



Universidade Federal
do Rio de Janeiro



Universidade Federal do Rio de Janeiro
Programa de Pós-Graduação em Informática

Diogo da Silva Magalhães Gomes

NÉBULA: UM SISTEMA INTEGRADO PARA GERAÇÃO DE INFERÊNCIAS NEBULOSAS

Rio de Janeiro
2011

Diogo da Silva Magalhães Gomes

Nébula: um Sistema Integrado para Geração de Inferências Nebulosas

Dissertação de Mestrado apresentada ao Programa de Pós-Graduação em Informática (PPGI), Universidade Federal do Rio de Janeiro, como parte dos requisitos necessários à obtenção do título de Mestre em Informática

Orientadores: Cláudia Lage Rebello da Motta, DSc
Adriano Joaquim de Oliveira Cruz, PhD

Rio de Janeiro
2011

G633 Gomes, Diogo da Silva Magalhães

Nébula: um sistema integrado para geração de inferências nebulosas / Diogo da Silva Magalhães.—2011.
144 f.: Il.

Dissertação (Mestrado em Informática) – Universidade Federal do Rio de Janeiro, Instituto de Matemática, Núcleo de Computação Eletrônica, Rio de Janeiro, 2011.

Orientadores: Claudia Lage Rebello da Motta; Adriano Joaquim de Oliveira Cruz

1. Nébula. 2. Sistemas Integrado. 3. Geração de Inferências Nebulosas – Teses.
I. Claudia Lage Rebello da Motta (Orient.). II. Adriano Joaquim de Oliveira Cruz (Orient.). III. Título.

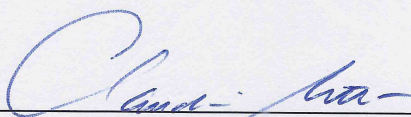
CDD

Diogo da Silva Magalhães Gomes

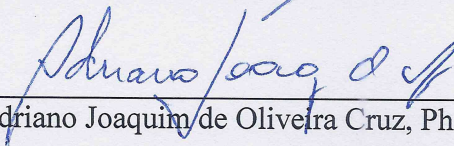
Nébula: um Sistema Integrado para Geração de Inferências Nebulosas

Dissertação submetida ao corpo docente do Programa de Pós-Graduação em Informática, Instituto de Matemática, Núcleo de Computação Eletrônica da Universidade Federal do Rio de Janeiro, como parte dos requisitos necessários para obtenção do grau de Mestre em Informática

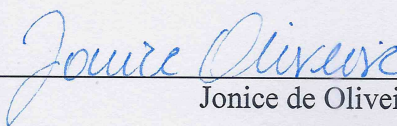
Aprovada em: Rio de Janeiro, 05 de agosto de 2011



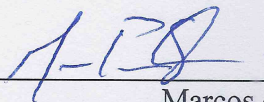
Prof. Claudia Lage Rebello da Motta, D.Sc., PPGI/UFRJ (orientadora)



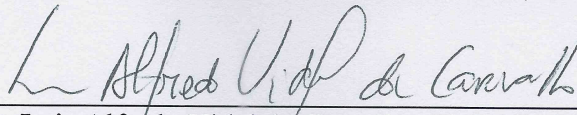
Prof. Adriano Joaquim de Oliveira Cruz, Ph.D, PPGI/UFRJ (Orientador)



Jonice de Oliveira Sampaio, D.Sc, PPGI/UFRJ



Marcos da Fonseca Elia, Ph.D, PPGI/UFRJ



Luis Alfredo Vidal de Carvalho, Ph.D, COPPE/UFRJ

*Para minha filha Larissa, minha grande força motivadora, encantadora fonte de
inocência e alegria que ilumina meus dias.*

Aos meus pais, meus grandes mestres, ídolos e verdadeiros amigos.

*À minha esposa Roberta, minha melhor amiga desde sempre, por seu positivismo e
paciência nos momentos adversos, encorajando-me a seguir adiante.*

Agradecimentos

Iniciar um projeto de pesquisa de mestrado é empreender uma longa jornada, que exige um grande esforço de dedicação e auto-aprendizado. No entanto, como todo projeto de longo prazo, deparamo-nos com alguns obstáculos no decorrer desta caminhada. E talvez seja justamente nestes momentos onde resida a verdadeira essência da descoberta, tornando claro o mais relevante aprendizado desta experiência, muito além dos aspectos técnicos e acadêmicos: a importância das pessoas, da amizade, seu companheirismo e seu apoio, que se manifestam nos momentos adversos.

Portanto, em primeiro lugar, agradeço especialmente à minha família, alicerce sobre a qual se apóia cada simples traço de minha personalidade, e essência de toda minha motivação.

A meus pais Luis e Leonor, verdadeiros mestres que me lecionaram por seus exemplos as lições mais importantes que eu poderia receber, meus referenciais de caráter, honestidade, paciência e amor, cuja abnegada dedicação fez de seus filhos seus grandes admiradores.

Agradeço especialmente à minha esposa Roberta, pelo privilégio de compartilhar de sua companhia, sua paciência e seu apoio. Suas palavras de motivação nos momentos mais adversos foram indispensáveis para a conclusão desta jornada.

À minha filha Larissa, por sua inocente compreensão nos pequenos momentos em que minha ausência se fez necessária, e por suas emocionantes gargalhadas em nossas brincadeiras. Qualquer ausência momentânea seria imediatamente recompensada por seus abraços e seus sorrisos.

A meus irmãos Danilo e Amanda, por seus exemplos de personalidade, inteligência, companheirismo e amizade, quem muito admiro e me espelho diariamente.

À minha avó Alexandrina, por seu extraordinário exemplo de abnegada dedicação, amor, paciência e compreensão a cada um de nossa grande família, especialmente acolhidos por seu coração imenso.

Acima de tudo, agradeço a Deus, pela oportunidade inicialmente concedida ao me permitir ingressar no mestrado, e especialmente por permitir-me prosseguir, mesmo perante as adversidades, rumo a esta realização.

Aos meus professores orientadores, Cláudia Motta e Adriano Cruz, por suas inestimáveis contribuições, opiniões perspicazes, paciência e bom humor, durante as aulas e reuniões de orientação, além de sua confiança atribuída ao trabalho no decorrer do projeto.

Ao meus grandes amigos desde sempre, Michele e Marcus Pereira, Suelen e Bruno Alpino, Camile e Bruno Santoro, Aline e Dário Minto, Nathália e Fred Diniz, por me permitirem crescer juntos, convivendo desde minha infância, e compartilhando de suas amizades.

Especialmente, agradeço também a meus amigos e companheiros de equipe da Petrobras, que colaboram diariamente, com muito bom humor e amizade, para uma rotina de trabalho produtiva e descontraída. O incentivo e a participação de cada um, e sua compreensão durante minha dedicação aos estudos, foram fundamentais para a realização deste trabalho. Meus agradecimentos ao Alexandre Lins, Anderson de Souza, André Moraes, Andréia Carvalho, Dalila Muniz, Jorge Morgado, Leonardo Bittencourt, Leonardo Tadeu, Luis Cogliatti, Max de Castro, Peter Santos, Renato Fishe, Roberto Cristoph e Sabrina Bettini. Em especial, agradeço ao amigo André Moraes, por sua parceria e amizade no cumprimento das tarefas acadêmicas.

Por fim, agradeço aos meus gerentes na Petrobras, Júlia Campello, Luis Antônio Araújo, e meu coordenador Flávio Gondin, pelo incentivo e liberação das horas necessárias para dedicação a este trabalho.

Resumo

GOMES, Diogo da Silva Magalhães. **Nébulas**: um sistema integrado para geração de inferências nebulosas. 2011. 144 f. Dissertação (Mestrado em Informática) – Programa de Pós-Graduação em Informática, Núcleo de Computação Eletrônica, Universidade Federal do Rio de Janeiro, Rio de Janeiro, 2011.

Jogos eletrônicos têm sido aplicados nas mais diversas áreas das ciências, utilizando-se de recursos computacionais inteligentes para agregar conhecimento às informações processadas. Neste cenário, destaca-se a utilização de jogos psicopedagógicos virtualizados em sistemas informatizados, onde algoritmos de inteligência computacional são aplicados dentro de processos pedagógicos para análise, intervenção e reabilitação cognitiva. No entanto, em função de sua natureza multidisciplinar e de suas características inerentemente imprecisas e inexatas, a modelagem de algoritmos matemáticos atuantes no processo de avaliação cognitiva é complexa. Este trabalho de pesquisa apresenta uma plataforma integrada para permitir o desenvolvimento de sistemas de inferências nebulosas de maneira simplificada e desacoplada da construção dos jogos psicopedagógicos, permitindo que as equipes multidisciplinares participem ativamente da construção da lógica para auxiliar a análise cognitiva, abstraindo, portanto, a complexidade das representações matemáticas e as limitações de algoritmos exatos para este tipo de problema.

Abstract

GOMES, Diogo da Silva Magalhães. **Nébulas**: um sistema integrado para geração de inferências nebulosas. 2011. 144 f. Dissertação (Mestrado em Informática) – Programa de Pós-Graduação em Informática, Núcleo de Computação Eletrônica, Universidade Federal do Rio de Janeiro, Rio de Janeiro, 2011.

Electronic games have been applied in several areas of science, using intelligent computational resources to aggregate knowledge to processed information. In this scenario, we highlight the use of psychopedagogical games, virtualized as computer systems, where computational intelligence algorithms are applied in educational processes for analysis, intervention and cognitive rehabilitation. However, due to its multidisciplinary nature and its uncertainties and inaccuracies, the modeling of these mathematical algorithms process is very complex. This research presents an integrated platform to enable the development of fuzzy inference systems in a simplified way, decoupled from the construction of the games, allowing multidisciplinary teams to participate actively in the construction of logic to assist the cognitive analysis, abstracting, however, the complexity of mathematical representations and the limitations of exact algorithms for this type of problem.

Lista de Figuras

Figura 1.1: Processo tradicional para construção de algoritmos nos jogos psicopedagógicos	20
Figura 1.2: Relevância do Nébulas, no contexto do projeto de virtualização de jogos psicopedagógicos	22
Figura 1.3: Diagrama esquemático da organização da dissertação	25
Figura 2.1: Fases componentes da metodologia de Fio Condutor. Fonte: Marques (2009b) ..	32
Figura 2.2: Imagens dos jogos virtualizados: Roda da Linguagem, CriaConto e Jogo dos Elásticos, respectivamente.....	35
Figura 3.1: Ilustração com representação conceitual da pertinência entre conjuntos clássicos e as infinitas possibilidades tratadas pela lógica nebulosa.....	38
Figura 3.2 Gráfico ilustrando conjunto clássico, representando a transição abrupta da pertinência de elementos entre os conjuntos	40
Figura 3.3 Gráfico representando a teoria dos conjuntos fuzzy, ilustrando o grau de pertinência de elementos entre os conjuntos	41
Figura 3.4: Exemplo de configuração possível para os conjuntos nebulosos por estatura	43
Figura 3.5: Exemplos de funções de transferência nos formatos triangular e trapezoidal, respectivamente	44
Figura 3.6: arquitetura geral de um sistema de inferências <i>fuzzy</i>	47
Figura 5.1: Modelo arquitetural em camadas para a construção de jogos neuropedagógicos.	62
Figura 5.2: Representação gráfica de uma possível partição fuzzy para a variável ‘quantidade de erros’	66
Figura 5.3: Diagrama do processo de construção de inferências com o sistema Nébulas	68
Figura 5.4: Modelo de arquitetura para o sistema Nébulas	70
Figura 5.5: Diagrama com os principais componentes do sistema Nébulas.....	71
Figura 5.6: Menu superior de seleção de módulos, com destaque para o Módulo de Configuração atualmente selecionado na barra de títulos.	73
Figura 5.7: Menu lateral de navegação e página principal da inferência criada.	74
Figura 5.8: Interface do sistema para o Passo 1: Seleção de Variáveis, do Módulo de Configuração	75
Figura 5.9: Editor de funções de pertinência do Nébulas.	76
Figura 5.10: Interface do sistema para o passo 2: Partições Nebulosas.....	77
Figura 5.11: Interface do sistema para o passo 3: Conjunto de Regras	78
Figura 5.12: Interface do sistema para o Módulo de Simulação	79
Figura 5.13: Diagrama do padrão de arquitetura <i>Model View Controller</i> (MVC)	82
Figura 6.1: Diagrama ilustrativo dos estudos de caso propostos.....	87
Figura 6.2: Esquema das variáveis envolvidas do jogo do Estacionamento do Caminhão	89
Figura 6.3: Funções de inclusão para as variáveis de entrada.....	90
Figura 6.4: Funções de inclusão para a variável de saída	91
Figura 6.5: Tela inicial do simulador de estacionamento de veículo.....	92
Figura 6.6: Tela do sistema Nébulas, com a instância criada para o simulador de estacionamento	93
Figura 6.7: Tela do sistema Nébulas, ilustrando as definições para as funções de pertinência.	95
Figura 6.8: Gráficos gerados pelo Nébulas para as funções de inclusão das variáveis de entrada	95
Figura 6.9: Gráficos gerados pelo Nébulas para as funções de inclusão da variável de saída ..	96
Figura 6.10: Tela do sistema Nébulas com a base de regras do simulador de estacionamento.	96
Figura 6.11: Rastros das trajetórias de diferentes simulações para o problema do estacionamento	98

Figura 6.12: Alteração do comportamento do veículo, conforme novas definições realizadas no sistema Nébula	99
Figura 6.13: Console de saída do simulador, representando os <i>logs</i> de intervalo de tempo para medição do desempenho do processamento da inferência	100
Figura 6.14: Toolbox do Matlab para definição das partições nebulosas	103
Figura 6.15: Toolbox Matlab para descrição da base de regras	104
Figura 6.16: Superfície representativa da matriz contendo o par de parâmetros de entrada e a variável de saída obtida pelo processamento da inferência	105
Figura 6.17: Console do Sistema Nébula para Simulação por Matriz de Valores	106
Figura 6.18: Mapa conceitual dos termos envolvidos na Função Cognitiva de Atenção	114
Figura 6.19: Representação das funções para cálculo dos coeficientes de avaliação de atenção	115
Figura 6.20: Modelo de tela e regras propostos para o Jogo “Monta Boneco”	116
Figura 6.21: Partições sugeridas para a variável ‘rapidez’	117
Figura 6.22: Partições sugeridas para a variável ‘precisão’	117
Figura 6.23: Partições sugeridas para a variável de saída ‘desempenho’	118
Figura 6.24: Jogo ‘Monta Boneco’ em execução, com destaque para a console de relatório	125

Lista de Tabelas

Tabela 3.1: conjunto de regras para um sistema de controle de velocidade de um veículo	45
Tabela 6.1: Partições Nebulosas identificadas para o sistema de inferências	90
Tabela 6.2: Base de regras para o sistema de inferências do estacionamento de veículo	91
Tabela 6.3: Comparação de resultados obtidos pela simulação de um algoritmo executada no sistema Nébula e no Matlab	109
Tabela 6.4: Estudo estatístico estratificado para o conjunto amostral, utilizando T-Student pareado	110
Tabela 6.5: Tabela de referência para a base de regras	118
Tabela 6.6: Matriz de referência das perguntas do questionário de avaliação com os critérios analisados	122
Tabela 6.7: Totalização de respostas obtidas para cada item do questionário	128

Lista de Equações

Equação 3.1: Função de pertinência de conjuntos clássicos	39
Equação 3.2: grau de pertinência de um elemento x a um conjunto X	40
Equação 3.3: função de pertinência de um conjunto X	40
Equação 6.1: Conjunto de equações que descrevem as funções de movimento do veículo	89
Equação 6.2: Equação para o cálculo do erro relativo percentual	106
Equação 6.3: Equação para o cálculo do Erro Médio Percentual Absoluto, calculado em função dos erros relativos percentuais em cada período, ou amostra	107

Lista de siglas e abreviaturas

API	Application Programming Interface ou Interface de Programação de Aplicativos
FIS	Fuzzy Inference System
HTTP	Hypertext Transfer Protocol
JAR	Java Archive
JEE	Java Enterprise Edition
JSF	Java Server Faces
MVC	Model-View-Controller
iNCE	Instituto Tércio Pacitti de Aplicações e Pesquisas Computacionais (Núcleo de Computação Eletrônica)
POO	Programação Orientada a Objetos
PPGI	Programa de Pós-graduação em Informática do iNCE
REST	Representational State Transfer
RIA	Rich Internet Applications
STI	Sistema de Tutor Inteligente
UFRJ	Universidade Federal do Rio de Janeiro

Sumário

CAPÍTULO I	15
1 INTRODUÇÃO	15
1.1 INTRODUÇÃO	16
1.2 MOTIVAÇÃO	17
1.3 CARACTERIZAÇÃO DO PROBLEMA	18
1.4 OBJETIVOS E HIPÓTESE	20
1.5 RELEVÂNCIA	21
1.6 ORGANIZAÇÃO DA DISSERTAÇÃO	23
CAPÍTULO II.....	26
2 AVALIAÇÃO COGNITIVA ATRAVÉS DE JOGOS PSICOPEDAGÓGICOS.....	26
2.1 VISÃO GERAL	27
2.2 REFERENCIAL TEÓRICO	28
2.2.1 <i>Elaboração Dirigida</i>	28
2.2.2 <i>Fio Condutor</i>	31
2.3 VIRTUALIZAÇÃO DE JOGOS PSICOPEDAGÓGICOS	32
CAPÍTULO III	36
3 SISTEMAS NEBULOSOS	36
3.1 FUNDAMENTOS DE LÓGICA NEBULOSA	37
3.2 LÓGICA NEBULOSA X LÓGICA CLÁSSICA	38
3.3 VARIÁVEIS E CONJUNTOS NEBULOSOS.....	39
3.4 FUNÇÕES DE PERTINÊNCIA	42
3.5 REGRAS E IMPLICAÇÕES LÓGICAS.....	44
3.6 SISTEMAS DE INFERÊNCIA NEBULOSOS	46
3.6.1 <i>Considerações sobre Sistemas Fuzzy</i>	49
CAPÍTULO IV.....	52
4 LÓGICA NEBULOSA EM JOGOS DIDÁTICOS.....	52
4.1 VISÃO GERAL	53
4.2 INTELIGÊNCIA COMPUTACIONAL EM JOGOS ELETRÔNICOS	54
4.3 DESCRIÇÃO DO PROBLEMA: A MODELAGEM DO ALGORITMO DE INTELIGÊNCIA COMPUTACIONAL	57
4.4 MODELAGEM DO PROBLEMA	58
CAPÍTULO V	60
5 NÉBULA – SISTEMA INTEGRADO PARA CONSTRUÇÃO DE INFERÊNCIAS NEBULOSAS	60
5.1 VISÃO GERAL	61
5.2 PROCESSO PARA CONSTRUÇÃO DE INFERÊNCIAS	64
5.3 MODELO ARQUITETURAL	68
5.4 MÓDULOS DO SISTEMA	72

5.4.1	Módulo de Configuração.....	73
5.4.1.1	Seleção de Variáveis.....	74
5.4.1.2	Definição dos Conjuntos Nebulosos (Editor de Funções de Pertinência).....	75
5.4.1.3	Especificação do Conjunto de Regras	77
5.4.2	Módulo de Simulação.....	78
5.4.3	Módulo de Inferências.....	80
5.4.4	Serviço de Inferências Fuzzy	80
5.5	AMBIENTE COMPUTACIONAL E TECNOLOGIAS	81
5.5.1	Arquitetura Geral do Sistema	81
5.5.2	Linguagem de Programação Java	82
5.5.3	Java ServerFaces.....	83
5.5.4	GlassFish Application Server	83
5.5.5	Banco de Dados JavaDB.....	83
5.5.6	Hibernate.....	84
5.5.7	JFuzzyLogic.....	84
CAPÍTULO VI.....		85
6	ESTUDO DE CASOS E ANÁLISE DE RESULTADOS	85
6.1	METODOLOGIA	86
6.2	SIMULADOR DO ESTACIONAMENTO DE VEÍCULO	88
6.2.1	Conceitos Gerais.....	88
6.2.2	Implementação do Simulador e Integração ao Nébulas	91
6.2.3	Testes integrados do simulador de estacionamento	97
6.2.4	Considerações Parciais.....	100
6.3	ANÁLISE MATEMÁTICA DO NÚCLEO DE INFERÊNCIAS	101
6.3.1	Conceitos Gerais.....	101
6.3.2	Implementação do Algoritmo no Matlab	102
6.3.3	Análise Matemática Comparativa dos Resultados.....	105
6.3.4	Considerações Parciais.....	111
6.4	ESTUDO DE CASO PRÁTICO EXPLORATÓRIO COM GRUPO DE ESPECIALISTAS	112
6.4.1	Conceitos Gerais.....	112
6.4.1.1	Modelo de Avaliação Cognitiva para as Funções de Atenção e o Jogo Psicopedagógico ‘Monta Boneco’.....	113
6.4.1.2	Modelo de Referência Proposto para as Configurações da Inferência.....	116
6.4.2	Metodologia para Análise de Usabilidade da Interface do Sistema Nébulas.....	119
6.4.3	Metodologia para Realização do Experimento.....	122
6.4.4	Considerações Parciais.....	126
CAPÍTULO VII		130
7	CONSIDERAÇÕES FINAIS	130
7.1	RESUMO DO TRABALHO	131
7.2	CONTRIBUIÇÕES	132
7.3	LIMITAÇÕES E DIFICULDADES ENCONTRADAS.....	134
7.4	TRABALHOS FUTUROS.....	135
REFERÊNCIAS BIBLIOGRÁFICAS.....		138
APÊNDICE		144
8	APÊNDICE: ROTEIRO UTILIZADO NA AVALIAÇÃO DE USABILIDADE DO SISTEMA NÉBULA	145

Capítulo I

Introdução

“Tudo é loucura ou sonho no começo. Nada do que o homem fez no mundo teve início de outra maneira — mas já tantos sonhos se realizaram que não temos o direito de duvidar de nenhum.”

Monteiro Lobato

Neste capítulo apresenta-se uma visão geral do projeto de pesquisa descrito nesta dissertação, abordando suas justificativas, identificando o problema e declarando as hipóteses e objetivos a serem perseguidos. Por fim, apresenta-se a metodologia de organização do texto que será utilizada no decorrer da dissertação.

1.1 Introdução

A utilização de recursos informatizados aplicados como ferramentas de apoio à decisão nas áreas de saúde e educação vem se consolidando a partir de uma série de experiências bem sucedidas em diversas áreas do conhecimento. Em especial, destaca-se a utilização de ambientes virtuais para a aplicação de jogos didáticos para avaliação e reabilitação cognitiva, que se privilegiam do uso das novas tecnologias, agregando novas possibilidades para atividades tradicionalmente desenvolvidas em meio físico.

Deste cenário de aplicação resulta um vasto campo de possibilidades, permitindo que novas abordagens para estimulação e análise de diversas funções cognitivas possam ser trabalhadas, cuja eficiência pode ser potencializada com a aplicação de recursos computacionais inteligentes.

Neste contexto, a utilização de jogos eletrônicos representa uma importante ferramenta de apoio para auxiliar psicólogos e especialistas em neurociências em suas avaliações cognitivas. Segundo Vygotsky (1989), através do jogo, a criança aprende fazendo, sua curiosidade é estimulada, adquire iniciativa e autoconfiança, e o desenvolvimento da linguagem, do pensamento e da concentração é proporcionado.

Portanto, o uso de jogos eletrônicos para estimulação e análise de funções cognitivas vem abrir novas perspectivas ao explorar abordagens diferenciadas que as tecnologias proporcionam. Neste cenário, diversas experiências positivas podem ser observadas nos trabalhos de Marques et al(2009b), Tordeschini et AL (2000) e Costa et AL (2005). Segundo Rieder et al (2004), os jogos educativos têm a capacidade de explorar atividades lúdicas que estimulam características específicas para o desenvolvimento da cognição. Ao apresentarem desafios em diferentes níveis, têm a capacidade de estimular funções cognitivas básicas, como memória, atenção e concentração.

Destaca-se, porém, o principal desafio de modelar matematicamente algoritmos informatizados que representem toda a complexidade e as incertezas inerentes aos modelos para avaliação cognitiva, em que uma série de nuances precisam ser observadas e consideradas, muitas das quais difíceis de serem quantificadas em valores precisos. Além disso, cada função cognitiva exige a elaboração de um modelo

matemático específico, feito sob medida para a função a ser avaliada, cujo processo de especificação é trabalhoso e igualmente complexo.

No modelo tradicional, o processo de análise cognitiva é realizado manualmente por psicólogos e especialistas em neurociências¹, a partir da observação e interpretação do andamento da sessão de jogo, baseado em seu conhecimento empírico e experiências práticas. Este conhecimento, porém, é de difícil formalização e representação em modelos matemáticos, devido à sua complexidade por tratarem de características abstratas e imprecisas.

Neste contexto, ressalta-se a utilização da técnica de lógica nebulosa, ou lógica *fuzzy*, para a representação das imprecisões e incertezas envolvidas na modelagem dos critérios de avaliação cognitiva.

Portanto, este trabalho apresenta uma plataforma integrada para auxiliar as equipes de especialistas na construção de modelos de inferências nebulosas para serem aplicados durante as sessões de jogo no processo de análise cognitiva. A plataforma apresenta uma interface simplificada que objetiva abstrair as representações matemáticas complexas de uma maneira didática, com orientações textuais e gráficos, de maneira que os algoritmos possam ser criados, testados e validados de maneira independente e desacoplada da construção dos jogos eletrônicos.

1.2 Motivação

A virtualização de jogos didáticos em sistemas informatizados representa a possibilidade de potencializar os resultados obtidos nos jogos tradicionais a partir da aplicação de recursos computacionais para a automatização do processamento e da análise de alguns conjuntos de dados. Adicionalmente, ao disponibilizar ambientes virtuais que forneçam mecanismos automatizados para a aplicação dos jogos, permite-se uma maior democratização do acesso a estes recursos pela sociedade, beneficiando um maior número de indivíduos.

Neste contexto, a complexidade envolvida no processo de elaboração dos algoritmos computacionais, em que diversas especialidades são requeridas com a

¹ O termo especialista em neurociências ou neuropedagogia, neurocientista ou, simplesmente, especialista, será utilizado neste trabalho para designar o profissional capacitado para aplicação dos jogos psicopedagógicos e avaliação final de seus resultados.

participação de equipes multidisciplinares, representa uma importante oportunidade de disponibilizar um ambiente integrado que viabilize este processo de maneira simples. Os aspectos técnico-científicos envolvidos, desde a complexidade de representar as características necessárias para a elaboração dos algoritmos de inteligência computacional, quanto à dificuldade em elaborar interfaces simplificadas que abstraíam as representações matemáticas, considerando os critérios de usabilidade e conceitos da área de neurociências, são especialmente desafiadores. A integração de áreas e conceitos essencialmente tão distintos torna o objeto deste estudo especialmente atraente do ponto de vista acadêmico.

Por fim, uma das principais motivações para o desenvolvimento deste trabalho consiste na importância que ele pode vir a representar no andamento da linha de pesquisa em jogos psicopedagógicos, oferecendo uma plataforma que pode viabilizar o desenvolvimento de demais estudos nas áreas tanto de inteligência computacional quanto nas neurociências, utilizando os dados fornecidos pelo sistema como subsídio para aplicação em seus respectivos estudos.

1.3 Caracterização do Problema

A construção dos jogos informatizados como aplicações didáticas envolve diversas etapas de elaboração, desde a identificação dos requisitos psicopedagógicos até a modelagem computacional das funcionalidades que irão atendê-los. Neste ponto, ressalta-se a complexidade de construir algoritmos matemáticos que consigam descrever de maneira eficiente o conhecimento do especialista na representação dos critérios a serem avaliados durante a sessão de jogo.

O processo de avaliação cognitiva envolve o trabalho de diversos profissionais, em equipes multidisciplinares, para determinar os critérios que serão utilizados, de acordo com a função cognitiva em questão, e como os resultados obtidos serão tratados. Estas equipes multidisciplinares, além de neurocientistas e psicólogos, precisam ainda envolver profissionais de matemática e de informática, que precisam compreender o problema e traduzi-lo em um modelo que possa ser informatizado. Em função de suas características inexatas e imprecisas, a construção dos modelos matemáticos pode exigir a utilização de técnicas de inteligência computacional, considerando especialmente neste contexto a técnica de lógica nebulosa.

Uma vez construídos os modelos matemáticos dos critérios de avaliação, o profissional de informática precisa integrá-lo ao jogo, codificando-o em uma linguagem de programação. Esta tarefa envolve minuciosos estudos sobre as possíveis técnicas de inteligência computacional a serem utilizadas, e complexas iterações de modelagem e implementação. Somente após ter disponível uma versão testável do *software* é que as equipes especialistas poderão avaliar se os resultados obtidos pelo algoritmo estão de acordo com o esperado. Naturalmente, no decorrer dos testes, as especificações do algoritmo precisam ser ajustadas, analisando e sintonizando seus parâmetros em busca de resultados mais precisos. Portanto, todo este ciclo deve ser refeito, culminando com uma nova implementação do *software* para, enfim, refazer os testes. Este custoso processo pode envolver várias etapas iterativas até que se consiga atingir o objetivo proposto.

Neste cenário, o funcionamento do algoritmo somente é possível de ser avaliado no contexto do jogo em execução. Disto decorre que, para cada jogo a ser criado, é necessário construir e agregar ao software o algoritmo de inferência que irá fornecer os recursos de inteligência computacional requeridos. Como premissa, os algoritmos inteligentes precisam de diversos ajustes e revisões, realizados em simulações iterativas, para que atinjam um grau satisfatório de precisão, o que pode ser dispendioso considerando sua implementação acoplada à codificação do jogo. Neste processo tradicional, após a elaboração do modelo matemático do algoritmo, ele precisa ser implementado e integrado ao jogo, que deve então ser compilado e publicado para execução. Uma vez disponível, o jogo precisa ser testado em diversas simulações, quando os dados obtidos são validados em comparação com os objetivos propostos. Neste momento, podem-se avaliar os ajustes que serão necessários no algoritmo, e todo um novo ciclo de desenvolvimento precisará ser refeito. A Figura 1.1 ilustra o processo tradicional para a criação de algoritmos inteligentes integrados aos jogos psicopedagógicos.

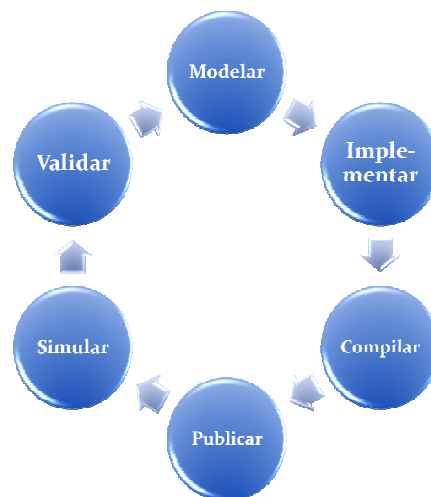


Figura 1.1: Processo tradicional para construção de algoritmos nos jogos psicopedagógicos

Neste contexto, identifica-se a principal questão que representa o problema ao qual este projeto de pesquisa se dedica: “Como viabilizar a definição dos algoritmos nebulosos, de forma dinâmica e desacoplada da codificação dos jogos psicopedagógicos, de maneira simplificada e permitindo a participação das equipes multidisciplinares envolvidas?”

1.4 Objetivos e Hipótese

O objetivo principal deste trabalho é propor a utilização de um sistema integrado para construção dos algoritmos de inferência nebulosos, que forneça uma interface simplificada e que permita que os algoritmos de inferência sejam construídos e testados de maneira independente do processo de codificação dos jogos eletrônicos.

Ao disponibilizar uma interface simplificada que oriente os usuários na construção dos algoritmos nebulosos, abstraindo as representações matemáticas em conceitos representados de maneira próxima à sua linguagem natural, objetiva-se permitir que as equipes multidisciplinares possam participar diretamente do processo de construção lógica dos algoritmos, interagindo diretamente com o sistema, testando e analisando os resultados.

Ao permitir que os algoritmos recém criados possam ser imediatamente testados, analisados e corrigidos, objetiva-se diminuir o custoso processo de criação dos jogos com recursos de inteligência computacional, permitindo melhorar e calibrar os algoritmos sem a necessidade de compilar e publicar o código-fonte do jogo. As

alterações feitas nos parâmetros do algoritmo poderão ser imediatamente visualizadas na ferramenta, obtendo-se melhor dinamismo e permitindo o envolvimento de todas as partes interessadas, participando ativamente deste processo a partir de um mesmo ambiente comum.

Portanto, as hipóteses de solução para a proposta deste trabalho podem ser formalmente descritas da seguinte maneira:

- **H1: “É possível utilizar uma mesma ferramenta integrada para permitir a construção, testes e ajustes nos algoritmos de inferência nebulosos, simplificando seu processo de construção e desacoplando-o da codificação dos jogos?”**
- **H2: “A simplificação do processo de construção dos algoritmos em uma ferramenta integrada que ofereça a abstração das representações matemáticas complexas pode facilitar o envolvimento dos perfis especialistas participantes das equipes multidisciplinares na tarefa de elaboração das inferências?”**

1.5 Relevância

A contribuição deste trabalho se concentra em disponibilizar uma ferramenta que permita construir e agregar aos jogos eletrônicos recursos computacionais inteligentes, de maneira simplificada e disponível para usuários de várias especialidades técnicas, diminuindo a concentração do conhecimento e democratizando o acesso a este tipo de informação. Sua importância pode ser avaliada na continuidade da linha de pesquisa de jogos psicopedagógicos e sua democratização na sociedade, objetivando atender a um maior número de pessoas necessitadas, considerando que a automatização do processamento de determinados conjuntos de resultados utilizando recursos computacionais permite a realização de um maior número de atendimentos.

Desta maneira, os profissionais de neurociências terão mais uma ferramenta de auxílio para suas tarefas de avaliação cognitiva, compartilhando seu conhecimento e democratizando o acesso à informação por toda a equipe multidisciplinar envolvida.

A Figura 1.2 representa um diagrama esquemático acerca da relevância do sistema Nébulas, inserido no contexto do projeto de virtualização de jogos

psicopedagógicos do PPGI/iNCE. Considerando que a linha de pesquisa já possui toda sua fundamentação teórica consolidada, a partir dos trabalhos de Marques e Seminério, e uma série de jogos disponíveis em meio físico, os aspectos técnicos foram alicerçados pela arquitetura definida para a plataforma computacional de desenvolvimento de jogos psicopedagógicos (FERREIRA, 2009). Neste sentido, esta plataforma arquitetural dispõe do framework Phidias (GOMES, MORAES, BETTINI, 2009), que oferece ferramentas para a construção dos jogos e disponibiliza uma base de dados estruturada para armazenamento das informações coletadas durante as sessões de jogo. Portanto, o sistema integrado Nébula se propõe a preencher uma importante lacuna dentro do contexto da linha de pesquisa, ao oferecer a possibilidade de utilizar recursos de inteligência computacional nos jogos psicopedagógicos, caracterizados por sistemas de inferência nebulosos.



Figura 1.2: Relevância do Nébula, no contexto do projeto de virtualização de jogos psicopedagógicos

1.6 Organização da Dissertação

A estrutura da dissertação foi definida buscando-se especificar um seqüenciamento lógico dos assuntos a serem abordados, orientando o leitor na condução de seu raciocínio, apresentando desde a fundamentação teórica dos conceitos envolvidos, até a descrição dos aspectos técnicos do trabalho e os resultados obtidos.

Após a introdução, são apresentados os conceitos teóricos envolvidos na concepção dos jogos psicopedagógicos e os fundamentos de lógica *fuzzy*. Em seguida, apresenta-se a proposta da dissertação, com o sistema de inferências resultante do projeto de pesquisa e os estudos realizados. Por fim, as considerações finais são apresentadas.

Portanto, este trabalho está organizado em sete capítulos:

- **Capítulo 1 – Introdução:** apresenta os conceitos gerais da dissertação, a motivação que originou o projeto de pesquisa, destacando a descrição do problema e os objetivos da dissertação;
- **Capítulo 2- Avaliação cognitiva através de Jogos Psicopedagógicos:** apresenta a fundamentação teórica presente na concepção dos jogos psicopedagógicos, descrevendo a metodologia utilizada na linha de pesquisa e relacionando alguns trabalhos correlatos encontrados na literatura;
- **Capítulo 3 – Sistemas Nebulosos:** apresenta os conceitos fundamentais envolvidos na teoria de lógica *fuzzy*, necessários para a contextualização e entendimento dos termos e metodologias abordadas nesta dissertação;
- **Capítulo 4 – Lógica Nebulosa em Jogos didáticos:** apresenta o cenário atual de utilização de técnicas de inteligência computacional, especialmente lógica nebulosa, em jogos de natureza didática. Este capítulo descreve o problema relacionado ao objeto principal de estudo deste trabalho, e a modelagem para a solução proposta;
- **Capítulo 5 – Nébula – Sistema Integrado para construção de Inferências Nebulosas:** descreve o principal produto deste trabalho, apresentando a proposta de utilização da plataforma integrada Nébula para a construção dos algoritmos de inferência nebulosos a serem utilizados nos jogos

psicopedagógicos. Define a arquitetura geral do sistema, descreve os seus módulos principais, o processo para construção dos algoritmos e o ambiente computacional utilizado pelo sistema;

- **Capítulo 6 – Estudo de Casos e Análise de Resultados:** apresenta os estudos realizados na utilização prática do sistema e descreve os resultados obtidos. Apresenta também os estudos para avaliar a precisão dos resultados dos algoritmos elaborados a partir da plataforma integrada, comparativamente a modelos reconhecidamente válidos, utilizando o Matlab;
- **Capítulo 7 – :** apresenta as considerações finais e as propostas para o desenvolvimento de trabalhos futuros que podem ser realizados.

A Figura 1.3 apresenta graficamente a maneira como esta dissertação está organizada em capítulos.

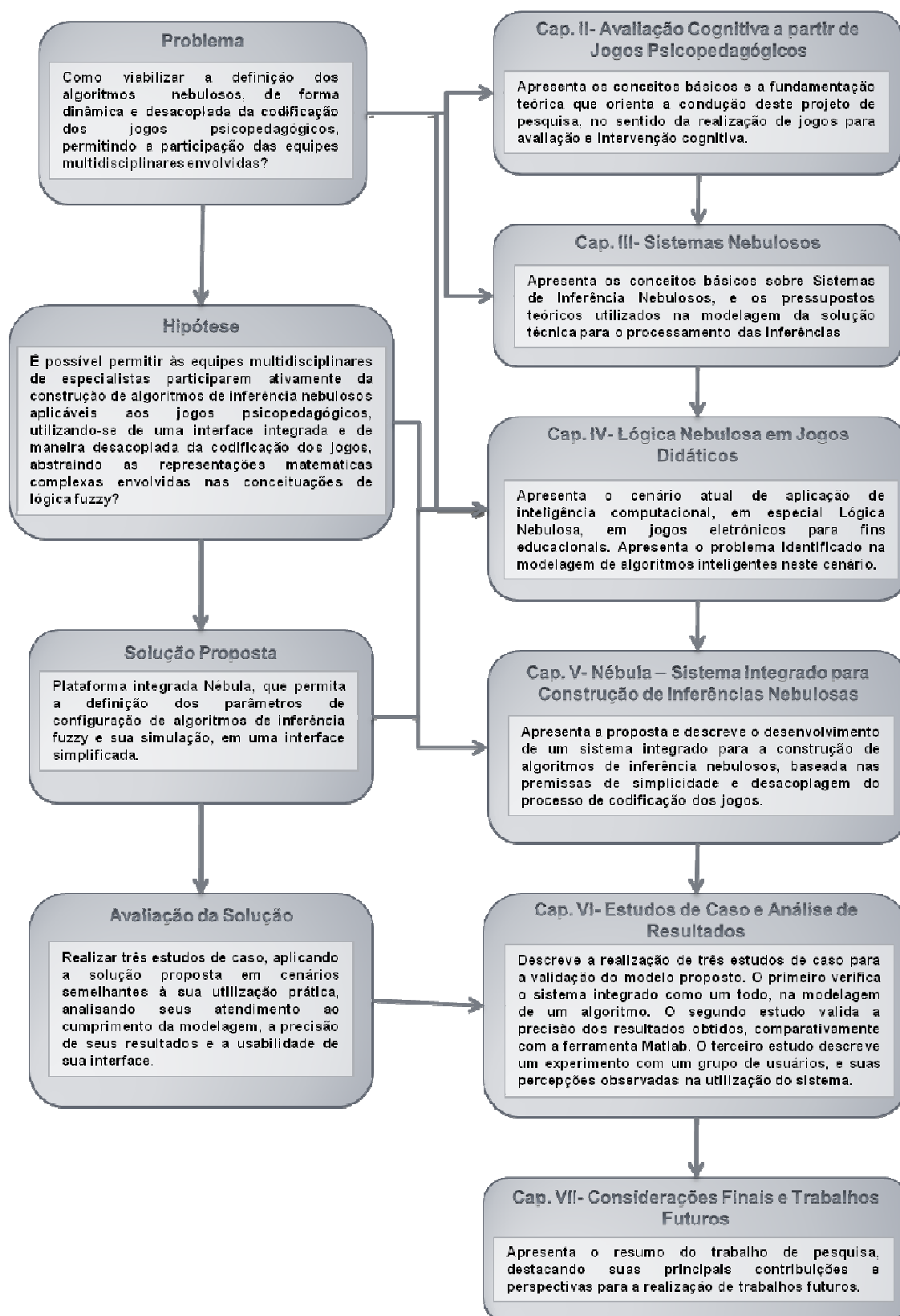


Figura 1.3: Diagrama esquemático da organização da dissertação

Capítulo II

Avaliação cognitiva através de Jogos Psicopedagógicos

“O que é necessário não é a vontade de acreditar, mas o desejo de descobrir, que é justamente o oposto.”

Bertrand Russell

Neste capítulo são apresentados os fundamentos teóricos que conceituam a linha de pesquisa de virtualização de jogos psicopedagógicos, sua contextualização histórica e seus principais pressupostos fundamentados nos trabalhos de Carla Marques e Franco Seminério para avaliação e intervenção no desenvolvimento cognitivo.

2.1 Visão Geral

A cognição humana diz respeito à maneira como lidamos com o conjunto de regras que especificamos para solucionar um determinado problema. E a maneira como elaboramos este conjunto de regras pode ser definido como o processo de aprendizado. Compreender a maneira como estas regras são formadas, no processo de desenvolvimento cognitivo, pode permitir a atuação de diversas frentes pedagógicas no sentido de potencializar o aprendizado e mapear suas áreas de deficiência.

Desta forma, as áreas de educação e pedagogia têm se beneficiado significativamente dos recentes avanços das neurociências cognitivas, que estudam a maneira como o desenvolvimento do cérebro estimula ou limita o desenvolvimento cognitivo (FERREIRA, 2009). Em especial, destaca-se a utilização de jogos para aplicação de metodologias pedagógicas, inseridas em processos educacionais contextualizados nesta metodologia, beneficiando-se especialmente de seu fator motivador, sua facilidade e abrangência de aplicação. Para Silveira e Barone (1998), um jogo educativo por computador é uma atividade de aprendizagem inovadora na qual as características do ensino apoiado em computador e as estratégias de jogo são integradas para alcançar um objetivo educacional específico.

Muitos dos jogos tradicionalmente conhecidos, como jogo da memória e quebra-cabeças, exploram diversas atividades lúdicas ao estimular a concentração (RIEDER, BRANCHER, 2004). Uma vez que, aplicados segundo uma metodologia psicopedagógica fundamentada, alguns destes jogos, especialmente desenhados para atender a uma função cognitiva específica, podem servir como poderosas ferramentas para apoiar a avaliação e o desenvolvimento cognitivo. Segundo Andrade et al (2004), jogos eletrônicos inteligentes, se guiados por objetivos psicopedagógicos bem definidos, podem ser utilizados para desenvolver uma extensa gama de capacidades cognitivas.

Rieder e Brancher (2004) afirmam que os jogos educativos têm a capacidade de explorar atividades lúdicas que estimulam características específicas para o desenvolvimento da cognição. Ao apresentarem desafios em diferentes níveis, têm a capacidade de estimular funções cognitivas básicas, como memória, atenção e concentração.

Neste sentido, embora diversas linhas teóricas possam ser observadas na literatura para a aplicação destes processos pedagógicos inseridos nos jogos, este trabalho se concentrará essencialmente nas teorias da Elaboração Dirigida, elaborada por Franco Seminério (1987), e do Fio Condutor, proposta por Marques (2009a).

A metodologia de Elaboração Dirigida, proposta por Seminério, objetiva aprimorar a forma como crianças de 7 a 12 anos formam suas regras. A essência de seu trabalho reside na intervenção psicopedagógica sobre a cognição na infância (MARQUES, 2009b).

Fundamentada nos estudos de Seminério, Marques propôs a teoria de Fio Condutor, que objetiva fornecer um caminho para conduzir a criança no sentido da reflexão sobre seus pensamentos, suas ações e as informações com que lida diariamente, levando-a a pensar de maneira diferente e a construir seu modo próprio de compreensão dos mecanismos que a cercam (MARQUES et al, 2009a).

Diante deste cenário, a utilização de jogos psicopedagógicos se apresenta como um importante recurso para estimular o desenvolvimento cognitivo, fundamentados nos pressupostos teóricos de Marques e Seminério.

2.2 Referencial Teórico

2.2.1 Elaboração Dirigida

Segundo Seminério (1987), a metacognição pode ser compreendida como o uso reflexivo da cognição, a partir da qual somos capazes de estruturar a lógica no processo de construção do conhecimento. Bertolini e Silva (2005) definem que a “metacognição implica o indivíduo ser capaz de conhecer e de auto-regular o seu próprio funcionamento cognitivo com a finalidade de solucionar problemas”. Segundo Marques et al (2009a), “a capacidade de controle reflexivo e consciente de uma regra identificada através da descrição da composição dos significados que a compõem caracterizam a metacognição”.

Desenvolver a metacognição, portanto, significa aprender a aprender, saber criar e utilizar estratégias próprias de aprendizado. No entanto, um grande desafio se apresenta na área de educação: como ensinar este processo de auto-descoberta?

A elaboração dirigida fundamenta-se em um modelo experimental que objetiva estimular o processo de metacognição por meio de atividades discursivas e reflexivas, privilegiando o uso metaprocessual para aquisição de conhecimentos através da intervenção do educador, com valorização do aspecto relacional e afetivo da aprendizagem e da motivação, ampliando a capacidade da criança em refletir sobre seus processos cognitivos. Em suma, “a Elaboração Dirigida busca oferecer à criança regras generativas, através de um modelo de interação interpessoal, para a dedução do conhecimento almejado” (MARQUES, 2009b). Este modelo orienta a aplicação dos jogos psicopedagógicos, em que a criança precisa detectar ativamente o modelo lógico presente no jogo, cujas características cognitivas encontram-se previamente mapeadas.

Segundo Seminério, existem dois canais morfogenéticos privilegiados para o desenvolvimento dos aspectos da cognição humana: o viso-motor e o áudio-fonético. Ambos se referem a competências inatas, responsáveis pelo modo como percebemos e construímos o mundo. Os canais morfogenéticos são capazes de sustentar e desenvolver os processos estruturados de nosso conhecimento (FERREIRA, 2009).

O canal viso-motor representa a transmissão e tratamento da informação, interligando a percepção visual e a ação motora. O canal áudio-fonético corresponde ao meio de transmissão da informação que interliga a percepção do meio auditivo e organização da ação motora com a produção dos fonemas de fala (FERREIRA, 2009).

No sentido computacional abrangido pela aplicação dos jogos psicopedagógicos, Seminério (1987) compreende estas competências como linguagens inatas, fundamentadas em quatro códigos da cognição humana:

- **L1:** competência para organizar estímulos em estruturas. Trata-se da linguagem-código mais primária e elementar do processo cognitivo. Para Seminério, esta é uma habilidade inata para organização de dados sensoriais em sistemas de formas regulares;
- **L2:** corresponde à habilidade de associar as estruturas lidas em L1, atribuindo-lhes um significado empírico. Corresponde à capacidade de atribuir significados a partir da percepção;
- **L3:** utiliza as figurais estruturais decorrentes da L2 para formar contextos significativos no sentido episódico, atuando como base para pensar criativamente, narrar e redigir, representados em termos imaginíficos no canal

visomotor, e frases discursivas no canal audiofonético. É a linguagem mais próxima à humanização;

- **L4:** refere-se à gênese da lógica, e constitui a competência que permite desenvolver o controle consciente sobre qualquer processo cognitivo, responsável pelo processo de captar e de gerar regras, elaborando qualquer tipo de raciocínio.

Diante deste cenário, Seminário propôs a teoria das Linguagens Código, fundamentada na hipótese de que estes canais estariam diretamente relacionados, e que “estes seriam os mecanismos responsáveis por permitir o uso da informação em termos de modos especializados de codificação, decodificação e recodificação” (MARQUES, 2009b).

Este mecanismo que decompõe a hierarquia de linguagens permitiu perceber os elementos prefixados em cada linguagem, responsáveis pelo acesso aos demais níveis, vislumbrando um caminho que permite o percurso do processo cognitivo. Isto permitiria aplicar processos pedagógicos que estimulem os alunos a identificar regras, fornecendo caminhos para permitir o salto cognitivo entre os níveis de aprendizado, auxiliando o aluno a compreender o seu próprio processo de aprendizado.

Neste contexto se apóia a fundamentação teórica da proposta pedagógica do projeto de jogos psicopedagógicos, que objetiva orientar a criança para a compreensão da existência de regras, em quaisquer situações, e a implícita capacidade para descobri-las, esperando-se alcançar um salto na escala de suas possibilidades metacognitivas (MARQUES, 2009a). Isto estimula o aluno a refletir sobre a regra que gerou a regra, em um processo recursivo que significa entender os mecanismos intrínsecos que determinam determinado problema ou comportamento.

Como instrumentos para a aplicação da técnica de Elaboração Dirigida, Seminário elaborou um conjunto de jogos, utilizando-se preferencialmente de materiais reciclados ou reutilizados, adaptados em função de suas possibilidades lúdicas e motivadoras. Além de sua eminente simplicidade, os materiais escolhidos focavam trabalhar os aspectos clássicos na aplicação de regras para seriação, classificação e inclusão, atuando especialmente com o objetivo de trabalhar o desenvolvimento dos mecanismos da lógica, da atenção, da percepção, da memória, da matemática, do imaginário, da alfabetização e da linguagem. Cada jogo está voltado à atividade de um

canal morfogênico específico, a ser trabalhado segundo a aplicação da técnica de Elaboração Dirigida.

2.2.2 Fio Condutor

Fundamentada nos estudos e propostas de Seminério, Marques elaborou a teoria do Fio Condutor (MARQUES, 2009b), como uma metodologia prática para aplicação da técnica de Elaboração Dirigida no contexto da avaliação e reabilitação cognitiva. Com base nestes estudos, foi elaborado um conjunto de jogos, especialmente designados para o trabalho e desenvolvimento metacognitivo, e que preconizam, conforme o modelo de fio Condutor, a presença de sete momentos distintos, assim denominados fases ou versões. Estes jogos auxiliam o psicólogo a localizar, estimular e intervir no processo de cognição da criança.

Desta forma, a técnica de Fio Condutor estrutura em organização prática a aplicação da Elaboração Dirigida, assim definidas por Marques (2009b):

- **Fase 1:** primeiro contato do jogador com o ambiente de jogo. Neste momento, nenhuma regra é apresentada ao jogador, e o jogo é iniciado a partir de sua compreensão intuitiva. Um mediador, denominado aplicador, acompanha todas as ações do jogador, testando o comportamento e o estágio da criança;
- **Fase 2:** o aplicador solicita à criança que o explique suas ações realizadas anteriormente. O aplicador registra as explicações da criança;
- **Fase 3:** é subdividida em dois momentos, em que no primeiro todas as peças do jogo são soltas e dispostas para que o jogador as organize no tabuleiro à sua maneira. No segundo momento, o aplicador solicita ao jogador que explique o que fez anteriormente.
- **Fase 4:** o jogador verbaliza a regra elaborada. O aplicador atua como mediador, formulando perguntas para auxiliar o jogador, no intuito de fazê-lo formular a regra para o jogo e a verbalize. O aplicador deve registrar todas as explicações fornecidas. É nesta fase que se caracteriza a aplicação da Elaboração Dirigida;
- **Fase 5:** o jogo é iniciado novamente, e o aplicador verifica se o jogador conseguiu atingir os objetivos. Funciona como um teste para verificar o resultado da aplicação da técnica de Elaboração Dirigida;

- **Fase 6:** O especialista intervém a partir dos dados coletados na aplicação da Elaboração dirigida;
- **Fase 7:** o mesmo jogo é aplicado, mas com tarefas diferentes para confirmar o resultado da aplicação. Neste fase, verifica-se a complexidade do jogo, podendo haver um aumento propositado da complexidade apresentada.

A Figura 2.1 ilustra sinteticamente as fases componentes da metodologia do Fio Condutor.

FASES DE CONSTRUÇÃO E APLICAÇÃO DO PROTOCOLO/JOGO						
VERSÃO 1	VERSÃO 2	VERSÃO 3	VERSÃO 4	VERSÃO 5	VERSÃO 6	VERSÃO 7
AVALIAÇÃO ESTÁTICA	AVALIAÇÃO DINÂMICA	AVALIAÇÃO DINÂMICA	AVALIAÇÃO DINÂMICA	AVALIAÇÃO ESTÁTICA	INTERVENÇÃO	REPETIÇÃO DAS VERSÕES 1 E 2 COM TAREFAS DIFERENTES
APRESENTAÇÃO DO ESTÍMULO	MEDIAÇÃO PARA A CONSTRUÇÃO DA REGRA	CONSTRUÇÃO DA REGRA COM OBJETOS MANIPULÁVEIS	ELABORAÇÃO DIRIGIDA	RETESTE MEDIADO		

Figura 2.1: Fases componentes da metodologia de Fio Condutor. Fonte: Marques (2009b)

2.3 Virtualização de Jogos Psicopedagógicos

Em função dos recentes avanços tecnológicos, e das diversas potencialidades representadas pela utilização de recursos computacionais como aliados na aplicação de processos pedagógicos, diversas iniciativas se estabeleceram no sentido de utilizar sistemas informatizados como ferramentas de apoio na área de educação. Em especial, destacam-se os cenários de utilização de jogos para finalidades didáticas.

Se comparado com os procedimentos realizados manualmente, a aplicação de jogos assistida por computador possui inúmeras vantagens, dentre as quais podem-se destacar: permite a repetição de exercícios, é possível sistematizar e alterar o nível de dificuldade de acordo com o desempenho do usuário, processa dados e disponibiliza os resultados, oferece uma motivação externa adicional, tendo ainda, no caso de programas

mais sofisticados, a possibilidade de dar imediato *feedback*, individualizado para cada tipo de resposta (GOMES et al, 2010 *apud* COSTA, 2000).

Desta forma, durante uma sessão de jogo, seja ela realizada em meios físicos ou digitais, diversas informações são geradas, que têm o potencial de auxiliar o aplicador no processo de avaliação cognitiva da criança em acompanhamento. Estas informações, quando trabalhadas de maneira unificada, podem servir de insumo para auxiliar o aplicador em suas decisões e avaliações. No entanto, quando realizada por meios físicos, muitas destas informações podem ser perdidas, uma vez que dependem da capacidade de percepção e registro imediato por parte do aplicador, o que poderia variar em função de seu grau de experiência, perfil especialista ou estilo. Além disto, em função das limitações humanas no tratamento de diversas informações em paralelo, muitas destas nuances poderiam passar despercebidas.

Desta forma, delegar a um sistema informatizado a captura e o processamento, durante a sessão de jogo, de todo um conjunto de informações de interesse previamente mapeadas, pode potencializar as possibilidades de análise e cruzamento de informações coletadas, aumentando o poder decisório do especialista e ampliando sua própria percepção sobre desempenho da criança, ao analisar relatórios e dados consolidados que poderiam ser oferecidos pelo sistema.

Neste cenário, ocorreu a iniciativa de um projeto conduzido pelo PPGI/iNCE, da Universidade Federal do Rio de Janeiro, com o objetivo de virtualizar em sistemas informatizados um conjunto de jogos psicopedagógicos criados por Marques, originalmente concebidos em meio físico, e seguindo os preceitos estabelecidos pelas técnicas de Elaboração Dirigida e do Fio Condutor. Associados aos benefícios de processamento computacional descritos anteriormente, a disponibilização dos jogos em meios eletrônicos buscou essencialmente democratizar o seu acesso, buscando um maior número de atendimentos, e com isto, “disseminar o processo de avaliação neuropsicológica às regiões carentes de profissionais especializados em educação especial e neuropedagogia, e de alunos que não têm condições econômicas para atendimento” (MARQUES, 2009c).

Portanto, a iniciativa de virtualizar os jogos psicopedagógicos criados por Marques contou com a participação de profissionais de diversos perfis especialistas, organizados em equipes multidisciplinares, envolvendo as áreas de psicologia, pedagogia, letras, medicina e computação, no intuito de abranger todos os aspectos

considerados nas metodologias pedagógicas envolvidas, minimizando as perdas decorrentes do processo de virtualização. Desta forma, o projeto de virtualização fundamentou-se nas seguintes premissas:

- Democratização do acesso aos jogos por um maior número de pessoas, em função da sua liberdade de uso comparativamente às limitações físicas dos jogos tradicionais;
- Preservação das características gráficas e do design metacognitivo propostos nos jogos originais;
- Preservação da figura fundamental do aplicador, especialista em neurociências, que não será substituída pelos engenhos computacionais propostos pelos jogos;
- Minimização das perdas decorrentes do processo de virtualização dos jogos, relativas à manipulação física dos elementos, a partir da inserção de recursos lúdicos adicionais, como animações, canções de fundo e respostas sonoras, previamente modeladas e aprovadas pelas equipes multidisciplinares que compõem os perfis especialistas envolvidos na concepção dos jogos;
- Possibilidades de registro e armazenamento automático de um conjunto de informações, que poderiam servir de insumo para processamentos posteriores, como projeções heurísticas em buscas de padrões;
- Possibilidade de incrementar algoritmos de inteligência computacional no processamento das informações coletadas durante a sessão de jogo, vislumbrando possibilidade de comportamento adaptativo, previsão, pré-análise, projeção e matematização de critérios e crivos de avaliação;
- Capacidade de reutilização dos jogos em cenários e condições semelhantes, e sua extensão para possibilidades adicionais, como atendimento a deficientes visuais e auditivos, utilizando-se de recursos proporcionados pelas tecnologias da informação;

Neste contexto, diante da diversidade de jogos existentes passíveis de serem virtualizados, Ferreira (2009) propôs um detalhado estudo sobre a definição de uma arquitetura computacional integrada para a construção de jogos neuropedagógicos, seguindo a linha teórica de Marques para o Fio Condutor. Esta arquitetura, que norteou

o desenvolvimento dos jogos, propõe uma infraestrutura composta por diferentes visões, representadas pela visão do jogador, do aplicador e do especialista avaliador, além de informações sobre repositórios, meios de comunicação e agentes computacionais.

Em conformidade com esta arquitetura, Gomes, Moraes e Bettini (2009) propuseram a construção de uma plataforma de desenvolvimento de jogos psicopedagógicos denominada Phidias, criada com a finalidade de abstrair a complexidade do processo de construção dos jogos, fornecendo uma infraestrutura sobre a qual os jogos possam ser desenvolvidos. Esta plataforma implementa um conjunto de requisitos para a construção do ambiente visual e processamento gráfico, além de fornecer a captura automática de diversos eventos de interação da criança durante a sessão de jogo, como: tempo entre jogadas, erros, persistências e desistências, movimentação de peças, entre outros. Estes dados são armazenados em uma base estruturada que centraliza as informações geradas por todos os jogos desenvolvidos na plataforma Phidias, que podem ser utilizados posteriormente para processamento e interpretação das informações.

Alguns dos principais jogos desenvolvidos são os Jogos da Roda da Linguagem, o Cria Conto e o Jogo dos Elásticos, ilustrados respectivamente pela Figura 2.2, e podem ser verificados nos trabalhos de Marques et al (2009a, 2009c) e de Ferreira (2009).



Figura 2.2: Imagens dos jogos virtualizados: Roda da Linguagem, CriaConto e Jogo dos Elásticos, respectivamente

Alguns resultados acerca da aplicação dos jogos psicopedagógicos para avaliação cognitiva em deficientes visuais são apresentados por Marques et al (2009). Em outro trabalho, Marques et al (2009c) descrevem a experiência acerca da virtualização em sistemas informatizados de diversos jogos psicopedagógicos originalmente construídos em meio físico.

Capítulo III

Sistemas Nebulosos

“Quando um monte de areia deixa de ser um monte de areia, caso retiremos um grão de cada vez?”

Eubulides de Mileto

A matemática clássica, caracterizada por modelos e equações descritos a partir de representações precisas, muitas vezes revela suas limitações ao representar problemas do mundo real onde um alto grau de imprecisão está envolvido. No contexto da aplicação de jogos psicopedagógicos, onde se buscam modelos quantitativos para aproximar as metodologias inerentemente qualitativas, a utilização de modelos matemáticos tradicionais não traduz adequadamente toda a complexa representação de suas características imprecisas.

Neste contexto, a lógica nebulosa se apresenta como uma maneira de modelar o modo de aproximação característico do raciocínio, imitando a habilidade humana de tomar decisões em ambientes de incerteza e imprecisão. (GOLDSHMIDT, 2007).

Este capítulo objetiva apresentar brevemente uma explicação sobre os conceitos essenciais relacionados à lógica nebulosa. Sua base teórica fundamenta os capítulos seguintes ao apresentar os principais aspectos envolvidos na construção de sistemas de inferências nebulosos, seus conceitos, aplicações e limitações.

3.1 Fundamentos de Lógica Nebulosa

Naturalmente, o raciocínio humano é capaz de lidar com processos complexos, processando informações imprecisas e aproximadas, conforme a maneira como representa sua abstração pessoal sobre o meio que o cerca. Ao descrever seu conhecimento tácito, utilizam-se expressões em termos lingüísticos para representar estas informações aproximadas nas estratégias criadas para a resolução de determinados problemas. Portanto, o ser humano opera a partir de representações subjetivas para abstrair as informações imprecisas, como os termos “aproximadamente”, “rápido” e “devagar”, ou “alto” e “baixo”.

A teoria dos conjuntos nebulosos, ou conjuntos *fuzzy*, foi inicialmente apresentada pelo matemático Lofti A. Zadeh (1965), com o principal objetivo de oferecer representações matemáticas para traduzir estes termos aproximados, ao expressá-las em um conjunto de regras linguísticas. Se um operador humano for capaz de articular sua estratégia de ação como um conjunto de regras da forma “se-então”, um algoritmo passível de ser implementado em computador pode ser construído. O resultado é um sistema de inferência baseado em regras, no qual a Teoria de Conjuntos *Fuzzy* e Lógica *Fuzzy* fornecem o ferramental matemático para se lidar com as tais regras linguísticas (TANSCHKEIT & SCHARF, 1990).

Embora inicialmente tenha provocado muita controvérsia, a teoria dos conjuntos *fuzzy* foi ganhando popularidade, especialmente no campo de sistemas controladores, à medida que o modelo lógico clássico se mostrava incapaz de modelar sistemas reais mais complexos, que demandavam forte tratamento para informações imprecisas.

Neste contexto, Zadeh (1979) formulou o princípio da incompatibilidade, assim definido: *“na medida em que a complexidade de um sistema aumenta, nossa habilidade para fazer afirmações precisas e que sejam significativas acerca deste sistema diminui até que um limiar é atingido, além do qual precisão e significância (ou relevância) tornam-se quase que características mutuamente exclusivas”*.

A lógica nebulosa, ou lógica *fuzzy*, de uma forma geral, é um modelo capaz de generalizar a lógica clássica binária de forma a ser aproveitada em situações em que há um alto grau de incerteza (DEMASI, 2003). Estes conceitos permitiriam o cálculo e

tratamento matemático de informações incertas, viabilizando uma melhor representação de problemas de natureza inerentemente imprecisa.

As técnicas de modelagem *fuzzy* permitem manusear informações qualitativas de maneira rigorosa, levando em consideração a maneira como a falta de exatidão e a incerteza são descritas, permitindo que o especialista descreva o seu conhecimento de forma conveniente. Ao considerar processos complexos, representados de maneira simples, as técnicas *fuzzy* permitem reduzir a complexidade de um projeto a um ponto em que problemas anteriormente intratáveis possam ser agora solucionados, na forma de sistemas de controle que proporcionem resultados acurados, com um desempenho estável e robusto (GOMIDE, 1995).

3.2 Lógica Nebulosa x Lógica Clássica

Tradicionalmente, uma proposição em lógica clássica possui dois extremos: ou a premissa é completamente verdadeira, ou é completamente falsa. Só existe um resultado possível. Entretanto, em lógica nebulosa, o resultado desta proposição poderá variar em um grau de pertinência, que poderá considerá-la como parcialmente verdadeira ou parcialmente falsa.

Neste sentido, a lógica *fuzzy* permite ver os “graus de verdade” que definem o ser e o não ser, como tons de cinza entre o preto e o branco, caracterizando, assim, uma generalização da lógica aritotélica (MORATORI, 2006). A Figura 3.1 ilustra uma representação estilizada de um comparativo entre os graus de pertinência da lógica clássica e da lógica nebulosa.



Figura 3.1: Ilustração com representação conceitual da pertinência entre conjuntos clássicos e as infinitas possibilidades tratadas pela lógica nebulosa.

Na teoria clássica de conjuntos, o conceito de pertinência fica bem definido. Dado um conjunto A em um universo X , os elementos deste universo simplesmente pertencem ou não àquele conjunto (TANSCHKEIT & SCHARF, 1990).

Segundo a teoria de lógica nebulosa, um conjunto *fuzzy* A em um universo X é definido por uma função de pertinência de forma que ela possa assumir um número infinito de valores no intervalo $[0-1]$. Isto implica dizer que um determinado elemento pode pertencer a mais de um conjunto *fuzzy*, em diferentes graus de pertinência.

3.3 Variáveis e Conjuntos Nebulosos

Na lógica clássica, a teoria dos conjuntos define o conceito de pertinência de um elemento a um determinado conjunto de maneira muito precisa, ou seja, é possível delimitar funções de avaliação para identificar se um elemento pertence ou não a um conjunto. Portanto, estas relações são definidas a partir de uma função de avaliação que recebe um dado elemento x e um conjunto X e retorna 1 (verdadeiro) se este elemento de fato pertencer ao conjunto ($x \in A$), ou 0 (falso) caso contrário ($x \notin A$). Isto pode ser expresso em termos da seguinte função característica:

$$f_A(x) = \begin{cases} 1 & \text{se e somente se } x \in A \\ 0 & \text{se e somente se } x \notin A \end{cases}$$

Equação 3.1: Função de pertinência de conjuntos clássicos

Segundo a função característica, um conjunto clássico define um limiar a partir do qual um determinado elemento deixa de participar de um conjunto. Neste contexto, poderíamos propor como exemplo didático, a separação dos conjuntos de indivíduos em função de sua idade, classificando-os pelos seguintes conjuntos: ‘*Criança*’, ‘*Jovem*’, ‘*Adulto*’ e ‘*Idoso*’. Note pelo exemplo a transição abrupta da pertinência dos elementos entre os conjuntos. No caso, um indivíduo que possua 19 anos e 11 meses é considerado jovem, mas ao completar 20 anos é diretamente reclassificado para o conjunto ‘*Adulto*’. Este tipo de representação ilustra didaticamente as limitações da matemática clássica na representação de situações reais, onde as fronteiras entre os conjuntos não ocorrem de maneira tão rígida, ocasionando eventualmente perdas de informações na avaliação do problema.

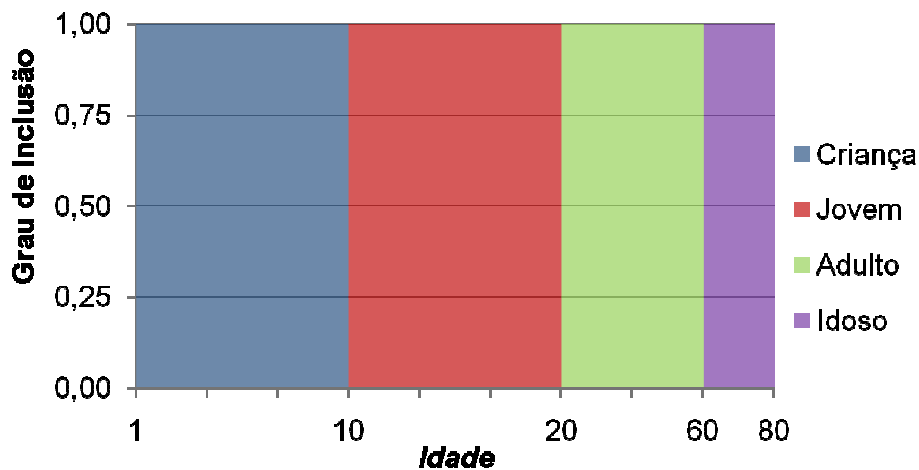


Figura 3.2 Gráfico ilustrando conjunto clássico, representando a transição abrupta da pertinência de elementos ente os conjuntos

A teoria dos conjuntos *fuzzy* pode ser considerada como uma extensão da teoria dos conjuntos clássica, onde se propõe uma caracterização mais ampla, generalizando a função de pertinência de modo que ela possa assumir um número infinito de valores no intervalo $[0,1]$ (GOMIDE, 1995). Ou seja, ao invés de termos 0 (falso) ou 1 (verdadeiro) como retorno da função de pertinência, qualquer valor entre 0 e 1 será possível. Portanto, um conjunto *fuzzy* é definido por uma função de pertinência conforme descrito na Equação 3.3, e representado por um conjunto de pares ordenados:

$$A = \{\mu_A(x) / x\} | x \in X$$

Equação 3.2: grau de pertinência de um elemento x a um conjunto X

onde μ_A é a chamada função de pertinência e $\mu_A(x)$ é o grau de pertinência de x em A . Um determinado elemento pode pertencer a mais de um conjunto, em diferentes graus de pertinência.

$$\mu_A(x) : X \rightarrow [0,1]$$

Equação 3.3: função de pertinência de um conjunto X

Em suma, podemos descrever em linhas gerais a função de pertinência μ_A , que define o grau de inclusão de um elemento x ao conjunto A , segundo as seguintes condições (MORATORI, 2006):

- $\mu_A(x) = 1$, quando x é completamente compatível com o conjunto A ;

- $\mu_A(x)=0$, quando x é completamente incompatível com o conjunto A ;
- $0 < \mu_A(x) < 1$ quando x é parcialmente compatível com A , com o grau $\mu_A(x)$.

Desta forma, podemos retornar ao exemplo da classificação etária de indivíduos, utilizando agora a notação dos conjuntos *fuzzy*. Destaca-se que neste caso um indivíduo com 19 anos e 11 meses pertencem tanto ao conjunto ‘Jovem’ quanto ao conjunto ‘Adulto’, porém em diferentes graus de pertinência. As fronteiras entre os conjuntos não são nitidamente definidos, havendo certa interseção sobre elas e, por isso, ocorre uma transição gradativa dos graus de pertinência entre os conjuntos à medida que a idade do indivíduo avança. Portanto, podemos definir que, neste exemplo, o indivíduo pertence aos conjuntos ‘Jovem’ e ‘Adulto’ em diferentes graus de inclusão, mas não pertence aos conjuntos ‘Criança’ e ‘Idoso’, pois nestes dois últimos casos o grau de pertinência é zero.

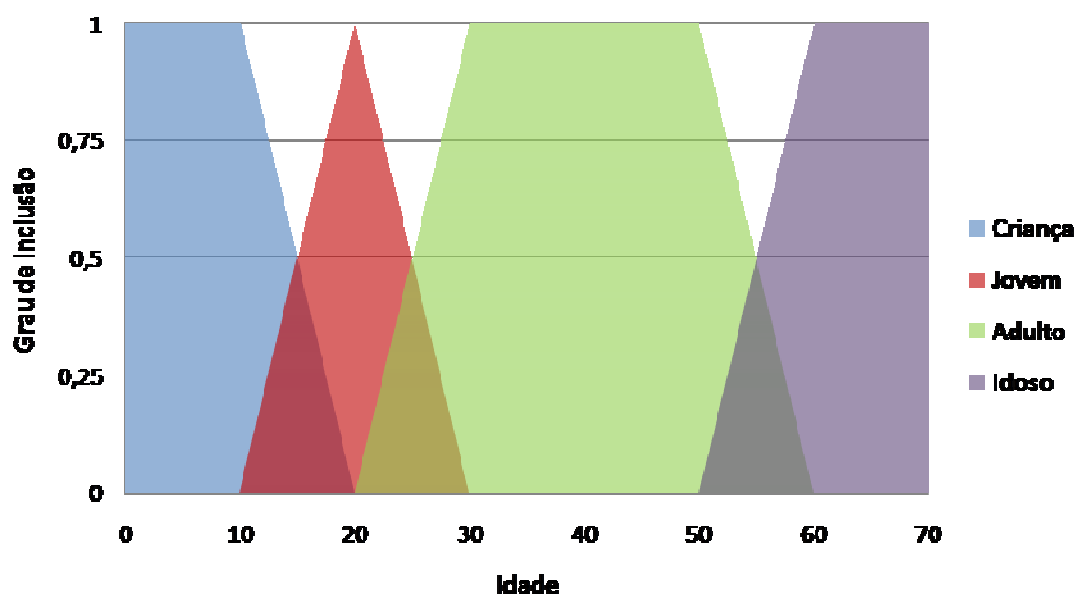


Figura 3.3 Gráfico representando a teoria dos conjuntos fuzzy, ilustrando o grau de pertinência de elementos entre os conjuntos

Esta representação permite identificar os elementos que são mais representativos da idéia geral dos conjuntos. Ou seja, quanto mais próximo o valor $\mu_A(x)$ estiver próximo de 1, maior o grau de representatividade do termo linguístico aplicado (MORATORI, 2006). Isto significa que um elemento pode ser membro apenas parcialmente de um conjunto.

Neste ponto, cabe ressaltar alguns conceitos empregados nos exemplos supracitados. No exemplo da classificação etária de indivíduos, o universo de discurso foi subdividido em conjuntos nebulosos, e para cada um destes conjuntos foi atribuído um rótulo de identificação. No caso, o universo de discurso é o que caracteriza o domínio de possibilidades para a idade do indivíduo, e os rótulos foram nomeados como ‘Criança’, ‘Jovem’, ‘Adulto’ e ‘Idoso’. Neste contexto, definimos o termo variável nebulosa, ou variável linguística, para designar a abstração matemática cujos possíveis valores são representados pelos conjuntos *fuzzy*, nomeados por cada um dos rótulos e representados por suas funções de pertinência.

A principal função das variáveis nebulosas é permitir a descrição lingüística empregada por seres humanos no lugar de variáveis quantificadas em termos matemáticos convencionais, permitindo, assim, a caracterização aproximada de valores imprecisos e o tratamento de sistemas complexos que envolvem certo grau de incerteza.

3.4 Funções de Pertinência

As funções de pertinência representam uma das principais características na modelagem de sistemas *fuzzy*, e definem como o universo de discurso será dividido em seus conjuntos nebulosos. Estas funções podem assumir diversas formas e diferentes valores de intervalos, dependendo do contexto empregado e da estratégia de modelagem assumida.

É relevante destacar que a modelagem da função de inclusão está diretamente ligada com a estratégia e com o ponto de vista da pessoa que a modela, pois representa a visão subjetiva sobre como uma pessoa entende um determinado problema. Para exemplificar o quanto estas funções podem variar de acordo com o contexto, podemos assumir o clássico problema da classificação de indivíduos conforme sua estatura. Tomemos como exemplo a variável linguística estatura, subdividida nos seguintes conjuntos nebulosos: $X_{(estatura)} = \{baixa, média, alta\}$. As funções de inclusão que definirão cada um dos conjuntos podem ser feitas de diversas maneiras, conforme a percepção da pessoa que a modela, de acordo com diversas características étnicas, geográficas ou etárias. A Figura 3.4 ilustra uma configuração possível para a definição dos conjuntos nebulosos conforme a estatura do indivíduo.

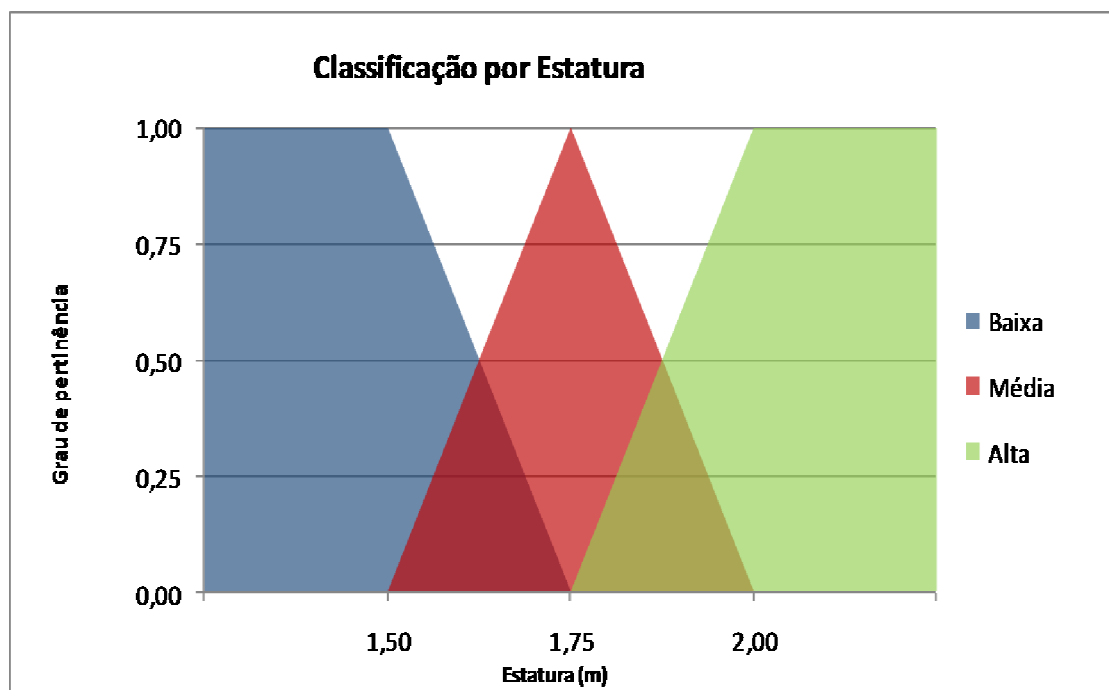


Figura 3.4: Exemplo de configuração possível para os conjuntos nebulosos por estatura

Podemos observar que o exemplo reflete apenas uma configuração possível. Neste caso, indivíduos com estatura de 1,75m são totalmente compatíveis com o conjunto de pessoas médias, enquanto pessoas com aproximadamente 1,80m pertenceriam predominantemente ao conjunto de pessoas altas. Estas configurações são compatíveis com as percepções do usuário que a modelou, mas podem não representar consenso quando analisadas por diferentes especialistas. Por exemplo, as características de biótipo da população de países como o Japão não são as mesmas de países de origem escandinava, por exemplo, tendo, portanto, intervalos distintos para definir seus conjuntos nebulosos. De maneira análoga, o mesmo se pode dizer de um sistema para controle de velocidade de um veículo, onde o senso de velocidade pode estar associado a diversos fatores, como condições da rodovia, condições do tempo e experiência do motorista. Certamente, o intervalo para o conjunto 'rápido' seria definido de maneiras diferentes quando analisado por um motorista recém-habilitado e por um profissional do automobilismo, dentro do contexto do problema analisado.

Além dos limiares das funções de pertinência, seu formato também pode assumir diferentes configurações como, por exemplo, as formas triangulares, trapezoidais e sigmoidais, entre outras. A Figura 3.5 ilustra graficamente os números *fuzzy* triangulares e trapezoidais.

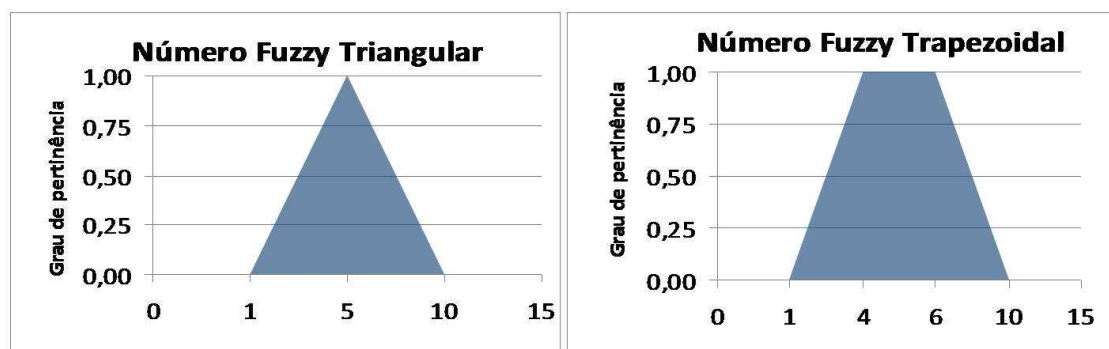


Figura 3.5: Exemplos de funções de pertinência nos formatos triangular e trapezoidal, respectivamente

Este aspecto do processo de modelagem pode demonstrar certa complexidade ao utilizar lógica nebulosa, porquanto as especificações que definem os intervalos e os formatos para as funções de pertinência podem necessitar de diversas iterações de testes, avaliações e ajustes, até que se consiga atingir um nível satisfatório nas configurações desejadas. Portanto, uma vez que cada conjunto, o formato e os limiares de suas funções de pertinência estiverem definidos, teremos então a definição completa de uma variável nebulosa.

3.5 Regras e Implicações Lógicas

Comumente, quando procuramos expressar nosso raciocínio, naturalmente desenvolvemos uma descrição que o representa em um conjunto de implicações lógicas, no formato *se* (*antecedente*), *então* (*consequente*). À medida que a descrição se desenvolve, combinamos diversas sentenças através da utilização de operadores lógicos “e” e “ou”, como na sentença a seguir: *se* (*antecedente1*) *e* (*antecedente2*), *então* (*consequente*). Por exemplo, podemos formar a seguinte proposição, considerando um sistema de controle de deslocamento de um veículo: *se* (*velocidade é alta*) *e* (*distância é perto*) *então* (*aceleração é frear forte*). Neste cenário, a partição nebulosa ‘frear forte’ assumiria intervalos negativos no universo de discurso da variável ‘aceleração’.

O conjunto de regras pode ser compreendido como o principal elemento de um sistema de inferências *fuzzy*, uma vez que representa a descrição do conhecimento de um especialista a respeito do problema a que o sistema se propõe a tratar. Desta forma, as definições das regras constituem um aspecto fundamental para o bom desempenho de

um sistema de inferências *fuzzy*, que somente será satisfatório em função da consistência das definições de sua base de regras.

O operador “*se-então*” é conhecido como declaração condicional *fuzzy*, e descreve como o valor de uma variável lingüística depende em função de outra. Uma frase deste tipo é normalmente denominada implicação, e um conjunto destas frases reunidas constituem as declarações condicionais comumente denominadas regras lingüísticas (TANSCHUIT, 2011).

Como exemplo, podemos considerar um sistema de controle para a velocidade de um veículo em deslocamento. Neste contexto, consideremos as variáveis *velocidade*=*{muito devagar, devagar, rápido, muito rápido}* e distância do ponto alvo, descrita como *distância*=*{muito perto, perto, distante, muito distante}*. Como saída do sistema de inferência, será considerada a variável *aceleração*=*{frear forte, frear, manter, acelerar, acelerar forte}*, correspondendo à ação esperada do veículo para as configurações de entrada correspondentes. A Tabela 3.1 descreve um exemplo para o conjunto de regras que descrevem o sistema de controle para este cenário.

Distância \ Velocidade	Muito perto	Perto	Distante	Muito distante
Muito devagar	Frear	Manter	Acelerar	Acelerar forte
Devagar	Frear forte	Frear	Acelerar	Acelerar forte
Rápido	Frear forte	Frear forte	Manter	Acelerar
Muito rápido	Frear forte	Frear forte	Manter	Manter

Tabela 3.1: conjunto de regras para um sistema de controle de velocidade de um veículo

O conceito de implicação está relacionado a um ramo da matemática conhecido como lógica proposicional, que é isomórfica à teoria dos conjuntos, sendo que ambas são isomórficas à álgebra booleana (TANSCHUIT, 2011).

Na lógica clássica, uma sentença formada por diversos antecedentes, conectados por operadores lógicos “e”, seu conseqüente somente será inferido se as afirmações de todos os seus antecedentes forem verdadeiras. Isto é, todas as premissas precisam ser positivas para se verificar a conclusão desta sentença. No caso de uma sentença formada pelo operador “ou”, pelo menos um dos antecedentes precisa ser verdadeiro.

Isto ocorre de outra forma em sistemas nebulosos, pois as premissas assumem graus de verdade, não precisando, portanto, serem totalmente verdadeiras para inferir a conclusão da sentença. Assim, a declaração condicional se “ x é A então y é B ” tem uma função de pertinência que mede o grau de verdade da relação de implicação entre x e y . Desta forma, as avaliações dos antecedentes podem ser manipuladas através de operações instituídas sobre a teoria dos conjuntos nebulosos, assumindo que as definições de *max* e *min*, respectivamente, constituem representações para os operadores “ou” (união) e “e” (interseção) (ZADEH, 1965). Isto reflete nos resultados do processamento da sentença, de forma que o resultado representa o grau de verdade da resposta, isto é, expressa o grau de pertinência que o conseqüente possui em relação ao conjunto que ele representa.

3.6 Sistemas de Inferência Nebulosos

Em 1977, o pesquisador Ebrahim H. Mandani aplicou a lógica fuzzy para criar um sistema baseado em regras, com o objetivo de controlar uma máquina à vapor (MANDANI, 1977).

Sistemas de inferência nebulosos buscam representar a modelagem do raciocínio humano em forma de regras, ao invés de um algoritmo explicitamente restrito a modelos matemáticos exatos. Portanto, os sistemas de inferência *fuzzy* (Fuzzy Inference Systems – FIS) objetivam gerar uma saída lógica a partir de um conjunto de entradas não necessariamente precisas. São compostos essencialmente por uma série de definições de variáveis, conjuntos nebulosos e regras, estruturadas de forma que as entradas apareçam nos antecedentes das regras, enquanto que a saída nos conseqüentes.

A arquitetura de um sistema de inferência *fuzzy* possui quatro componentes principais: um processador de entrada (*fuzzifier*), que transforma os dados de entrada precisos em representações nebulosas; uma base de conhecimento, representados pelas variáveis nebulosas, suas funções de pertinência e um conjunto de regras; uma máquina de inferência *fuzzy*, responsável pela avaliação das regras e inferência da saída; e um processador de saída (*defuzzifier*), que fornece um número real como saída da inferência. A Figura 3.6 ilustra conceitualmente a arquitetura geral de um sistema de inferências *fuzzy*.

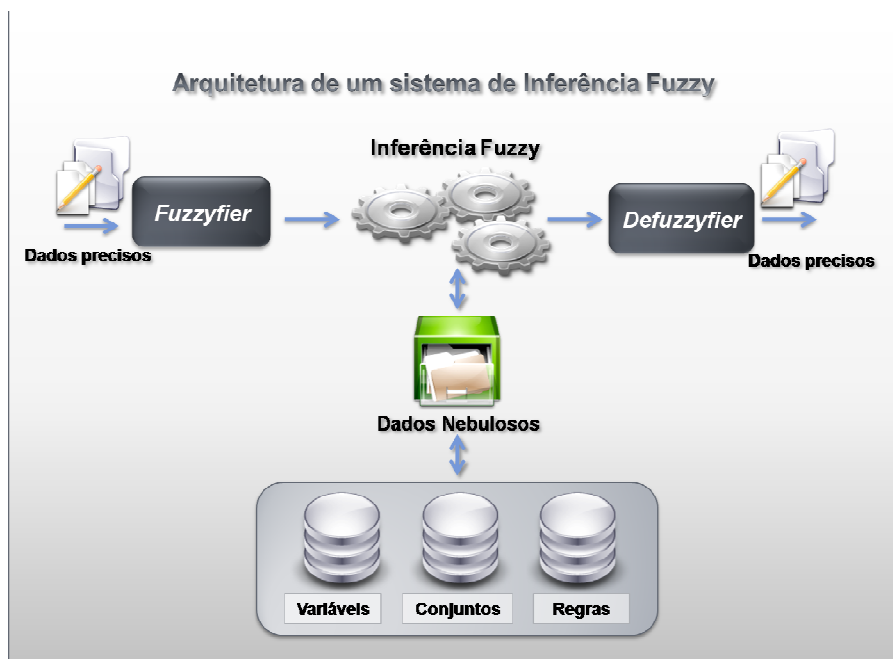


Figura 3.6: arquitetura geral de um sistema de inferências fuzzy.

Sistemas de inferência *fuzzy* traduzem em modelos matemáticos informações imprecisas expressa por um conjunto de regras lingüísticas e suas definições para as partições nebulosas. Desta forma, os dados precisos fornecidos como entradas para o sistema de inferências são transformados em dados nebulosos (*fuzzification*). Estes são processados, em função do mapeamento de regras definidas e, então, transformados novamente em dados precisos (*defuzzification*), que são retornados como saída pelo sistema de inferências.

Uma vez configurados os parâmetros de um sistema *fuzzy* – suas definições de regras, variáveis e conjuntos – o primeiro passo corresponde ao processo de nebulização (*fuzzification*). Neste processo, os valores numéricos da entrada são avaliados dentro dos conjuntos nebulosos definidos para sua respectiva variável. Isto é, conforme o exemplo descrito na Figura 3.4, um sistema que possui como entrada uma variável denominada ‘*estatura*’, deve transformar o valor fornecido, digamos 1.65m, em graus de pertinência para seus respectivos conjuntos definidos para esta variável: ‘*baixa*’, ‘*média*’ e ‘*alta*’.

De posse dos dados devidamente representados de maneira nebulosa, as definições de regras para o sistema são aplicadas. Desta forma, cada uma das regras definidas para o sistema, conforme descrito na seção ‘3.5 Regras e Implicações Lógicas’, são avaliadas de maneira a inferir a saída. Dois dos principais métodos

utilizados neste componente para implementar sistemas de inferência *fuzzy* são o Método de Mamdani (MANDANI, 1975, 1977) e o Método de Takagi-Sugeno (TAKAGI & SUGENO, 1985). A diferença básica entre eles se refere na especificação do conseqüente, no qual o método de Takagi-Sugeno o representa como uma função linear das variáveis dos antecedentes: **se x_1 é A_1 e x_2 é A_2 , então $z=f(x_1, x_2)$** . A função f é, em geral, um polinômio e o sistema de inferências é geralmente referenciado em função do grau deste polinômio (TANSCHKEIT, 2011). O Método Mamdani é o mais comumente utilizado e considera a saída da inferência como funções de pertinência que correspondam a conjuntos nebulosos, de maneira que após a avaliação de cada regra, haja um conjunto *fuzzy* a ser desnebulizado para cada variável de saída. No caso de haver mais de uma regra ativada para o contexto de entrada, comumente se utiliza o grau de pertinência mais alto (função *max*) entre as regras ativadas.

Após a avaliação das regras, têm-se os graus de pertinência atribuídos às saídas, que precisam ser novamente traduzidos da linguagem nebulosa para uma saída que possa de fato ser retornada pelo sistema, em um processo denominado desnebulização (*defuzzification*). Por exemplo, consideremos o exemplo para o sistema de controle de velocidade de um veículo, com seu conjunto de regras definidas na Tabela 3.1. Não é suficiente para o sistema de controle de velocidade que sistema de inferências retorne como saída os graus de pertinência destes conjuntos, resultantes da avaliação das regras. Estes resultados devem ser transformados em um valor preciso, representando o resultado do processo de avaliação, que será de fato utilizado pelo sistema de controle. No exemplo citado, a saída do sistema desnebulizada poderia ser, por exemplo, $-2m/s^2$, representando o valor que foi inferido pelo sistema nebuloso e que será de fato utilizado pelo sistema de controle de velocidade do veículo para realizar a ação esperada. Existem diversos métodos de defuzzificação descritos na literatura, os mais comumente empregados são: o *centro de gravidade*, em que a saída é o valor no universo que divide a área sob a curva da função de pertinência em duas partes iguais; a *média dos máximos*, onde a saída precisa é obtida tomando-se a média entre os dois elementos extremos no universo correspondente aos maiores valores da função de pertinência do conseqüente; ou simplesmente o valor de *máximo*.

3.6.1 Considerações sobre Sistemas Fuzzy

Especificar o conjunto de definições a serem utilizadas por um sistema de inferências *fuzzy*, como as funções de pertinência das variáveis envolvidas e seu conjunto de regras, pode não ser uma tarefa simples, mesmo que o especialista detenha todo o conhecimento acerca do problema em questão.

Inicialmente, pode-se considerar o problema do potencial crescimento da base de regras, uma vez que o número de regras possíveis é função direta do número de conjuntos *fuzzy* identificados para cada antecedente. Por exemplo, um sistema composto por duas variáveis de entrada, cada uma delas compostas por 5 funções de pertinência, resultaria em uma combinação de $5 \times 5 = 25$ regras. Portanto, quanto maior o número de conjuntos nebulosos atribuídos às variáveis de entrada, maior a complexidade em se estabelecer uma base de regras consistente.

Comumente, os sistemas de inferência são calibrados e melhorados de maneira iterativa, em diversas sucessões de testes e verificações, adequando e ajustando as funções de pertinência e a base de regras a fim de se obter melhores resultados. Alternativamente, existem métodos automatizados para extração de regras a partir de bases de conhecimento, que podem ser particularmente úteis em funções de classificação e previsão de séries temporais. Estes métodos empregam técnicas como redes neurais, neuro-fuzzy e algoritmos genéticos, tanto para gerar a base de regras como para ajustar as funções de pertinência. Demasi e Cruz (2003) demonstram a experimentação de técnicas de auto-aprendizado para fornecer comportamento adaptativo ao algoritmo inteligente, ajustando a base de regras conforme o aprendizado do sistema de inferências em tempo real.

Uma das principais vantagens dos sistemas de inferência *fuzzy* são sua estabilidade e robustez na modelagem de problemas complexos. Moratori et al (2005, 2006) apresentam resultados sobre experimentos que avaliam a utilização de sistemas de inferência *fuzzy*, descrevendo-os como robustos, concluindo que grande parte dos resultados ainda são obtidos corretamente mesmo quando há parte das regras retiradas ou incorretas. Segundo Moratori et al, sua simplicidade, facilidade de construção e sua robustez sugerem que é possível utilizar sistemas fuzzy bastante precisos sob condições reais.

Algumas vantagens de sistemas fuzzy:

- Fáceis de serem descritos e modificáveis, por serem baseados em regras representadas por termos lingüísticos simples, em forma de sentenças causais lógicas;
- Permite a modelagem de problemas complexos, que lidem com fatores de incerteza e imprecisão;
- Extensíveis, em função de permitir a adição, edição ou exclusão de regras por especialistas;
- Robusta e eficiente, funcionando corretamente mesmo contendo ruídos ou regras faltantes;

Em contrapartida, a utilização de sistemas *fuzzy* apresenta também algumas desvantagens, que podem ser descritas segundo a listagem abaixo:

- Necessitam de diversas simulações e testes, para sintonizar os parâmetros do algoritmo;
- Potencial crescimento da base de regras, à medida que novas variáveis e partições nebulosas são adicionados;
- Dificuldade em se estabelecer corretamente os parâmetros para as partições nebulosas;

Embora sistemas de inferência fuzzy facilitem a descrição do conhecimento de especialistas em uma linguagem simples, similar à sua forma natural de raciocínio, uma de suas principais desvantagens está relacionada à complexidade de utilização de softwares disponíveis no mercado, que são produzidos visando atender a uma vasta diversidade de áreas de aplicação, especialmente as engenharias e demais ciências exatas, e são complexas para serem utilizadas pelos usuários finais, que são de fato os especialistas nos problemas a serem resolvidos (HUDEC, 2011). Um dos principais requisitos para a correta modelagem de um determinado problema em um FIS é a participação direta do especialista e, portanto, a simplicidade para sua descrição é essencial. Desta forma, a utilização de softwares disponíveis de mercado não permite a fácil integração com bases de dados dos sistemas de controle, além de exigirem a

assistência de analistas em sistemas de informação para orientar na utilização dos softwares e para prepararem os dados de entrada a serem utilizados pelo FIS. Esta pesquisa se concentra a atuar neste problema, cuja proposta de solução encontra-se descrita no Capítulo V, atuando na disponibilização de uma interface simplificada e integrada para descrição e testes de sistemas *fuzzy*, permitindo sua utilização pelos diversos especialistas participantes das equipes multidisciplinares envolvidas.

Capítulo IV

Lógica Nebulosa em Jogos didáticos

"Não se imagina como tudo é vago, até que se tente fazê-lo de maneira precisa."

Bertrand Russel

A fim de contextualizar o cenário de utilização de recursos computacionais inteligentes em jogos didáticos, abrangendo especialmente os algoritmos de inferência nebulosos, serão analisados os trabalhos relacionados na área de cognição disponíveis na literatura, e serão discutidos os métodos mais comumente utilizados para modelar os algoritmos.

A utilização de técnicas de inteligência computacional em jogos eletrônicos é relativamente antiga, e alguns métodos tradicionais vêm sendo utilizados há bastante tempo para implementar a inteligência dos jogos (DEMASI, 2003).

O objetivo principal deste capítulo, portanto, é apresentar o cenário atual de utilização de técnicas de inteligência computacional nos jogos didáticos, especialmente lógica nebulosa, e analisar como esta inteligência é agregada aos jogos, sob a ótica da engenharia de software. Especialmente, será analisada a aplicação de sistemas de inferência *fuzzy* nos jogos e serão descritos os aspectos dos principais problemas relacionados à modelagem dos algoritmos e ao ciclo de construção dos jogos.

4.1 Visão Geral

Jogos e diversão fazem parte do crescimento humano. A criança que joga desenvolve a percepção, inteligência, sua tendência à experimentação e seus instintos sociais (PIAGET, 1988). Piaget afirma ainda que o jogo tem uma relação estreita com a construção da inteligência e possui uma efetiva influência como instrumento incentivador e motivador no processo de ensino e aprendizagem

Em especial, considerando-se o apelo inerentemente associado à tarefa de jogar, os jogos desempenham um importante fator motivador para sua utilização em projetos de pesquisa acadêmicos e na realização de testes e avaliações.

Neste cenário, destaca-se a utilização de jogos eletrônicos na aplicação de processos educacionais, atuando como mais uma ferramenta à disposição do educador para promover o desenvolvimento cognitivo, nas mais diversas áreas de aplicação. Tordeschini et al (2000) afirmam que os jogos desenvolvem a percepção, a inteligência, a curiosidade e estimulam o afloramento de instintos sociais.

Portanto, a utilização de recursos informatizados como aliada nas metodologias educacionais abre novas possibilidades para o desenvolvimento de estratégias pedagógicas mais eficientes, como a utilização de ambientes virtuais na aplicação de jogos didáticos para avaliação e reabilitação cognitiva. Sobretudo, utilizando-se dos recentes avanços na tecnologia, especialmente considerando-se a popularização do acesso a recursos informatizados, um vasto ramo de possibilidades se abre com a construção de jogos em sistemas computacionais.

Especialmente, o uso de recursos computacionais nos jogos permite aumentar a eficiência das metodologias pedagógicas designadas para o propósito do jogo em questão, à medida que permite a automatização da coleta e processamento de diversas informações coletadas durante a sessão de jogo. Se comparado com os procedimentos realizados manualmente, a aplicação de jogos assistida por computador possui inúmeras vantagens: permite a repetição de exercícios, realizados nos mesmos cenários; possibilita sistematizar e alterar o nível de dificuldade de acordo com o desempenho do usuário; disponibiliza os resultados imediatamente e, em alguns casos, em tempo real; atua como fator motivador, tendo ainda, no caso de programas mais sofisticados, a

possibilidade de dar imediato feedback, individualizado para cada tipo de resposta (BURDA,1994).

4.2 Inteligência Computacional em Jogos Eletrônicos

Ao pensar em jogos eletrônicos, não se pode dissociar a imediata relação existente com a área de inteligência computacional. Uma das principais características dos jogos eletrônicos é sua habilidade em lidar com os eventos de interação do usuário, respondendo de maneira natural, à semelhança que houvesse uma inteligência controlando estes eventos. Esta capacidade dos jogos é responsável por suas características motivadoras e de imersão que promovem no usuário.

Para isto, diversas técnicas de inteligência computacional podem ser empregadas, conforme o propósito que o jogo se propõe a atender, como Lógica Fuzzy e Redes Neurais. De toda forma, existe uma ampla diversidade de possibilidades nesta área de atuação, onde diversos métodos de inteligência computacional podem ser agregados aos jogos (DEMASI, 2003b; RABIN, 2002) e, portanto, analisá-los não é tarefa trivial, e não será objeto de estudo deste trabalho.

Neste contexto, este trabalho se concentrará em analisar a utilização de técnicas de lógica nebulosa para fornecer os recursos computacionais inteligentes presentes nos jogos eletrônicos, considerando especialmente os cenários de jogos didáticos.

A utilização de Lógica Nebulosa para modelagem de diversos aspectos existentes em cenários de aplicação para jogos eletrônicos destaca-se por sua capacidade de expressar as imprecisões e incertezas do conhecimento nele representado, permitindo a elaboração de modelos mais próximos da realidade. Segundo Demasi e Cruz (2003), sua versatilidade faz da Lógica *Fuzzy* uma excelente opção para aplicações que têm certo grau de incerteza, ou que precisem de grande flexibilidade e capacidade de adaptação.

Outra grande vantagem da utilização de lógica Nebulosa consiste na facilidade de representação do conhecimento em forma de regras, que podem ser expressas como sentenças gramaticais lógicas, no formato “*se-então*”. Segundo Moratori et al (2005), “a Lógica Fuzzy permite uma melhor aproximação do conhecimento do especialista,

através de representações semânticas de termos lingüísticos, permitindo definir poucas regras para trabalhar com termos imprecisos, criando uma modelagem mais próxima da realidade”.

Uma característica inerentemente presente nos jogos eletrônicos é sua tentativa de representar cenários, objetos e eventos à semelhança de situações reais, ou em contextos estilizados que sugerem uma determinada aproximação com a realidade. Ocorre que modelar este tipo de problema evidentemente esbarra com as dificuldades de se representar as imprecisões e incertezas características deste cenário. Portanto, os conceitos de lógica *fuzzy* podem ser utilizados para traduzir em termos matemáticos a informação imprecisa expressa por um conjunto de regras lingüísticas. Se um especialista humano for capaz de articular seu conhecimento em forma de um conjunto de regras, no formato de sentenças lógicas do tipo “*se-então*”, um algoritmo pode, portanto, ser implementado (RIEDER et al, 2004).

Um cenário de aplicação de lógica nebulosa em um jogo didático é apresentado por Rieder et al (2004), com a construção de um jogo eletrônico que utiliza recursos de inteligência computacional implementados por sistemas de inferência *fuzzy*, para auxiliar professores de matemática, baseado em um sistema de tutoria inteligente. Este jogo, denominado “Mercadão GL”, visa fornecer uma ferramenta de supervisão de ensino, auxiliando o professor na escolha de abordagens pedagógicas individualizadas e adequadas a cada aluno com dificuldades de aprendizagem (Rieder et al, 2004).

O jogo do Mercadão GL utiliza um sistema de inferências *fuzzy* baseado em regras para descrever o funcionamento do sistema de tutor inteligente (STI), que auxiliam o professor na aplicação do processo de ensino-aprendizagem. De acordo com o mapeamento das regras definidas, o STI define a ação a ser tomada para um determinado conjunto de ações do usuário, dentro do contexto do cenário em questão. Com isto, o autor afirma que através dos mecanismos de inferência *fuzzy* e do trabalho do STI, pretende-se mapear as ações do usuário durante a interação com o ambiente lúdico, buscando formular uma abordagem pedagógica personalizada, diagnosticando suas principais dificuldades de aprendizagem (Rieder et al, 2004). Observa-se que há um trabalho prévio de identificação, mapeamento e modelagem das regras que serão utilizadas para descrever a lógica que define o comportamento do sistema tutor inteligente.

Outro trabalho semelhante é descrito por Demasi e Cruz (2002), em que um jogo de naves espaciais é modelado utilizando-se técnicas de lógica *fuzzy*. O trabalho objetivou analisar as facilidades da modelagem usando lógica *fuzzy*, beneficiando-se de sua capacidade de adaptação e flexibilidade, e analisar a construção de regras para a representação de conhecimento no formato de descrições linguísticas. No trabalho, foram utilizadas entre 30 a 40 regras para analisar o comportamento da nave espacial, variando-se as definições das regras e verificando as alterações no comportamento da nave. O trabalho concluiu pela flexibilidade da utilização da técnica e sua capacidade de adaptação e manipulação, especialmente porque as regras foram implementadas separadamente do jogo, o que permitiu maior versatilidade na modificação das regras e execução dos testes. A facilidade da interpretação das regras, representadas por definições linguísticas, também contribuiu positivamente para a execução do projeto.

Embora os resultados dos trabalhos anteriores tenham sido positivos, destaca-se certa complexidade na identificação e modelagem das regras que definem o algoritmo de inteligência. Portanto, uma técnica que vem ganhando destaque na área de inteligência computacional está relacionada ao auto-aprendizado de regras. Em outro trabalho complementar, Demasi e Cruz (2003) experimentam técnicas de auto-aprendizado para fornecer comportamento adaptativo ao algoritmo inteligente.

Neste contexto, Moratori et al (2005) apresentam um estudo sobre a construção de sistemas adaptativos inteligentes aplicados à neuropsicologia cognitiva, utilizando-se de técnicas de Lógica *Fuzzy*.

Observa-se, portanto, que a construção de jogos didáticos que se utilizem de sistemas de inferência *fuzzy* baseado em regras depende fundamentalmente da pré-definição dos parâmetros que definirão o comportamento do algoritmo inteligente, assim como se beneficiam de ferramentas que auxiliem no processo de construção e adaptação destes parâmetros (Demasi e Cruz, 2003). Uma vez que os algoritmos precisam ser testados e seus parâmetros precisam ser ajustados, é importante a separação das definições do algoritmo de inferências da construção do jogo em si.

4.3 Descrição do Problema: a modelagem do algoritmo de inteligência computacional

Durante o processo de construção dos jogos eletrônicos, há uma etapa de modelagem do sistema de inferências nebuloso, em que são necessárias algumas definições para os parâmetros que comporão o sistema. Estes parâmetros podem ser considerados sob dois aspectos: os parâmetros fixos, ou aqui denominados **parâmetros estruturais**, que são aqueles identificados e configurados dentro das condições normais de operação; e os parâmetros alteráveis, identificados como **parâmetros de sintonização**, que frequentemente sofrem ajustes enquanto o sistema de inferências está sendo projetado (GOMIDE, 1995).

Portanto, para caracterização do problema, consideremos os principais componentes que compõem um sistema de inferências, assim agrupados:

- **Parâmetros estruturais:**
 - variáveis lingüísticas;
 - número de variáveis de entrada;
 - identificação dos conjuntos nebulosos para cada variável;
 - determinação das funções de pertinência para cada variável;
 - estrutura da base de regras;
 - conjunto básico de regras.
- **Parâmetros de sintonização:**
 - universo de discurso das variáveis;
 - parâmetros das funções de pertinência (intervalos, interseções, formato);
 - definições da base de regras e conectivos

Claramente, certas propriedades que definem o sistema de inferências precisam ser testadas e analisadas, especialmente aquelas que se referem aos parâmetros de sintonização, como as relacionadas às definições das funções de pertinência e ao conjunto de regras, sua completude, consistência e robustez. Em função da flexibilidade

decorrente da existência de diversos parâmetros, esta etapa de sintonização pode ser extremamente custosa, pois envolve diversas iterações de testes, análises e ajustes.

Em grande parte dos trabalhos relacionados, identificou-se como característica comum que os jogos são projetados contendo o algoritmo do sistema de inferências rígido e pré-estabelecido, construídos para atender a um determinado cenário específico de utilização, conforme apresentado na Seção 4.2. Em outras palavras, as regras de inferência são modeladas e codificadas no próprio software do jogo. Embora algumas abordagens diferenciadas para adaptação dinâmica do algoritmo de inferência possam ser encontradas em pesquisas de técnicas com auto-aprendizado, como apresentado por Demasi e Cruz (2003), este cenário onde o algoritmo inteligente faz parte do jogo é o mais comum.

Disto decorre que, para cada jogo a ser criado, é necessário construir e agregar ao jogo o algoritmo de inferência que irá lhe fornecer os recursos de inteligência computacional requeridos. Isto implica em grande complexidade, especialmente se considerando a interdisciplinaridade de áreas envolvidas neste processo de conhecimento, necessitando a interação de profissionais de informática para extrair, documentar e implementar o conhecimento descrito pelo especialista em neuropedagogia responsável pela função cognitiva à qual o jogo se propõe a avaliar. Adicionalmente, os algoritmos inteligentes precisam de diversos ajustes e revisões, realizados em simulações iterativas, até que atinjam um grau satisfatório de precisão, o que pode ser trabalhoso considerando sua implementação acoplada à codificação do jogo.

4.4 Modelagem do Problema

É desejável, portanto, uma ferramenta que permita a interação direta do especialista em neurociências, detentor do conhecimento, na modelagem e especificação das definições que serão utilizadas pelo sistema de inferências na avaliação cognitiva, permitindo-o testar e alterar estes parâmetros de uma maneira simplificada e intuitiva.

Desta forma, o objetivo deste trabalho é apresentar a proposta de construção de uma ferramenta integrada para permitir a modelagem dos algoritmos de inferência de forma desacoplada da construção dos jogos. Ao disponibilizar uma interface simples e intuitiva, objetiva-se permitir o envolvimento de toda a equipe multidisciplinar nas

definições do algoritmo, formalizando o conhecimento do especialista a partir da descrição de regras nebulosas em termos linguísticos.

Aliam-se a isto as principais vantagens da Lógica Nebulosa no que se refere à representação do conhecimento, segundo Demasi e Cruz (2002), principalmente pela facilidade que a analogia lingüística permite a quem cria os modelos. Por sua forma de expressão, descritas por regras expressas em sentenças lógicas simples e por representações de variáveis nebulosas, objetiva-se facilitar a modelagem do conhecimento do neurocientista, ao fazê-lo diretamente no sistema de inferências, sem a necessidade expressa de codificação por especialistas em computação.

Adicionalmente, obtém-se um importante benefício na construção do algoritmo inteligente, representado pela sua natureza dinâmica de modelagem, de forma que as configurações das variáveis e regras podem ser testadas, validadas e reajustadas em sucessivas iterações, dentro de um mesmo ambiente integrado, até que os resultados obtidos se tornem satisfatórios. Estes ajustes permitem calibrar a precisão do algoritmo de maneira independente do funcionamento dos jogos, utilizando-se dos módulos de configuração e de simulação fornecidos pelo sistema.

No Capítulo 5, a seguir, será descrito em detalhes a construção da ferramenta integrada Nébulas, que objetiva atender a modelagem do problema proposta nesta seção, apresentando uma abordagem sistêmica para integrar os procedimentos relacionados ao processo de construção de sistemas de inferências nebulosos.

Capítulo V

Nébula – Sistema Integrado para construção de Inferências Nebulosas

"Pensamentos valem e vivem pela observação exata ou nova, pela reflexão aguda ou profunda; não menos querem a originalidade, a simplicidade e a graça do dizer."

Machado de Assis

Este capítulo apresenta o sistema Nébula, destinado à construção de algoritmos de inferências nebulosos para serem utilizados nos jogos psicopedagógicos. Descreve-se uma visão geral do problema considerado, a proposta de solução, a descrição dos modelos arquiteturais e o processo de construção dos algoritmos, destacando a contribuição da proposta na simplificação dos processos tradicionais e nos recursos gráficos oferecidos pela interface, que contribuem para uma melhor utilização por usuários não especialistas em ciências exatas.

5.1 Visão Geral

Para implementar o modelo descrito na Seção 4.4, foi construído um sistema gerador de inferências denominado **Nébula**, objetivando fornecer um ambiente integrado que permita a participação de toda a equipe multidisciplinar no processo de construção dos algoritmos de inferência nebulosos, propondo o desacoplamento deste processo da construção dos jogos psicopedagógicos (GOMES, MOTTA, CRUZ, 2010).

O Nébula é orientado pelas premissas de simplicidade, integração e usabilidade, de maneira a facilitar o processo de especificação dos parâmetros para o desenvolvimento de sistemas de inferência, sem exigir grandes conhecimentos de computação ou de lógica nebulosa. A ferramenta objetiva fornecer uma interface intuitiva, capaz de abstrair a complexidade da representação de termos matemáticos com a utilização de gráficos e textos explicativos, conduzindo o usuário em uma sequência de passos que naturalmente orienta a descrição de seu raciocínio. Embora naturalmente alguns conceitos básicos sobre a técnica de lógica nebulosa precisem ser percebidos pelos usuários, objetiva-se que seja possível utilizar o sistema após uma breve iniciação ao tema, em que a utilização em conjunto pelos diversos perfis especialistas das equipes fará o conhecimento geral se consolidar, à medida que a ferramenta for utilizada e os resultados observados.

A partir de uma única ferramenta, os usuários podem criar, testar, analisar os resultados e realizar alterações nos algoritmos recém-criados, antes mesmo de integrá-los aos jogos, ou mesmo a qualquer momento em que se faça necessário. Mesmo após sua integração ao jogo, uma simples modificação nos parâmetros do algoritmo a partir da interface Nébula já produz efeito imediato e não exige que o jogo seja recompilado ou republicado – se o código do jogo seguir as práticas de implementação recomendadas, este não precisa nem mesmo ser reiniciado, e as alterações serão imediatamente percebidas.

A construção do sistema Nébula fundamenta-se nas definições de arquitetura propostas por Ferreira (2009) para construção de jogos neuropedagógicos integrados e demais plataformas relacionadas aos jogos, que descreve uma infraestrutura composta por diferentes visões, além de prever informações sobre repositórios, meios de comunicação e agentes computacionais. Esta arquitetura define diferentes

responsabilidades para cada uma das visões, conforme detalhado na Figura 5.1, de forma que a inteligência computacional fornecida pelos algoritmos construídos a partir do Nébula atuarão na camada de Modelo (conforme item 5 destacado na figura), definindo as regras e o comportamento do jogo. De fato, o sistema Nébula atuará como uma plataforma auxiliar para o processo de construção dos jogos, fornecendo os insumos para o desenvolvimento dos algoritmos que atuarão nesta camada de modelo, e não integrará diretamente, porém, com esta camada nos jogos.

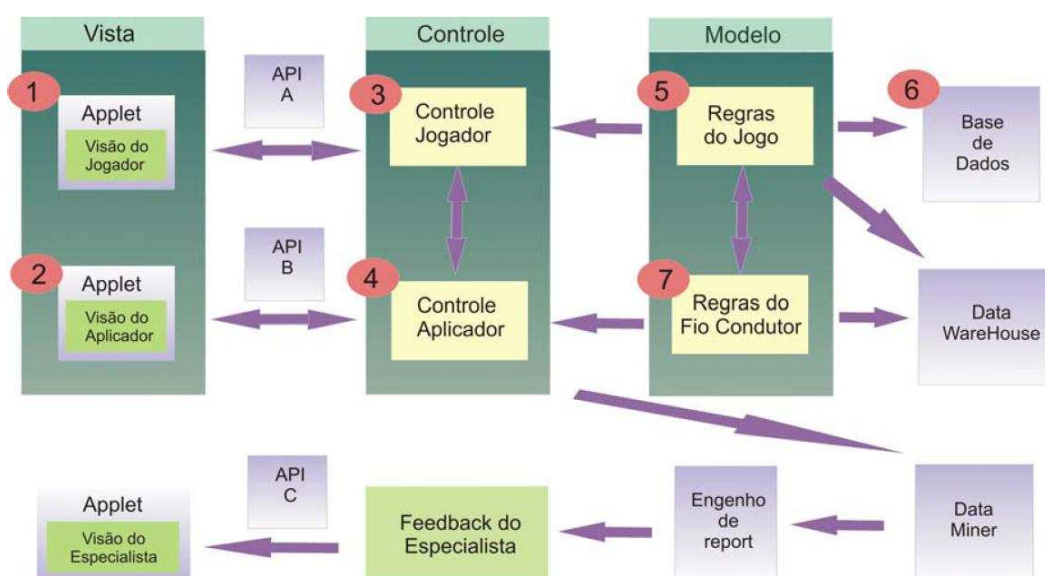


Figura 5.1: Modelo arquitetural em camadas para a construção de jogos neuropedagógicos.

Fonte: FERREIRA (2009)

Neste mesmo contexto, o sistema objetiva a facilitação do processamento das informações oriundas da base de dados integrada fornecida pela plataforma de construção de jogos neuropedagógicos Phidias (GOMES et al, 2009), cuja infraestrutura de desenvolvimento foi utilizada na construção de diversos jogos na linha de pesquisa, onde são armazenados os dados coletados durante as sessões de jogo e que poderão eventualmente ser utilizados para o processamento da inferência. A Figura 5.1 ilustra a utilização da base de dados prevista no modelo arquitetural de Ferreira (2009), destacada conforme o item 6 na figura, implementada a partir da plataforma Phidias. Entretanto, a ferramenta Nébula não está diretamente associada a nenhuma base de dados específica, sendo desacoplada e independente de qualquer fonte de origem de dados, que é especificada através de uma *Application Programming Interface* (API), tornando-a independente de soluções tecnológicas específicas. Portanto, a ferramenta

Nébula pode utilizar os dados provenientes da base de dados Phidias ou de qualquer outra fonte, como dados coletados em tempo real, ou ainda arquivos XML ou texto, desde que previamente tratados pelo *software* do jogo e fornecidos através da API especificada.

A interface gráfica do Nébula objetiva unificar todo o processo de construção dos algoritmos inteligentes, de maneira desacoplada da construção dos jogos psicopedagógicos, conferindo aos usuários a capacidade de testar e analisar os resultados dos algoritmos dentro de uma única ferramenta, realizando os ajustes necessários de maneira dinâmica para se buscar melhores resultados, de acordo com os objetivos dos especialistas envolvidos.

Portanto, a fim de atingir seus propósitos, o sistema Nébula baseia-se em algumas premissas básicas que definem as diretrizes de seus principais requisitos.

O principal objetivo do sistema é integrar o processo de construção de algoritmos de inferência nebulosos, de maneira desacoplada da codificação dos jogos psicopedagógicos, permitindo a participação das diferentes especialidades presentes nas equipes multidisciplinares, auxiliando para que possam formalizar seus conhecimentos de maneira simples e objetiva, utilizando-se de sentenças lógicas semelhantes ao seu raciocínio natural, com expressões lingüísticas causais organizadas em forma de regras. Estas definições, traduzidas em modelos matemáticos através de técnicas de lógica *fuzzy*, irão definir o comportamento do algoritmo de inferências nos diversos cenários em que este for aplicado.

Desta forma, a primeira premissa básica se refere à existência de uma interface simples e intuitiva, que conduza o usuário em seu raciocínio para a definição dos critérios de modelagem do algoritmo inteligente, permitindo a seleção de variáveis, definição de conjuntos nebulosos e descrição de regras. Com isto, objetiva-se possibilitar o envolvimento e a participação dos integrantes das equipes multidisciplinares atuando conjuntamente na construção do algoritmo, sem exigir conhecimentos aprofundados de informática e, especialmente, de lógica nebulosa. Como os resultados poderão ser imediatamente perceptíveis a partir dos testes realizados dentro da própria ferramenta, objetiva-se que os conceitos técnicos básicos envolvidos se consolidem na equipe à medida que a ferramenta for utilizada.

A segunda premissa consiste em construir um núcleo de processamento para as inferências geradas que forneça resultados precisos e confiáveis, a fim de que possam apoiar a decisão dos especialistas em seu papel na análise cognitiva. O atendimento a este critério será buscado ao permitir que o algoritmo seja testado na própria ferramenta, possibilitando que os critérios sejam revistos e reajustados em sucessivas simulações, em um módulo integrado ao sistema. Esta premissa fundamenta-se nos estudos apresentados por Moratori et al (2005) sobre a representação do conhecimento em forma de regras e sobre a robustez e confiabilidade dos sistemas de inferência fuzzy, operando em condições satisfatórias mesmo com regras incompletas. Adicionalmente, para garantir que o núcleo de processamento das inferências atue de maneira precisa, objetiva-se a realização de um estudo de caso para validação dos seus resultados, conforme apresentado no Capítulo VI.

Por fim, a terceira premissa se refere à existência de um módulo onde o especialista possa simular o comportamento do algoritmo recém-modelado, procedendo a ajustes pontuais nas definições estabelecidas e garantindo o cumprimento das premissas anteriores. Com isto, à medida que os descritores e regras são ajustados, procura-se obter maior precisão para os resultados fornecidos pelo sistema de inferências, validados antes mesmo de sua integração ao jogo.

5.2 Processo para construção de inferências

O sistema Nébula permite a construção dos algoritmos de inferências organizados sob o conceito de instância. Portanto, cada conjunto de configurações, seleção de variáveis, definição de parâmetros para os conjuntos nebulosos, funções de pertinência e base de regras, compõem uma instância de inferências. Cada instância é identificada por um nome atribuído no sistema, que será utilizado pelo software do jogo para recuperar o algoritmo criado para uma determinada finalidade. Por definição, um mesmo jogo pode rodar simultaneamente diversas instâncias de inferência, tantas quantas se fizerem necessárias, cada qual processando uma determinada característica necessária dentro do contexto do jogo. Desta forma, pode ser criada uma instância de inferências para acompanhar o nível de interação do usuário dentro do jogo, fornecendo estímulos eventuais conforme o comportamento do usuário, e outra para validar o acompanhamento semântico das regras esperadas para o jogo, por exemplo, analisando

os critérios que compõem o crivo metacognitivo. Neste mesmo exemplo, diversas instâncias de inferência poderiam ser construídas e utilizadas para modelar os diversos contextos presentes no crivo metacognitivo.

Portanto, para iniciar a construção de uma instância do algoritmo de inferências, o primeiro passo é designar um identificador que o represente. Este identificador será utilizado pelo desenvolvedor do jogo para obter, através de uma *Application Programming Interface* (API), a instância da inferência correspondente para ser utilizada no cenário desejado. Uma vez obtida a instância da inferência apropriada, o jogo pode submeter os conjuntos de dados de entrada para processamento e receber como retorno os valores processados pela inferência.

Todo o processo de construção de uma instância de inferências é realizado dentro do Módulo de Configuração, cujos passos serão descritos detalhadamente na seção 5.4.1.

Em seguida, são selecionadas quais variáveis nebulosas serão consideradas na composição desta instância de inferência. O sistema disponibiliza, inicialmente, um conjunto pré-selecionado de variáveis globais comumente utilizados em jogos psicopedagógicos, normalmente componentes da base de dados integrada da plataforma Phidias (GOMES et al, 2009), como tempo de resposta, quantidade de erros e de acertos, e alguns outros eventos de interação coletados automaticamente pelo Phidias. Além destas, o especialista pode ainda especificar sua própria variável, que será válida apenas no contexto desta inferência, conforme a necessidade para o problema a ser tratado.

Com o auxílio da interface do Nébulas, o usuário seleciona quais variáveis irão compor os conjuntos de entrada e de saída a serem informados ao sistema de inferência. Neste cenário, cabe definir o termo variável no contexto de lógica *fuzzy*, definindo-se como variável lingüística uma entidade utilizada para representar de modo impreciso e, portanto, lingüístico, um conceito ou uma variável de um determinado problema (ALMEIDA, 2005). Estas variáveis admitem como valor expressões lingüísticas (comumente chamadas de termos primários), como, por exemplo, “baixo”, “médio” ou “alto” para representar um determinado valor dentro de um intervalo, descrito por um conjunto nebuloso. Esses valores nebulosos contrastam com os valores assumidos por uma variável numérica, que admite apenas valores precisos (ou seja, números)

(ALMEIDA, 2005). Os conceitos de variáveis e conjuntos nebulosos encontram-se descritos na seção 3.3.

Após serem definidas as variáveis, devem-se especificar os termos primários associados a cada uma delas, que representam as abstrações dos conjuntos nebulosos para o intervalo numérico correspondente. Estes termos são representados por números *fuzzy*, ou também chamados intervalos *fuzzy*. Os números *fuzzy* são representados por intervalos no formato $\tilde{N}(1;5;10)$ ou $\tilde{N}(1;5;7;10)$, que correspondem aos descritores para os números *fuzzy* triangulares e trapezoidais, respectivamente. Porém, no sistema Nébola, estes intervalos são representados de maneira gráfica, abstraindo para o usuário a complexidade de suas representações matemáticas e facilitando a definição de seus termos.

Os termos primários definidos para uma determinada variável lingüística formam a sua estrutura de conhecimento, chamada de partição *fuzzy* da variável (ALMEIDA, 2005). Para exemplificar, uma variável de entrada denominada quantidade de erros poderia apresentar a seguinte partição *fuzzy*: ‘baixo’, ‘médio’ e ‘alto’. Como esta partição representa abstrações de valores numéricos, para cada um destes termos, haverá um conjunto nebuloso associado que descreverá a abstração do intervalo numérico a ele correspondente. Para facilitar o uso do sistema por neurocientistas e para facilitar a compreensão dos valores representados, o sistema irá disponibilizar representações gráficas para estes conjuntos nebulosos. A Figura 5.2 ilustra um exemplo para a representação da partição *fuzzy* possível para a variável ‘quantidade de erros’.

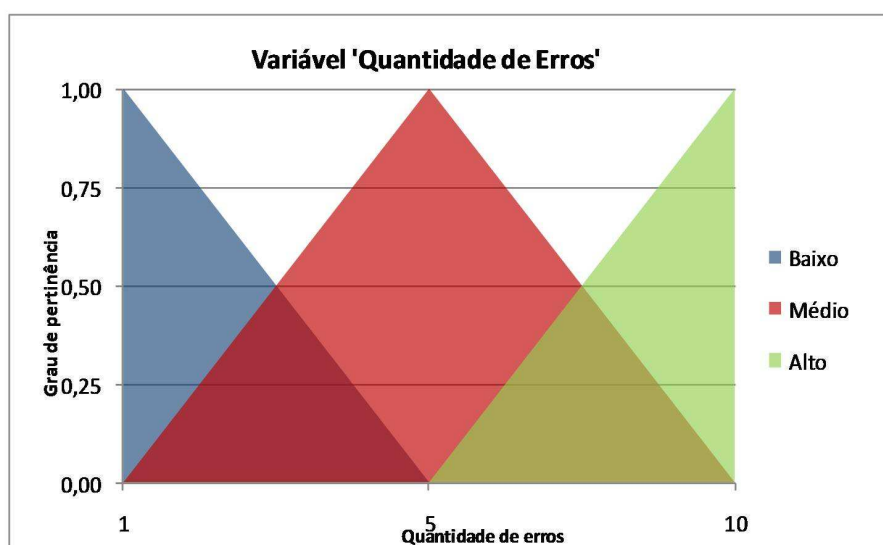


Figura 5.2: Representação gráfica de uma possível partição fuzzy para a variável ‘quantidade de erros’

Por fim, o comportamento do algoritmo de inferência no processamento das informações é definido ao se constituir o conjunto de regras lingüísticas, que correlacionam logicamente as variáveis de entrada com as variáveis de saída, em expressões semânticas causais, no formato “*se-então*”. O conhecimento do especialista é formalizado em expressões que determinam qual saída será equivalente a uma determinada combinação de condições fornecidas como entrada. Desta forma, a cada associação relevante de um quantificador para uma determinada variável com outra variável, será especificado um valor nebuloso correspondente para a saída. Os conceitos e alguns exemplos sobre regras e implicações lógicas são apresentados na seção 3.5.

Para exemplificar, considere-se um cenário hipotético em que existam as variáveis de entrada *quantidade_de_erros*={*pouco*, *médio*, *alto*}, a variável *tempo_de_resposta*={*baixo*, *médio*, *alto*} e a variável de saída *aspecto_atenção*={*ansiedade*, *concentração*}. Neste cenário, pode-se propor, apenas como exemplo didático, a seguinte regra: “**se** *quantidade_de_erros*=*alto* **e** *tempo_de_resposta*=*baixo* **então** *aspecto_atenção*=*ansiedade*”. No exemplo, a regra pode inferir a correspondência de uma característica de ansiedade identificada em um comportamento que apresenta persistência de respostas incorretas em um intervalo curto de tempo, demonstrando uma insistência do jogador na repetição do erro.

Este processo para construção dos algoritmos de inferência permite a construção de modelos para o processamento de diversos aspectos identificados durante a sessão de jogo pelos especialistas, auxiliando-os na interpretação dos resultados, na avaliação do comportamento do jogador e na inferência de informações a partir do cruzamento de características identificadas, modeladas a partir do conhecimento do especialista e descrita pela base de regras. Segundo Rieder et al (2004), “a lógica *fuzzy* pode sistematicamente traduzir os termos difusos desta comunicação humana em valores compreensíveis por computadores”.

Em suma, a configuração de uma instância de inferências no sistema é constituída, portanto, das seguintes etapas:

- Identificação da instância de inferências por um nome único;
- Seleção das variáveis de entrada e de saída;
- Definição das partições nebulosas para cada variável de entrada e de saída;
- Composição do conjunto de regras nebulosas;

- Execução e testes do sistema de inferência;
- Análise de resultados e sintonização do sistema, a partir de alterações na base de regras e nas funções de pertinência.

A Figura 5.3 ilustra sinteticamente o processo iterativo para a construção dos algoritmos de inferências com o sistema Nébula, destacando-se seu aspecto iterativo para simulações e ajustes.



Figura 5.3: Diagrama do processo de construção de inferências com o sistema Nébula

5.3 Modelo Arquitetural

A arquitetura geral do sistema Nébula foi modelada sob a concepção de módulos independentes que interagem entre si, cada um assumindo um diferente nicho de responsabilidade. Esta modelagem permite o melhor encapsulamento da lógica de negócio e reaproveitamento de código, à medida em que os módulos podem ser reaproveitados em diferentes situações.

O sistema conta com um núcleo central de processamento, o Serviço de Inferências *Fuzzy*, que é responsável por avaliar matematicamente as entradas e inferir a saída. Este módulo é utilizado pelos demais componentes do Nébula e pelos jogos psicopedagógicos, para realizar simulações e obter os resultados de inferências a partir

de um conjunto de entradas. A integração do núcleo de processamento das inferências com os demais módulos e com os jogos é realizada pelo Módulo de Inferências, que fornece interfaces de chamadas de métodos para instanciação da inferência, submissão de dados de entrada e obtenção dos valores processados.

Para isto, o módulo de inferências interage com a Base de Conhecimento da instância de inferência, onde estão definidos os parâmetros de configuração que determinam o comportamento do algoritmo de inferências. Adicionalmente, o módulo de inferências é responsável por obter e processar os dados de entrada que serão submetidos ao núcleo de inferências, que podem ser proveniente de diversas fontes de dados, como a base de dados integrada do Phidias ou de outros sistemas de banco de dados, arquivos texto ou XML, ou ainda dados coletados em tempo real, assim como obter os valores processados que serão retornados ao jogo psicopedagógico ou ao Módulo de Simulações do sistema Nébulas.

O Módulo de Simulação, por sua vez, permite a submissão pontual de valores para o núcleo de processamento da inferência, realizado por intermédio do Módulo de Inferências. Este módulo atua simulando as tarefas realizadas por um jogo psicopedagógico, onde os dados coletados são submetidos para o processamento da inferência, exibindo os valores retornados para o usuário analisar.

Para viabilizar o funcionamento de todo este processo, é necessário definir os parâmetros de configuração que compõem a Base de Conhecimento da instância de inferências. Esta tarefa é realizada pelo especialista a partir do Módulo de Configuração, que fornece uma interface gráfica simplificada para seleção das variáveis, definição das partições nebulosas e especificação do conjunto de regras.

A Figura 5.4 ilustra graficamente o modelo de arquitetura proposta para o sistema, com destaque para seus principais Módulos e suas inter-relações.

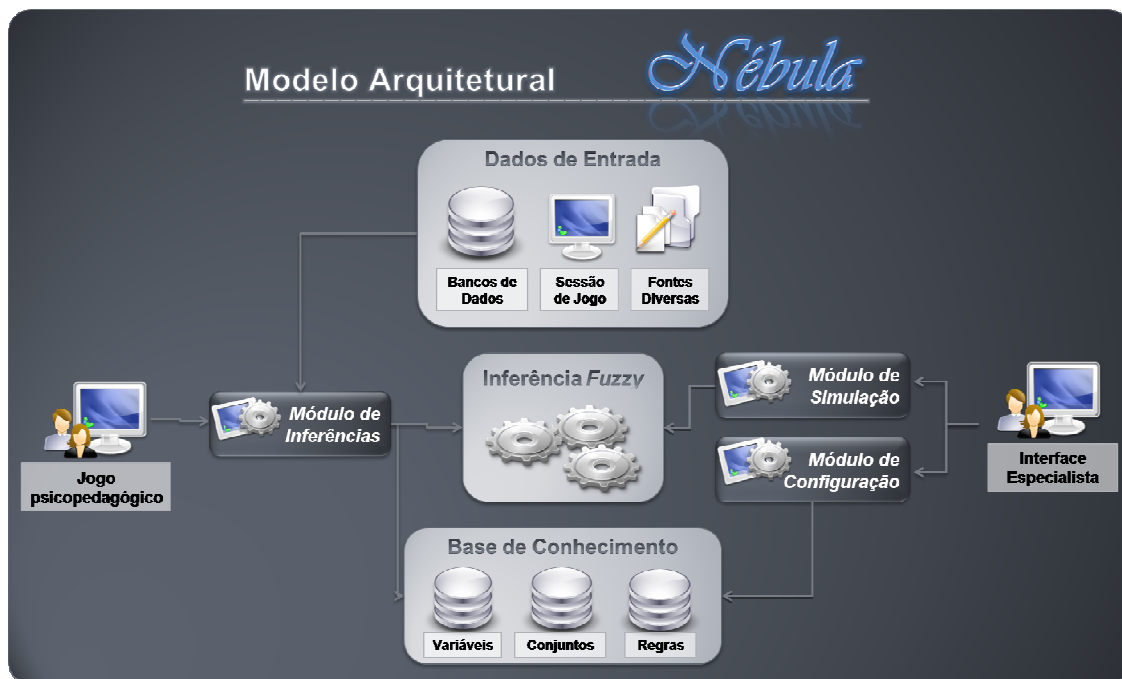


Figura 5.4: Modelo de arquitetura para o sistema Nébula

A comunicação entre os módulos é realizada a partir de componentes de software, que encapsulam um conjunto de responsabilidades, podendo ser reutilizados em diferentes situações.

Desta forma, o núcleo de processamento do serviço de inferências é disponibilizado como uma biblioteca JAR (*Java ARchive*), o que permite sua inclusão diretamente nos jogos psicopedagógicos. Esta abordagem pode ser particularmente útil para os casos onde um melhor tempo de resposta é exigido do processamento da inferência, como os casos de avaliação de dados em tempo real. Utilizando diretamente o núcleo de processamento de inferências, o software do jogo cria uma instância de inferências a partir dos parâmetros de configuração e submete os dados diretamente para a biblioteca, que irá realizar o processamento da inferência. Neste caso, todo o processamento será executado localmente, na mesma máquina onde o jogo estiver sendo executado.

Outra maneira de realizar o processamento da inferência é através da utilização de um *WebService*, que disponibiliza um conjunto de métodos simplificados para a instanciação da inferência adequada, submissão dos dados de entrada e obtenção dos valores processados. Neste caso, a comunicação dos jogos psicopedagógicos é feita a partir do protocolo HTTP, onde o sistema Nébula disponibiliza uma classe que provê

um serviço REST tanto para a camada servidor quanto para a camada Cliente, o que facilita o desenvolvedor do jogo a realizar a comunicação. Utilizando este tipo de chamada ao serviço de inferências, todo o processamento é realizado remotamente na máquina servidora, particularmente útil nos casos em que não são exigidos tempos de resposta imediatas e quando o equipamento local não possui grande poder de processamento. Adicionalmente, o desenvolvimento das chamadas aos métodos de avaliação de inferências é simplificado utilizando o serviço de inferências. Outra vantagem é permitir que, caso os parâmetros de configuração da inferência sejam alterados a partir do sistema Nébula, estas modificações possam ser imediatamente refletidas no jogo em execução, uma vez que os dados são processados diretamente no servidor em que as informações são armazenadas.

A Figura 5.5 ilustra sinteticamente os principais componentes disponibilizados pelo sistema Nébula para o processamento dos dados de inferências.

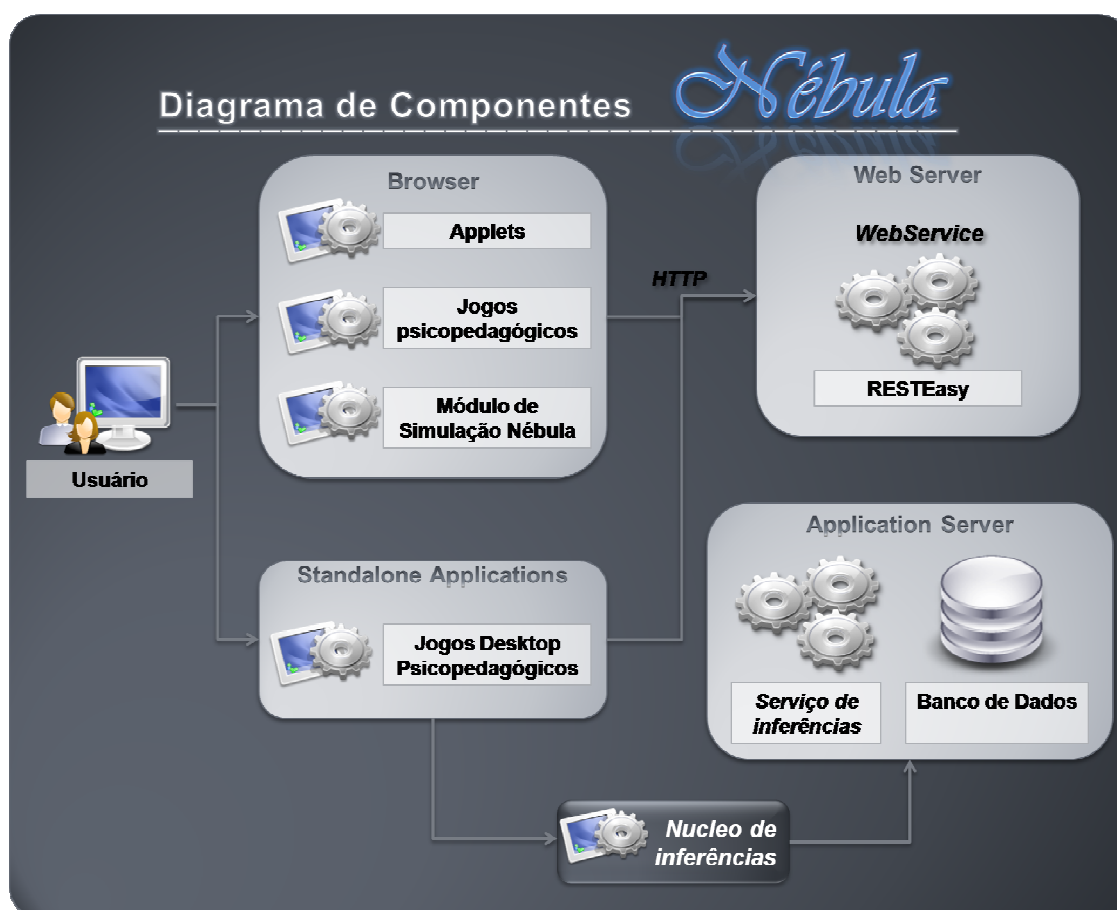


Figura 5.5: Diagrama com os principais componentes do sistema Nébula

5.4 Módulos do Sistema

O modelo de arquitetura apresentado na seção 5.3 proposto para o sistema de inferências é composto essencialmente por três módulos principais, responsáveis por fazer a interface com o usuário e com o jogo, além do serviço de inferências propriamente dito. São assim descritos:

- **Módulo de configuração:** utilizado para selecionar as variáveis, definir os conjuntos nebulosos e especificar o conjunto de regras necessárias para definir o algoritmo de inferências
- **Módulo de Simulação:** utilizado para executar pontualmente a instância de inferência recém-criada, submetendo dados de testes, permitindo realizar simulações e validações de resultados, buscando a melhor configuração para resultados mais precisos;
- **Módulo de Inferências:** acessado diretamente pelo aplicativo do jogo psicopedagógico. Atua como uma interface para a instância do Serviço de Inferências, repassando ao algoritmo as entradas fornecidas pelo jogo e retornando o resultado processado pela inferência.
- **Serviço de Inferência Fuzzy:** representa o núcleo do sistema de inferências, fornecendo os recursos computacionais inteligentes para processamento dos dados do jogo. Utiliza-se do conjunto de definições que compõem a base de conhecimento (regras, variáveis e partições nebulosas) estabelecidos no Módulo de Configuração para processar os valores fornecidos como entrada pelo jogo. Este componente implementa os algoritmos de inteligência computacional para transformar as entradas em valores nebulosos (*fuzzyfication*), processar as regras e transformar os resultados obtidos novamente em valores numéricos (*defuzzyfication*).

O usuário pode alternar entre os diferentes módulos a qualquer momento, a partir do menu superior de navegação, acionando o item Módulos. O módulo atualmente selecionado é exibido em destaque na barra superior de título, ao lado do nome do usuário, conforme ilustrado na Figura 5.6.



Figura 5.6: Menu superior de seleção de módulos, com destaque para o Módulo de Configuração atualmente selecionado na barra de títulos.

5.4.1 Módulo de Configuração

O Módulo de Configuração é voltado para utilização por parte do especialista em Neurociências e demais participantes das equipes multidisciplinares para modelar o algoritmo de inferências, a partir da definição do conjunto de parâmetros necessários para criar uma instância de inferências: seleção das variáveis envolvidas, definição das partições nebulosas e especificação do conjunto de regras nebulosas.

Este módulo foi construído com a premissa de orientar o usuário na condução do processo de formalização de seu conhecimento, disponibilizando um *design* no estilo passo-a-passo, onde o usuário é conduzido através das etapas necessárias para a construção do algoritmo de inferências, representados por itens seqüenciados no menu lateral de navegação.

As instâncias de inferências criadas pelo Nébula são agrupadas no menu de navegação lateral, organizadas na forma de projetos. Cada projeto contempla os passos necessários para a completa definição de uma instância de inferências, e é identificada por um código único, que será utilizado posteriormente para o jogo referenciar uma esta instância específica. A Figura 5.7 ilustra a organização do menu lateral de navegação e a página principal da definição da inferência.

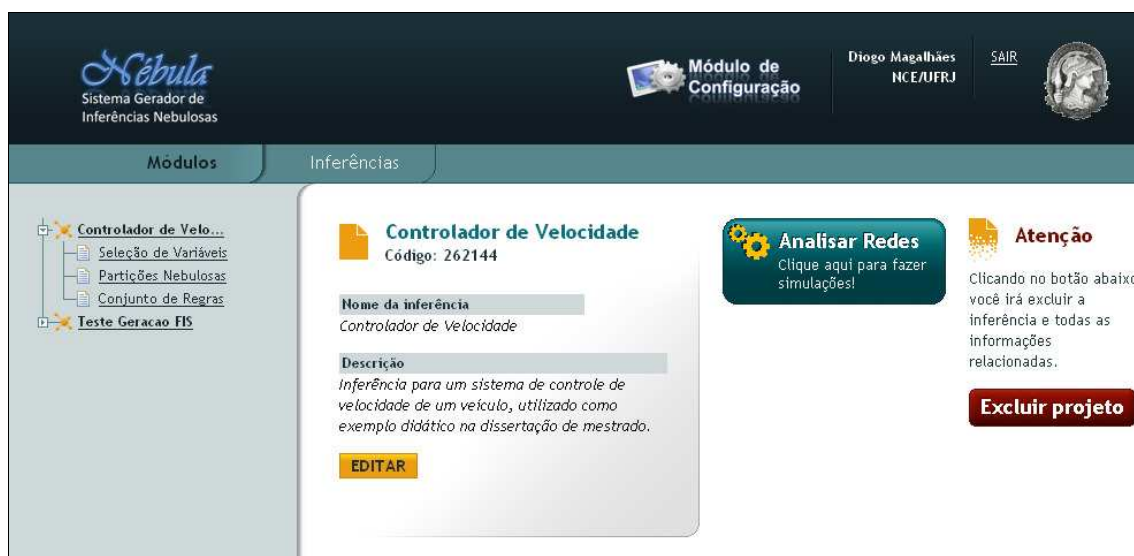


Figura 5.7: Menu lateral de navegação e página principal da inferência criada.

Nas subseções seguintes, serão apresentados cada um dos passos necessários para a complexa configuração de uma instância de inferência. Para ilustração do cenário de uso do sistema Nébula, será considerado o exemplo de um sistema controlador de velocidade de um veículo, conforme anteriormente mencionado na Seção 3.5.

5.4.1.1 Seleção de Variáveis

A seleção de variáveis é o primeiro passo no processo de construção do algoritmo. Nesta etapa, são selecionadas quais variáveis irão compor esta instância de inferências.

O usuário deve selecionar as variáveis de entrada e de saída a partir de um conjunto pré-disponibilizado de opções, composto pelas variáveis comumente utilizadas nos jogos psicopedagógicos ou, ainda, pode criar a sua própria variável de inferência, para os casos mais específicos.

A Figura 5.8 representa a interface do sistema para o passo de seleção de variáveis.

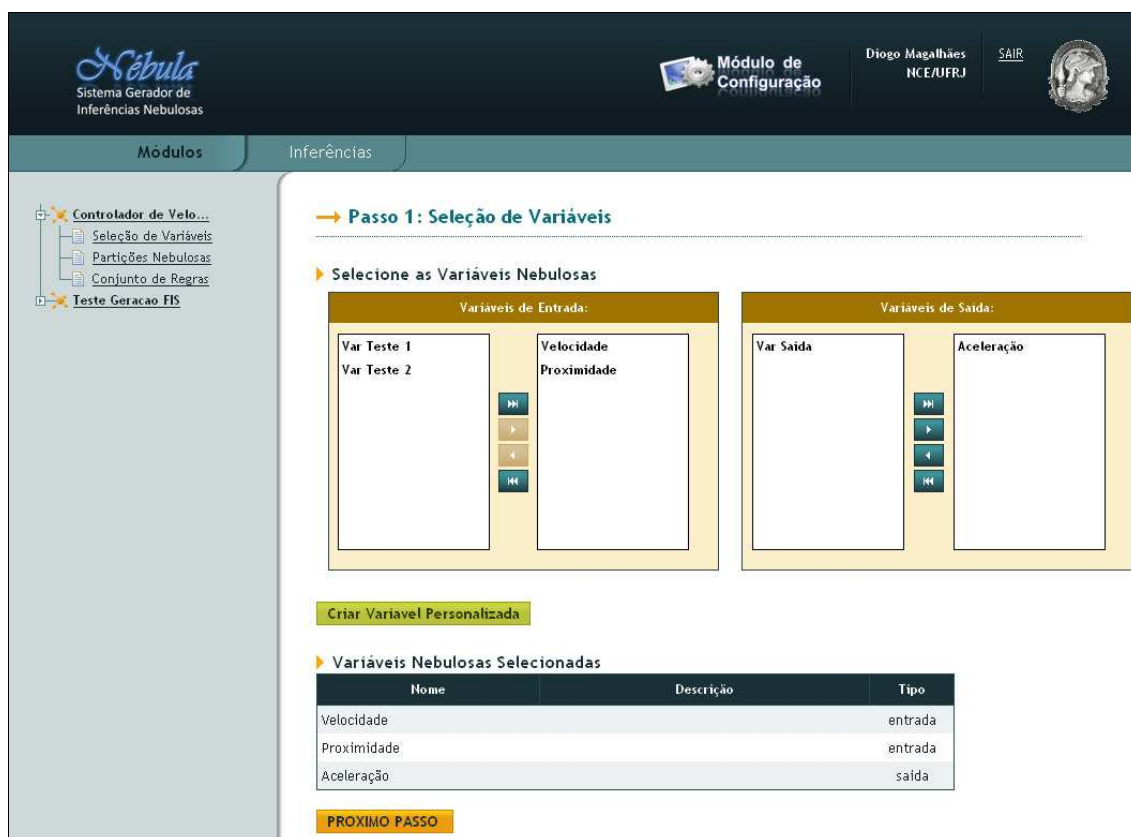


Figura 5.8: Interface do sistema para o Passo 1: Seleção de Variáveis, do Módulo de Configuração

5.4.1.2 Definição dos Conjuntos Nebulosos (*Editor de Funções de Pertinência*)

Em seguida, o segundo passo corresponde à definição dos conjuntos nebulosos associados a cada variável. Neste passo, o usuário especifica abstrações nebulosas para representar os intervalos correspondentes a cada uma das variáveis selecionadas no passo anterior. O sistema Nébula disponibiliza um editor de funções de pertinência, que auxilia o usuário na especificação dos intervalos de valores, abstraindo as representações numéricas associadas aos conjuntos *fuzzy*. Ao selecionar o tipo da partição e, conforme são informados os valores dos parâmetros para as bases e os topos, imediatamente é gerado um gráfico representando a forma atual da função de pertinência, alterando-se dinamicamente à medida que os valores são modificados, conforme ilustrado na Figura 5.9.

Criação de Nova Partição Nebulosa FECHAR

► **Configurações da Partição Nebulosa**

Nome da Partição
Muito Devagar

Descrição
Partição referente à variável de entrada 'Velocidade do Veículo'

Tipo da Partição
triangular

Base 1: 0.0 **Topo 1:** 10.0 **Base 2:** 20.0

► **Pré-Visualização da Partição**

Muito Devagar

1.00
0.75
0.50
0.25
0.00

0 2 4 6 8 10 12 14 16 18 20 22

SALVAR

Figura 5.9: Editor de funções de pertinência do Nébula.

O usuário pode especificar tantas partições quantas se fizerem necessárias para cada uma das variáveis nebulosas, objetivando uma melhor representação do problema. Para auxiliar o usuário na especificação destes conjuntos, o sistema imediatamente cria representações gráficas dos conjuntos adicionados.

A Figura 5.10 ilustra a tela do sistema para o segundo passo, demonstrando a representação gráfica para as funções de pertinência para cada variável envolvida na instância de inferência. Destaca-se que cada uma das variáveis envolvidas encontra-se representada em abas, e o processo de especificação das funções de pertinência é realizado para cada uma destas variáveis.

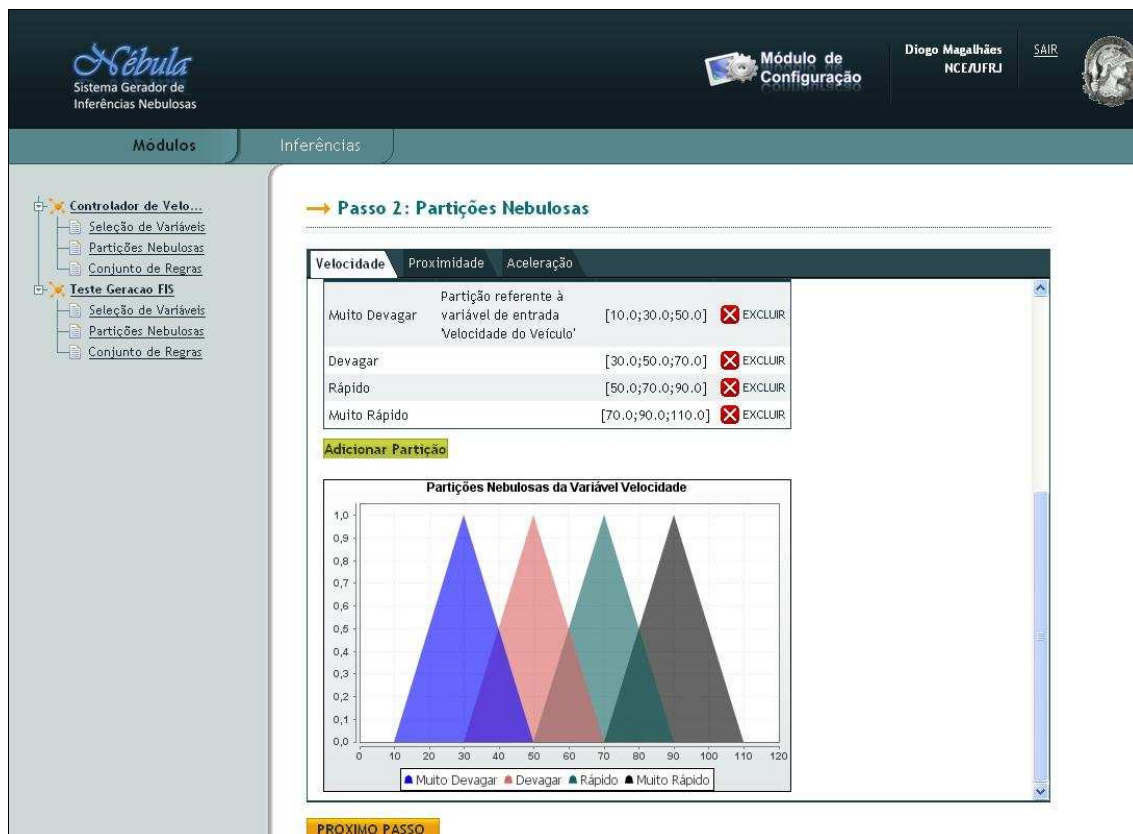


Figura 5.10: Interface do sistema para o passo 2: Partições Nebulosas

5.4.1.3 Especificação do Conjunto de Regras

Finalmente, o ultimo passo no processo de construção do algoritmo de inferências a partir do Módulo de configuração corresponde à especificação do conjunto de Regras.

Neste passo, as representações dos conjuntos nebulosos especificadas para cada uma das variáveis devem ser correlacionados logicamente, especificando-se um seqüenciamento causal no formato “se-então”.

A Figura 5.11 ilustra a interface do sistema Nébula para permitir a criação da base de regras.

Nebula
Sistema Gerador de Inferências Nebulosas

Módulo de Configuração
Diogo Magalhães
NCE/UFRJ

SAIR

Módulos | **Inferências**

Passo 3: Conjunto de Regras

Selecione as Configurações da Regra

Se:

Velocidade:

Proximidade:

Conector:

Então:

Aceleração:

Incluir Regra

Variáveis Nebulosas Seleccionadas

Descrição	Excluir
SE 'Proximidade'='Muito Perto' E 'Velocidade'='Muito Devagar' ENTÃO 'Aceleração'='Frear'	☒ EXCLUIR
SE 'Proximidade'='Muito Perto' E 'Velocidade'='Devagar' ENTÃO 'Aceleração'='Frear Forte'	☒ EXCLUIR
SE 'Proximidade'='Muito Perto' E 'Velocidade'='Rápido' ENTÃO 'Aceleração'='Frear Forte'	☒ EXCLUIR
SE 'Proximidade'='Muito Perto' E 'Velocidade'='Muito Rápido' ENTÃO 'Aceleração'='Frear Forte'	☒ EXCLUIR
SE 'Proximidade'='Perto' E 'Velocidade'='Muito Devagar' ENTÃO 'Aceleração'='Manter'	☒ EXCLUIR
SE 'Proximidade'='Perto' E 'Velocidade'='Devagar' ENTÃO 'Aceleração'='Frear'	☒ EXCLUIR
SE 'Proximidade'='Perto' E 'Velocidade'='Rápido' ENTÃO 'Aceleração'='Frear Forte'	☒ EXCLUIR

Figura 5.11: Interface do sistema para o passo 3: Conjunto de Regras

5.4.2 Módulo de Simulação

O módulo de simulação é responsável por permitir a execução de simulações pontuais na instância de inferências recém criada. Isto confere ao especialista a possibilidade de melhorar a qualidade do algoritmo criado, calibrando seu comportamento a partir da redefinição dinâmica dos critérios estabelecidos para o algoritmo, como a formação dos conjuntos nebulosos e o mapeamento de regras, validando os resultados obtidos até que se tornem satisfatórios.

Este módulo possibilita a realização de adequações e ajustes nos parâmetros de sintonização especificados para o algoritmo de inferências, conforme os conceitos descritos na Seção 4.3.

O Módulo de Simulações utiliza as definições especificadas a partir do Módulo de Configuração para a instância de inferências, e recebe como entrada um valor para cada variável prevista no mapeamento, retornando como saída o resultado de processamento da inferência.

Após a execução de cada teste, o especialista pode analisar o resultado e retornar ao módulo de configuração para realizar as alterações pertinentes, em sucessivas iterações até que os resultados obtidos representem o modelo proposto.

Para auxiliar o usuário na compreensão e análise dos resultados obtidos pela execução da inferência, o sistema apresenta dinamicamente os gráficos representativos das partições nebulosas das variáveis envolvidas, destacando as regiões acionadas no processamento para as entradas fornecidas, os graus obtidos em cada partição nebulosa da variável de saída, assim como as regras que foram ativadas no processamento de inferência para os valores submetidos para as variáveis de entrada, conforme ilustrado na Figura 5.12.

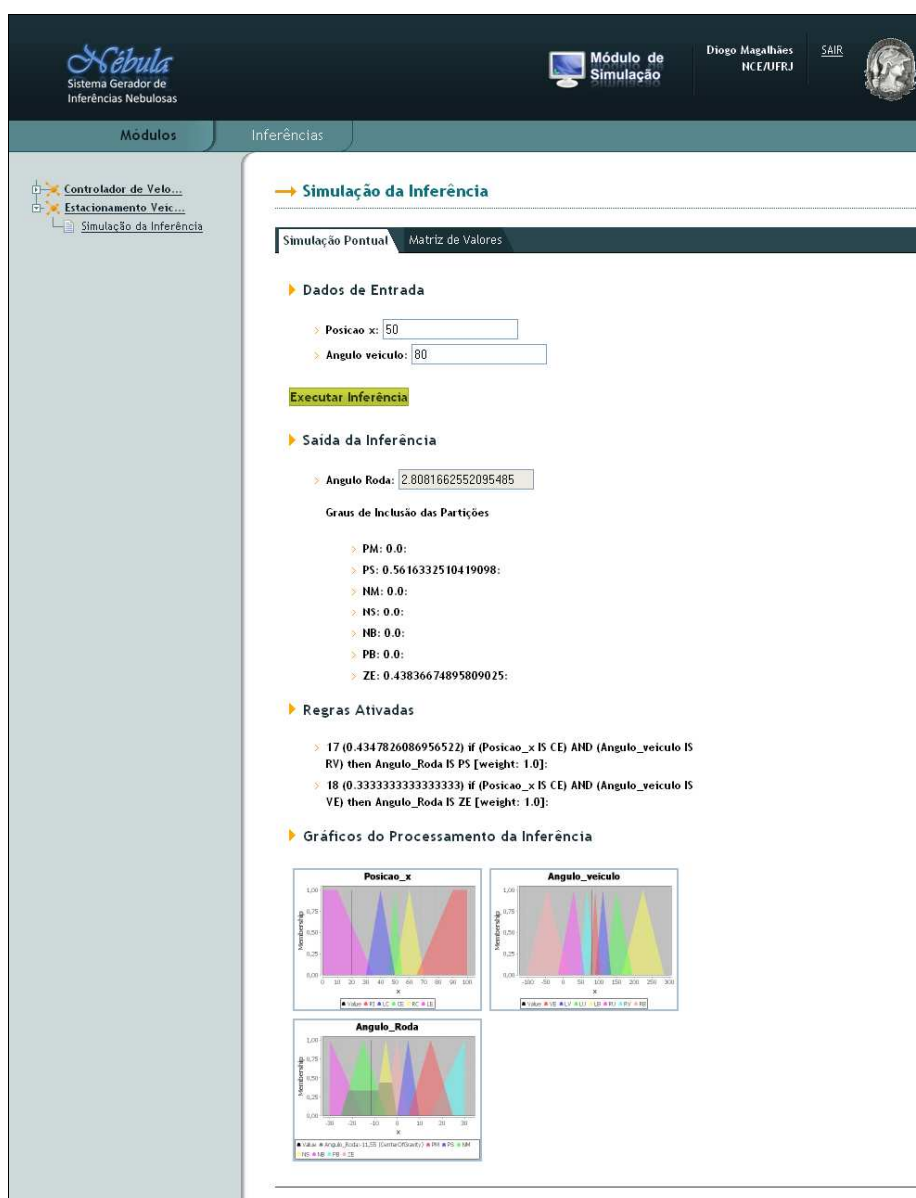


Figura 5.12: Interface do sistema para o Módulo de Simulação

Opcionalmente, o usuário pode realizar uma simulação em lote, especificando uma matriz de valores a ser submetida para processamento da inferência, avaliando-se diretamente um conjunto de resultados.

Após a execução dos testes a partir do módulo de inferências, ou a qualquer momento, o especialista pode retornar ao módulo de configuração para fazer ajustes na composição das variáveis, suas partições nebulosas e o conjunto de regras. Ao acessar o módulo de simulação, o usuário pode visualizar imediatamente os reflexos das alterações realizadas nas novas simulações, permitindo melhorar a qualidade do algoritmo segundo seus objetivos.

5.4.3 Módulo de Inferências

O módulo de inferências atua como a interface do desenvolvedor do jogo psicopedagógico com o serviço de inferências *fuzzy*. Este módulo expõe para o desenvolvedor uma interface de desenvolvimento, atuando como API, oferecendo o conjunto de métodos necessários para o jogo obter a instância da inferência apropriada, criada pelo especialista para atender a um propósito específico a partir do Módulo de Configuração, fornecer o conjunto dos dados de entrada e receber o valor de retorno processado pelo sistema de inferência nebuloso.

Os resultados obtidos por este módulo serão utilizados pelo jogo dentro do seu contexto de atuação e do objetivo a que a inferência se propõe, e poderão constituir insumos para ajustes nos jogos ou para apoiar a decisão do especialista na condução de sua análise cognitiva.

Este módulo também é utilizado pontualmente pelo módulo de simulações, para execução de testes e validações a partir de um conjunto de dados de entrada fornecidos pelo especialista através da interface do sistema.

5.4.4 Serviço de Inferências *Fuzzy*

O serviço de inferência nebuloso é responsável por transformar as definições especificadas para a instância de inferência, descrita pelos conjuntos nebulosos das variáveis e o mapeamento de regras, em termos matemáticos, e processar os resultados correspondentes ao conjunto de dados submetido como entrada.

Este módulo utiliza técnicas de lógica *fuzzy* para traduzir as imprecisões expressas pelos termos lingüísticos e variáveis nebulosas em abstrações de valores

(*fuzzification*) e os modelos matemáticos capazes de processá-los. Em função do mapeamento de regras, o algoritmo obtém o valor de saída correspondente, que é então transformado de volta em um número preciso (*deffuzification*), a ser fornecido como retorno para o usuário.

Este módulo constitui o núcleo funcional inteligente do sistema de inferências, responsável por interpretar o conhecimento do especialista expresso no mapeamento de regras, fornecendo o comportamento inteligente a ser agregado aos jogos.

5.5 Ambiente Computacional e Tecnologias

O ambiente computacional configurado para o desenvolvimento do sistema Nébula priorizou essencialmente a utilização de tecnologias abertas (*open source*) e com licenciamento de livre utilização, de maneira a permitir que futuras modificações e evoluções na ferramenta possam ser realizadas.

Focando-se especialmente na qualidade de código e na usabilidade do produto final, foram aplicadas as melhores práticas de desenvolvimento utilizando padrões de projeto (*design patterns*) recomendados na literatura (FOWLER, 2000), e ambientes para aplicações de interfaces ricas (*RIA – Rich Internet Applications*).

A seguir, são apresentadas a arquitetura geral do sistema e a descrição das principais tecnologias utilizadas.

5.5.1 Arquitetura Geral do Sistema

A arquitetura geral do sistema segue o clássico padrão de engenharia de projetos denominado *Model View Controller* (MVC) (2011), conhecido pela separação dos módulos do sistema nas camadas de apresentação (interface gráfica), camada de negócios (modelo), e uma camada intermediária que gerencia o fluxo de informação entre as camadas anteriores (controlador). Desta forma, alterações na interface gráfica podem ser realizadas sem interferirem diretamente na lógica do sistema, implementada na camada de modelo. De maneira análoga, os dados podem ser reorganizados sem afetar a maneira como a informação é apresentada na interface gráfica. Este modelo de organização de tarefas mostra-se de relevante importância no desenvolvimento de aplicações complexas, viabilizando que as manutenções evolutivas em um determinado

módulo possam ocorrer com o mínimo de impacto nos demais, a partir da separação de tarefas em cada uma das camadas.

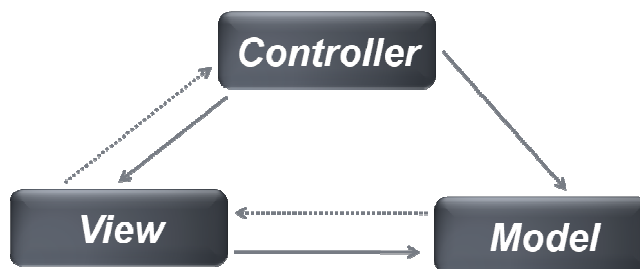


Figura 5.13: Diagrama do padrão de arquitetura *Model View Controller* (MVC)

A utilização desta arquitetura permitiu desenvolver o módulo de inferências e publicá-lo em uma biblioteca à parte, que pôde ser reutilizada tanto no módulo de simulações quanto no serviço de inferências. As interfaces do serviço de inferências, por sua vez, foram modeladas e disponibilizadas de forma a poderem ser acessadas diretamente a partir de uma API Java, ou opcionalmente por chamadas do tipo REST, viabilizando sua utilização por sistemas desenvolvidos em outras plataformas além do Java, como os jogos desenvolvidos em Flash, por exemplo.

A utilização da biblioteca Java contendo o núcleo de inferências do Nébula permite que as aplicações executem seu modelo matemático diretamente no mesmo ambiente computacional destas aplicações, permitindo seu melhor gerenciamento de desempenho. Desta forma, uma instância de inferências é criada e executada localmente, eliminando a necessidade de se efetuar múltiplas requisições REST através da rede, e permitindo que as aplicações possam desenvolver recursos de execução em tempo real.

5.5.2 Linguagem de Programação Java

Para o desenvolvimento do sistema Nébula foi utilizada a plataforma Java, por ser uma linguagem computacional completa para o desenvolvimento de aplicações baseadas na Web, e com fácil compatibilidade com grande parte dos jogos, também desenvolvidos nesta linguagem. Em especial, além de sua ampla utilização em ambiente acadêmico, a plataforma Java apresenta o diferencial de ser multiplataforma, isto é, pode ser executada em qualquer sistema operacional – o que é um requisito importante uma vez que não há definições a respeito da composição dos recursos computacionais disponíveis para os especialistas realizarem as sessões de jogo. Desta forma, o sistema

fica independente das escolhas que vem a ser tomadas no momento da construção destes ambientes.

5.5.3 Java ServerFaces

Para o desenvolvimento das interfaces ricas para o usuário, foi utilizada a tecnologia Java ServerFaces (JSF) (ORACLE, 2011), que é um *framework web* baseado no padrão MVC, que fornece diversos componentes para a construção de interfaces complexas.

Existem diversas bibliotecas de componentes que implementam o framework JSF. Para o sistema Nébula, foi utilizada a biblioteca de componentes Richfaces (JBOSS, 2011), em função de sua extensa disponibilidade de componentes, além de sua documentação completa e de fácil utilização. Disponibilizado e mantido pela fundação JBoss, o Richfaces é de utilização livre, e fornece a automatização do gerenciamento de recursos Ajax dentro da aplicação.

5.5.4 GlassFish Application Server

O GlassFish Application Server é um servidor de aplicações de código aberto e amplamente utilizado por sua robustez e confiabilidade. O servidor de aplicações foi escolhido por sua fácil integração com a IDE de desenvolvimento NetBeans, sua rapidez e por ser mantido pela mesma empresa que detém os direitos da plataforma Java – a Oracle, além de ser atualmente a implementação de referência da plataforma Java Enterprise Edition (JEE).

5.5.5 Banco de Dados JavaDB

Para o sistema Nébula optou-se por utilizar o banco de dados JavaDB (ORACLE, 2011). Trata-se de um banco de dados relacional, gratuito, de código aberto e extremamente leve, podendo ser embutido em programas Java. Possui apenas 2 megabytes, mas com um tempo de resposta que atende aos requisitos iniciais para a execução do projeto de pesquisa e dos experimentos. No futuro, com o aumento da carga de dados a ser tratada e com a disponibilização do sistema em ambientes distribuídos, poderão ser utilizados outros sistemas gerenciadores de banco de dados, necessitando apenas de poucas modificações nas configurações do ambiente de desenvolvimento.

5.5.6 Hibernate

O Hibernate (JBOSS, 2011) é um framework de persistência em Java que objetiva facilitar o desenvolvimento de sistemas orientado a objetos que necessitem realizar a persistência de informações em bancos de dados relacionais. Sua principal função é realizar as tarefas de mapeamento objeto-relacional, automatizando a transformação das classes em Java em formatos equivalentes com os modelos relacionais. É um software de código aberto, sob licença LGPL. O termo mapeamento objeto-relacional corresponde à técnica de se mapear as propriedades de classes Java em colunas correspondentes das tabelas do modelo relacional.

5.5.7 JFuzzyLogic

JFuzzyLogic é um componente inteiramente escrito em Java que implementa um sistema de inferências *fuzzy* (*FIS-Fuzzy Inference System*) completo, bem como as linguagens de controle Fuzzy, conforme a especificação IEC 61131-7. Sua utilização objetiva evitar a necessidade de se reescrever todo um conjunto de algoritmos e modelos matemáticos, utilizando uma ferramenta amplamente testada e validada, concentrando-se apenas no assunto principal da dissertação.

Capítulo VI

Estudo de Casos e Análise de Resultados

“Conhecer não é demonstrar nem explicar, é aceder à
visão”

Antoine de Saint-Exupéry

Neste capítulo, são apresentados os estudos de caso especificados com a finalidade de verificar o cumprimento dos objetivos principais deste projeto de pesquisa e validar suas hipóteses. Inicialmente, apresenta-se a metodologia utilizada, seguindo-se à apresentação de cada estudo de caso e seus respectivos resultados.

6.1 Metodologia

Para avaliar o modelo proposto e a ferramenta Nébula, criada como produto principal deste trabalho, foram realizados três estudos de caso especialmente designados para validar a adequação do modelo de funcionamento da construção dos algoritmos de inferência e sua inclusão no contexto dos jogos psicopedagógicos.

O primeiro estudo utiliza-se de um jogo especialmente desenvolvido para avaliar de maneira integrada o comportamento do sistema Nébula, explorando-o em seus diversos aspectos, analisando suas possibilidades e limitações, e submetendo-o a testes de processamento de informações em um ambiente de execução em tempo real. Este jogo serviu como implementação de referência ao utilizar-se de todos os recursos oferecidos pelo sistema Nébula, desde sua interface gráfica até seus recursos de programação e de integração. O jogo escolhido foi o do “Estacionamento do Caminhão” (KOSKO, 1992), amplamente utilizado na literatura para exemplificar os conceitos de sistema de controle *fuzzy*.

O segundo estudo de caso se propõe a analisar matematicamente a precisão dos resultados obtidos pelo núcleo de inferências do sistema Nébula para um conjunto de dados de entrada, comparando-os aos resultados obtidos pela execução de um modelo equivalente implementado no software Matlab (MATHWORKS, 2008). A partir de análises gráficas e por tabelas, objetiva-se validar os modelos de cálculos utilizados no núcleo de inferências, garantindo sua confiabilidade no processamento dos dados de entrada no contexto dos algoritmos propostos.

O terceiro caso corresponde à realização de um estudo prático exploratório, envolvendo a participação de neurocientistas na utilização do sistema. São analisadas as suas impressões na utilização da interface gráfica e na compreensão dos conceitos envolvidos, verificando se houve sucesso ao buscar descrever seu conhecimento na modelagem de um algoritmo de inferências através do sistema Nébula, baseadas na observação prática e nos registros escritos dos usuários. Este estudo objetiva validar os critérios de usabilidade do sistema Nébula na abstração das representações matemáticas envolvidas nos sistemas *fuzzy* em conceitos mais facilmente compreensíveis.

A Figura 6.1 ilustra graficamente os estudos de caso propostos para a validação dos objetivos e hipóteses propostos para este trabalho de pesquisa.



Figura 6.1: Diagrama ilustrativo dos estudos de caso propostos

Enquanto a análise técnico-matemática, proposta no segundo estudo de caso, objetiva analisar quantitativamente a precisão e a confiabilidade dos dados apresentados, os outros dois estudos podem ser considerados essencialmente qualitativos, visto que se preocupam em analisar as impressões e interpretações oferecidas pelas pessoas participantes dos experimentos práticos.

A partir destes estudos, pretende-se demonstrar que a ferramenta é capaz de apresentar resultados satisfatórios para os algoritmos modelados a partir de sua interface, assim como apresentar critérios satisfatórios de usabilidade e de interação, de forma a permitir o envolvimento dos diferentes participantes das equipes multidisciplinares no processo de construção dos algoritmos.

6.2 Simulador do Estacionamento de Veículo

6.2.1 Conceitos Gerais

O primeiro estudo de caso foi designado para verificar o completo funcionamento de todas as etapas envolvidas na utilização do sistema Nébula em condições reais de uso, desde a construção de uma instância de inferências até sua integração a um jogo psicopedagógico. Para isto, foi desenvolvido um simulador baseado no clássico problema do “estacionamento de veículo” (KOSKO 1992; CRUZ, 2002; MORATORI, 2006; TANSCHKEIT, 2011), amplamente referenciado na literatura técnica para exemplificação do funcionamento de sistemas de inferência *Fuzzy*.

Este simulador foi concebido para servir de implementação de referência, utilizando de maneira integrada as diversas possibilidades e recursos oferecidos pelo sistema Nébula, como: requisições REST para obtenção das configurações do algoritmo, instanciação e processamento local do núcleo de inferências, execução de inferências em tempo real, atualização automática de alterações nos parâmetros do algoritmo refletidas imediatamente no jogo em execução, além de validar o uso da API para integração do Nébula nos jogos psicopedagógicos.

A construção do simulador “Estacionamento de Veículo” propõe especificar o conjunto de definições *fuzzy* que determinarão o algoritmo necessário para direcionar um veículo, posicionado em uma posição x de um pátio (deslocamento no eixo horizontal) e rotacionado em um ângulo ϕ (ângulo do veículo em relação ao eixo horizontal), para um determinado local que delimita uma região representativa de uma vaga no estacionamento. É relevante destacar que, no problema originalmente proposto por Kosko (1992), a posição y do veículo do pátio não é considerada no contexto da modelagem *fuzzy* do problema. A variável de saída do sistema de inferências θ representa o ângulo da roda, que irá direcionar o deslocamento do veículo em um determinado sentido. As condições de parada satisfatórias envolvem o posicionamento do veículo dentro da região delimitada, considerando sua posição x como *central (CE)*, e seu ângulo ϕ como 90° (*vertical – VE*) A Figura 6.2 representa esquematicamente as variáveis envolvidas no jogo do caminhão.

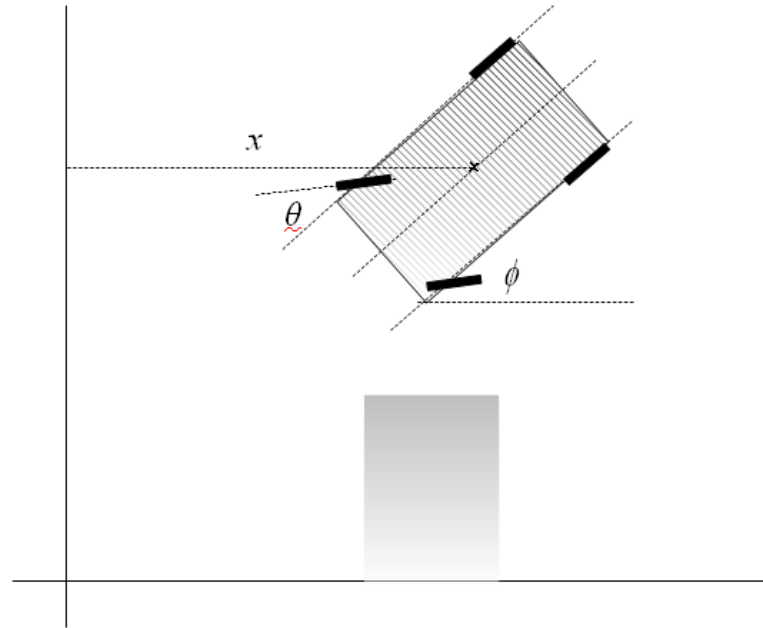


Figura 6.2: Esquema das variáveis envolvidas do jogo do Estacionamento do Caminhão

O funcionamento do simulador ocorre da seguinte forma: o veículo pode ser posicionado em qualquer local do pátio, e rotacionado em um ângulo qualquer, à escolha do usuário. Ao iniciar a simulação, o veículo deverá ser automaticamente direcionado por uma trajetória que o conduza para a posição final determinada pela vaga no estacionamento. O deslocamento do veículo é simulado pelo sistema em uma sequência de passos, onde a cada passo sua posição x e seu ângulo ϕ são submetidos ao sistema de inferências, e o veículo é então rotacionado na direção θ , determinada como saída pelo sistema de inferências, e deslocado em uma distância fixa w .

A Equação 6.1 descreve as funções de movimento que determinam a rotação e o deslocamento do veículo, onde as variáveis x e y representam suas coordenadas no pátio, e a variável w representa a distância fixa em que o veículo é deslocado a cada passo. A posição do veículo é modificada atualizando-se a posição das rodas, incrementando o ângulo atual ϕ com o valor θ retornado pelo sistema de inferências.

$$\begin{aligned}x' &= x + w \cos(\phi') \\y' &= y + w \sin(\phi') \\ \phi' &= \phi + \theta\end{aligned}$$

Equação 6.1: Conjunto de equações que descrevem as funções de movimento do veículo

As partições nebulosas para cada variável de entrada e para a variável de saída foram determinadas conforme descrito na Tabela 6.1 (CRUZ, 2002; MORATORI, 2006);

Partições Nebulosas e Rótulos					
Variável de entrada ϕ - ângulo do veículo		Variável de Entrada x - deslocamento horizontal		Variável de Saída θ - ângulo da roda	
RB	Right Below	LE	Left	NB	Negative Big
RU	Right Upper	LC	Left Center	NM	Negative Med
RV	Right Vertical	CE	Center	NS	Negative Small
VE	Vertical	RC	Right Center	ZE	Zero
LV	Left Vertical	RI	Right	OS	Positive Small
LU	Left Upper			PM	Positive Med
LB	Left Below			PB	Positive Big

Tabela 6.1: Partições Nebulosas identificadas para o sistema de inferências

Para cada partição nebulosa identificada, suas funções de pertinência foram especificadas conforme ilustrado na Figura 6.3 e na Figura 6.4, que definem os intervalos para as funções de inclusão das variáveis de entrada e da variável de saída, respectivamente, conforme as definições propostas nos trabalhos de Cruz (2002) e Tanscheit (2011).

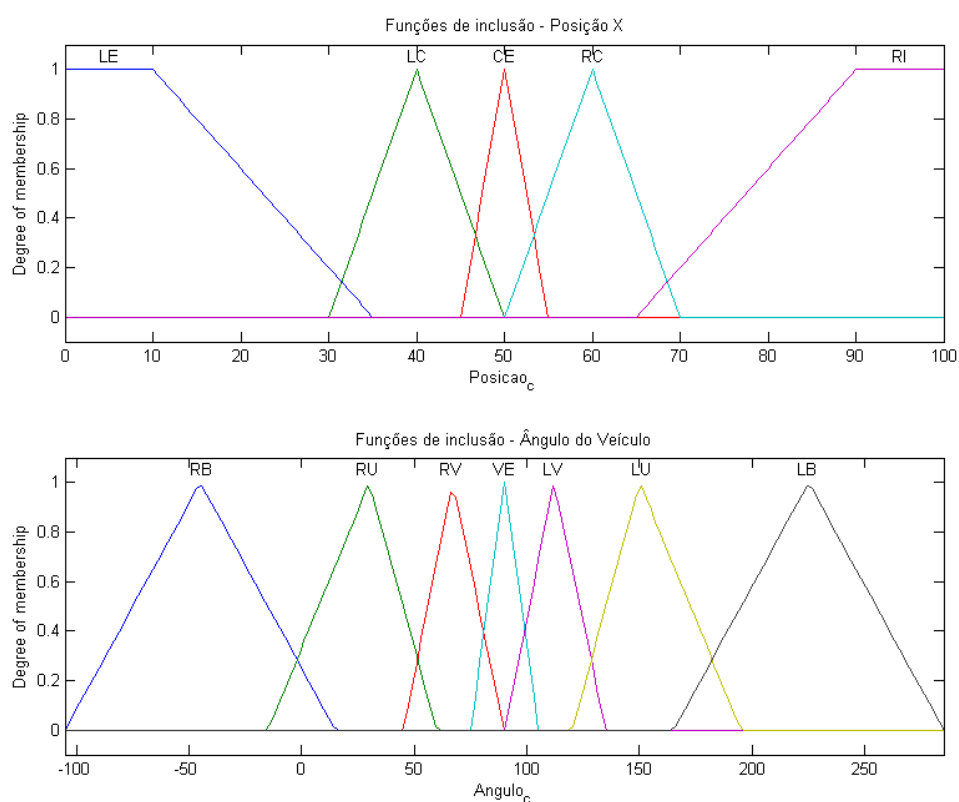


Figura 6.3: Funções de inclusão para as variáveis de entrada

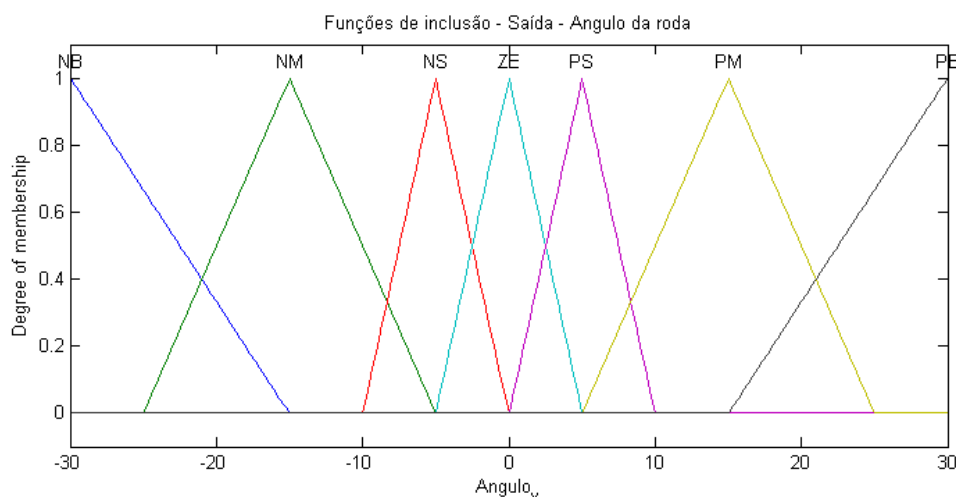


Figura 6.4: Funções de inclusão para a variável de saída

Considerando-se as partições nebulosas especificadas para as variáveis de entrada e de saída, puderam-se estabelecer as definições para o conjunto de regras descrito pela Tabela 6.2, onde a variável de saída corresponde ao ângulo da roda, representado pela variável θ .

$\phi \backslash x$	LE	LC	CE	RC	RI
RB	PS	PM	PM	PB	PB
RU	NS	OS	PM	PB	PB
RV	NM	NS	OS	PM	PB
VE	NM	NM	ZE	PM	PM
LV	NB	NM	NS	OS	PM
LU	NB	NB	NM	NS	OS
LB	NB	NB	NM	NM	NS

Tabela 6.2: Base de regras para o sistema de inferências do estacionamento de veículo

Portanto, a partir da matriz de regras descritas pela Tabela 6.2, pode-se ler uma determinada célula conforme a regra: **Se** (x é RB) **e** (ϕ é LE) **então** (θ é PS). Este formato natural de interpretação e leitura será utilizado para descrever a base de regras para o simulador a partir do sistema Nébulas, conforme descrito na Seção 6.2.2.

6.2.2 Implementação do Simulador e Integração ao Nébulas

O simulador de estacionamento de veículos foi desenvolvido em conformidade com a infraestrutura de desenvolvimento de jogos psicopedagógicos, visando adequar-se mais fielmente ao contexto dos demais jogos que seriam desenvolvidos. Utilizou-se a plataforma Phidias (GOMES et al, 2009) para desenvolver os recursos gráficos de interação com os elementos na tela e a animação do veículo, assim como para

possibilitar o recurso iterativo de processamento realizado a cada passo do deslocamento do veículo, capturando os eventos de clique e arrasto de mouse, além de obter a posição e rotação do veículo em cada momento.

No simulador, foram implementados apenas os recursos gráficos e as equações de deslocamento do veículo, de forma que o algoritmo para o cálculo da variável de saída, correspondente ao ângulo da roda, seria definido pela inferência a ser realizada pelo sistema Nébulas. Portanto, não houve nenhuma referência ou acoplamento no código-fonte do jogo à inteligência computacional necessária para o cálculo desta inferência. Neste contexto, apenas a instânciação do conjunto de configurações do algoritmo é utilizada, e os dados de entrada são submetidos para processamento pelo núcleo de inferências do sistema Nébulas.

A Figura 6.5 ilustra a tela inicial para o simulador, demonstrando a interface gráfica fornecida pela Plataforma Phidias, a figura estilizada do veículo a ser movimentado, e os botões para permitir o início/pausa do jogo, a rotação do veículo em ambas as direções, a atualização da instância de inferências e a habilitação do recurso de rastro de trajetória, nesta ordem. No canto superior direito, um quadro apresenta a situação atual das variáveis envolvidas na simulação: a posição do veículo, seu ângulo de rotação e o ângulo da roda.

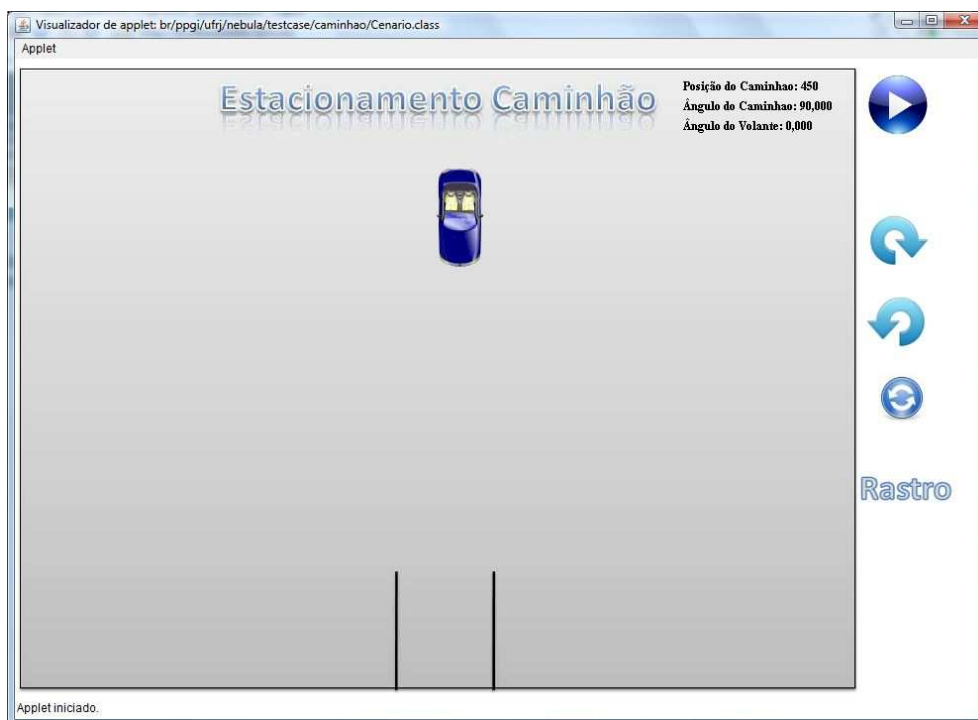


Figura 6.5: Tela inicial do simulador de estacionamento de veículo

Para modelar o algoritmo que representará o comportamento do veículo durante a simulação, criou-se uma instância de inferências a partir da interface Web do sistema Nébula. Esta instância é identificada por um número único, atribuído automaticamente pelo sistema, que é utilizado pelo simulador para obter a referência apropriada do algoritmo a ser utilizado. A Figura 6.6 ilustra a tela do sistema Nébula correspondente à instância de inferências criada para o simulador de estacionamento.



Figura 6.6: Tela do sistema Nébula, com a instância criada para o simulador de estacionamento

Para que a simulação do movimento do veículo ocorra de maneira mais natural, com transições suaves em seu deslocamento, é desejável que o processamento das variáveis de entrada seja executável com o máximo de rapidez, de maneira que o processamento da inferência ocorra em um intervalo de tempo menor do que o especificado para a transição entre os passos. Isto o assemelha em condições de uso de um sistema em tempo real. Portanto, o simulador utiliza um recurso disponibilizado pelo sistema Nébula que o permite instanciar um conjunto de definições do banco de dados e então executar localmente o núcleo do sistema de inferências, eliminando a necessidade de múltiplas requisições de rede em um sistema centralizado. Isto permite ao desenvolvedor melhor controle do desempenho da aplicação, restrita a um mesmo ambiente computacional de execução.

Desta forma, ao iniciar, o simulador solicita ao sistema Nébula, através de uma requisição REST enviada pela rede, a instância apropriada de inferências, cujas definições são carregadas a partir das configurações registradas no banco de dados.

Estas definições são então instanciadas localmente pela API do núcleo de inferências, que passa a executar o processamento das entradas que lhe são submetidas, considerando a base de conhecimento especificada para a inferência. Este modelo corresponde à forma *offline* de funcionamento do núcleo de inferências, que permite diminuir significativamente o tempo de resposta, ao eliminar a necessidade de realizar requisições de rede a cada passo do simulador que necessite de processamento.

Porém, como o processamento dos dados do simulador ocorre *offline*, as modificações nas definições do seu algoritmo, realizadas pelo sistema Nébula no banco de dados, não seriam imediatamente observáveis. Para isto, o simulador inclui um botão de *refresh*, que permite novamente obter do banco de dados as configurações da inferência recém-alterada, que são novamente instanciadas e podem ser percebidas no jogo sem a necessidade de reiniciá-lo.

Em situações normais de uso no contexto do projeto de virtualização dos jogos psicopedagógicos, a maioria do processamento de inferências deverá ser realizado de maneira centralizada pelo núcleo incluído no sistema Nébula, a partir de requisições REST, permitindo uma melhor centralização do algoritmo e a imediata sensibilização das alterações realizadas no banco de dados para as definições das inferências.

Ao finalizar a construção do simulador, foi especificado no sistema Nébula o conjunto de definições que irão compor a base de conhecimento para o sistema de inferência *fuzzy* que irá determinar o comportamento do veículo durante a simulação. Inicialmente, após selecionar as variáveis envolvidas, especificou-se o conjunto de partições nebulosas para as variáveis de entrada e de saída, em conformidade com os intervalos descritos pela Figura 6.3 e Figura 6.4, respectivamente. A tela do sistema Nébula descrita Figura 6.7 ilustra a definição das funções de pertinência correspondentes à variável de entrada posição x do veículo.

Para cada partição nebulosa criada, o sistema Nébula possui um editor de funções de pertinência, gerando o gráfico resultante das definições especificadas para os intervalos *fuzzy*. A Figura 6.8 e a Figura 6.9 ilustram os gráficos gerados pelo sistema para as funções de pertinência das variáveis de entrada e a variável de saída, para os intervalos definidos pela Figura 6.3 e Figura 6.4, respectivamente.

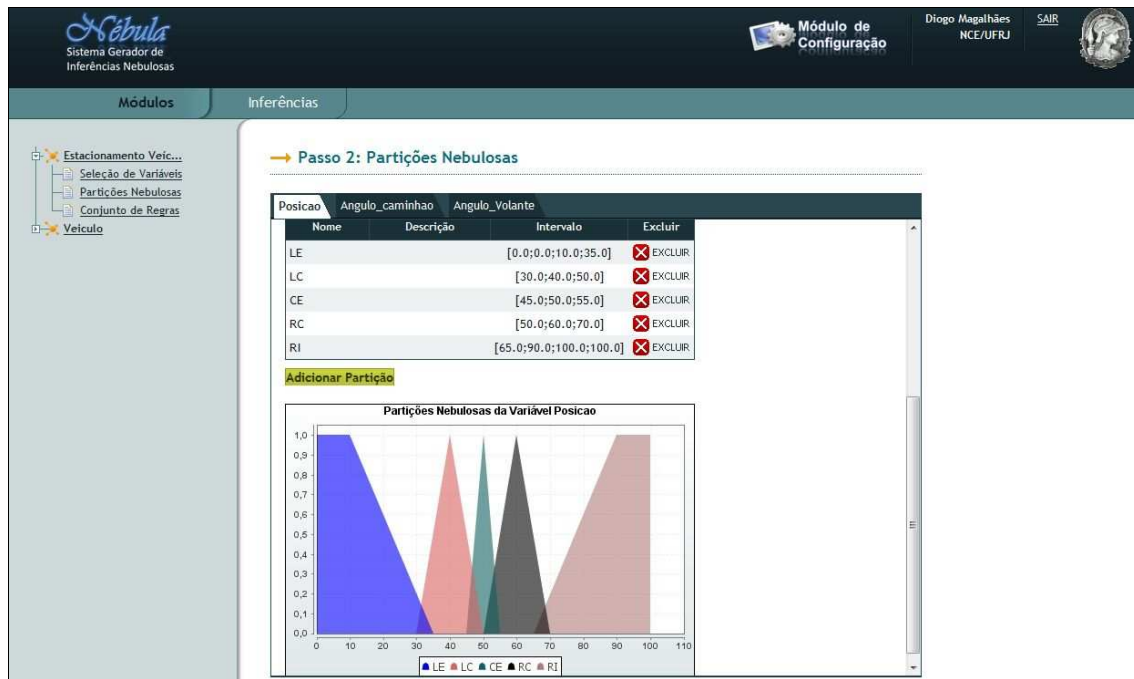


Figura 6.7: Tela do sistema Nébula, ilustrando as definições para as funções de pertinência

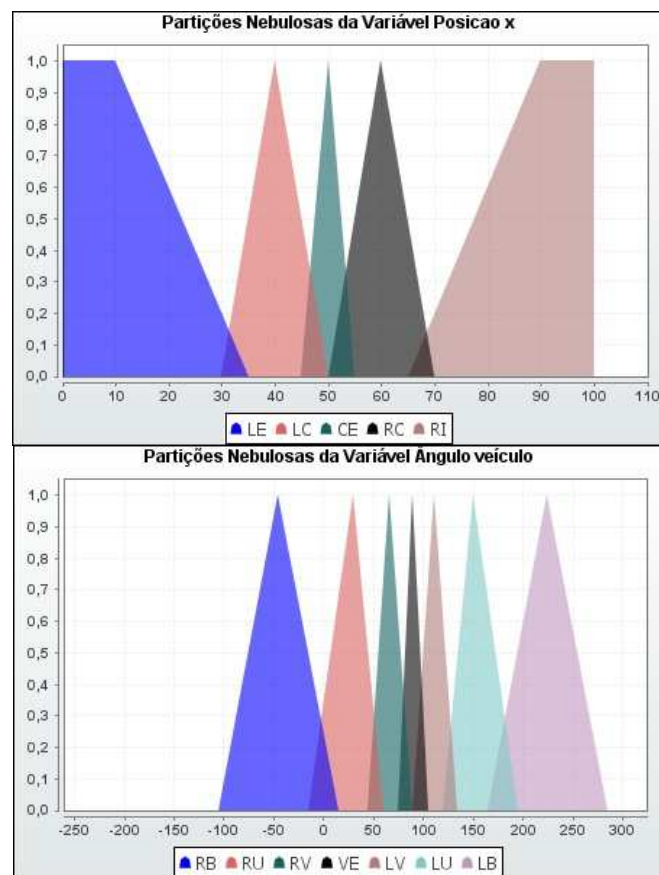


Figura 6.8: Gráficos gerados pelo Nébula para as funções de inclusão das variáveis de entrada

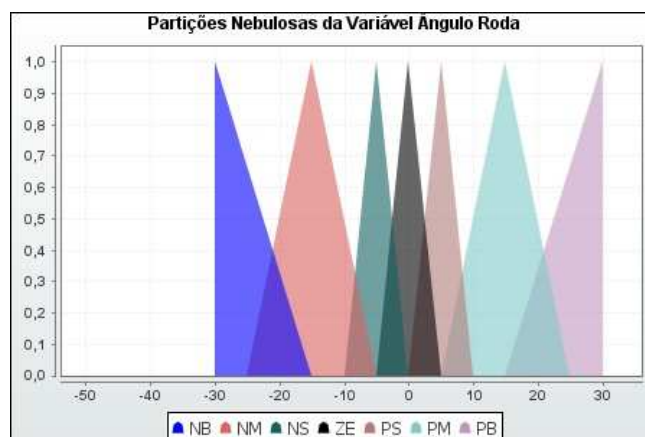


Figura 6.9: Gráficos gerados pelo Nébula para as funções de inclusão da variável de saída

Por fim, como último passo do processo de configuração da instância de inferências para o simulador de estacionamento, é realizada a correspondência das partições nebulosas com as respectivas saídas esperadas, mapeada segundo a base de regras descritas pela Tabela 6.2, conforme pode ser observado na captura de tela do sistema Nébula, descrita pela Figura 6.10.

Descrição	Excluir
SE 'Posicao'='LE' E 'Angulo_caminhao'='RB' ENTÃO 'Angulo_Roda'='PS'	☒ EXCLUIR
SE 'Posicao'='LE' E 'Angulo_caminhao'='RV' ENTÃO 'Angulo_Roda'='NS'	☒ EXCLUIR
SE 'Posicao'='LE' E 'Angulo_caminhao'='VE' ENTÃO 'Angulo_Roda'='NM'	☒ EXCLUIR
SE 'Posicao'='LE' E 'Angulo_caminhao'='LV' ENTÃO 'Angulo_Roda'='NB'	☒ EXCLUIR
SE 'Posicao'='LE' E 'Angulo_caminhao'='LU' ENTÃO 'Angulo_Roda'='NB'	☒ EXCLUIR
SE 'Posicao'='LE' E 'Angulo_caminhao'='LB' ENTÃO 'Angulo_Roda'='NB'	☒ EXCLUIR

Figura 6.10: Tela do sistema Nébula com a base de regras do simulador de estacionamento

6.2.3 Testes integrados do simulador de estacionamento

Uma vez configurados os parâmetros do algoritmo de inferências a partir do Sistema Nébula, diversos testes foram realizados diretamente no simulador em execução, avaliando-se a integração do simulador ao algoritmo descrito na plataforma.

Os testes consistem em analisar o comportamento no sistema de inferências em sua inter-relação com o jogo em execução, assim como analisar a carga do conjunto de informações e o efetivo processamento das inferências, considerando-se um ambiente integrado que represente uma situação real de utilização.

Portanto, após as configurações iniciais para a descrição do algoritmo que descreve o comportamento do veículo e a preparação do jogo para a instanciação desta inferência, conforme descrito na Seção 6.2.2, foram realizados sucessivos testes analisando a trajetória do veículo quando posicionado em diferentes coordenadas no pátio, conforme prevê a descrição do problema, além de considerar aleatoriamente diferentes ângulos de rotação. A modelagem do algoritmo propõe solucionar o percurso do veículo para o local de destino pré-determinado em quaisquer combinações destas situações iniciais. A partir deste conjunto de definições para o algoritmo, fundamentadas nos trabalhos de Cruz (2002) e Tanscheit (2011), criadas e configuradas diretamente na interface integrada do sistema Nébula, pôde-se verificar o comportamento esperado para o deslocamento do veículo em direção ao ponto de destino definido na proposição do problema. A Figura 6.11 ilustra as trajetórias de diferentes simulações executadas para o problema do estacionamento, considerando-se diferentes situações iniciais para o posicionamento e rotação do veículo no pátio. Pode-se observar que as trajetórias conduzem adequadamente para o local de destino apropriado, indicando a correta modelagem para a representação do algoritmo de inferências.



Figura 6.11: Rastros das trajetórias de diferentes simulações para o problema do estacionamento

Para analisar o comportamento do jogo diante de alterações pontualmente realizadas nas configurações do algoritmo de inferências, feitas a partir do ambiente integrado do sistema Nébula e, portanto, desacopladas de sua implementação, foram feitas simulações em que os parâmetros do algoritmo foram propositalmente alterados, e as modificações no comportamento do veículo observadas durante a execução do simulador. A Figura 6.12 ilustra uma situação em que o conjunto de regras foi propositalmente alterado de forma a incluir definições incorretas, com o objetivo de inserir um comportamento inadequado, destacando a imediata percepção das alterações no comportamento do veículo. Pôde-se verificar que o comportamento do jogo foi imediatamente alterado, conforme previsto nas definições realizadas no sistema Nébula.

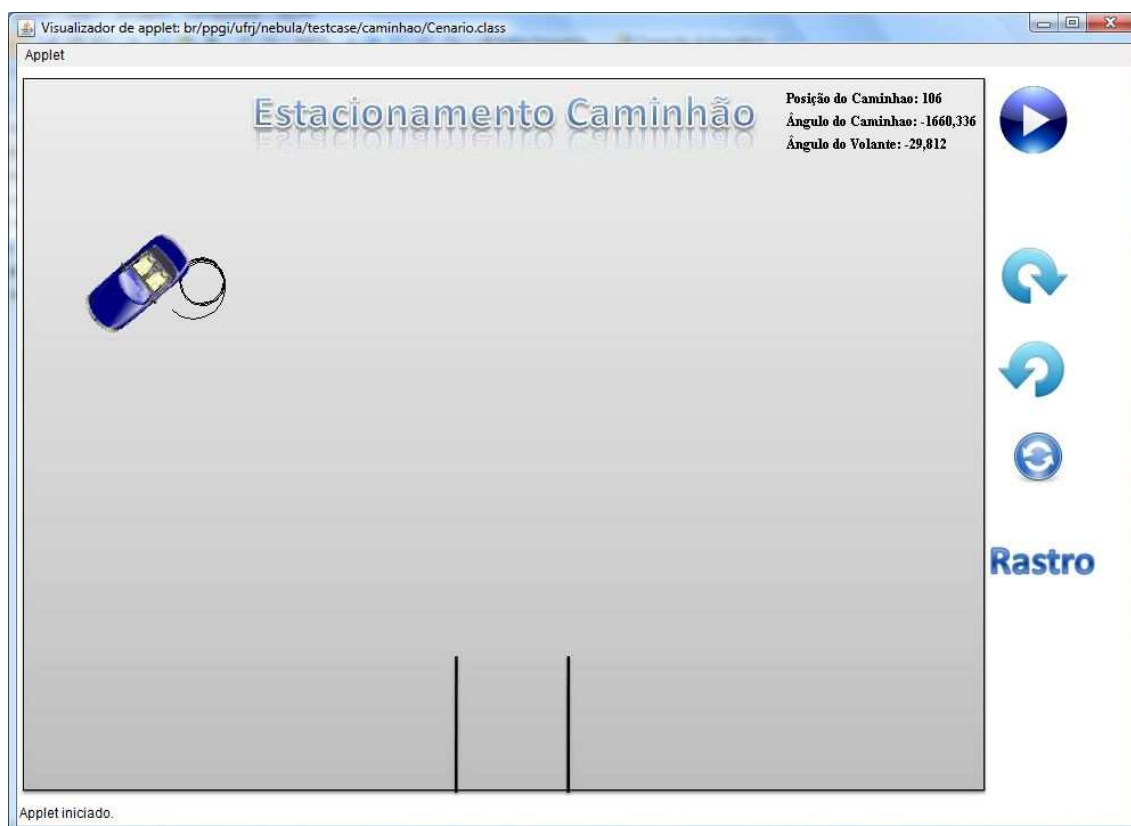


Figura 6.12: Alteração do comportamento do veículo, conforme novas definições realizadas no sistema Nébula

As simulações foram realizadas em um computador AMD Athlon 64 X2 dual core processor TK-55, com 2GB de memória RAM, e obteve-se tempos de respostas para a execução do processamento da inferência variando entre 6 e 18 milisegundos, conforme o melhor e o pior caso, respectivamente. Como o intervalo de tempo considerado para cada passo da animação foi de 100ms, o desempenho do processamento da inferência foi pequeno o suficiente para permitir os requisitos de execução em tempo real do simulador, com significativa margem de segurança. Para a instanciação da inferência, que necessita realizar requisições de rede via protocolo REST, e acessos a bancos de dados, o tempo de resposta obtido, em média, foi em torno de 2100 ms. A Figura 6.13 ilustra a saída da console de execução do jogo, com as medições do intervalo de tempo decorrido para processamento da inferência em cada passo.

```
Instância de inferências carregada com sucesso, em 2159 ms

Jogo em execucao...
Tempo decorrido para processamento da inferência: 18 ms
Tempo decorrido para processamento da inferência: 13 ms
Tempo decorrido para processamento da inferência: 10 ms
Tempo decorrido para processamento da inferência: 9 ms
Tempo decorrido para processamento da inferência: 6 ms
Tempo decorrido para processamento da inferência: 7 ms
Tempo decorrido para processamento da inferência: 8 ms
Tempo decorrido para processamento da inferência: 6 ms
Tempo decorrido para processamento da inferência: 7 ms
Tempo decorrido para processamento da inferência: 6 ms
Tempo decorrido para processamento da inferência: 6 ms
Tempo decorrido para processamento da inferência: 7 ms
```

Figura 6.13: Console de saída do simulador, representando os logs de intervalo de tempo para medição do desempenho do processamento da inferência

6.2.4 Considerações Parciais

A partir do conjunto de testes realizados, pôde-se validar a proposição inicial da utilização do sistema Nébulas para definir o conjunto de configurações para um sistema de inferências nebuloso, utilizado de maneira desacoplada por um jogo em execução. As definições realizadas pelo usuário a partir da interface Web do sistema Nébulas puderam ser instanciadas e executadas por um jogo criado na plataforma Phidias, que pode inclusive ser executado em outra máquina diferente daquela onde o Nébulas está sendo executado. Com isto, pôde-se verificar que o desacoplamento do algoritmo de inferências do código dos jogos é possível, eliminando o custoso ciclo de desenvolvimento de inteligência computacional nos jogos psicopedagógicos, conforme descrito na Figura 1.1.

A execução dos cálculos de inferência em um jogo que simula condições de processamento em tempo real permitiu verificar o desempenho satisfatório do núcleo de inferências nebuloso, com medições do tempo de resposta obtidos entre 6 e 18 milissegundos, considerados suficientemente pequenos para permitir que o processamento dos cálculos ocorresse de forma a não influenciar na fluência e naturalidade da animação de deslocamento do veículo. Os resultados obtidos em relação ao desempenho do processamento da inferência demonstraram grande rapidez nos cálculos, validando sua possibilidade de utilização em outros cenários diversos para jogos que exijam condições semelhantes, com restrições no tempo de resposta.

Adicionalmente, verificou-se que as alterações nas definições do algoritmo realizadas diretamente na interface *web* do sistema Nébula puderam ser imediatamente percebidas durante a execução do jogo, sem a necessidade de reiniciá-lo. Este recurso, além de facilitar o desenvolvimento de sistemas de inferência integrados aos jogos, permite realizar diversas simulações do comportamento do algoritmo, analisando os dados obtidos, calibrando e sintonizando seus parâmetros de maneira a buscar uma melhor aproximação dos resultados esperados. Com isto, obtêm-se maior dinamismo nos ajustes dos algoritmos, permitindo a imediata percepção dos impactos das alterações, diretamente observáveis no jogo em execução, sem a necessidade de atuar diretamente em seu código-fonte.

6.3 Análise Matemática do Núcleo de Inferências

6.3.1 Conceitos Gerais

Uma vez que os algoritmos de inferência sejam adequadamente modelados pelas equipes de especialistas, o sistema Nébula deve garantir que os resultados obtidos correspondam ao objetivo esperado. Portanto, seu processamento matemático deve ser preciso o suficiente para atender a seus propósitos de analisar os valores de entrada e, em função dos parâmetros de configuração do algoritmo, fornecer um valor satisfatoriamente preciso para a variável de saída inferida.

Desta forma, para validar a precisão do processamento matemático do núcleo de inferências do sistema Nébula, realizou-se um estudo de caso onde um mesmo modelo de algoritmo de inferências foi criado no sistema Nébula, e em uma outra ferramenta de engenharia consolidada no mercado, com confiabilidade reconhecidamente aceita na comunidade científica. Para este fim, escolheu-se o software Matlab (MATHWORKS, 2008), onde as mesmas definições de parâmetros utilizadas no Nébula foram especificadas, conforme especificado na Seção 6.2.

Ao realizar uma análise comparativa das matrizes de resultados processadas pelas duas plataformas – o Nébula e o Matlab – considerando um mesmo conjunto de parâmetros de configuração para o algoritmo, esperam-se resultados equivalentes e, com isso, pretende-se validar a precisão dos resultados obtidos, estendendo esta validação

para os demais modelos que virão a ser modelados na ferramenta, garantindo assim a confiabilidade de seus resultados.

6.3.2 Implementação do Algoritmo no Matlab

A partir das definições para o algoritmo de estacionamento do veículo previamente definidas no sistema Nébula, este mesmo conjunto de definições foi implementada no software Matlab, com o auxílio do toolbox Fuzzy.

Conforme descrito na Seção 6.1, foi analisada a utilização da plataforma Nébula para modelar o algoritmo que define o comportamento do veículo, corretamente validado no decorrer do simulador em execução. Portanto, ao modelar este algoritmo no software Matlab, pode-se comparar os resultados obtidos para o mesmo conjunto amostral de pares de valores para as variáveis de entrada.

É relevante destacar que, embora ambas as plataformas permitam a modelagem do algoritmo, visando, portanto, obter o mesmo resultado final como processamento de inferências, o uso do Matlab restringe-se ao seu perfil de especialistas na área de ciências exatas, como matemáticos, engenheiros e cientistas da computação, ao exigir conhecimentos específicos de manipulação de dados e de linguagem de programação. O Sistema Nébula, por sua vez, procura abstrair a complexidade deste procedimento para pessoas sem conhecimentos aprofundados destes conceitos, tornando-o acessível para os demais participantes das equipes multidisciplinares de especialistas envolvidos nos projetos de virtualização de jogos psicopedagógicos, permitindo-os obter um modelo equivalente a partir de processos simplificados de definição de parâmetros do algoritmo, orientados por sua interface gráfica amigável.

O primeiro passo para a modelagem do algoritmo de inferências no Matlab corresponde à especificação das variáveis de entrada e de saída, e a descrição das suas partições nebulosas. Utilizando o toolbox Fuzzy, as partições das variáveis de entrada e de saída foram especificadas, de acordo com os intervalos estabelecidos na Figura 6.3 e na Figura 6.4, respectivamente.

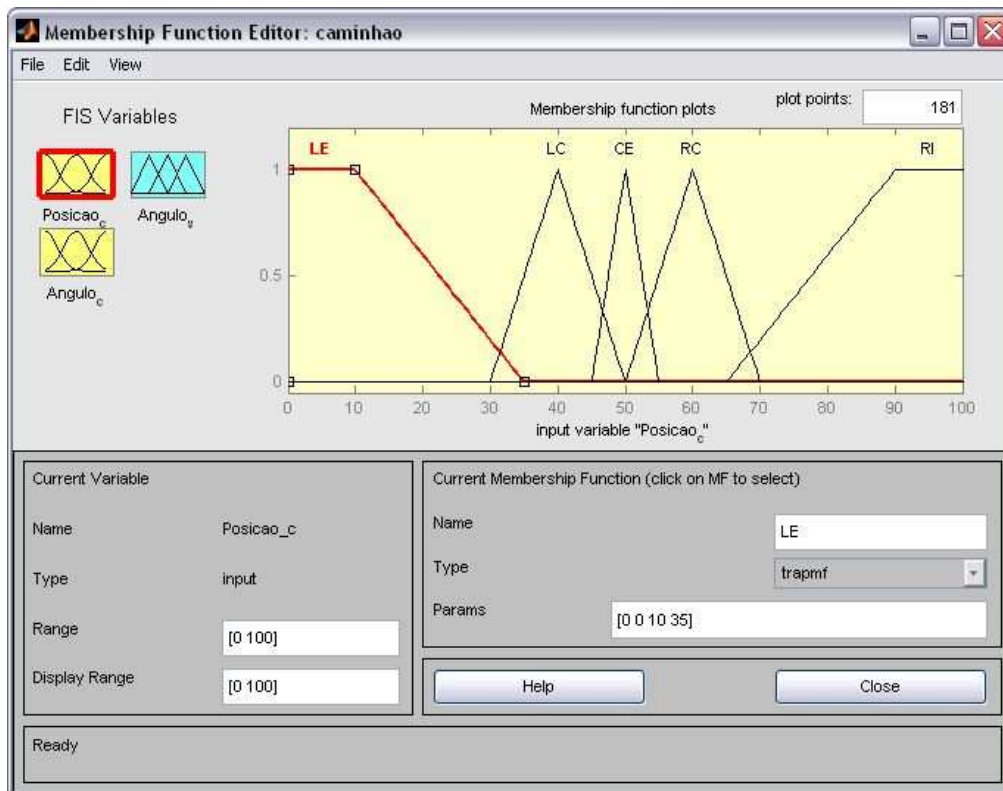


Figura 6.14: Toolbox do Matlab para definição das partições nebulosas

Conforme o conjunto de definições especificadas na Tabela 6.2, foram descritas as regras que definem o comportamento do algoritmo de inferências, correlacionando cada uma das partições das variáveis de entrada com sua correspondência lógica na variável de saída. A Figura 6.15 ilustra o mapeamento das regras definidos no Toolbox Fuzzy, do Matlab.

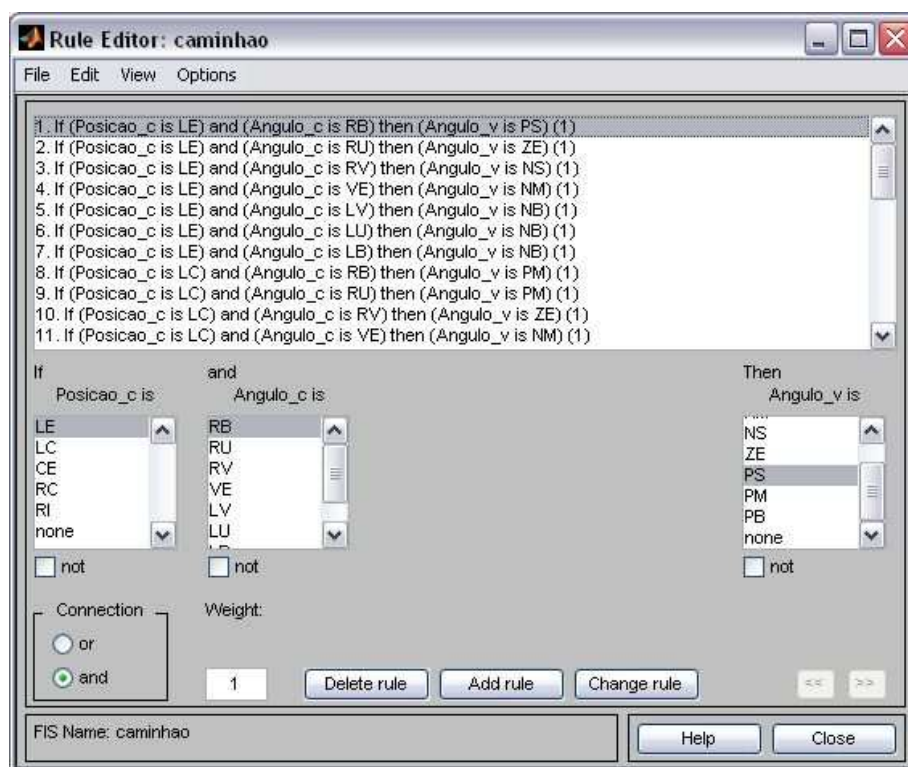


Figura 6.15: Toolbox Matlab para descrição da base de regras

Como ponto de partida, a fim de analisar inicialmente a correção e adequação dos parâmetros para o algoritmo recém-estabelecido, criou-se um cenário de validação completa, submetendo o algoritmo às diversas combinações possíveis para as variáveis de entrada, analisando seu comportamento sob a superfície de cobertura dos parâmetros iniciais. Desta forma, estabeleceu-se um conjunto de dados significativamente representativo para o problema proposto, em que, para cada parâmetro de entrada (posição x e ângulo do veículo), foram geradas 25 amostras, distribuídas linearmente em intervalos regulares, abrangendo variações em todas as partições nebulosas. A matriz resultante da combinação destes parâmetros resultou em 625 amostras, que foram submetidas à análise do algoritmo de inferência. As matrizes contendo as amostras para os dois parâmetros de entrada, e o resultado do processamento da inferência correspondente, compõem uma superfície que pode ser representada conforme o gráfico ilustrado na Figura 6.16.

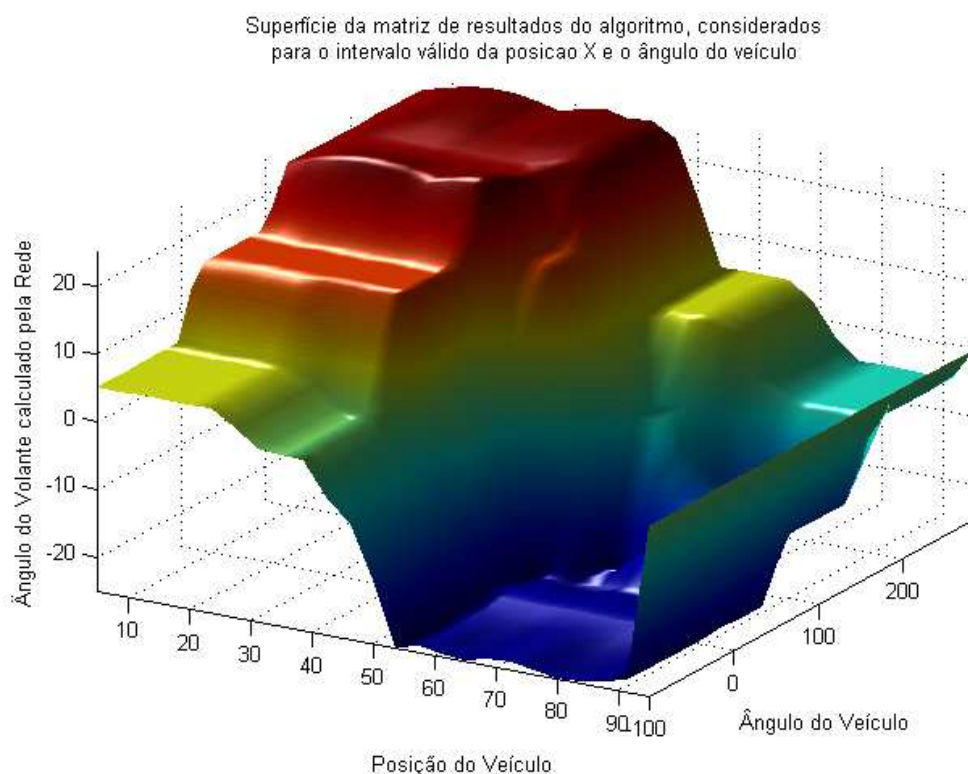


Figura 6.16: Superfície representativa da matriz contendo o par de parâmetros de entrada e a variável de saída obtida pelo processamento da inferência.

6.3.3 Análise Matemática Comparativa dos Resultados

Para composição da tabela de amostras comparativa entre os resultados do Nébulas e do Matlab, foi gerado um conjunto amostral minimamente significativo das partições nebulosas representadas. Este conjunto foi gerado a partir da combinação de 14 valores diferentes para a variável de entrada Ângulo do Veículo, e 10 valores diferentes para a posição do caminhão. Estes valores, obtidos linearmente dispostos no intervalo válido para cada uma destas variáveis, resultou em uma matriz de 140 amostras, que foram submetidas para processamento do algoritmo de inferências modelado no Nébulas, a ser comparativamente analisado perante os resultados obtidos para as mesmas amostras de entradas processados pelo Matlab.

O sistema Nébulas possui a funcionalidade integrada de simulação das inferências criadas a partir de sua plataforma, permitindo ao especialista modelador do algoritmo validar instantaneamente a adequação dos parâmetros definidos aos objetivos propostos, tornando possível realizar modificações e ajustes pontuais buscando melhores resultados. Portanto, duas maneiras de simulação estão disponíveis: a

Simulação Pontual e a Simulação por Matriz de Valores. Nesta última, é possível carregar um arquivo contendo o conjunto dos diversos valores para as variáveis de entrada a serem submetidos em lote para processamento pela inferência. Desta forma, o conjunto de valores de entrada gerados aleatoriamente no algoritmo implementado no Matlab, foi estruturado e armazenado em um arquivo texto, que foi importado pelo sistema Nébula, e submetido ao processamento. A Figura 6.17 ilustra os resultados apresentados pelo sistema Nébula para esta simulação.

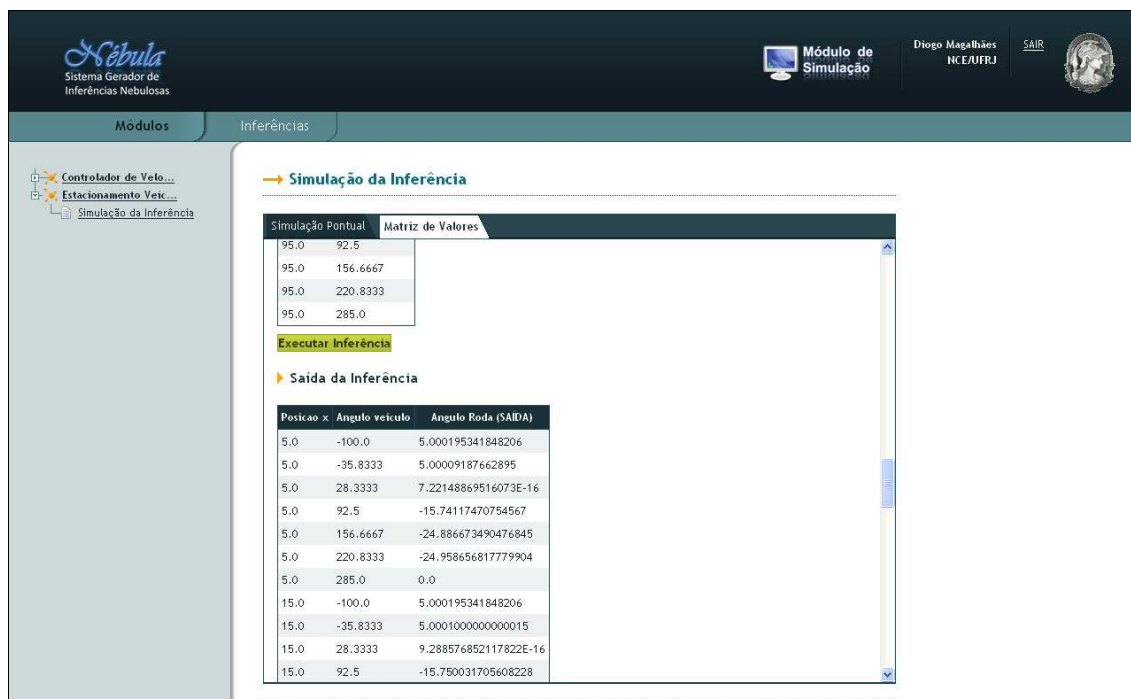


Figura 6.17: Console do Sistema Nébula para Simulação por Matriz de Valores

Em função do retorno da simulação fornecido pelo sistema Nébula, a Tabela 6.3 descreve os resultados obtidos, analisados comparativamente quanto à aproximação dos resultados do sistema Nébula em relação aos obtidos pelo Matlab. Para indicar a aproximação dos valores obtidos entre as duas ferramentas, utilizou-se o cálculo do erro relativo percentual (ERP), conforme ilustrado pela Equação 6.2, calculado individualmente para cada unidade amostral, que pode ser entendido como a discrepância identificada entre os dois valores medidos.

$$ERP = \frac{(V_{Matlab} - V_{Nébula})}{V_{Matlab}} \times 100$$

Equação 6.2: Equação para o cálculo do erro relativo percentual, onde V_{Matlab} e $V_{Nébula}$ correspondem aos valores retornados como saída pelo Matlab e pelo Nébula, respectivamente

Para servir como indicador único de precisão absoluta dos testes realizados, que resume a tabela comparativa apresentada, utilizou-se como índice a discrepância absoluta entre os valores obtidos, calculado a partir da função do erro médio percentual absoluto (EMPA), conforme descrito pela Equação 6.3.

$$EMPA = \frac{\sum_{t=1}^n |ERP_t|}{n}$$

Equação 6.3: Equação para o cálculo do Erro Médio Percentual Absoluto, calculado em função dos erros relativos percentuais em cada período, ou amostra

Para a bateria de testes realizadas comparativamente no Nébula e no Matlab, obteve-se o índice EMPA de 0,4398 %, evidenciando uma aproximação numérica satisfatória para os resultados obtidos pelo Nébula em relação ao Matlab. Deve-se apontar entre as possíveis causas de perdas de precisão o uso APIs para processamento matemático em Java, ou pequenas variações nos métodos de *fuzzificação* ou *defuzzificação* empregados por ambas as plataformas.

Comparativo de Resultados para a Inferência					
	Posição x	Ângulo do Veículo ϕ	Saída θ Nébula	Saída θ Matlab	Discrepância Percentual
1	5	-100,0000	5,0002	5,0887	1,7392%
2	5	-70,3846	5,0001	5,0106	0,2102%
3	5	-40,7692	5,0001	5,0085	0,1681%
4	5	-11,1538	4,1883	4,1962	0,1880%
5	5	18,4615	0,0000	0,0000	0,0000%
6	5	48,0769	-1,2876	-1,2782	-0,7328%
7	5	77,6923	-9,7324	-9,7448	0,1276%
8	5	107,3077	-24,7954	-24,9838	0,7540%
9	5	136,9231	-24,3226	-24,4974	0,7135%
10	5	166,5385	-24,4901	-24,6680	0,7212%
11	5	196,1538	-24,2039	-24,3769	0,7096%
12	5	225,7692	-24,9794	-25,1953	0,8568%
13	5	255,3846	-24,1333	-24,3062	0,7114%
14	5	285,0000	0,0000	0,0000	0,0000%
15	15	-100,0000	5,0002	5,0887	1,7392%
16	15	-70,3846	5,0001	5,0106	0,2102%
17	15	-40,7692	5,0001	5,0100	0,1976%
18	15	-11,1538	4,1883	4,1962	0,1880%
19	15	18,4615	0,0000	0,0000	0,0000%
20	15	48,0769	-1,2876	-1,2782	-0,7328%
21	15	77,6923	-9,7324	-9,7448	0,1276%
22	15	107,3077	-24,7954	-24,9838	0,7540%
23	15	136,9231	-24,3226	-24,4974	0,7135%
24	15	166,5385	-24,4901	-24,6680	0,7212%
25	15	196,1538	-24,2039	-24,3769	0,7096%
26	15	225,7692	-24,8161	-25,0065	0,7613%
27	15	255,3846	-24,1333	-24,3062	0,7114%
28	15	285,0000	0,0000	0,0000	0,0000%
29	25	-100,0000	5,0002	5,0887	1,7392%
30	25	-70,3846	5,0001	5,0120	0,2369%

31	25	-40,7692	5,0001	5,0120	0,2369%
32	25	-11,1538	4,1416	4,1425	0,0227%
33	25	18,4615	0,0000	0,0000	0,0000%
34	25	48,0769	-1,2876	-1,2782	-0,7328%
35	25	77,6923	-10,3182	-10,3329	0,1423%
36	25	107,3077	-23,8597	-24,0293	0,7058%
37	25	136,9231	-23,8597	-24,0293	0,7058%
38	25	166,5385	-23,8597	-24,0293	0,7058%
39	25	196,1538	-23,8597	-24,0293	0,7058%
40	25	225,7692	-23,8597	-24,0293	0,7058%
41	25	255,3846	-23,8597	-24,0293	0,7058%
42	25	285,0000	0,0000	0,0000	0,0000%
43	35	-100,0000	15,0000	15,0000	0,0000%
44	35	-70,3846	15,0000	15,0000	0,0000%
45	35	-40,7692	15,0000	15,0000	0,0000%
46	35	-11,1538	15,0000	15,0000	0,0000%
47	35	18,4615	15,0000	15,0000	0,0000%
48	35	48,0769	12,4561	12,4486	-0,0600%
49	35	77,6923	-6,9844	-6,9786	-0,0835%
50	35	107,3077	-24,1511	-24,3243	0,7120%
51	35	136,9231	-24,1511	-24,3243	0,7120%
52	35	166,5385	-24,1511	-24,3243	0,7120%
53	35	196,1538	-24,1511	-24,3243	0,7120%
54	35	225,7692	-24,1511	-24,3243	0,7120%
55	35	255,3846	-24,1333	-24,3062	0,7114%
56	35	285,0000	0,0000	0,0000	0,0000%
57	45	-100,0000	15,0000	15,0000	0,0000%
58	45	-70,3846	15,0000	15,0000	0,0000%
59	45	-40,7692	15,0000	15,0000	0,0000%
60	45	-11,1538	15,0000	15,0000	0,0000%
61	45	18,4615	15,0000	15,0000	0,0000%
62	45	48,0769	12,4561	12,4486	-0,0600%
63	45	77,6923	-6,9844	-6,9786	-0,0835%
64	45	107,3077	-24,1511	-24,3243	0,7120%
65	45	136,9231	-24,1511	-24,3243	0,7120%
66	45	166,5385	-24,1511	-24,3243	0,7120%
67	45	196,1538	-24,1511	-24,3243	0,7120%
68	45	225,7692	-24,1511	-24,3243	0,7120%
69	45	255,3846	-24,1333	-24,3062	0,7114%
70	45	285,0000	0,0000	0,0000	0,0000%
71	55	-100,0000	22,7931	22,9640	0,7444%
72	55	-70,3846	24,1511	24,3243	0,7120%
73	55	-40,7692	24,1511	24,3243	0,7120%
74	55	-11,1538	23,9674	24,1384	0,7085%
75	55	18,4615	24,1511	24,3243	0,7120%
76	55	48,0769	20,8304	21,0211	0,9073%
77	55	77,6923	15,0000	15,0000	0,0000%
78	55	107,3077	5,0002	5,0077	0,1506%
79	55	136,9231	-5,0002	-5,0077	0,1506%
80	55	166,5385	-6,1496	-6,1621	0,2036%
81	55	196,1538	-15,0000	-15,0000	0,0000%
82	55	225,7692	-15,0000	-15,0000	0,0000%
83	55	255,3846	-15,0000	-15,0000	0,0000%
84	55	285,0000	0,0000	0,0000	0,0000%
85	65	-100,0000	22,7931	22,9640	0,7444%
86	65	-70,3846	24,1511	24,3243	0,7120%
87	65	-40,7692	24,1511	24,3243	0,7120%
88	65	-11,1538	23,9674	24,1384	0,7085%
89	65	18,4615	24,1511	24,3243	0,7120%
90	65	48,0769	20,8304	21,0211	0,9073%
91	65	77,6923	15,0000	15,0000	0,0000%
92	65	107,3077	5,0002	5,0077	0,1506%
93	65	136,9231	-5,0002	-5,0077	0,1506%
94	65	166,5385	-6,1496	-6,1621	0,2036%
95	65	196,1538	-15,0000	-15,0000	0,0000%
96	65	225,7692	-15,0000	-15,0000	0,0000%

97	65	255,3846	-15,0000	-15,0000	0,0000%
98	65	285,0000	0,0000	0,0000	0,0000%
99	75	-100,0000	22,7931	22,9640	0,7444%
100	75	-70,3846	23,8597	24,0293	0,7058%
101	75	-40,7692	23,8597	24,0293	0,7058%
102	75	-11,1538	23,8597	24,0293	0,7058%
103	75	18,4615	23,8597	24,0293	0,7058%
104	75	48,0769	23,8519	24,0219	0,7077%
105	75	77,6923	20,2791	20,4691	0,9281%
106	75	107,3077	5,0001	5,0120	0,2369%
107	75	136,9231	5,0001	5,0120	0,2369%
108	75	166,5385	4,2671	4,2881	0,4886%
109	75	196,1538	-5,0001	-5,0120	0,2369%
110	75	225,7692	-5,0001	-5,0120	0,2369%
111	75	255,3846	-5,0001	-5,0120	0,2369%
112	75	285,0000	0,0000	0,0000	0,0000%
113	85	-100,0000	22,7931	22,9640	0,7444%
114	85	-70,3846	24,3553	24,5309	0,7159%
115	85	-40,7692	24,8161	25,0065	0,7613%
116	85	-11,1538	23,9674	24,1384	0,7085%
117	85	18,4615	24,7216	24,9064	0,7419%
118	85	48,0769	23,8519	24,0219	0,7077%
119	85	77,6923	21,0698	21,2700	0,9414%
120	85	107,3077	5,0001	5,0094	0,1857%
121	85	136,9231	5,0001	5,0134	0,2662%
122	85	166,5385	4,4472	4,4630	0,3534%
123	85	196,1538	-5,0001	-5,0099	0,1957%
124	85	225,7692	-5,0001	-5,0100	0,1976%
125	85	255,3846	-5,0001	-5,0069	0,1363%
126	85	285,0000	0,0000	0,0000	0,0000%
127	95	-100,0000	22,7931	22,9640	0,7444%
128	95	-70,3846	24,3553	24,5309	0,7159%
129	95	-40,7692	24,9580	25,1638	0,8179%
130	95	-11,1538	23,9674	24,1384	0,7085%
131	95	18,4615	24,7216	24,9064	0,7419%
132	95	48,0769	23,8519	24,0219	0,7077%
133	95	77,6923	21,0698	21,2700	0,9414%
134	95	107,3077	5,0001	5,0094	0,1857%
135	95	136,9231	5,0001	5,0134	0,2662%
136	95	166,5385	4,4472	4,4630	0,3534%
137	95	196,1538	-5,0001	-5,0099	0,1957%
138	95	225,7692	-5,0001	-5,0077	0,1522%
139	95	255,3846	-5,0001	-5,0069	0,1363%
140	95	285,0000	0,0000	0,0000	0,0000%

Tabela 6.3: Comparação de resultados obtidos pela simulação de um algoritmo executada no sistema Nébula e no Matlab

Por fim, em complementação aos cálculos de erro percentual médio absoluto para medir a discrepância entre os valores apurados pelas diferentes ferramentas, realizou-se um estudo estatístico a partir do cálculo T-Student pareado. Esta técnica permite validar a hipótese nula, que sugere a inexistência de diferenças significativas entre os valores apurados, ou que a diferença exista dentro de uma medida de tolerância aceitável. Isto é, espera-se que os valores obtidos a partir do Sistema Nébula sejam equivalentes aos valores obtidos sob as mesmas condições na ferramenta Matlab. O cenário de teste pareado, por sua vez, caracterizou-se em função da dependência das

amostras, uma vez que cada conjunto de valores foi submetido para cálculo por ambas as ferramentas.

Desta forma, dois estudos estatísticos foram realizados no conjunto de dados amostrais descritos na Tabela 6.3. O primeiro estudo consistiu em realizar uma análise estatística a partir de uma distribuição estratificada destas amostras, agrupadas em conjuntos de maneira que o valor do primeiro parâmetro de entrada permanece constante, enquanto o segundo parâmetro tem seus valores variantes. Este agrupamento estratificado gerou conjuntos com 14 amostras cada, e o *valor p* foi obtido utilizando-se o cálculo T-Student pareado. Observa-se portanto que, de acordo com os resultados para o *valor p* comparativamente à tabela de referência T-Student, não há evidências para rejeitar a hipótese-nula, em que os valores apurados por ambas as ferramentas são iguais, considerando-se um nível de significância de 99% ($\alpha=0,01$).

Testes Student Estratificados	
<i>Intervalo amostras</i>	<i>valor p</i>
1 - 14	0,025125681
15 - 28	0,02561006
29 - 42	0,026043808
43 - 56	0,007976581
57 - 70	0,007976581
71 - 84	0,00929573
85 - 98	0,00929573
99 - 112	0,003028157
113 - 126	0,002944197
127 - 140	0,002913838

Tabela 6.4: Estudo estatístico estratificado para o conjunto amostral, utilizando T-Student pareado

Adicionalmente, calculou-se o resultado do valor *p* considerando-se a totalidade do conjunto de valores, composto por 140 amostras, onde se obteve $p=0,517216$, evidenciando a inexistência de variações significativas.

Portanto, pode-se concluir que os resultados fornecidos pelo sistema Nébula, em igualdade de condições para os parâmetros de configuração, são equivalentes aos obtidos pelo Matlab, demonstrando que não foram observadas diferenças significativas entre os resultados obtidos, com nível de confiança de 99%, segundo o teste T-Student.

6.3.4 Considerações Parciais

A partir do conjunto de testes realizados, pôde-se validar a precisão matemática satisfatoriamente apresentada pelo sistema Nébula durante o processamento de um conjunto amostral de valores de entrada, em seus correspondentes resultados inferidos. Ao executar um conjunto de simulações em lote, submetendo uma matriz de 140 pares de valores de entrada para processamento pelo algoritmo de inferência, quando comparados com os mesmos resultados obtidos por um modelo semelhante desenvolvido no software Matlab, obteve-se como índice para o erro médio percentual absoluto o valor de 0,4398%, denotando uma precisão significativamente satisfatória para a natureza dos problemas que serão desenvolvidos com a Plataforma Nébula. Em complementação, testes estatísticos estratificados de conjuntos amostrais foram realizados com a técnica de cálculo T-Student Pareado, evidenciando a inexistência de diferenças significativas entre os valores apurados pelo sistema Nébula e pelo Matlab, considerando-se o nível de confiança em 99% segundo a tabela T-Student.

Com base nestes resultados, pode-se afirmar que o núcleo matemático de processamento de inferências fornecido pela plataforma Nébula é capaz de fornecer resultados precisos, com base nas definições dos parâmetros de configuração para os algoritmos de inferências modelados em sua plataforma. Esta afirmativa satisfaz uma das premissas deste projeto, que preconiza a confiabilidade dos resultados fornecidos pelo núcleo de processamento de inferências, em conformidade com os parâmetros de configuração para o algoritmo nebuloso em questão.

Diante deste cenário, é relevante destacar uma das vantagens do sistema Nébula, caracterizada por uma de suas principais contribuições: a simplicidade de uso. Ao fornecer uma interface mais intuitiva para utilização pelo usuário, o Nébula substitui a necessidade de utilização de ferramentas complexas, mais voltadas para perfis especialistas das áreas de ciências exatas, como é o caso do Matlab, ao fornecer um conjunto de funcionalidades voltadas mais especificamente para a solução de problemas referentes aos jogos psicopedagógicos. O mesmo algoritmo, que no Matlab envolveu o desenvolvimento de código-fonte, a utilização do sistema Nébula viabiliza a participação por não-especialistas em matemática e computação, abstraindo representações matemáticas em gráficos e expressões mais simplificadas.

6.4 Estudo de Caso Prático Exploratório com Grupo de Especialistas

6.4.1 Conceitos Gerais

O presente estudo de caso consiste em realizar um estudo prático propondo a utilização do sistema Nébula por um grupo de usuários, composto essencialmente pelos diferentes perfis de especialidade presentes nas equipes multidisciplinares de neurocientistas, participantes do projeto de virtualização de jogos psicopedagógicos, com o objetivo de validar a hipótese de sua viabilidade de uso, caracterizada essencialmente por seu critério de usabilidade. Este quesito será validado a partir do registro das percepções, opiniões e comentários dos usuários, além de características percebidas pela observação prática no decorrer do estudo de caso.

O Sistema Nébula objetiva permitir que, embora as equipes sejam compostas por diferentes especialidades, e considerando a participação de não-especialistas em lógica nebulosa nos grupos de trabalho, o conhecimento comum do grupo seja suficiente para utilização da ferramenta, que os conduzirá em uma sequência de passos na formalização de seu raciocínio para descrever o comportamento do algoritmo de inferências. A premissa considerada é que os múltiplos conhecimentos presentes em cada especialidade se complementarão, e que o conhecimento do grupo seja suficiente para permitir que pedagogos, psicólogos e lingüistas atuem conjuntamente com cientistas da computação, concentrando na ferramenta suas tarefas de modelagem do algoritmo.

Desta forma, a meta considerada para a realização dos objetivos consiste em permitir que os usuários disponham de uma interface integrada para a completa modelagem do algoritmo de inferências, incluindo a especificação de seus parâmetros de configuração e regras, a realização de simulações e testes, e a correspondente alteração de seus parâmetros de sintonização, de forma desacoplada da construção e da execução dos jogos.

Portanto, a metodologia proposta para este estudo de caso se concentra em apresentar a um grupo de especialistas um modelo de avaliação cognitiva previamente especificado, inserido no contexto de um jogo psicopedagógico escolhido para este

estudo, e solicitá-los que realizem na ferramenta Nébula a criação de uma instância de inferências, de acordo com as premissas estabelecidas por este modelo de avaliação.

O modelo de avaliação e o jogo psicopedagógico escolhidos para a realização deste estudo correspondem às funções cognitivas de Atenção, cujos critérios foram propostos pelo grupo de trabalho interdisciplinar e estabelecidos durante a disciplina de Neuropedagogia e Informática II, do curso de mestrado em informática do INCE, e foram discutidos e validados conjuntamente em sala de aula.

Ressalva-se que o estudo não se propõe a validar o modelo em si, nem tampouco os requisitos considerados no jogo de atenção utilizado, mas sim a capacidade do Sistema Nébula em representá-lo, a partir da definição dos parâmetros de configuração apropriados para um sistema de inferências nebuloso, de acordo com as percepções e conhecimento do especialista que irá modelá-lo.

6.4.1.1 Modelo de Avaliação Cognitiva para as Funções de Atenção e o Jogo Psicopedagógico ‘Monta Boneco’

Neste contexto, o modelo de avaliação cognitiva considerado neste estudo de caso corresponde às funções cognitivas de Atenção: Atenção Focal, Seletiva, Dividida e Alternada, cujos termos podem ser assim definidos:

- Atenção Focal (ou concentrada) é a focalização da atenção a um determinado objeto ou a uma ação (Brickenkamp, 2002);
- Atenção Seletiva é definida pela capacidade do indivíduo privilegiar determinados estímulos em detrimento de outros, ou seja, está ligada ao mecanismo básico que subsidia o mecanismo atencional (Lima, 2005);
- Atenção Alternada (ou alternância da atenção) é “a capacidade do indivíduo em alternar o foco atencional, ou seja desengajar o foco de um estímulo e engajar em outro.” (Lima, 2005)
- Atenção Dividida (ou distributiva) é quando esta é dividida para o desempenho de duas tarefas simultaneamente. (Lima, 2005). Segundo Lima (2005) “a atenção dividida é caracterizada por um processamento automático, ao contrário de outros tipos de atenção que ocorre por processamento controlado e, portanto, consciente do sujeito”.

Neste contexto, pode-se definir ainda a Atenção Sustentada, que representa a capacidade do indivíduo em mantê-la focalizada em uma sequência de estímulos, por um período de tempo, para conseguir desempenhar determinada tarefa, sendo caracterizada por uma habilidade em detectar estímulos (nível de vigilância) e por uma diminuição no desempenho ao longo do tempo (Sternberg, 2000; Sarter, Givens & Bruno, 2001).

A Figura 6.18, elaborada pelo grupo de trabalho durante a disciplina de Neuropedagogia e Informática II, ilustra resumidamente um mapa conceitual, correlacionando os termos envolvidos na Função Cognitiva de Atenção.

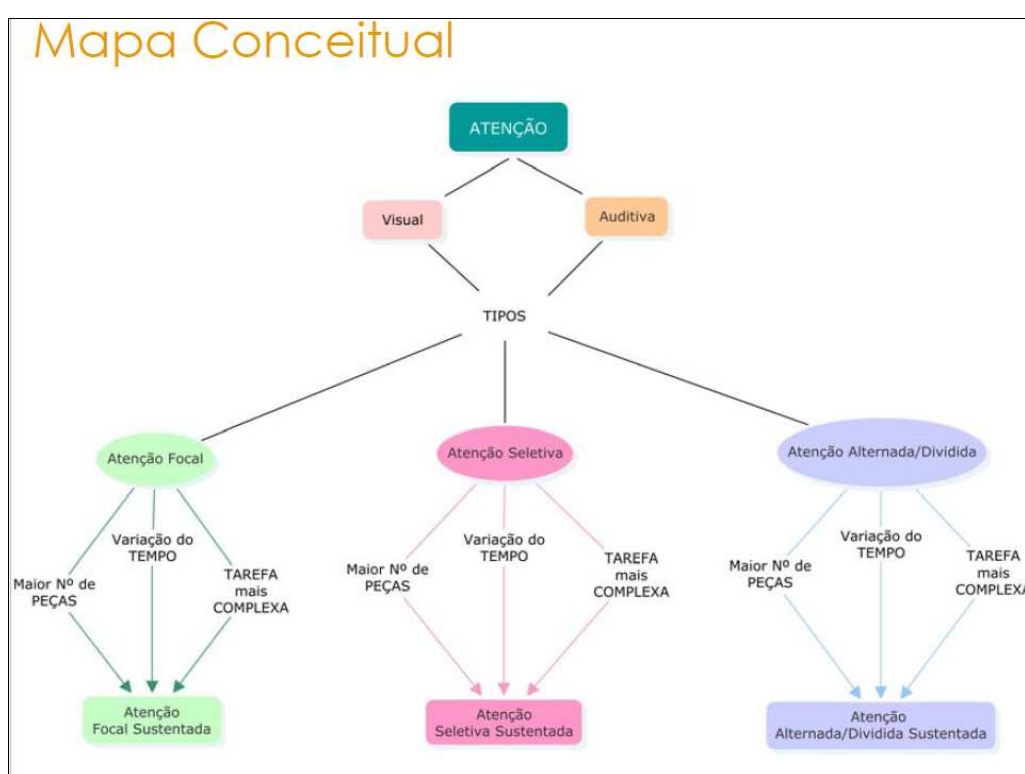


Figura 6.18: Mapa conceitual dos termos envolvidos na Função Cognitiva de Atenção

O modelo de avaliação cognitiva em questão propõe a utilização de funções matemáticas para calcular o nível de atenção em que a criança está situada, calculados essencialmente a partir de uma proporção direta entre a precisão dos resultados aferidos, pela rapidez das respostas. Estas funções estabelecem índices numéricos para posicionar o desempenho da criança durante o jogo, no contexto da função cognitiva analisada. Este índice obtido, portanto, pode ser compreendido como um coeficiente, ou *escore*, cujo significado deve ser apropriado a partir de uma comparação com uma tabela de referência previamente estabelecida. No entanto, esta tabela de referência, que seria

criada a partir de um conjunto de experimentos realizados com grupos de controle, não está ainda disponível. Desta forma, na ausência de significância diretamente compreensível para estes coeficientes, o estudo de caso propõe ao especialista especificar uma instância de inferências que represente sua percepção acerca dos critérios apropriados para a avaliação destas funções cognitivas, tendo como fundamento os critérios estabelecidos pelo modelo de avaliação considerado.

O modelo de cálculo para avaliação das funções cognitivas de atenção pode ser resumidamente observado a partir da ilustração descrita na Figura 6.19, originalmente apresentado na proposta para o modelo de avaliação cognitiva das funções de atenção.

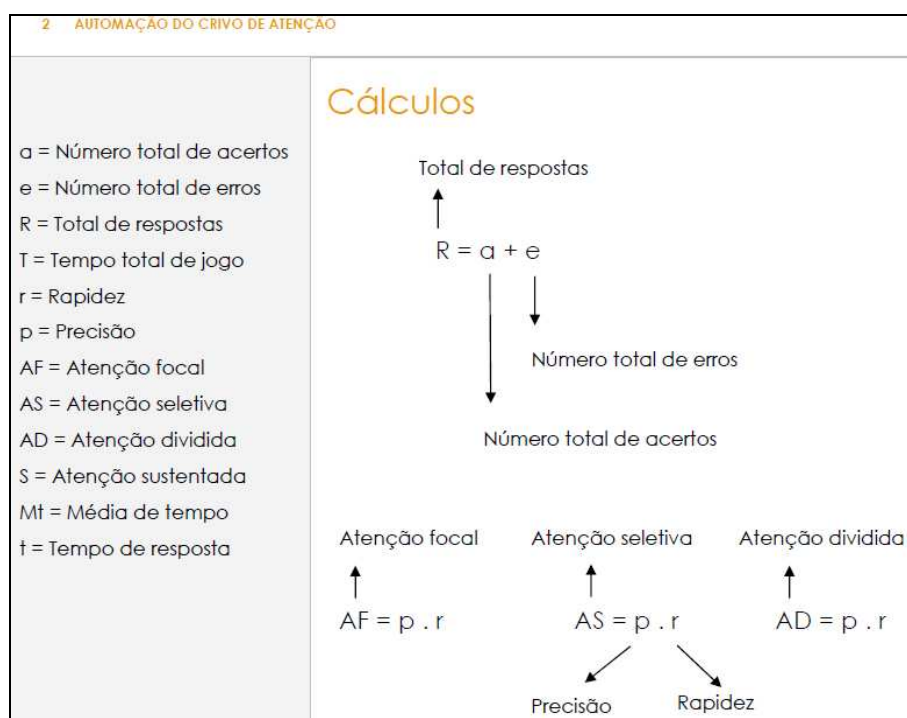


Figura 6.19: Representação das funções para cálculo dos coeficientes de avaliação de atenção

Os resultados obtidos pelo algoritmo de inferências modelado pelo especialista no Sistema Nébulas podem ser observados na própria interface da ferramenta integrada, a partir do Módulo de Simulação. Porém, para tornar sua percepção mais facilmente observável pelo especialista durante o experimento, propôs-se utilizar um jogo psicopedagógico previamente estabelecido, de forma que a instância de inferências recém-criada pudesse ser executada dentro do contexto do jogo, e seus resultados observados em tempo real, durante a sessão de jogo.

Desta forma, escolheu-se o jogo Monta Boneco, também criado pelo grupo de tecnologia durante a disciplina de Neuropedagogia e Informática II, e descrito no trabalho de Ferreira (2009). A Figura 6.20 ilustra as principais características de tela e regras propostas para este jogo.

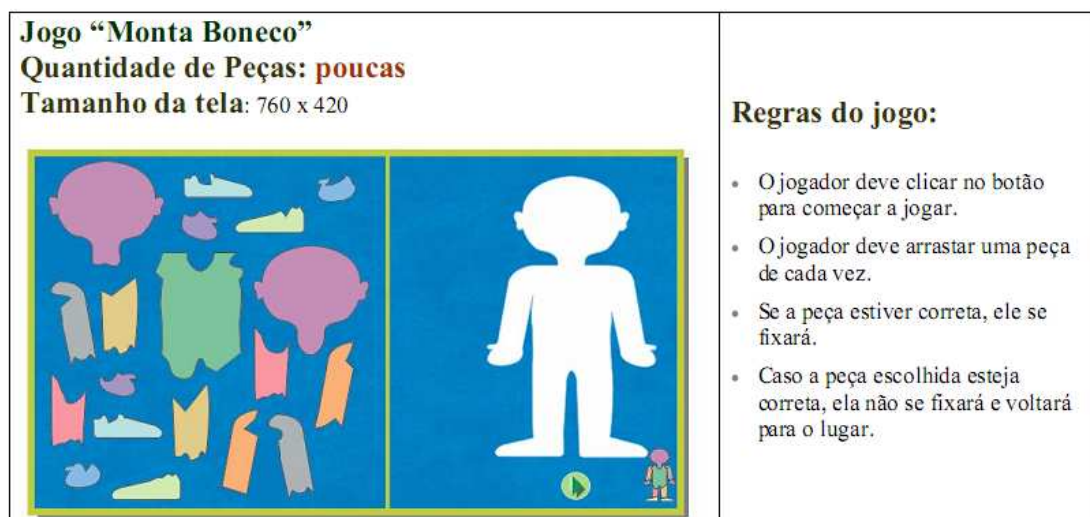


Figura 6.20: Modelo de tela e regras propostos para o Jogo “Monta Boneco”

6.4.1.2 Modelo de Referência Proposto para as Configurações da Inferência

Para auxiliar o especialista na especificação dos parâmetros iniciais de configuração do algoritmo de inferências a ser utilizado no jogo, um modelo de referência foi sugerido, para servir-lhe de orientação para a construção de seu raciocínio na elaboração das regras e especificação dos intervalos para os conjuntos nebulosos.

Embora durante o experimento o especialista tenha sido orientado a se expressar livremente, um modelo de referência mostrou-se particularmente útil ao apresentar os conceitos envolvidos no sistema, considerando-se que a demonstração a partir de exemplos em muitos casos mostra-se como uma das melhores maneiras de se transmitir um conhecimento, ou explicar determinados conceitos.

Portanto, uma vez que o modelo de avaliação cognitiva a ser utilizado, descrito na seção 6.4.1.1 considera para cálculo do desempenho para a função de Atenção as

variáveis ‘Tempo’ e ‘Precisão’, as partições nebulosas e seus intervalos sugeridos para estas variáveis encontram-se descritos na Figura 6.21 e Figura 6.22, respectivamente.

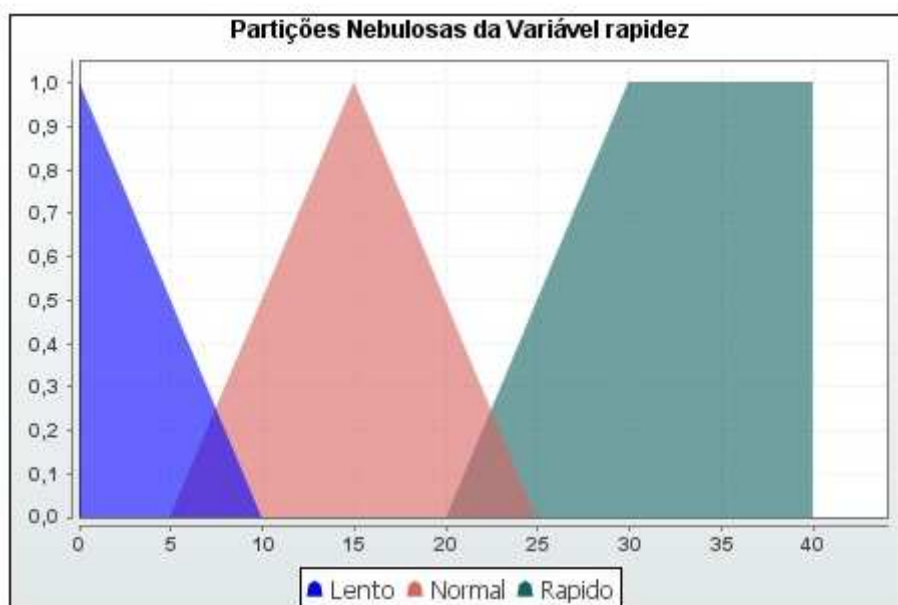


Figura 6.21: Partições sugeridas para a variável ‘rapidez’

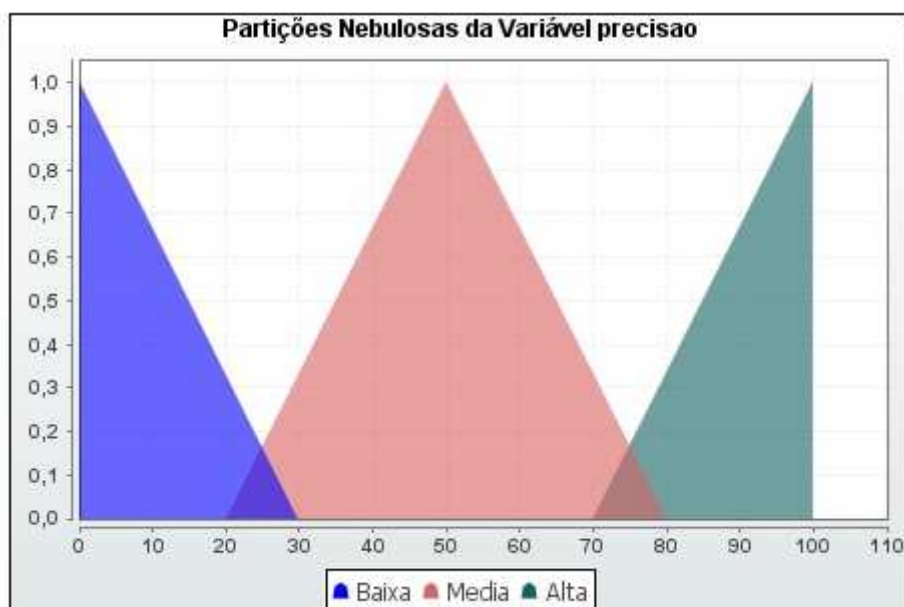


Figura 6.22: Partições sugeridas para a variável ‘precisão’

Para a variável de saída a ser inferida, aqui denominada ‘desempenho’, que objetiva representar o nível obtido durante a sessão de jogo para a função cognitiva em

questão, arbitrou-se um intervalo de variação estabelecido em termos percentuais, em uma escala de 0 a 100, conforme descrito na figura Figura 6.23.

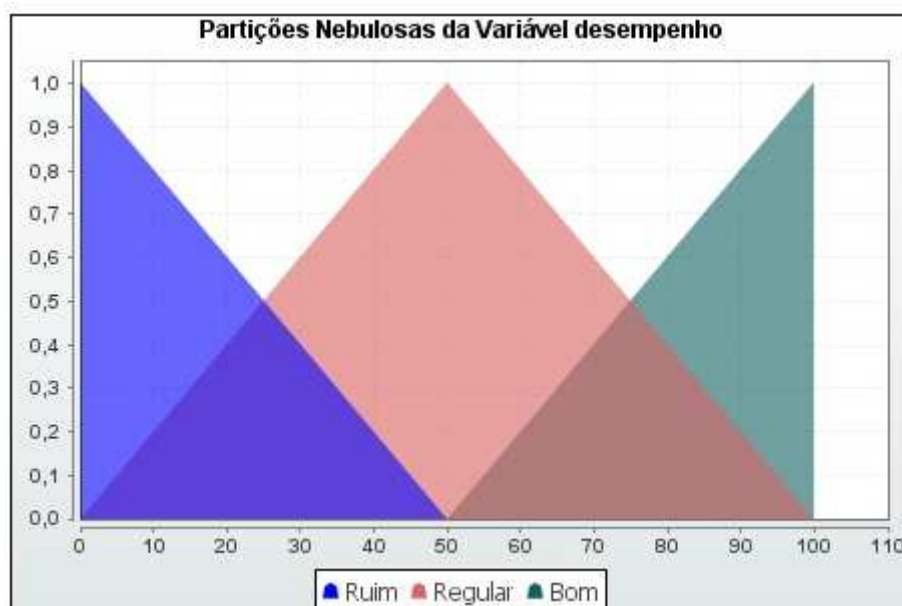


Figura 6.23: Partições sugeridas para a variável de saída ‘desempenho’

Para a base de regras a ser formada, que correlacionam as partições das variáveis de entrada de maneira causal com as partições da variável de saída, uma sugestão para sua composição encontra-se descrita na Tabela 6.5. Porém, durante o experimento, o especialista teve a liberdade de estabelecer livremente o conjunto de regras a ser formado, a partir de seu conhecimento e de suas opiniões, optando por utilizar a tabela sugerida apenas como exemplificação do formato apropriado a ser seguido. O mesmo se aplica para as sugestões de referência sobre as partições nebulosas das variáveis de entrada e de saída.

<i>Tempo</i> <i>Precisão</i>	Lento	Normal	Rápido
Baixa	Ruim	Ruim	Ruim
Médio	Ruim	Regular	Regular
Alta	Regular	Bom	Bom

Tabela 6.5: Tabela de referência para a base de regras

6.4.2 Metodologia para Análise de Usabilidade da Interface do Sistema Nébula

A fim de estruturar uma metodologia para padronizar a medição das opiniões dos usuários a respeito da ferramenta durante o experimento, propôs-se um roteiro para orientar uma entrevista, organizado na forma de um questionário, a fim de servir como protocolo para a avaliação, fundamentado nas especificações descritas pela Norma ISO 9241, que define o termo usabilidade como "a capacidade de um produto ser usado por usuários específicos para atingir objetivos específicos com eficácia, eficiência e satisfação em um contexto específico de uso" (ABNT, 2002). Resumidamente, pode-se conceituar o termo usabilidade como os atributos de um produto que o torna mais fácil de usar.

Neste contexto, um índice de usabilidade satisfatório é alcançado quando os objetivos do sistema são realizados com eficácia, eficiência e satisfação. Como estes contextos são de natureza subjetiva, de difícil mensuração direta, optou-se, conforme as recomendações da norma ISO 9241, obtê-las a partir da percepção dos usuários, colhidas a partir de um formulário de avaliação.

É relevante conceituar, neste ponto, os termos eficácia, eficiência e satisfação, no contexto tratado pela Norma ISO 9241, que orienta a condução deste estudo de caso. Segundo a norma (ABNT, 2002), estes termos podem ser assim definidos:

- **Eficácia:** refere-se à acurácia ou completude sobre como os objetivos podem ser alcançados. No contexto do sistema Nébula, este objetivo se refere à capacidade do sistema em permitir ao usuário a construção de algoritmos de inferência nebulosos para os jogos psicopedagógicos, permitindo-os representar graficamente e de maneira simples os conceitos matemáticos envolvidos.
- **Eficiência:** relacionam o nível de eficácia alcançada, comparativamente ao dispêndio de recursos necessários para a realização do objetivo. Estes recursos podem ser mentais ou físicos, tempo ou custos financeiros. No contexto deste estudo, esta medida se refere ao esforço mental e de tempo envolvidos na tarefa de especificação de um algoritmo de inferências nebuloso, comparativamente à situação normal que exigiria o uso de ferramentas de engenharia, como Matlab, e ao desenvolvimento de códigos em linguagens de

programação específicas, a serem acoplados aos jogos. A medida de eficiência se refere à economia de esforços referentes à desacoplagem do algoritmo ao código-fonte do jogo, e as conseqüentes tarefas de desenvolvimento, compilação, testes e publicação, inerentemente envolvidos no processo tradicional, conforme descritos na Figura 1.1.

- **Satisfação:** refere-se à forma pela qual os usuários estão livres de desconforto e suas atitudes em relação ao uso do produto. Portanto, pode ser especificada pela avaliação subjetiva pelo nível de conforto, gosto ou aceitação experimentado pelo usuário durante o uso do produto para realização de suas tarefas. No contexto do sistema Nébulas, este item se refere à maneira como o sistema representa a informação, abstraindo a complexidade matemática inerentemente envolvida nos processos tradicionais, e presentes nas demais ferramentas e linguagens de programação.

Em suma, pode-se dizer que a usabilidade do sistema pode ser mensurada a partir da percepção dos usuários durante a utilização do software, por sua facilidade de aprendizado, disponibilidade de recursos e pelo atendimento de seus objetivos.

Portanto, para mensurar estas percepções do usuário sobre os critérios de eficiência, eficácia e satisfação, estabeleceu-se um conjunto de questões com o objetivo de atuar como um protocolo semi-estruturado para registro de suas percepções durante o estudo de caso exploratório, orientando a condução da observação prática. Estas perguntas foram estruturadas em forma de um roteiro de entrevista, organizado na forma de um questionário, com respostas objetivas, baseadas na escala Likert, em função de sua eficiência em coletar o grau de intensidade da opinião sobre um determinado assunto (PARKER, 2000). Para complementar o registro de informações com mais riqueza de detalhes, algumas questões adicionais com respostas livres foram incluídas, com preenchimento opcional, para que os respondentes pudessem manifestar seus comentários e críticas sobre a utilização do sistema, buscando obter informações adicionais que pudessem colaborar com sua elaboração ou melhorias futuras.

Desta forma, as perguntas foram elaboradas com polaridade positiva, com respostas organizadas em cinco categorias, variando entre os valores 0 a 4, assim definidas:

0. Discordo totalmente;

1. Discordo parcialmente;
2. Estou indeciso;
3. Concordo parcialmente;
4. Concordo Totalmente.

Este protocolo de entrevista, disponível no Apêndice, procura avaliar a maneira como o usuário interpretou as facilidades oferecidas pela ferramenta, em uma visão comparativa às demais opções normalmente realizadas pelo modelo tradicional de desenvolvimento de algoritmos de inferência, utilizando-se de ferramentas de engenharia, como o Matlab, e de técnicas dependentes de desenvolvimento de código em linguagens de programação.

Portanto, as perguntas foram especificadas de maneira a auxiliar a avaliação das opiniões do usuário no cumprimento dos objetivos propostos pelo sistema Nébulas, no contexto dos critérios de usabilidade considerados. A matriz de referência que relaciona as perguntas com os objetivos e os critérios analisados encontra-se descrita na Tabela 6.6.

Objetivo Geral	Critério	Questão
Permitir a construção de algoritmos de inferência em uma interface integrada	Eficácia	1. A interface do sistema Nébulas facilita o usuário no desenvolvimento do algoritmo, orientando por uma sequência de passos.
Simplificar as representações matemáticas complexas em gráficos	Eficácia Satisfação	2. As facilidades oferecidas através de gráficos, tabelas, dicas e menus contribuíram para a compreensão das representações matemáticas envolvidas no processo de construção dos algoritmos.
Permitir o envolvimento dos participantes de diferentes especialidades das equipes multidisciplinares	Eficácia	3. A utilização do sistema Nébulas permite melhorar a integração das equipes multidisciplinares, ao atuarem conjuntamente na definição dos algoritmos.

Permitir a simulação dos algoritmos criados	Eficácia	4. A funcionalidade de simulação auxiliou no processo de testes, validação e sintonização dos parâmetros do algoritmo.
Permitir a sintonização dos parâmetros do algoritmo após testes	Eficácia	5. Foi possível realizar alterações nos parâmetros de sintonização do algoritmo a partir do sistema Nébula, e percebê-las imediatamente no jogo em execução.
Permitir o desacoplamento das definições do algoritmo do código, verificando como as alterações nos parâmetros refletem no jogo em execução	Eficiência	
Simplificação do processo de desenvolvimento, em relação ao modelo tradicional de desenvolvimento de software	Eficiência	6. A utilização do sistema Nébula melhorou o processo de desenvolvimento dos algoritmos utilizados nos jogos psicopedagógicos, em relação ao modelo tradicional.
	Satisfação	7. Foi percebida vantagem no sistema Nébula, em comparação às demais opções que envolvem o uso de ferramentas de engenharia (como o Matlab) e a utilização de técnicas que exigiam desenvolvimento de código (Java, C, etc). 8. Os resultados da inferência estavam de acordo com o esperado.
	Satisfação	9. Você se sentiu confortável em utilizar o sistema?
	Satisfação	10. O que você sentiu falta na ferramenta? Quais os principais problemas identificados? 11. Comentários

Tabela 6.6: Matriz de referência das perguntas do roteiro para avaliação dos critérios analisados

6.4.3 Metodologia para Realização do Experimento

Em suma, o estudo de caso propõe que o especialista possa realizar na ferramenta integrada Nébula a modelagem de uma instância de inferências, em

conformidade com as diretrizes estabelecidas pelo modelo de avaliação cognitiva, descrevendo os parâmetros de configuração conforme seu entendimento, e visualizando os resultados do algoritmo tanto na própria ferramenta quanto em uma sessão de jogo simulada.

Para viabilizar sua realização, o jogo ‘Monta Boneco’ foi adaptado para coletar, processar e exibir um conjunto de informações relevantes no contexto do estudo, e importantes para serem consideradas durante uma avaliação para a função cognitiva de atenção. Desta forma, foi criada uma console, orientada ao perfil de aplicador, que neste estudo, para finalidades didáticas, foi integrada à mesma interface do jogo. Em uma situação real de uso, esta console estaria disponível em um ambiente distinto daquele em que a sessão de jogo é executada. Esta console exibe um relatório consolidado, apresentando a situação presente das diversas variáveis e índices que estão sendo considerados pelo sistema, especialmente os resultados da inferência que estão sendo processados em tempo real.

O conjunto de variáveis coletadas considerada pelo jogo e exibidas pela console são:

- Quantidade de Acertos
- Quantidade de Erros
- Número de Perseverações (*erros consecutivos em uma mesma ação*)
- Quantidade Total de Respostas
- Tempo Total
- Tempo de Reação (*início do jogo até a primeira ação*)
- Tempo de Resposta (*instantâneo, entre cada ação*)
- Índice de Precisão (*acerto/total de respostas*)
- Índice de Rapidez (*respostas/tempo total*)
- Percentual de erros (*erros/total respostas*)
- Escore (*obtido pela multiplicação do índice de rapidez pelo índice de precisão*)
- Resultados da inferência

A console de relatórios também exibe algumas informações sobre o processamento do algoritmo de inferências, de acordo com os valores informados para as variáveis de entrada, em cada momento. Estas informações podem ser utilizadas para verificar a maneira como o algoritmo se comporta, e analisar a possibilidade de realizar modificações pontuais nos parâmetros de configuração. Estas informações da inferência exibidas pela console correspondem a:

- Índice numérico retornado como saída do processamento da inferência
- Graus de inclusão de Cada Partição
- Descrição das regras ativadas durante o processamento da inferência

Para abstrair a representação matemática dos índices precisos que são retornados pelo sistema de inferências para cada partição nebulosa da variável de saída, a console exibe seus resultados em forma de um gráfico estilizado, que descreve a situação de cada uma das partições de saída em um diagrama. Neste diagrama, cada uma destas partições está associada a um círculo, e o grau de inclusão de cada partição obtido na inferência é representado pela transparência ou opacidade do círculo a ela associado. Isto é, quando menor o grau de inclusão da partição, maior é a transparência do círculo que a descreve, de maneira que com o grau de inclusão zero o círculo não é exibido.

A Figura 6.24 ilustra a tela do Jogo “Monta Boneco” em uma execução simulada durante um dos experimentos, apresentando em destaque a console de relatório, apresentando a situação presente do conjunto de variáveis e o resultado do processamento da inferência para cada uma das partições da variável de saída, representadas graficamente.

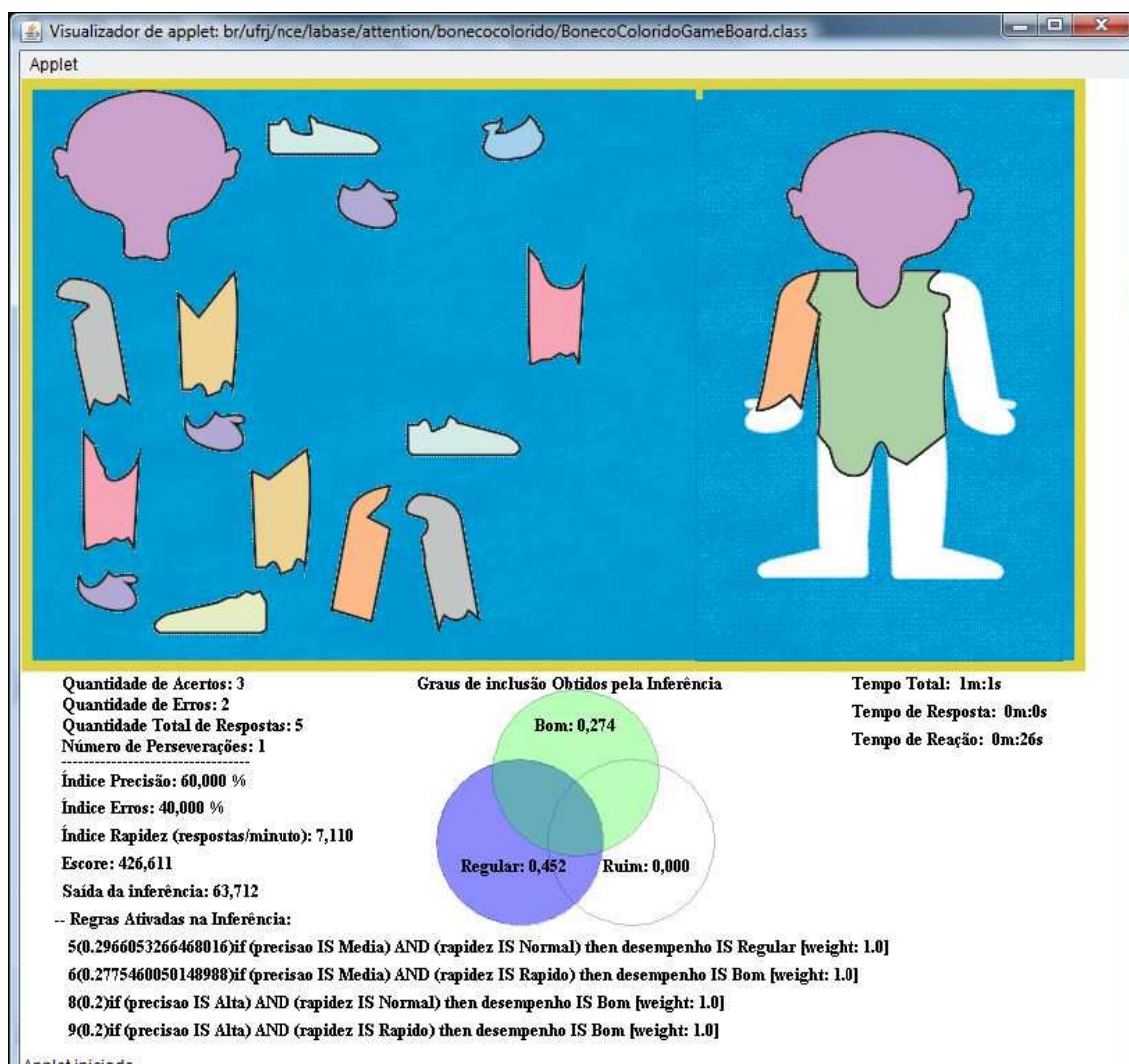


Figura 6.24: Jogo ‘Monta Boneco’ em execução, com destaque para a console de relatório

Neste cenário, o estudo de caso foi planejado e conduzido a partir de uma sequência de passos, assim descritos:

1. O especialista é inicialmente apresentado ao modelo de avaliação para a função cognitiva de Atenção, proposto pela equipe multidisciplinar, e ao jogo “Monta Boneco”;
2. Em seguida, é apresentado o Sistema Nébula, seus principais conceitos e objetivos, abordando uma breve introdução à Lógica Nebulosa;
 - 2.1. Apresenta-se brevemente o histórico sobre o processo tradicional de desenvolvimento de inferências para os jogos psicopedagógicos, destacando o problema a que este estudo se propõe a solucionar;

3. Então, o especialista é solicitado a realizar no sistema Nébula a modelagem de uma instância de inferências, para estimar o desempenho obtido, de acordo com o modelo apresentado e segundo seu entendimento pessoal;
 - 3.1. Os parâmetros de configuração devem ser *inicialmente estimados*, a partir do conhecimento do especialista, conforme preconizam as práticas para os sistemas de inferência;
 - 3.2. Um *modelo sugerido* é apresentado, para facilitar sua compreensão e conduzir o experimento. Todavia, o especialista tem liberdade para expressar livremente seu entendimento sobre a modelagem do algoritmo.
4. O especialista é apresentado ao *módulo de simulação*, e solicitado a realizar testes na inferência recém-criada
 - 4.1. No decorrer do estudo, o especialista pode realizar testes e modificações nos parâmetros de configuração, sintonizando-os à medida que os resultados são observados;
5. O jogo de atenção é executado, e o especialista simula a sua utilização, no intuito de verificar na prática os reflexos do modelo de inferência que ele criou na ferramenta Nébula;
 - 5.1. A console de relatório é apresentada, destacando o conjunto de variáveis coletadas e processadas, explicando sobre os resultados obtidos pela inferência;
 - 5.2. O especialista é estimulado a alterar alguns parâmetros de configuração no sistema Nébula, e observar como essas alterações se refletem no algoritmo do jogo em execução;
6. O Questionário de avaliação é apresentado para registrar suas percepções, críticas e comentários.

6.4.4 Considerações Parciais

O estudo de caso prático exploratório foi realizado com quatro indivíduos, com diferentes perfis especialistas, todos participantes do grupo de pesquisa do projeto de virtualização de jogos psicopedagógicos, conduzido pelo curso de mestrado em informática do INCE/PPGI, da Universidade Federal do Rio de Janeiro. Os perfis especialistas envolveram as áreas de psicologia, letras, pedagogia e informática.

A metodologia para a realização do estudo de caso consistiu em fazer uma breve explicação dos conceitos básicos sobre lógica nebulosa e o que é um algoritmo de inferência, seus resultados esperados, as variáveis envolvidas e suas representações gráficas para as suas partições. Como a ferramenta objetiva que os profissionais de diferentes especialidades atuem em conjunto, incluindo tanto profissionais de educação como de ciências exatas, atuando em conjunto, permitiu-se realizar o experimento na forma de uma sessão dirigida, onde o usuário operava o sistema modelando os parâmetros de acordo com o seu entendimento, recorrendo ao mediador para discutir sobre as melhores opções para a modelagem e esclarecimentos de eventuais dúvidas. O processo de modelagem do algoritmo foi discutido em conjunto durante a realização do experimento, como assim o seria em uma situação real de uso.

A Tabela 6.7 descreve resumidamente a totalização de respostas para cada item, para cada grau de avaliação correspondente.

Questão	0- Discordo Totalmente	1- Discordo Parcialmente	2- Indeciso	3- Concordo Parcialmente	4- Concordo Totalmente
1. A interface do sistema Nébulas facilita o usuário no desenvolvimento do algoritmo, orientando-o por uma sequência de passos.				1	3
2. As facilidades oferecidas através de gráficos, tabelas, dicas e menus contribuíram para melhorar a compreensão das representações matemáticas envolvidas no processo de construção dos algoritmos.				1	3
3. A utilização do sistema Nébulas permite melhorar a integração das equipes multidisciplinares, ao atuarem conjuntamente na definição dos algoritmos.				1	3
4. A funcionalidade de simulação auxiliou no processo de testes, validação e sintonização dos parâmetros do algoritmo.					4
5. Foi possível realizar alterações nos parâmetros de sintonização do algoritmo a partir do sistema Nébulas, e percebê-las imediatamente no jogo em execução.				1	3
6. A utilização do sistema Nébulas melhorou o processo de desenvolvimento dos algoritmos utilizados nos jogos psicopedagógicos, em relação ao modelo tradicional.				1	3

7. Foi percebida vantagem no sistema Nébula, em comparação às demais opções que envolvem o uso de ferramentas de engenharia (como o Matlab) e a utilização de técnicas que exigiam desenvolvimento de código (Java, C, etc).				1	3
8. Os resultados da inferência estavam de acordo com o esperado.					4
9. Você se sentiu confortável em utilizar o sistema?					4

Tabela 6.7: Totalização de respostas obtidas para cada item do questionário

Em complementação às respostas objetivas, alguns registros escritos voluntariamente manifestados evidenciaram alguns pontos positivos do sistema Nébula e sua relevância, assim como demonstraram alguns pontos de atenção.

Primeiramente, alguns dos relatos nos campos de comentários expressaram-se positivamente, mencionando a relevância potencial percebida como contribuição pelo sistema, ao permitir diminuir a distância existente entre os perfis especialistas (pedagogos, psicólogos) e técnicos (matemáticos, analistas). Ao integrá-los em um ambiente comum, o sistema Nébula permitiu facilitar a comunicação entre estes distintos perfis, atuando conjuntamente no processo de construção da lógica para dar suporte à avaliação cognitiva. Percebeu-se também um potencial de aplicação para além dos jogos psicopedagógicos, de maneira que outras frentes de trabalho interessadas em utilizar lógica nebulosa como técnica para solucionar seus problemas propostos, poderiam fazer uso do Nébula para auxiliar no processo de modelagem e testes de seus algoritmos.

Em contrapartida, percebeu-se que alguns conceitos não foram facilmente percebidos, especialmente pelos participantes com perfis não-técnicos, como pedagogia e psicologia. Estas dificuldades se manifestaram principalmente no passo de descrição das partições nebulosas, quando alguns relataram não compreender exatamente o que significavam os gráficos. Isto poderia ser resolvido melhorando-se a interface do sistema, incluindo-se novos recursos de ajuda e explicações em tela, como também por dedicar um tempo maior no treinamento dos usuários, uma vez que a experimentação prática precisou ser feita em função das limitações de tempo disponível pelos participantes voluntários. Outro ponto relevante é considerar o envolvimento de diferentes especialidades nas equipes multidisciplinares, integrando técnicos e

especialistas, de maneira que o conhecimento do grupo se complemente e seja possível preencher os pré-requisitos necessários ao completo processo de modelagem dos algoritmos de inferência.

Desta forma, os resultados apresentados evidenciaram a aceitação do Sistema Nébula pelo público-alvo de especialistas, que conseguiram modelar na ferramenta o algoritmo de inferências propostos, integrá-lo a um jogo psicopedagógico e observaram em tempo real os resultados processados pela instância de inferências recém-criadas. Os recursos de simulação e de sintonização de parâmetros foram experimentados e analisados, e os registros obtidos a partir da observação prática e das respostas formalizadas nos questionários de avaliação evidenciaram que o sistema Nébula permitiu que usuários de diferentes áreas de especialidades realizassem a modelagem de algoritmos de inferência de maneira simples, utilizando a ferramenta integrada e abstraindo a complexidade das representações matemáticas, simplificando o modelo tradicional de desenvolvimento de algoritmos de inteligência computacional a serem incluídos nos jogos psicopedagógicos.

Capítulo VII

Considerações Finais

"Palavra puxa palavra, uma idéia traz outra, e assim se faz um livro, um governo, ou uma revolução, alguns dizem que assim é que a natureza compôs as suas espécies."

Machado de Assis

Neste capítulo é apresentado um resumo sobre o trabalho de pesquisa tratada nesta dissertação, destacando-se especialmente suas contribuições, limitações e perspectivas para o desenvolvimento de trabalhos futuros.

7.1 Resumo do Trabalho

O projeto de virtualização de jogos psicopedagógicos está inserido no escopo de desenvolvimento de novas metodologias pedagógicas para potencializar as práticas de avaliação, intervenção e reabilitação cognitivas, ao utilizar-se de recursos informatizados tanto como ferramentas de democratização do acesso aos jogos, como para automatizar o processamento de informações com o uso de algoritmos de inteligência computacional.

No entanto, em função de sua natureza multidisciplinar e de suas características inerentemente inexatas e imprecisas, a modelagem de mecanismos que auxiliem o processo de avaliação cognitiva em algoritmos matemáticos é uma tarefa complexa. Especialmente, ao considerarmos a interação de diversas especialidades presentes nas equipes, que definem o conjunto de regras que determinam o comportamento do algoritmo, torna-se necessário a participação de um especialista em informática, para traduzir estas definições em modelos computacionais, a partir de uma linguagem de programação.

Portanto, incorporar os algoritmos de inteligência computacional aos jogos é uma tarefa que demanda muito esforço, em um conjunto de passos que vai desde a determinação do conjunto de variáveis, parâmetros e regras envolvidos no problema, sua implementação em uma linguagem de programação, sua incorporação aos jogos e, por fim, a execução de diversos testes e ajustes, em sucessivas iterações até que se obtenha o comportamento desejado.

Este longo processo, em que a lógica do algoritmo está acoplada aos jogos, demanda muito retrabalho do especialista em informática ao constantemente alterar a programação e publicar os jogos, além de dificultar a participação das demais especialidades das equipes multidisciplinares.

Neste sentido, este projeto de pesquisa apresentou uma plataforma integrada, que permite a descrição dos algoritmos de uma maneira simplificada e desacoplada do processo de codificação dos jogos psicopedagógicos, utilizando-se de uma interface intuitiva que permite a participação de toda a equipe multidisciplinar na modelagem do

algoritmo, viabilizando a execução de testes diretamente na ferramenta, visualizando os dados e efetuando imediatamente os ajustes necessários.

Esta simplificação do processo de construção de algoritmos de inferência eliminou o acoplamento das tarefas de modelagem e de construção dos jogos, permitindo que a modelagem evolua independentemente do jogo a ser construído. Com isto, o especialista pode visualizar na ferramenta integrada Nébula o resultado de simulações para o algoritmo, sintonizando os parâmetros até obter o comportamento adequado. Além disto, ao permitir a participação direta de todas as especialidades envolvidas, obteve-se melhor interação na equipe, incentivando a troca de experiências e contribuindo para a melhoria do conhecimento de grupo

Os resultados obtidos pelo sistema foram validados a partir da realização de três estudos de caso, que avaliaram sua aplicabilidade em situações reais de uso quanto à modelagem de algoritmos de inferência, a precisão de seus resultados e sua aceitação pelo público alvo. O primeiro estudo de caso verificou sua aplicação em um cenário real de uso, modelando um algoritmo para um simulador de estacionamento de veículo, onde foram explorados os principais recursos oferecidos pela ferramenta, inclusive executando processamento de inferências em tempo real. O segundo estudo de caso validou a precisão de seus resultados, comparativamente aos resultados fornecidos para o mesmo algoritmo construído no *software* Matlab. O terceiro estudo de caso tratou da realização de um estudo de caso prático exploratório, propondo a utilização do sistema por um grupo de especialistas, avaliando seu critério de usabilidade e evidenciando sua aceitação pelo público-alvo de especialistas.

Portanto, conclui-se, que os resultados obtidos nos estudos de caso realizados, conforme descritos no Capítulo 6, evidenciaram a aceitação das hipóteses propostas, ao concluir a adequação do sistema Nébula no atendimento aos requisitos identificados para a modelagem dos algoritmos de inferência, nos seus critérios de precisão, viabilidade e usabilidade.

7.2 Contribuições

No contexto do projeto de virtualização de jogos pedagógicos, as contribuições deste trabalho podem ser observadas como uma ferramenta adicional para

contribuir no desenvolvimento de mecanismos de apoio às decisões dos especialistas no processo de avaliação cognitiva.

Entre as principais contribuições, podem-se destacar:

- Viabilização do uso de inteligência computacional agregada aos jogos psicopedagógicos, ao propor a técnica de lógica nebulosa para solucionar os problemas relacionados à natureza inerentemente imprecisa e inexata dos algoritmos de avaliação cognitiva;
- Simplificação do processo de construção de algoritmos, ao desacoplar sua lógica da codificação dos jogos. Portanto, o desenvolvimento dos jogos psicopedagógicos pode seguir em frentes de trabalho paralela e distintas, sem haver acoplamento de dependência entre elas.
- Fornecimento de uma interface única, em que os algoritmos podem ser simulados e os resultados imediatamente visualizados, de maneira que os parâmetros possam ser sintonizados em busca de melhores resultados, em sucessivas iterações de testes e ajustes;
- Minimização do longo processo para disponibilização dos algoritmos de inferência nos jogos, ao simplificar as diversas etapas envolvidas no modelo tradicional: modelar, implementar, compilar, publicar, simular, validar. Estas etapas resumem-se em execuções iterativas a partir da ferramenta integrada Nébula;
- Facilitação da descrição do algoritmo de inferência, ao abstrair as representações matemáticas complexas dos conceitos *fuzzy* em modelos gráficos apresentando uma interface intuitiva, viabilizando a participação dos diversos perfis especialistas das equipes multidisciplinares no processo de construção dos algoritmos;
- Permite a generalização no tratamento de problemas que possam ser modelados em sistemas de inferência *fuzzy*, viabilizando o processamento de informações de diversos cenários de aplicação, estendendo seu uso a áreas mais amplas de atuação, não necessariamente restritos aos jogos psicopedagógicos;

- Como diversas características podem ser simultaneamente observadas durante a sessão de jogo, os especialistas podem criar várias instâncias a executá-las paralelamente em um mesmo jogo, a seu critério – cada qual, representando um aspecto específico de informações relevantes a serem processadas;

Por fim, pode-se compreender que os resultados processados pelos algoritmos modelados a partir da ferramenta Nébulas poderão servir de insumos para a realização de processamentos adicionais por outros mecanismos de inteligência computacional, conforme descrito na Seção 7.4. Portanto, além de fornecer mecanismos de apoio à decisão dos especialistas, o sistema Nébulas pode abrir novos horizontes de pesquisa para compreensão das informações que podem ser identificadas durante as sessões de jogo, constituindo uma base de conhecimento consolidada em que novas propostas inovadoras possam atuar para processamento e interpretação, agregando novos valores para as informações coletadas.

7.3 Limitações e Dificuldades Encontradas

Ao longo do desenvolvimento desta pesquisa, deparamo-nos com alguns cenários e limitações, os quais tornam-se relevantes destacar.

Primeiramente, cabe ressaltar que esta ferramenta, assim como os algoritmos de inteligência computacional desenvolvidos a partir dela, não pretende substituir o papel do especialista humano, cujas observações e conhecimento inato são fundamentais para a tarefa de avaliação cognitiva. Sua proposta é apenas fornecer mais um conjunto de insumos que possam apoiar suas decisões durante e após a realização das sessões de jogo.

Desta mesma forma, os mecanismos de inferência não pretendem ser, por si só, suficientes para atender a todas as nuances envolvidas na lógica para avaliação cognitiva – o que precisa ser feito em um nível macro pelo especialista. Seu objetivo é atuar no processamento de um conjunto informações pontuais, que podem ser entendidas como pequenos nós deste complexo conjunto de mecanismos interligados, cujos aspectos de relacionamento são abordados por outros projetos de pesquisa em andamento.

Neste mesmo cenário, a lógica nebulosa não pretende resolver todos os problemas de diferentes naturezas, mas se propõe a apresentar solução para aqueles que possam ser modelados como algoritmos matemáticos que lidem com informações imprecisas, e que possam ser especialmente descritos por um conjunto de regras em forma de sentenças lógicas. De maneira análoga, a ferramenta não se propõe a resolver todas as diferentes nuances observadas nas sessões de avaliação cognitiva.

Ao descrever o conjunto de regras que caracterizam os sistemas de inferência *fuzzy*, comumente é gerado um produto cartesiano que correlaciona as diferentes combinações para os conjuntos nebulosos de cada variável com sua saída correspondente. Portanto, ao considerar muitas variáveis de entrada, ou muitos conjuntos nebulosos para cada variável, pode ocorrer o efeito denominado “explosão combinatória de regras”, já evidenciado e referenciado na literatura (DEMASI, CRUZ, 2002; FIGUEIREDO et al, 2007). Para evitar este problema, modelos muito complexos podem ser divididos em duas ou mais instâncias distintas, simplificando o número de variáveis envolvidas em cada uma delas. Inclusive, diversas instâncias podem ser organizadas em modelos hierárquicos, em que a saída de uma instância alimenta a instância seguinte. O desenvolvedor e o especialista têm liberdade para planejar estas modelagens durante o planejamento e elaboração dos jogos.

7.4 Trabalhos Futuros

Dentro da linha de pesquisa para a virtualização de jogos psicopedagógicos do INCE/UFRJ, este trabalho pode ser considerado pioneiro ao permitir a utilização de inteligência computacional agregada aos jogos construídos. Portanto, diversas oportunidades para o desenvolvimento de trabalhos futuros podem ser vislumbradas.

Primeiramente, o grande desafio da modelagem de algoritmos de inferência está relacionado à correta descrição do conjunto de regras. Portanto, podem-se propor frentes de estudo que se dediquem a pesquisar a possibilidade de auto-aprendizado, ou aprendizado adaptativo, em que técnicas adicionais de inteligência computacional podem ser utilizadas para analisar bases de dados para servirem de exemplo para determinarem tanto o conjunto de regras, como as funções de pertinência para as variáveis de entrada. Com isto, podem-se aperfeiçoar algoritmos existentes ou

generalizá-los para aplicações semelhantes. Para isto, podem-se utilizar técnicas de redes neurais, análises bayesianas, ou sistemas mistos neuro-fuzzy.

Ainda em função do desafio referente à modelagem das regras, pode-se propor a construção de um módulo supervisor que possa orientar o especialista na elaboração e na sintonização dos parâmetros do algoritmo e na definição de regras, otimizando o algoritmo em busca de melhores resultados. Este módulo supervisor poderia se encarregar de analisar a superfície de cobertura das regras, buscando por regiões não cobertas ou regras conflitantes, ou ainda analisar as funções de pertinência por intersecções inadequadas, ou que pudessem ser otimizadas. Resumidamente, este módulo supervisor poderia fornecer os seguintes recursos, que em função de limitações de prazo e escopo, não puderam ser cobertas por este trabalho de pesquisa:

- Realizar a análise sequenciada de amostras ao longo do tempo, identificando oscilações, desvios e comportamentos inadequados, passíveis de otimizações;
- Sugerir alterações nos parâmetros de sintonização dos algoritmos, nas etapas de simulação e ajustes, buscando melhor otimização dos resultados, propondo melhores configurações para os intervalos das funções de pertinência;
- Analisar a base de regras e identificar a completude da superfície de cobertura, regras conflitantes ou pouco utilizadas que possam ser desconsideradas, melhorando a legibilidade das definições pelo especialista e o desempenho do processamento da inferência.

O estudo de caso com um grupo de especialistas evidenciou que a principal dificuldade na utilização do sistema se refere ao entendimento dos conceitos envolvidos na especificação das funções de pertinência para as variáveis. Além dos recursos anteriormente descritos, pode-se propor um estudo de usabilidade de interface, objetivando melhorar as representações visuais e acrescentar recursos de orientação ao usuário, como *hints* ou *help online*, no formato de *wizards* ou de tutoriais.

Outro recurso proposto para auxiliar a modelagem dos algoritmos pode ser compreendido como uma funcionalidade de *debugger*, onde o processamento da inferência possa ser analisado ao longo do tempo. Esta funcionalidade armazenaria as informações da inferência em cada momento, como os valores de entrada, as regras ativadas e o valor inferido, e forneceria mecanismos para que o usuário pudesse

visualizar essas informações de maneira consolidada. Desta forma, uma determinada regra que fosse inadvertidamente ativada em um determinado momento poderia ser corrigida na modelagem.

Neste contexto, acredita-se que o conjunto de informações processadas pelos mecanismos de inferência implementados na ferramenta Nébula, caracterizados como produto principal deste trabalho, pode representar um potencial insumo para o desenvolvimento de novas frentes de pesquisa, que possam se dedicar a análise e pós-processamento destas informações, realizando projeções heurísticas e análises bayesianas em busca de padrões, que possam auxiliar os pesquisadores a melhor compreenderem os mecanismos que atuam no processo de avaliação cognitiva.

Esta representação consolidada de todo um conjunto de dados historicamente coletados em diversas sessões de jogo, apoiado por modelos estatísticos que identifiquem e evidenciem padrões, grupamentos e linhas de tendências, pode auxiliar os pesquisadores a melhorar seus modelos de avaliação cognitiva, a partir de uma melhor compreensão do processo em uma visão macroscópica.

Por fim, uma série de estudos adicionais pode ser proposta com a utilização do sistema Nébula, abrangendo uma diversidade de áreas para aplicação de inteligência computacional, segundo os preceitos preconizados por este trabalho. Seu estudo, inicialmente orientado aos jogos psicopedagógicos, pode ser estendido para outras possibilidades, como sistemas de controle, sistemas especialistas, sistemas de suporte à decisão orientados a prognósticos médicos, entre outros. Desde que possam ser representados a partir de um sistema de inferências fuzzy baseado em regras, novos experimentos podem ser realizados, utilizando o sistema Nébula para integrar o envolvimento de especialidades distintas em cenários multidisciplinares.

Referências Bibliográficas

ABNT- Associação Brasileira de Normas Técnicas. **NBR ISO 9241-11/2002 - Requisitos ergonômicos para trabalho de escritórios com computadores: orientações sobre usabilidade.** Rio de Janeiro: ABNT, 2002.

ALMEIDA, V. R. S. A. **Sistemas de recomendação para grupos na difusão do conhecimento organizacional.** 2005. Dissertação (Mestrado em Informática) – Programa de Pós-Graduação em Informática, IM/NCE, Universidade Federal do Rio de Janeiro, Rio de Janeiro, 2005

BATTAIOLA, André Luiz. Jogos por computador - histórico, relevância tecnológica e mercadológica, tendências e técnicas de implementação. In: JORNADA DE ATUALIZAÇÃO EM INFORMÁTICA, 19., 2000, Curitiba. **Anais ...** Curitiba: SBC, 2000. p. 83-122.

BERTOLINI, Eni A. Sivera ; SILVA, Miguel A. de Mello. Metacognição e Motivação na Aprendizagem: Relações e Implicações Educacionais. **Revista Técnica IPEP**, São Paulo, v. 5, n. 1/2, p. 51-62, jan./dez. 2005. Disponível em: <<http://www.ipep.edu.br/portal/publicacoes/revista/revista2005/TEXT0%205.pdf>> acesso em: 05 mai. 2011

BRICKENKAMP, R. Teste d2: atenção concentrada. In: ALVES, Irai Cristina Boccato. **Manual/padronização brasileira.** São Paulo: Casa do Psicólogo, 2002.

BURDA, Philip ; STARKEY, Timothy ; DOMINGUEZ, Florentino ; VERA, Vivian; Computer-assisted cognitive rehabilitation of chronic psychiatric inpatients. **Computers in Human Behavior**, Amsterdam; v.10, n. 3, p. 359-368, Aut. 1994.

COSTA, R. M. E. M. ; CARVALHO, Luis Alfredo Vidal de. O Uso de jogos digitais na reabilitação cognitiva. In: SIMPÓSIO BRASILEIRO DE INFORMÁTICA NA EDUCAÇÃO, 16., 2005, Juiz de Fora.WORKSHOP DE JOGOS DIGITAIS NA EDUCAÇÃO, 2005, Juiz de Fora. Anais... Juiz de Fora: SBC, 2005. CD-Rom

COSTA, R. M. E. M. **Ambientes virtuais na reabilitação cognitiva de pacientes neurológicos e psiquiátricos.** 2000. Tese (Doutorado em: Engenharia de Sistemas e Computação) – COPPE Sistemas, Universidade Federal do Rio de Janeiro, Rio de Janeiro, Brasil, 2000.

CRUZ, A. J. O. **Introdução à lógica nebulosa.** Rio de Janeiro: NCE/UFRJ, 2004. Material de aula do curso de Mestrado em Informática do NCE/UFRJ.

_____. **Estacionando um caminhão.** Rio de Janeiro: NCE/UFRJ, 2002. Notas de aula do curso de Mestrado em Informática do NCE/UFRJ. Disponível em: <<http://www.infostat.sk/cevavstat/fuzzy/prispevky/ICEIS07.pdf>>. Acesso em: 07 abr. 2011.

DEMASI, Pedro. **Estratégias adaptativas e evolutivas em tempo real para jogos eletrônicos.** 2003. Dissertação (Mestrado em Informatica) – Instituto de Matemática, Núcleo de Computação Eletrônica, Universidade Federal do Rio de Janeiro, Rio de Janeiro, 2003.

DEMASI, Pedro ; CRUZ, A. J. O. Aprendizado de regras nebulosas em tempo real para jogos eletrônicos. In: SIMPÓSIO BRASILEIRO DE SISTEMAS MULTIMÍDIA E WEB. JOGOS, 11., 2003. Salvador; WORKSHOP BRASILEIRO DE JOGOS E ENTRETENIMENTO DIGITAL, 2., **Anais ...** Salvador: SBC, 2003.

_____. Modelagem fuzzy para um jogo de naves espaciais. In: WORKSHOP BRASILEIRO DE JOGOS E ENTRETENIMENTO DIGITAL, 1., 2002, São Paulo. **Anais ...** São Paulo: USP, 2002.

FERREIRA, Paula Andrea Prata. **Um projeto arquitetural para sistemas reuopedagógicos integrados**. 2009. 161 f. Dissertação (Mestrado em Informática) – Instituto de Matemática, Núcleo de Computação Eletrônica, Universidade Federal do Rio de Janeiro, Rio de Janeiro, 2009.

FIGUEIREDO, Karla et al . Modelo neuro-fuzzy hierárquico politree com aprendizado por reforço para agentes inteligentes. **Controle & Automação**, Natal. Sociedade Brasileira de automática, v. 18, n. 2, jun. 2007 . Disponível em: <http://www.scielo.br/scielo.php?script=sci_arttext&pid=S0103-17592007000200009&lng=en&nrm=iso>. Acesso em: 04 maio 2011.

FOWLER, Martin. **Padrões de arquitetura de aplicações corporativas**. Porto Alegre: Bookman, 2000.

GOLDSCHMIDT, Ronaldo R. **KDD e mineração de dados**. noções introdutórias em lógica nebulosa. Rio de Janeiro: Univercidade, 2007

GOMES, Diogo da ; MOTTA, C. ; CRUZ, A. J. O Sistema integrado para construção de inferências aplicáveis a jogos psicopedagógicos. **Revista Brasileira de Computação Aplicada**, Passo Fundo, v.2, n.2, p. 58-68, 2010.

GOMES, Diogo et al. **Phidias**: construção para agilizar a virtualização de jogos metacognitivos. Rio de Janeiro: NCE/UFRJ, 2009. 10 p. (Relatório Técnico, 05/09).

GOMIDE, F. ; GUDWIN, R. ; TANSCHKEIT, R. Conceitos fundamentais da teoria de conjuntos fuzzy, lógica fuzzy e aplicações. In: IFSA WORLD CONGRSS, 6.,1995, São Paulo. **Tutorials ...** São Paulo: IFSA, p.1-38, 1995.

HUDEC, Miroslav ; Vujošević, Mirko. **Integration of a fuzzy system and an information System for the territorial units ranking**. Disponível em: <<http://www.infostat.sk/cevavstat/fuzzy/prispevky/ICEIS07.pdf>>. Acesso em: 17 fev. 2011.

JBOSS, **RichFaces project**. Disponível em: <<http://www.jboss.org/richfaces>>. Acesso em: 9 maio 2011.

_____. **Hibernate project**. Disponível em: <<http://www.hibernate.org/>>. Acesso em: 9 maio 2011.

KOSKO, B. **Neural networks and fuzzy systems**: a dynamical systems approach to machine intelligence. New Jersey: Prentice Hall, 1992.

LIMA, Ricardo F. de. Compreendendo os mecanismos atencionais. **Ciências & Cognição**, Rio de Janeiro, Ano 2, v.6, nov. 2005. Disponível em: <<http://www.cienciasecognicao.org/pdf/v06/m24554.pdf>>. Acessado em: 03 jun. 2011.

MAMDANI, E. H. ; ASSILIAN, S. An experiment in linguistic synthesis with a Fuzzy LogicController. **International Journal of Man-Machine Studies**, London, v. 7, n. 1, p. 1-13, 1975.

MANDANI, E. H. Application of fuzzy logic to approximate reasoning using linguistic synthesis. **IEEE Transactions on Computers**, Piscataway, v. 16, n. 12, p. 1182-1191, Dec. 1977.

MARQUES, C. V. M. et al. **Avaliação de crianças deficientes visuais através de jogos neuropedagógicos**. SIMPÓSIO DE COMPUTAÇÃO APLICADA, 2009, Passo Fundo. **Anais ...** Passo Fundo: UPF, 2009.

MARQUES, C. V. ; OLIVEIRA, C. E. T. ; MOTTA, C. (Org.). **A revolução cognitiva: um estudo sobre a teoria de Franco Lo Presti** Seminário. Rio de Janeiro: PPGI/IM/NCE, 2009. (Relatório Técnico, 04/09).

_____. **Jogo da roda da linguagem** - Estudo proposicional para virtualização de jogos psicopedagógicos. Rio de Janeiro: PPGI/IM/NCE, 2009. No Prelo.

MORATORI, P. B. **Análise de estabilidade e robustez de controladores nebulosos: aplicação ao controle de trajetória de robôs**. 2006. Dissertação (Mestrado em Informática) – Instituto de Matemática, Núcleo de Computação Eletrônica, Universidade Federal do Rio de Janeiro, Rio de Janeiro, 2006

MORATORI, P. B. et al. Analysis of the stability of a fuzzy control system developed to control a simulated robot. In: IEEE INTERNATIONAL CONFERENCE ON FUZZY SYSTEMS, 14., 2005, Reno Nevada. **Proceedings ...** Reno, Nevada: IEEE, 2005. p.726-730.

MATHWORKS, **Matlab – the language of technical computing**. Matlab Software Version 7.6, 2008. Disponível em: <<http://www.mathworks.com/products/matlab/>> Acesso em: 15 abr. 2011.

MVC, MODEL-VIEW-CONTROLLER. **Java BluePrints**, 2011. Disponível em: <<http://java.sun.com/blueprints/patterns/MVC-detailed.html>>. Acesso em: 14 jan 2011.

ORACLE, **JavaServer faces technology**. Disponível em: <<http://www.oracle.com/technetwork/java/javasee/javaserverfaces-139869.html>>. Acesso em: 9 mai. 2011.

_____. **JavaDB**. Disponível em: <<http://www.oracle.com/technetwork/java/javadb/overview/index.html>>. Acesso em: 09 mai. 2011.

PARKER, Richard ; REA, Louis M. **Metodologia de pesquisa: do planejamento à execução**. 1. ed.. São Paulo: Thomsom Pioneira, 2000.

PIAGET, Jean. **Psicologia e pedagogia**. Rio de Janeiro: Forense, 1988.

RABIN, Steve. **AI Game programming wisdom**. Hingham: Charles River Media, 2002. 672p.

RIEDER, R. ; BRANCHER, J. D. Aplicação da lógica fuzzy a jogos didáticos de computador - a experiência do mercado GL. In: CONGRESSO IBEROAMERICANO DE INFORMÁTICA EDUCATIVA, 7., 2004, Monterrey. **Actas ...** Montrrey, 2004.

SARTER, M. ; GIVENS, B. ; BRUNO, J. P. The cognitive neuroscience of sustained attention: where top-down meets bottom-up. **Brain Research Reviews**, Amsterdam, v. 35, n. 2, p. 146-160, Apr. 2001. Disponível em: <http://www.sciencedirect.com/science?_ob=ArticleURL&_udi=B6SYS-42Y1BCS-5&_user=10&_rdoc=1&_fmt=&_orig=search&_sort=d&view=c&_acct=C000050221&_version=1&_urlVersion=0&_userid=10&md5=c1e7deed882d1a8d7db68f58429ebac6>. Acessado em: 03 jun. 2011.

SEMINÉRIO, Franco **Lo Presti. Elaboração dirigida: um caminho para o desenvolvimetro metaprocessual da cognição humana**. Rio de Janeiro: Instituto Superior de Estudos e Pesquisas Psicossociais, FGV, 1987. (Cadernos do ISOP, no. 10)

_____. Metacognição: um caminho para ultrapassar os limites da audição. **Espaço Informativo Técnico Científico do INES**. Rio de Janeiro, n. 14, p. 23-36, 2000. Volume. único

SILVEIRA, Sidnei Renato ; BARONE, Dante Augusto Couto. Jogos educativos computadorizados utilizando a abordagem de algoritmos genéticos. In: IV CONGRESSO IBEROAMERICANO DE INFORMÁTICA EDUCATIVA, 4., 1998, Brasília. **Anais ...** Brasília, 1998.

STERNBERG, Robert J. **Psicologia cognitiva**. Porto Alegre: Artmed, 2000.

TAKAGI, T. ; SUGENO, M. Fuzzy identification of systems and its applications to modelling and control. **IEEE Transactions. on Systems, Man & Cybernetics**, New York, v. 15: n. 1, p. 116-132, 1985.

TANSCHUIT, R. **Sistemas fuzzy. PUC-Rio/DEE**. Disponível em: <<http://www.inf.ufsc.br/~mauro/ine5377/leituras/ICA-Sistemas%20Fuzzy.pdf>>. Acesso em: jan. 2011.

TANSCHUIT, R. ; SCHARF, E. M.. Controle de um braço mecânico através de um controlador fuzzy self- organising. In: CONGRESSO BRASILEIRO DE AUTOMÁTICA, 8., 1990, Belém. **Anais ...** Belém: SBA, 1990.p. 481-487.

TORDESCHINI, R. ; PEREIRA, A. ; STRADIOTTO, C. Jogos e entretenimentos em realidade virtual: perspectivas e possibilidades. In: CONGRESSO INTERNACIONAL DE ENGENHARIA GRÁFICA NAS ARTES E NO DESENHO, 4., 2000, Ouro Preto, **Anais ...** Ouro Preto, 2000.

VYGOTSKY, L. S. **A Formação social da mente**. São Paulo: Martins Fontes, 1989.

ZADEH, L.A. A Theory of appoximate reasoning. In: HAYES, J. ; MITCHIE, D. E ; MIKULICH, L. **Machione intelligence**, New York: WILEY, 1979. v. 9. p 149-194.

ZADEH, L. A. Fuzzy sets. **Information and Control**, New York, v. 8, n.3, p 338 – 353, 1965.

Apêndice

Apêndice: Roteiro utilizado na avaliação de usabilidade do sistema Nébula

O roteiro em forma de questionário foi elaborado e diagramado graficamente de maneira a facilitar a marcação das respostas durante a avaliação, e também a interpretação dos resultados posteriormente, atuando como um protocolo semi-estruturado para registro das observações e percepções durante o estudo de caso exploratório. Foi solicitado a cada respondente que, para cada linha de resposta, marcasse a coluna correspondente à sua opinião. A escala de resposta das questões foi ordenada de maneira gradativa crescente, facilitando a interpretação de seus significados.

Questão	0- Discordo Totalmente	1- Discordo Parcialmente	2- Indeciso	3- Concordo Parcialmente	4- Concordo Totalmente
1. A interface do sistema Nébula facilita o usuário no desenvolvimento do algoritmo, orientando por uma sequência de passos.					
2. As facilidades oferecidas através de gráficos, tabelas, dicas e menus contribuíram para a compreensão das representações matemáticas envolvidas no processo de construção dos algoritmos.					
3. A utilização do sistema Nébula permite integrar as equipes multidisciplinares ao atuarem conjuntamente na definição dos algoritmos.					
4. A funcionalidade de simulação auxiliou no processo de testes, validação e sintonização dos parâmetros do algoritmo.					
5. Foi possível realizar alterações nos parâmetros de sintonização do algoritmo a partir do sistema Nébula, e percebê-las imediatamente no jogo em execução.					
6. A utilização do sistema Nébula melhorou o processo de desenvolvimento dos algoritmos utilizados nos jogos psicopedagógicos, em relação ao modelo tradicional.					

7. Foi percebida vantagem no sistema Nébula, em comparação às demais opções que envolvem o uso de ferramentas de engenharia (como o Matlab) e a utilização de técnicas que exigiam desenvolvimento de código (Java, C, etc).					
8. Os resultados da inferência estavam de acordo com o esperado.					
9. Você se sentiu confortável em utilizar o sistema?					
10. O que você sentiu falta na ferramenta? Quais os principais problemas identificados?					
11. Comentários					