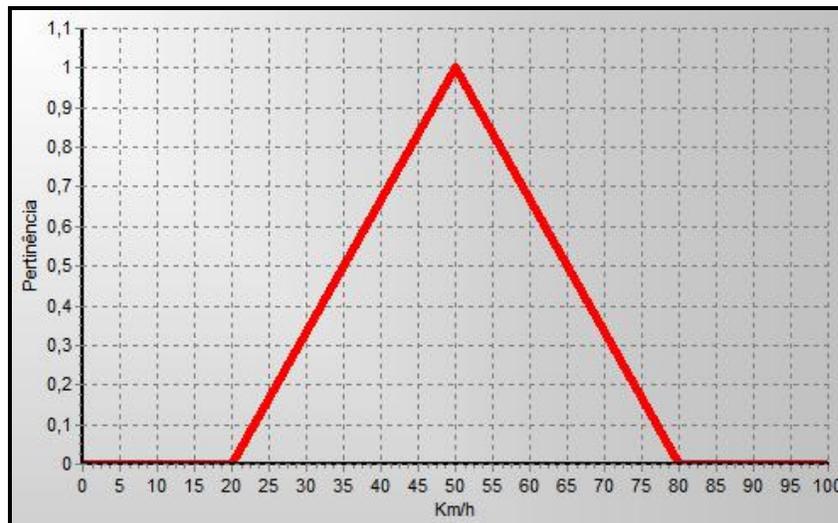


Figura 5 - Função triangular.

Fonte: do autor.

b) Função de pertinência trapezoidal: obedece ao modelo matemático de 4 parâmetros, sendo que $a < b \leq c < d$. A função (eq. 2) realiza a pertinência e na Figura 6 pode ser visualizado um exemplo de função de pertinência trapezoidal. Uma das características de uma função trapezoidal é o fato de um intervalo de valores assumirem a

possibilidade de veracidade de 100%. Observando o gráfico da Figura 6 percebe-se que qualquer valor entre 20 e 60 possui grau de pertinência igual a 1.

$$f(x; a, b, c, d) \begin{cases} 0, & x < a \\ (x-a)/(b-a), & a \leq x < b \\ 1, & b \leq x < c \\ (d-x)/(d-c), & c \leq x < d \\ 0, & x \geq d \\ \text{Restrição:} & a < b \leq c < d \end{cases}$$

...(2)

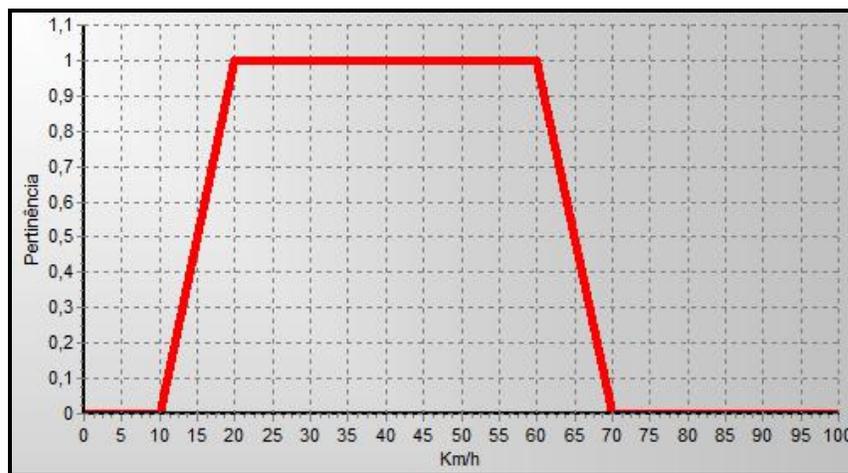


Figura 6 - Função de pertinência trapezoidal.

Fonte: do autor.

- c) **Função de pertinência gaussiana:** utiliza três parâmetros para a sua construção: x , média e desvio padrão. A seguir é apresentada a função (eq. 3) de pertinência gaussiana. A Figura 7 apresenta um exemplo de curva gaussiana.

$$f(x, \text{média}, \text{desvio padrão}) = \text{exponencial} \left(- 1 / 2 * ((x - \text{média}) / \text{desvio padrão})^2 \right)$$

...(3)

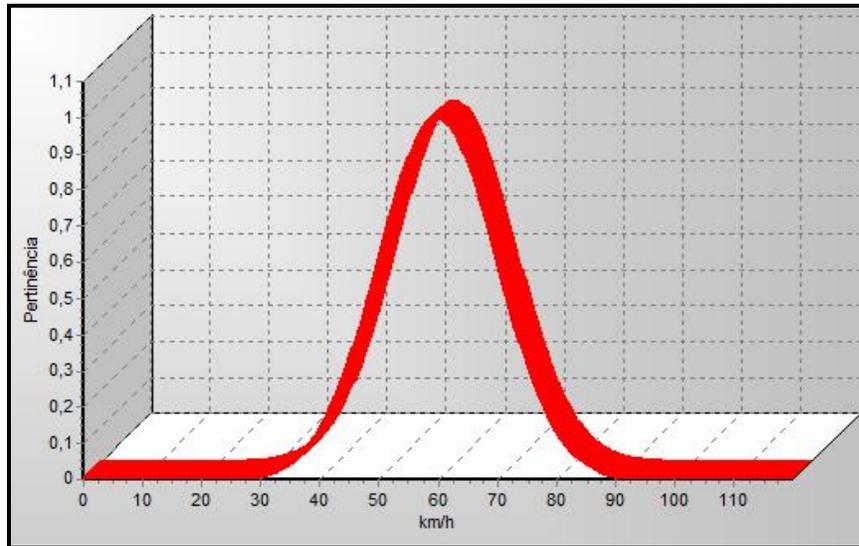


Figura 7 - Função de pertinência gaussiana.

Fonte: Material de aula (FERREIRA, 2007).

- d) Função de pertinência de *Cauchy* (Função de Sino):** A Figura 8 apresenta um exemplo da curva em forma de sino. A pertinência é calculada a partir da função (eq. 4).

$$f(x; a, b, c) = 1 / ((1 + |(x - c) / a|^{(2 * b)})$$
; $b > 0$

...(4)

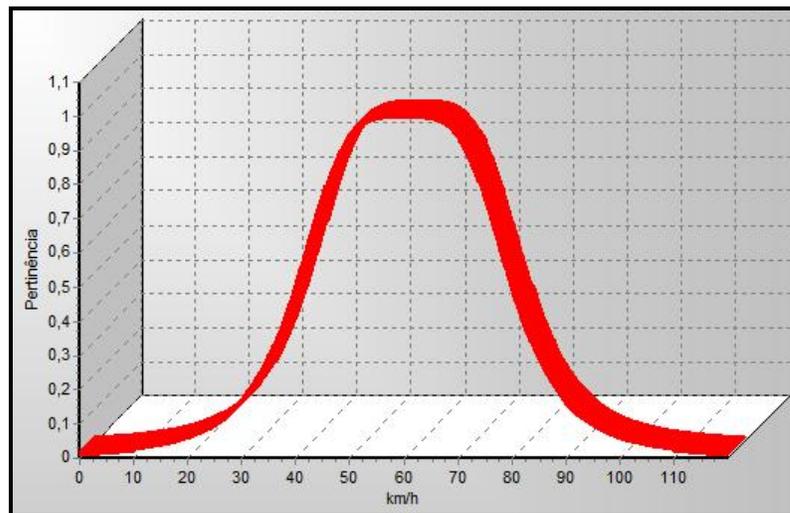


Figura 8 - Função de pertinência em forma de sino (*Cauchy*).

Fonte: Do autor.

e) **Pertinência de conjuntos discretos:** é a mais fácil, uma vez que para cada valor possível, existe uma pertinência definida. Esse é um método que não exige cálculos. Observando a Figura 9, o reconhecimento da pertinência para $x=30$ ($\mu(30) = 1$) é simplificado. Esse método é o mais rápido no processo de fuzificação, indicado para aplicações embarcadas que possuam processadores de baixo desempenho. Deve ser utilizado em casos em que as entradas são inteiras ou conhecidas. Não é aconselhada a utilização em casos em que o intervalo de x contenha um número alto de elementos. É importante observar que os dados discretos poderiam ser substituídos por uma função triangular.

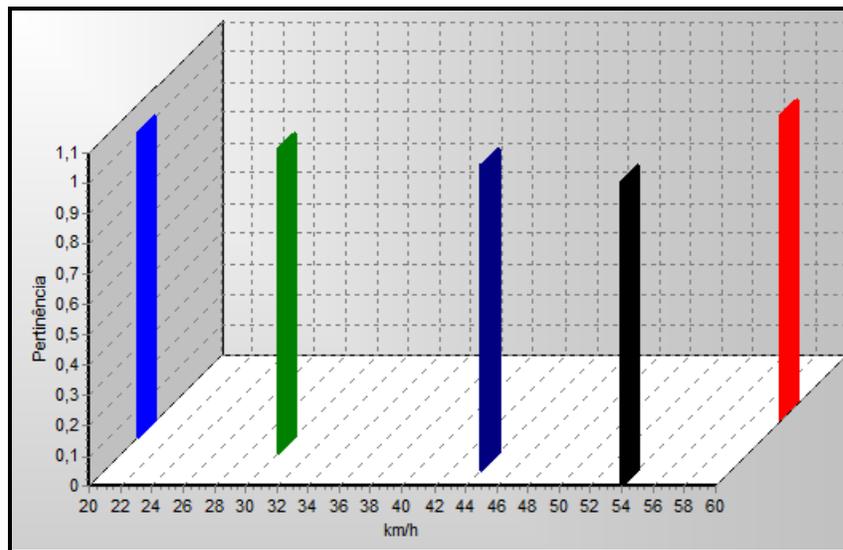


Figura 9 - Função de pertinência em conjuntos discretos.

Fonte: Do autor.

A última etapa de um controlador difuso é a defuzificação. Essa etapa é fundamental e será apresentada a seguir.

2.1.5 Defuzificação

Defuzificação é o processo de conversão de conjuntos difusos em uma saída discreta. Essa atividade é necessária pelo fato de muitos equipamentos exigirem um valor discreto. Para exemplificar, suponha-se que o modelo difuso seja responsável por movimentar um motor, o

equipamento espera como entrada o número de giros e não a informação de “Avanço Médio com pertinência 0,5”. A defuzificação é a etapa que exige o maior processamento, dependendo dos algoritmos escolhidos (WEBER; KLEIN, 2003). Nesta seção, serão abordados os seguintes métodos de defuzificação: centro da gravidade, primeiro dos máximos, centro dos máximos, último dos máximos, altura e centro da soma. É importante destacar que existem outros métodos, porém esses foram abordados por serem os mais utilizados.

2.1.5.1 Defuzificação centro de gravidade

O método do centro de gravidade, também conhecido como centro da área, é o mais notório dos métodos de defuzificação (DRIANKOV; HELLENDROORN; REINFRANK, 1996). A equação (eq. 5) apresenta o processo de defuzificação, observa-se que a solução está na razão entre o somatório de “u” vezes o seu grau de pertinência pelo somatório do grau de pertinência.

$$u_{CG} = \frac{\sum_{i=1}^N u_i \cdot \mu_u(u_i)}{\sum_{i=1}^N \mu_u(u_i)}$$

...(5)

Onde:

U_i = Valor do eixo das abscissas

$\mu_u(U_i)$ = Valor da pertinência de U_i

N = Limite do somatório

Na Figura 10 é possível observar um exemplo do resultado de uma defuzificação através do método do centro da gravidade. Repare que a largura de um conjunto está diretamente relacionada ao grau de precisão do sistema, quanto menor a amplitude mais precisão no resultado. Em uma análise, percebe-se que, onde o segundo conjunto possui grau de pertinência igual a 0,4 e o terceiro conjunto grau de pertinência igual a 0,5 a escala atingida por esses

conjuntos vai de 40 até 110. Aplicando a fórmula do centro da gravidade, obtém-se como valor de saída discreta aproximadamente 80.

$$D_{cg} = \frac{40*0 + 50*0,4 + 60*0,4 + 70*0,4 + 80*0,5 + 90*0,5 + 100*0,5 + 110*0}{0 + 0,4 + 0,4 + 0,4 + 0,5 + 0,5 + 0,5 + 0}$$

$$D_{cg} = 77$$

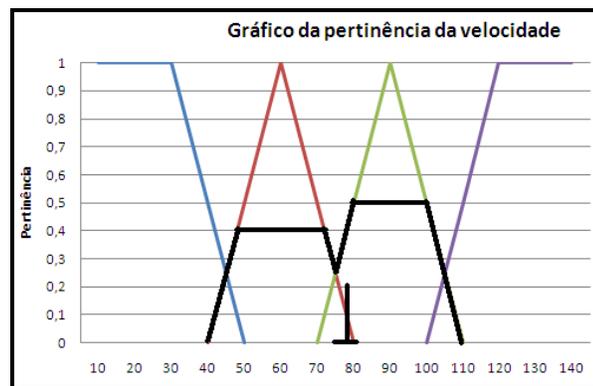


Figura 10 - Centro da gravidade.

Fonte: Do autor.

2.1.5.2 Defuzificação primeiro dos máximos

O processo de defuzificação pelo método do primeiro dos máximos consiste em três etapas (DRIANKOV; HELLENDROORN; REINFRANK, 1996):

- Localizar o conjunto com a maior pertinência.
- Localizar dentro do universo o maior grau de pertinência.
- A saída será o primeiro elemento que tiver a pertinência igual à maior pertinência.

A equação (eq. 6), obtida do livro “*An Introduction to Fuzzy Control*” (DRIANKOV; HELLENDROORN; REINFRANK, 1996) expressa a solução da defuzificação pelo método do primeiro dos máximos.

$$\begin{aligned} \text{hgt}(U) &= \sup \mu_U(u) \\ u^* &= \inf \{ \mu \in U \mid \mu_U(u) = \text{hgt}(U) \} \end{aligned}$$

...(6)

Onde:

$\mu_U(u)$ = Pertinência de um valor do eixo das abscissas

sup = Supremo

u^* = Primeiro valor do eixo das abscissas com a maior pertinência

hgt(U) = Resultado supremo

inf= Ínfimo

μ =grau de pertinência

U=universo

A Figura 11 apresenta um exemplo do processo de defuzificação pelo método do primeiro dos máximos. Supondo que a variável analisada seja “Velocidade”, e que o segundo e terceiro conjunto sejam “Velocidade Média” e “Velocidade Alta”. Observa-se grau de pertinência de 0,4 para “Velocidade Média” e 0,5 para “Velocidade Alta”. Seguindo a primeira etapa desse processo, identifica-se que o conjunto difuso com a maior pertinência é “Velocidade Alta”. A segunda etapa é identificar o maior grau de pertinência dentro do conjunto “Velocidade alta”. Como resposta tem-se o intervalo de 80 a 100 Km/h com pertinência de 0,5. Como última etapa desse método de defuzificação, basta selecionar o primeiro valor de x com o maior grau de pertinência, no caso do gráfico da Figura 11, 80 km/h.

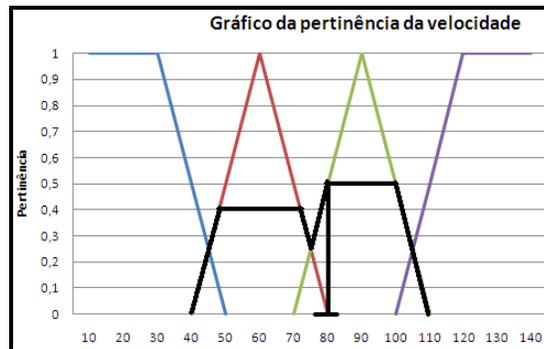


Figura 11 - Primeiro dos máximos.

Fonte: Do autor.

2.1.5.3 Defuzificação centro dos máximos

O método de defuzificação do centro dos máximos se utiliza do valor resultante do processo de defuzificação do primeiro dos máximos e do último dos máximos para calcular a média (DRIANKOV; HELLENDROORN; REINFRANK, 1996). A Figura 12 exhibe o processo de defuzificação do centro do máximo. Seguindo a mesma linha de raciocínio do exemplo apresentado no processo de defuzificação do primeiro dos máximos, tem-se o conjunto “Velocidade alta” com o maior grau de pertinência. Como citado anteriormente, o método do centro dos máximos se vale do resultado do método do primeiro dos máximos e do último dos máximos. Sendo os valores de 80 Km/h como saída do método do primeiro dos máximos e 100 Km/h como saída do último dos máximos, basta realizar a média dos valores para obter o resultado, que no exemplo é 90 km/h.

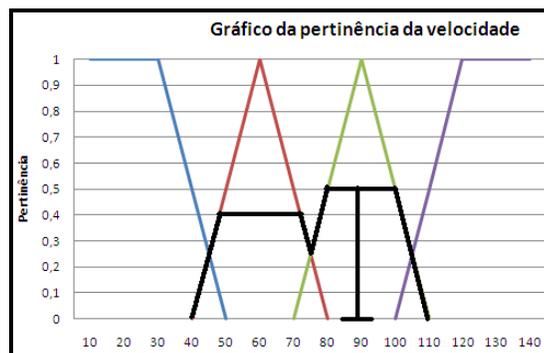


Figura 12 - Centro dos máximos.

Fonte: Do autor.

A equação (eq. 7), obtida do livro “*An Introduction to Fuzzy Control*” (DRIANKOV; HELLENDROORN; REINFRANK, 1996) expressa a solução da defuzificação pelo método do cento dos máximos, onde ocorre a média entre o resultado da defuzificação pelo primeiro dos máximos e a defuzificação do último dos máximos.

$$u^* = \frac{\inf\{u \in X \mid \mu_U(u) = \text{hgt}(U)\} + \sup\{u \in U \mid \mu_U(u) = \text{hgt}(U)\}}{2}$$

...(7)

Onde:

$\mu_U(u)$ = Pertinência de um valor do eixo das abscissas

$\text{hgt}(U)$ = Maior pertinência encontrada no universo

sup= Supremo

inf= Ínfimo

u^* = Média do maior e menor valor do eixo das abscissas

2.1.5.4 Defuzificação último dos máximos

O método de defuzificação do último dos máximos segue a mesma estratégia da defuzificação do primeiro dos máximos, porém selecionando o último valor de x com a maior pertinência. Um exemplo desse processo é exibido na Figura 13. Utilizando-se novamente do exemplo apresentado no método do primeiro dos máximos e do centro dos máximos, tem-se a seguinte situação, o conjunto difuso “Velocidade Alta” é o conjunto de maior pertinência. Como etapa seguinte do processo, basta localizar o último elemento de x que possua o maior grau de pertinência. No exemplo apresentado o valor de saída será 100 km/h.

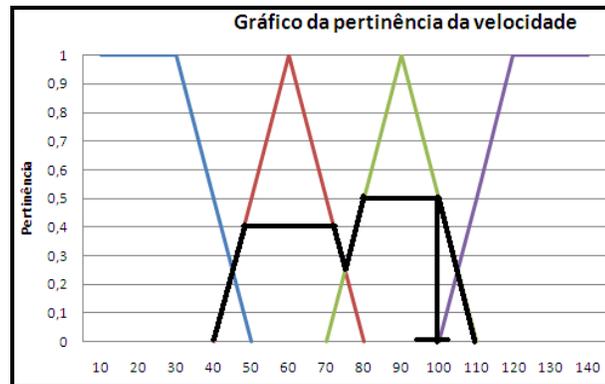


Figura 13 – Método de defuzzificação último dos máximos.

Fonte: Do autor.

A equação (eq. 8), obtida do livro “*An Introduction to Fuzzy Control*” (DRIANKOV; HELLENDROORN; REINFRANK, 1996) expressa a solução da defuzzificação pelo método do último dos máximos.

$$\text{hgt}(U) = \sup \mu_U(u)$$

$$u^* = \sup \{u \in U \mid \mu_U(u) = \text{hgt}(U)\}$$

...(8)

Onde:

$\mu_U(u)$ = Pertinência de um valor do eixo das abscissas

sup = Supremo

u^* = Último valor do eixo das abscissas com a maior pertinência

hgt(U) = Resultado supremo

2.1.5.4 Defuzzificação pelo método da altura

O método de defuzzificação por altura tem como objetivo principal a utilização de pesos. Esse método é considerado simples e rápido e consiste em realizar o somatório dos valores de x com pertinência maior que zero multiplicado pelas suas respectivas pertinências, dividido pelo somatório dos valores de x (WEBER; CLEIN, 2003). A Figura 14 apresenta um exemplo de

defuzificação utilizando o método por altura. Nesse método é possível perceber que o conjunto com o maior grau de pertinência possui uma maior influência sobre o resultado.

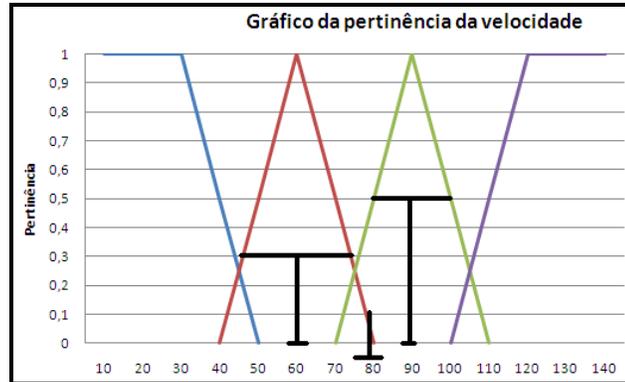


Figura 14 - Método de defuzificação por altura.

Fonte: Do autor.

A equação (eq. 9), obtida do livro “*An Introduction to Fuzzy Control*” (DRIANKOV; HELLENDROORN; REINFRANK, 1996) expressa a solução da defuzificação pelo método da altura. Nesta equação $C^{(k)}$ representa o conjunto de saída e f_k a pertinência. O valor de $C^{(k)}$ é o valor do conjunto com pertinência igual a 1, se tratando de um triângulo, existe apenas um valor, porém no caso de funções trapezoidais, deve se escolher o ponto médio entre os valores com pertinência igual a 1. Essa regra se aplica às demais funções de entrada.

$$u^* = \frac{\sum_{k=1}^m c^{(k)} \cdot f_k}{\sum_{k=1}^n f_k}.$$

...(9)

Onde:

$C^{(k)}$ = Valor de pico do conjunto original

f_k = Valor do conjunto cortado

u^* = Valor do eixo das abscissas após o processamento da equação

Para exemplificar a aplicação da equação, segue exemplo:

$$u^* = \frac{60 * 0,3 + 90 * 0,5}{0,3+0,5}$$

$$u^* = \sim 79$$

2.1.5.5 Defuzificação centro da soma

O método de defuzificação do centro da soma é uma evolução do método centro da área. A diferença entre esses dois métodos está no fato do método centro da soma fazer a contagem da interseção mais de uma vez, sendo que o resultado gerado pelo método centro da soma difere do método do centro de gravidade. Observando a Figura 15, é possível visualizar o resultado do processo de defuzificação.

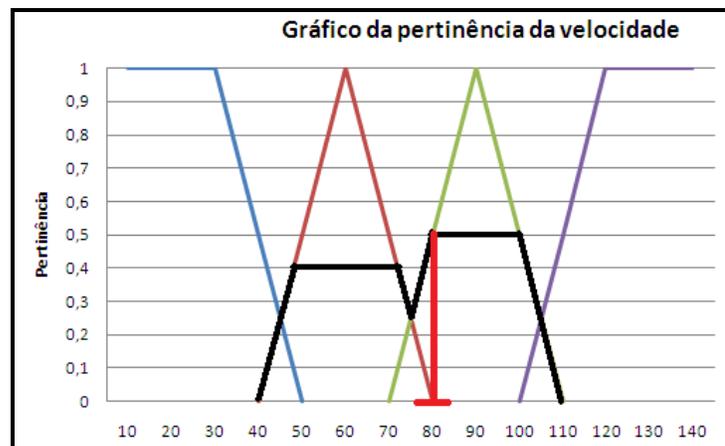


Figura 15 - Método de defuzificação do centro da soma.

Fonte: Do autor.

A equação (6), obtida do livro “*An Introduction to Fuzzy Control*” (DRIANKOV; HELLENDROORN; REINFRANK, 1996) expressa a solução da defuzificação pelo método do centro da soma. Nessa equação realiza-se o somatório do valor central do conjunto de saída vezes o conjunto de saída multiplicado pela sua pertinência, dividido pelo conjunto de saída multiplicado pela sua pertinência.

$$u^* = \frac{\sum_{i=1}^{\ell} u_i \cdot \sum_{k=1}^n \mu_{CLU^{(k)}}(u_i)}{\sum_{i=1}^{\ell} \sum_{k=1}^n \mu_{CLU^{(k)}}(u_i)}.$$

...(6)

Onde:

u_i = Valor no universo das abscissas

μ = Valor de pertinência

CLU= Conjunto difuso de saída

(k)= elemento do conjunto difuso de saída

Após o estudo dos métodos de fuzificação e defuzificação, é momento de compreender as máquinas de inferência. A seguir, serão estudadas as máquinas de inferência de Mamdani e Sugeno.

2.1.6 Sistema de controle difuso

O controlador difuso é composto de três fases distintas, sendo elas: fuzificação, inferência e defuzificação.

A fase da inferência é responsável pela operação sobre os conjuntos propriamente dita. Nela são determinadas as regras que serão ativadas e combinadas, e como resultados desse processo são produzidos conjuntos difusos de saída. (TANSCHKEIT, 2003).

Em um controlador difuso as regras têm um papel relevante, pois o bom desempenho do sistema está vinculado à produção de regras consistentes. A base do conhecimento pode ser fornecida por especialistas através de termos linguísticos. Esse processo pode ser complicado, mas, independente do conhecimento do especialista, uma alternativa para formação da base de regras é através da mineração de dados numéricos. Esse método tem um bom retorno em problemas de classificação (REZENDE, 2003).

A construção de uma base de conhecimento pode dar-se através de regras de produção, sendo essa a maneira mais comum. Uma regra de produção é formada por duas partes principais: antecedente e conseqüente. A estrutura de uma regra de produção é definida da seguinte forma: “Se <antecedente> então <conseqüente>” (REZENDE, 2003).

O sistema de inferência, na maioria das vezes, deve disponibilizar dados precisos como saída. Isso se deve pelo fato da maioria das aplicações necessitarem dessa informação em formato numérico.

Os modelos de inferência mais conhecidos são: modelo de Mamdani, Larsen, Tsukamoto e Takagi-Sugeno (SANDRI; CORREA, 1999). É importante mencionar que existem outros modelos, porém esses não serão abordados neste trabalho em virtude do tempo e foco do trabalho.

O modelo de Mamdani utiliza a função $\min(a,b)$ na etapa das premissas e na implicação e a função $\max(a,b)$ para a agregação. O modelo de Larsen utiliza o produto ($a * b$) nas etapas de premissas e implicação e a função $\max(a,b)$ na etapa de agregação. Os modelos de Mamdani e Larsen necessitam do processo de defuzificação.

O modelo de Tsukamoto e Takagi-Sugeno diferem dos modelos de Mamdani e Larsen na etapa de defuzificação. Esses modelos são conhecidos como modelos de interpolação. O conseqüente é obtido através de uma função gerando um único valor. Através de uma média ponderada obtém-se um valor de saída.

A seguir serão abordados os modelos de Mamdani e Takagi-Sugeno para uma explicação detalhada sobre esses dois métodos que se diferenciam na etapa final.

2.1.6.1 Sistema de controle difuso baseado no modelo Mamdani

O modelo de inferência Mamdani foi proposto na década de 1970 por Mamdani (MAMDANI, 1973). Por muitos anos esse foi o padrão de modelo e continua sendo muito utilizado atualmente. A característica desse modelo é o fato das relações difusas ocorrerem tanto nos antecedentes quanto nos conseqüentes das regras (REZENDE, 2003). Para exemplificar a citação segue uma regra.

Se Temperatura = *AumentandoMuito* **e** Pressão = *MuitoAlta*
Então PerigoExplosão=*Grande*

A Figura 16 exemplifica um controlador difuso Mamdani. Esse controlador recebe uma informação precisa, podendo ela ser de um *hardware* ou de outro *software*. Através do método de fuzificação essa variável é convertida de um valor discreto para um conjunto difuso. Na máquina de inferência são ativadas as regras e gerados os conjuntos difusos de saída. O processo de defuzificação é necessário para se obter uma saída precisa. Essa etapa converte conjuntos difusos em um valor discreto. Essa saída discreta é utilizada para a regulação de algum periférico ou como entrada para outro sistema. O “Sistema Físico” apresentado na figura é utilizado para exemplificar a interação de um controlador difuso com um determinado equipamento. Suponha-se que o sistema físico seja um motor. A entrada no controlador difuso é a carga desse motor e como saída é gerado um novo valor de potência para o motor. Esse ciclo se repete infinitamente para manter a velocidade do motor adequada ao processo.

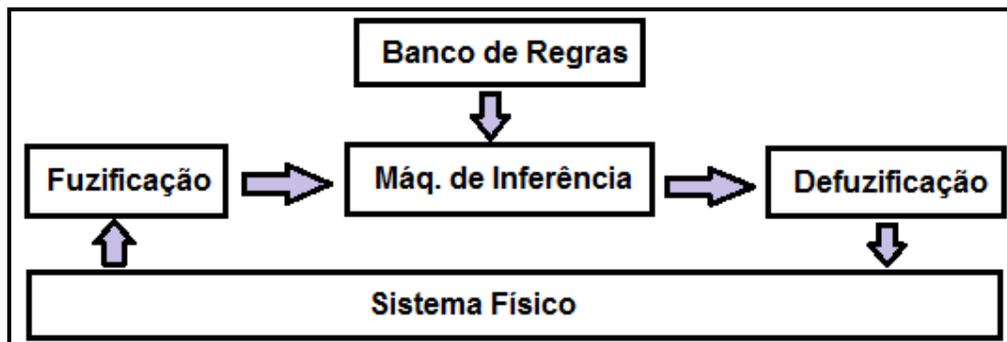


Figura 16 - Sistema de controle difuso baseado no modelo Mamdani.

Fonte: (REZENDE, 2003).

A Figura 17 apresenta um controlador difuso completo, que tem como objetivo determinar a gorjeta de um cliente. Esse sistema possui três variáveis (serviço, comida e gorjeta). As variáveis comida e serviço são variáveis de entrada e gorjeta é a variável de saída. Nesta figura são exibidas 3 regras e as suas saídas no conjunto difuso gorjeta. No processo final é realizada a soma de todas as saídas das regras e aplicado o método do centro da gravidade. Como resultado do processo, obteve-se uma gorjeta de 16,7%.

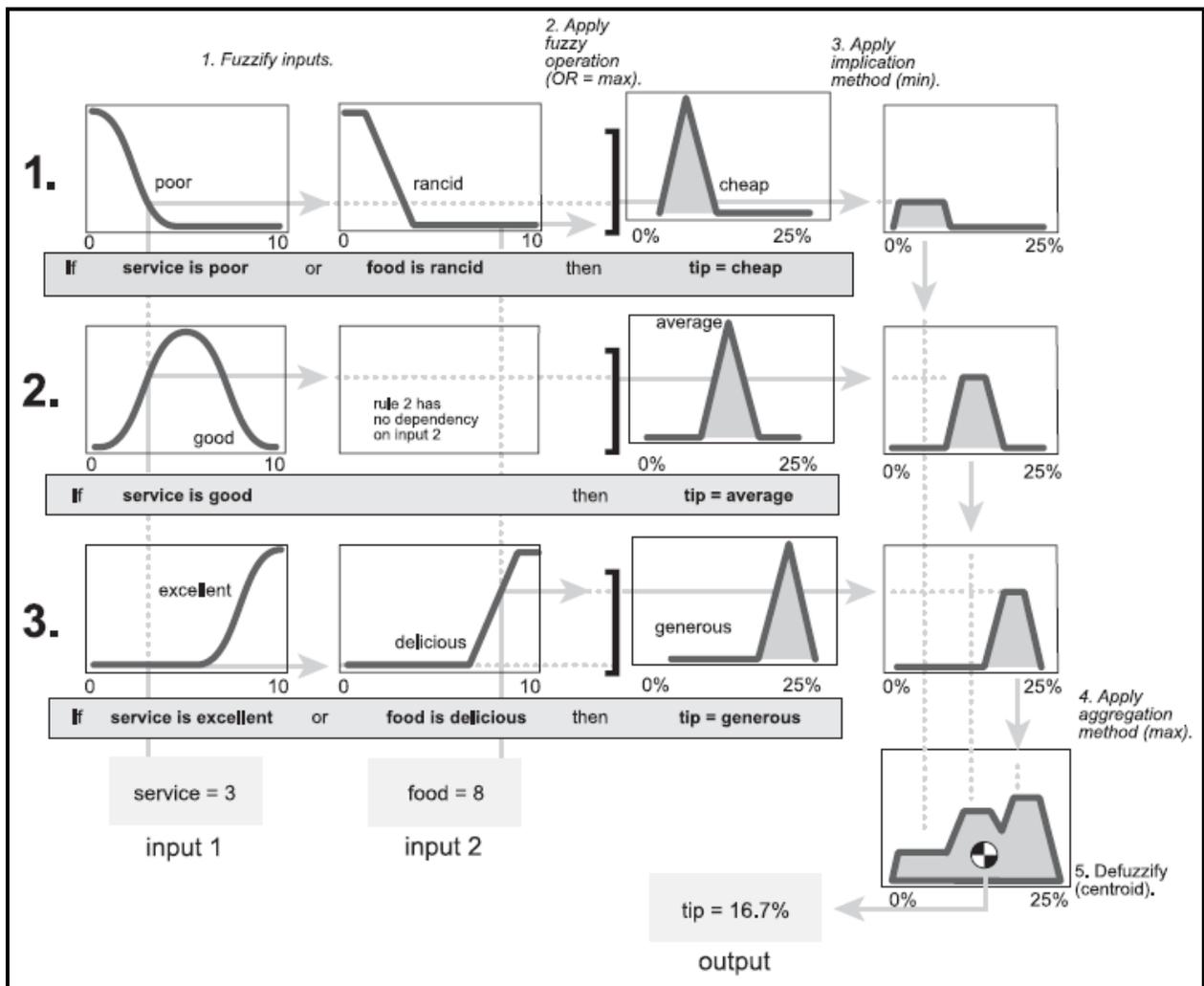


Figura 17 - Diagrama de inferência Mamdani completo.

Fonte: Manual Matlab® (MATHWORKS, 2009).

Conforme MATHWORKS (2009) pode-se citar como vantagens do sistema de inferência Mamdani os seguintes itens:

- É intuitivo.
- Possui ampla aceitação.
- É bem adequado para a entrada humana.

2.1.6.2 Sistema de controle difuso baseado no modelo Takagi-Sugeno

O modelo de inferência Takagi-Sugeno foi proposto na década de 1980 por Takagi e Sugeno (SUGENO, 1985). Esse modelo é similar ao modelo de Mamdani, porém a diferença está no tratamento do consequente, onde ao invés de utilizar relações difusas, utilizam-se equações paramétricas (representação de uma curva). Outra diferença está no fato de na maioria das vezes não utilizar a norma $T \min()$ (REZENDE, 2003). Para exemplificar, segue exemplo de uma regra. A diferença está no consequente, enquanto que no modelo de Mamdani o consequente é um conjunto difuso, no modelo Sugeno, a saída é um valor ou função.

$\text{Se } x_1 = A_i \text{ e } x_2 = A_j \text{ Então } y = 0$
--

A Figura 18 apresenta o modelo difuso da gorjeta. Observa-se a diferença em relação ao modelo de Mamdani. As saídas das regras não ativam conjuntos de difusos, os seus resultados são processados ao final para obtenção de uma saída.

A saída numérica é calculada através da soma das saídas das regras dividida pela soma dos valores de ativação. Como resultado dessa operação matemática, obtém-se o valor preciso de saída, equação (7) (MATHWORKS, 2009).

$$\text{Output} = \frac{\sum_{i=1}^N w_i z_i}{\sum_{i=1}^N w_i} \quad (7)$$

Onde:

W_i = Valor de ativação

Z_i = Valor de saída das regras

N = Limite do somatório

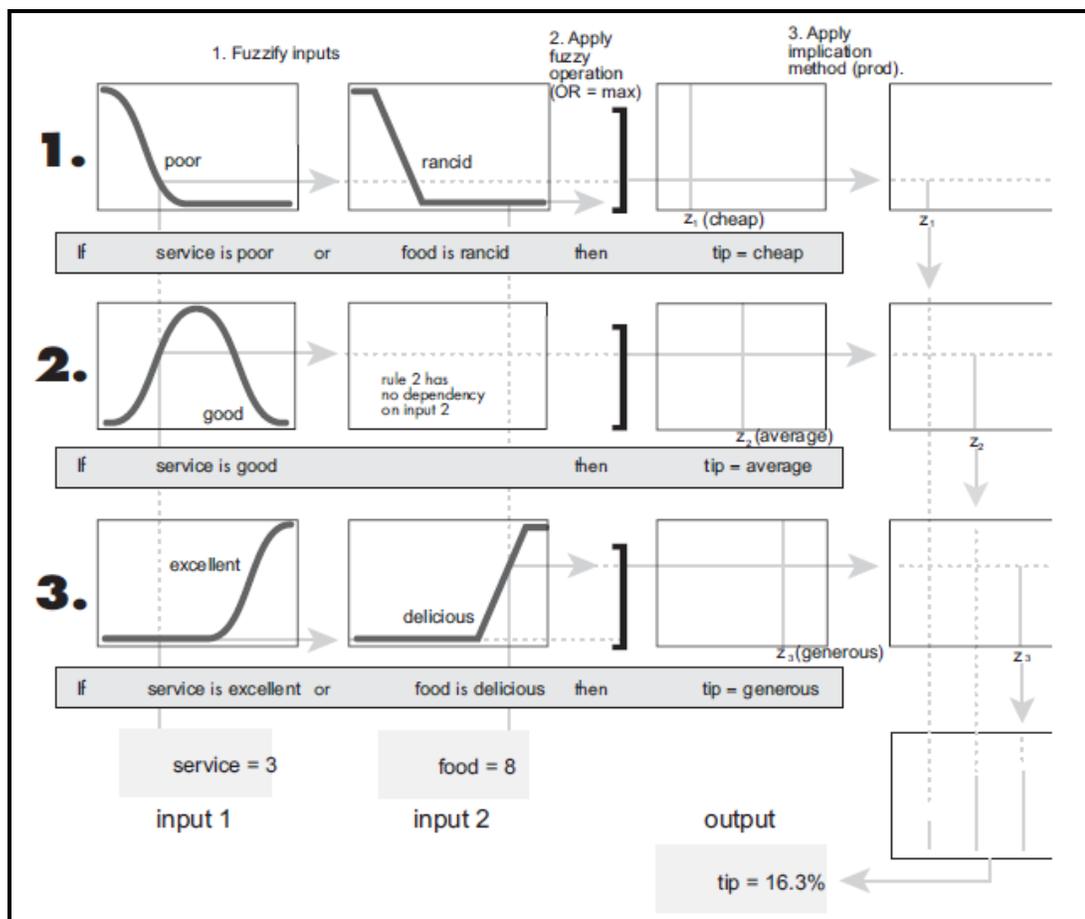


Figura 18 - Sistema de inferência Sugeno completo.

Fonte: Manual Matlab® (MATHWORKS, 2009).

Conforme MATHWORKS (2009) pode-se citar como vantagens do sistema de inferência Sugeno os seguintes itens:

- É computacionalmente eficiente.
- Trabalha bem com técnicas lineares (controle PID (Proporcional+Integral+derivativo)).
- Trabalha bem com otimização e técnicas adaptativas.
- Foi garantida a continuidade da superfície de saída.
- É bem adequado para a análise matemática.

Nesse capítulo explanou-se de forma não exaustiva sobre a lógica difusa. Essa abordagem foi importante para um estudo sobre as técnicas de fuzificação, defuzificação e máquinas de inferência. Na próxima seção será dado foco às ferramentas já existentes no mercado, bem como uma diferenciação entre as mesmas, buscando um posicionamento do presente trabalho dentro desse espectro de aplicativos.

2.2 Ferramentas de lógica difusa existentes no mercado

Para validar a importância do trabalho faz-se necessário o conhecimento das ferramentas de sistemas difusos existentes, reconhecendo oportunidades de mercado para o desenvolvimento de uma nova ferramenta. Nos próximos tópicos serão analisadas as ferramentas, levando-se em conta seus pontos positivos e negativos.

2.2.1 FuzzyF – *Fuzzy Logic Framework*

A ferramenta FuzzyF foi desenvolvida pelo Mestre em Ciência da Computação João Ricardo Bittencourt pela Universidade do Vale do Rio dos Sinos (UNISINOS). A ideia principal

desse projeto foi a elaboração de um conjunto de classes em Java que programa rotinas da lógica difusa (BITTENCOURT; OSÓRIO, 2002). As características do projeto são:

- Código livre e aberto baseado na licença GNU GPL.
- Multiplataforma, desenvolvido em Java.
- Código extensível, permite que outras pessoas possam dar continuidade na implementação.
- Possui documentação de como realizar a integração de sistemas com as classes desenvolvidas.
- Possui exemplos para serem aplicados e testados disponíveis no *site* do autor da ferramenta.
- Possui o modelo de Mamdani implementado.
- Possui três métodos de defuzificação (centro de gravidade, ponto central da área e a média dos máximos).
- Não possui interface gráfica para construção de modelos.
- É indicada apenas para programadores com conhecimentos da lógica difusa.
- A definição do modelo é dada através de um arquivo texto formatado dentro de um padrão definido pelo autor.
- Última atualização disponível para *download* em 26 de junho de 2006.

2.2.2 Fuzzy Logic ToolBox™

Fuzzy Logic ToolBox™² é um módulo do *software* Matlab® desenvolvido pela MathWorks™, empresa fundada em 1984 nos Estados Unidos na cidade Natick, Massachusetts. Esse módulo oferece um conjunto de ferramentas completas para a manipulação de sistemas difusos, tanto para a construção de modelos quanto para a simulação (MATHWORKS, 2009).

A construção de modelos difusos pode ser realizada de três formas: a primeira sendo através de comandos; a segunda, a partir de uma interface gráfica; e a terceira, utilizando diagramas de bloco. A interface gráfica disponibilizada é bastante intuitiva, favorecendo o aprendizado.

O sistema permite que seja exportado código fonte em C para projetos de *software* embarcado. A ferramenta também possibilita realizar simulações de funcionalidades, podendo essas serem acompanhadas de forma visual. É possível configurar o sistema para que os valores correspondentes às entradas do controlador difuso sejam obtidos através de uma interface externa.

Um ponto forte do sistema é a clareza no processo de construção de sistemas difusos. No momento da construção é possível definir os algoritmos e métodos a serem usados no modelo. Um exemplo dessa afirmação seria a possibilidade da escolha do método de defuzzificação, e essa configuração é clara e simples de ser realizada. Outro exemplo é a possibilidade da escolha do tipo de sistema de inferência, sendo que estão implementados os modelos de inferência de Mamdani e Sugeno.

Um fato importante de ser lembrado é a documentação. O guia do usuário é extremamente completo, deixando bem definido a forma de operação do sistema, além de contar com inúmeros exemplos completos e explicados. Atualmente existem aproximadamente 1.200 livros que tratam sobre a ferramenta Matlab® e Simulink®, o que deixa claro o potencial dessa ferramenta.

² - Trademark

O valor da licença que atualmente varia em torno de 2.500 dólares e a necessidade de computadores com alta performance para a operação do sistemas tornam-se fatos inibitórios para uma gama de usuários. Existe a possibilidade da utilização de versões para estudantes, mas essas versões possuem limitações de funcionalidades. Na Figura 19, é apresentada a tela principal do sistema.

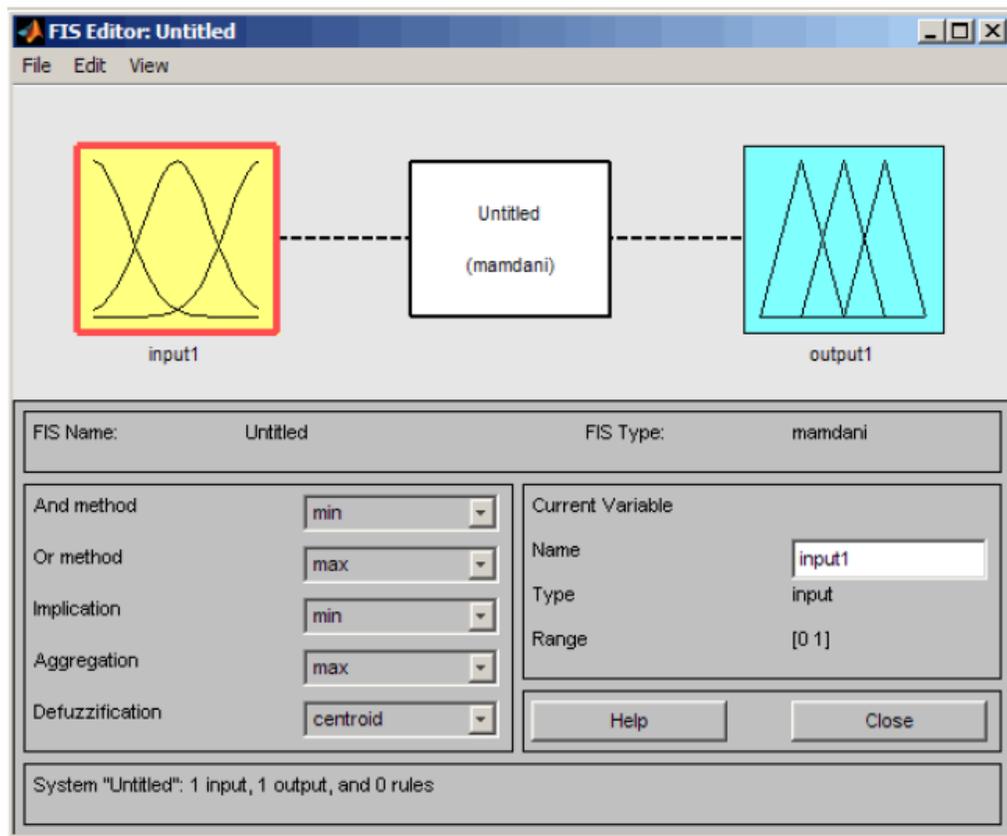


Figura 19 - Tela do *Software Fuzzy Logic ToolBox*TM.

Fonte: Manual Matlab[®] (MATHWORKS, 2009).

2.2.3 Mathematica[®] Fuzzy Logic

Mathematica[®] é um produto desenvolvido pela Wolfram Research, empresa fundada em 1987 por Stephen Wolfram. Trata-se de um *software* de matemática onde é possível adicionar o módulo de lógica difusa (WOLFRAM, 2003).

O sistema desenvolvido pela Wolfram é completo no que diz respeito a funcionalidades, sendo que implementa quase todas, senão todas as rotinas necessárias para a operação com sistemas difusos.

A ferramenta Mathematica[®] possui como forma de construção de modelos difusos apenas comandos, não existindo uma interface gráfica. Esse fato faz com que seja necessário um conhecimento maior por parte do usuário para a utilização dos recursos disponibilizados pelo *software*.

Esse sistema, além de pecar na falta de uma interface gráfica para a construção de modelos, possui como agravante o valor das licenças. Uma licença de uso individual profissional está na faixa dos 2.500 dólares, sem o pacote de lógica difusa. Existem licenças com valores diferenciados para o meio acadêmico, essas licenças são adquiridas por valores mais baixos, porém a licença possui tempo limite de uso. Uma licença individual para estudantes custa na faixa de 70 dólares por ano.

Um ponto forte do sistema é a documentação. O sistema conta com manuais completos com ótima descrição das funcionalidades e com exemplos descritos. Na Figura 20 é apresentada uma tela do *software*.

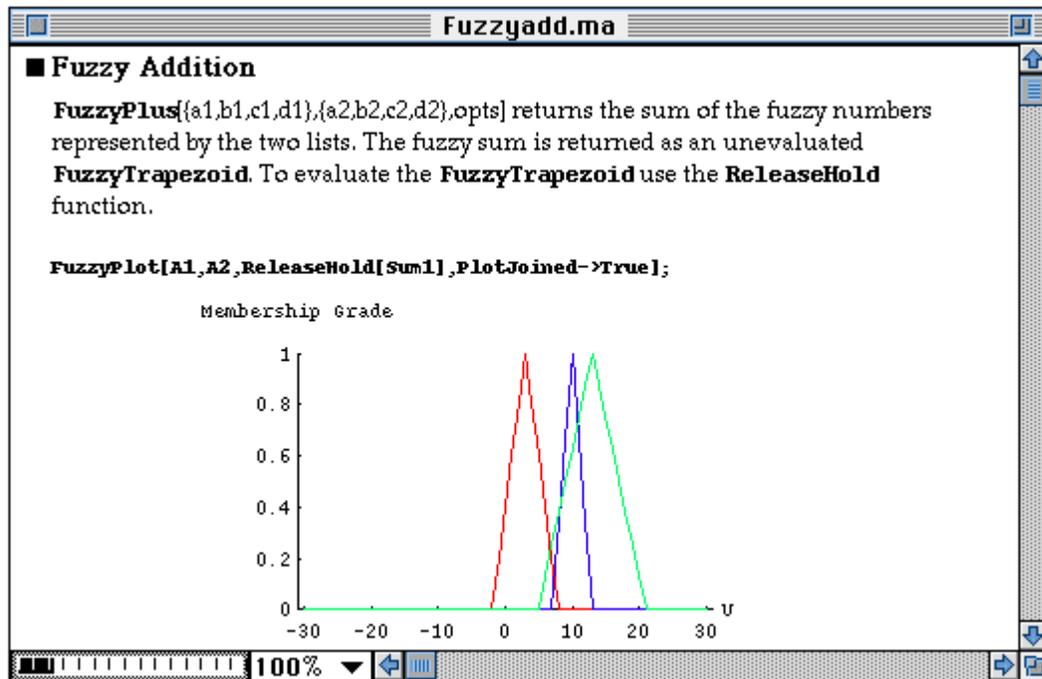


Figura 20 - Tela do *Software Mathematica*®.

Fonte: Site da empresa Wolfram Research (WOLFRAM, 2003).

2.2.4 FIDE™

FIDE™ (*Fuzzy Inference Development Environment*) é uma ferramenta de sistemas difusos desenvolvida pela Apronix, empresa fundada em Tokyo no Japão em 1988 por um grupo de matemáticos e engenheiros. A empresa conta com filiais em outros países (ZHU, 1997).

Essa ferramenta possui duas formas de construção de modelos, sendo uma delas através de comandos e a outra através de uma interface gráfica. Um diferencial dessa ferramenta encontra-se em uma funcionalidade de conferência (*Trace*), essa funcionalidade permite que o sistema exiba todas as operações que são efetuadas durante a simulação, a fim de encontrar e solucionar problemas.

Outro detalhe interessante dessa ferramenta é a funcionalidade do analisador, implementação que permite encontrar erros nas regras de forma visual e, além disso, indicar as regras que as geraram. O sistema também traz a funcionalidade de um simulador, para utilizar

essa funcionalidade basta criar um arquivo que servirá de base para a entrada no controlador difuso. Durante o processo de simulação é possível rastrear prováveis erros.

É possível, através de comando simples gerar código fonte para as seguintes linguagens de programação: *C*, *Assembly*, *Java* e *Matlab*[®]. Pode ser considerado como ponto fraco dessa ferramenta, a falta de documentação, quando comparado com as demais ferramentas, como *Matlab*[®], bem como o fato de não existir uma versão gratuita. A Figura 21 apresenta uma das telas do *software*.

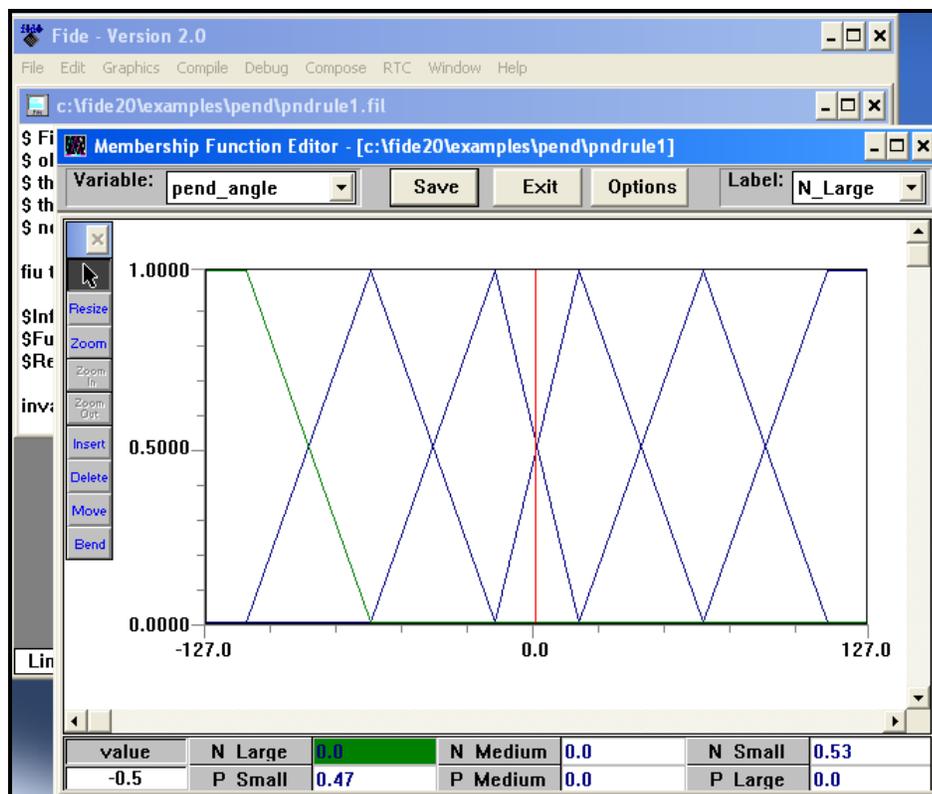


Figura 21 - Tela do *Software* FIDE[™] (*Fuzzy Inference Development Environment*).

Fonte: *Site* do desenvolvedor (ZHU, 1997).

2.2.5 TILShell 3.0

TILShell é uma ferramenta de criação de modelos difusos criada pela empresa Togai InfraLogic, companhia fundada pelo Dr. Masaki Togai em 1987 (TOGAI INFRALOGIC, 2009).

Esse sistema possui a interface com o usuário através de comandos ou interface gráfica. Esse sistema é distribuído em três versões, profissional, padrão e básica. Para utilizar a ferramenta é necessário apenas um computador com configurações de pouca performance.

O sistema apresenta recursos interessantes como: simulação (função que permite que o modelo difuso criado possa ser testado) e geração de código fonte em linguagem C. Essa ferramenta deixa a desejar em vários aspectos, sendo eles: pouca documentação, não deixa explícitas as condições de aquisição da ferramenta, não possui uma versão de demonstração para *download* da internet. Nos exemplos disponíveis, é possível verificar que o sistema permite a definição dos algoritmos de inferência e defuzificação. A Figura 22 apresenta uma das telas do *software*.

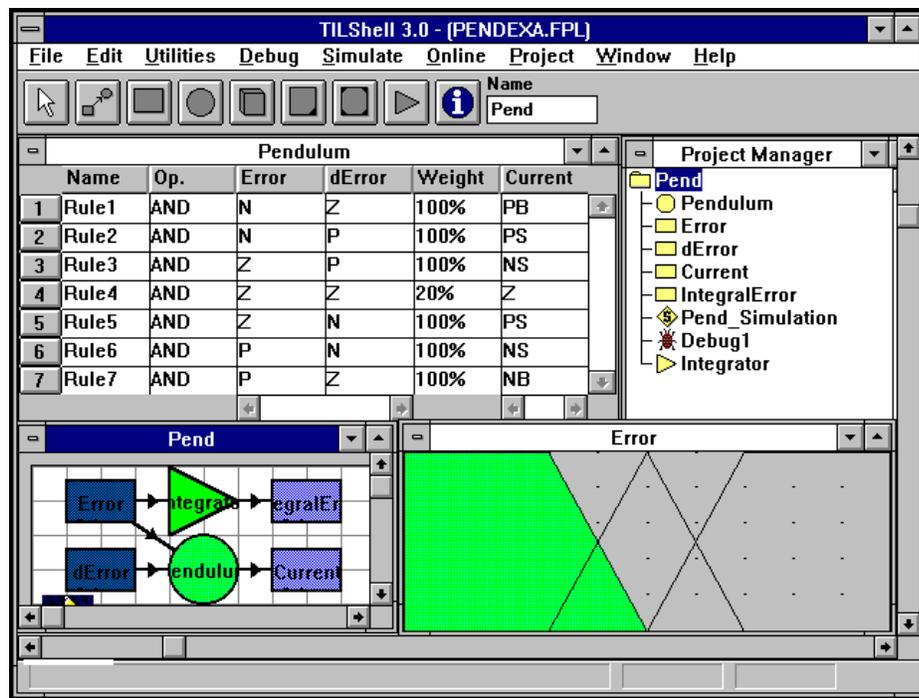


Figura 22 - Tela do *Software* TILShell.

Fonte: *Site* do desenvolvedor (TOGAI INFRALOGIC, 2009).

2.2.6 FuzzyTECH[®]

FuzzyTECH[®] é uma ferramenta de modelagem de sistemas difusos desenvolvida pela INFORM GmbH. Um item que diferencia o sistema FuzzyTECH[®] do restante são as características de ergonomia e usabilidade (FUZZYTECH, 2009).

A criação de controladores difusos é simples, desde a criação das variáveis e os respectivos graus de pertinência até as variáveis de saída e os métodos de defuzzificação. Baseado nas variáveis de entrada e de saída, o sistema facilita a criação das regras através de uma grade previamente preenchida. O sistema disponibiliza ferramentas de análise e simulação com recursos gráficos 2D e 3D, além de permitir integração com periféricos através da porta serial RS232 ou conexões TCP/IP.

Outro diferença dessa ferramenta é o fato de gerar documentação sobre os modelos de sistemas difusos implementados. O sistema conta também com um manual de uso completo.

O sistema é distribuído em diversas versões, onde cada uma possui determinados recursos, permitindo que o cliente possa escolher a versão que mais seja adequada a sua realidade. Existe uma versão de demonstração disponível no *site*, porém essa versão possui limitações de uso, e uma delas é o fato de não permitir a geração de código fonte nas seguintes linguagens: C e Java.

O fato dessa ferramenta não ser gratuita inviabiliza a sua utilização por parte de alguns usuários e até mesmo instituições de ensino.

Na Figura 23 é possível visualizar uma das telas do *software*. Uma das características mais interessantes nesse modelo é a forma fácil de criar os modelos difusos. A maioria das operações ocorre com funções *drag and drop* (funções de arrastar e soltar).

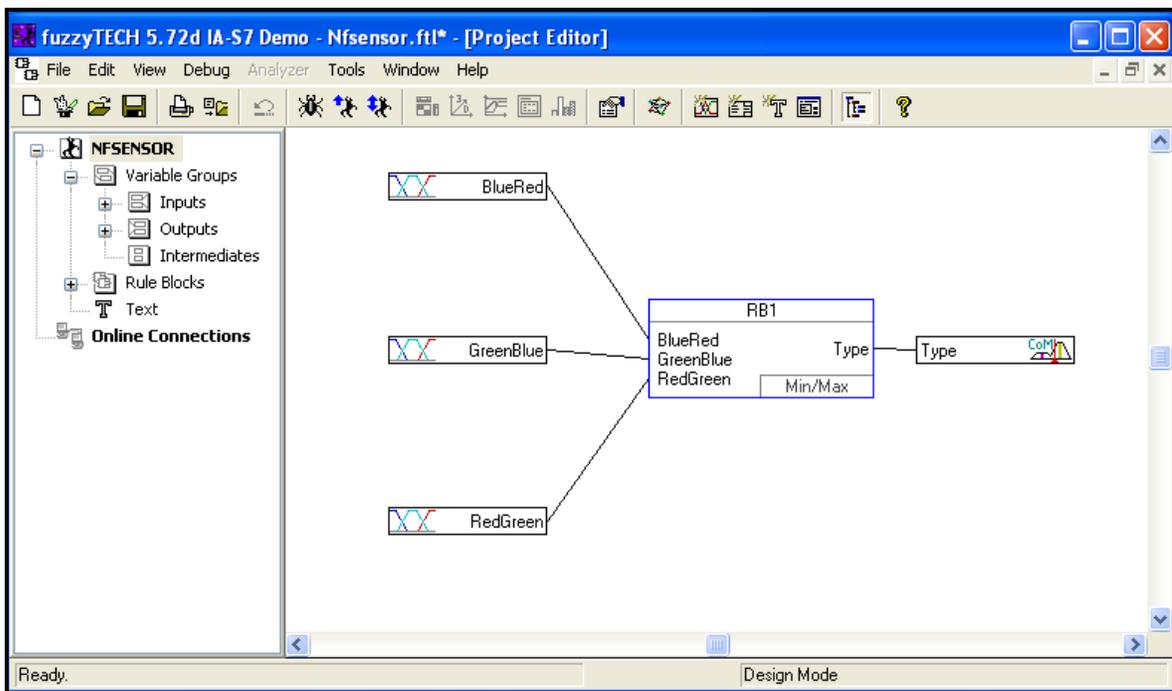


Figura 23 - Tela do Software FuzzyTECH®.

Fonte: Site do desenvolvedor (FUZZYTECH, 2009).

2.2.7 RockOn Fuzzy Tool

RockOn é um *software* de código aberto que implementa funcionalidades da lógica difusa. Esse *software* foi desenvolvido na universidade de Kassel na Alemanha durante um curso de lógica difusa pelos programadores Florian Backmann, Stefan Stützer e Stefan Lind (ROCKON FUZZY TOOL, 2009).

O *software* mantém como principal característica a árvore de projeto no lado direito. Isso facilita a criação de modelos difusos e sua manutenção. O sistema implementa o modelo de inferência Mamdani e os seguintes métodos de defuzificação: centro da gravidade, média dos máximos, maior dos máximos e menor dos máximos. A Figura 24 apresenta a tela principal do software RockOn.

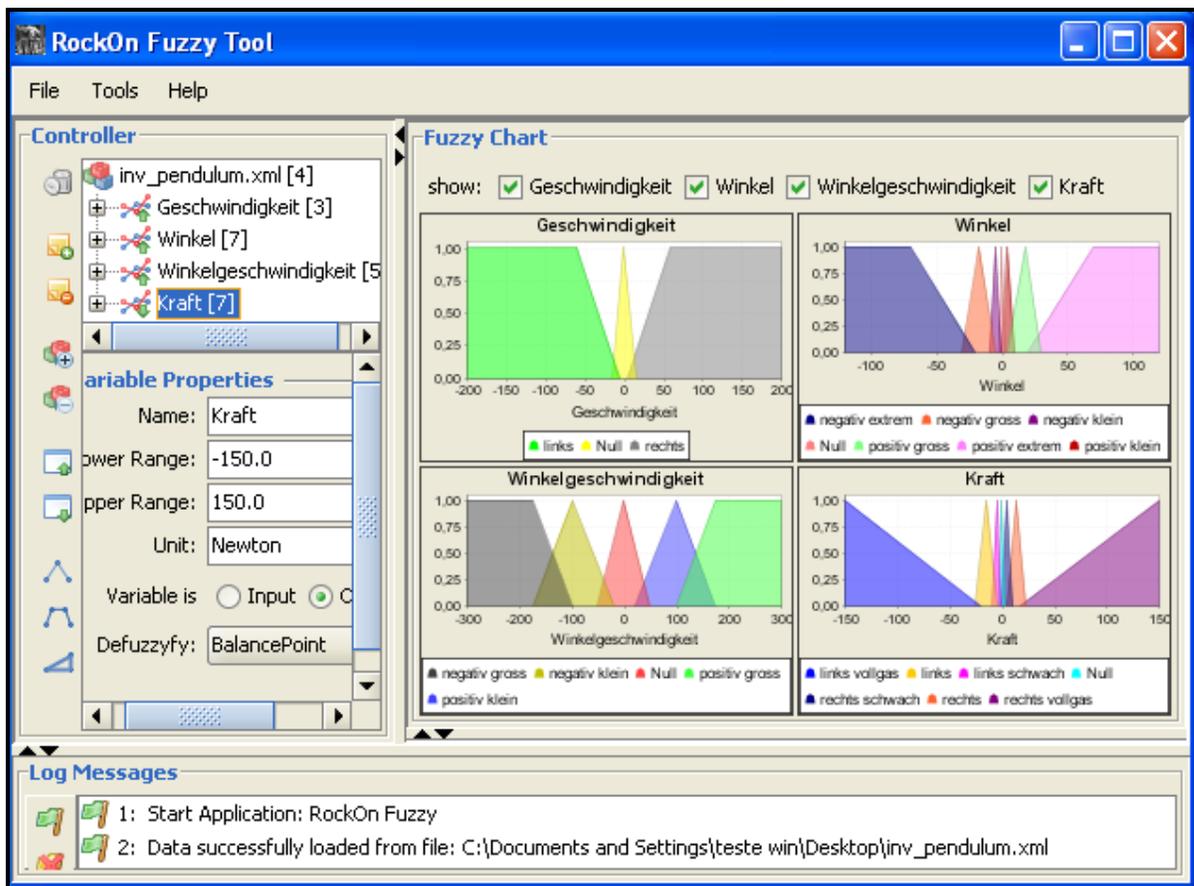


Figura 24 - Tela do *Software* RockOn.

Fonte: *Site* do desenvolvedor (ROCKON FUZZY TOOL, 2009).

Os pontos a seguir foram observados na ferramenta:

- Falta de documentação.
- Apresenta erro na execução do módulo de simulação.
- Interface gráfica agradável.
- Sistema multiplataforma.
- Código fonte aberto.

2.2.8 SciLab Fuzzy Tool

SciLab é um *software* matemático distribuído gratuitamente na *internet* e de código aberto desde 1994. Atualmente é dirigido pelo consórcio SciLab fundado em Maio de 2003 (SCILAB, 2009).

Para que a ferramenta possa fazer o uso de recursos da lógica difusa é necessário realizar o *download* de *plugins*. Esses incrementos no SciLab permitem que o usuário do sistema faça uso de rotinas difusas. Estão implementados os modelos de inferência de Mamdani e Takagi-Sugeno.

Pode se citar como vantagens dessa ferramenta o fato dela ser gratuita e a possibilidade de adicionar novas extensões. Isso quer dizer que um programador pode desenvolver as próprias rotinas e distribuí-las na *internet* para outros usuários do sistema.

Como ponto negativo, fica o fato da extensão disponível não possuir interface gráfica, isso faz com que o usuário tenha que ter um bom conhecimento na lógica difusa para iniciar as atividades com essa ferramenta. Outro ponto fraco é o fato de não ser muito simples a instalação dessa ferramenta e suas extensões, é necessário que o usuário tenha conhecimentos razoáveis em informática para realizar a tarefa de instalação. Na Figura 25 é possível visualizar a imagem de uma das telas do sistema.

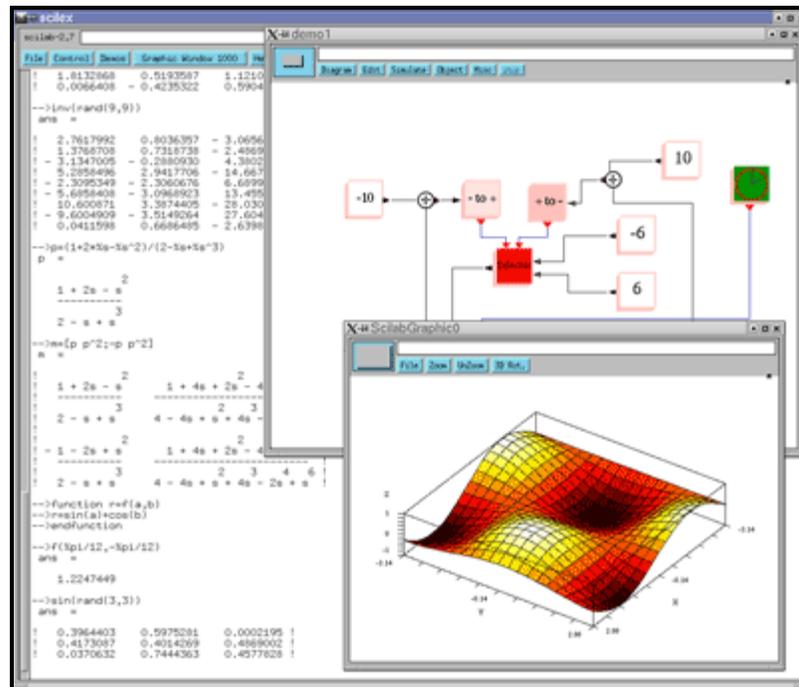


Figura 25 - Tela do *Software* SciLab.

Fonte: *Site* do desenvolvedor (SCILAB, 2009).

2.2.9 UNFUZZY: Sistema de lógica difusa

UNFUZZY (Sistema de lógica difusa) foi desenvolvido pelo professor Óscar Germán Duarte Velasco orientado pelo professor Gustavo Pérez como tese de mestrado em 1997 pela universidade da Colômbia (VELASCO, 2009).

Esse *software* é de distribuição gratuita que implementa o modelo de Mamdani e está disponível em 4 idiomas: espanhol, inglês, italiano e checo. Entre os *softwares* analisados até o momento, somente esse apresenta a característica de estar disponível em mais de um idioma.

O sistema é gráfico e possui boa parametrização das funcionalidades de métodos. Para desenvolver um controlador difuso através dessa ferramenta, basta clicar sobre os principais componentes e ajustá-los conforme necessário para a aplicação. Ao ser executada, essa ferramenta apresentou alguns problemas gráficos, como distorção das imagens.

Os requisitos de *hardware* para executar esse *software* são baixos. Essa ferramenta foi desenvolvida e homologada para ser executada sobre o sistema operacional Windows.

Pode-se citar como vantagens dessa ferramenta o fato de exportar código fonte para C e C++. Além disso, a ferramenta conta com suporte à simulação e depuração dos controladores difusos implementados através da interface gráfica disponibilizada.

A Figura 26 exibe telas do *software* desenvolvido por Velasco. Nessa tela é possível observar na parte superior o controlador difuso na íntegra e abaixo uma tela de parametrização das entradas do sistema.

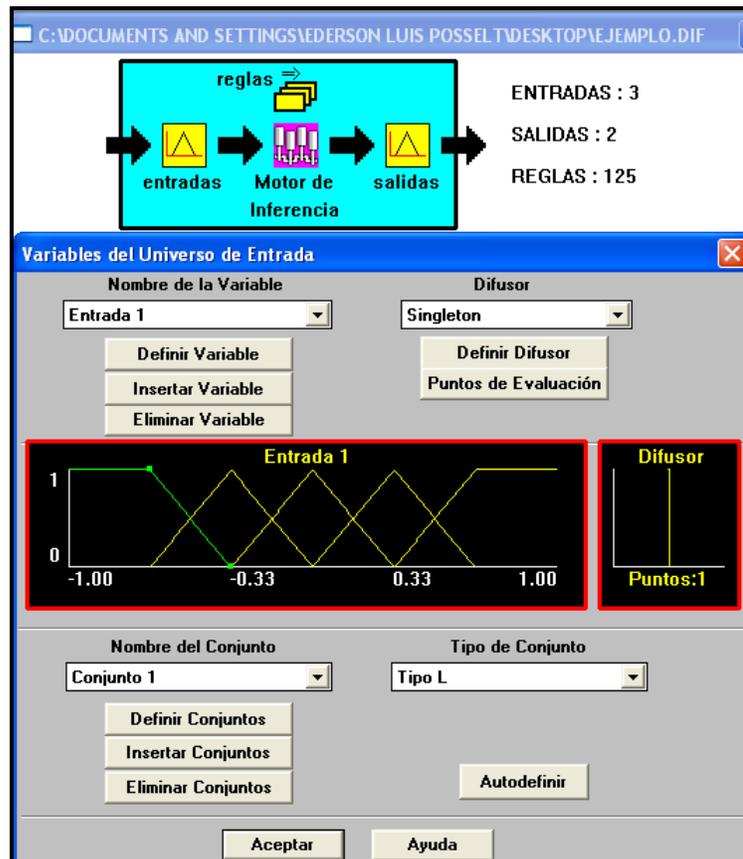


Figura 26 – UNFUZZY Sistema de lógica difusa desenvolvido por Velasco em 1997.

Fonte: *Site* do desenvolvedor (VELASCO, 2009).

2.2.10 XFuzzy

XFuzzy é uma ferramenta livre. A distribuição e modificação são regidas pela licença GNU (*General Public License*). Essa ferramenta pertence aos desenvolvedores e ao IMSE-CNM (Instituto de Microelectrónica de Sevilla – Centro Nacional de Microeletrônica) (IMSE-CNM, 2003).

Esse sistema foi desenvolvido em Java e isso o torna multiplataforma. A Figura 27 demonstra a organização da ferramenta.

- *Description stage*: Destinado à definição do sistema difuso.
- *Verification Stage*: Ferramentas para simulação e monitoramento.
- *Tuning Stage*: Aprendizagem.
- *Synthesis stage*: Funcionalidade de exportação de código fonte para outras linguagens.

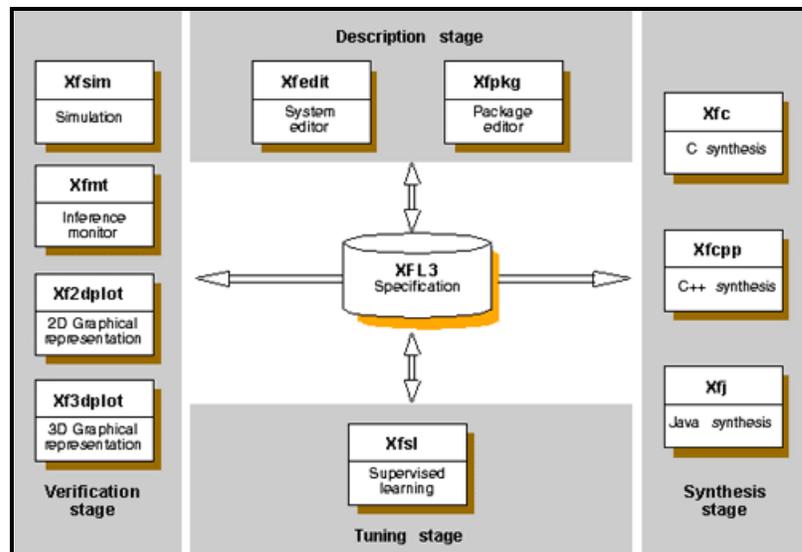


Figura 27 - Organização da ferramenta XFuzzy.

Fonte: Site do desenvolvedor.

Essa ferramenta apresenta vários pontos positivos, entre os aspectos mais favoráveis pode-se citar:

- Documentação completa. Existem vários vídeos no *site youtube* que fazem demonstração da ferramenta.
- Multiplataforma. Pode ser executada em qualquer sistema operacional que possua a máquina virtual Java instalada (JRE).
- Entrada de regras. As regras podem ser criadas de três formas: livre, nesse caso o usuário as cria; em formato de tabela, onde o usuário somente preenche lacuna; e o modo cruzamento, onde o usuário determina a saída para duas variáveis de entrada.
- Exporta código fonte para C, C++ e Java.
- Desenvolve os modelos de inferência de Mamdani e Takagi-Sugeno.
- Implementa 10 métodos de defuzificação.

A Figura 28 apresenta a tela inicial do *software* XFuzzy. Nessa tela, é possível reparar que o menu superior faz as chamadas das funções descritas na Figura 26.

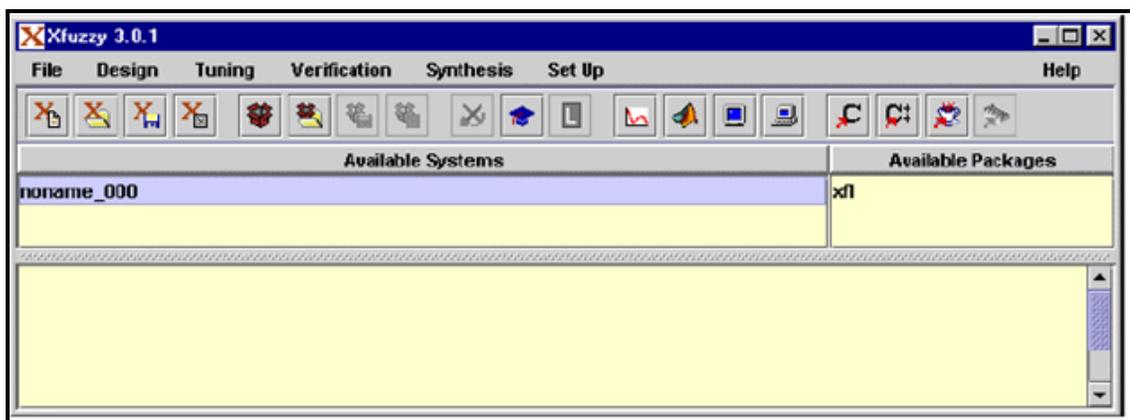


Figura 28 - Tela principal do *software* XFuzzy.

Fonte: Manual do *software* XFuzzy.

2.2.11 Comparativo entre as ferramentas analisadas

Após a análise de várias ferramentas, pode-se chegar a algumas conclusões. A primeira é o fato de existirem vários projetos já implementados no mercado, alguns pagos e outros gratuitos.

A avaliação dos sistemas foi realizada através de um processo investigativo no *site* dos desenvolvedores e nos manuais das ferramentas. Não há tempo hábil para instalar cada uma das ferramentas e realizar a homologação. Nos próximos parágrafos serão abordados os principais itens que diferenciam as melhores ferramentas encontradas.

Entre as ferramentas pagas, o Matlab[®] e o FuzzyTECH[®] apresentaram o maior número de recursos e uma interface gráfica adequada. A ferramenta FuzzyTECH[®] leva vantagem sobre o Matlab[®] no quesito exportação de código fonte e simulação, onde é possível obter entradas externas como: porta serial.

Entre as ferramentas livres, XFuzzy se mostrou a melhor. Essa ferramenta possui os recursos presentes nos *softwares* pagos como Matlab[®] e FuzzyTECH[®], e além disso, possui código fonte aberto, o que permite novas implementações, sendo elas regidas pela licença GNU.

Todavia, nenhuma das ferramentas possui todos os recursos implementados. A seguir, são citados alguns recursos importantes distribuídos entre as ferramentas:

- Somente o FuzzyTECH[®] disponibiliza rotinas de relatório do projeto. Esse relatório é gerado automaticamente durante o processo de montagem do controlador difuso. Esse recurso é interessante para fins de controle e manutenção do sistema.
- O Matlab[®] não possui exportação de código fonte para linguagens de desenvolvimento.
- XFuzzy não gera relatórios dos projetos implementados e também não oferece recursos de entrada de dados para simulação através de interfaces como *sockets* e portas seriais.

- FuzzyTECH[®] não traz transparência nas suas rotinas. Tanto o Matlab[®] quanto o XFuzzy exibem em seus manuais fórmulas e métodos utilizados.

Concluindo, após a análise de várias ferramentas, foi possível observar que todas apresentam pontos fracos e fortes. Baseando-se na análise realizada sobre os *softwares* de lógica difusa, o presente trabalho aborda as boas características evitando os pontos fracos.

O Anexo A apresenta uma tabela comparativa dos sistemas difusos estudados, segundo os seguintes critérios: interface, curvas de fuzificação, métodos de defuzificação, simulação, códigos gerados, licença, modelos e documentação.

Para que o projeto atenda aos conceitos de usabilidade e ergonomia, a próxima seção irá tratar os conceitos dessa técnica e as formas de validação das implementações.

2.3 Ergonomia e usabilidade de interfaces

O sucesso de um sistema está atrelado diretamente com a sua ergonomia e usabilidade. É comum escutar reclamações de usuários referentes à dificuldade de uso e aprendizado de um novo sistema. Um *software* ou até mesmo um eletrodoméstico pode estar dotado das melhores e mais complexas funções, porém se essas funcionalidades não estiverem claras e concisas, os usuários não farão o uso.

Conforme Cybis (2003), usabilidade é característica responsável pela eficácia e eficiência de um *software*. Ergonomia é a aplicação da ciência no que diz respeito a conforto, segurança e eficácia de instrumentos, máquinas e dispositivos.

Levando em consideração a importância da qualidade das interfaces para o presente trabalho, nos próximos tópicos serão abordados os principais conceitos e técnicas. Iniciando com o entendimento de metáforas de *software*, na sequência aprofundando os conceitos de usabilidade de *software* e, por último, verificando as técnicas existentes de avaliação de interfaces.

2.3.1 Metáforas de interfaces

Um dos conceitos mais relevantes em se tratando de interfaces, são as metáforas. As metáforas são utilizadas em interfaces para facilitar a compreensão do usuário quanto às funcionalidades de um sistema. Um exemplo de metáfora que pode ser ilustrado é o sistema de diretórios do sistema operacional *Microsoft Windows*, onde é possível arrastar um determinado conteúdo para dentro de uma pasta. O usuário ao executar essa tarefa fica com a sensação clara de que o arquivo foi movido para a pasta, porém o que aconteceu realmente foi a alteração da estrutura dos índices do sistema operacional (ERICKSON, 1995).

Metáforas mal aplicadas geram dúvidas aos usuários. Para ilustrar um exemplo de metáfora incorreta, é possível descrever a funcionalidade de *eject* do disquete nos computadores Macintosh. Para executar essa tarefa, o usuário necessitava arrastar o ícone do disquete para dentro da lixeira, entretanto, a execução dessa metáfora gerava no usuário a sensação de exclusão dos documentos existentes no disquete.

Para a elaboração de metáforas adequadas, Erickson (1995) sugere que sejam adotadas 3 etapas para essa tarefa. A primeira etapa está relacionada a entender a funcionalidade. A segunda etapa está relacionada a modelar na metáfora as características mais difíceis para o usuário. A terceira etapa consiste em produzir e validar a metáfora.

Contudo, as metáforas devem ser utilizadas nos sistemas, pois facilitam o entendimento do usuário quanto às funcionalidades do mesmo. As metáforas transcrevem situações concretas, já conhecidas e compreendidas pelo usuário para dentro de um *software*.

2.3.2 Usabilidade de interfaces

A usabilidade de interfaces é um requisito para o sucesso ou falha de um projeto de *software*. Conforme NIELSEN (1993), a usabilidade está relacionada a cinco atributos, cada um dos atributos será abordado a seguir.

Facilidade de aprendizado é o atributo mais importante da usabilidade. O sistema deve ser intuitivo no seu uso, fazendo com que o usuário aprenda com a utilização. Na Figura 29, é possível observar um gráfico que exibe curvas de aprendizado em relação ao tempo. Com o sistema projetado para usuários iniciantes, o aprendizado é rápido, porém limitado. Quando o aprendizado é focado para usuários experientes, se obtém resultados mais eficientes, contudo é necessário um tempo maior.

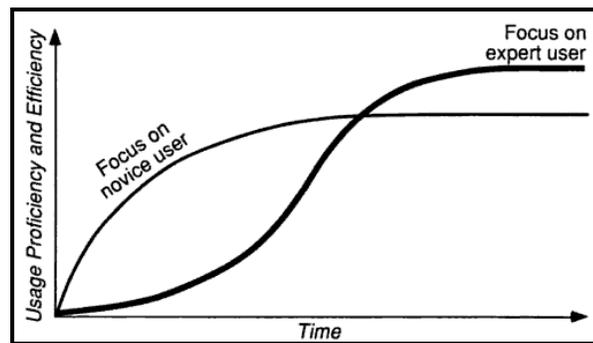


Figura 29 - Relação de aprendizado focado em usuários iniciantes e usuários avançados.

Fonte: NIELSEN, 1993.

O segundo atributo está relacionado à eficiência. Um sistema pode ser considerado eficiente se os usuários experientes (usuários que conheçam o sistema) conseguem realizar as tarefas de forma eficiente. Para verificar se um determinado sistema atende a esse requisito, é necessário monitorar os usuários e avaliar a utilização das funcionalidades do sistema.

O atributo da facilidade de lembrar do sistema é fundamental. O usuário ao utilizar o sistema pela segunda vez não deve passar pelo processo de aprendizado novamente. Esse atributo é importante em sistemas que são usados em períodos espaçados. O *software* de declaração de imposto de renda é um exemplo de sistema utilizado com baixa frequência.

O quarto atributo trata de erros. Nesse caso, erros são classificados como operações onde o usuário não atingiu o seu objetivo. Quando um erro desse nível ocorrer, o usuário deve de forma fácil reverter a situação. Erros graves de grandes perdas não são admissíveis.

E por fim, o quinto e não menos importante atributo, satisfação subjetiva. Esse atributo é pertinente ao quanto o sistema é agradável. Esse atributo tem grande relevância em sistemas externos a instituições, jogos podem ser utilizados como exemplo. A questão de um *software* ser agradável é mais importante quanto a questões de desempenho dentro deste atributo.

2.3.3 Avaliação de interfaces

Desenvolvimento de *software* é muito mais do que simplesmente escrita de códigos. Desenvolver *softwares* requer planejamento e processos. Para a execução de um projeto estão envolvidas técnicas e métodos (ROSSON; CARROLL, 2002).

Avaliar um *software* como etapa única ao final do projeto pode representar uma remodelagem do projeto como um todo. O ideal é que durante as etapas de desenvolvimento ocorram as validações de usabilidade (DIX; et al., 2001).

A melhor forma de saber se um sistema atende aos requisitos é através de avaliação. Para a ergonomia e usabilidade de interfaces isso não é diferente. Existem diversas técnicas para a avaliação da usabilidade. Nos próximos tópicos serão exibidas as seguintes técnicas: *Cognitive walkthrough*, heurística, GOMS e *checklist*. Para cada técnica será abordado o conceito e descrito um exemplo de aplicação.

2.3.3.1 Cognitive walkthrough

Cognitive walkthrough é a técnica que tem por objetivo analisar a facilidade de aprendizado de um *software*, baseando-se nas ações necessárias para realizar uma atividade (DIX; et al., 2001). Para a execução dessa técnica são coletadas informações referentes ao nível do usuário, funções do sistema e a sequência de etapas necessárias para executar as funções. Com base nos dados coletados, avaliadores fazem uma crítica em relação à usabilidade (PREECE; ROGERS; SHARP, 2002). Como vantagens dessa técnica têm-se: ênfase nos problemas do

usuário, ausência de protótipos e usuários durante a avaliação. A desvantagem dessa técnica está no fato de ser muito trabalhosa e não ser aplicável a qualquer tipo de sistema (LUNARDI, 2004).

Para exemplificar a aplicação da técnica *cognitive walkthrough* será abordado o trabalho elaborado por MANO & CAMPOS (2004). Esse trabalho faz a avaliação da aplicação *FLY* (*Software* de registro e manipulação de sumários de aula) desenvolvida na Universidade de Minho em Portugal. A Tabela 2 traz o processo de avaliação, o mesmo não está completo, por motivos de redundância, porém apresenta a clareza do processo de execução.

Tabela 2 - Tabela para exemplificar o processo de avaliação *walkthrough*.

Tarefa	Editar sumário
Ações	<p>a) Clicar o link “Sumário..Editar”.</p> <p>b) Selecionar o sumário a editar com a <i>check box</i> respectiva.</p> <p>c) Clicar o botão “Editar”.</p> <p>d) Preencher os campos que se pretende atualizar.</p> <p>e) Clicar o botão “Atualizar”.</p>
Questões ação A	<p>Questão 1: A ação que o utilizador pretende realizar está disponível na interface? Resposta: Sim.</p> <p>Questão 2: A ação correta é suficientemente evidente para o utilizador? Resposta: Sim, o <i>link</i> é visível no menu da aplicação.</p> <p>Questão 3: Irá o utilizador associar as ações corretas ao que pretende fazer? Resposta: Sim, o texto do <i>link</i> corresponde aos objetivos do utilizador.</p> <p>Questão 4: Irá o utilizador interpretar de forma correta a resposta do sistema à ação escolhida? Resposta: Sim, é apresentada uma página para seleção do(s) sumário(s) a editar.</p>
Questões ação B	<p>Questão 1. Resposta: Sim.</p> <p>Questão 2. Resposta: Provavelmente não para alguns utilizadores. Os utilizadores do tipo B que apenas pretendam editar um sumário poderão sentir-se</p>

	<p>confusos, já que clicar no sumário não o abre para edição, mas apenas o seleciona para edição.</p> <p>Questão 3.</p> <p>Resposta: Não. Na maioria das aplicações para a <i>web</i>, o conceito de hipertexto é amplamente utilizado, e os utilizadores estão habituados a ele. Clicar no texto do sumário ou num botão seria uma solução mais simples e coerente para editar um sumário. É de esperar que os utilizadores do tipo A sejam capazes de ultrapassar esta dificuldade. É duvidoso que os utilizadores do tipo B tenham essa capacidade.</p> <p>Questão 4.</p> <p>Resposta: Neste ponto a resposta não é imediatamente clara. Pelos motivos expostos anteriormente, os utilizadores do tipo B podem não interpretar corretamente a resposta do sistema. Porém, como o efeito esperado do clique numa <i>check box</i> é marcar o controle selecionando, a resposta deve ser sim, já que o sumário selecionado aparece marcado.</p>																																			
<p>Resultado resumido</p> <p>S = Sim</p> <p>N = Não</p> <p>Na = Não se aplica</p>	<table border="1"> <thead> <tr> <th></th> <th colspan="4">Questões</th> </tr> <tr> <th>Ações</th> <th>1</th> <th>2</th> <th>3</th> <th>4</th> </tr> </thead> <tbody> <tr> <th>A</th> <td>S</td> <td>S</td> <td>S</td> <td>S</td> </tr> <tr> <th>B</th> <td>S</td> <td>N</td> <td>N</td> <td>S</td> </tr> <tr> <th>C</th> <td>N</td> <td>N</td> <td>N</td> <td>S</td> </tr> <tr> <th>D</th> <td>S</td> <td>S</td> <td>S</td> <td>S</td> </tr> <tr> <th>E</th> <td>S</td> <td>S</td> <td>S</td> <td>S</td> </tr> </tbody> </table>		Questões				Ações	1	2	3	4	A	S	S	S	S	B	S	N	N	S	C	N	N	N	S	D	S	S	S	S	E	S	S	S	S
	Questões																																			
Ações	1	2	3	4																																
A	S	S	S	S																																
B	S	N	N	S																																
C	N	N	N	S																																
D	S	S	S	S																																
E	S	S	S	S																																

Fonte: Informações coletadas (MANO; CAMPOS, 2004)

O processo de avaliação *Cognitive walkthrough* é trabalhoso. O seu emprego na prática não é aplicável em todas as funcionalidades de um sistema complexo. Após executada a tarefa de avaliação, faz-se necessário gerar um relatório com os problemas encontrados e possíveis soluções.

2.3.3.2 Avaliação heurística

Avaliação heurística foi desenvolvida por Jacob Nielsen e Rolf Molich. Essa técnica consiste em avaliar o *software* com o usuário através de um conjunto de heurísticas. Os dez princípios definidos são: visibilidade do *status* do sistema, compatibilidade entre o sistema e o mundo real, liberdade e controle do usuário, consistência e padrões, prevenção contra erros, reconhecimento em lugar de lembrança, flexibilidade e eficiência de uso, projeto minimalista e estético, auxiliar os usuários a reconhecer, diagnosticar e recuperar-se de erros, ajuda e documentação. Essa técnica exige a participação de vários avaliadores, que ao final se reúnem e consolidam um relatório sobre os problemas encontrados. A vantagem desse método está no fato de qualquer nível de avaliador estar habilitado a executar a checagem do sistema, porém a qualidade do resultado pode ficar atrelada à capacidade dos mesmos (LUNARDI, 2004).

A Tabela 3 é utilizada apenas para exemplificar o processo de avaliação pela técnica de heurísticas. Reparem que são analisadas apenas quatro heurísticas, porém o processo completo leva em consideração os dez princípios.

Tabela 3 - Exemplo de avaliação de heurística.

Heurística	Observação dos avaliadores	Observação dos <i>designers</i>
Visibilidade do <i>status</i> do sistema	a) Não há indicação de que está em uma pasta de aluno. b) Tela Exames: campo docente(s) responsável(is) não haverá como ler o campo. c) O menu flutuante fica mostrado quando o mouse sai de cima dele, mesmo mostrando uma opção não selecionada.	a) OK. Implementado. b) OK. Corrigido. c) OK. Alterado o menu, não possui mais sub menu flutuante.
Compatibilidade entre o sistema e o mundo real	-----	-----
Liberdade e	a) Falta opção para listar alunos por ordem	a) OK. Implementado.

controle do usuário	alfabética ou outros itens. b) Até chegar na pasta do aluno o usuário tem que passar por muitas telas sequenciais.	b) OK. Alterada a estrutura do sistema para diminuir o número de telas intermediárias.
Consistência e padrões	a) Botão Cancelar faz a função de voltar, o que foi difícil perceber.	a) Não foi alterado, pois a equipe não concordou com a opinião dos avaliadores.

Fonte: *Internet* (<http://www.cnptia.embrapa.br/~luciana/disciplinas/mo645/avaliacao.html>)

2.3.3.3 GOMS

GOMS (*Gols, Operators, Methods, Seletion rules*) é o modelo de avaliação elaborado por Card, Moran e Nenwell em 1983. Essa técnica baseia-se nos passos necessários para realizar uma determinada tarefa no sistema. Depois de descrito o objetivo, os métodos, os operadores e as regras, é possível extrair informações de performance sem que o usuário tenha que realizar os procedimentos (PREECE; ROGERS; SHARP, 2002). Existem variações da técnica de GOMS, entre elas: CMNGOMS, KLM, NGOMSL e COM-GOMS. Como vantagem desse método pode-se citar a possibilidade de execução sem a presença de usuários, e como desvantagem a dificuldade de mapear o conjunto de ações em sistemas flexíveis (LUNARDI, 2004).

A Tabela 4 traz um exemplo de aplicação da avaliação pela técnica de GOMS. Observando a tabela, é possível perceber a dificuldade de realizar esse processo em um sistema complexo.

Tabela 4 - Exemplo de avaliação GOMS

Objetivo	Transformar em negrito uma determinada palavra de uma frase.
Operadores	Movimentar o mouse até a palavra desejada. Duplo clique do mouse sobre a palavra.
Método	Método: Utilização da opção do menu - Mover o mouse até a opção Formatação

	- Localizar a opção Negrito - Clicar na opção negrito Método: Utilização do teclado - Manter pressionar a tecla CTRL - Pressionar a tecla “N” (Negrito) - Soltar a tecla CTRL
Seleção de Regras	Utilizar opção “Menu” se usuário pouco experiente Utilizar opção “Teclado” se usuário experiente

Fonte: Material didático de Mário Rui Gomes e Joaquim Jorge, disciplina de IHC

2.3.3.4 Checklist

Checklist são listas de verificação de ergonomia de *software*. Através dessas listas, profissionais como programadores e analistas podem executar a validação da interface sem a necessidade de especialistas. O sucesso deste tipo de avaliação está mais relacionado à qualidade da lista de verificação do que ao avaliador. Os resultados das avaliações costumam ser mais regulares uma vez que os profissionais seguem a lista de verificação (CYBIS, 2003).

Conforme Cybis (2003), a utilização de listas de checagem apresenta as seguintes potencialidades:

- Não exige que a avaliação seja feita por especialista em IHC. Isso se deve ao fato do conhecimento estar embutido no *checklist*.
- Garantia de um resultado mais coerente, mesmo executado por avaliadores diferentes.
- Facilidade de identificação de problemas de usabilidade.
- Eficácia, devido à redução da subjetividade.
- Baixo custo, pois se trata de um método de rápida aplicação.

Uma lista de checagem pode ser encontrada em <http://www.labiutil.inf.ufsc.br/ergolist/>. Ergolist é um *software* de validação da ergonomia e usabilidade de *software online* desenvolvido em uma parceria entre SoftPólis, núcleo Softex-2000 de Florianópolis, e o LabIUtil, Laboratório de Utilizabilidade.

Na Figura 30 é possível observar a tela do *software* Ergolist. É interessante ressaltar que essa lista de checagem oferece as opções de resposta “Sim”, “Não” e “Não aplicável”. Também é possível adicionar comentários a cada questão presente na lista. Na lateral direita da tela, pode-se verificar que existe uma referência contextualizada para cada questão. Ao final de uma avaliação o *software* Ergolist gera um laudo sobre a usabilidade e ergonomia do *software*.

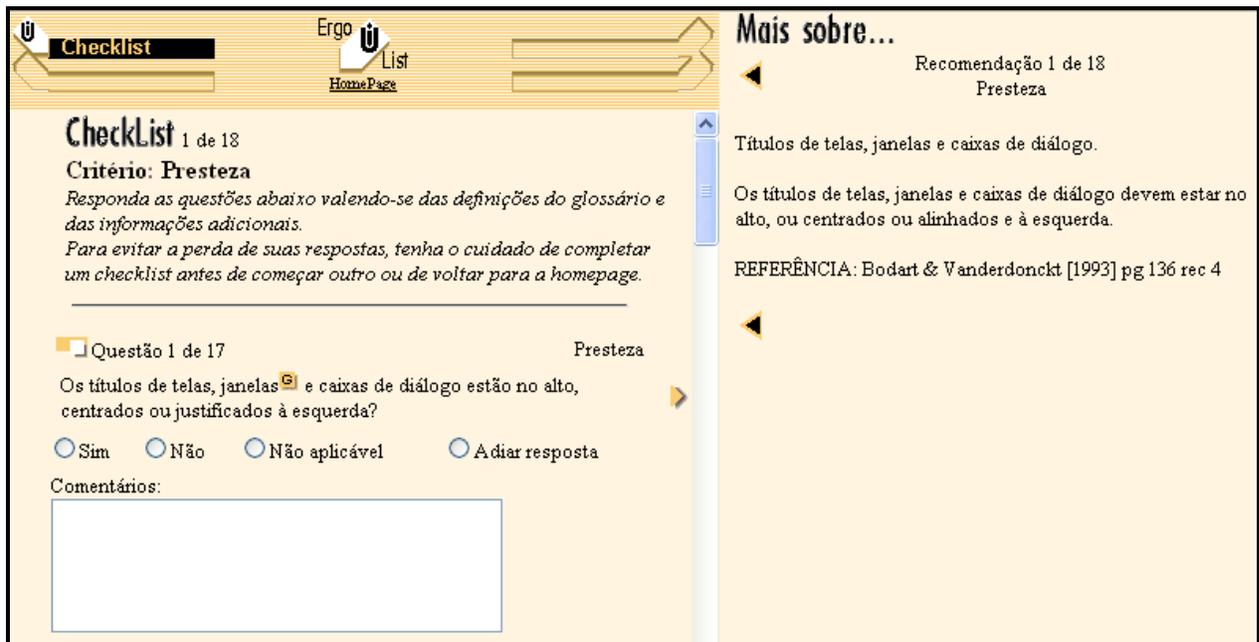


Figura 30 - Tela do *software* Ergolist.

Fonte: <http://www.labiutil.inf.ufsc.br/ergolist/>

Na próxima seção será apresentada a técnica utilizada no trabalho e as argumentações para o seu uso.

2.3.3.5 Definição da técnica a ser utilizada

Todas as técnicas de avaliação da usabilidade e ergonomia estudadas nos tópicos anteriores apresentam vantagens e desvantagens como já mencionado. O fato mais importante diz respeito ao fato de que o sistema tem de ser submetido à avaliação para que seja um sistema de qualidade, como forma de atender as normas de usabilidade e ergonomia.

A técnica de *checklist* foi adotada neste trabalho. Essa escolha se deve ao fato da facilidade de aplicação e execução por programadores e analistas. É importante salientar que existem listas de checagem bem elaboradas disponíveis de forma gratuita. O *checklist* é aplicado durante o processo de desenvolvimento e não somente ao final do projeto. O ANEXO B exibe as questões de checagem aplicadas, extraídas do *software* Ergolist.

Com a execução do processo de avaliação por *checklist* obteve-se como resultado um *software* padrão dentro das normas oficiais de ergonomia e usabilidade, facilitando o aprendizado do usuário.

3 METODOLOGIA

Este capítulo apresenta a metodologia utilizada nesta pesquisa, descrevendo os passos e procedimentos utilizados para a realização dos objetivos propostos.

3.1 Métodos de pesquisa

Este tipo de estudo exige inicialmente a elaboração de uma problemática de pesquisa onde a partir de então, demanda um levantamento de dados a ser pesquisado na literatura (livros, periódicos, monografias, dissertações, teses, anais de eventos impressos, eletrônicos/digitais, entre outras fontes) (MARCONI; LAKATOS, 2007).

Para sucesso da pesquisa, considera-se fundamental que o pesquisador imponha limites no estudo, definindo nitidamente o problema de pesquisa, os fenômenos e o contexto a serem pesquisados. Identificando as distintas concepções, fundamentos, as correntes teóricas, os autores mais importantes, bem como posicionar-se claramente em relação a cada um deles (MARCONI; LAKATOS, 2007).

Assim, o problema de pesquisa deste trabalho contempla a construção de uma nova ferramenta para o desenvolvimento de sistemas difusos, trazendo funcionalidades satisfatórias que auxiliam o usuário desde a compreensão da técnica até a modelagem para a solução de problemas específicos com a possibilidade de integração através do protocolo UDP/IP.

Portanto, este estudo se enquadra, segundo os fins, em uma pesquisa aplicada, ou seja, pesquisa que objetiva gerar conhecimentos para aplicação prática dirigidos para soluções de problemas específicos (GIL, 1991); de natureza descritiva-exploratória, uma vez que visa à pesquisa e exploração, baseada em dados bibliográficos para a criação de uma ferramenta de desenvolvimento de sistemas difusos.

Segundo Cervo, Bervian (1996, p. 49),

“... a pesquisa descritiva procura descobrir, com a precisão possível, a frequência como um fenômeno ocorre, sua relação e conexão, com os outros, sua natureza e características, correlacionando fatos ou fenômenos sem manipulá-lo...”.

Gil (1991, p. 45) concorda com tal afirmativa, destacando que a pesquisa descritiva adota como objetivo primordial a descrição das características de determinada população ou fenômeno.

Em relação à pesquisa de natureza exploratória, Gil (1991) afirma que esta forma de pesquisa é realizada sobre um problema ou questão que geralmente são assuntos com pouco ou nenhum estudo anterior a seu respeito. O objetivo desse tipo de estudo é procurar padrões, ideias ou hipóteses. A ideia não é testar ou confirmar uma determinada hipótese. As técnicas tipicamente utilizadas para a pesquisa exploratória são estudos de caso, observações ou análises históricas, e seus resultados fornecem geralmente dados qualitativos ou quantitativos. A pesquisa exploratória avaliará quais teorias ou conceitos existentes podem ser aplicados a um determinado problema ou se novas teorias e conceitos devem ser desenvolvidos.

3.2 Síntese dos procedimentos metodológicos

Para melhor compreensão dos procedimentos propostos e utilizados, apresenta-se a seguir uma síntese do passo a passo que foi realizado para o desenvolvimento dessa pesquisa. Para a sua execução, foram realizados os seguintes procedimentos metodológicos:

- Pesquisa bibliográfica: Levantamento de referencial teórico. O objetivo foi fundamentar a importância e as contribuições que as pesquisas nesta área proporcionam para novas ideias de aprimoramento e melhorias científicas e práticas.
- Avaliação de ferramentas existentes: Esta avaliação foi realizada através das funcionalidades apresentadas por cada ferramenta.

- Aplicação de regras de ergonomia e usabilidade: Como a ferramenta será visual, é importante ao usuário que a interface da ferramenta seja compreensível e de fácil utilização. Para isso, foram contemplados aspectos de ergonomia e usabilidade.
- Projeto da ferramenta levando em consideração os parâmetros de entrada, os métodos utilizados para o processamento de um sistema difuso e os parâmetros de saída. Implementação da ferramenta: Este passo envolveu a definição da linguagem de programação utilizada e o próprio desenvolvimento da ferramenta.
- Teste e validação da ferramenta desenvolvida: Foi realizada a interação prática com usuários para uso da ferramenta, a fim de coletar informações referentes à sua utilidade, funcionalidades e interface.
- Comparação da ferramenta desenvolvida com ferramentas existentes no mercado: A partir de critérios definidos para a realização da comparação.
- Aplicação da ferramenta em pelo menos um projeto: Essa etapa prevê que o sistema seja utilizado na indústria ou integrado a um sistema já existente com a finalidade de comprovar a sua eficácia.

4 MODELAGEM E DESENVOLVIMENTO DA FERRAMENTA INFUZZY

Após a realização de pesquisas sobre lógica difusa, ferramentas de modelagem existentes, ergonomia e usabilidade de *software*, foi possível desenvolver uma ferramenta que atendesse grande parte dos usuários, denominada de InFuzzy. A ferramenta será apresentada através das seguintes seções:

- 4.1 Modelagem da ferramenta InFuzzy.
- 4.2 Aplicação das características de ergonomia e usabilidade de *software*.
- 4.3 Funcionalidades da ferramenta InFuzzy.
- 4.4 *Ferramenta* InFuzzy.
- 4.5 Estrutura do arquivo gerado pelo sistema InFuzzy.
- 4.6 Protocolo de comunicação UDP.
- 4.7 Homologação da ferramenta InFuzzy.
- 4.8 Processo de instalação da ferramenta InFuzzy.
- 4.9 Projetos implementados na ferramenta InFuzzy.

4.1 Modelagem da ferramenta InFuzzy

Essa seção é destinada a apresentar o diagrama de uso, a estrutura de dados e o fluxograma do projeto. Através desses itens é possível gerar uma visão macro do sistema, esclarecendo as suas principais funcionalidades e a forma de interação e ordem lógica de construção de um sistema difuso.

A Figura 31 apresenta o diagrama de caso de uso referente à ferramenta InFuzzy. Pela Figura 31 é possível observar a interação do usuário com as principais funcionalidades do sistema, sendo elas:

- Definir preferências: Permite ao usuário definir as configurações referentes ao sistema difuso, como: operador AND, operador OR, implicação, defuzificação e agregação.
- Definir variáveis de entrada e os termos linguísticos: Configuração das variáveis de entrada e os seus respectivos termos linguísticos.
- Definir variáveis de saída e os termos linguísticos: Configuração das variáveis de saída e os seus respectivos termos linguísticos.
- Definir regras: Configuração das regras que atuarão no sistema difuso.
- Realizar simulação: Definição do valor das variáveis de entrada para a simulação do sistema difuso.
- Restaurar Projeto: Possibilidade de reabrir um projeto, recarregando as variáveis do sistema a partir de um arquivo.
- Salvar projeto: Persistir as informações do sistema em um arquivo.

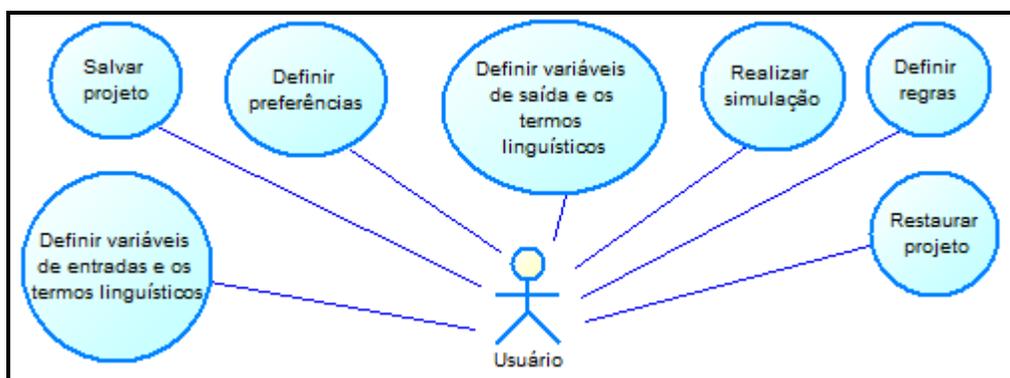


Figura 31- Diagrama de caso de uso

Fonte: do autor

A Figura 32 apresenta a estrutura dos dados aplicada a ferramenta InFuzzy. Através do modelo de dados é possível reconhecer a organização dos dados nos seguintes níveis de relacionamento:

- Uma variável pode conter vários termos linguísticos, porém um termo linguístico deve estar relacionado a uma e somente uma variável.
- Um bloco de regras pode possuir várias regras, porém uma regra deve estar relacionada a um e somente um bloco de regras.
- Um bloco de regras pode possuir várias conexões, porém uma conexão deve estar atrelada a somente um bloco de regras.
- Uma simulação pode possuir várias “simulações variáveis”, porém uma “simulação variável” deve estar vinculada a uma e somente uma simulação. “Simulação variável” é a estrutura utilizada para organizar as variáveis de entrada utilizadas na simulação.

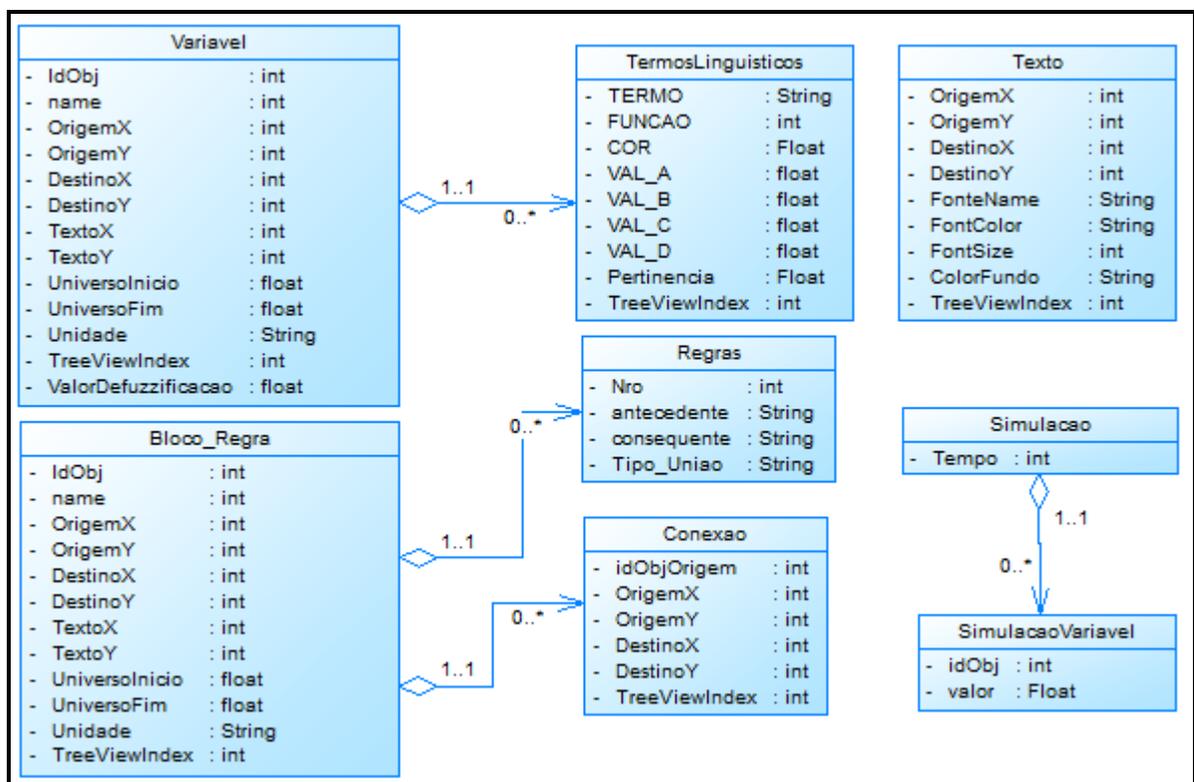


Figura 32- Modelagem da estrutura dos dados

Fonte: do autor

A Figura 33 apresenta um fluxograma de uso da ferramenta. Com a ferramenta InFuzzy em execução, a primeira opção no fluxograma está relacionada a reabrir um projeto ou criar um novo projeto difuso. Com o projeto difuso aberto, é possível executar diversas tarefas sem sequência pré-definida como:

- Adicionar variáveis de entrada.
- Adicionar variáveis de saída.
- Editar regras.
- Simular projeto.

Com as alterações efetuadas, o projeto pode ou não ser persistido. Caso o usuário opte por não salvar as alterações, essas serão descartadas.

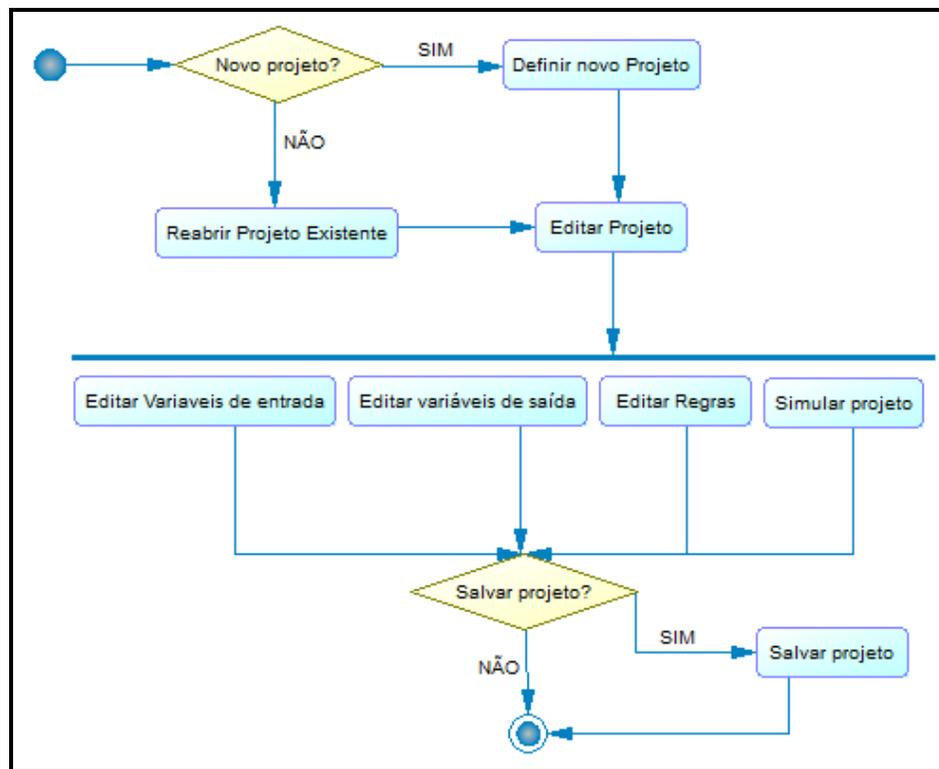


Figura 33- Fluxograma InFuzzy

Fonte: do autor

4.2 Aplicação das características de ergonomia e usabilidade de *software*

O sucesso de um projeto está fortemente relacionado a sua interface gráfica e consequentemente à ergonomia e usabilidade. Como forma de garantir uma interface adequada ao projeto, realizou-se o estudo de várias técnicas de avaliação de interfaces, sendo elas:

- *Cognitive walkthrough*: Técnica que tem por objetivo analisar a facilidade de aprendizado de um *software*.
- Avaliação heurística: Técnica que consiste em avaliar o *software* com o usuário através de um conjunto de heurísticas.
- GOMS: Técnica que se baseia nos passos necessários para realizar uma determinada tarefa no sistema.
- *Checklist*: Listas de verificação de ergonomia de *software*.

A seção 2.3 do capítulo 2, sobre ergonomia e usabilidade de interfaces descreve a diferença entre as técnicas e exemplifica a sua aplicação. Após a análise dos métodos, optou-se pelo método de *checklist* pelos motivos listados a seguir:

- Não exige que a avaliação seja feita por especialista em IHC. Isso se deve ao fato do conhecimento estar embutido no *checklist*.
- Garantia de um resultado mais coerente mesmo executado por avaliadores diferentes.
- Facilidade de identificação de problemas de usabilidade.
- Eficácia, devido à redução da subjetividade.
- Baixo custo, pois se trata de um método de rápida aplicação.

A Tabela 5 apresenta o resumo da aplicação da lista de checagem baseada no *software Ergolist* (ERGOLIST, 2009). As questões estão divididas em grupos. Cada grupo possui um total de questões que estão divididas entre “Atende”, “Não atende” e “Não se aplica”. No ANEXO B, é possível visualizar a aplicação da lista na íntegra. Lembrando que, as questões para avaliação foram obtidas no *software Ergolist* e a aplicação dessas questões foi realizada pelo autor do projeto.

Tabela 5- Tabela resumida da aplicação do método *checklist* baseado em questões do Ergolist

Grupo de questões	Atende	Não atende	Não se aplica	Total de questões
Presteza: Verifica se o sistema informa e conduz o usuário durante a interação.	12	2	3	17
Agrupamento por localização: Verifica se a distribuição espacial dos itens traduz as relações entre as informações.	8	0	3	11
Agrupamento por formato: Verifica os formatos dos itens como meio de transmitir associações e diferenças.	12	1	4	17
Legibilidade: Verifica a legibilidade das informações apresentadas nas telas do sistema.	18	1	8	27
Feedback: Avalia a qualidade do <i>feedback</i> imediato às ações do usuário.	7	4	1	12
Concisão: Verifica o tamanho dos códigos e termos apresentados e introduzidos no sistema.	11	0	3	14
Ações Mínimas: Verifica a extensões dos diálogos estabelecidos para a realização dos objetivos do usuário.	5	0	0	5
Densidade Informacional: Avalia a densidade informacional das telas apresentadas pelo sistema.	6	0	3	9
Ações Explícitas: Verifica se é o usuário quem comanda explicitamente as ações do sistema.	4	0	0	4

Controle do Usuário: Avalia as possibilidades do usuário controlar o encadeamento e a realização das ações.	4	0	0	4
Flexibilidade: Verifica se o sistema permite personalizar as apresentações e os diálogos.	1	1	1	3
Experiência do Usuário: Avalia se usuários com diferentes níveis de experiência têm iguais possibilidades de obter sucesso em seus objetivos.	1	3	2	6
Proteção contra erros: Verifica se o sistema oferece as oportunidades para o usuário prevenir eventuais erros.	3	2	2	7
Mensagens de erro: Avalia a qualidade das mensagens de erro enviadas aos usuários em dificuldades.	6	2	1	9
Correção de erros: Verifica as facilidades oferecidas para que o usuário possa corrigir os erros cometidos.	0	3	2	5
Consistência: Avalia se é mantida uma coerência no projeto de códigos, telas e diálogos com o usuário.	8	1	2	11
Significados: Avalia se os códigos e denominações são claros e significativos para os usuários do sistema.	10	1	1	12
Compatibilidade: Verifica a compatibilidade do sistema com as expectativas e necessidades do usuário em sua tarefa.	13	2	6	21
TOTAL	129	23	42	194

Fonte: do autor

A Figura 34 apresenta o resultado apresentado na Tabela 5 em formato de gráfico levando em consideração as questões atendidas, não atendidas e as questões que não se aplicavam ao *software*. Desta forma observa-se que 66% das questões aplicadas são atendidas pelo sistema, 12% não são atendidas e 22% das questões não são aplicáveis.

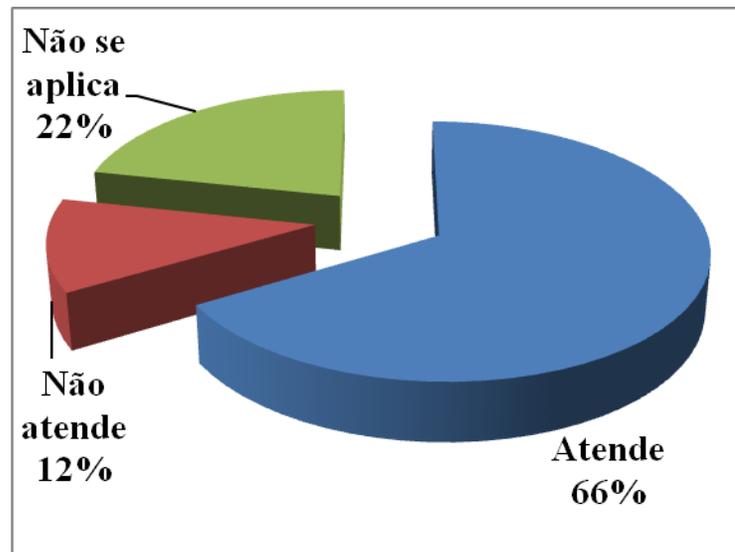


Figura 34- Resultado da aplicação da lista de checagem

Fonte: do autor

Para possuir uma visão mais real sobre a aplicação das listas de checagem, é importante considerar apenas os itens aplicáveis ao sistema. Computando-se apenas os itens aplicáveis obtém-se o gráfico apresentado na Figura 35. Através desse gráfico é possível observar que 85% das questões de ergonomia são atendidas e apenas 15% não são atendidas.

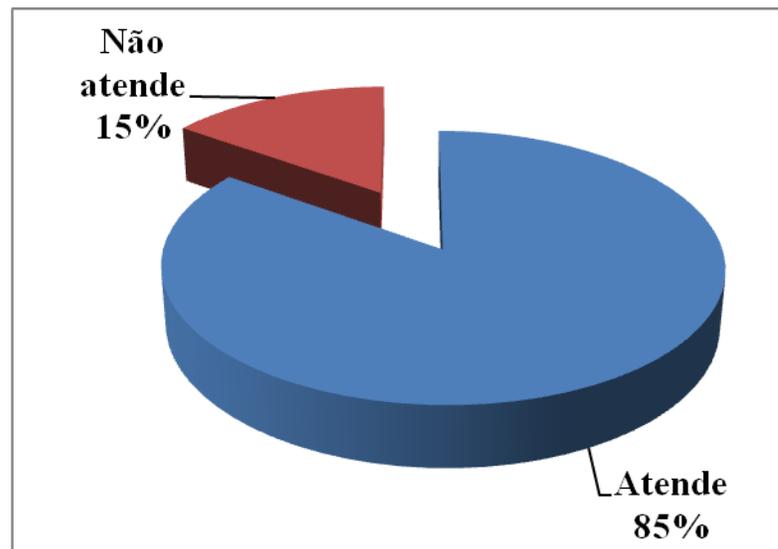


Figura 35- Resultado aplicação *checklist*, itens aplicáveis

Fonte: do autor

Uma das vantagens apresentadas pelas listas de checagens é a capacidade de manutenção sobre os itens pendentes. Atender em 100% as questões do *checklist* é uma tarefa trabalhosa, uma vez que essa demanda programação e cuidados específicos. A utilização de uma IDE (*Integrated Development Environment*) facilita esse trabalho, uma vez que na sua essência estão inseridos vários itens de usabilidade e ergonomia.

4.3 Funcionalidades da ferramenta InFuzzy

A ferramenta foi desenvolvida em Delphi para a plataforma Microsoft nas versões XP, Vista e *Seven*. Essa escolha leva em conta o fato do sistema operacional da Microsoft ser o mais utilizado no mundo. Esse *software* também funciona no sistema operacional Linux, executado pelo *software* Wine (sistema capaz de executar aplicações Windows em sistemas operacionais Linux), atingindo a grande maioria dos usuários de computadores. Na Figura 36 é apresentada a tela inicial do sistema InFuzzy. As principais características da tela inicial são: menu principal de acesso a todas as funções de controle de projetos, barra de acessos rápidos e informações sobre o sistema na parte inferior.

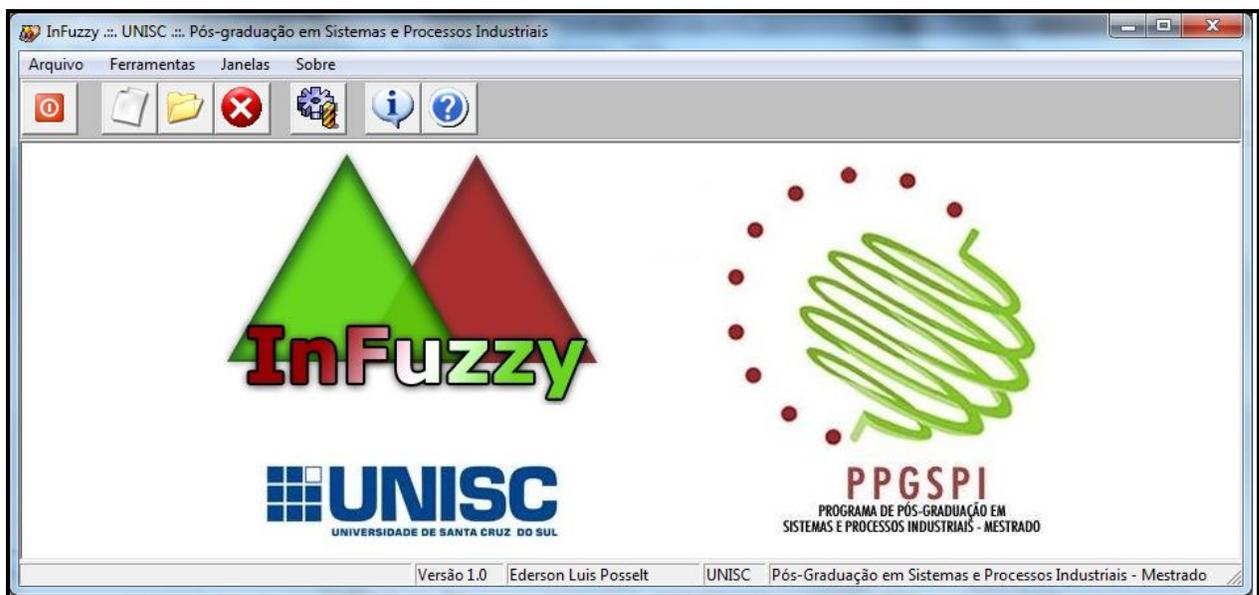


Figura 36- InFuzzy - Ferramenta de desenvolvimento de sistemas difusos.

Fonte: Do autor, extraída da ferramenta InFuzzy.

A Tabela 6 apresenta as possibilidades de configuração da ferramenta para o desenvolvimento de um sistema difuso. As funções de pertinência são aplicadas aos termos linguísticos de cada variável, seja ela de entrada ou saída. As relações entre as regras podem ser estabelecidas pelo operador OR ou AND, sendo que para cada um desses operadores é possível determinar uma função a ser aplicada. A implicação é a função destinada a determinar a forma como o resultado dos antecedentes ocorre sobre os consequentes. Na etapa de agregação é possível selecionar uma função para determinar como o resultado de cada regra influencia o resultado de pertinência da variável de saída. A última configuração é referente ao número de passos, essa configuração determina a precisão dos resultados de defuzificação, pois é ela quem informa o deslocamento dentro do universo. Quanto menor o valor do passo, maior a precisão e maior a necessidade computacional. Percebe-se a diferença nessa configuração quando existe o uso de funções não lineares como: gaussiana, sigmóide e seno.

Tabela 6- Tabela com os opcionais de configuração do sistema difuso

Configuração	Opções
Funções de pertinência	Triangular Trapézio Gaussiana <i>Cauchy</i> Sigmóide Rampa Direita e Esquerda Inclinação Direita e Esquerda Discreto
AND	Mínimo (Norma-T) Produto Algébrico (Norma-T) Produto Limitado (Norma-T)
OR	Máximo (Conorma-T) Soma Algébrica (Conorma-T) Soma Limitada (Conorma-T)
Implicação	Mínimo (Norma-T)
Agregação	Máximo (Conorma-T) Soma Algébrica (Conorma-T) Soma Limitada (Conorma-T)
Defuzificação	Centro da Gravidade Primeiro dos Máximos Centro dos Máximos Último dos Máximos

	Média dos Máximos Altura
Passo	0,01; 0,10; 0,50 1,00; 2,00; 5,00 10,00

Fonte: Dados obtidos da ferramenta InFuzzy.

A ferramenta InFuzzy permite a criação de modelos de inferência Mamdani (mínimo para implicação e máximo para agregação), porém, podem ser definidas variações a partir do momento em que podem ser definidas outras funções para os métodos de agregação.

Nas próximas seções serão abordadas individualmente as funcionalidades de cada parte do sistema, dessa forma elucidando o uso da ferramenta.

4.4 Ferramenta InFuzzy

Nesta seção será apresentado a ferramenta InFuzzy de forma a abordar todas as funcionalidades envolvidas no processo de modelagem de um sistema difuso, desde a criação de um modelo até a etapa de simulação dos processos difusos, passando pelas seguintes etapas:

- Criação de um modelo.
- Definições iniciais do projeto.
- Modelagem de um projeto difuso.
- Edição de variáveis de entrada e saída.
- Edição de regras.
- Edição de preferências.
- Edição de texto.
- Edição de simulação.

Para criar um sistema difuso, basta selecionar a opção “Arquivo/ Novo”. A execução dessa etapa é apresentada na Figura 37. Também é possível acessar essa funcionalidade através do botão “Novo” presente na barra de acesso rápido. Observa-se a presença de três pontos na opção “Novo...”, que indica que haverá uma nova interface de diálogo após a escolha dessa opção.

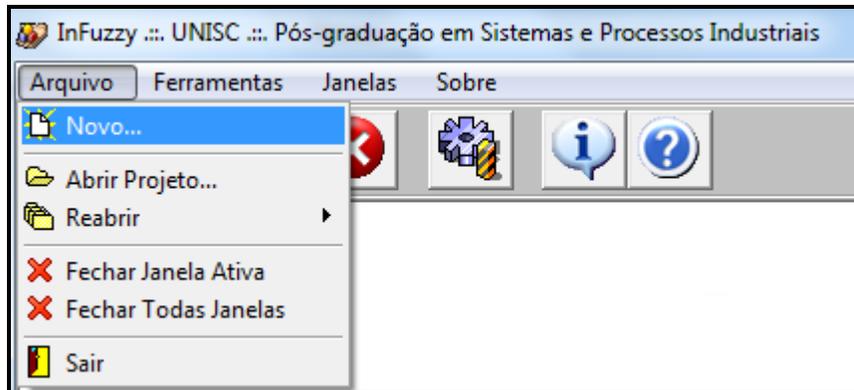


Figura 37- Menu de seleção de novo projeto

Fonte: Do autor, extraída da ferramenta InFuzzy.

Depois de selecionada a opção “Novo...”, será apresentada uma janela onde devem ser informados três parâmetros, sendo eles: “Título”, “Autor” e “Descrição”. Essa janela pode ser visualizada na Figura 38. Esses campos são livres e o seu uso é destinado à identificação do projeto.

A imagem mostra a janela "Formulário de criação de Projeto". Ela contém três campos de texto: "Título" com o valor "Gorjeta", "Autor" com o valor "Manual MatLab" e "Descrição" com o valor "Sistema que visa a determinação do valor da gorjeta do cliente em percentual". Na base da janela, há dois botões: "Confirmar..." com um ícone de marca de seleção verde e "Cancelar" com um ícone de X vermelho.

Figura 38- Janela de novo projeto.

Fonte: Do autor, extraída da ferramenta InFuzzy.

Após as definições iniciais do projeto, o *software* apresenta a tela de modelagem de projetos difusos. A descrição das funcionalidades desta tela é apresentada na seção a seguir.

4.4.1 Modelagem de projetos difusos

A construção dos modelos difusos ocorre através de interface gráfica, que está de acordo com as normas de ergonomia e usabilidade apresentadas na fundamentação teórica deste projeto. A determinação das entradas, saídas e blocos de regras são realizados por operações de *drag and drop* (arrastar e soltar). Esses componentes serão ligados através de linhas. A união de todos os componentes na tela dá origem ao controlador difuso modelado pelo usuário.

A Figura 39 apresenta a tela de construção do projeto, sendo essa, a tela mais importante do sistema, pois é através dessa que ocorrem todas as manipulações sobre variáveis de entrada, saída, bloco de regras e mensagens. O sistema recebe duas variáveis de entrada, onde se conectam ao bloco de regras. Por sua vez, este está conectado a uma variável de saída.

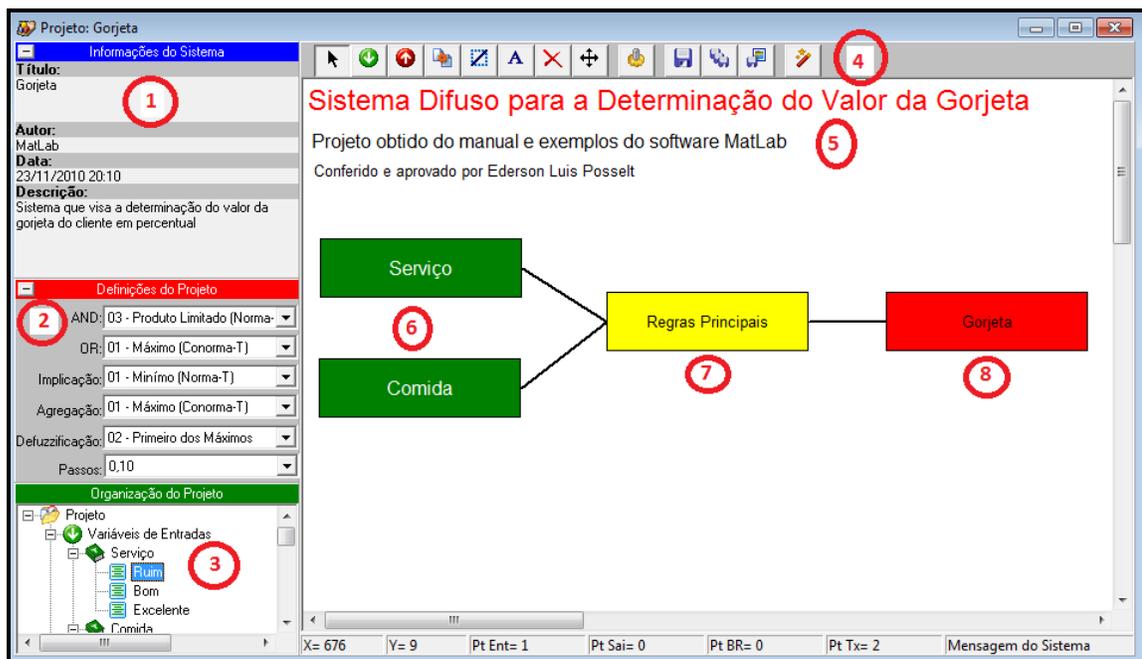


Figura 39- Tela de controle sobre o projeto difuso.

Fonte: Do autor, extraída da ferramenta InFuzzy.

Para um maior entendimento da interface apresentada na Figura 39, a Tabela 7 apresenta de forma detalhada os principais componentes e sua descrição.

Tabela 7- Tabela com informações sobre a tela de modelagem de projetos difusos

Componente	Descrição	Informações Extras
1	Informações referentes ao projeto.	As informações presentes são: Título, Autor, Data e Descrição.
2	Configuração do projeto.	Nesse espaço são realizadas as opções de controle do projeto. Todas as escolhas são realizadas através de uma seleção. As possíveis configurações são: AND, OR, Implicação, Agregação, Defuzificação e Passos.
3	Estrutura do projeto.	O sistema utiliza uma estrutura de árvore para permitir que o usuário tenha uma visão geral do projeto.
4	Barra de ferramentas.	<p>Através dessa barra de ferramentas é possível construir sistemas difusos. A seguir é apresentada a descrição de cada componente.</p> <ul style="list-style-type: none">  Seleção de componentes.  Adição de variáveis de entrada.  Adição de variáveis de saída.  Adição de bloco de regras.  Conexão entre variáveis de entrada e saída com blocos de regras.  Adição de texto ao projeto.  Exclusão de componentes do projeto.  Ferramenta para movimentar componentes da tela.  Configuração de preferências do usuário.

		 Salvar projeto.  Salvar como projeto.  Salvar uma imagem do projeto.  Iniciar o modo de simulação.
5	Ambiente de desenvolvimento.	Essa área do sistema é utilizada para a modelagem do sistema difuso.
6	Variáveis de entrada	Os blocos verdes indicam as variáveis de entrada.
7	Bloco de regras	O bloco amarelo indica o bloco de regras.
8	Variáveis de saída	O bloco vermelho indica a saída.

Fonte: Do autor, extraída da ferramenta InFuzzy.

A Figura 40 apresenta a árvore de estruturação do projeto difuso implementado. Através dessa estruturação, o usuário consegue ter uma visão global e identada sobre os componentes utilizados no projeto. Observando essa estrutura, facilmente se reconhece as variáveis de entrada e os seus termos linguísticos, as variáveis de saída e seus termos linguísticos, os blocos de regras e suas conexões com as variáveis de entrada e saída e, por último, as mensagens de texto criadas para identificar o projeto. É importante ressaltar que, além de permitir visualizar a estrutura do projeto, o usuário pode realizar operações através do clique do botão direito do mouse, chamando um menu de ações, como pode ser visto na Figura 40.



Figura 40- Árvore da estrutura do projeto difuso modelado.

Fonte: Do autor, extraída da ferramenta InFuzzy.

A partir da tela de projetos, são acionadas as demais telas que permitem configurar os parâmetros do sistema difuso. As próximas seções tratam de detalhar cada uma das telas envolvidas no processo, iniciando pela tela de edição de variáveis de entrada e saída.

4.4.2 Tela de edição de variáveis de entrada e saída

As configurações referentes às variáveis de entrada e saída ocorrem pela mesma tela. Nessa tela ocorrem definições como: nome da variável, universo, unidade e termos linguísticos.

A Figura 41 apresenta a tela de edição de variáveis de entrada e saída. A chamada dessa tela pode ser realizada por três eventos, sendo eles:

- Duplo clique sobre a caixa da variável de entrada ou saída presente na área de modelagem do projeto;
- Clique com o botão direito sobre a caixa da variável de entrada ou saída presente na área de modelagem do projeto e na sequência selecionar a opção “editar”;
- Clique com o botão direito sobre a variável desejada na árvore de estrutura do projeto e na sequência selecionar a opção “Editar”.

Os próximos tópicos, numerados de 1 a 5 fazem uma explanação dos principais componentes dessa tela:

1. A primeira configuração a ser realizada é a configuração do nome da variável, o universo dessa variável e o tipo de unidade. O nome da variável é empregado no controle das regras, o universo é utilizado na persistência da criação dos termos linguísticos e a unidade é aplicada ao eixo das abscissas dos gráficos, conforme normas de ergonomia.
2. Lista dos termos linguísticos criados pelo usuário. É possível reparar que as cores desta lista refletem nas cores das funções apresentadas no item 4. As operações

possíveis nesta etapa são: adição de termos linguísticos, edição de termos linguísticos e exclusão de termos linguísticos. As operações realizadas nos termos linguísticos refletem nas regras criadas no bloco de regras da seguinte forma: a edição de um termo linguístico faz a edição do nome da variável em todas as regras; já a exclusão, elimina todas as regras que contenham esse termo. Para uma maior segurança nessa etapa, o sistema apresenta diálogos de confirmação para o usuário.

3. Essa etapa é utilizada para a definição das variáveis. Após a configuração do tipo de função a ser utilizada, o sistema ajusta os parâmetros de entrada, isso quer dizer que se for uma função triangular, o sistema vai solicitar a entrada de três valores; no caso de uma função trapezoidal, o sistema irá solicitar quatro valores de entrada. O próximo parâmetro a ser informado é o nome do termo linguístico e a cor desejada para a diferenciação entre as demais funções. Para concluir a etapa, é necessário salvar as edições realizadas. Na parte inferior, o sistema apresenta a fórmula da função escolhida. É importante ressaltar que os parâmetros informados são validados em relação ao universo informado e às condições impostas pelas próprias funções.
4. Essa área é destinada à apresentação gráfica das funções. Através desta, o usuário consegue visualizar com maior facilidade as funções de pertinência criadas por ele.
5. Barra de funções da tela, sendo elas: largura da linha (utilizada para determinar a largura da linha quando o gráfico é apresentado em 2D); 3D (possibilidade de apresentar o gráfico em 3D. A barra localizada ao lado permite o ajuste do nível de 3D; Imprimir (essa função permite a impressão do gráfico presente na tela); Salvar BMP (função destinada a gerar um *bitmap* do gráfico).

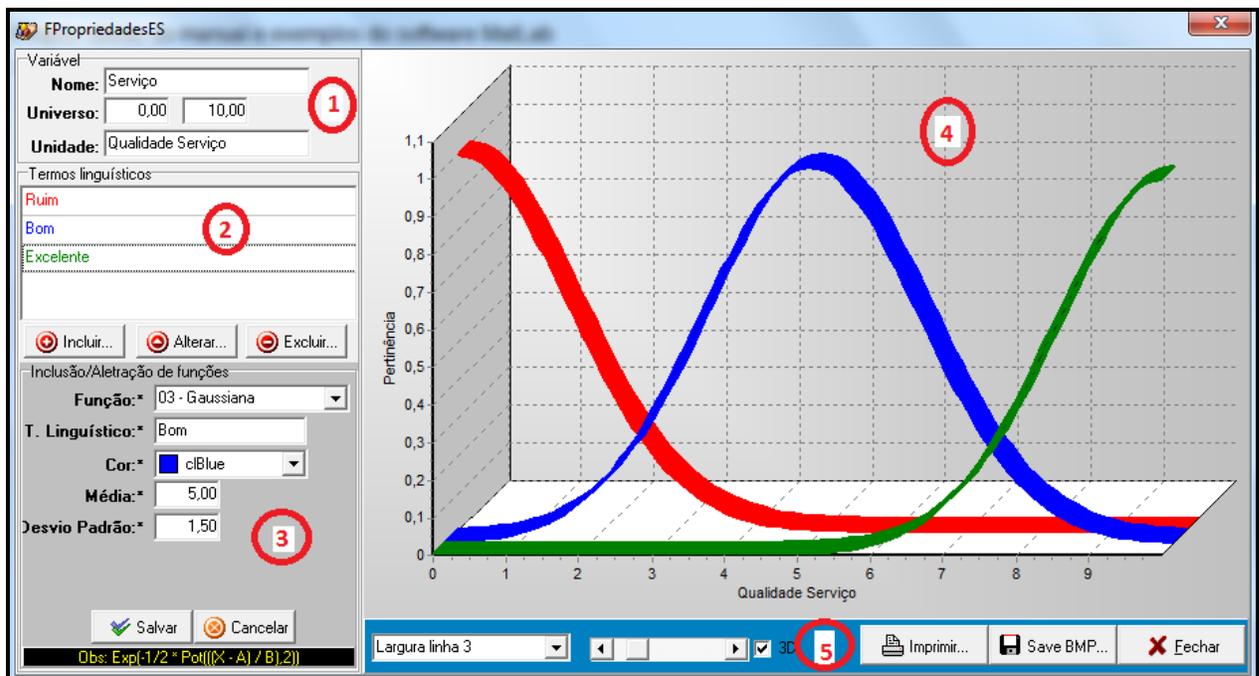


Figura 41- Tela de edição de variáveis de entrada e saída.

Fonte: Do autor, extraída da ferramenta InFuzzy.

Com as variáveis de entrada e saída criadas e configuradas, a próxima etapa é a definição das regras. Na seção a seguir é apresentada a tela responsável por permitir as operações de inclusão, edição e exclusão de regras.

4.4.3 Tela de edição de regras

As regras fazem parte da etapa de inferência de um sistema difuso. A definição das regras do projeto ocorre através da tela apresentada na Figura 42. As descrições das funcionalidades dessa tela estão divididas em oito etapas, conforme a separação apresentada na figura.

1. Nome do bloco de regras: A ferramenta permite a criação de mais de um bloco de regras, sendo assim, é sugerido que o usuário crie um nome para cada bloco de regras.
2. Antecedentes: São criados de forma dinâmica em virtude da associação das entradas aos blocos de regra. Para cada variável de entrada é carregada uma lista com os

termos linguísticos. Na figura é possível verificar que as duas variáveis de entrada são: “Serviço” e “Comida”.

3. Consequentes: São criados de forma dinâmica obedecendo a mesma teoria dos antecedentes. Observa-se que apenas uma saída está conectada a esse bloco de regras, sendo ela: “Gorjeta”.
4. Conexão: Tipo de conexão entre os antecedentes. Para cada regra o sistema permite apenas a escolha de um tipo de conexão. É importante lembrar que a função aplicada pelo operador é configurada na tela inicial do projeto.
5. Permite duas opções: “Adicionar Regra” (através dessa opção o sistema grava uma nova regra na base) e “Editar Regra” (edita uma determinada regra selecionada pelo usuário). O sistema faz o controle da criação e edição de regras a fim de verificar a integridade dos parâmetros e a duplicidade de regras.
6. Lista de regras geradas: Todas as regras geradas pelo sistema são apresentadas nesta lista. Para editar uma regra basta selecioná-la e na sequência, realizar o ajuste necessário e clicar no botão “Editar Regra”. Nessa etapa existe um facilitador, se a necessidade for criar uma nova regra semelhante a uma já existente, basta selecioná-la e em seguida fazer os ajustes, porém, ao invés de clicar no botão “Editar regra” utiliza-se o botão “Adicionar Regra”.
7. Existem duas possibilidades de excluir regras, a primeira é através do botão “Excluir Regra”, onde o usuário deve selecionar a regra a ser escolhida e na sequência utilizar a função. Uma caixa de diálogo será aberta para confirmar a exclusão. A outra maneira é através do botão “Excluir todas as regras”. A utilização desse método realiza a exclusão de todas as regras, porém é necessário que o usuário confirme essa ação através de uma caixa de diálogo.
8. Essa é a última etapa da definição das regras. Para que as regras realmente sejam salvas no projeto é necessário que seja utilizado o botão “Salvar”. Isso se deve pelo fato de que as regras são manipuladas em uma área fora dos dados do projeto e quando acionada a opção salvar, essas regras são copiadas para a estrutura de dados

do projeto. Caso não se deseje salvar nenhuma das alterações ou inclusões, basta utilizar o botão “Cancelar”.

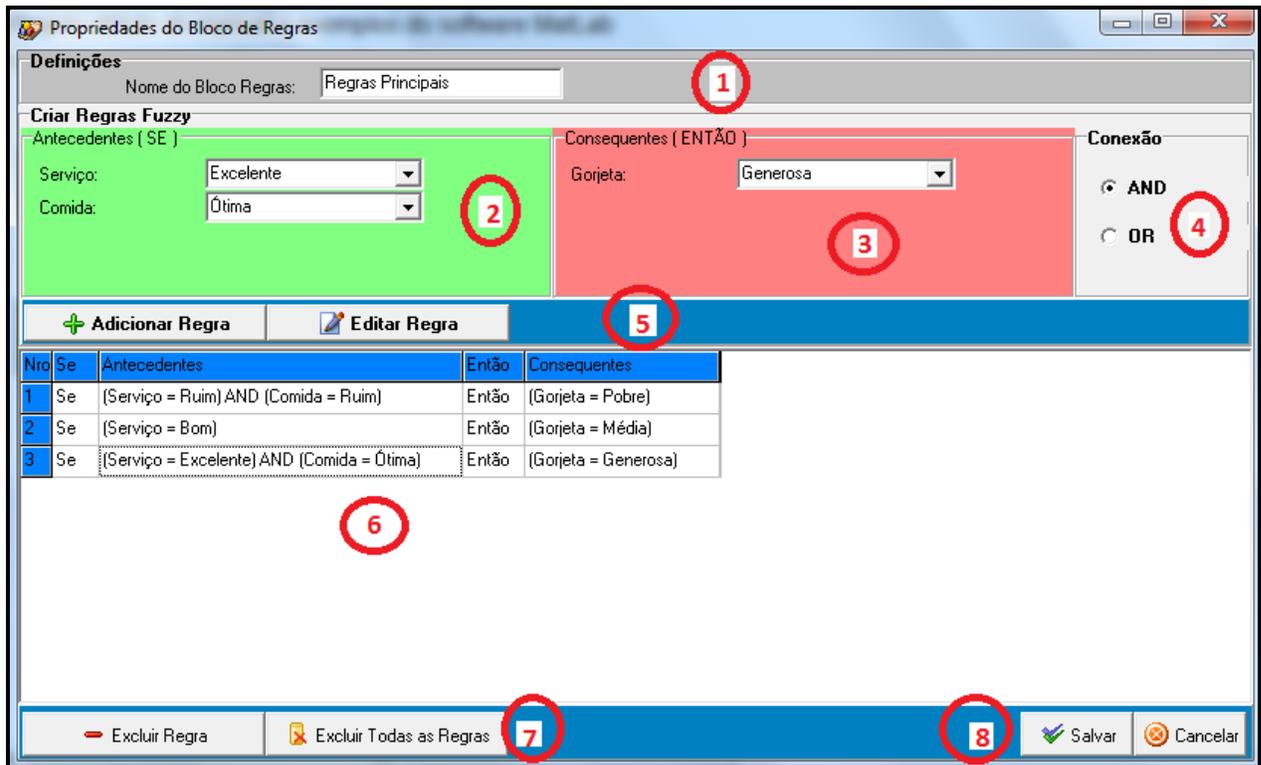


Figura 42- Tela de definição das regras

Fonte: Do autor, extraída da ferramenta InFuzzy.

Com as regras e variáveis definidas, o usuário do sistema possui a oportunidade de configurar as suas preferências do projeto. Na próxima seção é apresentada a descrição da tela de preferências.

4.4.4 Tela de configuração de preferências

A ferramenta proposta permite a configuração de cores e fontes dos modelos, isso se deve ao fato dos usuários sentirem a necessidade de personalizar os projetos de acordo com suas vontades. A tela que permite essa personalização é apresentada na Figura 43 e o seu entendimento será explicado em cinco etapas.

1. Configurações do componente de entrada: As configurações possíveis são largura da borda, cor da borda, cor de preenchimento, tamanho da fonte, tipo de fonte e cor da fonte.
2. Configurações do componente de saída: As configurações possíveis são largura da borda, cor da borda, cor de preenchimento, tamanho da fonte, tipo de fonte e cor da fonte.
3. Configurações do componente de bloco de regras: As configurações possíveis são largura da borda, cor da borda, cor de preenchimento, tamanho da fonte, tipo de fonte e cor da fonte.
4. Configuração de transições: Trata-se da configuração das linhas que conectam as variáveis aos blocos de regras. É possível configurar a largura e a cor dessa linha.
5. O botão restaurar permite ao usuário restaurar as configurações padrões do sistema; isso pode ser aplicado no seguinte caso: Um determinado professor recebe de um aluno um projeto todo personalizado de forma exagerada, com apenas um clique o professor pode alterar o leiaute para o seu padrão desejado. A opção “Salvar” persiste a nova definição de leiaute do projeto. Essa informação ficará salva no arquivo do projeto.

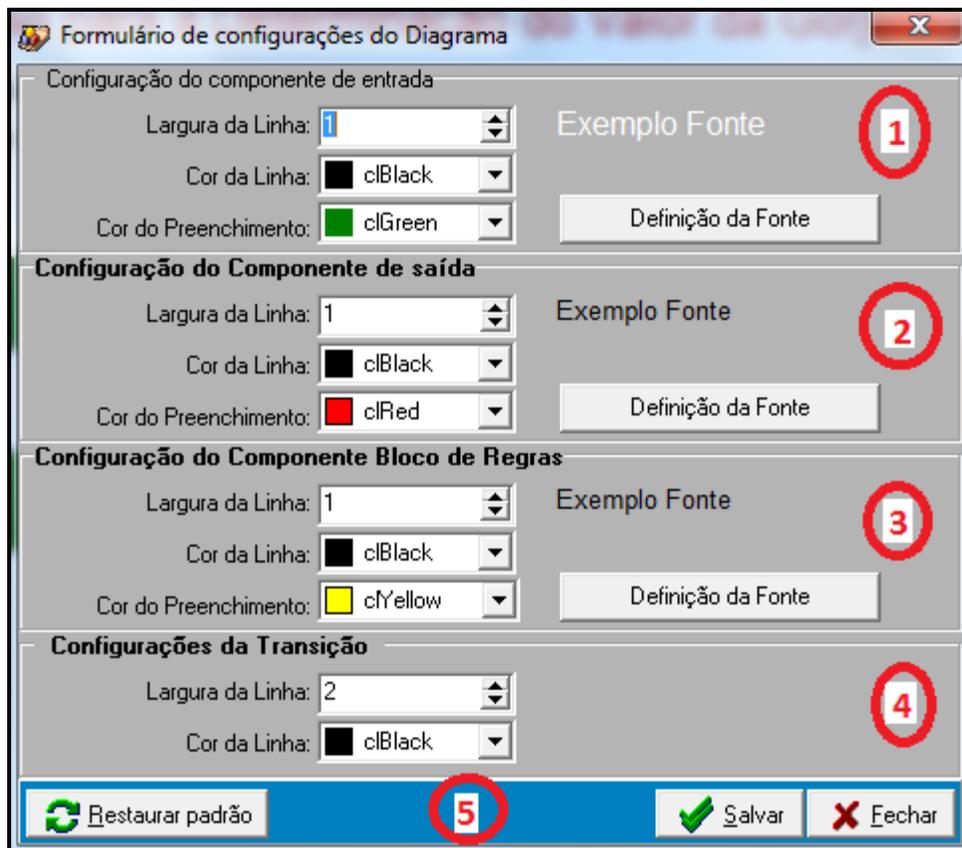


Figura 43- Tela de personalização do projeto difuso.

Fonte: Do autor, extraída da ferramenta InFuzzy.

Outro recurso importante da ferramenta InFuzzy está relacionada à manipulação de textos no projeto. Os textos criados podem ser utilizados para fins de autoria, comentários, entre outras possíveis utilizações. Na seção a seguir são detalhadas as operações com textos.

4.4.5 Tela de adição de texto no projeto

A Figura 44 apresenta a tela de adição e edição de textos do projeto. Esses textos se tornam importantes à medida que informações sobre o projeto precisam estar presentes nele. Os textos adicionados e a sua configuração são livres e depende apenas do usuário. Os textos gerados não influenciam no projeto, apenas na apresentação visual. Para adicionar um texto, basta selecionar a ferramenta de texto na barra de ferramentas e clicar na área desejada para inserir o texto. As informações solicitadas para a criação de um texto são: mensagem a ser

exibida, cor de fundo da mensagem e definição da fonte. Na parte inferior da tela, o sistema já fornece uma prévia da configuração selecionada. Para finalizar, basta clicar no botão salvar.

Para editar um texto existente, existem três formas de realizar essa tarefa, sendo elas:

1. Duplo clique sobre um texto na área de trabalho.
2. Clique com o botão direito sobre o texto e na sequência seleção da opção editar.
3. Clique com o botão direito sobre a estrutura em árvore do projeto e na sequência utilizar-opção editar.

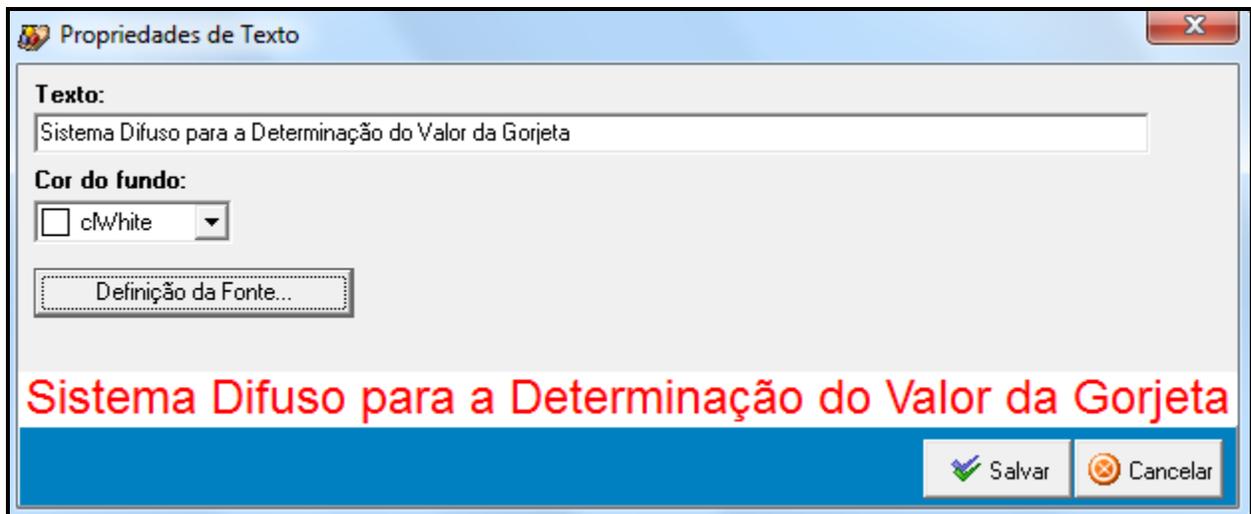


Figura 44- Tela de adição e configuração de textos do projeto.

Fonte: Do autor, extraída da ferramenta InFuzzy.

Através do processo de simulação é possível validar o sistema difuso implementado. A próxima seção aborda as funcionalidades da tela de simulação.

4.4.6 Tela de simulação

A Figura 45 apresenta a tela de simulação do sistema. A partir dessa tela, o usuário tem a capacidade de avaliar o sistema em relação às variáveis de entrada. Na simulação o sistema apresenta de forma analítica o processo de fuzificação, inferência e defuzificação, permitindo ao projetista analisar desvios de comportamento, decorrente de regras mal formuladas. Essa tela será explicada em seis passos, conforme assinalado na Figura 45.

1. Entrada dos valores: Nessa etapa o usuário (projetista) informa os respectivos valores para cada uma das variáveis. Para facilitar esse processo e atender as normas de ergonomia, o sistema apresenta ao lado do campo de entrada dos valores, as informações referentes à amplitude e unidade da variável.
2. Barra de controle sobre as simulações: Através do botão “Adicionar” é possível incrementar o conjunto de testes a serem realizados. O botão “Editar” permite que uma determinada simulação possa ser editada. As informações referentes às simulações são persistidas no projeto, ficando salvas para futuras simulações.
3. Através de um conjunto de paletas são exibidas informações da simulação, sendo elas: Conjunto de valores que compõe a base de testes, descrição analítica do último processo de simulação realizado, descrição textual do sistema implementado, processo de defuzificação apresentado em um gráfico, histórico de saída apresentado em gráfico e, por último, *debug* de comunicação. Essas etapas serão abordadas com mais detalhes na sequência do texto.
4. Conjunto de valores que compõe a simulação. A primeira coluna (Tempo (ms)) é utilizada para informar ao usuário o tempo em milissegundos necessário para realizar o processo. Esse tempo pode variar em decorrência dos seguintes motivos: *hardware* utilizado, definição da variável, passos, processos do sistema operacional, entre outros. É importante lembrar que esse tempo sempre será maior do que em uma aplicação real, pois nesse processo são geradas muitas etapas de depuração, assim

consumindo muito tempo. As colunas em verde representam as entradas e as colunas em vermelho representam as saídas.

5. Barra de funções: A primeira função “Executar simulação” deve ser utilizada para executar uma simulação previamente selecionada na lista de simulações. O botão “Executar todas as simulações” faz com que todas as simulações cadastradas sejam executadas em ordem cronológica de cadastro. A opção “Excluir simulação” permite a eliminação de uma determinada simulação previamente selecionada. Essa ação exige uma confirmação do usuário através de uma caixa de diálogo. Para excluir todas as simulações, o usuário deve selecionar o botão “Excluir todas as simulações”, será necessário uma confirmação através de uma caixa de diálogo também.
6. Comunicação UDP: Através de uma porta o sistema InFuzzy permite a comunicação com outros sistemas através do protocolo UDP. O protocolo utilizado para a troca de mensagens será abordado no decorrer do texto.



Figura 45- Tela de simulação do sistema difuso.

Fonte: Do autor, extraída da ferramenta InFuzzy.

As próximas seções abordam de forma detalhada as funcionalidades desenvolvidas em cada aba da tela de simulação, representada pelo item 3 da Figura 45.

4.4.6.1 Descrição das etapas de simulação

Conforme mencionado anteriormente, a descrição das etapas do processo completo do controlador difuso é apresentada na Tabela 8. A ferramenta InFuzzy permite ao usuário salvar, copiar e manipular as informações de todas as etapas realizadas, já que estão em formato de texto. As etapas de depuração do sistema difuso apresentadas na Tabela 8 serão descritas a seguir, a fim de um entendimento sobre o processo.

A primeira etapa apresentada é a fuzificação. Nessa etapa são atribuídos os valores de pertinência a cada termo linguístico em relação ao valor de entrada. Para exemplificar, observa-se que a pertinência para o termo linguístico “Ruim” é 0,14; para “Bom” é 0,41 e para “Excelente” é 0. Esse procedimento é realizado para todas as variáveis.

O próximo passo é a inferência, onde são computadas as regras, implicação e agregação. Para que o usuário tenha facilidade na depuração das rotinas, são apresentadas as regras na íntegra e na sequência com a substituição dos valores de pertinência dos termos linguísticos. Na sequência é realizada a agregação.

A última etapa realizada é a defuzificação, onde é apresentado o método de defuzificação aplicado e o valor das variáveis de saída.

Tabela 8- Descrição do processo de analítico das etapas do controlador difuso.

===== FUZIFICAÇÃO =====	
Valor Entrada "Serviço" = 3,00	
Pertinência "Ruim" = 0,14	
Pertinência "Bom" = 0,41	
Pertinência "Excelente" = 0,00	
Valor Entrada "Comida" = 2,00	
Pertinência "Ruim" = 0,50	

Pertinência "Ótima" = 0,00

===== INFERÊNCIA =====
 BLOCO DE REGRA = Regras Principais

REGRA = 0

Se (Serviço = Ruim) AND (Comida = Ruim) Então (Gorjeta = Pobre)

Se (Serviço = 0,14) AND (Comida = 0,50) Então (Gorjeta = 0,64)

AGREGAÇÃO

Valor Saída "Gorjeta

Pertinência "Pobre" = 0,64

Pertinência "Média" = 0,00

Pertinência "Generosa" = 0,00

REGRA = 1

Se (Serviço = Bom) Então (Gorjeta = Média)

Se (Serviço = 0,41) Então (Gorjeta = 0,41)

AGREGAÇÃO

Valor Saída "Gorjeta

Pertinência "Pobre" = 0,64

Pertinência "Média" = 0,41

Pertinência "Generosa" = 0,00

REGRA = 2

Se (Serviço = Excelente) AND (Comida = Ótima) Então (Gorjeta = Generosa)

Se (Serviço = 0,00) AND (Comida = 0,00) Então (Gorjeta = 0,00)

AGREGAÇÃO

Valor Saída "Gorjeta

Pertinência "Pobre" = 0,64

Pertinência "Média" = 0,41

Pertinência "Generosa" = 0,00

===== RESULTADO FINAL AGREGAÇÃO =====

Valor Saída "Gorjeta

Pertinência "Pobre" = 0,64

Pertinência "Média" = 0,41

Pertinência "Generosa" = 0,00

===== DEFUZIFICAÇÃO =====

Defuzificação pelo método do Primeiro dos Máximos

Variável de saída

Gorjeta = 3,20

Fonte: Do autor, extraída da ferramenta InFuzzy.

4.4.6.2 Descrição do sistema difuso

Outra facilidade apresentada pela ferramenta InFuzzy é a descrição textual do sistema difuso projetado pelo usuário. Na Tabela 9 é apresentada a descrição textual do projeto “Gorjeta”, onde é possível observar as variáveis de entrada, as funções utilizadas para cada termo linguístico e os valores que compõem as funções. O mesmo ocorre com as variáveis de saída. Por último são apresentadas as regras criadas pelo projetista do sistema difuso.

Tabela 9- Descrição textual do sistema difuso implementado.

<p>===== VARIÁVEIS DE ENTRADA =====</p> <p>Serviço (0 a 10 Qualidade Serviço) Ruim - Gaussiana(0 , 1,5) Bom - Gaussiana(5 , 1,5) Excelente - Gaussiana(10 , 1,5)</p> <p>Comida (0 a 10 Qualidade da comida) Ruim - Rampa Esquerda(0 , 1 , 3) Ótima - Rampa Direita(7 , 9 , 10)</p> <p>===== VARIÁVEIS DE SAÍDA =====</p> <p>Gorjeta (0 a 30 Percentual da Gorjeta) Pobre - Triangular(0 , 5 , 10) Média - Triangular(10 , 15 , 20) Generosa - Triangular(20 , 25 , 30)</p> <p>===== REGRAS =====</p> <p>Regras Principais Se (Serviço = Ruim) AND (Comida = Ruim) Então (Gorjeta = Pobre) Se (Serviço = Bom) Então (Gorjeta = Média) Se (Serviço = Excelente) AND (Comida = Ótima) Então (Gorjeta = Generosa)</p>
--

Fonte: Do autor, extraída da ferramenta InFuzzy.

4.4.6.3 Gráfico de defuzificação

A Figura 46 apresenta o gráfico da última etapa do processo, a defuzificação. Através de um gráfico de área é possível visualizar o resultado da agregação. A linha vertical é destinada a sinalizar o resultado da defuzificação. O resultado do processo de defuzificação “Primeiro dos Máximos” foi de 3,2. Nesta tela existem as seguintes possibilidades:

- Escolha da variável de saída: No caso de existir mais de uma variável de saída, o usuário possui a opção de selecionar a desejada e projetar o gráfico resultante do processo de agregação.
- Cor da linha: Possibilita ao usuário selecionar a cor da linha que sinaliza o resultado do processo de defuzificação.
- Cor da área de defuzificação: Permite ao usuário selecionar a cor da área do gráfico.

As configurações das cores são destinadas a permitir ao usuário uma personalização do gráfico do resultado. Isso se deve ao fato do usuário ter a possibilidade de exportar um *bitmap* desse gráfico ou realizar a sua impressão.

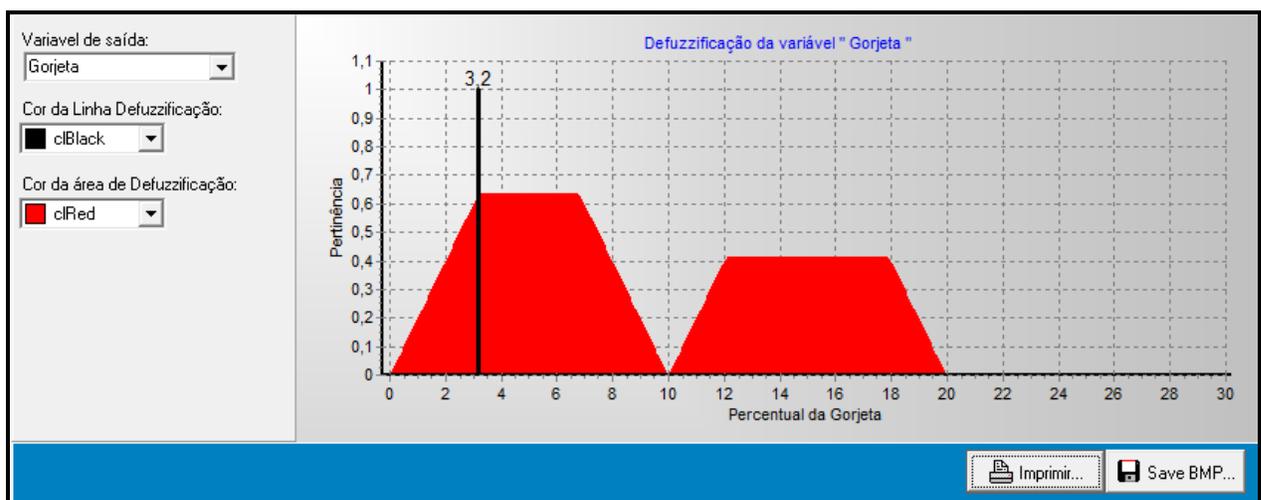


Figura 46- Tela de simulação etapa defuzificação.

Fonte: Do autor, extraída da ferramenta InFuzzy.

4.4.6.4 Histórico de saídas

O histórico de saídas é destinado à visualização dos resultados das variáveis de saída em um espaço de tempo através de gráficos de linha. Esse gráfico se torna importante à medida que o sistema esteja relacionado a um projeto de controle. Para exemplificar, imagina-se a seguinte situação: manter a temperatura de um forno a 50 °C iniciando com o forno desligado. Através do gráfico será possível visualizar o aumento da temperatura até alcançar os 50 °C e, após isso, manter a temperatura.

Nesta aba o sistema permite o controle do nível de 3D para o gráfico, como também permite a exportação do gráfico no formato BMP.

A Figura 47 apresenta o gráfico gerado a partir da execução de várias simulações. As interações apresentadas no gráfico foram obtidas através do projeto “Gorjeta”. Observando o gráfico é possível perceber que o eixo das abscissas representa o número de simulações executadas.



Figura 47- Gráfico do histórico de saídas.

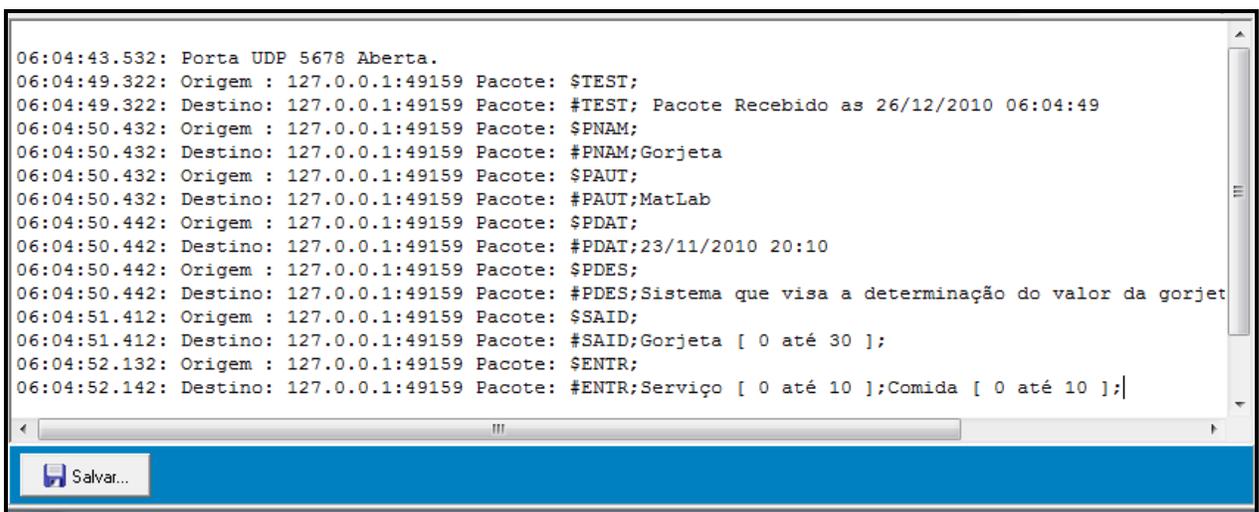
Fonte: Do autor, extraída da ferramenta InFuzzy.

4.4.6.5 Debug Comunicação

A aba “*Debug Comunicação*” é destinada a armazenar toda a troca de mensagens decorrente do processo de comunicação realizada entre a ferramenta Infuzzy e *softwares* de terceiros, através do protocolo UDP.

A Figura 48 exibe a aba de “*Debug Comunicação*”. Observando essa tela é possível perceber que ela mantém as seguintes informações:

- Hora: Exibe o horário no formato “hora:minuto:segundos:milissegundos” em que uma determinada mensagem foi recebida ou enviada.
- Origem: Endereço do IP e porta da mensagem recebida.
- Destino: Endereço e porta do destinatário da mensagem.
- Pacote: Informação recebida foi enviada.



```

06:04:43.532: Porta UDP 5678 Aberta.
06:04:49.322: Origem : 127.0.0.1:49159 Pacote: $TEST;
06:04:49.322: Destino: 127.0.0.1:49159 Pacote: #TEST; Pacote Recebido as 26/12/2010 06:04:49
06:04:50.432: Origem : 127.0.0.1:49159 Pacote: $PNAM;
06:04:50.432: Destino: 127.0.0.1:49159 Pacote: #PNAM;Gorjeta
06:04:50.432: Origem : 127.0.0.1:49159 Pacote: $PAUT;
06:04:50.432: Destino: 127.0.0.1:49159 Pacote: #PAUT;MatLab
06:04:50.442: Origem : 127.0.0.1:49159 Pacote: $PDAT;
06:04:50.442: Destino: 127.0.0.1:49159 Pacote: #PDAT;23/11/2010 20:10
06:04:50.442: Origem : 127.0.0.1:49159 Pacote: $PDES;
06:04:50.442: Destino: 127.0.0.1:49159 Pacote: #PDES;Sistema que visa a determinação do valor da gorjeta
06:04:51.412: Origem : 127.0.0.1:49159 Pacote: $SAID;
06:04:51.412: Destino: 127.0.0.1:49159 Pacote: #SAID;Gorjeta [ 0 até 30 ];
06:04:52.132: Origem : 127.0.0.1:49159 Pacote: $ENTR;
06:04:52.142: Destino: 127.0.0.1:49159 Pacote: #ENTR;Serviço [ 0 até 10 ];Comida [ 0 até 10 ];|
  
```

Figura 48- Log de comunicação UDP.

Fonte: Do autor, extraída da ferramenta InFuzzy.

4.5 Estrutura do arquivo gerado pelo sistema InFuzzy

A ferramenta InFuzzy adota o padrão de arquivos INI para salvar o projeto. O esclarecimento e detalhamento do padrão utilizado permitem a utilização do arquivo gerado em outros projetos futuros. Essa escolha se deve ao fato da facilidade de manipulação da informação dentro deste padrão. O padrão de arquivos INI também é adotado pelo Matlab[®]. Arquivos INI possuem três pontos, sendo eles:

1. Seção: Declarado entre colchetes e destinado a organizar um grupo de variáveis.
2. Variável: Utilizado como identificador de um parâmetro.
3. Valor: Conteúdo vinculado a uma variável.

A vantagem deste tipo de arquivo é a nitidez, clareza das informações e a facilidade de acesso aos resultados. As linhas a seguir exibem a função utilizada no Delphi para realizar a leitura de uma variável. A linha 1 exibe a função e os parâmetros e a linha 2 apresenta um exemplo. É necessário passar 3 parâmetros para a função, sendo o primeiro a seção desejada, o segundo a variável desejada e o terceiro o conteúdo a ser retornado, caso não exista a seção ou a variável. Na linha 2 é possível visualizar que o sistema está realizando a leitura do título do projeto; caso o mesmo não exista, a função retornará uma *string* vazia.

```

Linha 1 = Arq.ReadString( Seção , Variável , Conteúdo );
Linha 2 = Arq.ReadString( 'PROJETO' , 'TITULO' , '' );

```

A Tabela 10 apresenta a seção de informações do projeto. As principais informações presentes são: “TITULO” (utilizado para nomear o projeto), “AUTOR” (identificação do autor do projeto), “DATA” (Dia da criação do projeto ou última edição das propriedades do projeto), “DESCRICA0” (uma breve informação destinada a descrever o projeto) e por último “IDENTIFICADOR_OBJETOS” (trata-se de um valor de incremento único utilizado para identificar objetos como variáveis de entrada, saída e bloco de regras, com esse parâmetro o

sistema evita duplicação de identificadores de objetos). Observa-se que os dados são referentes ao projeto “Gorjeta”.

Tabela 10- Arquivo do sistema: Seção projeto

[PROJETO]
TITULO =Gorjeta AUTOR = Matlab® DATA =23/11/2010 20:10 DESCRICA O=Sistema que visa à determinação do valor da gorjeta do cliente em percentual IDENTIFICADOR_OBJETOS =4

Fonte: Do autor, extraída da ferramenta InFuzzy.

Na Tabela 11 é apresentada a estrutura das informações salvas para a definição das principais funções do sistema. Observando a tabela identificam-se os seguintes parâmetros: “PROJETO_FUNCAO_AND” (destinado a declarar a função a ser utilizada no cálculo dos antecedentes unidos por “AND”), “PROJETO_FUNCAO_OR” (destinado a declarar a função a ser utilizada no cálculo dos antecedentes unidos por “OR”), “PROJETO_DEFUZZIFICACAO” (determina qual o algoritmo empregado no processo de defuzzificação das variáveis de saída), “PROJETO_FUNCAO_AGREGACAO” (determina a função a ser aplicada no processo de agregação dos resultados obtidos após a implicação), “PROJETO_FUNCAO_IMPLICACAO” (determina a função a ser utilizada no processo de implicação, ou seja, como o resultado do cálculo dos antecedentes influencia as variáveis de saída) e por último “PROJETO_PASSOS” (determina a precisão do cálculo da defuzzificação, quanto menor esse valor, maior a precisão do resultado, porém maior a necessidade computacional).

Tabela 11- Arquivo do sistema: Seção definição das funções

[DEFINICAO_FUNCOES]
PROJETO_FUNCAO_AND =FUNC_PRODUTO_LIMITADO PROJETO_FUNCAO_OR =FUNC_MAXIMO PROJETO_DEFUZZIFICACAO =DEFUZZIFICACAO_PRIMEIRO_MAXIMOS PROJETO_FUNCAO_AGREGACAO =FUNC_MAXIMO

PROJETO_FUNCAO_IMPLICACAO=FUNC_MINIMO PROJETO_PASSOS=0,10
--

Fonte: Do autor, extraída da ferramenta InFuzzy.

A seção “JANELA” apresentada na Tabela 12 é destinada a controlar preferências do usuário quanto à disposição de atributos presentes na janela. Existem duas possibilidades de configuração para as variáveis “INFORMACOES_SISTEMA” e “DEFINICOES_SISTEMA”, sendo elas “+” (indica que a janela está minimizada e pode se ampliada) e “-” (indica que a janela está maximizada e pode ser minimizada). O objetivo dessa configuração é a possibilidade do usuário minimizar a janela de informações referentes ao projeto e as definições do sistema ampliando o tamanho da área utilizada para a montagem da árvore de estrutura do projeto difuso.

Tabela 12- Arquivo do sistema: Seção janela

[JANELA]
INFORMACOES_SISTEMA=-
DEFINICOES_SISTEMA=+

Fonte: Do autor, extraída da ferramenta InFuzzy.

A Tabela 13 apresenta a estrutura de uma variável de entrada. No final do título da seção é acrescentado um valor inteiro, iniciando em zero, tendo a seguinte estrutura “ENTRADA_X”, onde “X” é um valor inteiro sequencial. A variável “IDOBJ” é um valor inteiro único no sistema gerado a partir da variável “IDENTIFICADOR_OBJETOS” na seção “PROJETO”. A variável “NAME” é destinada ao nome da variável de entrada. As variáveis “ORIGEMX”, “ORIGEMY”, “DESTINOX” e “DESTINOY” são destinadas ao desenho gráfico do retângulo que representa a variável de entrada. As variáveis “TEXTOX” e “TEXTTOY” são utilizadas para determinar as coordenadas de escrita do nome da variável de entrada. As variáveis “UNIVERSOINICIO” e “UNIVERSOFIM” determinam a amplitude da variável podendo essas serem especificadas em valores flutuantes com precisão de duas casas decimais. A última variável de configuração de uma variável de entrada é a “UNIDADE”, campo textual destinado a exibir a unidade da variável no eixo das abscissas.

Tabela 13- Arquivo do sistema: Definição de variáveis de entrada

[ENTRADA_0]
IDOBJ=1 NAME=Serviço ORIGEMX=14 ORIGEMY=134 DESTINOX=184 DESTINOY=184 TEXTOX=72 TEXTOY=150 UNIVERSOINICIO=0 UNIVERSOFIM=10 UNIDADE=Qualidade Serviço

Fonte: Do autor, extraída da ferramenta InFuzzy.

A Tabela 14 exibe a estrutura dos termos linguísticos. O nome da seção define um relacionamento entre o termo e a variável. Observando o nome da seção “ENTRADA_X_TERMO_Y”, onde “X” representa o sequencial da variável e “Y” representa o sequencial do termo linguístico. As seguintes variáveis são utilizadas para cada termo linguístico, “TERMO” (nome informado pelo usuário ao termo linguístico), “FUNCAO” (Função escolhida para o termo linguístico. É utilizado um valor inteiro para representar as funções seguindo a seguinte lógica: 1- triangular, 2- trapézio, 3- gaussiana, 4- cauchy, 5- sigmóide, 6- rampa esquerda, 7- rampa direita, 8- inclinação esquerda, 9- inclinação direita, 10- discreto). A variável “COR” é utilizada para definir a cor da função do gráfico de pertinências. As variáveis “VAL_A”, “VAL_B”, “VAL_C” e “VAL_D” são utilizadas para armazenar o valor dos parâmetros das funções.

Tabela 14- Arquivo do sistema: Termo linguístico de entrada

[ENTRADA_0_TERMO_0]
TERMO=Ruim FUNCAO=2

COR=cIRed
VAL_A=0
VAL_B=1,5
VAL_C=0
VAL_D=0

Fonte: Do autor, extraída da ferramenta InFuzzy.

A Tabela 15 apresenta a estrutura de um objeto de saída, todas as considerações realizadas sobre a estrutura das variáveis de entrada se aplicam à estrutura das variáveis de saída com exceção do nome da seção, que é composto da palavra “SAIDA_” mais um valor inteiro sequencial.

Tabela 15- Arquivo do sistema: Definição de variáveis de saída

[SAIDA_0]
IDOBJ=3
NAME=Gorjeta
ORIGEMX=490
ORIGEMY=179
DESTINOX=660
DESTINOY=229
TEXTOX=554
TEXTOY=196
UNIVERSOINICIO=0
UNIVERSOFIM=30
UNIDADE=Percentual da Gorjeta

Fonte: Do autor, extraída da ferramenta InFuzzy.

Os termos linguísticos das variáveis de saída obedecem a mesma descrição realizada para os termos linguísticos das variáveis de entrada, com exceção do nome da seção que segue a seguinte estratégia “SAIDA_X_TEMO_Y”, onde “X” representa o sequencial das variáveis de saída e “Y” representa o sequencial inteiro dos termos referentes à variável “X”. Na Tabela 16 pode ser visualizada essa estrutura.

Tabela 16- Arquivo do sistema: Termo linguístico de saída

[SAIDA_0_TERMO_0]
TERMO =Pobre FUNCAO =0 COR =clRed VAL_A =0 VAL_B =5 VAL_C =10 VAL_D =0

Fonte: Do autor, extraída da ferramenta InFuzzy.

A Tabela 17 apresenta a seção de definição dos blocos de regras. Por padrão, o nome da seção é composto por “BLOCO_REGRA_” mais o sequencial inteiro, iniciando em zero. A variável “IDOBJ” é obtida através do incremento único da variável “IDENTIFICADOR_OBJETOS” na seção “PROJETO”. A variável “NAME” é utilizada apenas para permitir ao usuário um nome mais amigável ao bloco de regras em nível visual. As variáveis “ORIGEMX”, “ORIGEMY”, “DESTINOX” e “DESTINOY” são utilizadas para desenhar o retângulo que representa o bloco de regras na tela. Por último, as variáveis “TEXTOX” e “TEXTOY” são utilizadas para escrever o valor da variável “NAME” dentro do retângulo do bloco de regras.

Tabela 17- Arquivo do sistema: Bloco de regra

[BLOCO_REGRA_0]
IDOBJ =0 NAME =Regras Principais ORIGEMX =255 ORIGEMY =179 DESTINOX =425 DESTINOY =229 TEXTOX =289 TEXTOY =196

Fonte: Do autor, extraída da ferramenta InFuzzy.

A Tabela 18 apresenta a estrutura do arquivo utilizada para armazenar a relação entre um determinado bloco de regras e as variáveis de entrada e saída. O título da seção é composto da seguinte forma: “BLOCO_REGRA_X_CONEXAO_ENTRADA_Y” para conexões com variáveis de entradas e “BLOCO_REGRA_X_CONEXAO_SAIDA_Y” para conexões com variáveis de saída, onde “X” representa o sequencial do bloco de regra e “Y” o sequencial das conexões tanto de entrada como de saída. A informação mais importante está relacionada à variável “IDOBJORIGEM” que é destinada a informar a relação com o objeto de entrada ou de saída. As variáveis “ORIGEMX”, “ORIGEMY”, “DESTINOX” e “DESTINOY” são destinadas a desenhar uma reta na área gráfica com o intuito de ligar uma variável de entrada ou saída a um bloco de regra.

Tabela 18- Arquivo do sistema: Conexão de variáveis e blocos de regras

[BLOCO_REGRA_0_CONEXAO_ENTRADA_1]
IDOBJORIGEM=2 ORIGEMX=183 ORIGEMY=260 DESTINOX=255 DESTINOY=204
[BLOCO_REGRA_0_CONEXAO_SAIDA_0]
IDOBJORIGEM=3 ORIGEMX=425 ORIGEMY=204 DESTINOX=490 DESTINOY=204

Fonte: Do autor, extraída da ferramenta InFuzzy.

A Tabela 19 apresenta a estrutura utilizada para armazenar as regras de um bloco de regras. No título da seção é determinada a sequência de regras através de um valor inteiro. A variável “NRO” é apenas um sequencial inteiro referente ao número da regra. A variável “ANTECEDENTE” é destinada a armazenar os antecedentes das regras. A variável “CONSEQUENTE” é destinada a armazenar os consequentes de uma regra. Tanto para os

consequentes quanto para os antecedentes a estrutura do valor é semelhante, sempre iniciará com “(” mais “**VARIÁVEL**” mais “=” mais ”**TERMO LINGUÍSTICO**” mais “)”. Se existir mais de um antecedente, os mesmos deverão estar unidos pelo operador “AND” ou “OR”. Para facilitar a leitura e remontagem do projeto, foi optado por manter uma variável (“TIPO_UNIAO”) para salvar o tipo de conexão entre os antecedentes.

Tabela 19- Arquivo do sistema: Regras

[BLOCO_REGRA_0_REGRA_0]
NRO=1 ANTECEDENTE =(Serviço = Ruim) AND (Comida = Ruim) CONSEQUENTE =(Gorjeta = Pobre) TIPO_UNIAO =AND
[BLOCO_REGRA_0_REGRA_1]
NRO=2 ANTECEDENTE =(Serviço = Bom) CONSEQUENTE =(Gorjeta = Média) TIPO_UNIAO =AND

Fonte: Do autor, extraída da ferramenta InFuzzy.

Como visto anteriormente, o sistema permite ao usuário criar textos, esses textos possuem uma estrutura e precisam ser persistidos no arquivo do projeto. Na Tabela 20 é apresentada a estrutura de mensagens de texto. O título da seção possui um valor inteiro de incremento para cada mensagem iniciando em zero. A variável “TEXTO” armazena o texto a ser projetado na tela. As variáveis “ORIGEMX”, “ORIGEMY”, “DESTINOX” e “DESTINOY” são utilizadas para definir o posicionamento do texto na tela. As coordenadas de destino são necessárias neste caso, pois permitem a projeção de um retângulo atrás do texto na cor definida na variável “COLORFUNDO”. As variáveis “FONTENAME”, “FONTCOLOR” e “FONTSIZE” são destinadas à configuração do texto na área do projeto no sistema.

Tabela 20- Arquivo do sistema: Mensagens de texto

[TEXT0_0]
TEXT0 =Sistema Difuso para a Determinação do Valor da Gorjeta ORIGEMX =4 ORIGEMY =5 DESTINOX =611 DESTINOY =32 FONTNAME =Arial FontColor =clRed Fontsize =18 ColorFundo =clWhite

Fonte: Do autor, extraída da ferramenta InFuzzy.

A Tabela 21 apresenta a estrutura dos dados das simulações. O título da seção possui valor de incremento para cada entrada de valores de entrada. A variável “TEMPO” armazena o tempo em milissegundos de processamento para concluir uma interação do sistema difuso. A variável “IDOBJ_X” é utilizada para apontar para a variável de entrada e a variável “VALOR_X” o valor utilizado na simulação. O sistema armazena as informações de simulações para que os mesmos testes possam ser realizados em um futuro e por outras pessoas.

Tabela 21- Arquivo do sistema: Simulação

[SIMULACAO_0]
TEMPO =20 IDOBJ_0 =1 VALOR_0 =5 IDOBJ_1 =2 VALOR_1 =5
[SIMULACAO_1]
TEMPO =9 IDOBJ_0 =1 VALOR_0 =2

IDOBJ_1=2
VALOR_1=2

Fonte: Do autor, extraída da ferramenta InFuzzy.

A última parte do arquivo trás as informações referentes às opções de configuração do projeto na parte gráfica. Essas configurações se aplicam aos seguintes itens: variáveis de entrada, variáveis de saída, blocos de regras e linhas de conexão. As configurações são apresentadas na Tabela 22.

Tabela 22- Arquivo do sistema: Opções do projeto

[OPCOES]
CONF_LINHA_LARGURA_ENTRADA=1
CONF_LINHA_LARGURA_SAIDA=1
CONF_LINHA_LARGURA_INFERENCIA=1
CONF_LINHA_LARGURA_CONEXAO=2
CONF_LINHA_COR_ENTRADA=clBlack
CONF_LINHA_COR_SAIDA=clBlack
CONF_LINHA_COR_INFERENCIA=clBlack
CONF_LINHA_COR_CONEXAO=clBlack
CONF_PREENCHIMENTO_COR_ENTRADA=clGreen
CONF_PREENCHIMENTO_COR_SAIDA=clRed
CONF_PREENCHIMENTO_COR_INFERENCIA=clYellow
CONF_FONTE_TAMANHO_ENTRADA=12
CONF_FONTE_TAMANHO_SAIDA=10
CONF_FONTE_TAMANHO_INFERENCIA=10
CONF_FONTE_NAME_ENTRADA=Arial
CONF_FONTE_NAME_SAIDA=Arial
CONF_FONTE_NAME_INFERENCIA=Arial
CONF_FONTE_COR_ENTRADA=clWhite
CONF_FONTE_COR_SAIDA=clBlack
CONF_FONTE_COR_INFERENCIA=clBlack

Fonte: Do autor, extraída da ferramenta InFuzzy.

4.6 Protocolo de comunicação UDP

A comunicação implementada através do protocolo UDP permite a ferramenta InFuzzy a interação com outros *softwares* de forma simples e rápida. A utilização dessa comunicação permite ao usuário implementar um *software* com base na lógica difusa sem a escrita de nenhuma linha de código para a lógica difusa.

A Figura 49 apresenta a estrutura de comunicação. A ferramenta InFuzzy mantém toda a estrutura do sistema difuso e o *software* do usuário implementa o sistema desejado com exceção das rotinas difusas. Quando for necessário, o *software* do usuário deve enviar um pacote para o InFuzzy com o valor das entradas. A ferramenta InFuzzy irá processar o sistema difuso e retornar o valor das variáveis de saída. A comunicação sempre é iniciada pelo *software* do cliente, seguindo a ordem apresentada na Figura 49. Essa estrutura tem a capacidade de operar sobre redes locais e *internet*. As Letras A, B, C e D representam o fluxo de comunicação entre os sistemas. O processo de comunicação sempre inicia pelo *software* do usuário (A), a ferramenta InFuzzy recebe e processa o pacote UDP recebido (B), após realizado o processo difuso completo, é disparado um pacote com os resultados (C), e por último o *software* do usuário recebe o retorno do sistema InFuzzy (D).



Figura 49- Estrutura de comunicação UDP.

Fonte: Do autor.

A vantagem da implementação através do protocolo de comunicação está relacionado à velocidade de validação da lógica difusa para um determinado projeto. Para esclarecer melhor essa informação imagina-se o seguinte cenário: em uma aplicação agrária, um determinado periférico dotado de processador e módulo rede encontra-se em um silo. Sabe-se que a

implementação da lógica difusa em processadores de baixa capacidade pode ser algo trabalhoso, consequentemente demorado. Nesse tipo de cenário, o usuário utilizaria a comunicação UDP para realizar o processamento da lógica difusa em um servidor.

A Tabela 23 apresenta o protocolo de comunicação. Os comandos são enviados em ASCII. Foi utilizado o protocolo NMEA (*National Marine Electronics Association*) como referência para a elaboração do protocolo adotado pelo InFuzzy. O protocolo respeita as seguintes características básicas:

- Os comandos enviados iniciam com “\$” + quatro letras em maiúsculo finalizando com “;”.
- Os comandos retornados pelo servidor seguem o mesmo princípio dos comandos enviados pelo cliente, porém utilizam “#” ao invés de “\$”. Dessa forma diferenciando mensagens do servidor e do cliente.

O único comando obrigatório é “\$EXEC”, os demais comandos têm apenas funcionalidades informativas, porém são muito importantes na medida em que não se conhece o modelo difuso implementado, sendo que a partir dessas informações é possível entender o projeto e realizar o uso das funções.

Outro comando especial é o “\$SAVE;”, esse comando executa praticamente todas as rotinas do comando “\$EXEC”, porém é retornado ao cliente apenas se o processo ocorreu com sucesso ou erro. No servidor ficam registrados apenas os valores de entrada enviados, não sendo executada a rotina de simulação para esses valores.

Tabela 23- Protocolo de comunicação

Comando	Descrição	Resposta do Servidor
\$TEST;	Comando para testar a comunicação com o servidor.	#TEST; <Horário do servidor>
\$PNAM;	Comando para receber o	#PNAM;<Nome do Projeto>

	nome do projeto.	
\$PAUT;	Comando para receber o nome do autor.	#PAUT;<Nome do Autor>
\$PDAT;	Comando para receber a data de criação do projeto.	#PDAT;<Data do Projeto>
\$PDES;	Comando para receber a descrição do sistema	#PDES;<Descrição Projeto>
\$ENTR;	Recupera do servidor a lista de variáveis de entrada e o tamanho do universo.	#ENTR;<variável + escala> Ex: #ENTR;Serviço [0 até 10]; Comida [0 até 10];
\$SAID;	Recupera do servidor a lista de variáveis de saída e o tamanho do universo.	#SAID;<variável + escala> Ex: #SAID;Gorjeta [0 até 30];
\$EXEC;<variável de entrada>=<valor>; Ex:\$EXEC;Serviço=5,9; Comida=4,8;	Comando para a execução do sistema difuso. Esse comando tem como retorno um valor para cada variável de saída.	#EXEC;<Status>;<variável de saída>=<valor de saída>; Ex: #EXEC;SUCES;Gorjeta=15,45;
\$SAVE;<variável de entrada>=<valor>; Ex:\$SAVE;Serviço=5,9; Comida=4,8;	Comando para salvar as entradas no sistema difuso. Esse comando tem como retorno apenas o <i>status</i> .	#EXEC;<Status>; Ex: #EXEC;SUCES;

Fonte: do autor

A Tabela 24 demonstra os possíveis erros gerados pelo comando “\$SAVE” e “\$EXEC”. Os erros são os mesmos para os dois comandos, pois esses erros são capturados no processo de validação dos parâmetros recebidos. Existem apenas seis erros possíveis, e os mesmos estão descritos e exemplificados na tabela.

Para facilitar o desenvolvimento de sistema de terceiros, todos os erros possuem um campo com a descrição do erro no final da instrução, dessa forma, caso algum erro aconteça, o programador poderá ler a mensagem de erro no momento, sem a necessidade de localizar a documentação sobre os erros.

Tabela 24- Retornos de erros de comunicação.

Erro	Descrição	Exemplo
ERRO1	Indica que o valor de uma determinada variável está abaixo do limite do universo.	#SAVE;ERRO1;<Variável + escala>; <Descrição do erro> Ex: #SAVE;ERRO1;Serviço [0 até 10]; Variável inferior ao início universo.
ERRO2	Indica que o valor de uma determinada variável está acima do limite do universo.	#SAVE;ERRO2;<Variável + escala>; <Descrição do erro> #SAVE;ERRO2;Serviço [0 até 10]; Variável superior ao fim universo.
ERRO3	Variável enviada ao servidor não encontrada.	#SAVE;ERRO3;<Variável>; <Descrição> Ex: #SAVE;ERRO3;Serviço; Variável não encontrada.
ERRO4	O número de variáveis de entrada enviadas ao servidor é menor do que o número de variáveis esperadas pelo servidor.	#SAVE;ERRO4; Número de parâmetros inferior ao número de variáveis.
ERRO5	O número de variáveis de entrada enviadas ao servidor é maior do que o número de variáveis esperadas pelo servidor.	#SAVE;ERRO5; Número de parâmetros superior ao número de variáveis.

ERRO6	Existe um erro na conversão do valor enviado ao servidor.	#SAVE;ERRO6;<variável>; <Descrição> Ex: #SAVE;ERRO6;Serviço; Problemas na conversão de valores, utilize "," para decimais.
-------	---	---

Fonte: Do autor.

Para facilitar a implementação do protocolo foi desenvolvida uma aplicação que realiza a programação de todas as funcionalidades mencionadas no protocolo. A descrição dessa aplicação pode ser visualizada no ANEXO D.

4.7 Homologação da ferramenta InFuzzy

Para realizar a homologação da ferramenta foi utilizada a seguinte estratégia. Primeiro, foi escolhida uma ferramenta de sistemas difusos para a comparação de resultados. A ferramenta escolhida foi o Matlab[®], ToolBox Fuzzy. A escolha desse *software* se deve ao reconhecimento desse fabricante no mundo e pela qualidade/confiabilidade apresentada em seus produtos. No segundo momento foi escolhida uma aplicação demo presente na instalação do Matlab[®] versão R2009a, chamada *Tipper*. Essa aplicação tem como objetivo definir o percentual de gorjeta em relação à qualidade da comida e do serviço.

Ao longo dessa comparação de resultados, foram realizadas alterações no projeto das duas ferramentas (Matlab[®] e InFuzzy) a fim de cobrir o máximo de recursos das ferramentas. A primeira etapa a ser analisada é a definição das variáveis de entrada, saída, blocos de regras e conexões.

A Figura 50 apresenta a modelagem do sistema difuso gorjeta implementado no Matlab[®] (A) e no InFuzzy (B). As duas ferramentas permitem a definição das variáveis de entrada e saída de forma gráfica. As funções projetadas nas caixas das variáveis no Matlab[®] são meramente ilustrativas, não representando os termos linguísticos das variáveis. No bloco de inferência o *software* Matlab[®] adiciona a informação do tipo de modelo Mamdani, porém essa informação

somente estaria correta se a função utilizada para AND e a implicação fosse mínimo e a agregação utilizasse a função máximo. A informação mais correta seria “Modelos Clássicos” para Larsen, Mamdani e Zadeh ou “Modelos de Interpolação” para Takagi-Sugeno e Tsukamoto. Nessa primeira etapa, a ferramenta InFuzzy permite a adição de textos que permitem a explanação do projeto, o que não é possível no sistema Matlab[®]. Outra diferença na modelagem está relacionada aos blocos de inferência, no *software* Matlab[®] é possível utilizar apenas uma, enquanto que na ferramenta InFuzzy é possível criar quantas forem necessárias.

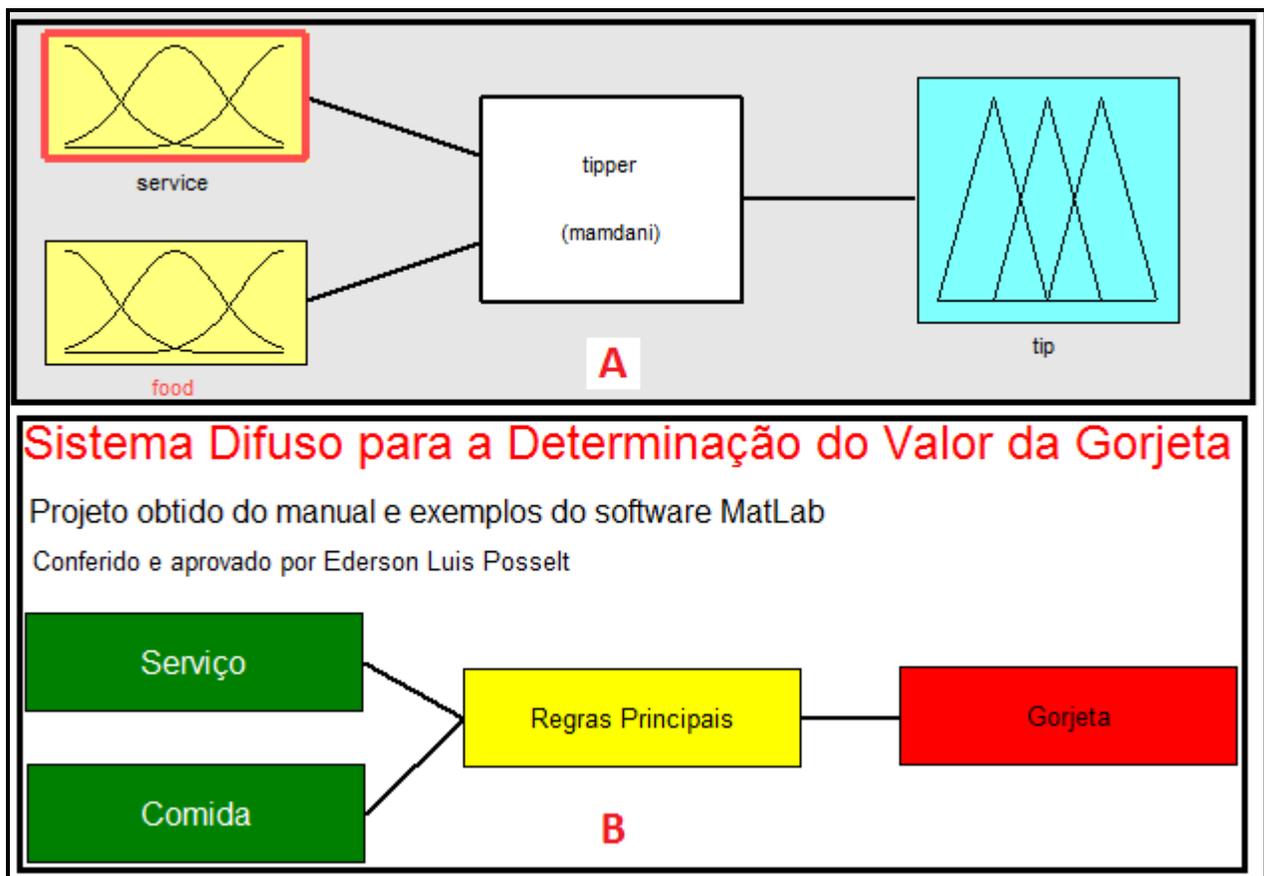


Figura 50- Modelagem do sistema difuso no Matlab[®] e no InFuzzy

Fonte: Do autor, extraída do *software* Matlab[®] e InFuzzy.

Continuando o projeto gorjeta, a próxima etapa é a definição dos termos linguísticos para a variável “Serviço”. O universo dessa variável inicia em 0 até 10 e está dividida em três termos linguísticos, sendo eles: pobre, bom e excelente. Conforme demonstrado na Figura 51, tanto a ferramenta Matlab[®] (A), quanto InFuzzy (B), foram capazes de modelar as funções gaussianas.

Conforme normas de ergonomia e usabilidade, o *software* Matlab[®] não apresenta a descrição do eixo das abscissas e ordenadas. Uma das vantagens da ferramenta InFuzzy nessa etapa é a demonstração das funções em 3D, o que favorece um melhor entendimento ao usuário.

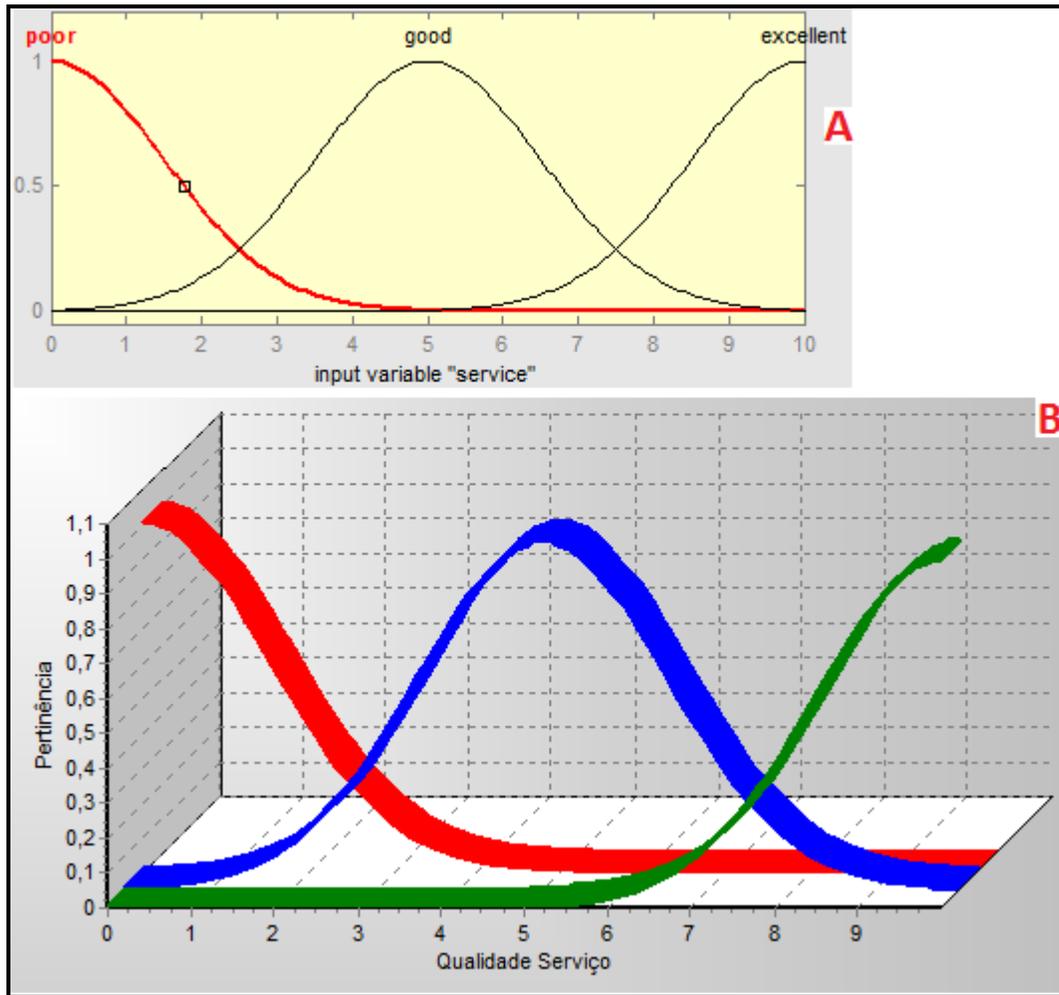


Figura 51- Variável de entrada "Serviço" e os termos linguísticos

Fonte: Do autor, extraída do *software* Matlab[®] e InFuzzy.

A Figura 52 apresenta a etapa da configuração dos termos linguísticos da variável “Comida”, essa variável tem o seu universo definido de 0 até 10. No *software* Matlab[®] (A), os termos linguísticos “ruim” e “delicioso” são projetados através da função de trapézio, porém parte da função trapezoidal não é apresentada, pois fica fora da área do universo. Na ferramenta InFuzzy (B) existe uma função especial para esses casos, chamada “Rampa”, essa função sempre

está vinculada a uma das extremidades do universo, sendo assim, deve-se optar pela função “Rampa Esquerda” e “Rampa Direita”.

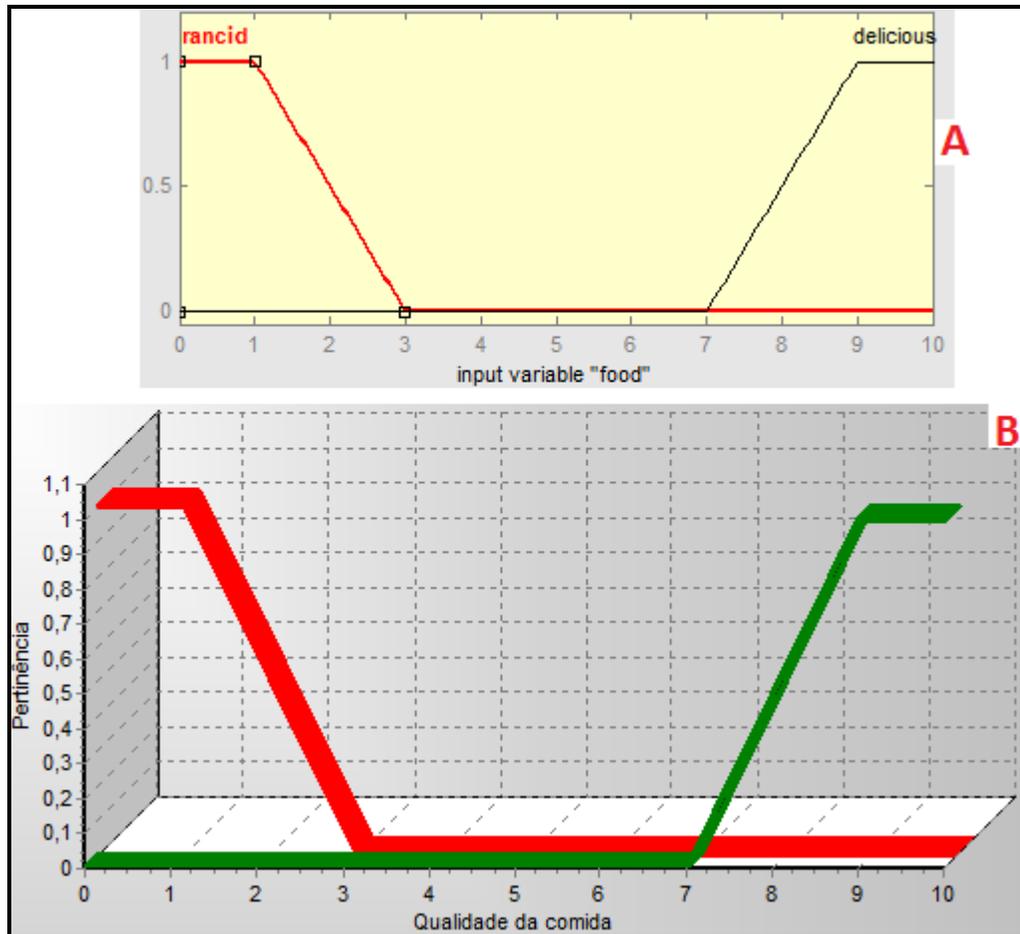


Figura 52- Variável de entrada "Comida" e os termos linguísticos

Fonte: Do autor, extraída do *software* Matlab® e InFuzzy.

A última variável a ser configurada no projeto Gorjeta, é a variável de saída “Gorjeta”, composta de três funções triangulares, “Pobre”, “Média” e “Generosa” e com o universo definido de zero a dez. No *software* Matlab® (A), por não possuir representação 3D fica um pouco difícil o entendimento da pertinência das funções. Uma observação interessante nos termos linguísticos está relacionada ao fato de não existirem áreas de interseção, como ocorre na variável de entrada “Serviço”. Na Figura 53 (A) é apresentada a configuração da variável de saída “Gorjeta” no *software* Matlab® e na Figura 53 (B) é apresentada a configuração da variável na ferramenta InFuzzy.

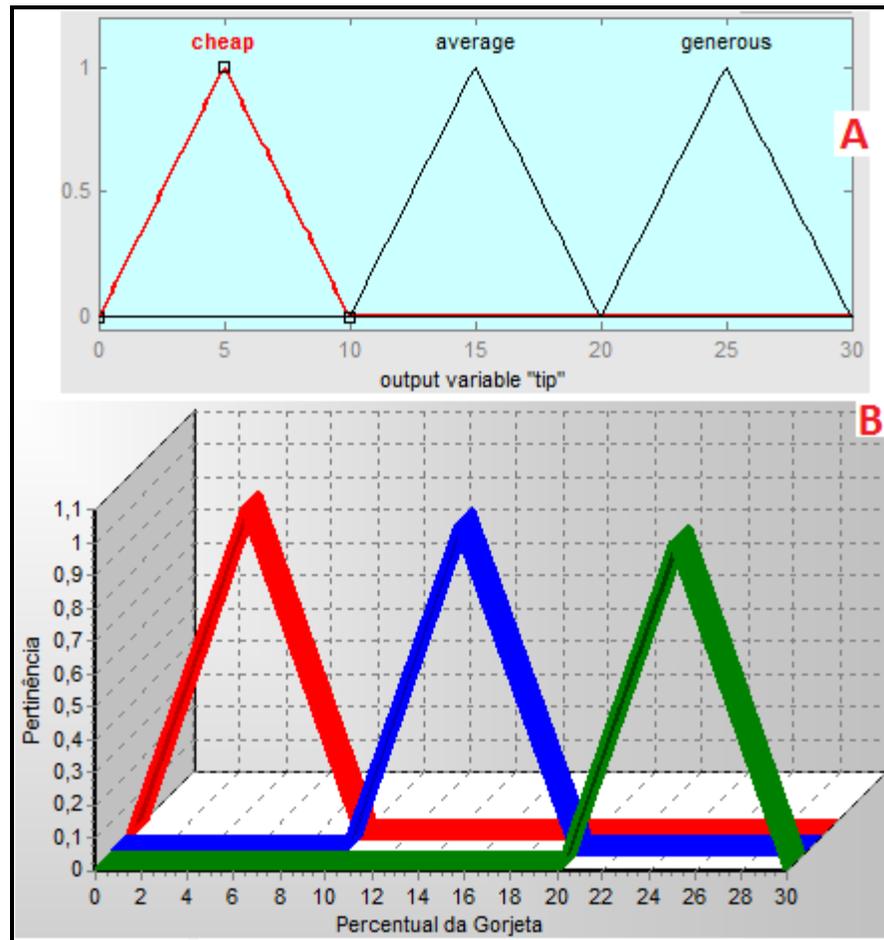


Figura 53- Variável de saída "Gorjeta" e termos linguísticos

Fonte: Do autor, extraída do *software* Matlab® e InFuzzy.

Após definidas as variáveis e os termos linguísticos, é o momento de montar as regras. Como visto na seção anterior, para acessar a interface de montagem das regras na ferramenta InFuzzy, basta clicar sobre o bloco de regras; no *software* Matlab® o procedimento é semelhante. Na Figura 54 é apresentada a montagem das regras no *software* Matlab® (A) e na ferramenta InFuzzy (B). O procedimento de criação das regras é semelhante nas duas ferramentas, porém existe uma pequena diferença na formatação. No *software* Matlab® as regras são dispostas em um componente conhecido como “*String List*”, enquanto que na ferramenta InFuzzy, as regras são dispostas em uma tabela com as colunas definidas de acordo com as etapas “Antecedentes” e “Consequentes”, facilitando a leitura e entendimento. O sistema Gorjeta é composto de apenas três regras, sendo essas apresentadas na Figura 54.

1. If (service is poor) or (food is rancid) then (tip is cheap) (1)
 2. If (service is good) then (tip is average) (1)
 3. If (service is excellent) or (food is delicious) then (tip is generous) (1)

Nro	Se	Antecedentes	Então	Consequentes
1	Se	(Serviço = Ruim) OR (Comida = Ruim)	Então	(Gorjeta = Pobre)
2	Se	(Serviço = Bom)	Então	(Gorjeta = Média)
3	Se	(Serviço = Excelente) OR (Comida = Ótima)	Então	(Gorjeta = Generosa)

Figura 54- Conjunto de regras do projeto Gorjeta

Fonte: Do autor, extraída do *software* Matlab® e InFuzzy.

Com as variáveis de entrada e saída com os seus termos linguísticos definidos, os conjuntos de regras formados, é o momento de realizar a última etapa de configuração. Essa etapa pode ser configurada e reconfigurada em qualquer etapa do projeto. A Figura 55 (A) apresenta a configuração pelo *software* Matlab® e a Figura 55 (B) pela ferramenta InFuzzy. As alterações realizadas nas configurações serão visualizadas nas etapas de simulação, pois é onde incidem. Uma das diferenças nesta etapa está relacionada a um detalhamento maior da descrição das funções no InFuzzy.

A	B
And method: min Or method: max Implication: min Aggregation: max Defuzzification: centroid	Definições do Projeto AND: 01 - Mínimo (Norma-T) OR: 01 - Máximo (Conorma-T) Implicação: 01 - Mínimo (Norma-T) Agregação: 01 - Máximo (Conorma-T) Defuzzificação: 01 - Centro da Gravidade Passos: 0,10

Figura 55- Configuração dos parâmetros do projeto Gorjeta

Fonte: Do autor, extraída do *software* Matlab® e InFuzzy.

Com todas as etapas configuradas, é o momento de testar a aplicação através do módulo de simulação. Na Figura 56 (A) é apresentada a simulação no *software* Matlab®. A técnica utilizada pelo Matlab® na apresentação da simulação é interessante ao passo que todas as regras são representadas através de gráficos, porém essa técnica fica um pouco pesada à medida que existam muitas variáveis e regras. Na ferramenta InFuzzy a etapa de simulação é dividida em 3 partes, a primeira é apenas o resultado do processo, a segunda uma descrição analítica do processo completo e a última um gráfico de área onde é sinalizado o resultado. A terceira etapa da simulação da ferramenta InFuzzy é apresentada na Figura 56 (B). Nesse momento foi atingido um objetivo importante no projeto, o resultado do *software* Matlab® é o mesmo que a ferramenta InFuzzy. Com isso, um dos objetivos do projeto foi atingido, a homologação do *software* desenvolvido neste projeto, InFuzzy homologado através de uma ferramenta reconhecida no mercado.

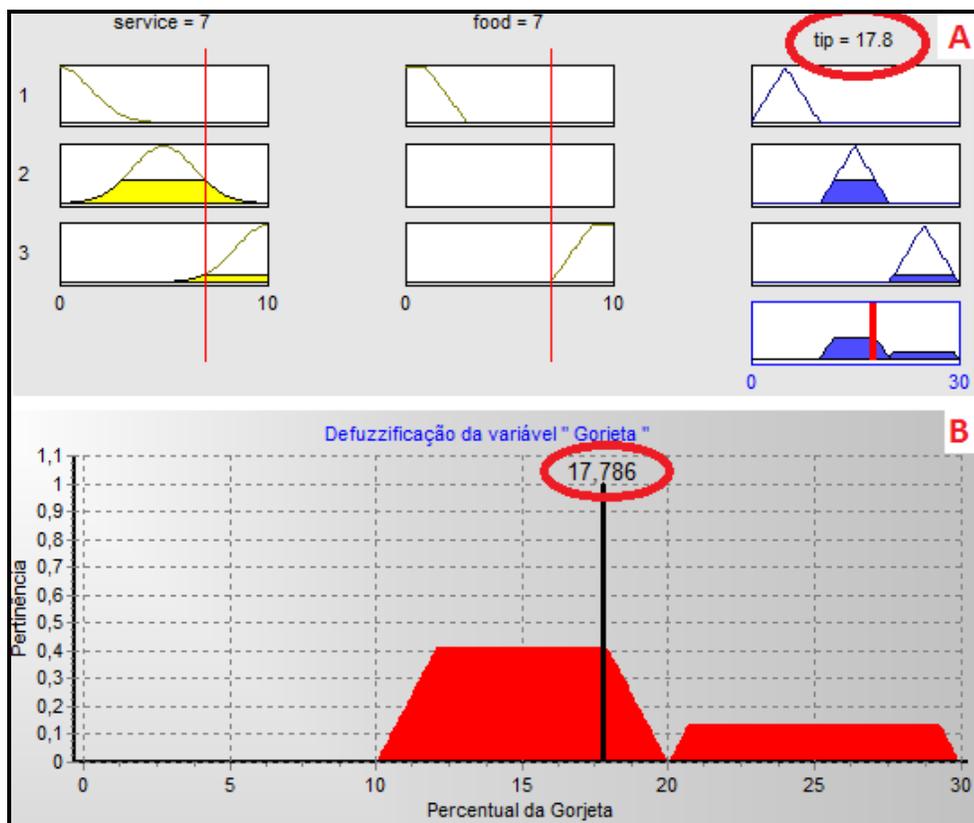


Figura 56- Etapa de simulação do projeto Gorjeta

Fonte: Do autor, extraída do *software* Matlab® e InFuzzy.

Para demonstrar a confiabilidade no projeto, foi criada uma lista de entradas que foi simulada. Todos os resultados foram conferidos no *software* Matlab® e coincidiram. A ferramenta InFuzzy apresenta uma vantagem em relação ao *software* Matlab®, onde é permitida a criação de uma lista de valores a serem testados e esses valores ficam salvos no projeto. Isso se torna mais importante na medida em que se realizam vários testes com configurações variadas. É importante lembrar que a coluna em azul trás o tempo de processamento em milissegundos para realizar o processamento completo. Na Figura 57 podem ser visualizados os resultados das simulações.

Tempo (ms)	Serviço	Comida	Gorjeta
16	5,00	5,00	15,00
16	2,00	2,00	7,52
16	7,00	7,00	17,79
15	9,00	9,00	24,47
16	10,00	0,00	15,00
16	0,00	10,00	15,00
15	10,00	10,00	24,92
16	0,00	0,00	5,08
16	3,00	2,00	9,66

Figura 57- Projeto gorjeta, lista de simulações

Fonte: Do autor, extraída da ferramenta InFuzzy.

Após a apresentação desta seção foi possível atingir um dos objetivos, implementar um projeto completo e comparar os resultados, a fim de demonstrar a confiabilidade da ferramenta. Na próxima seção serão apresentadas algumas inconformidades encontradas no *software* Matlab®.

4.7.1 Inconformidades encontradas no *software* Matlab®

Após a utilização do Matlab® como ferramenta de referência foi possível observar alguns pontos falhos no sistema. Pontos que geram desconfiança no usuário, pois esses erros nem sempre ocorrem.

Um dos primeiros pontos observados é apresentado na Figura 58. Na etapa “A” o resultado obtido foi 12,3; porém esse não é o resultado correto e esperado. Após realizar um ajuste no parâmetro “Plot points” para um valor superior a 101, o *software* Matlab® ajusta o resultado para 12,1. O resultado obtido na ferramenta InFuzzy foi de 12,1 para esse mesmo teste.

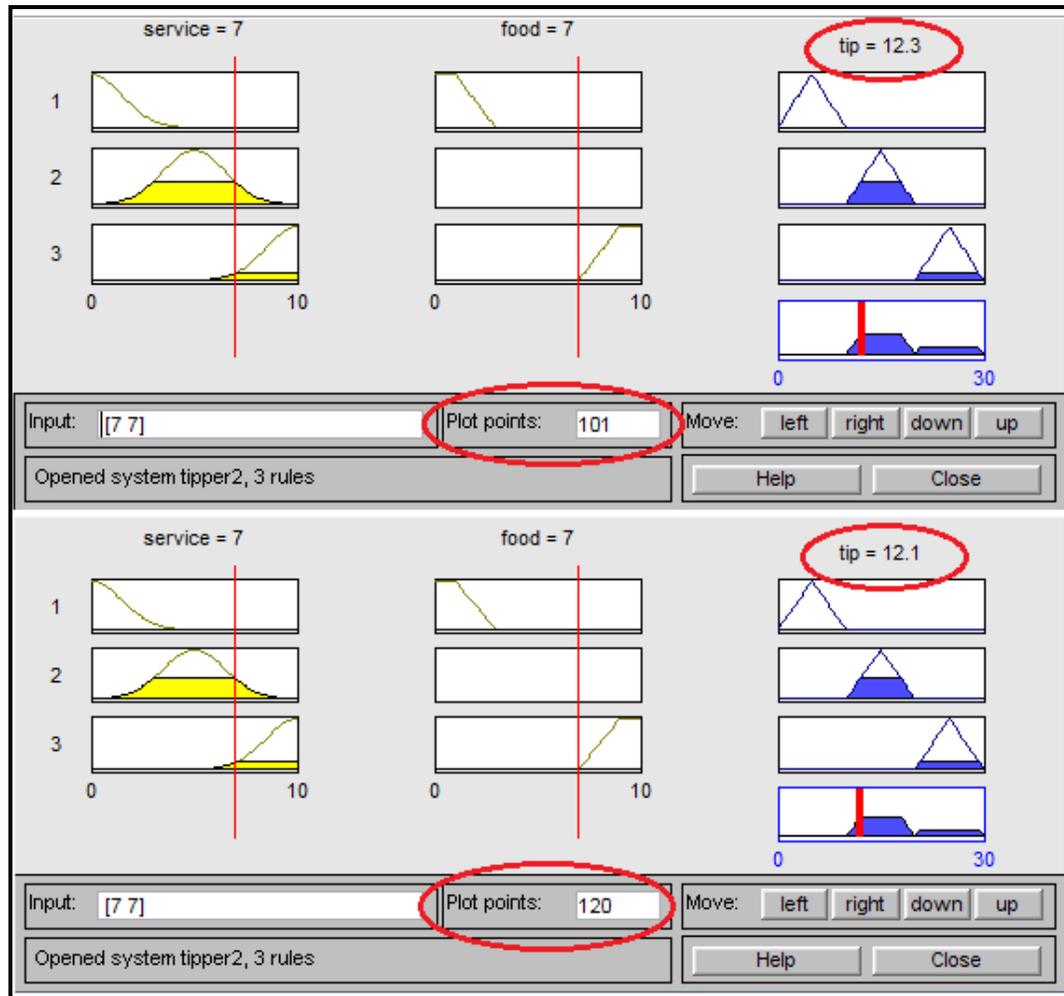


Figura 58- Erro apresentado pelo *software* Matlab® no resultado final.

Fonte: Figura obtida através do *software* Matlab®.

Um erro mais grave foi encontrado na etapa de defuzificação do *software* Matlab®, esse erro foi percebido após a seleção do método do último dos máximos, esse processo de defuzificação seleciona o elemento do eixo das abscissas mais à direita no universo com a maior pertinência. Na Figura 59 (A) é apresentado o resultado com erro, esse erro gerou uma desconfiança no projeto, uma vez que o resultado do *software* diferia do conceito dos processos

estudados até o momento. Após, um fato casual foi descoberto que, alterando o valor da variável “Plot points” para um valor diferente do padrão, o algoritmo é recalculado ajustando o resultado para o valor correto. O processo de defuzzificação correto pode ser visualizado na Figura 59 (B).

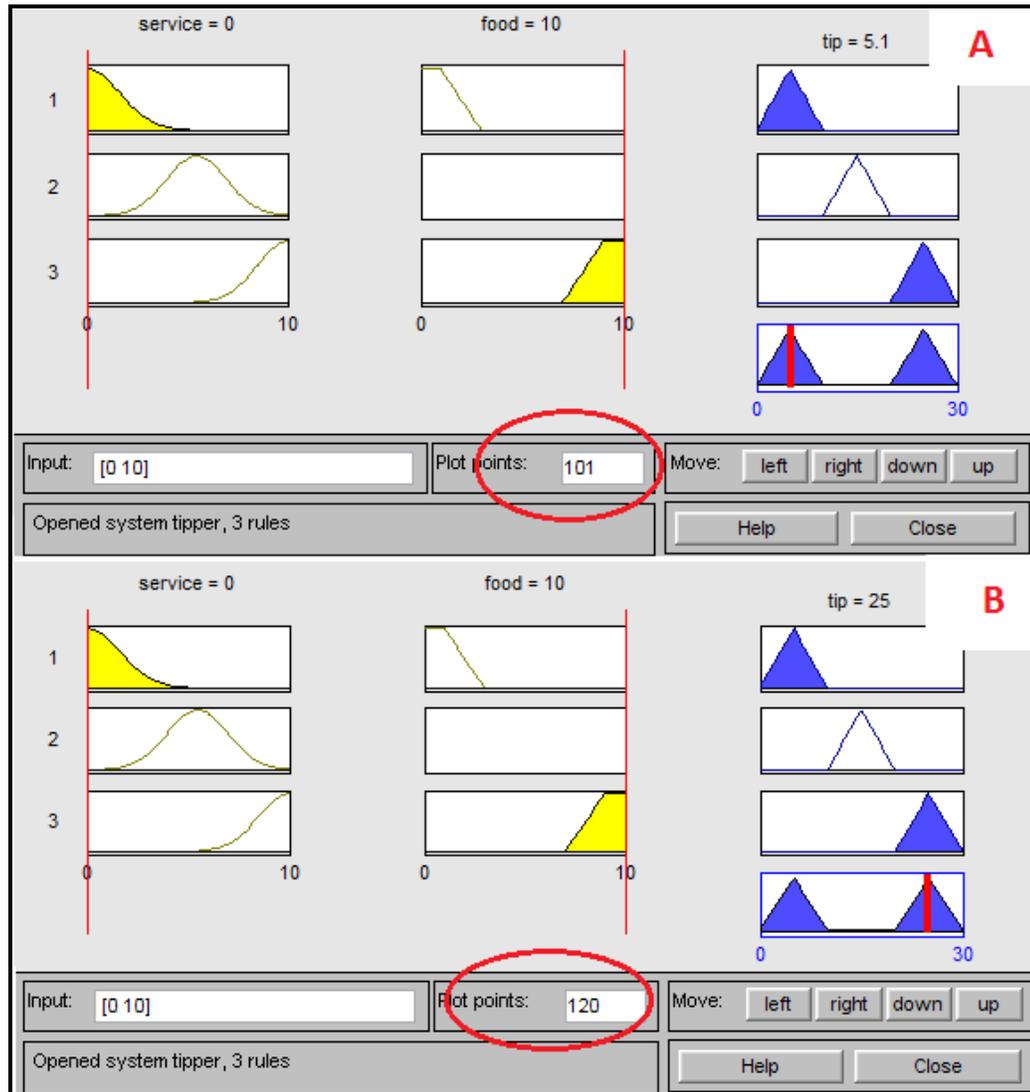


Figura 59- Erro apresentado pelo *software* Matlab[®] no processo de defuzzificação.

Fonte: Figura obtida através do *software* Matlab[®].

4.8 Processo de instalação da ferramenta InFuzzy

Para facilitar a instalação do sistema foi desenvolvido um instalador utilizando o *software* gratuito *Inno Setup versão 5.1.5*, o instalador se torna importante à medida que vários arquivos

são necessários para o sistema. O instalador realiza a instalação dos seguintes itens: executável do programa, manual, vídeo aulas, projeto de dissertação, exemplos implementados e os respectivos artigos dos quais foram extraídos e o desinstalador. A seguir serão explicadas as etapas do processo de instalação da ferramenta InFuzzy.

A Figura 60 apresenta a primeira tela do instalador. Essa tela apresenta algumas informações sobre o *software*. Para prosseguir a instalação basta clicar em “Avançar”. Para cancelar o processo de instalação basta clicar em “Cancelar”.



Figura 60- Instalador do sistema InFuzzy etapa 1

Fonte: Do autor, extraída da ferramenta InFuzzy.

A segunda etapa do instalador consiste na escolha de um diretório para a instalação do sistema. Por padrão o instalador está direcionado para a pasta de programas do sistema operacional Windows, porém o usuário pode alterar o destino da instalação. Alterar o destino da instalação pode ser importante caso o usuário não possuir permissão para instalar o sistema nos diretórios do sistema operacional. A Figura 61 apresenta essa etapa. Caso seja necessário alterar o

local de instalação, basta utilizar o botão “Procurar...” para selecionar o novo local. Para continuar a instalação, utilize o botão “Avançar” e para cancelar utilize o botão “Cancelar”. Nessa tela, para fins de informação, é apresentado o espaço requerido em disco para a instalação.

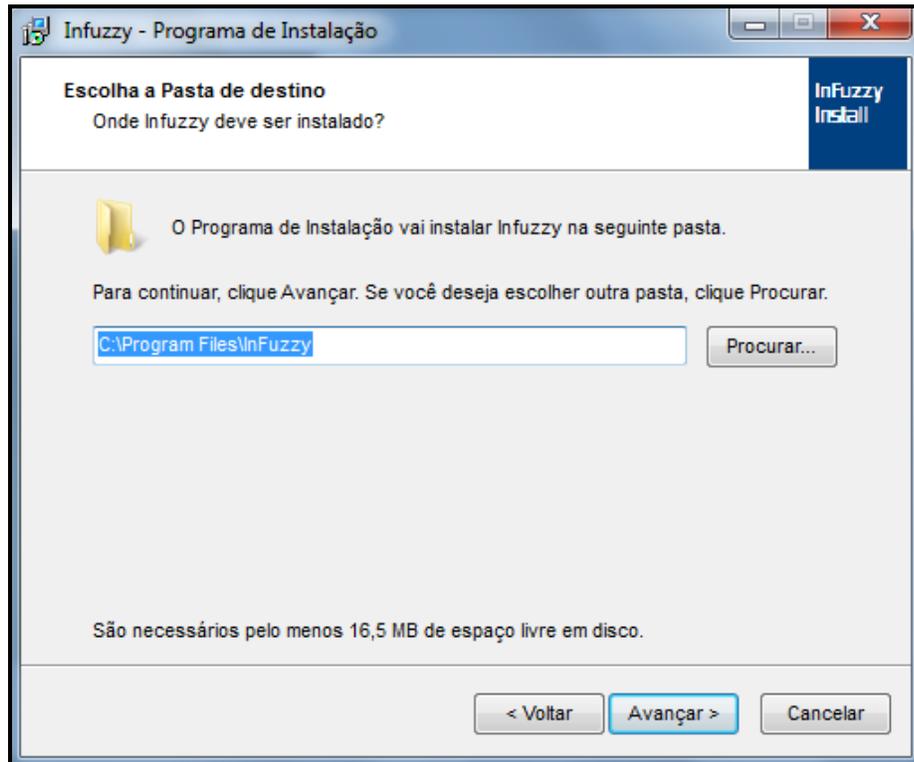


Figura 61- Instalador do sistema InFuzzy etapa 2.

Fonte: Do autor, extraída da ferramenta InFuzzy.

A terceira etapa do instalador é utilizada para definir o grupo dos ícones no menu iniciar do Windows. Por padrão o sistema sugere o nome do grupo como “InFuzzy”, mas esse pode ser alterado pelo usuário para o nome que achar mais conveniente. Nessa tela é possível acessar três funções, “Voltar” (voltar para a tela anterior), “Avançar” (Seguir para a próxima etapa do instalador) e por último “Cancelar” (utilizado para abandonar a instalação). Essa etapa pode ser visualizada na Figura 62.

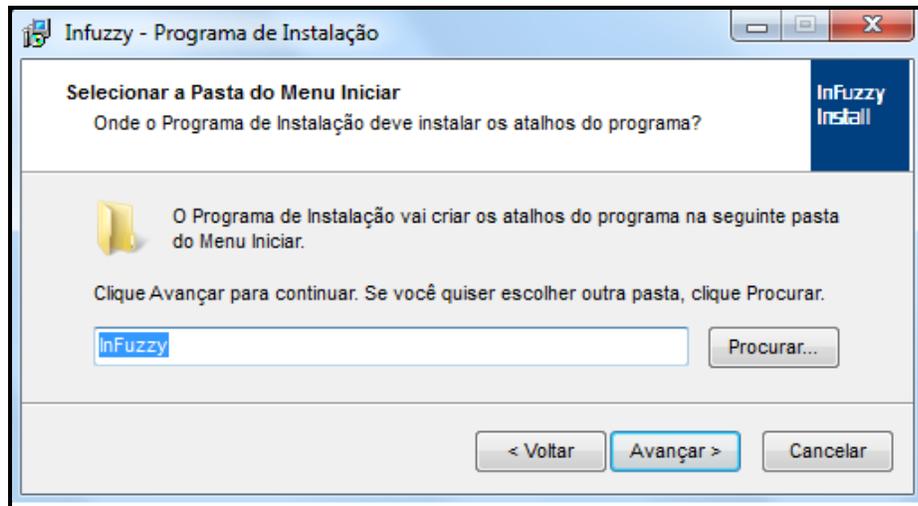


Figura 62- Instalador do sistema InFuzzy etapa 3

Fonte: Do autor, extraída da ferramenta InFuzzy.

A quarta etapa do instalador é destinada a uma tela de confirmação das configurações realizadas pelo usuário até o momento. Caso alguma das informações não esteja correta, o usuário pode selecionar a opção “Voltar” e ajustar as incoerências. Caso tudo esteja correto, basta utilizar o botão “Avançar” e por último, se necessário, basta utilizar a opção “Cancelar” para finalizar a instalação. A Figura 63 apresenta a tela do instalador na etapa 4.

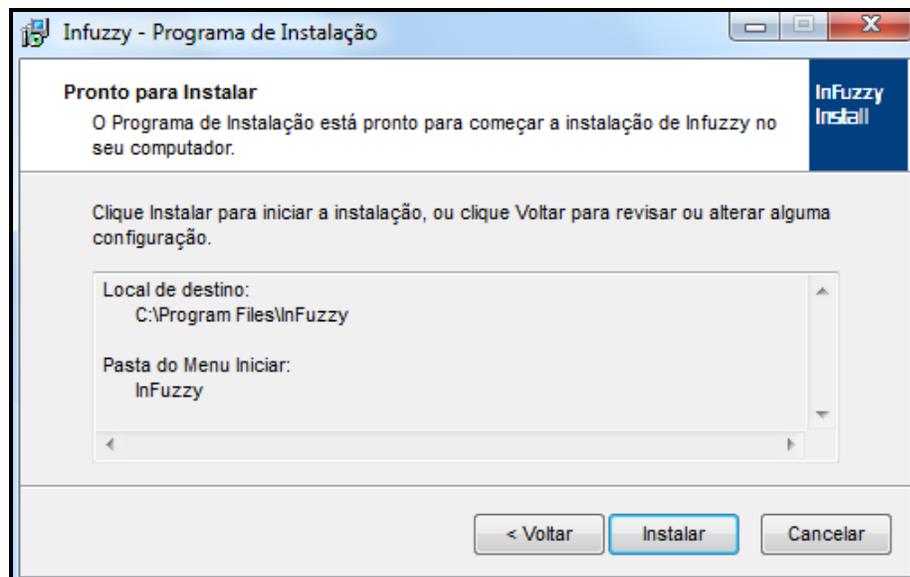


Figura 63- Instalador do sistema InFuzzy etapa 4.

Fonte: Do autor, extraída da ferramenta InFuzzy.

E finalmente, caso tudo tenha ocorrido perfeitamente, a tela apresentada na Figura 64 será exibida ao usuário. Nessa tela não existe nenhuma opção além de concluir a instalação.

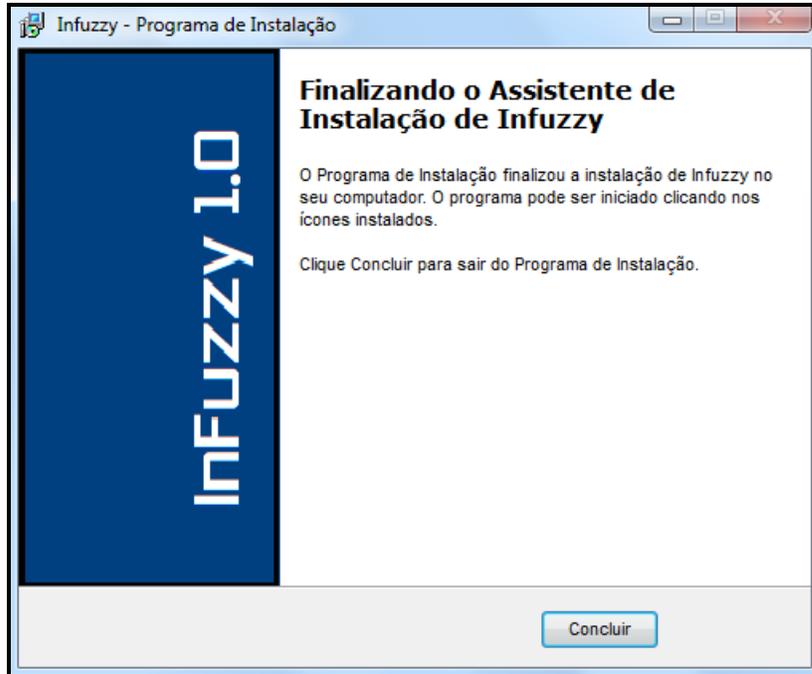


Figura 64- Instalador do sistema InFuzzy etapa 5.

Fonte: Do autor, extraída da ferramenta InFuzzy.

A Figura 65 apresenta o menu iniciar do Windows após a finalização da instalação. Através desse menu é possível acessar cinco itens, sendo eles:

1. Desinstalar InFuzzy: Ferramenta para remover a instalação do programa InFuzzy.
2. InFuzzy: *Software* de modelagem de sistemas difusos.
3. Manual: Manual no formato HTML do *software*.
4. Projeto.pdf: dissertação apresentada ao Programa de Pós-Graduação em Sistema e Processos Industriais.
5. Vídeo Aula: Vídeo aula com instruções de uso do sistema.

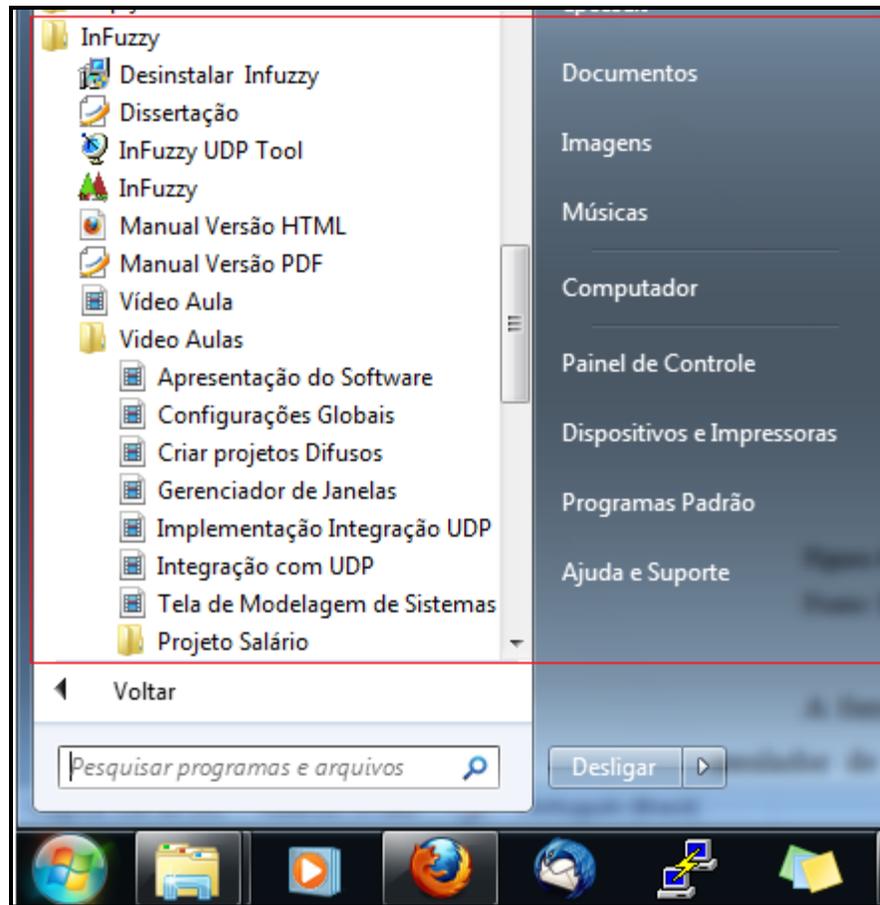


Figura 65- Instalador menu iniciar

Fonte: Do autor.

A ferramenta InFuzzy pode ser instalada no Linux com a utilização do *software Wine*, um emulador de *softwares win32* para a plataforma Linux. A homologação da ferramenta InFuzzy ocorreu na versão Ubuntu 10.04 LTS com kernel 2.6.32-24 release em abril de 2010. Nas versões derivadas do Debian, a instalação do Wine ocorre através do comando “*apt-get install wine*”, porém é necessário estar conectado com o usuário “*root*”.

Para realizar a instalação da ferramenta InFuzzy, é necessário estar conectado como “*root*”, na Figura 66 é apresentado o comando realizado no terminal e na parte inferior da imagem é exibida a primeira tela do processo de instalação. Todas as etapas de instalação no sistema Microsoft Windows são apresentadas na instalação no Linux. O comando “*wine Setup_InFuzzy.exe*” é responsável por iniciar o processo de instalação.

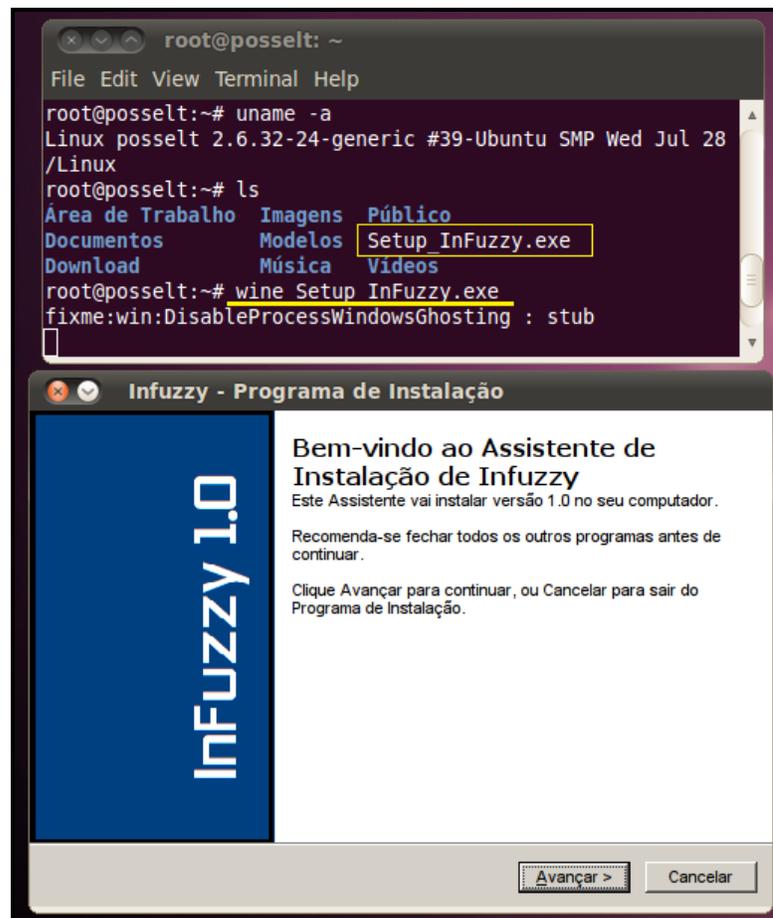


Figura 66- Instalação da ferramenta InFuzzy no Linux

Fonte: Do autor, extraída da ferramenta InFuzzy.

Para iniciar a ferramenta InFuzzy no Linux existem duas maneiras, a primeira é a execução do *software* através de linha de comando. Essa execução é apresentada na Figura 67, onde o primeiro comando é destinado para acesso ao diretório da instalação do sistema e o segundo comando é destinado à execução do *software* propriamente dito.

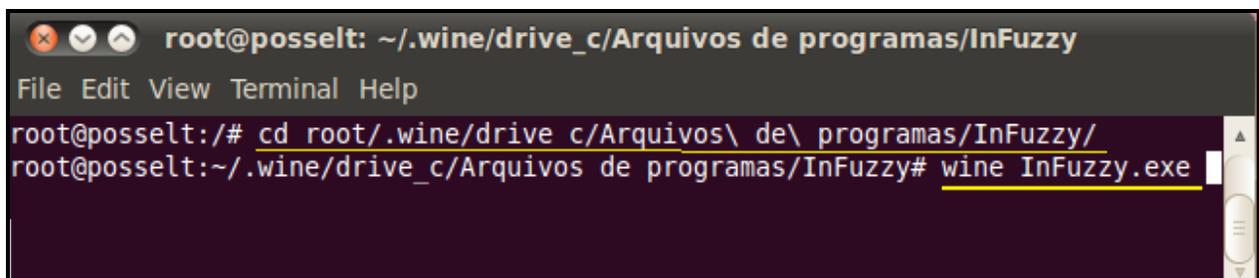


Figura 67- Iniciar a ferramenta InFuzzy através de linha de comando.

Fonte: Do autor.

A segunda maneira de iniciar o sistema ocorre através do menu da interface gráfica. No exemplo apresentado na Figura 68, foi utilizada a interface GNOME. É importante mencionar que o sistema operacional Linux pode executar diversas interfaces gráficas como: BlackBox, FluxBox, KDE, entre outras. Em algumas dessas interfaces é provável que não seja criado o menu de acesso gráfico.



Figura 68- Iniciar a ferramenta InFuzzy através do menu do Gnome.

Fonte: Do autor.

Para que seja possível acessar o restante dos arquivos que compõem o projeto (Manual, Projeto.pdf, Vídeo Aula) é necessário proceder de duas formas:

1. Instalar *softwares* da plataforma Microsoft para vincular o tipo de arquivo a um *software*.
2. Acessar a pasta do *software* e executar os complementos através de *softwares* nativos Linux.

Caso não exista um *software* vinculado ao arquivo desejado e se tente realizar a sua chamada através do menu, a seguinte mensagem será apresentada: “*Não existe um programa Windows configurado para abrir este tipo de arquivo*”.

A execução da ferramenta InFuzzy possui um performance inferior quando executado no Linux. Isso se deve ao fato de ser emulado através do *software* Wine. Na próxima seção serão apresentados alguns projetos implementados na ferramenta InFuzzy.

4.9 Projetos implementados na ferramenta InFuzzy

Essa seção é destinada a apresentar projetos difusos desenvolvidos utilizando a ferramenta InFuzzy. Os projetos abordados nos próximos tópicos são de autoria do autor ou extraídos de trabalhos de graduação, dissertações, teses e publicações na área. O objetivo dessa etapa é elucidar através de implementações a potencialidade da ferramenta InFuzzy, e para isso serão abordados os seguintes projetos:

- **Definição salário:** Projeto utilizado como demonstração nas vídeo aulas. O objetivo desse projeto é definir o salário de um trabalhador através de sua experiência e nível escolar.
- **Energia limpa:** Projeto extraído de uma publicação, esse projeto se baseia nas condições climáticas e nível de bateria para determinar o comportamento do sistema em relação à economia de energia.
- **Controle de ventilador:** Esse projeto tem o objetivo de verificar a temperatura e determinar a velocidade do ventilador do gabinete. Nesse projeto é utilizada a integração de um *hardware* com a ferramenta InFuzzy através do protocolo UDP/IP.

4.9.1 Projeto difuso: Definição salário

Esse projeto difuso foi desenvolvido pelo autor para demonstrar o uso do *software* de modelagem difusa InFuzzy. O desenvolvimento na íntegra desse projeto pode ser visualizado nas vídeo aulas distribuídas juntamente com o instalado do *software*.

O projeto difuso “Definição de Salário” é constituído de duas variáveis de entrada (Escolaridade, Experiência), um bloco de regras e uma variável de saída (Salário). Esse projeto tem como objetivo determinar o salário em relação à experiência e escolaridade. É importante lembrar que esse projeto é ilustrativo e suas regras são baseadas no conhecimento humano, neste caso, as regras foram elaboradas pelo autor.

A Tabela 25 apresenta as variáveis envolvidas no projeto, os termos linguísticos, suas funções e os respectivos parâmetros utilizados. Ao lado de cada variável é apresentado entre parênteses o universo e unidade da variável. Um fato interessante é a mescla de funções, esse sistema usufrui de sete funções distintas para modelar as variáveis. A ferramenta InFuzzy permite que sejam usadas todas as funções para uma mesma variável, não existindo restrições sobre esse aspecto.

Tabela 25- Projeto definição salário: Etapa definição variáveis

Termo linguístico	Função	Parâmetros
Variável de entrada: Experiência (0 a 30 anos)		
Pouca	Rampa Esquerda	(1, 2)
Média	Triangular	(1, 4, 7)
Alta	Trapézio	(5, 8, 12, 15)
Muito Alta	Rampa Direita	(13, 16)
Variável de entrada: Escolaridade (0 a 10 nível)		
Pouca	Inclinação Esquerda	(0, 3)
Média	Gaussiana	(5, 1)
Alta	Inclinação Direita	(7, 10)
Variável de saída: Salário (540 a 3000 Reais)		

Baixo	Rampa Esquerda	(600, 800)
Médio	Triangular	(700, 1000, 1300)
Alto	Trapézio	(1200, 1500, 1800, 2100)
Muito Alto	Rampa Direita	(2000, 2300)

Fonte: do autor

A Tabela 26 apresenta o conjunto de regras utilizadas no projeto. Essas regras foram estabelecidas pelo especialista, neste caso, o autor do projeto difuso. Foram definidas 12 (doze) regras que atendem a todas as combinações possíveis entre as variáveis de entrada. A tabela está dividida em duas colunas, antecedentes e consequentes.

Tabela 26- Projeto definição salário: Etapa definição das regras

Antecedente (Se)	Consequente (Então)
(Experiência = Pouca) AND (Escolaridade = Pouca)	(Salário = Baixo)
(Experiência = Pouca) AND (Escolaridade = Media)	(Salário = Baixo)
(Experiência = Pouca) AND (Escolaridade = Alta)	(Salário = Médio)
(Experiência = Media) AND (Escolaridade = Pouca)	(Salário = Baixo)
(Experiência = Media) AND (Escolaridade = Media)	(Salário = Médio)
(Experiência = Media) AND (Escolaridade = Alta)	(Salário = Alto)
(Experiência = Alta) AND (Escolaridade = Pouca)	(Salário = Médio)
(Experiência = Alta) AND (Escolaridade = Media)	(Salário = Médio)
(Experiência = Alta) AND (Escolaridade = Alta)	(Salário = Alto)
(Experiência = Muito Alta) AND (Escolaridade = Pouca)	(Salário = Médio)
(Experiência = Muito Alta) AND (Escolaridade = Media)	(Salário = Alto)
(Experiência = Muito Alta) AND (Escolaridade = Alta)	(Salário = Muito Alto)

Fonte: do autor

Na Tabela 27 são apresentados os operadores aplicados ao projeto “Definição Salário”. Trata-se de um sistema de inferência Mamdani genuíno. Qualquer trabalho desenvolvido com lógica difusa deveria possuir essas três etapas (definição de variáveis, definição das regras e a

definição dos operadores) de forma clara, pois a partir delas é possível refazer o projeto difuso em outras ferramentas a fim de realizar validações.

Tabela 27-Projeto definição salário: Etapa definição dos operadores

AND	Mínimo
OR	Máximo
Implicação	Mínimo
Agregação	Máximo
Defuzificação	Centro da gravidade

Após a implementação do projeto difuso, é o momento de realizar simulações para validar a funcionalidade do sistema. A Tabela 28 apresenta algumas simulações realizadas. Nessas simulações, o sistema recebe os parâmetros discretos para as variáveis de entrada e retorna um valor discreto como saída. A tabela apresenta os valores como uma caixa preta, porém quando realizadas as simulações pela ferramenta InFuzzy, são apresentadas de forma analítica diversas informações sobre o processo de simulação.

Uma característica que se observa no projeto “Definição Salário” é o fato do método de defuzificação “Centro da Gravidade” nunca alcançar os valores dos extremos do universo, isso pode ser visto na primeira simulação, onde o funcionário não possuía experiência e possuía o nível de escolaridade igual a 0 (zero), porém o salário não foi definido com o mínimo e sim com o valor de R\$ 630,00. Nesse momento percebe-se com clareza a importância da seleção adequada do método de defuzificação, a primeira simulação sendo executada com o método de defuzificação “Primeiro dos Máximos” teria como saída o valor de R\$ 540,00.

Tabela 28-Projeto definição salário: Etapa de simulação

Experiência (0 a 30 anos)	Escolaridade (0 a 10 nível)	Salário (R\$ 540,00 a R\$ 3000,00)
0	0	630,39
1	1	640,47
8	5	1.000,00

11	6	1.000,00
18	9	2.547,75
25	10	2.570,61

Fonte: do autor

4.9.2 Projeto difuso: Uso da lógica *fuzzy* para a otimização de sistemas energéticos baseados em energia limpa

Esse projeto teve como objetivo gerenciar o consumo de energia em relação a três variáveis de entrada, sendo elas: tempo, nível de bateria e velocidade dos ventos. Esse sistema realiza a captura de energia solar e eólica armazenando em um banco de baterias. Essa energia armazenada é utilizada na iluminação de corredores, banheiros e outros setores de uma empresa. O sistema difuso utilizado é responsável por determinar o comportamento do sistema no que diz respeito à economia de energia baseada na previsão do tempo e do nível de bateria, isso quer dizer, se a previsão é de tempo nublado e ventos vagarosos nos próximos dias, o sistema se ajustará de forma econômica, a fim de manter energia durante os dias de baixa produção energética. Esse projeto foi desenvolvido por Ederson Posselt, Fabiano Horn e Rolf Fredi Molz, sendo publicado e apresentado no XXIX Encontro Nacional de Engenharia de produção em 2009. O artigo e a implementação do projeto encontram-se na íntegra junto à instalação do projeto (POSSELT; MOLZ; HORN, 2009).

Esse projeto possui alguns diferenciais em relação ao projeto apresentado na seção anterior (Definição salário). A primeira delas é o fato de existirem dois blocos de regras, a segunda é o fato de uma variável de entrada estar associada a mais de um bloco de regras e por último, dois blocos de regras convergirem o resultado em apenas uma variável de saída. Essas observações ficam evidentes observando-se a Figura 69.

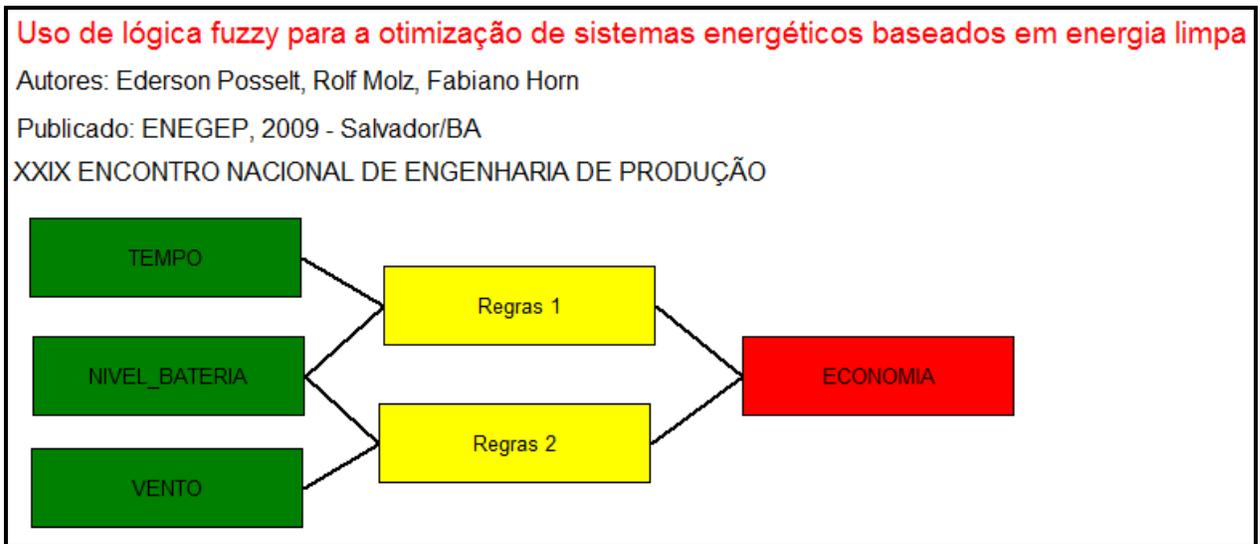


Figura 69- Implementação do projeto energia limpa na ferramenta InFuzzy

Fonte: do autor

A Tabela 29 apresenta o detalhamento das variáveis e termos linguísticos empregados neste projeto. Através dessa tabela é possível reconstruir as variáveis em qualquer sistema difuso existente no mercado. A unidade das variáveis “Vento” e “Tempo” refere-se o quanto um dia é ensolarado e a velocidade dos ventos, quanto melhor essas variáveis para a produção energética, mais próximo o valor será de 16. A unidade “Nível” utilizada na variável de saída “Economia” é destinada a informar o nível de economia do sistema; quanto mais próximo de zero, maior é a economia realizada pelo sistema, em outras palavras, a disponibilidade de energia é reduzida.

Tabela 29- Projeto Energia Limpa: Etapa definição variáveis

Termo linguístico	Função	Parâmetros
Variável de entrada: Nível Bateria (0 a 10 carga)		
Baixo	Rampa Esquerda	(0, 2, 4)
Média	Trapézio	(3, 5, 6, 8)
Alta	Rampa Direita	(6, 8, 10)
Variável de entrada: Vento (0 a 16 qualidade)		
Ruim	Rampa Esquerda	(0, 4, 6)
Médio	Trapézio	(4, 6, 10, 12)
Bom	Rampa Direita	(10, 12, 16)

Variável de entrada: Tempo (0 a 16 qualidade)		
Ruim	Rampa Esquerda	(0, 4, 6)
Médio	Trapézio	(4, 6, 10, 12)
Bom	Rampa Direita	(10, 12, 16)
Variável de saída: Economia (0 a 10 nível)		
Máxima	Rampa Esquerda	(0, 3, 5)
Alerta	Triangular	(3, 5, 7)
Padrão	Triangular	(5, 7, 9)
Nenhuma	Inclinação Direita	(8, 10)

Fonte: do autor

A Tabela 30 apresenta as regras dos dois blocos de regras envolvidos neste projeto. Essa tabela está dividida em duas colunas, sendo a primeira os antecedentes e a segunda os consequentes. É importante observar que nessa tabela estão listadas as regras dos dois blocos de regras presentes no projeto difuso.

Tabela 30- Projeto Energia Limpa: Etapa definição regras

Antecedente	Consequente
Bloco de Regras 1	
(NIVEL_BATERIA = Baixo) AND (TEMPO = Ruim)	(ECONOMIA = Máxima)
(NIVEL_BATERIA = Baixo) AND (TEMPO = Médio)	(ECONOMIA = Máxima)
(NIVEL_BATERIA = Baixo) AND (TEMPO = Bom)	(ECONOMIA = Máxima)
(NIVEL_BATERIA = Médio) AND (TEMPO = Ruim)	(ECONOMIA = Alerta)
(NIVEL_BATERIA = Médio) AND (TEMPO = Médio)	(ECONOMIA = Alerta)
(NIVEL_BATERIA = Médio) AND (TEMPO = Bom)	(ECONOMIA = Padrão)
(NIVEL_BATERIA = Alto) AND (TEMPO = Ruim)	(ECONOMIA = Padrão)
(NIVEL_BATERIA = Alto) AND (TEMPO = Médio)	(ECONOMIA = Padrão)
(NIVEL_BATERIA = Alto) AND (TEMPO = Bom)	(ECONOMIA = Nenhuma)
Bloco de Regras 2	
(NIVEL_BATERIA = Baixo) AND (VENTO = Ruim)	(ECONOMIA = Máxima)
(NIVEL_BATERIA = Baixo) AND (VENTO = Médio)	(ECONOMIA = Máxima)

(NIVEL_BATERIA = Baixo) AND (VENTO = Bom)	(ECONOMIA = Máxima)
(NIVEL_BATERIA = Médio) AND (VENTO = Ruim)	(ECONOMIA = Alerta)
(NIVEL_BATERIA = Médio) AND (VENTO = Médio)	(ECONOMIA = Alerta)
(NIVEL_BATERIA = Médio) AND (VENTO = Bom)	(ECONOMIA = Padrão)
(NIVEL_BATERIA = Alto) AND (VENTO = Ruim)	(ECONOMIA = Padrão)
(NIVEL_BATERIA = Alto) AND (VENTO = Médio)	(ECONOMIA = Padrão)
(NIVEL_BATERIA = Alto) AND (VENTO = Bom)	(ECONOMIA = Nenhuma)

Fonte: do autor

Na Tabela 31 estão definidas as funções para cada operador. Com as informações apresentadas até o momento é possível recriar esse projeto sem dificuldades. As configurações definidas neste projeto representam a máquina de inferência de Mamdani.

Tabela 31- Projeto Energia Limpa: Etapa definição dos parâmetros

AND	Mínimo
OR	Máximo
Implicação	Mínimo
Agregação	Máximo
Defuzificação	Centro da gravidade

Fonte: do autor

Os resultados obtidos no artigo conferem com os resultados obtidos no processo de simulação da ferramenta InFuzzy. Na Tabela 32 são apresentados os resultados de alguns testes de simulação executados, quanto menor o resultado da variável economia, maior deverá ser o gerenciamento de energia, a fim de economizar energia em momentos de baixa produção. Para exemplificar melhor, observa-se a primeira simulação, onde o nível da bateria é igual a 5, qualidade dos ventos é igual a 5 e a qualidade do tempo é igual a 5. Após o processo de fuzificação, inferência e defuzificação o sistema retornou o valor discreto de 5 para o nível de economia, o que quer dizer, que o sistema deve economizar energia.

Tabela 32- Projeto Energia Limpa: Etapa de simulação

Nível Bateria (0 a 10 carga)	Vento (0 a 16 qualidade)	Tempo (0 a 16 qualidade)	Economia (0 a 10 nível)
5	5	5	5
8	9	9	7
7	9	9	6
8	14	14	9,67

Fonte: do autor

4.9.3 Projeto difuso: Controle da velocidade de um ventilador aplicado em *hardware*

Esse projeto tem o objetivo de definir a velocidade do ventilador instalado em quadros de comandos baseado na temperatura interna, objetivando uma economia de energia e uma redução de ruído. Os ventiladores são aplicados nos quadros de comando com o objetivo de manter uma temperatura adequada para o funcionamento dos componentes eletrônicos nela instalados. Atualmente esses ventiladores se mantêm ligados à capacidade máxima de operação enquanto existir energia no quadro de comando, o que gera desperdício em temperaturas baixas. É possível afirmar que a economia de energia é baixa, porém essa aplicada a milhares de ventiladores, pode ser representativa.

A Figura 70 apresenta o *hardware* utilizado no projeto. Esse *hardware* é composto de uma CPU e um ventilador cedidos pela Impley Tecnologia Eletrônica SA. É possível observar os principais componentes presentes na placa, sendo eles:

- Modulo de rede Tibbo (2) – esse componente é responsável por receber de forma serial informações do processador e encaminhar um pacote UDP/IP para o destinatário configurado. Ao receber um pacote UDP/IP esse componente o encaminha de forma serial ao processador.

- Conexão elétrica com o ventilador (3) – saída de 12 volts para alimentação do ventilador.
- Processador (4) – Processador da linha 16F da Microchip, responsável por realizar a leitura do sensor de temperatura, encaminhar essa leitura para o servidor que está com a ferramenta InFuzzy em execução; na sequência recebe a resposta do servidor com a determinação da velocidade do ventilador, aplicando-a.
- Leitura do sensor de temperatura (5) – Entrada do sensor de temperatura, responsável por capturar a temperatura no interior do quadro de comando.
- Entrada de alimentação da CPU (6) – Alimentação de 12 volts para a CPU.

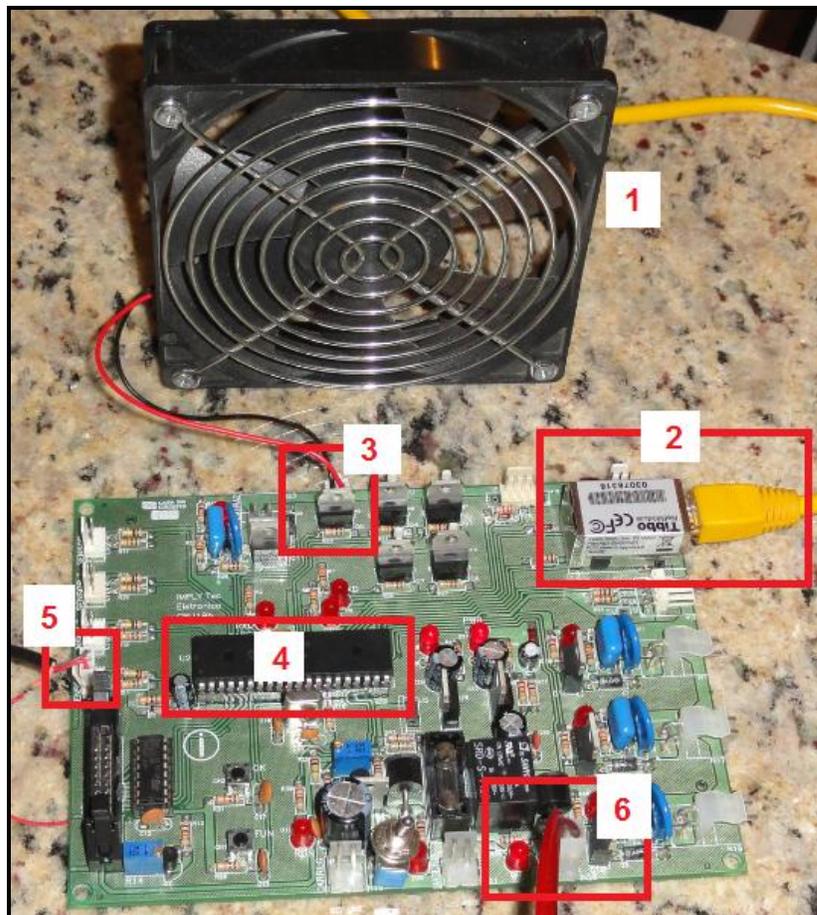


Figura 70- Projeto difuso - Controle de ventilador

Fonte: do autor

A Figura 71 apresenta o projeto implementado na ferramenta InFuzzy, observando essa imagem é possível verificar que existe apenas uma variável de entrada “Temperatura”, um bloco de regras e apenas uma variável de saída.

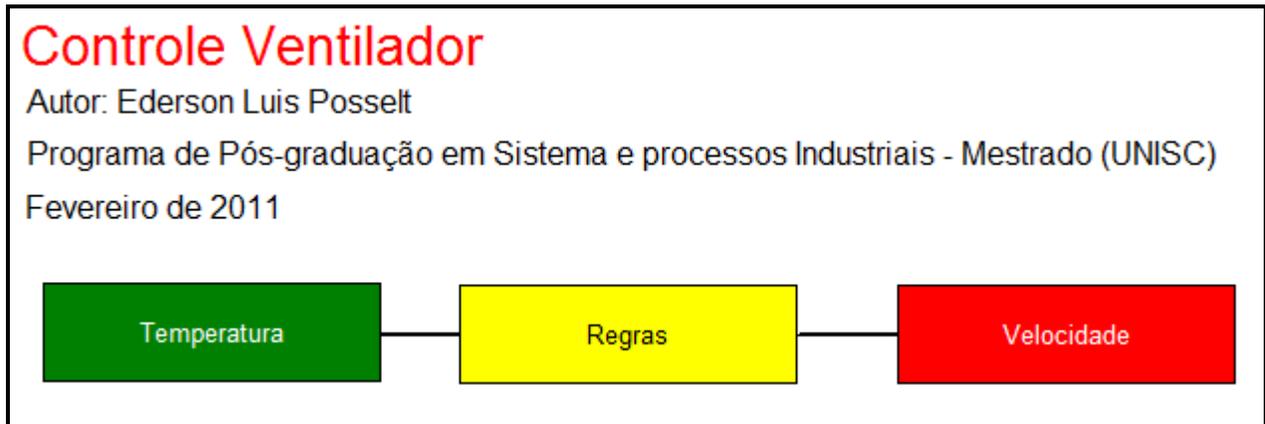


Figura 71- Projeto controle de ventilador implementado na ferramenta InFuzzy

Fonte: do autor

A Tabela 33 apresenta os termos linguísticos das variáveis de entrada temperatura e variável de saída velocidade.

Tabela 33- Projeto difuso controle ventilador: Definição das variáveis

Termo linguístico	Função	Parâmetros
Variável de entrada: Temperatura (-20 a 100 Celsius)		
Muito Baixa	Rampa Esquerda	(-20, -10, 5)
Baixa	Triangular	(0 ,10, 20)
Média	Triangular	(10, 20, 30)
Muito Alta	Rampa Direita	(40, 50, 100)
Variável de saída: Velocidade (0 a 100 Velocidade)		
Baixa	Triangular	(1 , 20 , 40)
Média	Triangular	(30 , 60 , 90)
Alta	Rampa Direita	(80, 90, 100)

Fonte: do autor

A Tabela 34 apresenta as regras definidas no projeto. A primeira coluna da tabela apresenta os antecedentes e a segunda coluna apresenta os consequentes.

Tabela 34- Projeto difuso controle ventilador: Definição das regras

Antecedente (Se)	Consequente (Então)
(Temperatura = Muito Baixa)	(Velocidade = Zero)
(Temperatura = Baixa)	(Velocidade = Zero)
(Temperatura = Média)	(Velocidade = Baixa)
(Temperatura = Alta)	(Velocidade = Média)
(Temperatura = Muito Alta)	(Velocidade = Alta)

Fonte: do autor

As funções utilizadas para os operadores encontram-se na Tabela 35. Com a definição das variáveis e seus termos linguísticos, das regras e dos operadores e suas funções, têm-se todas as informações necessárias para reproduzir esse projeto.

Tabela 35- Projeto difuso controle ventilador: Definição dos parâmetros

AND	Mínimo
OR	Máximo
Implicação	Mínimo
Agregação	Máximo
Defuzificação	Centro da gravidade

Fonte: do autor

Esse projeto tem como objetivo exemplificar a integração do *hardware* com a ferramenta InFuzzy através do protocolo UDP/IP. A Tabela 36 apresenta o *debug* do processo de comunicação. Observando-se esse protocolo, percebe-se as seguintes informações:

- A primeira linha representa a abertura da porta 5678 UDP.
- A comunicação sempre inicia pelo *hardware*, com intervalos próximos de 1 segundo.

- Em todos os casos é possível observar o pacote de origem e o de retorno da ferramenta InFuzzy.
- A última linha demonstra o fechamento da porta de comunicação.

Tabela 36- Projeto difuso controle ventilador: *Debug* da comunicação

```

00:41:16.021: Porta UDP 5678 Aberta.
00:41:41.449: Origem : 192.168.3.110:4200 Pacote: $EXEC;Temperatura=0;
00:41:41.512: Destino: 192.168.3.110:4200 Pacote: #EXEC;SUCES;Velocidade=0,00;
00:41:42.619: Origem : 192.168.3.110:4200 Pacote: $EXEC;Temperatura=0;
00:41:42.651: Destino: 192.168.3.110:4200 Pacote: #EXEC;SUCES;Velocidade=0,00;
00:41:44.257: Origem : 192.168.3.110:4200 Pacote: $EXEC;Temperatura=0;
00:41:44.304: Destino: 192.168.3.110:4200 Pacote: #EXEC;SUCES;Velocidade=0,00;
00:41:45.895: Origem : 192.168.3.110:4200 Pacote: $EXEC;Temperatura=0;
00:41:45.942: Destino: 192.168.3.110:4200 Pacote: #EXEC;SUCES;Velocidade=0,00;
00:41:47.533: Origem : 192.168.3.110:4200 Pacote: $EXEC;Temperatura=5;
00:41:47.565: Destino: 192.168.3.110:4200 Pacote: #EXEC;SUCES;Velocidade=0,00;
00:41:49.171: Origem : 192.168.3.110:4200 Pacote: $EXEC;Temperatura=6;
00:41:49.203: Destino: 192.168.3.110:4200 Pacote: #EXEC;SUCES;Velocidade=0,00;
00:41:50.809: Origem : 192.168.3.110:4200 Pacote: $EXEC;Temperatura=9;
00:41:50.841: Destino: 192.168.3.110:4200 Pacote: #EXEC;SUCES;Velocidade=0,00;
00:41:52.447: Origem : 192.168.3.110:4200 Pacote: $EXEC;Temperatura=14;
00:41:52.494: Destino: 192.168.3.110:4200 Pacote: #EXEC;SUCES;Velocidade=20,31;
00:41:54.085: Origem : 192.168.3.110:4200 Pacote: $EXEC;Temperatura=12;
00:41:54.132: Destino: 192.168.3.110:4200 Pacote: #EXEC;SUCES;Velocidade=20,22;
00:41:55.723: Origem : 192.168.3.110:4200 Pacote: $EXEC;Temperatura=20;
00:41:55.770: Destino: 192.168.3.110:4200 Pacote: #EXEC;SUCES;Velocidade=20,33;
00:41:57.361: Origem : 192.168.3.110:4200 Pacote: $EXEC;Temperatura=18;
00:41:57.393: Destino: 192.168.3.110:4200 Pacote: #EXEC;SUCES;Velocidade=20,32;
00:41:58.999: Origem : 192.168.3.110:4200 Pacote: $EXEC;Temperatura=19;
00:41:59.031: Destino: 192.168.3.110:4200 Pacote: #EXEC;SUCES;Velocidade=20,33;
00:42:00.637: Origem : 192.168.3.110:4200 Pacote: $EXEC;Temperatura=17;
00:42:00.669: Destino: 192.168.3.110:4200 Pacote: #EXEC;SUCES;Velocidade=20,32;
00:42:02.275: Origem : 192.168.3.110:4200 Pacote: $EXEC;Temperatura=16;
00:42:02.322: Destino: 192.168.3.110:4200 Pacote: #EXEC;SUCES;Velocidade=20,32;
00:42:03.913: Origem : 192.168.3.110:4200 Pacote: $EXEC;Temperatura=16;
00:42:03.960: Destino: 192.168.3.110:4200 Pacote: #EXEC;SUCES;Velocidade=20,32;
00:42:05.551: Origem : 192.168.3.110:4200 Pacote: $EXEC;Temperatura=18;
00:42:05.598: Destino: 192.168.3.110:4200 Pacote: #EXEC;SUCES;Velocidade=20,32;
00:42:07.189: Origem : 192.168.3.110:4200 Pacote: $EXEC;Temperatura=24;
00:42:07.221: Destino: 192.168.3.110:4200 Pacote: #EXEC;SUCES;Velocidade=20,37;
00:42:08.827: Origem : 192.168.3.110:4200 Pacote: $EXEC;Temperatura=24;
00:42:08.859: Destino: 192.168.3.110:4200 Pacote: #EXEC;SUCES;Velocidade=20,37;
00:42:10.465: Origem : 192.168.3.110:4200 Pacote: $EXEC;Temperatura=23;
00:42:10.512: Destino: 192.168.3.110:4200 Pacote: #EXEC;SUCES;Velocidade=20,36;
00:42:12.103: Origem : 192.168.3.110:4200 Pacote: $EXEC;Temperatura=36;
00:42:12.150: Destino: 192.168.3.110:4200 Pacote: #EXEC;SUCES;Velocidade=60,00;
00:42:13.741: Origem : 192.168.3.110:4200 Pacote: $EXEC;Temperatura=40;
00:42:13.788: Destino: 192.168.3.110:4200 Pacote: #EXEC;SUCES;Velocidade=50,00;
00:42:15.395: Origem : 192.168.3.110:4200 Pacote: $EXEC;Temperatura=45;
00:42:15.426: Destino: 192.168.3.110:4200 Pacote: #EXEC;SUCES;Velocidade=91,22;

```

```

00:42:17.017: Origem : 192.168.3.110:4200 Pacote: $EXEC;Temperatura=51;
00:42:17.049: Destino: 192.168.3.110:4200 Pacote: #EXEC;SUCES;Velocidade=92,25;
00:42:18.655: Origem : 192.168.3.110:4200 Pacote: $EXEC;Temperatura=52;
00:42:18.702: Destino: 192.168.3.110:4200 Pacote: #EXEC;SUCES;Velocidade=92,25;
00:42:20.293: Origem : 192.168.3.110:4200 Pacote: $EXEC;Temperatura=66;
00:42:20.340: Destino: 192.168.3.110:4200 Pacote: #EXEC;SUCES;Velocidade=92,25;
00:42:21.931: Origem : 192.168.3.110:4200 Pacote: $EXEC;Temperatura=69;
00:42:21.978: Destino: 192.168.3.110:4200 Pacote: #EXEC;SUCES;Velocidade=92,25;
00:42:23.569: Origem : 192.168.3.110:4200 Pacote: $EXEC;Temperatura=73;
00:42:23.616: Destino: 192.168.3.110:4200 Pacote: #EXEC;SUCES;Velocidade=92,25;
00:42:25.223: Origem : 192.168.3.110:4200 Pacote: $EXEC;Temperatura=72;
00:42:25.254: Destino: 192.168.3.110:4200 Pacote: #EXEC;SUCES;Velocidade=92,25;
00:42:26.627: Porta UDP 5678 Fechada.

```

Fonte: do autor

As informações apresentadas na Tabela 36 são apresentadas no gráfico da Figura 72, porém apenas a variável de saída. Em uma observação, pode-se observar que a temperatura aumentou gradativamente e com isso o retorno da velocidade do ventilador também aumentou. Além disso, pode-se observar que ocorreram aproximadamente 27 interações entre o *hardware* e a ferramenta InFuzzy.



Figura 72- Projeto difuso controle ventilador: Gráfico de saída da variável "Velocidade"

Fonte: do autor

Junto ao instalador é disponibilizado um vídeo que demonstra o funcionamento dessa aplicação. Através desse vídeo e da vídeo aula de instruções sobre integração utilizando o protocolo UDP/IP, este processo é esclarecido.

4.10 Diferenciais da ferramenta InFuzzy em relação as demais ferramentas apresentadas

A ferramenta InFuzzy comparada com as ferramentas existentes no mercado trás uma série de vantagens e melhorias. Nos próximos parágrafos serão abordados os dez principais pontos da ferramenta InFuzzy, utilizados como critérios de comparação com outras ferramentas.

Primeiro: Está relacionado à interface gráfica. Trata-se de uma aplicação MDI (*Multiple Document Interface*), propriedade que permite a manipulação de um ou mais projetos simultaneamente. A interface é de fácil manuseio baseada em metáforas com funções *drag-and-drop*. A interface de uma aplicação define o grau de aceitação pelo público e para garantir uma boa interface, essa foi submetida à técnica de *checklist* baseada nas questões obtidas no *software Ergolist*.

Segundo: Está relacionada à modelagem de sistemas difusos. Nessa etapa, a ferramenta InFuzzy mantém uma estrutura de projeto baseada nos conceitos da lógica difusa. A estrutura é reconhecida com clareza através das cores e conexões entre os blocos de variáveis de entrada, blocos de regras e blocos de regras de saída.

Terceiro: O sistema permite adicionar juntamente ao modelo mensagens de texto explicativas sobre o projeto difuso modelado, formatadas de acordo com as preferências do usuário em relação à fonte e posicionamento.

Quarto: O sistema permite a divisão das regras em vários blocos de acordo com as variáveis de entrada e saída.

Quinto: Definição de uma lista de entradas a serem simuladas. Essa lista é persistida e pode ser executada novamente quando houver mudanças nas definições do modelo difuso.

Sexto: O sistema não é uma caixa preta, isso significa que todas as etapas para a execução de um modelo difuso são descritas de forma a elucidar o resultado obtido. Essa característica foi visualizada apenas na ferramenta InFuzzy.

Sétimo: Histórico de resultados obtidos com a execução das simulações do modelo difuso implementado é exibido ao usuário através de um gráfico temporal. Somente a ferramenta InFuzzy apresenta essa característica.

Oitavo: Permite a integração com modelos difusos com *softwares* de terceiros através de pacotes UDP/IP. Essa comunicação está documentada através de um protocolo definido no projeto, além disso, é acoplado no projeto um *software* chamado “InFuzzy UDP Tool”, responsável por demonstrar a integração entre os diferentes sistemas. Essa característica foi encontrada apenas na ferramenta InFuzzy.

Nono: Documentação completa. O projeto é baseado nesta dissertação, além disso, possui um manual e vídeo aulas elaborados pelo autor. Em nenhuma das ferramentas estudadas foi encontrada vídeo aulas desenvolvidas pelo autor.

Décimo: Única ferramenta nacional de modelagem de sistemas difusos gráfica.

InFuzzy se torna forte a partir do momento em que reúne funcionalidades dispersas nas mais diversas ferramentas estudadas além de agregar funcionalidades inovadoras e fundamentais.

CONCLUSÃO

A lógica difusa é uma técnica capaz de operar sobre dados imprecisos, mostrando-se mais adequada para tratar modos de raciocínio aproximados ao invés de exatos. Atualmente essa técnica é pouco difundida na indústria e em empresas de desenvolvimento de *software*. A carência observada a partir da análise das ferramentas existentes abriu espaço para um *software* gratuito, com interação gráfica de fácil manuseio, documentação completa, capacidade de realizar simulações e integração com outros sistemas através do protocolo IP/UDP.

Com base na problemática da pesquisa apresentada na metodologia, houve a busca de subsídios necessários para atingir os objetivos propostos pelo trabalho. Após a concretização de várias etapas, obteve-se como resultado um *software* de distribuição gratuita, com as seguintes características:

- 1) A instalação do *software* é executada de forma rápida e simples, método já conhecido pela maioria dos usuários de computadores. Todos os arquivos necessários para o funcionamento e aprendizado encontram-se no instalador, sendo eles: ferramenta InFuzzy, DLLs (*Dynamic-link library*) necessárias, manual do *software*, a presente dissertação, exemplos de projetos modelados, vídeo aulas e aplicativo para desinstalar o *software*.
- 2) Interface gráfica amigável de fácil aprendizado, homologada através da técnica de *Checklist*.
- 3) Módulo de simulação com *trace*, esse módulo permite que a cada simulação sejam acompanhados os procedimentos realizados para a obtenção dos resultados, além da visualização do processo de defuzificação através de um gráfico.
- 4) Módulo de integração com outros *softwares* através do protocolo IP/UDP. Esse módulo é um diferencial em relação aos *softwares* de modelagem analisados neste trabalho. Esse módulo permite ao usuário testar de forma ágil e fácil a lógica difusa para uma determinada aplicação com o mínimo de esforço.

- 5) As vídeo aulas, que totalizam um tempo superior a 105 minutos, demonstram de forma detalhada o uso da ferramenta InFuzzy na seguinte organização: todos os módulos do sistema (definição de variáveis, termos lingüísticos, regras, simulações), desenvolvimento de um sistema difuso completo, desenvolvimento de um sistema em hardware integrando com a ferramenta InFuzzy e o uso da ferramenta InFuzzy UDP Tool.

A validação da interface gráfica ocorreu através de listas de verificação (*Checklist*) em várias etapas do projeto. As questões utilizadas na validação foram extraídas do *software* Ergolist. Após a execução, observou-se que 85% dos itens aplicáveis foram atendidos, determinando que a interface está em sua grande maioria de acordo com as normas de usabilidade e ergonomia, proporcionando facilidade de uso e aprendizado ao usuário. Os 15% não atingidos não foram cumpridos devido ao curto tempo necessário para o desenvolvimento. A etapa de validação pode ser visualizada na íntegra na seção 4.2 e no ANEXO B. A interface gráfica foi avaliada na seguinte situação: o *software* foi instalado para dois usuários graduados na área de tecnologia com pouco ou nenhum conhecimento sobre lógica difusa. Após uma breve explanação sobre o *software* e a visualização das vídeo aulas, os dois usuários obtiveram o entendimento sobre o funcionamento da lógica difusa e foram capazes de modelar sistemas, sendo que um deles desenvolveu o modelo difuso para o controle de um pêndulo invertido e o outro um controle de umidade. Além dessas implementações, um dos usuários implementou o *software* do PIC utilizado na seção 4.9.3 onde é apresentada a integração da ferramenta InFuzzy com um periférico.

Como forma de validar os resultados da ferramenta InFuzzy, foi implementado um determinado modelo difuso no *software* Matlab[®] e na ferramenta InFuzzy, observando-se que os resultados em ambas as ferramentas coincidia nas mais diversas combinações. A descrição dos testes realizados está detalhada na seção 4.7. Outra forma de validar a ferramenta ocorreu através da modelagem de sistemas difusos encontrados em dissertações, trabalhos de conclusão de curso e artigos. Nestes casos, observou-se os resultados obtidos pelos estudantes e os resultados apresentados na ferramenta InFuzzy. Os testes realizados podem ser visualizados na íntegra na seção 4.9 e nos exemplos que acompanham a instalação do sistema.

Como forma de ampliar as funcionalidades da ferramenta InFuzzy, sugere-se para trabalhos futuros o desenvolvimento de cinco funcionalidades descritas a seguir:

- 1) Exportação de código fonte: Esse módulo terá a responsabilidade de transformar o modelo difuso implementado na ferramenta InFuzzy em código fonte para um ou mais linguagens como: C, C++, C#, Java, Pascal, PHP, ASP, entre outras. Essa implementação pode ocorrer basicamente de duas formas, sendo uma delas na edição do código fonte da ferramenta InFuzzy, onde essa já exportaria o código fonte; ou no desenvolvimento de uma ferramenta independente, que teria como entrada o arquivo gerado pela ferramenta InFuzzy e como saída um arquivo de código fonte.
- 2) Desenvolvimento de modelos de inferência de interpolação, tendo como exemplo o modelo de Takagi-Sugeno. Neste trabalho, foram desenvolvidos os modelos de inferência clássicos de Mamdani e suas variações. Essa implementação teria que ser realizada diretamente no código fonte da ferramenta InFuzzy.
- 3) A ferramenta InFuzzy permite atualmente a integração com outros *softwares* utilizando o protocolo de rede IP/UDP. Para estender essa facilidade será importante que sejam implementados módulos de integração através da porta serial, porta paralela, porta USB, protocolo de rede IP/TCP. Com essa gama de formas de integração, a ferramenta InFuzzy se torna uma boa ferramenta no que diz respeito à facilidade de testes e implementações de modelos difusos.
- 4) Ampliação das funções existentes na ferramenta InFuzzy como: operador AND, operador OR, implicação, agregação, métodos de fuzificação e novas funções de fuzificação.
- 5) Desenvolvimento de um validador de regras, esse módulo teria como objetivo evitar que o usuário definisse regras contraditórias. Para exemplificar essa situação, segue o exemplo: “**Se** temperatura é quente **então** ar condicionado é máximo” “**Se** temperatura é fria **então** ar condicionado é máximo”. Essas situações são conflitantes para o gerenciamento da temperatura de uma determinada sala na qual se deseja climatizar.

A ferramenta InFuzzy, desenvolvida neste trabalho, resultou em um *software* capaz de alterar o cenário atual, aproximando a teoria da prática, gerando resultados ao usuário em um tempo curto, capaz de validar a técnica de lógica difusa em uma determinada situação. Este projeto mostra a sua importância a partir do momento em que introduz no meio acadêmico e industrial uma ferramenta gratuita que possibilita modelar, implementar, simular e integrar modelos difusos, suprimindo uma carência existente. A ferramenta InFuzzy contribuirá para a divulgação do uso de lógica difusa como solução em problemas reais e poderá impulsionar o uso da técnica na indústria.

REFERÊNCIAS

- BITTENCOURT, J. R.; OSÓRIO, F.. **FuzzyF - Fuzzy Logic Framework: Uma Solução Software Livre para o Desenvolvimento, Ensino e Pesquisa de Aplicações de Inteligência Artificial Multiplataforma**. In: WORKSHOP SOBRE SOFTWARE LIVRE, 3., 2002, Porto Alegre. **Anais...** . Porto Alegre, RS: Sociedade Brasileira de Computação, 2002. p. 58 – 61
- CALDEIRA, A. M. et al. **Inteligência Computacional Aplicada à Administração, Economia e Engenharia em Matlab**. São Paulo: Thomson, 2007. 370 p.
- CERVO, A. L.; BERVIAN, P. A. **Metodologia científica**. 4ª edição, São Paulo: Makron Books, 1996, 90 p.
- COSTA, A. et al. **LÓGICA FUZZY: CONCEITOS E APLICAÇÕES**. Disponível em: <http://www.inf.unisinos.br/~cazella/dss/fuzzy_relatorio.pdf>. Acesso em: 10 Jun. 2009.
- CYBIS, W. A. **Engenharia de Usabilidade: uma abordagem ergonômica**. Laboratório de Utilizabilidade de Informática. Florianópolis, 2003
- DIX, A. et al. **Human-Computer Interaction**. 2ª Ed. Harlow: Prentice Hall, 2001
- DRIANKOV, D.; HELLENDROORN, H.; REINFRANK, M.. **An Introduction to Fuzzy Control, Second Edition**. Berlin: Springer, 1996
- ERGOLIST. **Software para avaliação de interfaces**. Disponível em: <<http://www.labiutil.inf.ufsc.br/ergolist/quest.htm>>. Acessado em: Nov de 2009.
- ERICKSON, T. D.. **Working with interface metaphors. In Human-Computer interaction: Toward the Year 2000**, R. M. Baecker, et al.. San Francisco: Morgan Kaufmann Publishers, 1995, p. 147-151.
- FERREIRA, J. M. P.. **Lógica Difusa (Fuzzy Logic)**. Material de aula da Universidade Estadual de Maringá, 2007
- FROZZA, R.. **Lógica Difusa (Fuzzy Logic)**. Material de aula do Programa de Pós-Graduação em Sistemas e processos industriais, UNISC, 2007.
- FUJIMOTO, R. Y.. **Diagnóstico Automático de Defeitos em Rolamentos Baseados em Lógica Fuzzy**. 2005. 158 f. Dissertação (Mestrado) - Departamento de Engenharia Mecânica, Universidade de São Paulo Escola Politécnica, São Paulo, 2005.
- FUZZYTECH. **Fuzzy Logic Advantage in Your Design**. Disponível em: <www.fuzzytech.com>. Acessado em: Nov de 2009.
- GIL, A. C.. **Como elaborar projetos de pesquisa**. São Paulo: Atlas, 1991

IMSE-CNM. **Fuzzy Logic Design Tools**. Manual do software. 2003

JÚNIOR, I. N.; NOVAKOWSKI, S. **Lógica Fuzzy Sistemas de Apoio a Decisão**. Rio Grande do Sul, UNISINOS, Abr. 2005. Disponível em: <http://www.inf.unisinos.br/~cazella/dss/SAD_Logica_Fuzzy.pdf>. Acesso em: maio de 2008

KOHAGURA, T.. **Lógica Fuzzy e suas aplicações**. 2007. 61 f. Trabalho de Conclusão de Curso (3) - Curso de Ciência da Computação, Departamento de Computação, Universidade Estadual de Londrina, Londrina, 2007.

LANDAUER T. K.; PRABHU P. V.. **Handbook of human-computer interaction**. In: KIERAS, D.. *A guide to GOMS Model Usability Evolution using NGOMSL*. Amsterdam, 1997. p.733-765.

LUNARDI, A.. **Uma estratégia para a avaliação de usabilidade com suporte de um ambiente corporativo**. 2004. Graduação (Graduado) - Curso de Ciência da Computação, Departamento de Informática, UNISC, Santa Cruz do Sul, RS, 2004.

MANO A.; CAMPOS J. C.. **Aplicação de um Cognitive Walkthrough - estudo de caso**. In *Interação 2004 - 1ª. Conferência Nacional em Interação Pessoa-Máquina*, p. 256-258. Grupo Português de Computação Gráfica, 2004.

MAMDANI, E. H. **Applications of fuzzy algorithms for control of simple dynamic plant**. Proc. IEEE 121, vol. 12, p. 1585-1588, 1973.

MARCONI, M. de A.; LAKATOS, E. M. **Metodologia científica**. 5.ed. São Paulo: Atlas, 2007. 312p.

MATHWORKS. **Fuzzy Logic Toolbox: Users Guide**. Disponível em: <<http://www.mathworks.com/>>. Acesso em: 01 ago. 2009.

NIELSEN, J. **Usability Engineering**. San Francisco, California, United States: Morgan Kaufmann, 1993.

PREECE, J.; ROGERS, Y.; SHARP, H.. **Interaction design: Beyond human-computer interaction**. USA: Jhon Wiley & Sons, 2002.

POSSELT, E. L.; MOLZ, R. F.; HORN, F.. **Uso De Lógica Fuzzy Para A Otimização De Sistemas Energéticos Baseados Em Energia Limpa**. In: ENCONTRO NACIONAL DE ENGENHARIA DE PRODUÇÃO, XXIX, 2009, Salvador, BA.

REZENDE, S. O.. **Sistemas inteligentes Fundamentos e Aplicações**. In: ALMEIDA, P. E. M.; EVSUKOFF, A. G. de (Org.) *Sistemas Fuzzy*. São Paulo: Manole, 2003. p.169-201.

ROCHA, Heloisa V., BARANAUSKAS, Maria C. C. **Design e avaliação de interfaces humano-computador**. Disponível em: <http://www.nied.unicamp.br>. Data de acesso: 01/11/09.

ROCKON FUZZY TOOL, **The Best Free Fuzzy Control And Simulation Tool**. Disponível em: < <http://www.timtomtam.de/rockonfuzzy/>>. Acessado em: Nov de 2009.

ROSSON, M. B.; CARROLL J. M. **Usability Engineering – Scenario-based Development of Human-Computer Interaction Desing**. San Diego: Academic Press, 2002.

SANDRI, S.; CORREA, C.. **Lógica Nebulosa**. In: ESCOLA DE REDES NEURAIAS, 5., 1999, São José Dos Campos. Anais... . São José Dos Campos: ITA, 1999. p. 73 - 90

SCILAB, **The open source platform for numerical computation**. Disponível em: < <http://www.scilab.org/> Acessado em: Nov de 2009.

SOUSA, C. P. de. **Lógica Fuzzy: Aplicações**. Disponível em: <<http://www.deti.ufc.br/~pimentel/fuzzy/aplic.html>>. Acesso em: 01 out. 2010.

SUGENO, M.. **An introductory survey of fuzzy control**. Information Sciences 36, p. 59-83, 1985.

TANSCHKEIT, R.. **Sistemas fuzzy**. In: VI Simpósio Brasileiro de Automação Inteligente, 2003, Bauru, SP. Anais de Minicursos do VI SBAI, 2003. p. 35.

TOGAI INFRALOGIC . **The World's Source For Fuzzy Logic Solutions**. Disponível em: <<http://www.ortech-engr.com/fuzzy/togai.html>>. Acessado em: Nov de 2009.

VELASCO, O. G. D.. **Sistema de lógica difusa**. Disponível em: <[www.ing.unal.edu.co/~ogduarte /softwareDetallado.htm](http://www.ing.unal.edu.co/~ogduarte/softwareDetallado.htm)> Acessado em: Dez de 2009

ZADEH, L. A.. *Fuzzy Sets, Information and Control*, Vol. 8, p. 338-353, 1965.

ZADEH, L. A.. **BISC: The Berkeley Initiative in Soft Computing**. Disponível em: <<http://www.cs.berkeley.edu/~zadeh/>>. Acesso em: 10 nov. 2009.

ZHU, D.. **FIDE -Fuzzy Inference Development Environment An Intelligent Software Tool for Mutli-Sensor Data Fusion**. Apronix Application Note, 1997.

WEBER, L.; KLEIN, P. A. T.. **Aplicação da Lógica Fuzzy em Software e Hardware**. Canoas: Ulbra, 2003. 110 p.

WOLFRAM. *MATHEMATICA FUZZY LOGIC*, **The most flexible environment for exploring fuzzy systems**. Wolfram Research, 2003. Disponível em: <www.wolfram.com>. Acessado em: Ago de 2009.

ANEXO A: Tabela comparativa de sistemas difusos

Na Tabela 38 é apresentado um comparativo das principais características das ferramentas de sistemas difusos analisadas. Para que seja possível entendimento da Tabela 38, a Tabela 37 traz um índice das ferramentas analisadas. É importante lembrar que muitas das informações foram extraídas das documentações das ferramentas, e isso se deve pelo fato dessas não estarem disponíveis em versões de demonstração. Logicamente, algumas das ferramentas possuem funcionalidades além das relacionadas na Tabela 38, porém foram utilizadas as características fundamentais para o desenvolvimento de sistemas difusos.

Tabela 37- Índice das ferramentas analisadas

Índice	Software
1	FuzzyF
2	Fuzzy Logic ToolBox™
3	Mathematica® Fuzzy Logic
4	FIDE™
5	TILShell
6	FuzzyTECH®
7	RockOn Fuzzy Tool
8	SciLab Fuzzy Fuzzy Tool
9	UNFUZZY
10	XFuzzy
11	InFuzzy

Fonte: Tabela formada a partir das pesquisas realizadas neste projeto

Tabela 38 - Comparativo das principais características de uma ferramenta difusa.

Atributos/ Ferramentas	1	2	3	4	5	6	7	8	9	10	11
Máquina de inferência											
Mamdani	x	x	x	x			x		x	x	x

ANEXO B: Lista de checagem de ergonomia ErgoList

A Tabela 39 apresenta as questões utilizadas para a avaliação da ergonomia do *software* desenvolvido. Essas questões estão divididas em grupos conforme semelhança de tema. Essa lista de questões foi extraída do *software* Ergolist, disponível em: <http://www.labiutil.inf.ufsc.br/ergolist/quest.htm>.

Tabela 39 - Questões de validação de ergonomia extraídas do ErgoList

Nro	Questões	Sim	Não	Não Aplicável
Avaliação de Presteza: Verifique se o sistema informa e conduz o usuário durante a interação.				
1	Os títulos de telas, janelas e caixas de diálogo estão no alto, centrados ou justificados à esquerda?	X		
2	Todos os campos e mostradores de dados possuem rótulos de identificação?	X		
3	Caso o dado a entrar possua um formato particular, esse formato encontra-se descrito na tela?	X		
4	As unidades para a entrada ou apresentação de dados métricos ou financeiros encontram-se descritas na tela?	X		
5	Os rótulos dos campos contêm um elemento específico, por exemplo, ":", como convite às entradas de dados?	X		
6	Caso o dado a entrar possua valores aceitáveis esses valores encontram-se descritos na tela?	X		
7	Listas longas apresentam indicadores de continuação, de quantidade de itens e de páginas?			X
8	As tabelas apresentam cabeçalhos para linhas e colunas consistentes e distinguíveis dos dados apresentados?	X		
9	Os gráficos possuem um título geral e rótulos para seus eixos?	X		

10	Os botões que comandam a apresentação de caixas de diálogo apresentam em seus rótulos o sinal "..." como indicador da continuidade e do diálogo?	X		
11	As páginas de menus possuem títulos, cabeçalhos ou convites à entrada?	X		
12	As opções de menu que levam a outros painéis de menu apresentam o sinal ">" como indicador desse fato?	X		
13	O usuário encontra disponíveis as informações necessárias para suas ações?	X		
14	Nas caixas de mensagens de erro, o botão de comando "AJUDA" está sempre presente?		X	
15	A resposta para uma solicitação de ajuda do usuário está estruturada no contexto da tarefa e da transação corrente?		X	
16	Existe a possibilidade do usuário obter a lista de comandos básicos da linguagem?			X
17	Na ocorrência de erros, o usuário pode acessar todas as informações necessárias ao diagnóstico e à solução do problema?			X
Avaliação Agrupamento por localização: Verifique se a distribuição espacial dos itens traduz as relações entre as informações.				
1	O espaço de apresentação está diagramado em pequenas zonas funcionais?	X		
2	A disposição dos objetos de interação de uma caixa de diálogo segue uma ordem lógica?	X		
3	Nos agrupamentos de dados, os itens estão organizados espacialmente segundo um critério lógico?	X		
4	Os códigos das teclas aceleradoras de opções de menu estão localizados à direita do nome da opção?			X
5	Nas listas de seleção, as opções estão organizadas segundo alguma ordem lógica?	X		
6	Os painéis de menus são formados a partir de um critério lógico de agrupamento de opções?	X		

7	Dentro de um painel de menu, as opções mutuamente exclusivas ou interdependentes estão agrupadas e separadas das demais?	X		
8	As opções dentro de um painel de menu estão ordenadas segundo algum critério lógico?	X		
9	A definição da opção de menu selecionada por default segue algum critério?			X
10	Os grupos de botões de comando estão dispostos em coluna e à direita, ou em linha e abaixo dos objetos aos quais estão associados?	X		
11	O botão de comando selecionado por default está na posição mais alta, se os botões estão dispostos verticalmente, ou na mais à esquerda, se os botões estão dispostos horizontalmente?			X
Agrupamento por formato: Verifique os formatos dos itens como meio de transmitir associações e diferenças.				
1	Os controles e comandos encontram-se visualmente diferenciados das informações apresentadas nas telas?	X		
2	Códigos visuais são empregados para associar diferentes categorias de dados distribuídos de forma dispersa nas telas?	X		
3	Os diferentes tipos de elementos de uma tela de consulta (dados, comandos e instruções) são visualmente distintos uns dos outros?	X		
4	Os rótulos são visualmente diferentes dos dados aos quais estão associados?	X		
5	Os rótulos são visualmente diferentes dos dados aos quais estão associados?	X		
6	Em situações anormais, os dados críticos e que requeiram atenção imediata são diferenciados através do uso de cores brilhantes como, por exemplo, o vermelho ou o rosa?	X		
7	Sinais sonoros são empregados para alertar os usuários em relação a uma apresentação visual?			X

8	Na apresentação de textos, os recursos de estilo, como itálico, negrito, sublinhado ou diferentes fontes são empregados para salientar palavras ou noções importantes?			X
9	Os itens selecionados para alteração, atualização ou acionamento estão destacados dos outros?	X		
10	Nas situações de alarme e nas telas de alta densidade de informação, o recurso de intermitência visual é empregado para salientar dados e informações?			X
11	Os campos obrigatórios são diferenciados dos campos opcionais de forma visualmente clara?	X		
12	Nas caixas de mensagens, o botão selecionado por default tem uma apresentação visual suficientemente distinta dos outros?	X		
13	Em situações em que se exija atenção especial do usuário, as mensagens de alerta e de aviso são apresentadas de maneira distinta?			X
14	A forma do cursor do mouse é diferente da de qualquer outro item apresentado?	X		
15	As formas de cursores (dois ou mais) apresentados simultaneamente são suficientemente distintas umas das outras?	X		
16	As caixas de agrupamento são empregadas para realçar um grupo de dados relacionados?	X		
17	Quando apresenta opções não disponíveis no momento, o sistema as mostra de forma diferenciada visualmente?		X	
Avaliação Legibilidade: Verifique a legibilidade das informações apresentadas nas telas do sistema.				
1	As áreas livres são usadas para separar grupos lógicos em vez de tê-los todos de um só lado da tela, caixa ou janela?	X		
2	Os grupos de objetos de controle e de apresentação que compõem as caixas de diálogo e outros objetos compostos encontram-se alinhados vertical e horizontalmente?	X		

3	Os rótulos de campos organizados verticalmente e muito diferentes em tamanho estão justificados à direita?	X		
4	A largura mínima dos mostradores de texto é de 50 caracteres?			X
5	A altura mínima dos mostradores de texto é de 4 linhas?			X
6	Os parágrafos de texto são separados por, pelo menos, uma linha em branco?			X
7	O uso exclusivo de maiúsculas nos textos é evitado?	X		
8	O uso do negrito é minimizado?	X		
9	O uso do sublinhado é minimizado?	X		
10	Nas tabelas, linhas em branco são empregadas para separar grupos?			X
11	As listas de dados alfabéticos são justificadas à esquerda?	X		
12	As listas contendo números decimais apresentam alinhamento pela vírgula?		X	
13	As linhas empregadas para o enquadramento e segmentação de menus (separadores, delimitadores etc.) são simples?	X		
14	As bordas dos painéis dos menus estão suficientemente separadas dos textos das opções de modo a não prejudicar a sua legibilidade?	X		
15	O uso de abreviaturas é minimizado nos menus?	X		
16	Os nomes das opções estão somente com a inicial em maiúsculo?	X		
17	Os números que indicam as opções de menu estão alinhados pela direita?	X		
18	Quando a enumeração alfabética é utilizada, as letras para seleção estão alinhadas pela esquerda?	X		
19	As opções das barras de menu horizontal estão separadas por, no mínimo, 2 caracteres brancos?	X		
20	Os rótulos de campos começam com uma letra maiúscula, e as letras restantes são minúsculas?	X		
21	Os itens de dados longos são particionados em grupos mais curtos, tanto nas entradas como nas apresentações?			X

22	Os códigos alfanuméricos do sistema agrupam separadamente letras e números?			X
23	Os ícones são legíveis?	X		
24	O sistema utiliza rótulos (textuais) quando pode existir ambiguidade de ícones?	X		
25	A informação codificada com o vídeo reverso é legível?			X
26	O uso de vídeo reverso está restrito à indicação de feedback de seleção?	X		
27	Os dados a serem lidos são apresentados de forma contínua, ou seja, não piscantes?			X
Avaliação Feedback: Avalie a qualidade do feedback imediato às ações do usuário.				
1	O sistema fornece feedback para todas as ações do usuário?	X		
2	Quando, durante a entrada de dados, o sistema torna-se indisponível ao usuário, devido a algum processamento longo, o usuário é avisado desse estado do sistema e do tempo dessa indisponibilidade?			X
3	O sistema fornece informações sobre o estado das impressões?		X	
4	Os itens selecionados de uma lista são realçados visualmente de imediato?	X		
5	A imagem do cursor fornece feedback dinâmico e contextual sobre a manipulação direta?		X	
6	O sistema fornece ao usuário informações sobre o tempo de processamentos demorados?		X	
7	O sistema apresenta uma mensagem informando sobre o sucesso ou fracasso de um processamento demorado?	X		
8	O sistema fornece feedback imediato e contínuo das manipulações diretas?	X		
9	O sistema define o foco das ações para os objetos recém criados ou recém abertos?	X		

10	O sistema fornece feedback sobre as mudanças de atributos dos objetos?	X		
11	Qualquer mudança na situação atual de objetos de controle é apresentada visualmente de modo claro ao usuário?	X		
12	O sistema fornece um histórico dos comandos entrados pelo usuário durante uma sessão de trabalho?		X	
Avaliação Concisão: Verifique o tamanho dos códigos e termos apresentados e introduzidos no sistema.				
1	O sistema oferece valores defaults para acelerar a entrada de dados?	X		
2	A identificação alfanumérica das janelas é curta o suficiente para ser lembrada facilmente?	X		
3	Os nomes das opções de menu são concisos?	X		
4	Os ícones são econômicos sob o ponto de vista do espaço nas telas?	X		
5	As denominações são breves?	X		
6	As abreviaturas são curtas?	X		
7	Os códigos arbitrários que o usuário deve memorizar são sempre menores do que 4 ou 5 caracteres?			X
8	Os rótulos são concisos?	X		
9	Códigos alfanuméricos não significativos para o usuário e que devem ser entrados no sistema são menores do que 7 caracteres?			X
10	Na entrada de dados alfanuméricos, o sistema considera as letras maiúsculas e minúsculas como equivalentes?			X
11	Na entrada de dados numéricos, o usuário é liberado do preenchimento do ponto decimal desnecessário?	X		
12	Na entrada de dados numéricos, o usuário é liberado do preenchimento do zeros fracionários desnecessários?	X		
13	Na entrada de valores métricos ou financeiros, o usuário é liberado do preenchimento da unidade de medida?	X		

14	É permitido ao usuário reaproveitar os valores definidos para entradas anteriores, podendo inclusive alterá-los?	X		
Avaliação Ações Mínimas: Verifique a extensões dos diálogos estabelecidos para a realização dos objetivos do usuário.				
1	Em formulário de entrada de dados o sistema posiciona o cursor no começo do primeiro campo de entrada?	X		
2	Na realização das ações principais em uma caixa de diálogo, o usuário tem os movimentos de cursor minimizados através da adequada ordenação dos objetos?	X		
3	O usuário dispõe de um modo simples e rápido (tecla TAB por exemplo) para a navegação entre os campos de um formulário?	X		
4	Os grupos de botões de comando possuem sempre um botão definido como default?	X		
5	A estrutura dos menus é concebida de modo a diminuir os passos necessários para a seleção?	X		
Avaliação Densidade Informacional: Avalie a densidade informacional das telas apresentadas pelo sistema.				
1	A densidade informacional das janelas é reduzida?	X		
2	As telas apresentam somente os dados e informações necessários e indispensáveis para o usuário em sua tarefa?	X		
3	Na entrada de dados codificados, os códigos apresentam somente os dados necessários estão presentes na tela de uma maneira distinguível?			X
4	O sistema minimiza a necessidade do usuário lembrar dados exatos de uma tela para outra?	X		
5	Na leitura de uma janela, o usuário tem seus movimentos oculares minimizados através da distribuição dos objetos principais segundo as linhas de um "Z" ?	X		
6	O sistema evita apresentar um grande número de janelas que podem desconcentrar ou sobrecarregar a memória do usuário?	X		

7	Na manipulação dos dados apresentados pelo sistema, o usuário está liberado da tradução de unidades?			X
8	As listas de seleção e combinação apresentam uma altura correspondente a um máximo de nove linhas?	X		
9	Os painéis de menu apresentam como ativas somente as opções necessárias?			X
Avaliação Ações Explícitas: Verifique se é o usuário quem comanda explicitamente as ações do sistema.				
1	O sistema posterga os processamentos até que as ações de entrada do usuário tenham sido completadas?	X		
2	Durante a seleção de uma opção de menu o sistema permite a separação entre indicação e execuções da opção ?	X		
3	Para iniciar o processamento dos dados, o sistema sempre exige do usuário uma ação explícita de "ENTER"?	X		
4	É sempre o usuário quem comanda a navegação entre os campos de um formulário?	X		
Avaliação Controle do Usuário: Avalie as possibilidades do usuário controlar o encadeamento e a realização das ações.				
1	O usuário pode terminar um diálogo seqüencial repetitivo a qualquer instante?	X		
2	O usuário pode interromper e retomar um diálogo seqüencial a qualquer instante?	X		
3	O usuário pode reiniciar um diálogo seqüencial a qualquer instante?	X		
4	Durante os períodos de bloqueio dos dispositivos de entrada, o sistema fornece ao usuário uma opção para interromper o processo que causou o bloqueio?	X		
Avaliação Flexibilidade: Verifique se o sistema permite personalizar as apresentações e os diálogos.				
1	Os usuários têm a possibilidade de modificar ou eliminar itens irrelevantes das janelas?			X

2	Ao usuário é permitido personalizar o diálogo, através da definição de macros?		X	
3	É permitido ao usuário alterar e personalizar valores definidos por <i>default</i> ?	X		
Avaliação Experiência do Usuário: Avalie se usuários com diferentes níveis de experiência têm iguais possibilidades de obter sucesso em seus objetivos.				
1	Caso se trate de um sistema de grande público, ele oferece formas variadas de apresentar as mesmas informações aos diferentes tipos de usuário?		X	
2	Os estilos de diálogo são compatíveis com as habilidades do usuário, permitindo ações passo-a-passo para iniciantes e a entrada de comandos mais complexos por usuários experimentados?		X	
3	O usuário pode se deslocar de uma parte da estrutura de menu para outra rapidamente?	X		
4	O sistema oferece equivalentes de teclado para a seleção e execução das opções de menu, além do dispositivo de apontamento (mouse,...)?		X	
5	O sistema é capaz de reconhecer um conjunto de sinônimos para os termos básicos definidos na linguagem de comando, isto para se adaptar aos usuários novatos ou ocasionais?			X
6	O usuário experiente pode efetuar a digitação de vários comandos antes de uma confirmação?			X
Avaliação Proteção contra erros: Verifique se o sistema oferece as oportunidades para o usuário prevenir eventuais erros.				
1	O sistema apresenta uma separação adequada entre áreas selecionáveis de um painel de menu de modo a minimizar as ativações acidentais?	X		
2	Em toda ação destrutiva, os botões selecionados por default realizam a anulação dessa ação?	X		

3	Os campos numéricos para entrada de dados longos estão subdivididos em grupos menores e pontuados com espaços, vírgulas, hífen ou barras?			X
4	Ao final de uma sessão de trabalho o sistema informa sobre o risco de perda dos dados?	X		
5	O sistema emite sinais sonoros quando ocorrem problemas na entrada de dados?		X	
6	As teclas de funções perigosas encontram-se agrupadas e/ou separadas das demais no teclado?			X
7	O sistema solicita confirmação (dupla) de ações que podem gerar perdas de dados e/ou resultados catastróficos?		X	
Avaliação Mensagens de erro: Avalie a qualidade das mensagens de erro enviadas aos usuários em dificuldades.				
1	As mensagens de erro ajudam a resolver o problema do usuário, fornecendo com precisão o local e a causa específica ou provável do erro, bem como as ações que o usuário poderia realizar para corrigi-lo?	X		
2	As mensagens de erro são neutras e polidas?	X		
3	As frases das mensagens de erro são curtas e construídas a partir de palavras curtas, significativas e de uso comum?	X		
4	As mensagens de erro estão isentas de abreviaturas e/ou códigos gerados pelo sistema operacional?	X		
5	O usuário pode escolher o nível de detalhe das mensagens de erro em função de seu nível de conhecimento?		X	
6	A informação principal de uma mensagem de erro encontra-se logo no início da mensagem?	X		
7	Quando necessário, as informações que o usuário deve memorizar encontram-se localizadas na parte final da mensagem de erro?			X
8	Em situações normais as mensagens de erro são escritas em maiúsculo/minúsculo?	X		

9	As mensagens de erro têm seu conteúdo modificado quando na repetição imediata do mesmo erro pelo mesmo usuário?		X	
Avaliação Correção de erros: Verifique as facilidades oferecidas para que o usuário possa corrigir os erros cometidos.				
1	Qualquer ação do usuário pode ser revertida através da opção DESFAZER?		X	
2	Através da opção REFAZER, a regressão do diálogo, também pode ser desfeita?		X	
3	Os comandos para DESFAZER e REFAZER o diálogo estão diferenciados?		X	
4	O sistema reconhece e através de uma confirmação do usuário, executa os comandos mais freqüentes mesmo com erros de ortografia?			X
5	Depois de um erro de digitação de um comando ou de dados, o usuário tem a possibilidade de corrigir somente a parte dos dados ou do comando que está errada?			X
Avaliação Consistência: Avalie se é mantida uma coerência no projeto de códigos, telas e diálogos com o usuário.				
1	A identificação das caixas, telas ou janelas são únicas?	X		
2	A organização em termos da localização das várias características das janelas é mantida consistente de uma tela para outra?	X		
3	A posição inicial do cursor é mantida consistente ao longo de todas as apresentações de formulários?			X
4	Uma mesma tecla de função aciona a mesma opção de uma tela para outra?		X	
5	Os ícones são distintos uns dos outros e possuem sempre o mesmo significado de uma tela para outra?	X		
6	A localização dos dados é mantida consistente de uma tela para outra?	X		
7	Os formatos de apresentação dos dados são mantidos consistentes de uma tela para outra?	X		

8	Os rótulos estão na mesma posição em relação aos campos associados?	X		
9	O símbolo para convite à entrada de dados é padronizado (por exemplo " : ")?	X		
10	As áreas de entrada de comandos estão na mesma posição de uma tela para outra?	X		
11	Os significados dos códigos de cores são seguidos de maneira consistente?			X
Avaliação Significados: Avalie se os códigos e denominações são claros e significativos para os usuários do sistema.				
1	As denominações dos títulos estão de acordo com o que eles representam?	X		
2	Os títulos das páginas de menu são explicativos, refletindo a natureza da escolha a ser feita?	X		
3	Os títulos das páginas de menu são distintos entre si?	X		
4	Os títulos das páginas de menu são combináveis ou componíveis?		X	
5	As denominações das opções de menu são familiares ao usuário?	X		
6	O vocabulário utilizado nos rótulos, convites e mensagens de orientação são familiares ao usuário, evitando palavras difíceis?	X		
7	O vocabulário utilizado em rótulos, convites e mensagens de orientação é orientado à tarefa, utilizando termos e jargão técnico normalmente empregado na tarefa?	X		
8	Os cabeçalhos de colunas de dados são significativos e distintos?	X		
9	O sistema adota códigos significativos ou familiares aos usuários?			X
10	As abreviaturas são significativas?	X		
11	As abreviaturas são facilmente distinguíveis umas das outras, evitando confusões geradas por similaridade?	X		
12	A intermitência luminosa (pisca-pisca) é usada com moderação e somente para atrair a atenção para alarmes, avisos ou mensagens críticas?	X		

Avaliação Compatibilidade: Verifique a compatibilidade do sistema com as expectativas e necessidades do usuário em sua tarefa.				
1	As telas são compatíveis com o padrão do ambiente?	X		
2	A imagem do formulário na tela do terminal assemelha-se com o formulário de entrada em papel?			X
3	O sistema propõe uma caixa de diálogo modal, quando a aplicação deve ter todos os dados antes de prosseguir ou quando o usuário tenha de responder a uma questão urgente?	X		
4	As caixas de diálogo do sistema apresentam um botão de validação, um botão de anulação e, se possível, um botão de ajuda?	X		
5	Os significados usuais das cores são respeitados nos códigos de cores definidos?			X
6	As opções de codificação por cores são limitadas em número?			X
7	As informações codificadas através das cores apresentam uma codificação adicional redundante?	X		
8	A taxa de intermitência para elementos piscantes está entre 2 e 5 Hz (2 a 5 piscadas por segundo)?			X
9	A apresentação sonora é compatível com o ruído do ambiente?			X
10	As mensagens são sempre afirmativas e na voz ativa?	X		
11	Quando uma frase descreve uma seqüência de eventos, a ordem das palavras na frase corresponde à seqüência temporal dos eventos?	X		
12	Ilustrações e animações são usadas para completar as explicações do texto?	X		
13	O sistema segue as convenções dos usuários para dados padronizados?	X		
14	O sistema utiliza unidades de medida familiares ao usuário?	X		
15	Dados numéricos que se alterem rapidamente são apresentados analogicamente?		X	

16	Dados numéricos que demandam precisão de leitura são apresentados digitalmente?	X		
17	Os itens são numerados com números, não com letras?	X		
18	Os identificadores numéricos de opção de menu iniciam de "1", e não de "0"?	X		
19	Os eixos de um gráfico apresentam escalas numéricas iniciando em zero, com intervalos padronizados, crescendo da esquerda para a direita e de cima para baixo?		X	
20	Os itens de um grupo de botões de rádio são mutuamente exclusivos?	X		
21	Os itens de um grupo de caixas de atribuição permitem escolhas independentes?	X		

Fonte: Questões extraídas de <http://www.labiutil.inf.ufsc.br/ergolist/quest.htm>

ANEXO C: Projetos Fuzzy implementados

As tabelas a seguir apresentam resumos de projetos apresentados em teses, dissertações, trabalhos de conclusão de curso e artigos. A organização dessas tabelas permite a extração de estatísticas importantes para este projeto, como:

- a) Aplicações da lógica difusa, tipos de projetos e áreas de atuação.
- b) Ferramentas utilizadas para implementar os modelos difusos.
- c) Funções de pertinência.
- d) Máquina de inferência.
- e) Operadores.
- f) Implicação.
- g) Agregação.
- h) Defuzificação.

Tabela 40- Diagnóstico automático de rolamentos baseado em lógica fuzzy

Índice	A1
Título:	Diagnóstico automático de rolamentos baseado em lógica fuzzy
Autor(es):	Rodrigo Yoshiaki Fujimoto
Orientador:	Prof. Dr. Linilson R. Padovese
Instituição:	Escola politécnica da universidade de São Paulo
Cidade:	São Paulo
Ano:	2005
Titulação:	Mestre em Engenharia
Referência	FUJIMOTO, Rodrigo Yoshiaki. Diagnóstico Automático de Defeitos em Rolamentos Baseados em Lógica Fuzzy . 2005. 158 f. Dissertação (Mestrado) - Departamento de Engenharia Mecânica, Universidade de São Paulo Escola Politécnica, São Paulo, 2005.

Resumo:	O projeto tem por finalidade diagnosticar defeitos em mancais de rolamentos industriais. Os dados para análise foram obtidos por um acelerômetro. Através da vibração do equipamento é possível obter assinaturas, e conseqüentemente verificar defeitos. Para facilitar a manutenção preditiva em indústrias o projeto propõe o estudo de duas metodologias: Inferência fuzzy e Fuzzy C-Means e ao final faz a comparação com o método atual mais empregado: Análise estatística.
Resultado	Após estudar e aplicar os métodos, os seguintes resultados foram obtidos: 1) Método estatístico: Detecção = 76% 2) Inferência fuzzy = Detecção = 97,3% e classificação = 94,7% 3) Fuzzy C-Means = Detecção = 90,7% e classificação = 51,1%
Funções	Gaussiana, Triangular, trapézio
Operadores	AND (Mínimo) e OR (Máximo)
Inferência	Mamdani
Implicação	Mínimo
Agregação	Máximo
Defuzificação	Centro Gravidade
Software:	Desenvolveu o <i>software</i>

Tabela 41- Controle fuzzy para a melhoria do perfil de tensão em sistema de distribuição de energia elétrica

Índice	A2
Título:	Controle fuzzy para a melhoria do perfil de tensão em sistema de distribuição de energia elétrica
Autor(es):	Wesley Fernando Usida
Orientador:	Prof. Denis Vinicius Coury
Instituição:	Escola de engenharia de São Carlos, da universidade de São Paulo
Cidade:	São Carlos
Ano:	2007
Titulação:	Mestre em Engenharia Elétrica
Referência	USIDA, Wesley Fernando. Controle fuzzy para melhoria do perfil de tensão em sistemas de distribuição de energia elétrica . 2007. 143 f. Dissertação (Mestrado) - Curso de Engenharia Elétrica, Escola de Engenharia de São Carlos da Universidades de São Paulo, São Carlos, 2007.
Resumo:	O projeto tem por finalidade a implementação de um relé regulador de tensão

	baseado em sistemas <i>fuzzy</i> que atua no comutador de <i>tap</i> de transformadores de potência instalados em subestações de distribuição. A implementação do algoritmo foi realizada no Matlab [®] , e após simulações o sistema se mostrou viável.
Resultado	O Relé <i>fuzzy</i> desenvolvido permitiu adaptação automática as variações que ocorrem na demanda do sistema elétrico. Com isso, foi evitado o fornecimento acima ou abaixo dos limites permitidos aos consumidores finais.
Funções	Triangular, trapézio
Operadores	AND (Mínimo) e OR (Máximo)
Software:	Matlab [®]
Observações	As rotinas implementadas no Simulink [®] / Matlab [®] utilizam muito processamento, tendo como consequência um tempo de processamento elevado.

Tabela 42- Implementação de um sistema de controle inteligente utilizando a lógica fuzzy

Índice	A3
Título:	Implementação de um sistema de controle inteligente utilizando a lógica fuzzy
Autor(es):	Marcelo Bilobrovec, Rui Francisco Martins Marçal, João Luis Kovaleski
Evento:	XI SIMPEP
Cidade:	Bauru/SP
Ano:	2004
Referência	BILOBROVEC, Marcelo; MARÇAL, Rui Francisco Martins; KOVALESKI, João Luis. Implementação de um sistema de controle inteligente utilizando a lógica fuzzy. In: SIMPEP, 11., 2004, Bauru. Anais... . Bauru: Simpep, 2004. p. 1 - 7
Resumo:	Apresentação de um sistema especialista desenvolvido para modelagem computacional da previsão do comportamento de um motor elétrico em função da tensão a ele aplicada. Os dados da pesquisa foram obtidos junto ao Laboratório de Informática Aplicada à Agricultura - INFOAGRO, da Universidade Estadual de Ponta Grossa (UEPG) e foram utilizados dados de teste de bancada de um motor elétrico, obtidos junto ao laboratório de

	Engenharia Elétrica - UNESP, Bauru. A partir dos resultados obtidos, pode-se dizer com segurança que o Controlador Fuzzy é aplicável a sistemas que ainda não dispõem de uma metodologia adequada ao seu controle.
Resultado	Os resultados obtidos para o caso do motor elétrico mostram a viabilidade da aplicação da lógica fuzzy no controle da velocidade de motores. Isto permite vislumbrar a possibilidade da criação de controladores de baixo custo, graças à simplicidade de sua implantação, permitindo, desta forma, uma fácil integração a sistemas já existentes, pois através da simulação chegou-se ao valor de 81,49 V para uma velocidade de 105,5330 rad/s, valor este muito próximo ao tabelado (quadro 4) de 81,60 V para a mesma velocidade.
Funções	Trapézio e Triangular
Operadores	and
Inferência	Mamdani
Implicação	Mínimo
Agregação	Máximo
Defuzificação	Centro da gravidade
Software:	Próprio, desenvolvido em Delphi 7

Tabela 43- Desenvolvimento de um algoritmo de controle para sistemas de elevadores, com base na lógica fuzzy

Índice	A4
Título:	Desenvolvimento de um algoritmo de controle para sistemas de elevadores, com base na Lógica Fuzzy
Autor(es):	Lívio Borges Requião Montalvão e Newton Cleber Martins de Carvalho
Resumo:	Este trabalho apresenta um algoritmo de controle para sistemas de elevadores com base na lógica fuzzy, que tem o objetivo de otimizar o consumo de energia e reduzir o tempo de espera dos usuários. Inicialmente, são apresentadas as etapas de desenvolvimento do algoritmo Fuzzy. Depois, se apresenta a metodologia de testes, compreendendo a criação de um ambiente de simulação onde o algoritmo proposto foi comparado com outros dois, sendo um fuzzy e outro convencional. Os testes comprovam a eficiência do algoritmo proposto, que proporciona, em média, 37% de economia de energia e 3% de redução do tempo de espera em relação

	a um algoritmo convencional.
Resultado	Por meio de simulações foi possível comprovar a eficiência dos sistemas de controles Fuzzy em aplicações dessa natureza. Em termos práticos, a economia financeira proporcionada pela implantação do algoritmo de controle em um condomínio residencial, além de recuperar em curto prazo as despesas de instalação, poderia permitir a aplicação do montante financeiro economizado em obras para a melhoria do edifício, ou mesmo a redução das taxas de condomínio cobradas aos moradores.
Funções	Triangular
Operadores	AND
Inferência	Mamdani
Implicação	Mínima
Agregação	Máxima
Defuzificação	Centro da gravidade
Software:	Matlab [®] nas etapas de projeto

Fonte: Tabela elaborada pelo autor com base em informações recolhidas da obra de Lívio Borges Requião Montalvão e Newton Cleber Martins de Carvalho

Tabela 44- *Optimal Control And Fuzzy Control Application In Nonlinear Vehicular Suspension Models With Magnetorheological Damper*

Índice	A5
Título:	<i>Optimal Control And Fuzzy Control Application In Nonlinear Vehicular Suspension Models With Magnetorheological Damper</i>
Autor(es):	Angelo Marcelo Tusset, Alberto Tamagna, Marat Rafikov
Evento:	XIII International Conference on Industrial Engineering and Operations Management
Cidade:	Foz do Iguaçu/PR
Ano:	2007
Referência	TUSSET, Angelo Marcelo; TAMAGNA, Alberto; RAFIKOV, Marat. Optimal Control And Fuzzy Control Application In Nonlinear Vehicular Suspension Models With Magnetorheological Damper. In: INTERNATIONAL CONFERENCE ON INDUSTRIAL ENGINEERING AND OPERATIONS MANAGEMENT, 13., 2007, Foz do Iguaçu. Anais... . Foz Iguaçu: Abepro, 2007. p. 1 - 12.
Resumo:	<i>In this work we study the using of the magnetorheological (MR)</i>

	<i>damper, controlled by the combination of linear feedback control and fuzzy logic. The linear feedback control determines the force fulfilled by the damper and the fuzzy controller determines the amps that will be used in the rheological material. The computer simulations show the efficiency of the proposed vibrations control in a nonlinear quarter-car vehicular model.</i>
Resultado	<i>In this paper we considered the magnetorheological damper, controlled by the combination of linear feedback control and fuzzy logic. The linear feedback control determines the force fulfilled by the damper and the fuzzy controller determines the amps that will be used in the rheological material. Numerical simulations of this control applied to the magnetorheological damper show the efficiency of the proposed vibrations control in a nonlinear quarter-car vehicular model.</i>
Funções	Triangular
Operadores	AND
Inferência	Mamdani
Implicação	Mínimo
Agregação	Máximo
Defuzificação	Centro Gravidade
Software:	Matlab®

Fonte: Tabela elaborada pelo autor com base em informações recolhidas da obra de Angelo Marcelo Tuset, Alberto Tamagna, Marat Rafikov (2007)

Tabela 45- Aplicação de sistemas fuzzy e neuro-fuzzy para a predição da temperatura retal de frangos de corte

Índice	A6
Título:	Aplicação de sistemas fuzzy e neuro-fuzzy para a predição da temperatura retal de frangos de corte
Autor(es):	Leandro Ferreira
Orientador:	Orientador: Dr. Tadayuki Yanagi Junior Co-Orientador: Dr. Wilian Soares Lacerda
Instituição:	Universidade Federal de Lavras
Cidade:	Lavras/MG
Ano:	2009
Titulação:	Mestre em engenharia de Sistemas
Referência	FERREIRA, Leandro. Aplicação de sistemas fuzzy e neuro-fuzzy para a predição da temperatura retal de frangos de corte. 2009. 56 f. Dissertação

	(3) - Curso de Pós-graduação em Engenharia de Sistemas, Universidade Federal de Lavras, Lavras, 2009.
Resumo:	Objetivo do projeto foi de desenvolver e validar um sistema de apoio a decisão, utilizando sistema fuzzy e um sistema neuro-fuzzy, para a predição da temperatura retal de frangos de corte. Foram utilizadas as seguintes variáveis de entrada: temperatura do ar, umidade relativa e velocidade do ar e tendo como variável de saída a temperatura retal.
Resultado	O sistema fuzzy desenvolvido para a predição da temperatura retal de frangos de corte, com base no ambiente térmico de criação, caracterizado pela temperatura do ar, umidade relativa e velocidade do ar, proporcionou um baixo desvio padrão, sendo esse de aproximadamente 0,13 °C.
Funções	Triangular
Operadores	AND(Mínimo)
Inferência	Mamdani
Implicação	Mínimo
Agregação	Máximo
Defuzificação	Centro da gravidade
Software:	Maple 8 para aplicação fuzzy e Scilab para neuro-fuzzy

Fonte: Tabela elaborada pelo autor com base em informações recolhidas da obra de Leandro Ferreira (2009)

Tabela 46- Sistema especialista em lógica fuzzy para o controle, gerenciamento e manutenção da qualidade

Índice	A7
Título:	Sistema Especialista em Lógica Fuzzy para o Controle, Gerenciamento e Manutenção da Qualidade
Autor(es):	Marcelo Bilobrovec
Orientador:	Rui Francisco Martins Marçal
Instituição:	Universidade Tecnológica Federal do Paraná
Cidade:	Ponta Grossa/PR
Ano:	2005
Titulação:	Mestre em Engenharia de Produção
Referência	BILOBROVEC, Marcelo. Sistema Especialista em Lógica Fuzzy Para o Controle, Gerenciamento e Manutenção da Qualidade em Processo de Aeração de Grãos . 2005. 74 f. Dissertação (3) - Curso de Engenharia de Produção, Departamento de Pós-graduação, Universidade Tecnológica

	Federal do Paraná, Ponta Grossa, 2005.
Resumo:	Sistema especialista baseado em Lógica Fuzzy para controle automático, gerenciamento e manutenção da qualidade do processo de aeração de grãos em silos. O sistema possibilita a aeração de grãos de forma racional em silos verticais atendendo às necessidade tanto para armazenamento como para a conservação destes. Foi desenvolvido um protótipo que permitiu ensaios com grãos de milho e a verificação da necessidade de aerar certa massa destes grãos, através da análise das condições de temperatura ambiente e da massa ensilada e da umidade relativa do ar ambiente e dos grãos. As variáveis de entrada são capturadas através de sensores de temperatura e umidade, essas são processadas através de um sistema <i>fuzzy</i> , onde as saídas fazem o controle da aeração.
Resultado	O sistema desenvolvido em Delphi 7 permitiu o controle da aeração de grãos baseado nas informações de umidade e temperatura, tanto em modo automático como manual.
Funções	Trapézio, Triangular
Operadores	AND
Inferência	Mamdani
Implicação	Mínimo
Agregação	Máximo
Defuzificação	Média, Máximo, Mínimo
Software:	Desenvolvido em Delphi 7

Fonte: Tabela elaborada pelo autor com base em informações recolhidas da obra de Marcelo Bilobrovec (2005)

ANEXO D: InFuzzy UDP Tool

Para facilitar o desenvolvimento e demonstrar o protocolo, foi desenvolvida a ferramenta InFuzzy UDP Tool. Através desse *software* é possível realizar conexão com o servidor permitindo realizar a chamada de todas as funções presentes no protocolo.

Para que o sistema funcione corretamente é necessário verificar se a porta de comunicação está aberta na ferramenta InFuzzy. A abertura de uma porta ocorre na tela de simulação no canto superior direito. Para realizar a tarefa basta digitar o número de uma porta entre 1024 até 65000 e clicar no botão “Abrir Porta UDP”. É importante observar que algumas portas já podem estar em uso por outros serviços como *Skype*, *Torrent*, vídeo conferência. Se tudo ocorrer bem, aparecerá o teste “Fechar Porta UDP” no botão. Caso seja clicado novamente, a porta será fechada e não receberá mais mensagens de clientes. A Figura 73 apresenta o processo de abertura de porta UDP na ferramenta InFuzzy.

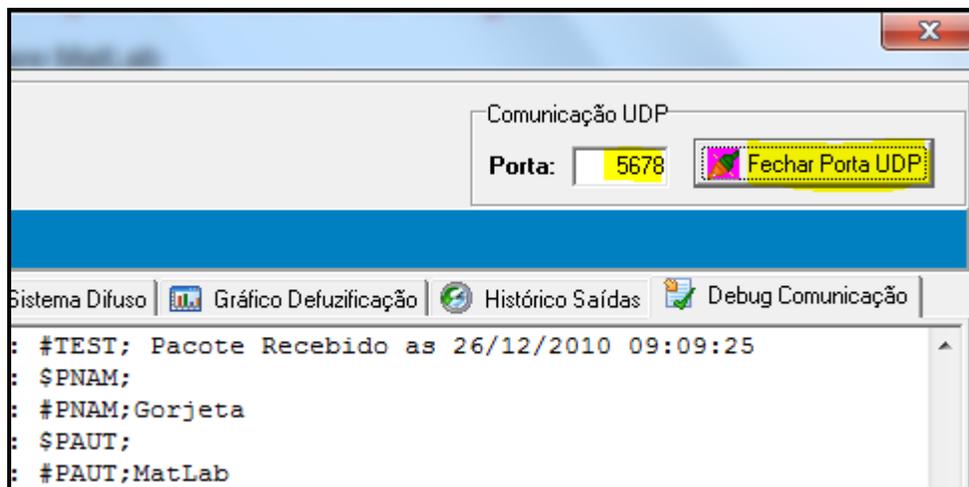


Figura 73- Tela de simulação da ferramenta InFuzzy

Fonte: Imagem extraída da ferramenta InFuzzy

A próxima etapa é executar a ferramenta InFuzzy UDP Tool que se encontra na pasta da instalação e no menu do sistema operacional. A Figura 74 está dividida em seis etapas, sendo elas:

1. Definição do endereço IP e Porta. É importante que a porta seja a mesma que foi definida na ferramenta InFuzzy. Ao clicar no botão “Teste de conexão com o servidor”, será realizada a chamada do protocolo “\$TEST”, se tudo ocorrer de forma correta, a circunferência em vermelho deverá ficar verde.
2. A etapa dois coleta do servidor as informações referentes ao projeto. Os seguintes comandos são ativados ao se clicar no botão “Capturar Informações do Servidor”: “\$PNAM;”, “\$PAUT;”, “\$PDAT;” e por último “\$PDES;”. Caso tudo ocorra de forma correta, todos os campos serão preenchidos.
3. A terceira etapa coleta informações referentes as variáveis de entrada. Para realizar essa tarefa o comando “\$ENTR;” é executado. O retorno desse comando traz as variáveis de entrada do projeto e os respectivos valores de início e fim do universo.
4. A quarta etapa coleta informações referentes as variáveis de saída. Para realizar essa tarefa o comando “\$SAID;” é executado. O retorno desse comando traz as variáveis de saída do projeto e os respectivos valores de início e fim do universo.
5. Etapa destinada ao envio de comandos do tipo “\$EXEC;” e “\$SAVE;”. A primeira parte é a escolha do comando na caixa de seleção e a segunda é a escrita das variáveis de entrada e os respectivos valores. Para enviar o comando ao servidor é necessário clicar no botão “Enviar Comando” após a instrução estar preenchida.
6. O espaço reservado na sexta etapa é destinada a escrever o retorno recebido do servidor após o envio do comando “\$EXEC;” e “\$SAVE;”.

É importante lembrar que não existe uma ordem nem a necessidade de executar todos os comandos para iniciar o envio de requisições de execução.

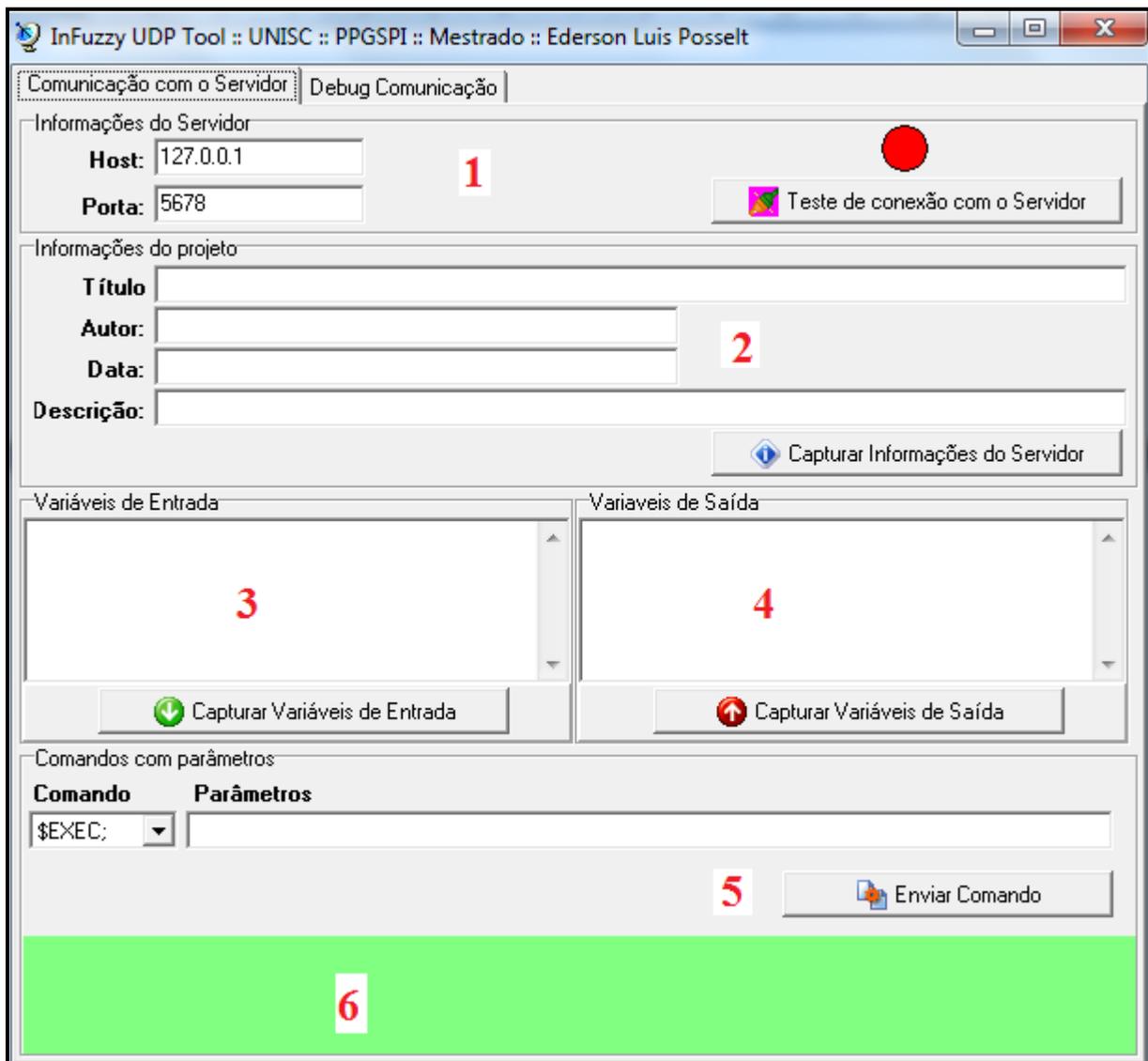


Figura 74- Ferramenta InFuzzy UDP Tool

Fonte: Imagem obtida da ferramenta InFuzzy UDP Tool

A Figura 75 apresenta a tela da ferramenta InFuzzy UDP Tool após a execução de todos os comandos do protocolo. Observando o projeto é possível perceber que se trata do clássico exemplo da “Gorjeta”.

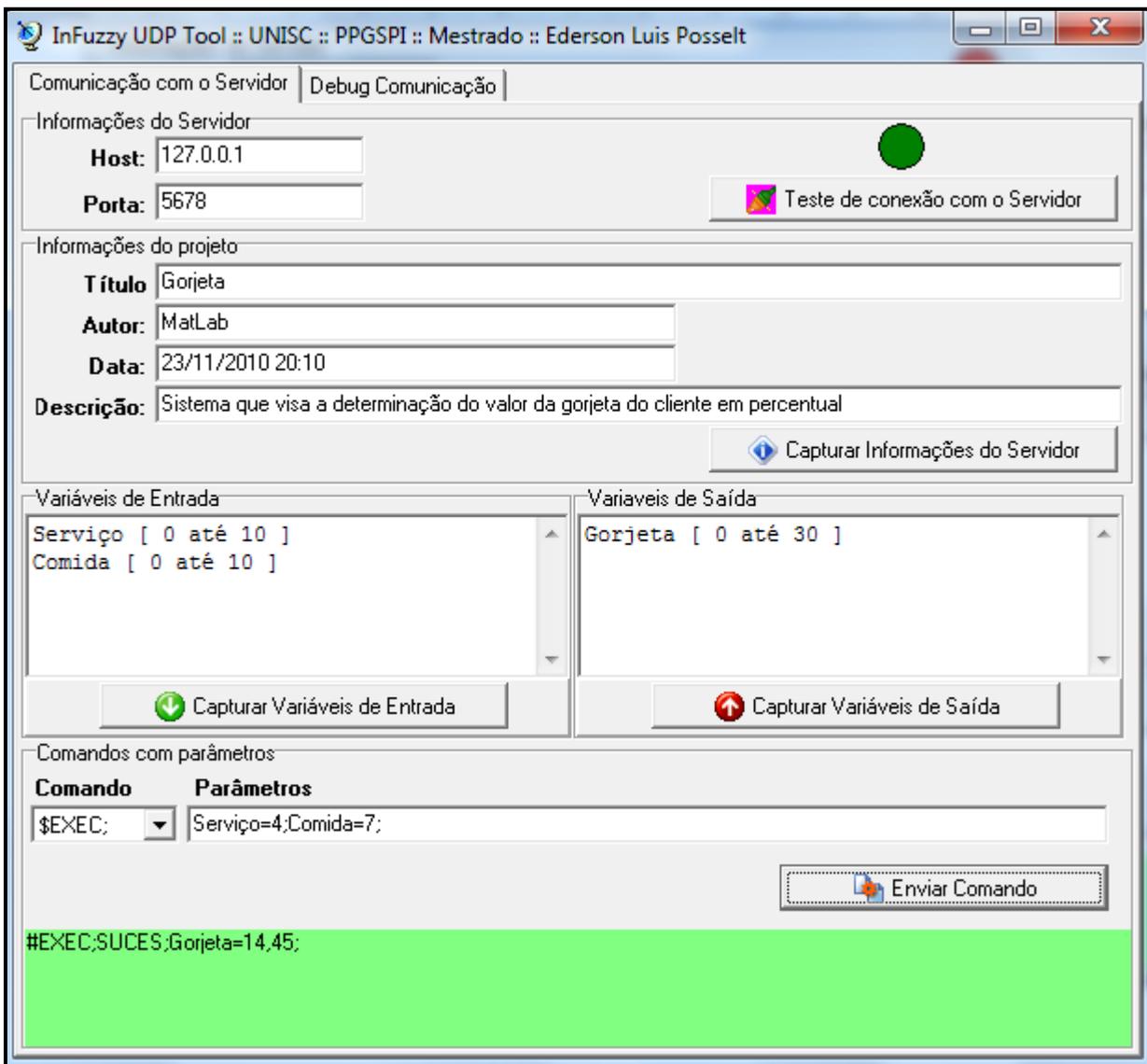
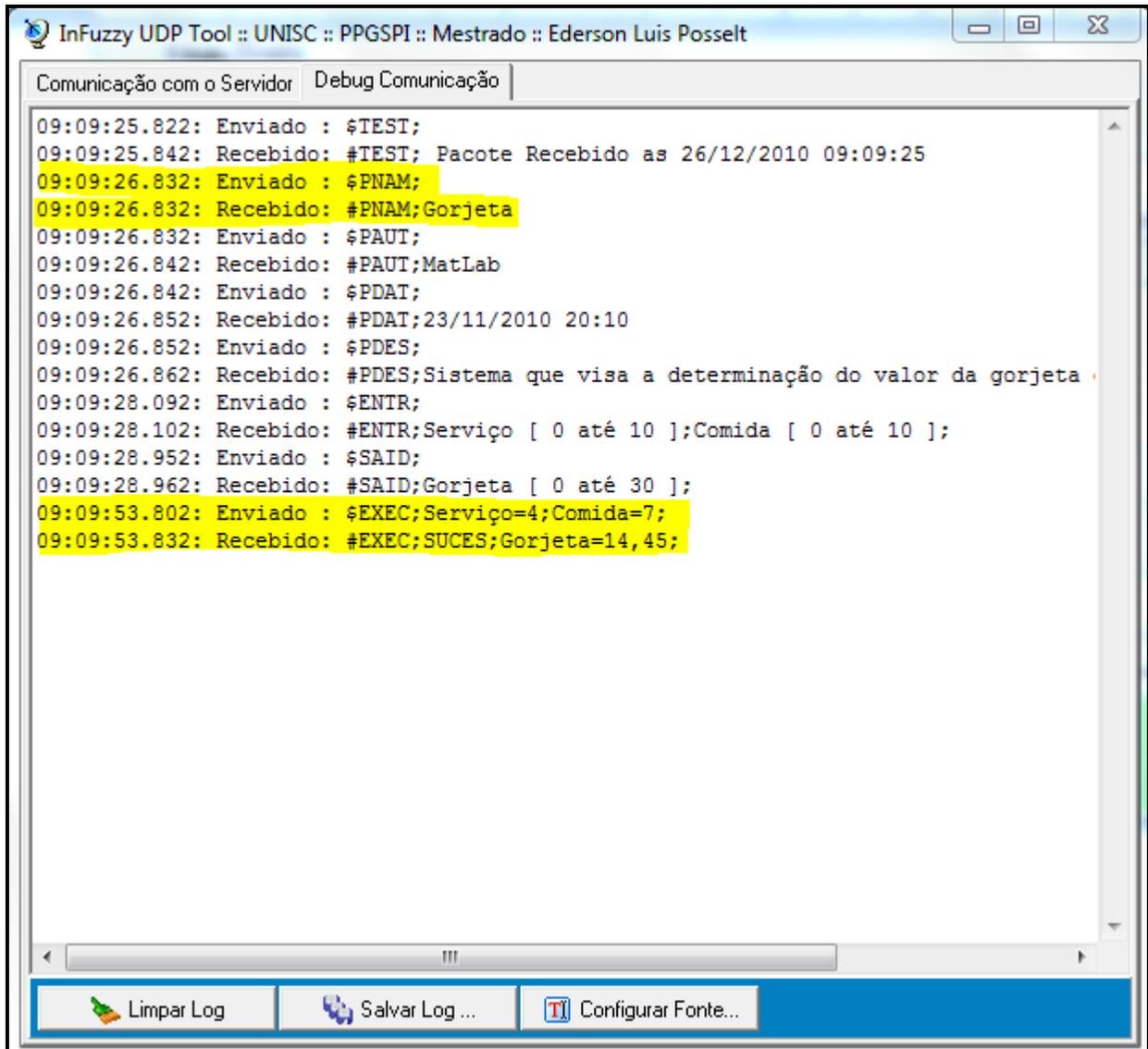


Figura 75- Ferramenta InFuzzy UDP Tool com os valores preenchidos

Fonte: imagem extraída da ferramenta InFuzzy UDP Tool

Todo o processo da execução dos comandos do protocolo gerado a partir dos acionamentos dos botões da primeira aba da ferramenta InFuzzy UDP Tool geraram um registro de mensagens que foi gravado na segunda aba do sistema (*Debug Comunicação*). O resultado desse registro pode ser visualizado na Figura 76, onde fica transparente o momento em que foi enviado a mensagem, qual foi a mensagem, o momento em que foi recebido a resposta e a resposta.

Observando as duas últimas instruções, é possível observar o envio do comando “\$EXEC;”, é importante observar que o comando foi realizado na máquina local, porém o tempo total entre o envio da mensagem, o processamento pela ferramenta InFuzzy e o recebimento da resposta foi de apenas trinta milissegundos, porém esse tempo pode ser alto para algumas aplicações específicas.



```
09:09:25.822: Enviado : $TEST;
09:09:25.842: Recebido: #TEST; Pacote Recebido as 26/12/2010 09:09:25
09:09:26.832: Enviado : $PNAM;
09:09:26.832: Recebido: #PNAM;Gorjeta
09:09:26.832: Enviado : $PAUT;
09:09:26.842: Recebido: #PAUT;MatLab
09:09:26.842: Enviado : $PDAT;
09:09:26.852: Recebido: #PDAT;23/11/2010 20:10
09:09:26.852: Enviado : $PDES;
09:09:26.862: Recebido: #PDES;Sistema que visa a determinação do valor da gorjeta
09:09:28.092: Enviado : $ENTR;
09:09:28.102: Recebido: #ENTR;Serviço [ 0 até 10 ];Comida [ 0 até 10 ];
09:09:28.952: Enviado : $SAID;
09:09:28.962: Recebido: #SAID;Gorjeta [ 0 até 30 ];
09:09:53.802: Enviado : $EXEC;Serviço=4;Comida=7;
09:09:53.832: Recebido: #EXEC;SUCES;Gorjeta=14,45;
```

Figura 76- Registro do envio de mensagens da ferramenta InFuzzy UDP Tool

Fonte: Imagem extraída da ferramenta InFuzzy UDP Tool