

Dissertação de Mestrado

**UM MODELO DE TOMADA DE DECISÃO BASEADO NA
TEORIA DA PERSUASÃO APLICADO À CLASSE DE JOGOS
MMORPG**

Helio Cavalcante Silva Neto
helio.hx@gmail.com

Maceió, Abril de 2011

HELIO CAVALCANTE SILVA NETO

**UM MODELO DE TOMADA DE DECISÃO BASEADO NA
TEORIA DA PERSUASÃO APLICADO À CLASSE DE JOGOS
MMORPG**

Dissertação apresentada como requisito parcial para obtenção do grau de mestre em Modelagem Computacional de Conhecimento na Universidade Federal de Alagoas.

Orientador:

Prof. Dr. Fábio Paraguaçu Duarte da Costa

Co-orientador:

Prof. Dra. Roberta Vilhena Vieira Lopes

Maceió, Abril de 2011

Catálogo na fonte
Universidade Federal de Alagoas
Biblioteca Central
Divisão de Tratamento Técnico
Bibliotecária Responsável: Helena Cristina Pimentel do Vale

S586u Silva Neto, Helio Cavalcante.
Um modelo de tomada de decisão baseado na teoria da persuasão aplicado à classe de Jogos MMORPG / Helio Cavalcante Silva Neto. – 2011.
195 f. : il.

Orientador: Fábio Paraguaçu Duarte da Costa.
Co-Orientadora: Roberta Vilhena Vieira Lopes.
Dissertação (mestrado em Modelagem Computacional de Conhecimento) –
Universidade Federal de Alagoas. Instituto de Computação. Maceió, 2011.

Bibliografia: f. 145-154.
Apêndices: f. 155-195.

1. Jogos por computador – Persuasão. 2. Tomada de decisão. 3. Redes de Petri.
4. Teoria dos jogos. 5. MMORPG (Massively Multiplayer Online Role-Playing
Game). 6. RPG (Role-Playing Game). I. Título.

CDU: 004.4

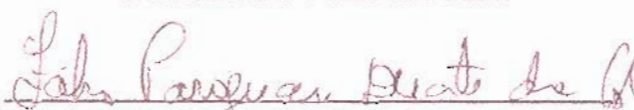


UNIVERSIDADE FEDERAL DE ALAGOAS/UFAL
**Programa Multidisciplinar de Pós-Graduação em
Modelagem Computacional de Conhecimento**
Avenida Lourival Melo Mota, Km 14, Bloco 09, Cidade Universitária
CEP 57.072-900 – Maceió – AL – Brasil
Telefone: (082) 3214-1364



Membros da Comissão Julgadora da Dissertação de Mestrado de Helio Cavalcante Silva Neto, intitulada: “Um Modelo de Tomada de Decisão Baseado na Teoria da Persuasão Aplicado à Classe de Jogos MMORPG”, apresentada ao Programa de Pós-Graduação em Modelagem Computacional de Conhecimento da Universidade Federal de Alagoas em 14 de abril de 2011, às 09h00min, na sala de aula do Mestrado em Modelagem Computacional de Conhecimento.

COMISSÃO JULGADORA



Prof. Dr. Fábio Paraguaçu Duarte da Costa

UFAL – Instituto de Computação

Orientador



Prof. Dra. Roberta Vilhena Vieira Lopes

UFAL – Instituto de Computação

Orientadora


Prof. Dr. Arturo Hernández-Domínguez

UFAL – Instituto de Computação

Examinador


Prof. Dr. Edilson Farneda

UCB – Pró-Reitoria de Pós-Graduação e Pesquisa

Examinadora

Maceió, abril de 2011.

A minha mãe Teresa Maria.
A minha Namorada Cristiane Nazaré.

AGRADECIMENTOS

Agradeço primeiramente a minha mãe, que sempre acreditou em mim e nunca mediu esforços para obter o melhor para nossa família.

Sou muito grato a minha namorada, pessoa que está sempre presente em minha vida e me ajudou de todas as formas. Desculpe-me pelas minhas ausências e meus erros, você me completa.

Aos meus orientadores, Roberta Vilhena, mãezona amiga, e Fábio Paraguaçu, por seus conselhos sábios. Sou muito grato aos dois por me acolherem e confiarem no meu trabalho.

Ao amigo Vitor Torres, disposto sempre a ajudar e a aconselhar. E a todos os meus amigos (Bruno Pastor, Bruno Nascimento, Douglas Feitosa, Valter Ramos, Rodrigo Peixoto, Cledson Calaça, Mário André, Tarsis Marinho, Elvys Soares e Henrique Pacheco) que me ajudaram direta e indiretamente. Agradeço em especial a um grande amigo irmão de viagens e de trabalho, Leonardo Filipe (só não coloco o seu nome todo porque não iria caber).

Lembro também dos meus amigos de fora da universidade, sou muito grato a todos.

Obrigado ao meu ventilador, grande companheiro que me proporciona brisas frescas em tardes quentinhas.

Por fim, peço desculpas caso esqueci alguém.

RESUMO

É notório que as pessoas a todo o momento têm de decidir diante de diferentes situações e sobre problemas dos mais diversos e utilizam, para isso, suas experiências passadas, valores, crenças, conhecimentos técnicos, suas habilidades e filosofias. Tomar decisões é algo crucial no âmbito dos jogos. Essa atividade acontece a todo o tempo e em vários níveis e influencia diretamente o desempenho do jogo, inclusive no âmbito do RPG, que é uma ferramenta onde há uma grande interação dos sujeitos em zonas de desenvolvimento proximal, pois que os jogadores têm que se unir para completar os objetivos do jogo (*Quest*), existindo, assim, uma maior troca do conhecimento prévio e social dos jogadores. Por se tratar de um jogo cooperativo que estimula a socialização, a interatividade e o desenvolvimento de habilidades comunicativas, surge, então, a motivação de trabalhar, na presente dissertação, com o RPG. Esta dissertação tem como objetivo construir, um modelo utilizando Rede de Petri e UML, com base na Teoria dos Jogos e na Teoria da Persuasão aplicado ao *Massively Multiplayer Online Role-Playing Game*, na busca de envolver o usuário em algum tipo de trama, trabalhando com elementos lúdicos, ao mesmo tempo, favoreça uma maior facilidade na tomada de decisão.

Palavras-chave: Princípios da Persuasão, Sistemas de Tomada de Decisão, MMORPG, RPG, Redes de Petri e Teoria dos Jogos.

ABSTRACT

It is of common knowledge that at every moment, people have to make decisions over different matters regarding to a wide range of issues and that, in making such decisions, they use of previous experiences, values, beliefs, expertise, skills and even of their own philosophies. From a videogame perspective, decision-making is a crucial activity that happens at all times and at different levels of perception. Moreover, it has direct influence over a game's performance, a fact that also concerns to RPGs as they can act as tools to enhance the improvement of the proximal development zones of the involved individuals, due the need these individuals have to create groups and work together in order to complete different objectives of a game (known as quests). A need that results in higher exchanges of previously acquired knowledge, as well as their social knowledge. As the RPG has an inherent cooperative character that stimulates socialization, interaction and the improvement of communication skills, it was thought that it would be interesting to take advantage of the RPG to built a model using UML and Petri Net, based on the Game Theory and on a application of the Theory of Persuasion to an Massively Multiplayer Online Role-Playing Game environment, that involves the user in some kind of plot, at the same time that favors a greater ease in decision-making.

Keywords: Psychology of Persuasion, Systems Decision Making, MMORPG, RPG, Petri Net and Game Theory.

LISTA DE FIGURAS

Figura 1: Trade do <i>World of Warcraft</i>	35
Figura 2: Fluxo da Reciprocidade.....	60
Figura 3: Fluxo da Consistência.....	68
Figura 4: Fluxo da Autoridade.....	73
Figura 5: Fluxo da Prova Social.....	78
Figura 6: Fluxo da Escassez.....	81
Figura 7: Fluxo da Empatia.....	86
Figura 8: Topologia do modelo.....	90
Figura 9: Topologia dos Agentes.....	91
Figura 10: Mapa do Conhecimento.....	91
Figura 11: Pacotes do Modelo.....	98
Figura 12: Diagrama de Caso de Uso Jogador e NPC.....	100
Figura 13: Diagrama de Caso de Uso Jogador e Jogador.....	101
Figura 14: Diagrama de Caso de Uso Jogador e Monstro.....	102
Figura 15: Diagrama de Caso de Uso Jogador e Jogador Vendedor.....	103
Figura 16: Diagrama de Caso de Uso Backup.....	104
Figura 17: Diagrama Estado Cria Conta.....	117
Figura 18: Diagrama Estado Cria Personagem.....	118
Figura 19: Diagrama Estado Caçar.....	119
Figura 20: Diagrama Estado Compra Item Jogador.....	120
Figura 21: Diagrama Estado Compra Item NPC.....	121

Figura 22: Diagrama Estado Compra Skill.	122
Figura 23: Diagrama Estado Cria Guild.	123
Figura 24: Diagrama Estado Cria Party.	124
Figura 25: Diagrama Estado Quest.	126
Figura 26: Diagrama Estado Vende Item NPC.	127
Figura 27: Diagrama Estado Fórum.	128
Figura 28: Diagrama Estado Itens a Venda.	129
Figura 29: Diagrama Estado Leilão.	130
Figura 30: Exemplo de Subsistema.	131
Figura 31: Diagrama Subsistemas do Ambiente.	133
Figura 32: Diagrama Subsistemas do ambiente virtual.	134
Figura 33: Exemplo de Componente.	138
Figura 34: Exemplo de Nó.	138
Figura 35: Diagrama de Classe cria conta e cria personagem.	155
Figura 36: Diagrama Classe Caçar.	156
Figura 37: Diagrama Classe Adquirir Quest.	157
Figura 38: Diagrama Classe Compra Item Jogador.	158
Figura 39: Diagrama Classe Compra ou Vende Item NPC.	159
Figura 40: Diagrama Classe Compra Skill.	160
Figura 41: Diagrama Classe Cria Guild.	161
Figura 42: Diagrama Classe Cria Party.	162
Figura 43: Diagrama Classe Fóruns, Leilão e Itens a Venda.	163
Figura 44: Diagrama Classe com Interface Cria Conta e Personagem Jogador.	164
Figura 45: Diagrama Classe com Interface Caçar.	165

Figura 46: Diagrama Classe com Interface Adquirir Quest.	166
Figura 47: Diagrama Classe com Interface Compra Item Jogador.	167
Figura 48: Diagrama Classe com Interface Compra ou Vende Item NPC.	168
Figura 49: Diagrama Classe com Interface Compra Skill.	169
Figura 50: Diagrama Classe com Interface Cria Guild.	170
Figura 51: Diagrama Classe com Interface Cria Party.	171
Figura 52: Diagrama Classe com Interface Fóruns, Leilão e Itens a Venda.	172
Figura 53: Diagrama Componentes com Implantação do jogo.	173
Figura 54: Diagrama Componentes com Implantação do ambiente virtual.	174
Figura 55: Reciprocidade na Rede de Petri.	175
Figura 56: Consistência na Rede de Petri.	176
Figura 57: Autoridade na Rede de Petri.	177
Figura 58: Prova Social na Rede de Petri.	178
Figura 59: Escassez na Rede de Petri.	179
Figura 60: Empatia na Rede de Petri.	180
Figura 61: Elementos básicos de uma rede de Petri.	182
Figura 62: Transformações preservando vivacidade, segurança e limitação.	188
Figura 63: Interface Visual Object Net ++.	190
Figura 64: Interface das propriedades dos places, transições e arcos.	191
Figura 65: Interface de informações relativas aos places, transições e arcos.	192
Figura 66: Simulando a execução de uma Rede de Petri.	193
Figura 67: Interface de opções.	193
Figura 68: Rede de Petri Lâmpada.	194

LISTA DE QUADROS

Quadro 1: Atores.	95
Quadro 2: Cores dos <i>Forms</i>	106

LISTAS DE ABREVIATURAS E SIGLAS

CAI: *Computer-Aided Instruction* ou *Computer-Assisted Instruction* significa (Instrução Assistida ou Auxiliada por Computador).

CBL: *Computer Based-Learning* (Aprendizagem Baseada no Computador).

D&D: *Dungeons & Dragons* (Caverna do Dragão).

HTML: *HyperText Markup Language* (Linguagem de Marcação de Hipertexto).

IGDA: *International Game Developers Association* (Associação Internacional de Desenvolvedores de Jogos).

LAN: *Local Network*.

MMORPG: *Massively Multiplayer Online Role-Playing Game* (Jogo de Interpretação de Personagem Online e em Massa para Múltiplos Jogadores).

MMOG: *Massively Multiplayer Online Game* (Jogo Eletrônico Online Multijogador em Massa).

MUDD: *Multi-User Dungeons&Dragons* (Labirintos e Dragões para Múltiplos Usuários).

NPC: *Non-Player Character* (Personagem que não do Jogador).

ODBC: *Open Data Base Connectivity* (é um padrão para acesso a sistemas gerenciadores de bancos de dados).

PC: *Player Character* (Personagem do Jogador).

RPG: *Role Playing Game* (Jogo de Interpretação de Personagens).

UML: *Unified Modeling Language* (Linguagem Unificada de Modelagem).

WLAN: *Wireless Local Area Network*.

GLOSSÁRIO

Backup: Ato de salvar todas as informações do ambiente ou sistema.

Chat: Termo do Inglês que significa bate-papo.

Ethos: Ambiente desenvolvido pelo autor do trabalho.

Forms: também chamado de fronteira, nada mais é que uma interface gráfica entre o jogador e o gerenciador de informações da plataforma.

Dropar: Termo técnico do RPG que vem do Inglês Drop significa cair, soltar, largar.

Get: Utiliza este método para solicitar alguma informação desejada do objeto na classe.

Guild: Termo técnico do RPG que em inglês significa corporação ou associação a qual o jogador pertence.

JUDE: Java and UML Developers Environment (Ambiente para Desenvolvedores UML e Java) - software livre para modelagem UML.

Level: Termo técnico do RPG que significa que o personagem ficou mais forte quanto maior o level.

Main-Frames: Computador de grande porte.

Money: Termo do inglês que significa dinheiro.

Monstro: Inimigo encontrado no ambiente RPG.

Moodle: Modular Object-Oriented Dynamic Learning Environment é um software livre, de apoio à aprendizagem, executado num ambiente virtual.

Multiplayer: Se permite vários usuários jogarem em uma rede local, modem ou pela *Internet*.

On-line: Um anglicismo advindo do uso da Internet, sendo em linha uma tradução literal de *on-line*, pouco usada no português.

Party: Termo técnico do RPG que em Inglês significa grupo de caça ou grupo de missão.

Quest: Missão ou objetivo ligado ao RPG.

Set: Utiliza este método para modificar alguma informação desejada do objeto na classe.

Skill: Termo técnico do RPG que em Inglês significa habilidade a qual pertence ao personagem exemplo: auto-cura.

Sprite: Aparência do personagem, monstro, item e etc (interface gráfica).

Top: Top Quest, Top Item, Top Monstro e Fórum, são fóruns mais comentados ou visualizados no ambiente.

Trade: Ferramenta típica dos jogos do gênero MMORPG, possibilita a venda ou troca de produtos entre dois jogadores.

SUMÁRIO

INTRODUÇÃO	17
1.1. Sistemas de Tomada de Decisão	17
1.2. Motivação	18
1.3. Objetivos	20
1.3.1. Geral	20
1.3.2. Específicos	20
1.4. Metodologia	21
1.5. Organização	21
2 O MASSIVELY MULTIPLAYER ONLINE ROLE-PLAYING GAME (MMORPG)	23
2.1. Conceitos básicos	25
2.2. Sistema de Regras e Mestre do Jogo	26
2.3. RPG e compartilhamento do conhecimento	29
2.4. Surgimento dos RPG digitais	31
2.5. RPG na <i>Internet</i> e o MMORPG	32
2.6. MMORPG no contexto do comércio	34
2.7. Conclusão do Capítulo	35
3 A TEORIA DA PERSUASÃO	37
3.1. Os princípios da persuasão	37
3.1.1. Reciprocidade	38
3.1.2. Consistência	40
3.1.3. Autoridade	41
3.1.4. Prova Social	44
3.1.5. Escassez	46
3.1.6. Empatia	47
3.2. Conclusão do Capítulo	49
4 SISTEMAS DE TOMADA DE DECISÃO	50
4.1. Tecnologia de Informação	51

4.2.	Componentes da Tecnologia de Informação	52
4.3.	Sistemas de Informação	52
4.4.	Classificação de Sistema de Informação	54
4.4.1.	Sistema de Apoio às Operações	54
4.4.1.1.	Sistema de Processamento de Transações (SPT)	54
4.4.1.2.	Sistemas de Trabalho do Conhecimento e de Automação de Escritório (STC e SAE).....	55
4.4.2.	Sistema de Apoio Gerencial	56
4.4.2.1.	Sistema de Suporte da decisão (SSD)	56
4.4.2.2.	Sistema de Suporte Executivo (SSE)	56
4.4.2.3.	Sistema de Informação Gerencial (SIG)	56
4.5.	Condições de Tomada de Decisão	57
4.6.	Conclusão do Capítulo	57
5	UM MODELO DE TOMADA DE DESCISÃO BASEADO NA TEORIA DA PERSUASÃO.....	59
5.1.	Reciprocidade aplicada ao MMORPG.....	59
5.1.1.	Modulo Reciprocidade fluxo e o princípio modelado na rede Petri	59
5.2.	Consistência aplicada ao MMORPG	67
5.2.1.	Modulo Consistência fluxo e o princípio modelado na rede Petri	67
5.3.	Autoridade aplicada ao MMORPG	72
5.3.1.	Modulo Autoridade fluxo e o princípio modelado na rede Petri.....	73
5.4.	Prova Social aplicada ao MMORPG	76
5.4.1.	Modulo Prova Social fluxo e o princípio modelado na rede Petri	77
5.5.	Escassez aplicada ao MMORPG	80
5.5.1.	Modulo Escassez fluxo e o princípio modelado na rede Petri.....	80
5.6.	Empatia aplicada ao MMORPG	85
5.6.1.	Modulo Empatia fluxo e o princípio modelado na rede Petri.....	85
5.7.	Conclusão do Capítulo	88
6	ESPECIFICAÇÃO DE UM MODELO DE TOMADA DE DECISÃO.....	89
6.1.	Descrição do ambiente	89
6.2.	Atores.....	94

6.3. Caso de Uso	95
6.3.1. Jogador	96
6.3.2. Jogador_Vendedor	97
6.3.3. Monstro	97
6.3.4. NPC	97
6.3.5. Sist. de Backup	98
6.4. Diagrama de caso de uso	98
6.4.1. Jogador e NPC	99
6.4.2. Jogador e Jogador	100
6.4.3. Jogador e Monstro	101
6.4.4. Jogador e Jogador Vendedor	102
6.4.5. Sistema de Backup	104
6.5. Diagrama de Classes	104
6.5.1. Diagrama de Classe Cria Conta e Cria Personagem Jogador	106
6.5.2. Diagrama de Classe Caçar	108
6.5.3. Diagrama de Classe Adquirir <i>Quest</i>	109
6.5.4. Diagrama de Classe Compra Item do Jogador	110
6.5.5. Diagrama de Classe Compra ou Vende Item do NPC	110
6.5.6. Diagrama de Classe Compra <i>Skill</i>	111
6.5.7. Diagrama de Classe Cria <i>Guild</i>	112
6.5.8. Diagrama de Classe Cria <i>Party</i>	112
6.5.9. Diagrama de Classe Fóruns, Leilão e Itens a Venda	113
6.6. Diagrama de Estado	114
6.6.1. Diagrama de Estado Cria Conta	116
6.6.2. Diagrama de Estado Cria Personagem	117
6.6.3. Diagrama de Estado Caçar	118
6.6.4. Diagrama de Estado Compra Item do Jogador	119
6.6.5. Diagrama de Estado Compra Item do NPC	120
6.6.6. Diagrama de Estado Compra <i>Skill</i>	121
6.6.7. Diagrama de Estado Cria <i>Guild</i>	122
6.6.8. Diagrama de Estado Cria <i>Party</i>	123

6.6.9.	Diagrama de Estado <i>Quest</i>	124
6.6.10.	Diagrama de Estado Vende Item para NPC.....	126
6.6.11.	Diagrama de Estado Fórum.....	127
6.6.12.	Diagrama de Estado Itens à Venda.....	128
6.6.13.	Diagrama de Estado Leilão.....	129
6.7.	Subsistemas.....	130
6.7.1.	Diagrama de Subsistemas do ambiente.....	131
6.7.2.	Diagrama de Subsistemas do ambiente virtual.....	134
6.8.	Diagrama De Classes Com Interface.....	135
6.8.1.	Diagrama de Classe com Interface Cria Conta e Cria Personagem Jogador.....	135
6.8.2.	Diagrama de Classe com Interface Caçar.....	135
6.8.3.	Diagrama de Classe com Interface Adquirir <i>Quest</i>	136
6.8.4.	Diagrama de Classe com Interface Compra Item do Jogador.....	136
6.8.5.	Diagrama de Classe com Interface Compra ou Vende Item do NPC..	136
6.8.6.	Diagrama de Classe com Interface Compra <i>Skill</i>	137
6.8.7.	Diagrama de Classe com Interface Cria <i>Guild</i>	137
6.8.8.	Diagrama de Classe com Interface Cria <i>Party</i>	137
6.8.9.	Diagrama de Classe com Interface Fóruns, Leilão e Itens à Venda ...	137
6.9.	Diagrama de Componentes com Implantação.....	138
6.9.1.	Diagrama de Componentes com Implantação do jogo.....	139
6.9.2.	Diagrama de Componentes com Implantação do ambiente virtual ...	140
6.10.	Conclusão do Capítulo.....	140
CONCLUSÃO	141
6.1.	Trabalhos Futuros.....	143
REFERÊNCIAS	144
APÊNDICE	154
APÊNDICE A	– Diagramas de Classes.....	155
APÊNDICE B	– Diagrama de Classes com Interface.....	164
APÊNDICE C	– Diagrama de Componentes com Implantação.....	173

APÊNDICE D – Princípios da Persuasão aplicados ao MMORPG utilizando Redes de Petri.....	175
APÊNDICE E – Rede de Petri	181
1.1.1. Redes de Petri Estendidas.....	183
1.1.2. Propriedades da Rede de Petri.....	184
1.1.3. Equação de Estado.....	186
1.1.4. Refinamento na Rede de Petri	187
1.1.5. Ferramenta de Simulação <i>Visual Object Net ++</i>.....	189
1.1.5.1. Utilizando o Visual Object Net ++	190
1.1.6. Conclusão do Apêndice E	194

INTRODUÇÃO

1.1. Sistemas de Tomada de Decisão

É notório que as pessoas a todo o momento têm de decidir diante de diversas situações e sobre problemas dos mais diferentes, utilizando para isso suas experiências passadas, valores, crenças, conhecimentos técnicos e habilidades. Algumas pessoas são mais conservadores; outras possuem características inovadoras, estando mais dispostas a assumir riscos em potencial. Esses diversos modos de tomada de decisão podem representar tanto o sucesso, como o fracasso daquelas que dependem desse processo (GARCIA, 2003).

Tomar decisões é algo crucial no âmbito dos jogos. Essa atividade acontece a todo o tempo e em todos os níveis, influenciando diretamente o desempenho do jogo. Sua importância é bastante clara e pode ser percebida empiricamente, ao se analisar a partida. De fato, é impossível pensar em jogos sem considerar a ocorrência constante do processo decisório (GARCIA, 2003).

A fim de favorecer uma maior facilidade na tomada de decisão, foi utilizada, no presente trabalho, a Teoria dos Jogos, que consiste em modelos que analisam situações de conflito e de cooperação dependentes de um comportamento estratégico, em que as ações dos jogadores parcialmente dependem da conduta dos outros jogadores (CONWAY, 1977; CONWAY, 1978). Isto é, a Teoria dos Jogos é a ciência da tomada de decisões em situações de interdependência. Ao contrário de uma decisão unilateral, aqui, trata-se de decidir levando-se em conta decisões de outro(s) envolvido(s) num mesmo problema de decisão.

Uma outra teoria também foi utilizada para dar suporte à tomada de decisão, qual seja, a dos Princípios da Persuasão, que abordam o uso da comunicação para alterar atitudes, crenças ou comportamentos de outras pessoas de maneira voluntária e não por meio da força ou da coerção. Em outras palavras, o indivíduo que persuade leva o seu interlocutor à aceitação de uma determinada idéia (CIALDINI, 1998). Essas duas teorias foram modeladas e aplicadas no ambiente *Role Playing Game* (RPG), como forma de favorecer o jogador a criar suas estratégias e decisões em busca dos seus objetivos (conhecimento).

O jogo de RPG foi criado na década de 70 e tornou-se, com o passar dos anos, uma interessante atividade cultural para muitas crianças, adolescentes e adultos. Desde sua criação, vários livros foram publicados no mundo inteiro, além de terem personagens e sistemas transpostos para outras mídias, como cinema e videogames.

Durante muito tempo, pessoas do mundo inteiro têm dedicado algumas horas a um "passatempo" chamado *Role Playing Game* (RPG). O RPG é um ambiente lúdico em que se define uma história onde várias pessoas participam e cada uma delas assume o papel de um personagem e, através de sua imaginação, cria novas cenas e continua a história (PAVÃO,1996).

Contudo, este jogo vem sofrendo grandes conquistas estruturais e operacionais, uma vez que as modernas tecnologias estão proporcionando uma melhor comunicação e expressão humanas. Tais tecnologias têm gerado profundas reestruturações e remodelagens, qualitativas e quantitativas, tanto para os métodos comunicativos (recepção, armazenagem e transmissão de informação) quanto para a tomada de decisão, no que diz respeito à veiculação de conhecimento.

1.2. Motivação

Os novos experimentos em educação, que enfatizam o uso de tecnologias ligadas à comunicação, permitem corresponder às novas tendências do mundo contemporâneo, no que se refere às necessidades de se fazer uso de vários meios de propagação do conhecimento para que o ser humano tenha o maior controle do processo de aprendizagem, determinando como, quando, onde e quanto deseja aprender.

A inspiração de ambientes flexíveis, tais como os jogos colaborativos (SILVA, 2008), orientados para o âmbito do conhecimento e as necessidades individuais do sujeito, é uma consequência da emergência para novas abordagens, como o construtivismo¹, e de perspectivas pedagógicas suportadas por modelos de instrução, ícones e cenários legítimos. A mudança no entendimento de novos ambientes é

¹ Na perspectiva construtivista, o sujeito não nasce inteligente, mas também não depende do meio em sua totalidade para o desenvolvimento intelectual. Ele se forma pela interação com o meio, respondendo aos estímulos externos, analisando, organizando e construindo o conhecimento como resultado desta relação.

marcada pela flexibilidade dos procedimentos de aprendizagem, pela decisão individual sobre quais materiais trabalhar, pela identificação dos objetivos a atingir e pela fixação de uma estratégia e tomada de decisão para a construção do conhecimento. Essa abordagem distancia-se do conceito de aprendizagem baseada na aquisição e retenção sistemática do conhecimento e competências externamente existentes.

Portanto, surge a necessidade de se criar mundos virtuais que envolvam o usuário em algum tipo de trama, trabalhando com elementos lúdicos que estimulem a fantasia e, por consequência, a criatividade. Nesse sentido, o RPG é uma classe de jogos cooperativos que estimulam a socialização, a interatividade e o desenvolvimento de habilidades comunicativas (LOPES, KLIMICK & CASANOVA, 2002).

O RPG, por ser um jogo não-competitivo, quando utilizado de maneira didática, traz benefícios para as práticas de sala de aula, como interação social, cooperação, incentivo à imaginação e à expressão oral, senso de pesquisa, entre outros. O fato de ser uma atividade lúdica, aparentemente, faz com que as pessoas participem de maneira espontânea.

Para Vygotsky (1994), o desenvolvimento psicológico do indivíduo é determinado pela *interação social*, ou seja, diretamente com os membros da cultura ou indiretamente, através dos diversos elementos do ambiente culturalmente estruturado. O fundamento do funcionamento psicológico humano é social e os elementos mediadores na relação entre o homem e o mundo (instrumentos e signos) são fornecidos pelas relações entre os homens.

Outra concepção importante da teoria interacionista de Vygotsky é a *zona de desenvolvimento proximal*. A intervenção constante nesta zona por indivíduos mais experientes contribui para o desenvolvimento dos indivíduos ainda imaturos. As brincadeiras são situações imaginárias com regras implícitas e os jogos possuem regras com situações imaginárias ocultas e ambos auxiliam no desenvolvimento do indivíduo pelo fato de atuarem na *zona de desenvolvimento proximal* (VYGOTSKY, 1994). O RPG é um jogo basicamente fundamentado na interação social. Partindo destas premissas, é possível sugerir que o RPG é uma ferramenta privilegiada para propiciar o desenvolvimento psicológico e intelectual, principalmente pelo fato da interação dos sujeitos em zonas de desenvolvimento proximal.

O grande diferencial deste modelo de ambiente lúdico é que não existem perdedores ou vencedores, ou seja, não existe uma concorrência entre jogadores vilões e heróis. Ao contrário, os jogadores têm que se unir para completar os objetivos do jogo (*Quest*), existindo uma melhor tomada de decisão, troca do conhecimento prévio e social dos jogadores. Esta foi a grande motivação de desenvolver um ambiente *Massively Multiplayer Online Role-Playing Game* (MMORPG), que posteriormente pode ser utilizado pelos professores em suas salas de aulas, fazendo com que os jogadores aprendam e fixem a matéria de uma forma descontraída e, principalmente, com o trabalho em equipe, troquem informações, favorecendo, assim, a construção do conhecimento (SILVA NETO *et al.*, 2010).

É importante mencionar que a Teoria dos Jogos e os Princípios da Persuasão serão uma das ferramentas em que o jogador poderá utilizar para criar a sua melhor estratégia de tomada de decisão. Outro fator importante é que novas ações e eventos podem ser facilmente inseridos à implementação, de acordo com o contexto da história ou com o objetivo a ser atingido. Isso indica, por exemplo, que um professor poderia, além de criar a história, adicionar novas funcionalidades ao sistema conforme o conteúdo a ser ensinado através do ambiente lúdico MMORPG (CONWAY, 1977; CONWAY, 1978).

1.3. Objetivos

1.3.1. Geral

Construir um modelo de tomada de decisão baseado na teoria da persuasão aplicado ao MMORPG, que possibilite a interação de vários jogadores em um ambiente lúdico, utilizando como artifício os Princípios da Persuasão e a Teoria dos Jogos, permitindo, assim, ajudar o jogador na construção do seu conhecimento.

1.3.2. Específicos

- Criação de ferramentas que possibilitem persuadir jogadores na busca do conhecimento;

- Verificação da efetividade da aplicabilidade dos Princípios da Persuasão junto com a Teoria dos Jogos para o favorecimento da formação de novas estratégias para os jogadores;

1.4. Metodologia

Conforme exposto acima, o presente trabalho visa desenvolver um modelo que dê suporte à tomada de decisão dos jogadores quando da escolha da melhor estratégia a ser aplicada durante o jogo. Tal processo de tomada de decisão foi aplicado no âmbito do MMORPG.

Para atingir o objetivo do trabalho, tomou-se como ponto de partida a pesquisa bibliográfica no sentido compreender os conceitos e regras que envolvem o RPG e sua variação, o MMORPG. Ocorreu também o levantamento bibliográfico no domínio da psicologia social, onde se abordou os seis Princípios da Persuasão, demonstrando de que forma eles iram atuar no ambiente, bem como no âmbito dos sistemas de tomada de decisão, que permitiu entender o funcionamento e os principais tipos de sistemas de tomada de decisão. Por fim, foi realizada uma análise sobre os conceitos matemáticos, a definição formal e as equações das Redes de Petri.

Após abordagem dos principais conceitos teóricos necessários ao embasamento do trabalho, foi modelado - com base nos princípios matemáticos das Redes de Petri e nos conceitos advindos da UML - o ambiente, que engloba todo o funcionamento, ações, atores, classes e estados.

1.5. Organização

A presente dissertação foi desenvolvida em cinco capítulos.

No primeiro, é apresentado o RPG, detalhando os seus conceitos básicos, junto com o seu sistema de regras, até chegar aos RPGs digitais, mais especificamente no MMORPG.

O segundo capítulo trata da Teoria da Persuasão e os seus seis princípios, quais sejam: reciprocidade, consistência, autoridade, prova social, escassez e empatia.

No terceiro capítulo, é explanado os conceitos dos sistemas de tomada de decisão e suas propriedades, bem assim as suas principais classificações e as reais condições para a tomada de decisão.

No quarto capítulo, é apresentado o modelo proposto pelo autor, analisando o funcionamento das Redes de Petri aplicadas aos Princípios da Persuasão.

No último capítulo, após a demonstração das redes modeladas, são exibidos todos os diagramas UML modelados e seus conceitos - diagramas esses que provam o funcionamento do ambiente com suas ferramentas de apoio à decisão.

Por derradeiro, salienta-se que o propósito desse trabalho de conclusão do mestrado não é esgotar a temática proposta, mas sim ressaltar a importância da utilização da Teoria dos Jogos e dos Princípios da Persuasão aplicados no âmbito dos jogos, a fim de que possa satisfazer as necessidades dos jogadores na tomada de decisão e na busca do conhecimento.

2 O MASSIVELY MULTIPLAYER ONLINE ROLE-PLAYING GAME (MMORPG)

A sigla RPG, que significa *Role Playing Game* (jogo de representação), foi designada em 1973 nos EUA por Dave Arneson. As representações no âmbito do RPG ocorrem, basicamente, por meio da fala e da imaginação de seus jogadores durante as sessões, uma vez que uma sessão nada mais é que a continuação da aventura a partir da interrupção feita ao final da sessão anterior (HUGHES, 1988) (JACKSON, 1994) (CALE, 2002).

No modelo original do RPG, os jogadores apenas descrevem situações, comportando-se verbalmente, não existindo representação corporal das ações de seus personagens (CARDWELL, 1994). Não obstante, existem diferentes formas de jogar RPG (diferentes sistemas), mas todas apresentam elementos em comum, quais sejam: um grupo de jogadores, normalmente de três a sete, escolhem um determinado sistema de jogo, definem personalidade, ideologia, habilidades e inabilidades de seus personagens, quantificados por meio de atribuições (valores) numéricas em tabelas próprias; e designa-se um Mestre do jogo, normalmente o mais experiente dos jogadores, para ser o responsável por criar previamente um esboço de futuras aventuras desenvolvidas por todos durante as sessões (HUGHES, 1988) (JACKSON, 1994).

Os efeitos das ações dos personagens são declarados pelo Mestre, que adota um modelo matemático determinado (dependendo do sistema de RPG utilizado), estipulando uma jogada com diferentes tipos de dados (quatro, seis, oito, dez e/ou vinte faces) característicos para cada ação do jogo.

Hughes (1988) afirma que, mesmo existindo vários sistemas possíveis de serem utilizados em um jogo de RPG, o principal foco característico e que talvez tenha contribuído para o aumento da popularidade do RPG é o fato de a ação do jogo ser provocada basicamente pelo processo de imaginação dos jogadores. Há também outros recursos que objetivam dar um maior realismo ao jogo, tais como: efeitos sonoros, maquiagem, figuras, fotos, etc. Estes sistemas poderão ser vistos na sessão 2.2.

Existem, segundo Otte (2002), dois tipos de jogos em RPG: a campanha e a aventura solo. Em uma aventura solo, os jogadores unem seus personagens para uma aventura apenas, que tem um fim e um objetivo definidos e uma vitória definitiva a ser conquistada, a qual determina o fim do jogo. Em uma campanha, o foco é o crescimento dos personagens dentro do cenário, o que ocorre por meio de uma seqüência de aventuras que não possuem uma vitória definitiva, mas apenas pequenos pontos específicos a serem conquistados para a melhoria da moral, bens e capacidades dos personagens no cenário. Os personagens não competem diretamente entre si, no entanto, personagens que são menos vitoriosos durante os jogos costumam tomar parte secundária na trama, com menos capacidades e possivelmente enfrentando uma situação de 'morte' de seus personagens.

Sugere Klimick (1992) que talvez seja este o único objetivo do jogo, ou seja, o personagem deve continuar vivo, mas fazendo isso sempre no limite, assumindo riscos e vivendo situações intensas. Os personagens podem morrer. Quando isso acontece, o jogador deve criar outro diferente.

É importante salientar que um jogo de RPG normalmente envolve um grupo de amigos ou de pessoas que interagem socialmente no mundo real - falhas de caráter que não sejam impostas pela personalidade do personagem podem ser muito mal vistas socialmente. A perda ou morte de um personagem em uma aventura de RPG pode ser remediada pela criação de um novo personagem, mas demonstrações de falhas graves de caráter são difíceis de serem superadas com o grupo.

A morte ou perda de um personagem é razoavelmente comum dentro de alguns tipos de cenários, e tem a função de ampliar o fator de "jogo" dentro da mera representação e participação em uma história. Esta experiência pode ser dura para o jogador, pois, com o tempo, este, assim como um ator, cria um forte vínculo afetivo com seu personagem. Isso porque desenvolve, dentro da estrutura básica do personagem previamente ou por ele mesmo criado, uma gama de reações, comportamentos e associações ao cenário e aos outros personagens que formam um malha sentimental muito delicada. Mas, acima de tudo, deve ser claro para todos os jogadores que se trata de um jogo e o mais importante é o aprendizado e o crescimento do jogador e não do personagem.

2.1. Conceitos básicos

Como toda a atividade, o RPG tem também uma linguagem própria. Por exemplo, os narradores são chamados de mestres, os ouvintes/participantes de jogadores, e as histórias de aventuras. Os conceitos básicos do RPG são classificados da seguinte maneira, segundo Debbio (1998).

- Jogador: Os jogadores são aqueles responsáveis por um ou mais personagens (PC, *player character*, personagem do jogador) dentro da trama, dentro do sistema de regras e do cenário do jogo, e tem controle e liberdade de ação com seu ou seus personagens;
- Mestre do Jogo: É responsável por todos os fatores do cenário e da trama que não envolvem as ações dos personagens (estas são de exclusividade dos jogadores). Controla os personagens da trama que interagem com os personagens dos jogadores, mas que não são controlados por eles (NPC, *Non-Player Character*, personagem que não do jogador). Controla o cenário, adaptando a medida da necessidade da trama e, por fim, é o responsável absoluto pelos objetivos secretos e desenrolar da trama. Como todos em um jogo de RPG, o Mestre do Jogo deve seguir e respeitar o sistema de regras, mas, para um bom desenrolar da trama, ele pode mudar as situações dentro de uma lógica razoável;
- Sistema de Regras: As ações que os jogadores tomam para seus personagens são indicadas ao mestre, que consulta o Sistema de Regras para se obter o resultado desta ação para o Personagem em questão dentro da situação em que ele se encontra. Para isso, existem regras claras para variadas situações e indicações de como proceder para situações inesperadas, ficando a cargo do Mestre do Jogo lidar com situações não definidas. É comum a utilização de um elemento aleatório, como um dado para se simular a incerteza do resultado das ações que é comum ao mundo real;
- Cenário: O cenário trata-se do mundo em que os jogadores estão, tanto geograficamente quanto temporalmente e às vezes mesmo em que

dimensão ou plano astral os personagens podem se encontrar, com realidades cotidianas ou totalmente diferentes do normal. Cenários costumam ser definidos em conjunto pelos jogadores e pelo mestre e são escolhidos por possuírem capacidades de comportar os personagens e tramas que interessam aos jogadores;

- **Personagens:** Os personagens são as projeções da fantasia dos jogadores dentro do cenário proposto pelo grupo de jogo. Eles podem ser construídos pelo jogador ou oferecidos como personagens prontos, mas são obrigatoriamente montados dentro do Sistema de Regras e Cenário estabelecidos pelo jogo. Dependendo da Trama, alguns personagens podem ser mais ou menos interessantes. Todos jogadores possuem habilidades que definem suas capacidades de interagir com o cenário. Tais habilidades são definidas pelo Sistema de Regras e adquiridas pelo jogador a medida de seu interesse na construção de um personagem específico;
- **Trama:** Dentro do Cenário, os Personagens unem-se para algum tipo de aventura ou ação em conjunto, a isto é chamado Trama. Os personagens, na maioria do tempo, estarão seguindo a Trama, encontrando pessoas, objetos ou localidades importantes para o desenrolar da Trama, mesmo que, na maioria das vezes, possam não saber exatamente quais são os objetivos ou que estão tomando decisões importantes dentro da mesma;
- **NPCs (*Non Player Charater*):** Personagens não jogadores. Termo emprestado de outros RPGs para designar personagens que são controlados pelo mestre e não pelos jogadores. NPCs geralmente servem como personagens coadjuvantes nas aventuras.

2.2. Sistema de Regras e Mestre do Jogo

Como não há apenas um sistema de regras (cada jogo de RPG costuma ter o seu), as possibilidades de jogos de RPG se multiplicam ainda mais, pois cada ambientação pode ser desenvolvida por diferentes sistemas de regras, segundo Klimick (1992). Ou seja, cada combinação ambientação-sistema dá origem a um jogo diferente. Além disso, a mesma ambientação pode ser desenvolvida de maneiras diferentes.

Há também as ambientações ficcionais, o que torna impossível imaginar um limite para as possibilidades de ambientação. Klimick (2001) fala daquelas inspiradas em filmes (Star Wars, Indiana Jones) ou na literatura (H.P. Lovecraft, Anne Rice, J.R.R. Tolkien).

De acordo com Klimick (2001), o jogador cria um personagem para a ambientação escolhida. Este personagem será criado de acordo com um sistema de regras como qualquer outro jogo, e obedecendo a lógica do mundo para o qual ele está sendo criado. Por exemplo: um personagem de velho oeste não vai poder voar ou ter uma arma de raio laser.

O sistema de regras de RPG é um livro que descreve um determinado “universo” de ficção, com características e regras próprias. Há “universos” medievais, futuristas e fantásticos. Há “universos” que abordam o jogo político, espionagem, e até a história brasileira. Cada sistema descreve cidades, características dos seus habitantes, seu modo de vida e ensina a criar aventuras ambientadas naquele “mundo”.

Klimick (1992) sugere que o sistema de regras serve para organizar a ação dos personagens durante o jogo, determinando os limites do que ele pode ou não pode fazer. Por exemplo: não basta um personagem saber atirar para acertar um alvo. Vai depender do alvo e das condições em que o personagem se encontra, além do bom atirador que ele é. O sistema de regras tem como finalidade fazer uma simulação da realidade (a realidade do jogo), influenciando a ação dos personagens nas ações mais complexas.

No RPG, raramente os jogadores respeitam fielmente as regras, segundo Klimick (1992). Trata-se de um jogo de fantasia, que nasce da fantasia do autor ou de um grupo de autores. O sucesso de um jogo nasce da capacidade desta criação representar a fantasia dos jogadores, assim como uma criação artística. Porém, é quase impossível que esta fantasia represente 100% as fantasias dos seus jogadores. Desta forma, como se trata de um jogo que estimula a criatividade e a imaginação, os jogadores, mais cedo ou mais tarde, acabam modificando um pouco aquela fantasia e adequando-as as fantasias do grupo. O mesmo ocorre com o sistema de regras. Sempre há uma situação da realidade que os jogadores julgam não estar bem representada pelas regras e a modificam.

O mestre escreve a situação inicial, que funciona como motivação para os jogadores. A partir dessa descrição, os jogadores começam a inserir seus personagens na trama e o mestre vai conduzindo o problema a partir das ações dos jogadores. Muitas vezes, utilizam-se de dados para verificar se a ação do jogador foi bem-sucedida ou não, como também utiliza-se da observação da ficha do personagem para verificar as habilidades e competências do jogador para desenvolver a ação. O mestre também pode se utilizar de cartas ao acaso, para definir o destino do personagem e a resolução da trama. No entanto, o mestre joga a favor da história. Suas decisões devem levar em consideração o que é melhor para o desenvolvimento das resoluções a serem tomadas.

O mestre não se limita apenas a narrar a história e descrever os cenários; ele também é responsável pela interpretação de todos os personagens da trama que não forem "controlados" pelos jogadores, normalmente os vilões, os coadjuvantes, e eventuais criaturas, animais ou outros seres. Os personagens controlados pelo mestre são chamados de NPC (*Non Player Character*). A função do mestre não é jogar contra os jogadores, nem a favor. O mestre joga a favor da história e da diversão (KLIMICK, 1992).

Suas decisões devem levar em conta o que é melhor para o desenvolvimento da história e para a diversão dos participantes. Muitas vezes, ajudar um jogador pode desagradá-lo, pois o desafio e a superação de limites é o que costuma dar forma ao jogo. Estes limites e desafios não são necessariamente físicos, como invadir um castelo ou vencer um combate, mas também de essência dramática, como deixar de conseguir a tão sonhada vingança para salvar a vida de um companheiro ou a própria dificuldade em interpretar o seu personagem (KLIMICK, 1992).

Uma aventura pode ter um objetivo definido: salvar a princesa, encontrar um tesouro, deter uma invasão; ou pode ser uma história sem fim, na qual os personagens passam de uma aventura a outra. A escolha cabe ao mestre e aos jogadores. Uma campanha² pode durar anos e, nesse tempo, ver a chegada e a partida de personagens (e jogadores). Para o ambiente educacional, a função do mestre não seria criar a

² Uma sucessão de aventuras, envolvendo a saga dos personagens, desde a primeira partida.

aventura, mas tornar a objetividade uma aventura dirigida para entender um objetivo de aprendizagem.

2.3. RPG e compartilhamento do conhecimento

O RPG é uma ferramenta para a criação de simulações práticas, vivenciais, incentivando a criatividade, a participação, a pesquisa e a integração das pessoas. Ele é adaptável a qualquer conteúdo didático, ambiente ou público.

Os jogos podem ser empregados em uma variedade de propósitos dentro do contexto de aprendizado. Um dos usos básicos é a possibilidade de se construir autoconfiança e o incremento da motivação (MARCATO, 1996). Para Fernandes (1995), até mesmo o mais simplório dos jogos pode ser empregado para proporcionar informações factuais e praticar habilidades, conferindo rapidez e competência aos jogadores.

A representação de um objeto, a partir de um jogo, organiza os conhecimentos. Numa visão construtivista, os conhecimentos estão sujeitos às estruturas e eles são interdependentes, de forma que a aquisição de um conhecimento pode gerar a estruturação de muitos outros que haviam sido adquiridos anteriormente. Portanto, o conhecimento do indivíduo não é composto de partes, mas de um conjunto de relações.

No jogo, é possível compartilhar emoções, crenças, valores e conhecimentos. Ao compartilhar aventuras e situações, os membros trocam sentimentos daí resultantes, ou seja, as emoções perceptíveis, mas não expressas. Todo o processo de trabalho conjunto é beneficiado pelos membros da equipe.

Como o conhecimento tácito está vinculado aos sentidos, às experiências pessoais e aos movimentos corporais, ele pode, a partir do jogo, ser transferido aos outros. Segundo Krogh (2001), existem algumas maneiras de compartilhar o conhecimento tácito:

- Observação direta – os membros da comunidade observam as tarefas e as habilidades dos demais na execução de uma atividade. Os observadores passam a compartilhar crenças sobre as ações eficazes e ineficazes, aprimorando a capacidade de agir em situações semelhantes;

- Observação direta e narração – os membros observam as tarefas e recebem explicações dos demais sobre a execução do trabalho, geralmente na forma narrativa sobre situações semelhantes. As crenças dos observadores são reforçadas ainda mais por essas histórias;
- Imitação – os membros tentam imitar os métodos de execução da tarefa, com base na observação direta dos outros membros;
- Experimentação e comparação – os componentes do grupo experimentam várias soluções e em seguida observam o trabalho de um especialista, comparando o trabalho deste com o seu;
- Execução conjunta – os membros da comunidade tentam executar a tarefa em conjunto. Os mais experientes oferecem dicas e idéias sobre como melhorar o desempenho dos menos experientes.

Dentro da tipologia apresentada acima, a *execução conjunta* é a que mais se assemelha às regras aplicadas no desenvolvimento do RPG. O Mestre descreve a situação e diz aos jogadores o que os seus personagens vivem, vêem e ouvem, enquanto os jogadores descrevem o que eles estão fazendo para vencer o desafio.

O processo de aprendizado, no entanto, vai envolver uma mistura de observação, imitação, narração, experimentação e execução para que os envolvidos no processo compartilhem o conhecimento tácito. Ele é compartilhado por meio de uma profunda socialização dos membros da organização, principalmente aqueles ligados à criação de conhecimento ou, como denominam Krogh (2001), a micro-comunidade de conhecimento. Para esses autores, é com a socialização que os membros da comunidade não apenas compreendem as perspectivas de cada um sobre as situações compartilhadas, mas também chegam a um consenso sobre uma visão comum e sobre a maneira de agir em determinada situação. Tais situações podem envolver, por exemplo, a execução de um projeto de engenharia, a identificação das necessidades de um cliente ou a análise de sofisticadas especificações técnicas (KROGH, 2001).

O conhecimento tácito, embora seja o mais importante para as organizações, dificilmente pode ser expresso e recuperado em documentos. Na visão de Krogh (2001), “a única maneira de recobrar esse capital organizacional é recriar as condições singulares daquela micro-comunidade de conhecimento – tarefa difícil, se não

impossível, considerando até que ponto as relações sociais apresentam características espontâneas e aleatórias”.

O jogador não é um mero espectador, mas ele é um ator, representa um papel, cria uma personagem; como um roteirista, escolhe caminhos e toma decisões nem sempre previstas pelo Mestre, contribuindo na recriação da aventura. Durante o jogo, os jogadores buscam elementos a partir de leituras, pesquisas e troca com os colegas em um misto de diálogo, de decisão e de consenso com o grupo para criar os personagens.

A troca de experiências e de conhecimentos é a condição básica para o sucesso das aventuras do grupo. Nesse sentido, o RPG é um mediador de aprendizagens, na medida em que os conhecimentos podem ser construídos e socializados, principalmente pelas leituras e pela produção escrita. Os jogadores escrevem e lêem por prazer, sentam-se em uma mesa com seus companheiros e produzem uma narrativa de anônimos, sem cobranças, sem formalização acadêmica dessa escrita; aprendem uns com os outros e aventuram-se por uma viagem em nível imaginário que contribui para a constituição de um conhecimento partilhado num espaço de vivência do prazer.

2.4. Surgimento dos RPG digitais

Os primeiros jogos de computador da história surgiram no fim dos anos 50 e começo dos 60. Foram jogos não gráficos baseados em terminais, chamados de MUDD (*Multi-User Dungeons&Dragons* – Labirintos e Dragões para Múltiplos Usuários), nos quais os jogadores incorporavam um herói que explorava mundos descritos nas telas do computador, em busca de poder e riquezas. Estes primeiros jogos eram essencialmente verbais: o computador era programado para responder primariamente com palavras para quaisquer das opções do jogador. Dentro dessa premissa simples, eram jogos extremamente complexos com uma infinidade de variações, opções, combinações de ações contra ações que ofereciam uma grande liberdade para o jogador, sendo que o banco de dados destes jogos era realimentado e ampliado constantemente.

Com a evolução dos computadores, houve uma grande evolução na parte gráfica dos jogos, oferecendo ao jogador cada vez mais recursos multimídia como botões e atalhos para se lidar com as dificuldades de uma aventura.

2.5. RPG na *Internet* e o MMORPG

Com o lançamento dos jogos de RPG computadorizados, que permitia o jogo na modalidade *multiplayer*, ou seja, em que se permitem vários usuários jogarem em uma rede local, modem ou pela *Internet*.

A seguir, serão listadas as principais características dos RPG's digitais na *Internet*:

- Possibilita a interação de múltiplos jogadores;
- Vasta exploração do(s) mundo(s) com vastas localidades existentes;
- A existência de diversas sub-tramas, permitindo o jogador criar sua própria história e aventuras;
- Uma grande semelhança com os RPG's de mesa, pelo fato da permissão de criação e evolução dos personagens. De forma geral, utiliza um sistema de regras já consagrado no RPG de mesa, como exemplo, o *D&D*³, pela vivência da cooperação e possibilidade de interpretação;
- Numa grande maioria de jogos, o usuário pode personalizar o personagem principal, como exemplo, criar aventuras, itens, armas e mundos;

Não menos importante que a interação, há a capacidade de os jogadores poderem construir objetos neste mundo bastante semelhantes aos MUDD (anteriormente citados). Resumindo, a grande diferença entre os MUDD e os MMOG é que os MUDD são uma espécie de realidade virtual baseada em texto e os MMOG possuem ricas interfaces gráficas. Nesse viés, pode-se afirmar que os MMOG's são uma evolução dos MUDD, e os MMORPG's são apenas uma modalidade dos MMOG's.

Os MMORPG's têm como principal característica a intervenção constante de uma equipe de Mestres de Jogo (NPC's ou humanos desenvolvedores do mundo) e atuam na criação de tramas e desafios para os personagens dos jogadores. As tramas

³ *Dungeons & Dragons*: labirintos e dragões.

não são lineares com começo, meio e fim: existe um conceito de mundo virtual para ser explorado, uma história em aberto. Outra característica do MMORPG, notada no termo *Massively Multiplayer Online Role-Playing Game*, é que a modalidade permite o acesso simultâneo de milhares de jogadores atuando no mundo “virtual”, como exemplo, o *World of Warcraft* (BLIZZARD ENTERTAINMENT, 2010) - MMORPG de maior sucesso na atualidade, que no início de 2010 tinha mais de 11 milhões de jogadores (FADEL, 2009).

De forma resumida as principais características da modalidade MMORPG, são:

- Permite que milhares de jogadores interajam em um mundo virtual (dependendo apenas dos *hardwares* que serão utilizados como servidores, permitindo o acesso de centenas ou milhares ou etc.);
- Exploração do mundo extremamente grande, uma vez que o jogador não ficaria obrigado a seguir uma rota ou lugar do mapa a cada momento;
- Tramas não lineares, existindo um mundo virtual para ser explorado;
- Alto grau de interpretação e cooperação, pelo fato de que os jogadores mantêm uma “vida” virtual;
- Possibilita a personalização do personagem, bem como a criação de objetos pertencentes ao mundo virtual, que poderão ser acessíveis por outros personagens;

Segundo relatório da *International Game Developers Association* - IGDA (IGDA, 2006) sobre jogos online, inclusive sobre os MMORPG, os jogadores precisam de intensa interação social e muitos deles buscam posições de alto status no mundo do jogo. O jogo deve dar suporte a sociabilização por meio do comércio (venda e compra de itens de jogo), aperfeiçoando habilidades através de treino e atividades cooperativas. Outro fato citado no relatório é o tempo de duração das sessões de jogo, que chegam ter em média doze horas. Além disso, os jogadores tendem a permanecer por muito tempo na comunidade do jogo em razão das amizades virtuais formadas no decorrer do tempo jogado.

2.6. MMORPG no contexto do comércio

A *Internet* vem revolucionando quase todas as atividades que os seres humanos utilizam para ganhar a vida, tal como ocorre com a atividade terceirizada. Os países de terceiro mundo normalmente possuem mão-de-obra de baixo custo, o que atrai os empregadores dos países ricos, que contratam esses trabalhadores como uma medida de reduzir os custos com a mão-de-obra. Uma indústria que está se beneficiando com tal forma de contratação são os jogos online (SRINIVASAN *et al.*, 2002).

Desde o surgimento do *World of Warcraft*⁴, que tem um total de milhões de inscritos, passou a existir grandes possibilidades de se lucrar economicamente no mundo virtual dos MMOs, bastando apenas ter uma mente perspicaz com o objetivo de ganhar dinheiro. Nestes mundos virtuais, há várias maneiras de se lucrar financeiramente direta ou indiretamente e até mesmo desonestamente.

Em razão da existência de um comércio desonesto, os desenvolvedores dos jogos criaram penalidades para o comércio desleal, por meio de uma ferramenta denominada *Trade*. Na Figura 1, visualiza-se como funciona um *Trade* na maioria dos jogos MMOs: (A) Encontra-se o valor a ser pago por determinado item; (B) Lista de Itens a ser negociado; (C) Opção que o jogador tem de *Trade* (confirmar) ou cancelar a venda.

Em quase todos os jogos MMOs, existem diferentes tipos de “shoppings” de item, o que permite gerar um comércio entre os jogadores. Os jogadores que geralmente comercializam tais itens em grande demanda e qualidade tendem a ter alguma influência no mundo virtual. Ou seja, seus itens podem ser mais caros e conseqüentemente mais raros, semelhantemente do que ocorre no comércio real, onde grandes empresas cobram valores bem maiores daqueles cobrados normalmente pelo mercado por conta da qualidade de seus produtos, garantia e etc (SRINIVASAN *et al.*, 2002).

⁴ Disponível em: <<http://us.battle.net/wow/en/>>. Acesso em: 16 dez. 2010.

Jogadores de alto nível normalmente em troca de dinheiro virtual e até mesmo de moeda real, oferecem itens muito difíceis de se adquirir, não sendo incomum, portanto, encontrar jogadores que vendem itens considerados raros em troca de dinheiro real. No entanto, ressalta-se que este tipo de prática ocorre de maneira discreta, pois as empresas desenvolvedoras dos mundos virtuais não permitem tal tipo de compra e venda, aplicando como única penalidade para essa espécie comércio a exclusão da conta.

Por fim, é importante salientar que cada MMO tem a sua determinada forma de negociação, uma vez que se desenvolve o comércio de modo a facilitar o intercâmbio ou venda ou troca de materiais de acordo com as características do ambiente MMO (SRINIVASAN *et al.*, 2002).



Figura 1: Trade do *World of Warcraft*.

2.7. Conclusão do Capítulo

No Capítulo 2, salientou-se a modalidade de jogo MMORPG convencionais, analisando o RPG e seus sistemas de regras e funcionamento, que, por sua vez, não se preocupam em utilizar estratégias de jogos e em ser uma ferramenta apta a construção do conhecimento. Salientou-se, então, que tais jogos possuem falhas quanto à busca

de uma eficiente estratégia de jogo, uma vez que ainda são carentes no que se refere ao desenvolvimento de melhores práticas de persuadir e de tomada de decisão.

3 A TEORIA DA PERSUASÃO

A fim de propiciar um melhor entendimento do modelo proposto neste trabalho, mostra-se necessário, antes de apresentá-lo, realizar um breve estudo sobre os princípios da persuasão, abordando os seus conceitos e fundamentos. Isso porque, com base nesta teoria, demonstrar-se-á como se pode aplicar a Persuasão aos MMORPG, propondo um modelo de aprendizado que facilite a troca de informações, aprendizado social e a tomada de decisão.

3.1. Os princípios da persuasão

Segundo Robert Cialdini (1998), persuasão é o uso de comunicação para alterar atitudes, crenças ou comportamentos de outras pessoas. É importante ressaltar que essa mudança deve ocorrer voluntariamente e não através da força ou da coerção. Em outras palavras, o indivíduo que persuade leva o seu interlocutor à aceitação de uma determinada idéia.

Para Citelli (1986), na própria etimologia da palavra, está embutido um conselho irônico: per + suadere = aconselhar. “Essa exortação possui um conteúdo que deseja ser verdadeiro: alguém “aconselha” outra pessoa acerca da procedência daquilo que está sendo enunciado”. Segundo o referido autor, é possível que o persuasor não trabalhe com uma verdade, mas com algo que seja verossímil. Nessa perspectiva, persuadir não é enganar, mas sim é o resultado de certa organização do discurso que cria um efeito de sentido e o constitui como verdadeiro para o interlocutor.

Os processos de influência e de persuasão são estudados desde a Grécia Antiga, com a Retórica de Aristóteles como referência máxima, mas continuam ainda hoje a fascinar os investigadores na área das ciências sociais, particularmente na área da psicologia social. Estudar os mecanismos que levam alguém a estar em melhores condições para persuadir, ou ser persuadido, tem sido preocupação de vários investigadores cujos trabalhos desmistificaram a idéia da persuasão como uma arte controlada por uns poucos, colocando-a num patamar científico e disponível para ser aprendida por qualquer pessoa.

O discurso persuasivo busca, portanto, representar “toda a verdade” por meio de recursos lingüísticos que são selecionados como expressões de “uma verdade”, instaurando uma superposição. Nesse sentido, o objetivo maior do discurso persuasivo é, por meio de recursos retóricos, “convencer ou alterar atitudes e comportamentos já estabelecidos” (CITELLI, 1986).

O estudo das estratégias de persuasão também tem sido um tópico de interesse por parte de vários psicólogos sociais, com especial destaque para Robert Cialdini (1998). Este autor definiu seis princípios gerais de persuasão, nas quais se destacam várias táticas e estratégias que podem ser usadas para persuadir quem desejar, quais sejam: reciprocidade, consistência, autoridade, prova social, escassez e empatia.

3.1.1. Reciprocidade

O aspecto relevante do princípio da reciprocidade é o do sentido da obrigação que se passa, tornando-se onipresente na cultura humana, afirmam os sociólogos, como Alvin Gouldner (1960), pois, segundo esse autor, não há sociedade humana que não se inscreva à regra. O arqueólogo Richard Leakey (1978) atribui a essência daquilo que nos torna humanos para a reciprocidade: "Nós somos humanos porque nossos antepassados aprenderam a compartilhar seus alimentos e suas habilidades em uma rede comunitária".

É o sentimento de obrigação que é crucial para existir o avanço social. Isto significa que uma pessoa poderia dar alguma coisa (por exemplo: alimentação, energia, atenção) para o outro com a confiança de que ele não estava sendo perdido. O resultado foi a diminuição das inibições naturais contra as operações que deve ser iniciada por uma pessoa - o fornecimento de recursos pessoais para o outro. Sistemas sofisticados e coordenados de ajuda, como dar presentes, a defesa, e o comércio tornaram-se possíveis, trazendo imensos benefícios para a sociedade. Com essas consequências claramente adaptativas para a cultura, não é de se estranhar que a regra da reciprocidade é tão profundamente implantada nas pessoas pelo processo de socialização.

As sociedades humanas competitivas seguem o princípio da reciprocidade e, conseqüentemente, esperam que seus membros respeitem e acreditem nesse princípio. Cada pessoa foi ensinada a viver de acordo com a regra, todos sabem sobre as sanções sociais e sobre os escárnios aplicados a quem o viola. Há rótulos que são atribuídos às pessoas que não as cumprem, tais como ingrato ou caloteiro. Isso porque é desagradável para a sociedade em geral deparar-se com pessoas que tomam “favores” e não fazem nenhum esforço para retribuí-los.

Esta pequena regra social tácita permite - quando convenientemente explorada - persuadir⁵ e influenciar outras pessoas de uma forma eficaz e que dificilmente será vista como persuasiva ou manipulatória.

Uma das razões da reciprocidade ser tão eficaz é o fato desta regra possuir força, muitas vezes produzindo uma resposta "sim" a um pedido que, salvo na existência de um endividamento, certamente teria sido recusado.

Há ainda uma outra característica da regra da reciprocidade que é aquela que permite que esta seja explorada para o lucro. Paradoxalmente, a regra desenvolvida para promover a igualdade, pode também ser usada para trazer resultados desiguais. Pequenos favores iniciais podem produzir uma sensação de obrigação.

Quando confrontada com um solicitador que emprega a regra da reciprocidade, enfrenta-se um adversário formidável. Seja por apresentar um primeiro favor ou concessão inicial, o requerente terá que recorrer a um poderoso aliado na campanha para adquirir o respeito. Poderia dar cumprimento ao desejo do solicitante e, assim fazer sucumbir a regra da reciprocidade. Ou, poderia recusar a cumprir e sofrer o impacto da regra sobre sentimento profundamente condicionado da equidade e da obrigação.

As estratégias da reciprocidade, de aparência simples, tornam-se extremamente eficazes e quase indetectáveis. A melhor solução para combater estas estratégias é refletir antes de se aceitar qualquer favor ou concessão da parte de alguém cujas verdadeiras intenções se desconhece.

⁵ Convencer ou aconselhar alguém, mudando o ponto de vista de forma não imposta.

Apesar de se tratar de uma norma que é explorada no sentido de manipular e influenciar, a reciprocidade é também um pilar fundamental da sociedade humana, e um dos motivos para o desenvolvimento das primeiras comunidades de seres humanos.

3.1.2. Consistência

Trata-se do desejo de ser (e aparecem) consistente com o que já foi feito. Ao realizar uma escolha ou tomar uma posição sobre uma determinada questão, pessoal ou interpessoal, deve-se portar de forma coerente perante à sociedade. Estas pressões irão levar a reagir de maneira que venha justificar as decisões anteriores (CIALDINI, 1998).

Este princípio também utiliza como artifício a norma social. Tem-se uma exploração da pressão social para se manter consistente com as crenças, atitudes ou comportamentos preponderantes na sociedade, que, por fim, devem ser vistos de forma positiva pelos outros.

Os psicólogos têm compreendido por muito tempo o poder do princípio da consistência para direcionar a ação humana. Tal princípio atrai o desejo à coerência como um motivador central do nosso comportamento. Mas, a tendência para ser coerente é realmente forte o suficiente para obrigar a fazer o que ordinariamente não se gostaria de fazer. A unidade a ser (e aparência) consistente constitui uma arma muito potente de influência social, muitas vezes levando a agir de maneira que são claramente contrários aos interesses.

Para entender por que a regra da consistência é tão poderosa, é importante reconhecer que, na maioria das circunstâncias, a consistência é valorizada e adaptativa. A incoerência é comumente tida como um indesejável traço de personalidade. Pessoas, cujas crenças, palavras e atos se contradizem, podem ser vistas como indecisas, confusas, “duas caras”, ou até mesmo mentalmente doentias. Por outro lado, um elevado grau de coerência é normalmente associado com a força pessoal e intelectual. É no centro da lógica, da racionalidade, estabilidade e honestidade.

O quebrar de um compromisso ou o alterar de um padrão de comportamento anterior é não só visto de forma negativa pelos outros, como provoca desconforto

psicológico para a própria pessoa, que tentará procurar minimizar estes problemas, comportando-se muitas vezes de forma prejudicial aos seus interesses. Tal como acontece com a reciprocidade, a consistência está de tal forma enraizada nos nossos padrões sociais que a sua ativação é praticamente espontânea.

Uma das formas mais comuns de explorar este princípio persuasivo é o de utilizar um comportamento passado do alvo da persuasão e torná-lo saliente na altura de implementar a estratégia. Na falta de dados passados, o recurso a um compromisso público, através da concordância com uma determinada premissa, também produz resultados eficazes.

Quanto mais pública e aparentemente menos forçada for a declaração do sujeito, maior será o seu sentimento de compromisso e a sua tendência em se comportar de forma consistente com a sua posição. Outra vantagem desta estratégia é que vai se reforçando ao longo do tempo, na medida em que os sujeitos vão procurando e definindo novas razões que expliquem o seu comportamento na tentativa de se manterem consistentes.

A maior prova de que as pessoas realmente sentem e acreditam vem menos de suas palavras do que de seus atos. Observadores tentando decidir o que é um “modelo” de homem é como olhar de perto suas ações. O comportamento diz muito sobre cada pessoa, pois constitui uma fonte primária de informação sobre suas crenças, valores e atitudes (CIALDINI, 1998).

Tal como ocorre com a norma da reciprocidade, defender-se do poder da consistência não é simples e pode ter custos pessoais e sociais. Ser consistente com as suas ações é um traço de personalidade muito apreciado na nossa cultura, por se tratar de um mecanismo psicológico, que pode ser explorado, de maneira útil e adaptativa.

3.1.3. Autoridade

Antes de iniciar o detalhamento do terceiro princípio, será preciso deliberar em que contexto é definida a palavra “autoridade”. Tal termo não se limita apenas no poder que advém de um cargo, mas também se refere ao estatuto atribuído a alguém com base nos seus conhecimentos ou perícia relativamente a um dado assunto ou ofício. A

título de exemplo, pode-se considerar um policial ou o chefe como alguém que possui autoridade que advém dos cargos que ocupam, mas também se pode considerar como autoridade, num dado campo do conhecimento, um professor ou estudioso que se dedicou a um determinado assunto (CIALDINI, 1998).

Um aspecto fundamental a se destacar é o fato de a autoridade ser uma característica que é notada pelo uso da comunicação, uma vez que não basta alguém auto proclamar-se como uma autoridade. É necessário que a audiência, ou seja, que meio social reconheça, na prática, esta qualidade, mesmo não sendo ela, formalmente, real.

Quando se refere à palavra autoridade, fala-se também de pessoas que detêm determinados poderes reais ou simbólicos, que, por si sós, são capazes de conseguir a anuência dos outros membros. Isto é, esse princípio não induz as pessoas a obedecer à autoridade por medo das represálias, mas sim por que se acredita que essa é a melhor solução, tendo em conta que foi admitida por alguém que é supostamente um entendedor no assunto.

A obediência unânime à autoridade é justificada pela própria posição por ele ocupada na organização social. Esta obediência, por sua vez, permite o desenvolvimento de estruturas sofisticadas para recurso de produção, comércio, defesa, expansão e controle social, que de outra maneira seria impossível de ocorrer. Em razão desses fatores positivos, desde o nascimento, a pessoa é treinada a obedecer à autoridade constituída, sendo essa ação atribuída como sendo a correta e, por conseqüência, a desobediência dada como conduta errada.

Deve-se ressaltar que a instrução religiosa contribuiu também na formação do princípio da autoridade, mas se utilizando de reprimendas. Por exemplo, o primeiro livro da Bíblia descreve que Adão e Eva foram punidos com a perda do paraíso pela falta de obediência à autoridade máxima. Nesse episódio, extrai-se a lição de que a obediência à autoridade é muito mais benéfica, uma vez que o não respeito às regras por ela editadas provocará a aplicação de castigos, sejam morais ou patrimoniais.

O poder persuasivo das autoridades é facilmente explicável: os cargos que ocupam ou os conhecimentos que demonstram colocam essas pessoas como as mais credíveis para emitir opiniões ou dar conselhos acerca de um dado assunto. Esta é uma

lógica que permite que o mundo social funcione sem grandes problemas, pois a atuação da autoridade traz conseqüências benéficas no sentido atribuir segurança às relações sociais. Por exemplo, caso um policial indique que é melhor não estacionar o carro perto de uma curva, as pessoas reconhecem a autoridade necessária para que essa indicação seja cumprida, do mesmo modo ocorre com o médico, que prescreve determinados medicamentos e não outros.

Tal como as restantes normas da persuasão, a norma da autoridade atua de forma mecânica, pois se leva a concordar com algo sem que se perca tempo em pensar se é o mais correto a se fazer. Isso permite que decisões propostas e evidentemente nocivas, simplesmente por serem tomadas por alguém reconhecido como uma autoridade, sejam acatadas por outras pessoas, apesar de não verificar se é a melhor solução para um problema.

Para se evitar o uso equivocado dessa norma, deve-se tentar definir se a suposta autoridade é relevante para o assunto em questão. Deve-se também tentar separar a opinião da pessoa que a emite, de forma que se avaliem os méritos desta e não a aceite com base apenas no estatuto de quem a emitiu.

Uma tática de proteção que pode ser usada contra o status da autoridade é prevenir-se de seu respeito mecânico, avaliando a sua atuação e as ordens por ele emitidas. Isso porque, sendo a obediência automática, as pessoas acabam interpretando de forma errônea a intenção da autoridade, podendo isso impedir que elas descubram a real intenção daquele que detém o poder, que muitas vezes está imbuído de interesses imorais e ilícitos.

Desse modo, a forma fundamental de defesa contra este problema é uma maior consciência do poder de autoridade. Quando esta consciência é acoplada a um reconhecimento de como o símbolo da autoridade pode ser facilmente falsificado, o benefício será uma abordagem devidamente correta, quando se tenta influenciar com o status da autoridade (CIALDINI, 1998).

Mas, não se deve querer resistir completamente, ou na maioria das vezes, ao princípio da autoridade, visto que, geralmente, as figuras da autoridade sabem o que estão falando. Os médicos, os juízes, os executivos, os líderes legislativos, e entre outros, tipicamente ganharam suas posições por conta do conhecimento superior e do

juízo perante a sociedade. Assim, como regra geral, suas diretrizes oferecem excelentes conselhos. Portanto, a dica é ser capaz de reconhecer quando propostas das autoridades são as melhores opções ou quando devem ser combatidas, por camuflarem interesses escusos.

3.1.4. Prova Social

Grande parte do comportamento, público e privado, é condicionado pelas normas sociais e pelo fato de querer dar a melhor impressão aos outros. Muitas vezes, para conseguir essa aprovação e sentir-se integrado, ou para que não se perceba que está desenquadrado, utiliza-se o comportamento de outras pessoas como pistas de como se deve comportar. A isso é chamado de prova social, ou seja, a utilização do comportamento de outras pessoas como modelos de conduta em situações ambíguas⁶ (CIALDINI, 1998).

No processo de examinar as reações de outras pessoas para resolver as incertezas, é bem provável que se ignore um importante mecanismo, qual seja, a evidência social - que consiste em avaliar a adequação da conduta em uma dada situação. Pois, em situações ambíguas, a tendência que as pessoas têm de checar qual o comportamento adotado pela maioria pode provocar a chamada "ignorância pluralística". Tal conceito foi criado por Floyd Henry Allport, que definiu este fenômeno como sendo o processo, por meio do qual, membros de um grupo - embora possuam percepções, crenças, ou atitudes diferentes do restante do grupo - acabam adotando o comportamento da maioria.

Ou seja, ocorre a "ignorância pluralística" quando os dissidentes, apesar de não apoiarem a norma do grupo, atuam como os outros membros por acreditarem que o comportamento adotado revela a posição unânime do grupo. Desse modo, mesmo discordando com aquela posição, os dissidentes acabam atuando como se concordassem com a mesma, a fim de não se distinguir dos demais, uma vez que, caso contrário, a conduta seria aprovada por todos, menos por eles. Nesse viés, percebe-se que, por conta da ignorância pluralística, as pessoas conformam-se com a opinião

⁶ Várias opções de escolha.

consensual percebida num grupo, em vez de pensar e agir de acordo com as suas próprias perspectivas (TRACY, 2003).

Sem dúvida, quando as pessoas estão incertas, é comum que elas estejam mais propensas a analisar as ações de outros para decidir a maneira que se deve agir. Mas, além disso, há uma outra condição importante: a similaridade. O princípio da prova social atua melhor, quando as pessoas começam a se identificar com o comportamento das outras. Isso porque é mais fácil seguir a liderança de um indivíduo semelhante do que as ordens daquele que possui condutas distintas.

O princípio da prova social é simples e faz todo o sentido no mundo em que cada vez mais se depara com situações novas. A lógica de “se as pessoas fazem, é porque é o mais correto” permite assumir um comportamento adequado num contexto desconhecido. Isso ocorre em inúmeras situações do dia-a-dia, como, por exemplo, para se saber qual a forma correta de estar em uma cerimônia oficial de estado ou para decidir qual o melhor filme a ser assistido no cinema.

É com base nesta norma da persuasão que se impõem milhares escalas de “melhor”, como: os melhores filmes, os melhores livros, e etc. O uso do termo “melhor” e afirmações proporcionam às pessoas um “atalho mental” na tomada de decisões, pois se confia na decisão dos outros, optando-a como “corretamente”, ao invés de gastar recursos mentais.

Os problemas podem surgir se o comportamento da maioria não for o mais correto ou se os dados que se recebe tiverem sido manipulados. Nesses casos, deve existir a confiança na validade social, que, por sua vez, pode ser um erro no ato de não se revelar a melhor ferramenta para se inferir a forma correta de proceder.

A melhor maneira de não se deixar influenciar por estratégias que recorrem a esta norma é tentar encontrar explicações alternativas para o comportamento dos outros. Por exemplo, é possível que um livro seja o mais vendido apenas porque foi feita uma promoção e não por ser o “melhor”.

Como base nesse exemplo e conforme já mencionado, o principal problema desse princípio é quando surgem dados incorretos, uma vez que, a melhor defesa contra estas desvantagens é reconhecer quando os dados estão errados. Deve-se criar, então, uma sensibilidade à prova social, de modo que, quando se deparar com

informações imprecisas, possa ser capaz de compreender a invalidade de tais dados, permitindo, assim, rejeitá-los (CIALDINI, 1998).

3.1.5. Escassez

O princípio da escassez funciona de forma poderosa sobre o montante que é atribuído às coisas; é natural que um determinado produto ou coisa tenha um “valor agregado” por motivos de escasses. Provavelmente, o uso mais simples do princípio da escassez ocorre no chamado “número limitado”, que é uma tática utilizada para informar aos clientes de que um determinado produto está em falta e de que não vai durar muito tempo no mercado ou na prateleira. Às vezes, a informação “número limitado” pode ser verdade, mas em todo o caso, a intenção é convencer os clientes da escassez de um item e, assim, aumentar o seu valor imediato perante os olhos dos clientes (CIALDINI, 1998).

De acordo com o princípio da escassez, as pessoas são convidadas a se comprometer a comprar o “aparelho” quando o mesmo está menos disponível e, portanto, mais desejável. Muitas pessoas não controlam uma compra, e, neste momento, encontram-se singularmente vulneráveis.

Este é um dos princípios mais fáceis de ser percebido: quanto mais raro for um objeto, maior é o desejo de possuí-lo. Em termos econômicos, um exemplo deste princípio, é a lei da oferta e da procura. É normal sentir impulso de comprar um determinado artigo por este encontrar-se em promoção ou ser uma “edição limitada”. Assim que um produto entra num período de “escassez”, o interesse dos consumidores aumenta.

É o fato de estar a perder algo potencialmente valioso que leva a atuar de acordo com esta norma de persuasão. Quando se sabe que algo existe em um número limitado, a perspectiva de estar a perder um bom negócio ou de não comprar um produto único torna esse objeto mais desejável. É isso que faz muitas vezes pagar mais por um produto, sabendo-se, na realidade, que o mesmo vale menos. Outra forma é saber que as “coisas” difíceis de possuir são tipicamente melhor do que aquelas de fácil aquisição.

Além disso, existe uma fonte única e secundária de poder dentro do princípio da escassez: a oportunidade de se tornar menos disponível proporciona a perda da liberdade de aquisição, e as pessoas não gostam de perder aquilo que antes poderiam ter. Este desejo preserva as prerrogativas estabelecidas, onde a peça central de suporte psicológico é a Teoria da Reatância, desenvolvida pelo psicólogo Jack Brehm para explicar a resposta humana que diminui o controle pessoal. De acordo com o psicólogo, sempre que a livre escolha é limitada ou ameaçada, a necessidade de manter a liberdade faz desejá-las (bem como os bens e serviços a eles associados) significativamente mais do que anteriormente. Assim, quando há o aumento de escassez ou de qualquer outra coisa que interfere no acesso prévio de algum "item", ocorre uma reação contra a interferência do querer ou do tentar possuir o item mais do que antes da interferência (BREHM,1966).

É bastante fácil sentir-se devidamente advertido contra pressões da escassez, mas é muito mais difícil agir com base nessa advertência. Parte do problema é que a reação típica à escassez dificulta a capacidade de pensar. Quando ver-se algo que se deseja tornar-se menos disponível, ocorre uma agitação física em conjunto, especialmente nos casos que envolvam concorrência direta (CIALDINI, 1998).

Para evitar essa norma, deve-se procurar avaliar o produto não à luz da sua escassez, mas sim de acordo com os seus méritos. Deve-se, portanto, tentar perceber se o valor que está sendo exigido é o justo a se pagar.

3.1.6. Empatia

Não seria nenhuma novidade sentir-se mais predisposto a fazer favores às pessoas de que se gosta mais: namorado(a), esposo(a), pais, irmãos, familiares próximos e amigos. Ser persuadido por alguma destas pessoas é algo natural. Menos óbvio, mas aceitável, é o fato de ser mais facilmente persuadido por pessoas de que se gosta. Todos precisam sentir-se queridos e gostados, e, quando alguém demonstra gostar do outro, estar-se mais disponível para fazer aquilo que pedem (CIALDINI, 1998).

Poucas pessoas ficariam surpresas ao saber que mais preferem dizer "sim" aos pedidos de alguém querido ou conhecido do que aqueles advindos de pessoas

desconhecidas e de que não se gosta. O que pode ser surpreendente, entretanto, é que esta simples regra é usada de várias maneiras, por pessoas totalmente estranhas, a fim de que possa satisfazer os pedidos.

Mas, o princípio da empatia na persuasão vai além do que simples relações humanas. Existem outras pequenas coisas que fazem este princípio atuar. Muitas vezes são óbvias, mas normalmente passam despercebidas, quer pela sua sutileza, ou pela tendência em minimizar o seu impacto nas ações. Como exemplo, a empatia física de uma pessoa proporciona um aumento ao potencial de persuasão. Apesar dos seres humanos serem racionais e socialmente evoluídos, não deixam de estar expostos a leis básicas da natureza.

A atratividade física atua em termos de avaliação da personalidade de uma pessoa. As pessoas mais atraentes são percebidas como sendo mais inteligentes, capazes, honestas e confiáveis. Isto se deve ao chamado efeito de halo⁷ (ROSENZWEIG, 2007), em que as características positivas ou negativas de um dado traço são transpostas para outros sem qualquer tipo de ligação aparente. Mas, a atratividade física é uma variável que só pode ser manipulada até certo ponto, porque se trata de uma avaliação bastante subjetiva, apesar de haver uma grande influência sócio-cultural.

No entanto, existe uma grande força da empatia quando se trata de seres humanos com características semelhantes. Quando as pessoas partilham os interesses, os gostos, as crenças e as atitudes iguais, torna-se mais favorável a prática da empatia. Coisas simples, como exemplo, local de nascimento, equipe de futebol preferida, nome, ou modo de vestir-se e etc, todos estes fatores influenciam positivamente na forma de reação aos pedidos feitos.

Outra estratégia da empatia é o elogio, quando simples palavras de agrado são lançadas, cria-se uma maior empatia. Em outras palavras, termos de bajulação ou simples alegações de afinidades criam uma avaliação positiva. De fato, procura-se beneficiar-se da “regra da amizade”, que irá dar suporte a atuação do princípio.

⁷ Capacidade de julgamento errônea das pessoas através da aparência física.

Ao invés de tentar reconhecer e prevenir a ação da empatia, antes, o ideal é criar uma chance de observar o objetivo final de tal estratégia. Isto é, a vigilância deve ser dirigida não para os motivos que podem produzir a empatia indevida, mas sim para as consequências provocadas por essa empatia indevida. Ao concentrar a atenção nos efeitos e não nas causas, pode-se evitar a tarefa trabalhosa, ou quase impossível de tentar detectar e desviar as influências psicológicas da empatia. Ao se perceber este princípio, será mais fácil tomar as medidas necessárias para combatê-lo. (CIALDINI, 1998).

Atenta-se que não se deve preocupar-se em conter a influência dos fatores da empatia. Mas, muito pelo contrário, deve-se permitir que esses fatores exercem sua força, para, então, utilizá-la de forma favorável (CIALDINI, 1998). Em todo o caso, a melhor forma de defesa do uso destas estratégias é tentar distinguir o pedido da pessoa que o faz, no sentido de poder avaliá-lo de uma forma mais objetiva e coerente.

3.2. Conclusão do Capítulo

No Capítulo 3, destrinchou-se detalhadamente o funcionamento de cada um dos Princípios bordados pela Teoria da Persuasão no meio social, constatando, através de suas funcionalidades, que a persuasão criará maiores possibilidades de atingir o objetivo desejado pelo prosador. Pois, foi demonstrado, em suma, que a persuasão atua na sociedade de forma quase que imperceptível.

4 SISTEMAS DE TOMADA DE DECISÃO

Inicialmente, torna-se necessário conceituar os elementos que conduzem as pessoas nas suas decisões. Segundo Batista (2004), do ponto de vista da administração de empresas em concordância com a definição de sistemas, existem dois elementos fundamentais para a tomada de decisões: os canais de informação e as redes de comunicação. Através dos canais de informação, as pessoas definem de onde serão adquiridos os dados, e as redes de comunicação definem para onde os dados serão direcionados.

Para a formação dos sistemas e a conseqüente obtenção dos elementos fundamentais para a tomada de decisão, é necessário o conhecimento dos conceitos de Dados, Informação e Conhecimento.

Para Davenport e Prusak (1999), é essencial para a realização bem-sucedida dos trabalhos ligados ao conhecimento que as pessoas saibam definir o que são dados, informações e conhecimento, pois o sucesso ou o fracasso organizacional muitas vezes pode depender da aplicação desses elementos para solução de problemas e tomada de decisões.

Os dados apresentam-se como elementos em sua forma bruta, os quais não podem por si só sustentar a estruturação necessária para tomada de ação. Os dados precisam passar por análise e transformações para se tornarem úteis. Oliveira (2002) cita que dado é qualquer elemento identificado em sua forma bruta que, por si só, não conduz a uma compreensão de determinado fato ou situação. Assim, para a compreensão de determinado fato ou situação em uma organização é necessário que os dados se transformem em informação.

Padoveze (2000) *apud* Nakagawa evidencia que informação é o dado que foi processado e armazenado de forma compreensível para seu receptor e que apresenta valor real percebido para suas decisões correntes ou prospectivas.

Segundo Oliveira (1992), a informação auxilia no processo decisório, pois, quando devidamente estruturada, é de crucial importância para as empresas, além disso, associa os diversos subsistemas e capacita a empresa a impetrar seus objetivos.

O valor atribuído pelos gestores às informações depende dos resultados alcançados pela empresa. Os benefícios oferecidos pelas decisões acertadas, baseadas em informações valiosas, representam o sucesso da empresa.

O conceito de valor da informação, segundo Padoveze (2000), está relacionado com:

- A redução da incerteza no processo de tomada de decisão;
- A relação do benefício gerado pela informação versus custo de produzi-la;
- Aumento da qualidade da decisão.

Para medir o valor da informação, o gestor deve dispor da informação de forma que ela reduza as incertezas encontradas no decorrer do processo de decisão, e conseqüentemente, aumente a qualidade da decisão.

Segundo Stair (1998), o conjunto de dados, as regras, os procedimentos e relações que devem ser seguidos para se atingir o valor informacional ou resultado adequado do processo estão contidos na base do conhecimento.

A base do conhecimento facilita reconhecer quais dados e informações são úteis para se atingir os objetivos traçados pela organização.

Para Laudon e Laudon (1999), conhecimento é o conjunto de ferramentas conceituais e categorias usadas pelos seres humanos para criar, colecionar, armazenar e compartilhar a informação.

As informações são criadas a partir da transformação dos dados, através da aplicação do conhecimento humano.

4.1. Tecnologia de Informação

O ambiente empresarial está mudando continuamente, tornando-se mais complexo e menos previsível, e cada vez mais dependente de informações e de toda a infraestrutura tecnológica que permite o gerenciamento de enormes quantidades de dados. Para Pereira e Fonseca (1997), a tecnologia da informação surgiu da necessidade de se estabelecer estratégias e instrumentos de captação, organização, interpretação e uso das informações.

As informações com qualidade e apresentadas em tempo hábil à tomada de decisão são de vital importância para as empresas modernas. O uso adequado dos

recursos da Tecnologia de Informação garante a qualidade e pontualidade das informações. Foina (2001) conceitua Tecnologia da Informação como um conjunto de métodos e ferramentas, mecanizadas ou não, que se propõe a garantir a qualidade e pontualidade das informações dentro da malha empresarial.

Batista (2004) define Tecnologia de Informação como todo e qualquer dispositivo que tenha a capacidade para tratar dados e/ou informações, tanto de forma sistêmica como esporádica, independentemente da maneira como é aplicada.

A gestão estratégica das informações, resultante da Tecnologia da Informação, é parte integrante de qualquer estrutura gerencial de sucesso.

4.2. Componentes da Tecnologia de Informação

O sucesso das empresas atualmente está totalmente vinculado à velocidade em que as informações são assimiladas e pela rapidez em que são tomadas as decisões. Os componentes que fundamentam a Tecnologia de Informação são os grandes precursores desse sucesso.

Segundo Rezende e Abreu (2000), a Tecnologia de Informação está fundamentada nos seguintes componentes:

- Hardware e seus dispositivos e periféricos;
- Software e seus recursos;
- Sistemas de telecomunicações;
- Gestão de dados e informações.

A união desses componentes eleva a potencialidade de atuação das empresas, agregando valor de mercado e capacidade de gerir as informações de forma eficiente.

4.3. Sistemas de Informação

Os sistemas de informação têm por objetivo gerar informações para a tomada de decisões. Os dados são coletados, processados e transformados em informação. Stair (1998) afirma que sistemas de informação são uma série de elementos ou componentes inter-relacionados que coletam (entrada), manipulam e armazenam (processo), disseminam (saída) os dados e informações e fornecem um mecanismo de *feedback*.

Gil (1999) define que os sistemas de informação compreendem um conjunto de recursos humanos, materiais, tecnológicos e financeiros agregados segundo uma seqüência lógica para o processamento dos dados e a correspondente tradução em informações.

Na visão de Pereira e Fonseca (1997), os sistemas de informação (*management information systms*) são mecanismos de apoio à gestão, desenvolvidos com base na tecnologia de informação e com suporte da informática para atuar como condutores das informações que visam facilitar, agilizar e otimizar o processo decisório nas organizações.

A gestão empresarial precisa cada dia mais do apoio de sistemas, pois estes dão segurança, agilidade e versatilidade para a empresa no momento em que se processam as decisões.

As empresas precisam estar preparadas para lidar com os problemas internos e externos do ambiente em que estão inseridas, para tanto, buscam no desenvolvimento de sistemas de informações suporte para a resolução desses problemas. Laudon e Laudon (1999), afirmam que a razão mais forte pelas quais as empresas constroem os sistemas é para resolver problemas organizacionais e para reagir a uma mudança no ambiente.

Os sistemas de informação objetivam a resolução de problemas organizacionais internos, e a conseqüente preparação para enfrentar as tendências da crescente competitividade de mercado.

Para Pereira e Fonseca (1997), os sistemas de informação têm por finalidade a captura e/ou a recuperação de dados e sua análise em função de um processo de decisão. Envolvem, de modo geral, o decisor, o contexto, o objetivo da decisão e a estrutura de apresentação das informações.

De forma estruturada, os sistemas de informação dão condições para que as empresas reajam às mutações do mercado e se sintam alicerçadas por um processo decisório forte o suficiente para garantir a resolução dos problemas.

Com o Sistema de Informação estruturado, com a apresentação das informações necessárias, também já propiciando uma visão das decisões, a empresa

garante um grande diferencial em relação aos concorrentes, e os gestores podem tomar decisões mais rápidas e de fontes seguras.

Para serem efetivos, os sistemas de informação precisam, segundo Pereira e Fonseca (1997), corresponder às seguintes expectativas:

- Atender as reais necessidades dos usuários;
- Estar centrados no usuário (cliente) e não no profissional que o criou;
- Atender ao usuário com presteza;
- Apresentar custos compatíveis;
- Adaptar-se constantemente às novas tecnologias de informação;
- Estar alinhados com as estratégias de negócios da empresa.

Ao visualizar um sistema que atenda os requisitos acima, a empresa sente-se confiante no momento de utilizá-lo no processo decisório de seus negócios.

4.4. Classificação de Sistema de Informação

Os sistemas, do ponto de vista empresarial, podem ser classificados de acordo com a sua forma de utilização e o tipo de retorno dado ao processo de tomada de decisões. Os sistemas podem ser de contexto operacional ou gerencial, ou seja, Sistemas de Apoio às Operações e Sistema de Apoio Gerencial. A seguir serão apresentadas algumas classificações dos sistemas.

4.4.1. Sistema de Apoio às Operações

Os sistemas de Apoio às Operações de uma empresa têm por principais metas processar transações, controlar processos industriais e atualizar banco de dados, fornecendo informações de âmbito interno e externo. Apesar da sua importância para o desenvolvimento normal das atividades da empresa, não consegue desenvolver informações específicas, necessitando do apoio do sistema de informação gerencial. Faz parte do Sistema de Apoio às Operações.

4.4.1.1. Sistema de Processamento de Transações (SPT)

Os sistemas de processamento de transações são utilizados no nível operacional da empresa. Afirmam Laudon e Laudon (2001), que um sistema de

processamento de transações é um sistema computadorizado que executa e registra as transações rotineiras diárias necessárias para a condução dos negócios.

A automatização dos trabalhos repetitivos e rotineiros comuns aos negócios da empresa agiliza e facilita a realização dos trabalhos, além de oferecer uma gama maior de informações. Como exemplo, pode-se citar a transação das rotinas da folha de pagamento, a computadorização, além de produzir os cheques para pagamento dos colaboradores, pode-se também fornecer relatórios exigidos pelos órgãos federais e estaduais. São exemplos de SPT, a emissão de notas fiscais e o controle de estoque.

4.4.1.2. Sistemas de Trabalho do Conhecimento e de Automação de Escritório (STC e SAE)

A necessidade do nível de conhecimento da empresa é suprida pelos sistemas de trabalho do conhecimento e de automação de escritório. Segundo Batista (2004), a definição que se aplica ao STC e SAE é descrita da seguinte forma: toda e qualquer tecnologia de informação que possui como objetivo principal aumentar a produtividade pessoal dos trabalhadores manipulam as informações de escritório.

Laudon e Laudon (2001) também definem que os sistemas de automação de escritório (SAE) são aplicações de informática projetadas para aumentar a produtividade dos trabalhadores de dados, dando suporte à coordenação e às atividades de comunicação de um escritório típico.

Os aplicativos dos escritórios são projetados com base na necessidade de manipulação e gerenciamento de documentos, aumentando assim a produtividade dos envolvidos com a atividade, por exemplo: a editoração eletrônica, arquivamento digital, planilhas de cálculo e outro que favorecem a qualidade e agilidade das tarefas.

Os sistemas de trabalho do conhecimento exigem uma visão ampla das pessoas, pois, além de saber usar os aplicativos dos escritórios, essas pessoas precisam saber utilizar o que o aplicativo oferece para criar informações novas.

4.4.2. Sistema de Apoio Gerencial

Quando se fala em fornecer informações para a tomada de decisão, toda a empresa deve estar envolvida nesse processo. A complexa relação entre os diversos gerentes de uma organização deve ser facilitada pelos sistemas de apoio gerencial.

O'Brien (2002) afirma que, quando os sistemas de informação se concentram em fornecer informação e apoio à tomada de decisão eficaz pelos gerentes, são chamados sistemas de apoio gerencial. Entre os vários tipos de sistemas de apoio gerencial, podem-se citar: Sistema de Suporte da Decisão (SSD), Sistema de Suporte Executivo (SSE) e Sistema de Informação Gerencial (SIG).

4.4.2.1. Sistema de Suporte da decisão (SSD)

Os sistemas de suporte da decisão são munidos de grande quantidade de dados e ferramentas de modelagem, permitindo uma flexibilidade, adaptabilidade e capacidade de resposta rápida ao nível gerencial da organização.

Os sistemas de suporte à decisão oferecem recursos cruciais que viabilizam o suporte às decisões de nível gerencial.

4.4.2.2. Sistema de Suporte Executivo (SSE)

Os sistemas de suporte executivo dão suporte ao nível estratégico da empresa e ajudam a definir os objetivos a serem estabelecidos, utilizando-se de tecnologia avançada para a elaboração de gráficos e relatórios. Os usuários desse sistema são os executivos seniores.

Os sistemas de suporte executivo não são projetados para resolver problemas específicos, em vez disso, fornecem uma capacidade de computação e telecomunicações que pode mudar a estrutura dos problemas.

4.4.2.3. Sistema de Informação Gerencial (SIG)

O sistema de informação gerencial dá suporte às funções de planejamento, controle e organização de uma empresa, fornecendo informações seguras e em tempo hábil para tomada de decisão. Oliveira (2002) define que o sistema de informação

gerencial é representado pelo conjunto de subsistemas, visualizados de forma integrada e capaz de gerar informações necessárias ao processo decisório.

Stair (1998), assim, define o propósito básico de um SIG que é ajudar a empresa a alcançar suas metas, fornecendo a seus gerentes detalhes sobre as operações regulares da organização, de forma que possam controlar, organizar e planejar com mais efetividade e com maior eficiência.

4.5. Condições de Tomada de Decisão

As decisões são tomadas sob diversas condições, sob condições de certeza, condições de incerteza e condições de risco. As decisões programadas normalmente oferecem um grau de risco menor do que as decisões não-programadas.

Em se tratando das decisões tomadas sob certeza, o decisor tem conhecimento das conseqüências ou resultados de todas as alternativas, sendo assim, pode escolher a melhor dentre as alternativas propostas. Stephen e Coulter (1996) enfatizam que a situação ideal para a tomada de decisões é a de certeza, ou seja, o administrador pode tomar decisões precisas, pois o resultado de cada alternativa é conhecido.

Com relação às decisões tomadas sob condições de incertezas, os resultados são desconhecidos e gerados sob probabilidades, onde o decisor tem pouco ou nenhum conhecimento das informações que formam as alternativas.

Na tomada de decisão com risco, todas as alternativas têm um resultado específico e são projetadas sob probabilidades conhecidas. O tomador de decisão conhece todas as alternativas e sabe que o risco é inevitável.

Diante do exposto, observa-se que a tomada de decisão está diretamente relacionada ao potencial informativo do Sistema de Informação da empresa, e este dever ser o mais útil possível na geração da melhor informação no auxílio ao gestor.

4.6. Conclusão do Capítulo

O Capítulo 4 explanou a importância de compreender as melhores práticas e aplicações para a tomada de decisão das áreas de sistemas de informação e tecnologia da informação nas empresas. Uma vez que o propósito básico da informação é o de

habilitar a empresa a alcançar seus objetivos pelo uso eficiente dos recursos disponíveis, portanto, as informações podem decidir o futuro da organização.

A utilização da tecnologia na tomada de decisão possibilita a criação de aplicações capazes de atender desde pequenos a grandes volumes de utilização. Ou seja, as empresas têm como grande aliado os sistemas de informação gerencial, os quais proporcionam benefícios significativos na gestão da empresa viabilizando a geração de relatórios de apoio ao processo decisório.

Os sistemas de informação gerenciais fortalecem o plano de atuação das empresas. A geração de informações rápidas, precisas e principalmente são úteis para o processo de tomada de decisão, pois garantem uma estruturação de gestão diferenciada, resultando em vantagem competitiva sobre as demais empresas.

5 UM MODELO DE TOMADA DE DECISÃO BASEADO NA TEORIA DA PERSUASÃO

Como já foi mencionado no início do capítulo 3, após uma breve explanação dos seis princípios da Persuasão, o modelo matemático denominado Rede de Petri (*APÊNDICE E – Rede de Petri*) e os sistemas de tomada de decisão (capítulo 4), pode-se dar continuidade à aplicação dos princípios modelados ao MMORPG.

5.1. Reciprocidade aplicada ao MMORPG

No ambiente, existe inúmeras possibilidades de se aplicar a reciprocidade. Tal fator é possível, pois o jogo aborda uma criação de um mundo real unido a um surreal. A seguir, serão citadas as principais ferramentas para se utilizar as estratégias da reciprocidade:

- Uma simples ajuda de caça para que possa subir de nível ou encontrar algum item ou completar alguma *Quest*. A pessoa que receber a ajuda ficaria em dívida com o ajudante, o que permite a troca do conhecimento;
- Dicas ou conselhos recebidos nos fóruns. Da mesma forma que a anterior, o ajudado ficaria em dívida com as postagens no fórum e para com o ajudante;
- Descontos em vendas de algum item. Existe também a possibilidade de que tal desconto não exista, bastando apenas anunciar “desconto” para que o item seja vendido. Isso ocorre no caso de que alguns itens necessitem de um “empurrão” para que possa ser vendido, ou pode ser usado para compra acima de X valor terá o desconto N . O mesmo vale para promoções e concursos;
- Para compradores ativos, dão-se descontos, criando-se um vínculo de fidelidade.

5.1.1. Modulo Reciprocidade fluxo e o princípio modelado na rede Petri

A Reciprocidade atua como uma troca de favores, ou seja, consiste em compartilhar aquilo que foi recebido.

No ambiente do MMORPG o princípio pode ser ativo de várias maneiras, já mencionadas na seção 5.1. Para facilitar uma melhor compreensão deste princípio modelado ao ambiente, foi criado um fluxo do seu funcionamento no ambiente, como pode ser visualizado na Figura 2.

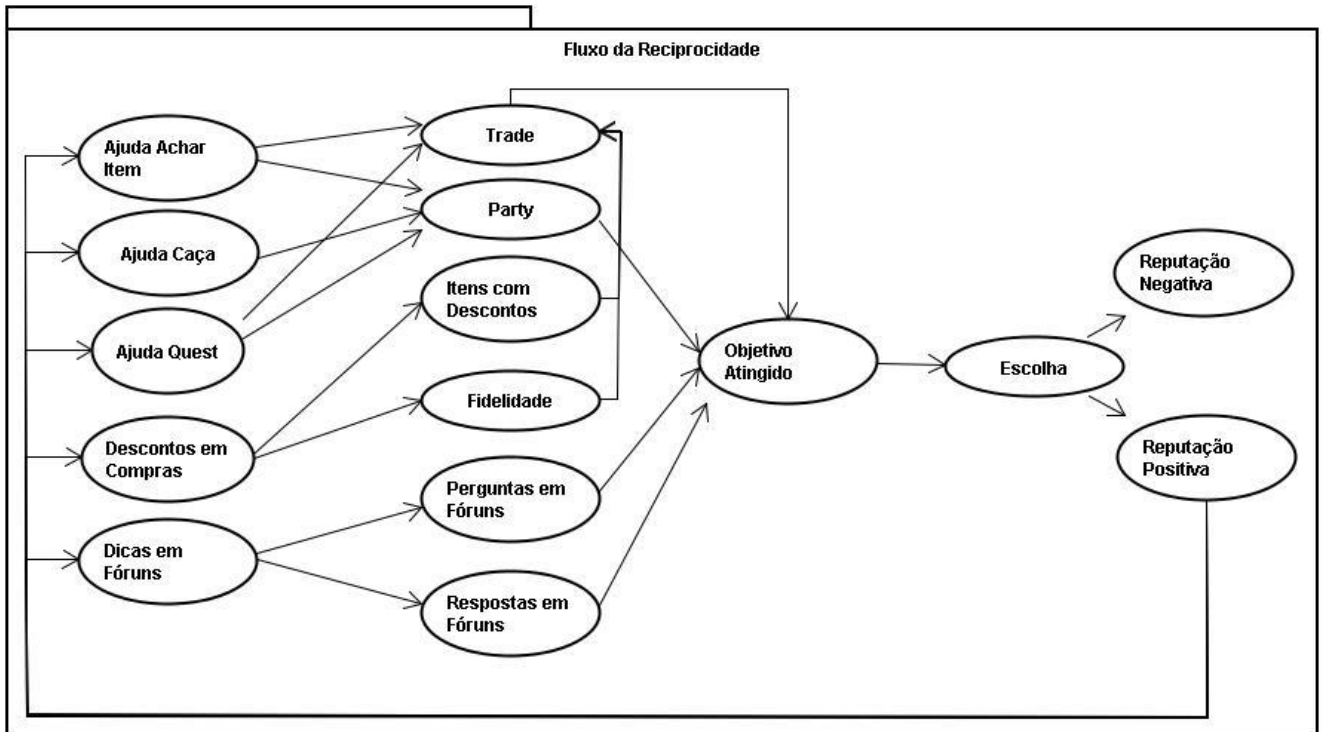


Figura 2: Fluxo da Reciprocidade.

A reciprocidade ativa-se quando o jogador solicita a outro simples ajudas, tais como encontrar algum item, completar alguma *Quest* e ajudar a caçar. Para estas solicitações, os jogadores utilizam ao seu favor ferramentas como o *Trade* e a *Party*, como pode ser visto no fluxo acima demonstrado. O início deste fluxo dar-se através do desejo de cada jogador, que solicita a reciprocidade, tais como: *Ajuda Achar Item*, *Ajuda Caça* e *Ajuda Quest*. Caso ajudar for achar um item e o outro jogar tenha este determinado objeto, o mesmo pode ser transferido via *Trade*. Caso contrário, cria-se a *Party* para ir à procura deste item ou para ir à caça ou, ainda, para completar alguma *Quest*. A *Party* possibilita que os jogadores dividam o dinheiro e as experiências da busca ou da missão.

Em relação a *Descontos em Compras*, trata-se do caso em que o jogador utiliza este princípio, a fim de que possa atingir descontos por fidelidade ou na compra de uma

grande quantia de objetos em negociação com outro jogador. Então, utiliza-se a ferramenta de transferência de objeto chamada *Trade*.

A *Dicas em Fóruns* dar-se quando o jogador acessa a plataforma do ambiente virtual, e realiza suas perguntas ou pesquisas. Após atingida a meta, os pesquisadores aprovam as respostas de outros membros, a fim de que possam facilitar novas buscas e motivar outros jogadores a melhorar suas reputações.

Ao utilizar todas as opções de ferramentas que concretizam este princípio, chega-se ao *Objetivo Atingido*. Isso ocorre por que o jogador conseguiu obter a sua meta ou missão. Ao final da meta, o jogador tem a escolha passiva de retribuir o princípio, criando um movimento cíclico da Reciprocidade, ou tem a escolha de não retribuí-lo, quebrando o ciclo do princípio, estando o jogador ciente das penalidades que na seção 3.1.1 foram mencionadas.

No APÊNDICE D – *Princípios da Persuasão aplicados ao MMORPG utilizando Redes de Petri*, encontra-se a Figura 55, que aborda a Reciprocidade modelada utilizando a Rede de Petri e permite visualizar os lugares (estados), as transições e o conjunto de arcos orientados.

Nesta rede, existe marcação inicial, para a definição formal da Rede de Petri, este caso envolve todas as variáveis *m1* de todos os princípios, no entanto, mesmo com a marcação inicial da rede permite-se criar um movimento cíclico da rede em todos os outros Princípios da Persuasão. Este movimento, por sua vez, apenas é quebrado quando o jogador ativo da Persuasão não desejar retribuir aquilo que lhe foi dado. Além disso, nesta mesma rede, existem também os lugares finais, também chamados estados finais, que são utilizados para identificar a quantidade de pessoas que completam o ciclo da Reciprocidade e as que não completam.

Para a rede da Reciprocidade, podem-se declarar os seguintes lugares e suas variáveis, tais como:

- Variável *m1* - *Início da Reciprocidade*: este estado aborda quando o jogador utiliza a Reciprocidade para obter algo de seu desejo, solicitando a transição *Solicitar Suporte*. O jogador apenas poderá prosseguir quando aquele solicitado estiver disponível;

- Variável *m2 - Solicitado Disponível*: é o estado onde o jogador inicial, que ativa a Reciprocidade, aguarda a disponibilidade de outro jogador, verificando se este está disponível ou não. Caso não tenha jogador disponível, passa-se para a transição *Solicitador não Disponível*. Caso exista um jogador disponível, passa-se para a transição *Solicitar Suporte*;
- Variável *m3 - Espera Pedido Suporte*: Ocorre quando o jogador ativo da reciprocidade está a escolher o desejo que irá informar ao jogador passivo à Persuasão. Neste estado, a Rede de Petri encontra-se em redundância de caminhos, uma vez que esta foi desenvolvida para modelar sistemas paralelos, concorrentes, assíncronos e não-determinísticos. Este fato justifica-se por que a Rede de Petri foi modelada com a atuação humana. Desse modo, a escolha do caminho a seguir irá depender apenas do jogador ativo, que poderá ser *Comprar Item*, *Pedir Ajuda* ou *Perguntar Fórum*;
- Variável *m5 - Espera Desconto*: o jogador ativo utiliza a vantagem do desconto de itens por meio de diversos fatores, tais como: cliente ativo, compra em grande quantia e etc. O jogador passivo garantirá a venda de seu produto através da troca de favores. Passa-se, então, para a transição *Item Descontar*;
- Variável *m4 - Espera do Pedido*: o jogador passivo só poderá prosseguir com o desconto, caso o item seja solicitado. Na hipótese de isso não ocorrer, passa-se para a transição *Pedido não Realizado*. Feito o pedido, o item prossegue para a transição *Item Descontar*, escolhendo o valor a ser negociado;
- Variável *m8 - Fiel a Compra*: ocorre quando se fecha o valor do pedido, criando uma fidelidade na compra. Pode-se prosseguir para *Solicitar Trade* ou *Compra não Efetuar*;
- Variável *m11 - Reputação Negativa ao Solicitante*: pode-se dizer que se trata de um dos estados finais que existem na rede. Este estado ocorre quando o jogador ativo da Reciprocidade deseja não prosseguir com a compra, criando assim um vínculo de não fidelidade. Nesse caso, o jogador quebra o ciclo da Reciprocidade e é penalizado com perdas de pontos em sua reputação, o

que não é o ideal para o presente trabalho, mas pelos motivos anteriormente mencionados existirá esta possibilidade;

- Variável *m9 - Espera Item Disponível*: como o nome já diz, trata-se da disponibilidade do item do jogador passivo, que pode negociar mesmo quando não tem o item. Nesse caso, passa-se para a transição *Item não Disponível*, que se mantém até que o vendedor tome posse do item, momento em que se segue para a transição *Solicitar Trade*;
- Variável *m10 – Trade*: neste estado, ocorre a real venda do produto. Após toda a fase de negociação e busca do item, passa-se para a transição *Confirmar Trade*, checando valores e itens de compra;
- Variável *m12 - Compra Realizada*: é onde ocorre todo o processo e finalização da compra, permitindo passar para a transição *Pontuar Reputação*;
- Variável *m13 - Reputação Pontuada por Venda*: é o estado final que permite analisar quantas vendas foram feitas utilizando este princípio, permitindo também uma maior reputação ao vendedor caso ocorram vendas futuras;
- Variável *m14 - Reputação Pontuada por Compra*: semelhante ao estado anterior, porém, em relação ao comprador;
- Variável *m25 - Objetivo da Reciprocidade Atingido*: como o próprio nome diz, a Reciprocidade não ocorre apenas quando se cria um ciclo, que no caso seria o ideal, mas quando o vendedor ou o ajudante ou quem respondeu a pergunta no fórum usou a Reciprocidade, pode esta funcionar ou não. Isso porque é este o objetivo do trabalho, qual seja, utilizar os Princípios da Persuasão. Em seguida, existe a transição *Ativando a Reciprocidade*, que aborda a escolha do jogador ativo em retribuir ou não;
- Variável *m26 - Espera Escolha do Solicitante*: este estado aguarda a ação do jogador para com as transições *Escolher não Retribuir* ou *Escolher Retribuir*;
- Variável *m27 - Reputação Negativa ao Solicitante*: é o estado final da rede e ocorre quando o jogador ativo não escolhe retribuir a Reciprocidade, pontuado, assim, negativamente a sua reputação;

- Variável *m28* - *Reputação ao Solicitante*: na transição *Escolher Retribuir*, existem duas rotas em paralelo a seguir. Uma das rotas é a *Reputação ao Solicitante*, em que o jogador ativo é pontuado positivamente em sua reputação, favorecendo-o quando desejar ativar novamente a *Persuasão*. A outra rota vai para o estado *Espera para Completar o Clico da Reciprocidade*, a seguir explicado;
- Variável *m29* - *Espera para Completar o Clico da Reciprocidade*: trata-se apenas de um estado que aguarda que alguém o ative, que, no caso, é o jogador passivo ou o ativo, seguindo, então, para a transição *Completa Ciclo da Reciprocidade*;
- Variável *m30* - *Espera do Início da Reciprocidade*: é o estado que indica quando a reciprocidade irá iniciar, seguindo para a transição *Ativando a Reciprocidade*. É nesse momento que se inicia realmente a rede, passando para os estados *m1* e *m2*;
- Variável *m6* - *Ajuda em Espera*: ocorre quando o jogador ativo escolhe a transição *Pedir Ajuda*, podendo seguir para a transição *Solicita Party* na hipótese de existir um jogador na pretensão de ajudá-lo;
- Variável *m15* - *Espera Party Disponível*: ocorre da mesma forma que o estado *m4*, porém, diferenciando-se na necessidade de verificar se a ajuda para *Party* estará disponível ou não. Caso esteja indisponível, irá para a transição *Party não Disponibilizada*. Já, estando disponível, passa-se para a transição *Solicitar Party*;
- Variável *m16* – *Party*: ocorre quando os membros envolvidos a ajudar estão em grupo e utilizam a ferramenta *Party*. Dependendo da conduta, pode-se prosseguir até a transição *Ajuda em Andamento* ou *Party Cancelada*, que indica o cancelamento do grupo, haja vista o não comprometimento deste ou do jogador em ajudar;
- Variável *m17* - *Reputação Negativa ao Solicitado*: caso o grupo ou jogador passivo solicite o cancelamento da *Party*, a(s) sua(s) reputação(ões) serão pontuadas como negativas, como forma de penalidade;

- Variável *m18 - Ajuda Realizada*: trata-se do fim do processo de ajuda e ocorre quando se atinge o objetivo do grupo, prosseguindo-se para a transição *Pontuar Reputação*;
- Variável *m19 - Reputação Pontuada por Ajuda*: é semelhante ao estado *m13*, porém, a pontuação, nesse caso, ocorrerá pelo fato da ajuda;
- Variável *m20 - Reputação Pontuada ao Ajudado*: é semelhante ao estado *m14*, porém, esta pontuação ocorrerá pelo fato da ajuda. Prossegue-se para o estado *m25*, dando continuidade à rede, que já foi explicado anteriormente;
- Variável *m7 - Espera Resposta*: é o outro caminho a ser seguido na rede. Trata-se quando o jogador ativo solicita a transição *Pergunta Fórum*, aguardando até que algum jogador passivo possa respondê-lo, podendo ou não o jogador ativo *Confirmar a Resposta*. Caso contrário, o jogador passa para a transição *Resposta não Realizada*;
- Variável *m21 - Espera da Pergunta*: ocorre quando os jogadores passivos aguardam alguma pergunta para responder. Não havendo pergunta alguma, passa-se para transição *Pergunta não Realizada*. Já, na hipótese de ocorrer a resposta relacionada à pergunta feita, cabe ao jogador ativo *Confirmar a Resposta*;
- Variável *m22 - Dica Realizada*: este estado ocorre após o jogador *Confirmar a Resposta*, passando para a transição *Pontuar Reputação*;
- Variável *m23 - Reputação Pontuada por Dica*: semelhante ao estado *m13*. Mas, aqui, a maior reputação ocorre pelo fato de existir a participação do fórum em transmitir a dica;
- Variável *m24 - Reputação Pontuada por Pergunta*: semelhante ao estado *m14*, dando continuidade à reputação por ter realizado a pergunta. Em paralelo, irá para o estado *m25*, dando continuidade a toda rede;

No em tanto, existe outra variável que compõe a Rede de Petri, chamada de função peso. Esta função é responsável pelo disparo da rede, ocorra ou não.

A função peso é determinada pela estratégia de cada jogador, uma vez que, este jogador poderá ter várias estratégias, o armazenamento das estratégias será responsabilidade da classe *Conta_Class* contém o atributo *Estrategia*, que é do tipo

vetor, possibilitando a criação de um perfil do jogador (esta classe será melhor explicada na sessão 6.5).

Para o desenvolvimento desta estratégia foi utilizada umas das equações do âmbito da Teoria dos Jogos. Para um melhor entendimento desta equação torna-se necessário explanar a teoria até sua equação aplicando ao modelo.

A Teoria dos Jogos é uma teoria matemática criada para se modelar fenômenos que podem ser observados quando dois ou mais “agentes de decisão” interagem entre si. Ela fornece a linguagem para a descrição de processos de decisão conscientes e objetivos envolvendo mais do que um indivíduo (CONWAY, 1977) (CONWAY, 1978).

Seguindo o princípio da Teoria dos Jogos aplicado ao modelo visa-se o estudo da escolha de decisões ótimas sob condições de conflito, quando deseja ativa um dos Princípios da Persuasão. Onde elemento básico do jogo é o conjunto de jogadores que dele participam. Visto que cada jogador tem um conjunto de estratégias. Quando cada jogador escolhe sua estratégia, tem-se então uma situação ou perfil no espaço de todas as situações (perfis) possíveis. Cada jogador tem interesse ou preferências para cada situação no jogo. Em termos matemáticos, cada jogador tem uma função utilidade que atribui um número real (o ganho do jogador) a cada situação do jogo (SARTINI, 2004).

Mais especificamente, o jogo tem os seguintes elementos básicos: existe um conjunto finito de jogadores, representado por $G = \{g_1, g_2, \dots, g_n\}$. Cada jogador $g_i \in G$ possui um conjunto finito $S_i = \{s_{i1}, s_{i2}, \dots, s_{imi}\}$ de opções, denominadas estratégias puras do jogador $g_i (m_i \geq 2)$. Um vetor $s = (s_{1j_1}, s_{2j_2}, \dots, s_{n_j_n})$, onde s_{ij_i} é uma estratégia pura para o jogador $g_i \in G$, é denominado um perfil de estratégia pura. O conjunto de todos os perfis de estratégia pura formam, portanto, o produto cartesiano (SARTINI, 2004):

$$S = \prod_{i=1}^n S_i = S_1 \times S_2 \times \dots \times S_N,$$

denominado espaço de estratégia pura do jogo. Para jogador $g_i \in G$, existe uma função utilidade:

$$u_i : \begin{array}{l} S \rightarrow R \\ s \mapsto u_i(s) \end{array}$$

que associa o ganho $u_i(s)$ do jogador g_i a cada perfil de estratégia pura $s \in S$ (SARTINI, 2004).

Desta maneira, possibilita o jogador escolher qual a melhor estratégia para cada Princípio da Persuasão e seu disparo em cada momento do jogo.

5.2. Consistência aplicada ao MMORPG

No ambiente, existem inúmeras possibilidades de se aplicar a consistência. A seguir, serão citadas as principais ferramentas para se utilizar as estratégias da consistência:

- O comprometimento de comprar algum item raro, sabendo que foi muito difícil para o vendedor achá-lo. Nesse caso, o valor que será cobrado não seria um problema, mesmo que o preço prejudique o comprador. O vendedor utiliza o compromisso para realizar a venda, criando um vínculo de fidelidade;
- Existe um atributo ligado ao personagem que é chamado de Reputação. Com esse atributo, outros jogadores poderão checar a consistência do vendedor, quem postou o fórum e quem responde ao fórum;
- No ambiente do jogo, há a possibilidade da popularidade do personagem. Quando um personagem é consistente na sua ética ou comprometimento, outros personagens saberão a respeito de seu comportamento, positiva ou negativamente;
- Entre os membros pertencentes a uma mesma facção, existe um comprometimento maior em ajudar uns aos outros.

5.2.1. Modulo Consistência fluxo e o princípio modelado na rede Petri

Prosseguindo com a descrição de cada modelo, seguindo a estrutura da Rede de Petri da seção 5.1.1. Encontra-se a Consistência que nada mais é que manter-se consistente perante os princípios sociais, ou seja, tem-se uma exploração da pressão social para se manter consistente com as crenças, atitudes ou comportamentos.

No ambiente do MMORPG o princípio pode ser ativo de várias maneiras, já mencionadas na seção 5.2. Para facilitar uma melhor compreensão deste princípio

modelado ao ambiente, foi criado um fluxo do seu funcionamento no ambiente, como pode ser visualizado na Figura 3.

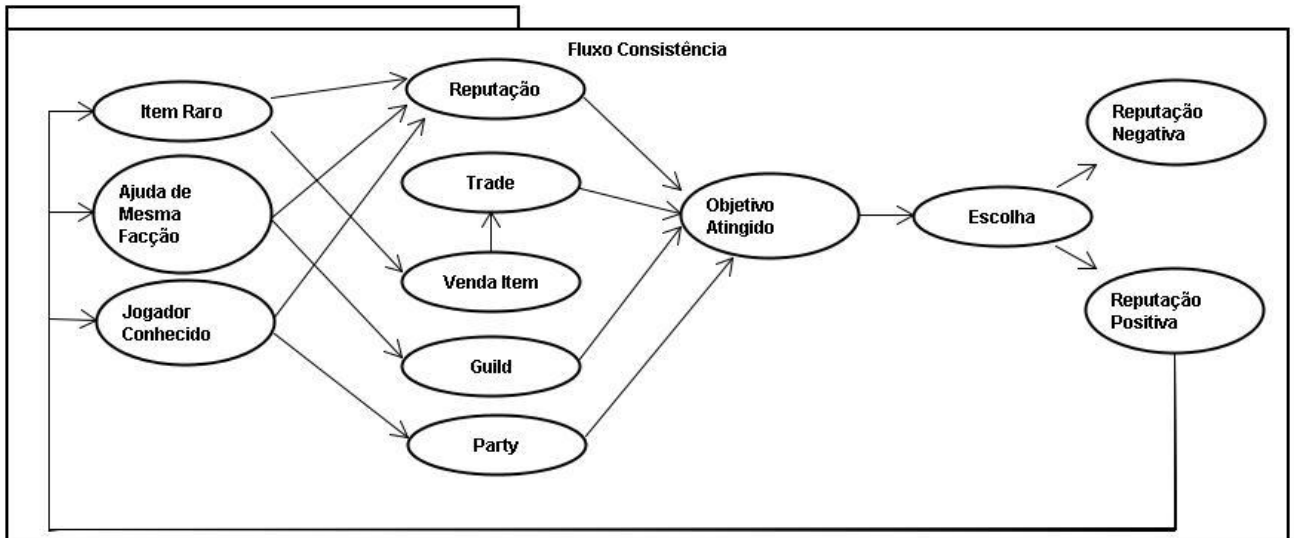


Figura 3: Fluxo da Consistência.

A Consistência inicia quando o jogador solicita a jogador simples ajuda ou solicita ajuda a jogadores de mesma facção, tais como encontrar algum item, completar alguma *Quest* e ajudar a caçar. Existe também outro caso que ocorre na Consistência é o caso da compra do *Item Raro*, mencionado na seção 5.2. Para estas solicitações, os jogadores utilizam ao seu favor ferramentas como o *Trade*, *Guild* e a *Party*, como pode ser visto no fluxo acima demonstrado. O início deste fluxo dar-se através do desejo de cada jogador, que solicita a Consistência, tais como: *Ajuda de Mesma Facção*, *Item Raro* e *Jogador Conhecido*. Este fluxo segue a estrutura do fluxo anterior.

Na possibilidade de *Jogador Conhecido* é o fato em que este jogador poderá ser um jogador conhecido (amigo) ou então um jogador de alta reputação, criando um perfil consistente.

No APÊNDICE D – *Princípios da Persuasão aplicados ao MMORPG utilizando Redes de Petri*, encontra-se a Figura 56, que aborda a Consistência modelada utilizando a Rede de Petri e permite visualizar os lugares (estados), as transições e o conjunto de arcos orientados.

Para a rede da Consistência, podem-se declarar os seguintes lugares e suas variáveis, tais como:

- Variável *m1 - Início da Consistência*: este estado aborda quando o jogador utiliza a Consistência para obter algo de seu desejo, solicitando a transição *Solicitar Suporte*. O jogador apenas poderá prosseguir quando aquele solicitado estiver disponível;
- Variável *m2 - Solicitado Disponível*: é o estado onde o jogador inicial, que ativa a Consistência, aguarda a disponibilidade de outro jogador, verificando se este está disponível ou não. Caso não tenha jogador disponível, passa-se para a transição *Solicitador não Disponível*. Caso exista um jogador disponível, passa-se para a transição *Solicitar Suporte*;
- Variável *m12 - Espera Pesquisa Reputação*: estado onde o jogador ativo do princípio irá verificar o perfil de consistência do jogador que irá ajudar. Existindo a possibilidade de seguir para a transição *Escolher Continuar*, caso o jogador seja consistente positivamente, ou *Escolher não Continuar*, caso este jogador disponível não seja consistente;
- Variável *m13 - Espera do Início da Consistência*: é o estado onde o jogador deseja ativar a Consistência, passando para a transição *Ativando Consistência*. É nesse momento que se inicia realmente a rede, passando para os estados *m1* e *m2*;
- Variável *m3 - Espera Pedido*: Ocorre quando o jogador ativo da consistência está a escolher o desejo que irá informar ao jogador passivo à Persuasão. Neste estado, o jogador poderá seguir para *Comprar Item Raro*, *Pedir Ajuda* ou *Perguntar Fórum*;
- Variável *m5 - Espera Item Raro*: o jogador passivo venderá o item raro ao jogador ativo, prosseguindo para a transição *Comprar*;
- Variável *m6 - Espera do Pedido*: o jogador passivo só poderá prosseguir, caso o item seja solicitado. Na hipótese de isso não ocorrer, passa-se para a transição *Pedido não Realizado*. Feito o pedido, o item prossegue para a transição *Comprar*, escolhendo o valor a ser negociado;
- Variável *m7 - Fiel a Compra*: ocorre quando se fecha o valor do pedido (envolvendo o seu custo), criando uma fidelidade na compra. Pode-se prosseguir para *Solicitar Trade* ou *Compra não Efetuar*;

- Variável *m9 - Reputação Negativa ao Solicitante*: um dos estados finais que existem na rede. Este estado ocorre quando o jogador ativo da Consistência deseja não prosseguir com a compra, quebrando a fidelidade da compra. Nesse caso, é penalizado com perdas de pontos em sua reputação;
- Variável *m8 - Espera Item Disponível*: como o nome já diz, trata-se da disponibilidade do item do jogador passivo, que pode negociar mesmo quando não tem o item. Nesse caso, passa-se para a transição *Item não Disponível*, que se mantém até que o vendedor tome posse do item, momento em que se segue para a transição *Solicitar Trade*;
- Variável *m4 - Trade*: neste estado, ocorre a real venda do produto. Após toda a fase de negociação e busca do item, passa-se para a transição *Confirmar Trade*, checando valores e itens de compra;
- Variável *m10 - Compra Realizada*: é onde ocorre todo o processo e finalização da compra, permitindo passar para a transição *Pontuar Reputação*;
- Variável *m11 - Reputação Pontuada por Venda*: é o estado final que permite analisar quantas vendas foram feitas utilizando este princípio, permitindo também uma maior reputação ao vendedor caso ocorram vendas futuras, criando um comportamento consistente;
- Variável *m26 - Reputação Pontuada por Compra*: semelhante ao estado anterior, porém, em relação ao comprador;
- Variável *m25 - Objetivo da Consistência Atingido*: como o próprio nome diz, a Consistência não ocorre apenas quando se cria um ciclo, que no caso seria o ideal, mas quando o vendedor ou o ajudante usou a Consistência, pode esta funcionar ou não. Em seguida, existe a transição *Ativando a Consistência*, que aborda a escolha do jogador ativo em ser consistente ou não;
- Variável *m25 - Espera Escolha do Solicitante*: este estado aguarda a ação do jogador para com as transições *Escolher não Retribuir* ou *Escolher Retribuir*;
- Variável *m27 - Reputação Negativa ao Solicitante*: é o estado final da rede e ocorre quando o jogador ativo não escolhe ser consistente a Consistência, pontuado, assim, negativamente a sua reputação;

- Variável *m28 - Reputação ao Solicitante*: na transição *Escolher Retribuir*, existem duas rotas em paralelo a seguir. Uma das rotas é a *Reputação ao Solicitante*, em que o jogador ativo é pontuado positivamente em sua reputação, favorecendo-o quando desejar ativar novamente a *Persuasão*. A outra rota vai para o estado *Espera para Completar o Clico da Consistência*, a seguir explicado;
- Variável *m29 - Espera para Completar o Clico da Consistência*: trata-se apenas de um estado que aguarda que alguém o ative, que, no caso, é o jogador passivo ou o ativo, seguindo, então, para a transição *Completa Ciclo da Consistência*, passa então para o estado *m13*;
- Variável *m15 - Ajuda em Espera*: ocorre quando o jogador ativo escolhe a transição *Pedir Ajuda*, podendo seguir para a transição *Solicita Party* na hipótese de existir um jogador na pretensão de ajudá-lo;
- Variável *m16 - Espera Party Disponível*: ocorre da mesma forma que o estado *m6*, porém, diferenciando-se na necessidade de verificar se a ajuda para *Party* estará disponível ou não. Caso esteja indisponível, irá para a transição *Party não Disponibilizada*. Já, estando disponível, passa-se para a transição *Solicitar Party*;
- Variável *m17 - Party*: ocorre quando os membros envolvidos a ajudar estão em grupo e utilizam a ferramenta *Party*. Dependendo da conduta, pode-se prosseguir até a transição *Ajuda em Andamento* ou *Party Cancelada*, que indica o cancelamento do grupo, haja vista o não comprometimento deste ou do jogador em ajudar;
- Variável *m17 - Reputação Negativa ao Solicitante*: caso o grupo ou jogador ativo solicite o cancelamento da *Party*, a sua reputação será pontuadas como negativas, como forma de penalidade;
- Variável *m14 - Ajuda Realizada*: trata-se do fim do processo de ajuda e ocorre quando se atinge o objetivo do grupo, prosseguindo-se para a transição *Pontuar Reputação*;
- Variável *m18 - Reputação Pontuada por Ajuda*: é semelhante ao estado *m11*, porém, a pontuação, nesse caso, ocorrerá pelo fato da ajuda;

- Variável *m29 - Reputação Pontuada ao Ajudado*: é semelhante ao estado *m26*, porém, esta pontuação ocorrerá pelo fato da ajuda. Prossegue-se para o estado *m24*, dando continuidade à rede, que já foi explicado anteriormente;
- Variável *m20 - Dica espera*: é o outro caminho a ser seguido na rede. Trata-se quando o jogador ativo solicita a transição *Pergunta Fórum*, aguardando até que algum jogador passivo possa respondê-lo, podendo ou não o jogador ativo *Confirmar Dica*. Caso contrário, o jogador passa para a transição *Não Confirma Dica*;
- Variável *m21 - Espera da Pergunta*: ocorre quando os jogadores passivos aguardam alguma pergunta para responder. Não havendo pergunta alguma, passa-se para transição *Pergunta não Realizada*. Já, na hipótese de ocorrer a resposta relacionada à pergunta feita, cabe ao jogador ativo *Confirmar Dica*;
- Variável *m22 - Dica Realizada*: este estado ocorre após o jogador *Confirmar Dica*, passando para a transição *Pontuar Reputação*;
- Variável *m23 - Reputação Pontuada por Dica*: semelhante ao estado *m18*. Mas, aqui, a maior reputação ocorre pelo fato de existir a participação do fórum em transmitir a dica. Em seguida, irá para o estado *m24*, dando continuidade a toda rede;

5.3. Autoridade aplicada ao MMORPG

No ambiente, existem inúmeras possibilidades de se aplicar a autoridade. A seguir, serão citadas as principais ferramentas para se utilizar as estratégias da autoridade:

- Na ordem de hierarquia das facções, onde cada membro da “*Guild*” desempenha o seu papel;
- Na ferramenta *Party*, onde existe o líder ou o personagem especialista na missão ou onde se acha o item;
- Os personagens mais experientes: esse atributo pode ser visto no *Level* (nível) do personagem;

- Existem também os heróis, que são personagens conhecidos no mundo do jogo.

5.3.1. Modulo Autoridade fluxo e o princípio modelado na rede Petri

A Autoridade nada mais é que jogadores que detêm certo conhecimento ou status durante o jogo, uma vez que estes não são auto proclamados.

No ambiente do MMORPG o princípio pode ser ativo de várias maneiras, já mencionadas na seção 5.3. Para facilitar uma melhor compreensão deste princípio modelado ao ambiente, foi criado um fluxo do seu funcionamento no ambiente, como pode ser visualizado na Figura 4.

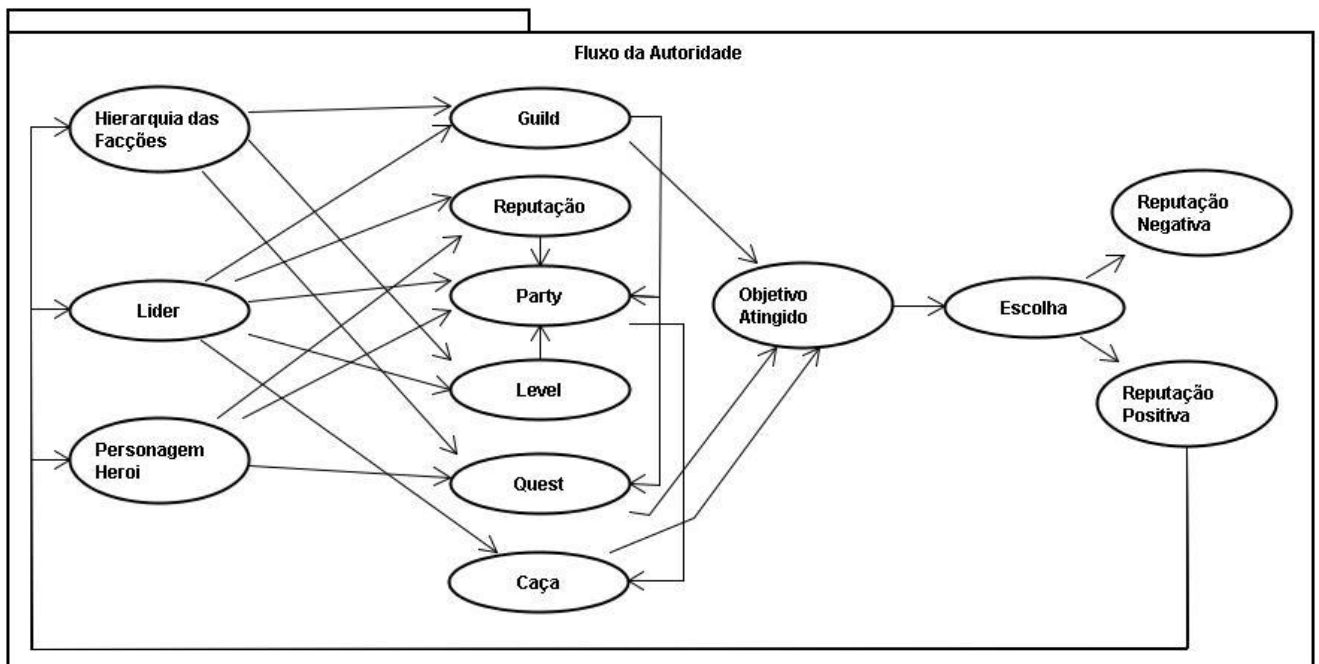


Figura 4: Fluxo da Autoridade.

A Autoridade inicia quando o jogador atua de forma hierática, esta possibilidade ocorre pela ferramenta *Guild*, ou escolhido para liderar a *Party*, com objetivo em comum, tais como, *Quest* ou *Caça*. A escolha do líder pode ser dada através de sua *Reputação* e/ou *Level* e/ou conhecimento de um determinado objetivo. Da mesma maneira que ocorre nos *Personagem Heroi*, onde este personagem é dotado de uma grande *Reputação*.

No APÊNDICE D – *Princípios da Persuasão aplicados ao MMORPG utilizando Redes de Petri*, encontra-se a Figura 57, que aborda a Autoridade modelada utilizando a Rede de Petri e permite visualizar os lugares (estados), as transições e o conjunto de arcos orientados.

Para a rede da Autoridade, podem-se declarar os seguintes lugares e suas variáveis, tais como:

- Variável *m1 - Início da Autoridade*: este estado aborda quando o jogador utiliza a Autoridade para obter algo de seu desejo, solicitando a transição *Solicitar Suporte*. O jogador apenas poderá prosseguir quando aquele solicitado estiver disponível;
- Variável *m2 - Solicitado Disponível*: é o estado onde o jogador inicial, que ativa a Autoridade, aguarda a disponibilidade de outro jogador, verificando se este está disponível ou não. Caso não tenha jogador disponível, passa-se para a transição *Solicitador não Disponível*. Caso exista um jogador disponível, passa-se para a transição *Solicitar Suporte*;
- Variável *m3 - Espera Pesquisa Reputação*: estado onde o jogador passivo e ativo do princípio irá verificar o perfil do jogador que irá ajudar. Existindo a possibilidade de seguir para a transição *Escolher Continuar*, caso o jogador seja consistente positivamente, ou *Escolher não Continuar*, caso este jogador disponível não seja consistente;
- Variável *m4 - Espera do Início da Autoridade*: é o estado onde o jogador deseja ativar a Autoridade, passando para a transição *Ativando a Autoridade*. É nesse momento que se inicia realmente a rede, passando para os estados *m1* e *m2*;
- Variável *m5 - Espera Pedido*: Ocorre quando o jogador ativo da consistência está a escolher o desejo que irá informar ao jogador passivo à Persuasão. Neste estado, o jogador poderá seguir para *Pedir Ajuda* ou *Pedir Ajuda Guild*;
- Variável *m6 - Espera Party*: ocorre quando o jogador ativo escolhe a transição *Pedir Ajuda*, podendo seguir para a transição *Solicita Party* na hipótese de existir um jogador na pretensão de ajudá-lo;

- Variável *m8 - Espera Party Disponível*: verificar se a ajuda para *Party* estará disponível ou não. Caso esteja indisponível, irá para a transição *Party não Disponibilizada*. Já, estando disponível, passa-se para a transição *Solicitar Party*;
- Variável *m10 - Party*: ocorre quando os membros envolvidos a ajudar estão em grupo e utilizam a ferramenta *Party*. Dependendo da conduta, pode-se prosseguir até a transição *Marcar Lider da Party* (definir no grupo quem é o líder) ou *Party Cancelada*, que indica o cancelamento do grupo, haja vista o não comprometimento deste ou do jogador em ajudar;
- Variável *m12 - Reputação Negativa ao Solicitante*: caso o grupo ou jogador ativo solicite o cancelamento da *Party*, a sua reputação será pontuadas como negativas, como forma de penalidade;
- Variável *m13 - Party em Andamento*: trata-se do estado onde o objetivo do grupo ainda estar em andamento, após terem definido o líder do grupo, depois de concluído passa-se para a transição *Concluir Objetivo*;
- Variável *m14 - Ajuda Realizada*: trata-se do fim do processo de ajuda e ocorre quando se atinge o objetivo do grupo, prosseguindo-se para a transição *Pontuar Reputação*;
- Variável *m15 - Reputação Pontuada por Ajuda*: ocorre a pontuação da reputação pelo fato da ajuda;
- Variável *m16 - Reputação Pontuada ao Ajudado*: é semelhante ao estado *m15*, porém, esta pontuação ocorrerá pelo fato de quem sofreu a ajuda. Em paralelo prossegue para o estado *m17*;
- Variável *m17 - Objetivo da Autoridade Atingido*: como o próprio nome diz, a Autoridade não ocorre apenas quando se cria um ciclo, que no caso seria o ideal, mas quando o ajudante ou ajudado usou a Autoridade, pode esta funcionar ou não. Em seguida, existe a transição *Ativando a Autoridade*, que aborda a escolha do jogador ativo em ser consistente ao principio ou não;
- Variável *m18 - Espera Escolha do Solicitante*: este estado aguarda a ação do jogador para com as transições *Escolher não Retribuir* ou *Escolher Retribuir*;

- Variável *m19 - Reputação Negativa ao Solicitante*: é o estado final da rede e ocorre quando o jogador ativo não escolhe ser consistente a Autoridade, pontuado, assim, negativamente a sua reputação;
- Variável *m20 - Reputação ao Solicitante*: na transição *Escolher Retribuir*, existem duas rotas em paralelo a seguir. Uma das rotas é a *Reputação ao Solicitante*, em que o jogador ativo é pontuado positivamente em sua reputação, favorecendo-o quando desejar ativar novamente a Persuasão. A outra rota vai para o estado *Espera para Completar o Clico da Autoridade*;
- Variável *m29 - Espera para Completar o Clico da Autoridade*: trata-se apenas de um estado que aguarda que alguém o ative, que, no caso, é o jogador passivo ou o ativo, seguindo, então, para a transição *Completa Ciclo da Autoridade*, passa então para o estado *m4*;
- Variável *m7 - Espera Party Guild*: estado semelhante ao *m6*, contudo este estado esta relacionado diretamente a *Guild*, pressupondo que esta *Guild* já tem a sua hierarquia. Este estado fica parado até que algum membro ou os membros estejam disponíveis, seguindo para a transição *Solicita Party Guild*;
- Variável *m9 - Espera Guild Disponível*: estado semelhante ao *m8*, porém está relacionado também diretamente a *Guild*. Este estado poderá seguir para *Party não Disponibilizada* (caso *Party* não disponível) ou *Solicita Party* (caso *Party* disponível);
- Variável *m11 - Party Guild*: semelhante a *m10*, no entanto como existe o compromisso da facção, neste estado não existirá a desistência do grupo. Para a transição *Marcar Lider da Party* existe apenas para definir o líder, podendo seguir a hierarquia da facção ou não. Que pode fim segue para o estado *m13*.

5.4. Prova Social aplicada ao MMORPG

No ambiente, existem inúmeras possibilidades de se aplicar a prova social. A seguir, serão citadas as principais ferramentas para se utilizar as estratégias da prova social:

- Nos fóruns, existe a possibilidade de visualizar a *Quest* mais comentada, a *Quest* mais fácil, os melhores monstros para caçar, os monstros ou *Quest* que dão mais itens, melhores itens a comprar e etc.
- Existe o melhor item para realizar determinado objetivo, essa informação pode ser fornecida por outros jogadores;
- Há a possibilidade de o jogador escolher a classe e raça que melhor o satisfaz, espelhando-se em opiniões de outros jogadores. O mesmo vale na escolha das habilidades (*Skills*);
- O jogador pode optar qual a melhor forma de atuar no ambiente, correta ou não, seguindo as propostas de outros jogadores.

5.4.1. Modulo Prova Social fluxo e o princípio modelado na rede Petri

A Prova Social ocorre quando parte do comportamento, público e privado, é condicionado pelas normas sociais, com o intuito de querer dar a melhor impressão, muitas vezes ocorre quando o jogador encontra-se em situações ambíguas.

No ambiente do jogo o princípio pode ser ativo de várias maneiras, já mencionadas na seção 5.4. Para facilitar uma melhor compreensão deste princípio modelado ao ambiente, foi criado um fluxo do seu funcionamento no ambiente, como pode ser visualizado na Figura 5.

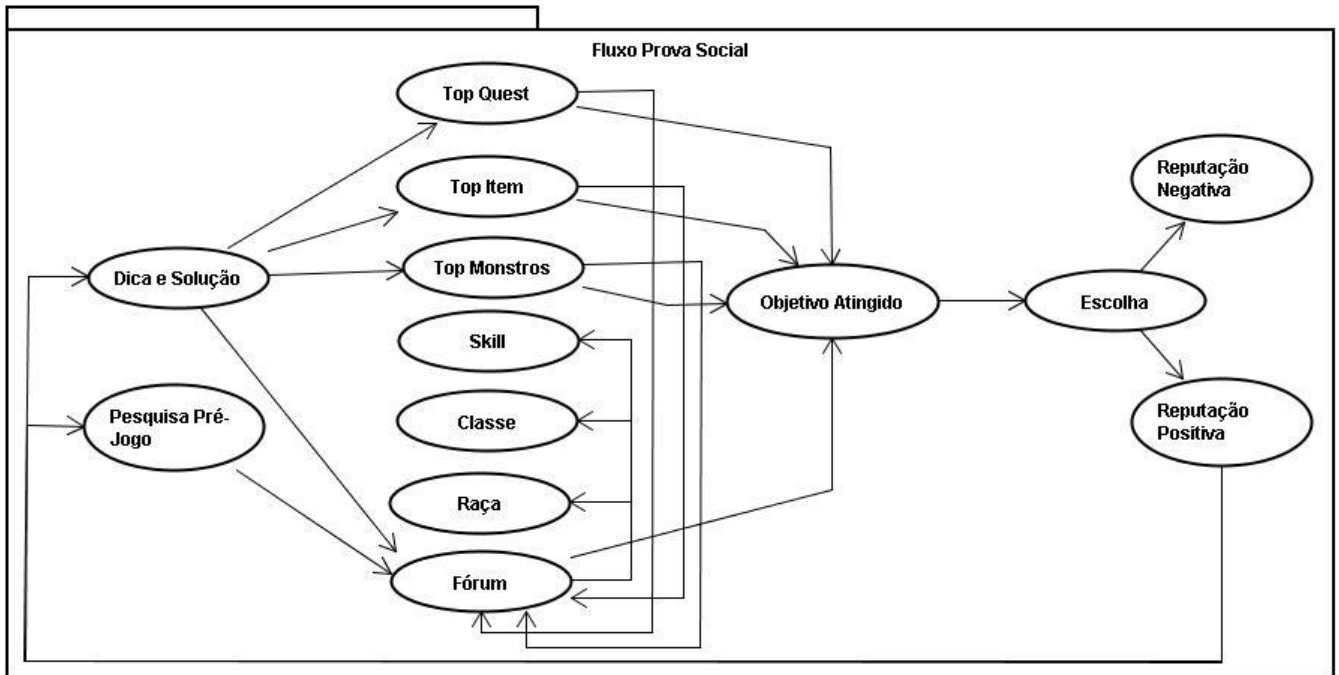


Figura 5: Fluxo da Prova Social.

A Prova Social é ativada quando o jogador busca dicas e soluções para suas eventuais dúvidas (*Dica e Solução*, *Pesquisa Pré-Jogo*). Para os tais problemas existe várias ferramentas tais como: *Top Quest*, *Top Item*, *Top Monstro* e *Fórum* (nesta ferramenta pesquisa-se o assunto da escolha do jogador).

No APÊNDICE D – *Princípios da Persuasão aplicados ao MMORPG utilizando Redes de Petri*, encontra-se a Figura 58, que aborda a Prova Social modelada utilizando a Rede de Petri e permite visualizar os lugares (estados), as transições e o conjunto de arcos orientados.

Para a rede da Prova Social, podem-se declarar os seguintes lugares e suas variáveis, tais como:

- Variável *m1* - *Início da Prova Social*: este estado aborda quando o jogador utiliza a Prova Social para obter algo de seu desejo, solicitando a transição *Acessar Fórum*;
- Variável *m2* - *Espera Tipo de Pesquisa*: é o estado que o jogador informa a modalidade da pesquisa que irá, passando para a transição *Pesquisa Fórum*;
- Variável *m13* - *Em pesquisa Fórum*: para o estado ocorre o fato da pesquisa do jogador, para encontrar uma solução relacionada à sua dúvida. Existindo

a possibilidade do jogar continuar na transição *Pesquisando* até encontrar a solução e *Confirmar Resposta* ou seguir para *Pesquisa não Encontrada*;

- Variável *m4 - Dica Realizada*: estado este quando soluciona-se a dúvida do jogador, prosseguindo para a transição *Pontuar Reputação*;
- Variável *m7 - Reputação Pontuada por Dica*: é o estado final que permite analisar quantas dicas foram realizadas utilizando este princípio, permitindo também uma maior reputação ao ajudante;
- Variável *m6 - Reputação Pontuada por Pergunta*: semelhante ao estado anterior, porém, em relação ao ajudado;
- Variável *m8 - Objetivo da Prova Social Atingido*: como o próprio nome diz, a Prova Social não ocorre apenas quando se cria um ciclo. Em seguida, existe a transição *Ativando a Prova Social*, que aborda a escolha do jogador ativo em retribuir ou não;
- Variável *m9 - Espera Escolha do Solicitante*: este estado aguarda a ação do jogador para com as transições *Escolher não Retribuir* ou *Escolher Retribuir*;
- Variável *m10 - Reputação Negativa ao Solicitante*: é o estado final da rede e ocorre quando o jogador ativo não escolhe retribuir a Reciprocidade, pontuado, assim, negativamente a sua reputação;
- Variável *m11 - Reputação ao Solicitante*: na transição *Escolher Retribuir*, existem duas rotas em paralelo a seguir. Uma das rotas é a *Reputação ao Solicitante*, em que o jogador ativo é pontuado positivamente em sua reputação, favorecendo-o quando desejar ativar novamente a Persuasão. A outra rota vai para o estado *Espera para Completar o Clico da Prova Social*;
- Variável *m12 - Espera para Completar o Clico da Reciprocidade*: trata-se apenas de um estado que aguarda que alguém o ative, que, no caso, é o jogador passivo ou o ativo, seguindo, então, para a transição *Completando Ciclo da Prova Social*;
- Variável *m15 - Espera do Início da Prova Social*: é o estado que indica quando a reciprocidade irá iniciar, seguindo para a transição *Ativando a Prova Social*. É nesse momento que se inicia realmente a rede, passando para o estado *m1*;

- Variável *m14 - Pesquisa não Encontrada*: estado onde o jogador desiste de sua pesquisa e irá realizar a transição *Perguntar Fórum*;
- Variável *m3 – Espera Resposta*: neste estado o jogador irá aguardar a resposta para sua pergunta realizada no fórum até que seja confirmada com a transição *Confirmar Resposta*, ou *Resposta não Disponibilizada*;
- Variável *m5 – Espera Pergunta*: o jogador ativo irá aguardar a pergunta, que pode ser confirmada ou *Pergunta não Disponibilizada*. Após a confirmação da resposta passa-se para o estado *m4*;

5.5. Escassez aplicada ao MMORPG

No ambiente, existem inúmeras possibilidades de se aplicar a escassez. A seguir, serão citadas as principais ferramentas para se utilizar as estratégias da escassez:

- As vendas em geral, onde o jogador estar disposto a pagar pelo item;
- Nos leilões envolvendo itens raros, onde há uma disputa entre jogadores para a posse de um determinado item;
- O “valor” que envolve na relação para se obter algum item ou *Skill* ou experiência ou objetivo na plataforma, exemplo: horas jogadas, sacrifícios de companheiros, *Quests*, itens e etc;
 - Brigas e disputas para obter o melhor lugar na cidade ou a melhor cidade ou o melhor ponto de caça.

5.5.1. Modulo Escassez fluxo e o princípio modelado na rede Petri

A Escassez funciona de forma poderosa sobre o montante que é atribuído às coisas, ou seja, o determinado valor que estar disposto a pagar para atingir o objetivo.

No ambiente do jogo o princípio pode ser ativo de várias maneiras, já mencionadas na seção 5.5. Para facilitar uma melhor compreensão deste princípio modelado ao ambiente, foi criado um fluxo do seu funcionamento no ambiente, como pode ser visualizado na Figura 6.

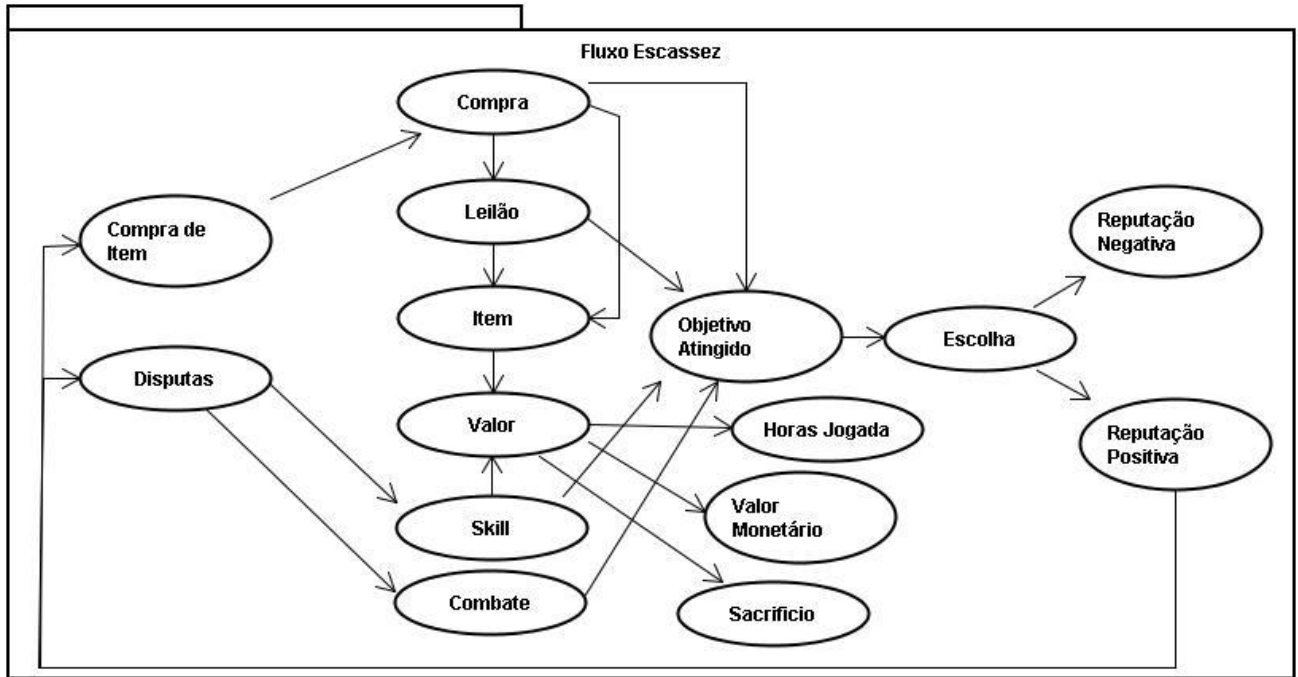


Figura 6: Fluxo da Escassez.

Perante o modelo este princípio atuara relacionado à *Compra de Item*, existindo duas modalidades de compra a *Compra* normal e a opção de *Leilão*, que para cada compra existe o *Valor* ligado ao produto. Outro caso é a *Disputas*, estas disputas por cidades, *Skill*, item e etc. para estas *Disputas* existem o(s) *Valor* agregado ao objetivo.

No APÊNDICE D – *Princípios da Persuasão aplicados ao MMORPG utilizando Redes de Petri*, encontra-se a Figura 59, que aborda a Escassez modelada utilizando a Rede de Petri e permite visualizar os lugares (estados), as transições e o conjunto de arcos orientados.

Para a rede da Escassez, podem-se declarar os seguintes lugares e suas variáveis, tais como:

- Variável *m1* - *Início da Escassez*: este estado aborda quando o jogador utiliza a Escassez para obter algo de seu desejo, solicitando a transição *Solicitar Suporte*. O jogador apenas poderá prosseguir quando aquele solicitado estiver disponível;
- Variável *m3* - *Solicitado Disponível*: é o estado onde o jogador inicial, que ativa a Consistência, aguarda a disponibilidade de outro jogador, verificando se este está disponível ou não. Caso não tenha jogador disponível, passa-se

para a transição *Solicitador não Disponível*. Caso exista um jogador disponível, passa-se para a transição *Solicitar Suporte*;

- Variável *m2* - *Espera Pedido Suporte*: Ocorre quando o jogador ativo da Escassez está a escolher o desejo que irá informar ao jogador passivo à Persuasão. Neste estado, o jogador poderá seguir para *Comprar Item Raro* ou *Pedir Ajuda*;
- Variável *m4* - *Escolher Tipo Compra*: o jogador passivo terá a opção de escolha da modalidade da venda. Em caso de uma compra normal passa-se para a transição *Compra Simples*, caso deseje o leilão segue para *Fazer Lance Leilão*;
- Variável *m9* - *Aguarda Fim Leilão*: para este estado ocorre quando o jogador efetua o seu lance e aguardo o fim do leilão, segue-se para *Leilão Finalizado*, quando o jogador não aceite o valor a se pagar encaminha-se para a transição *Valor não Aceito*, voltando ao estado *m4*. Caso não seja fiel no pagamento do valor usa-se a transição *não Fiel Leilão*.
- Variável *m10* - *Reputação Negativa ao Solicitante*: um dos estados finais da rede onde o jogador decide não pagar pelo valor mesmo depois de ter efetuado o lance de compra, criando uma penalidade na reputação do solicitante;
- Variável *m7* - *Fiel a Compra*: ocorre quando se fecha o valor do pedido (envolvendo o seu custo), apenas aguardando recebê-lo. Pode-se prosseguir para *não Fiel a Compra* indo o estado *m10* ou *Solicitar Trade*;
- Variável *m11* - *Trade*: neste estado, ocorre a real venda do produto. Após toda a fase de negociação e busca do item, passa-se para a transição *Confirmar Trade*, checando valores e itens de compra;
- Variável *m12* - *Compra Realizada*: é onde ocorre todo o processo e finalização da compra, permitindo passar para a transição *Pontuar Reputação*;
- Variável *m13* - *Reputação Pontuada por Venda*: é o estado final que permite analisar quantas vendas foram feitas utilizando este princípio, permitindo

também uma maior reputação ao vendedor caso ocorram vendas futuras, criando um comportamento consistente;

- Variável *m14* - *Reputação Pontuada por Compra*: semelhante ao estado anterior, porém, em relação ao comprador;
- Variável *m15* - *Objetivo da Escassez Atingido*: como o próprio nome diz, a Escassez não ocorre apenas quando se cria um ciclo, que no caso seria o ideal, mas quando o vendedor ou o ajudante usou a Escassez, pode esta funcionar ou não. Em seguida, existe a transição *Ativando a Escassez*, que aborda a escolha do jogador ativo em ser consistente ou não;
- Variável *m16*- *Espera Escolha do Solicitante*: este estado aguarda a ação do jogador para com as transições *Escolher não Retribuir* ou *Escolher Retribuir*;
- Variável *m17* - *Reputação Negativa ao Solicitante*: é o estado final da rede e ocorre quando o jogador ativo não escolhe ser consistente a Consistência, pontuado, assim, negativamente a sua reputação;
- Variável *m18* - *Reputação ao Solicitante*: na transição *Escolher Retribuir*, existem duas rotas em paralelo a seguir. Uma das rotas é a *Reputação ao Solicitante*, em que o jogador ativo é pontuado positivamente em sua reputação, favorecendo-o quando desejar ativar novamente a Persuasão. A outra rota vai para o estado *Espera para Completar o Clico da Escassez*, a seguir explicado;
- Variável *m19* - Em *Espera para Completar o Clico da Escassez*: trata-se apenas de um estado que aguarda que alguém o ative que, no caso, é o jogador passivo ou o ativo, seguindo, então, para a transição *Completa Ciclo da Escassez*;
- Variável *m27* - *Espera do Inicio da Escassez*: trata-se do estado onde o jogador ativa o princípio seguindo para os estados *m1* e *m3*;
- Variável *m5* - *Espera do Valor*: este estado aborda quando o jogador estar aguardando o valor do produto, possibilitando seguir para a transição *Valor não Aceito* chegando ao estado *m4*, ou *Valor Aceito* passando para o estado *m7*;

- Variável *m6 - Espera do Pedido*: o jogador passivo só poderá prosseguir, caso o item seja solicitado. Na hipótese de isso não ocorrer, passa-se para a transição *Pedido não Realizado*. Feito o pedido, o item prossegue para a transição *Valor Aceito*, escolhendo o valor a ser negociado;
- Variável *m8 - Espera Item Disponível*: como o nome já diz, trata-se da disponibilidade do item do jogador passivo, que pode negociar mesmo quando não tem o item. Nesse caso, passa-se para a transição *Item não Disponível*, que se mantém até que o vendedor tome posse do item, momento em que se segue para a transição *Valor Aceito*;
- Variável *m20- Ajuda em Espera*: ocorre quando o jogador ativo escolhe a transição *Pedir Ajuda*, podendo seguir para a transição *Solicita Party* na hipótese de existir um jogador na pretensão de ajudá-lo;
- Variável *m21 - Espera Party Disponível*: ocorre da mesma forma que o estado *m6*, porém, diferenciando-se na necessidade de verificar se a ajuda para *Party* estará disponível ou não. Caso esteja indisponível, irá para a transição *Party não Disponibilizada*. Já, estando disponível, passa-se para a transição *Solicitar Party*;
- Variável *m22 - Espera Valor da Ajuda*: para este estado os jogadores irão negociar o valor a ser pago com base no sacrifício do item, caso o *Valor não Aceito* o jogador segue para o estado *m20* e *m21*, caso confirmem valor usa-se a transição *Valor Aceito*;
- Variável *m23 - Party*: ocorre quando os membros envolvidos a ajudar estão em grupo e utilizam a ferramenta *Party*. Dependendo da conduta, pode-se prosseguir até a transição *Ajuda em Andamento*;
- Variável *m24 - Ajuda Realizada*: trata-se do fim do processo de ajuda e ocorre quando se atinge o objetivo do grupo, prosseguindo-se para a transição *Pontuar Reputação*;
- Variável *m25 - Reputação Pontuada por Ajuda*: é semelhante ao estado *m13*, porém, a pontuação, nesse caso, ocorrerá pelo fato da ajuda;

- Variável *m26 - Reputação Pontuada ao Ajudado*: é semelhante ao estado *m14*, porém, esta pontuação ocorrerá pelo fato da ajuda. Prossegue-se para o estado *m15*, dando continuidade à rede, que já foi explicado anteriormente;

5.6. Empatia aplicada ao MMORPG

No ambiente, existem inúmeras possibilidades de se aplicar a empatia. A seguir, serão citadas as principais ferramentas para se utilizar as estratégias da empatia:

- O ambiente proporciona a ferramenta *Chat*, onde jogadores que não se conhecem podem criar vínculos de amizade na medida em que interagem. O mesmo vale para jogadores de raça semelhantes ou que o mesmo se identifique;
- Existe também a possibilidade de que outros jogadores já tenham um vínculo de amizade ou familiar, o que torna mais fácil ativar essa técnica;
- Tendo em vista o sexo feminino ainda ser minoria entre os jogadores, um jogador do sexo feminino tem uma maior probabilidade de persuadir os jogadores do sexo oposto;
- Nos personagens pertencentes da mesma facção, é mais fácil ocorrer a empatia, pois participam do mesmo grupo.

5.6.1. Modulo Empatia fluxo e o princípio modelado na rede Petri

Para a Empatia sentir-se mais predisposto a fazer favores às pessoas de que se gosta é praticamente uma coisa normal, ou seja, menos óbvio, mas aceitável, é o fato de ser mais facilmente persuadido por pessoas de que se gosta.

No ambiente do jogo o princípio pode ser ativo de várias maneiras, já mencionadas na seção 5.6. Para facilitar uma melhor compreensão deste princípio modelado ao ambiente, foi criado um fluxo do seu funcionamento no ambiente, como pode ser visualizado na Figura 7.

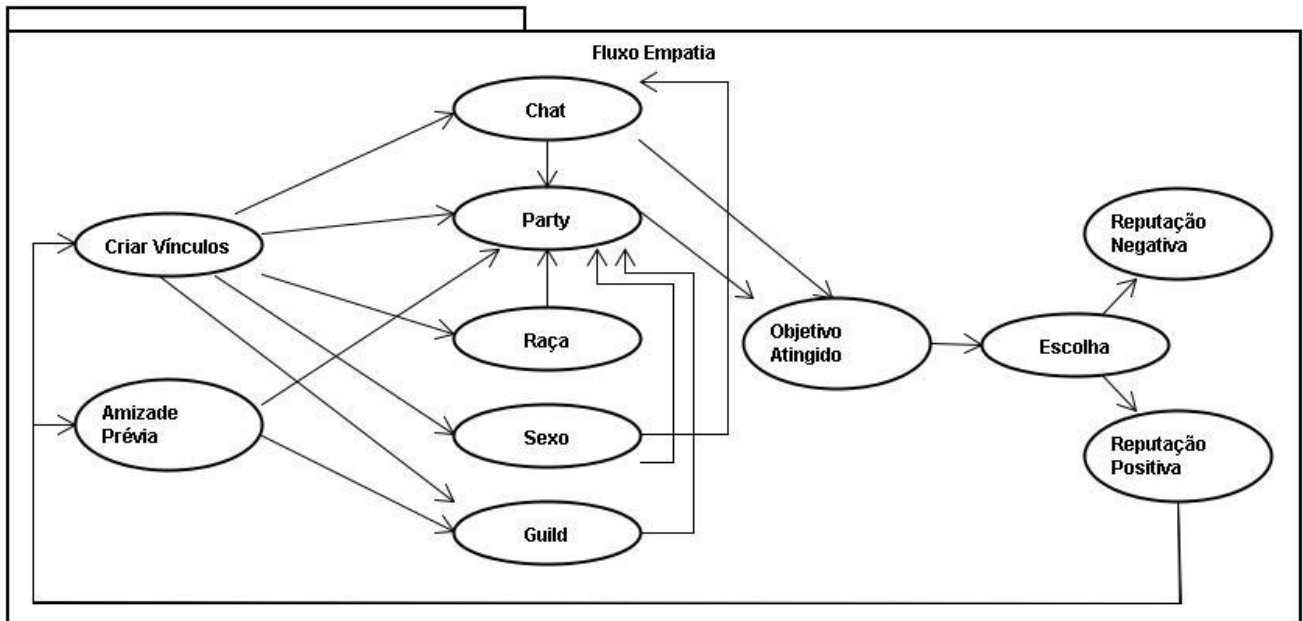


Figura 7: Fluxo da Empatia.

O modelo possibilita *Criar Vínculos* durante o acesso ao ambiente ou manter a *Amizade Prévia*, para permitir que isto ocorra existe o *Chat*, *Party*. Há também o caso da empatia criada em semelhanças de *Raça* ou *Guild* ou *Sexo* (semelhante ou opostos).

No APÊNDICE D – *Princípios da Persuasão aplicados ao MMORPG utilizando Redes de Petri*, encontra-se a Figura 60, que aborda a empatia modelada utilizando a Rede de Petri e permite visualizar os lugares (estados), as transições e o conjunto de arcos orientados.

Para a rede da Empatia, podem-se declarar os seguintes lugares e suas variáveis, tais como:

- Variável *m1* - *Início da Empatia*: este estado aborda quando o jogador utiliza a Empatia para obter algo de seu desejo, solicitando a transição *Solicitar Suporte*. O jogador apenas poderá prosseguir quando aquele solicitado estiver disponível;
- Variável *m2* - *Solicitado Disponível*: é o estado onde o jogador inicial, que ativa a Consistência, aguarda a disponibilidade de outro jogador, verificando se este está disponível ou não. Caso não tenha jogador disponível, passa-se

para a transição *Solicitador não Disponível*. Caso exista um jogador disponível, passa-se para a transição *Solicitar Suporte*;

- Variável *m3 - Espera Pedido Suporte*: Ocorre quando o jogador ativo da Empatia está a escolher o desejo que irá informar ao jogador passivo à Persuasão. Neste estado, o jogador poderá seguir para *Ajuda Mesma Guild*, *Ajuda Empatia Prévia* e *Ajuda Pós Chat*;
- Variável *m4 - Ajuda em Espera*: ocorre quando o jogador ativo escolhe uma das transições *Ajuda Mesma Guild*, *Ajuda Empatia Prévia* e *Ajuda Pós Chat*, todas as transições levam a o mesmo estado, porém o grande diferencial é de como será a forma de persuadir usando a Empatia. Quando tem o jogador disponível passa-se para o *Solicita Party*. Para este mesmo estado existe também a transição *Ajuda Negada*, esta transição ocorre quando o jogador solicita a ajuda e o jogador que iria sofrer a empatia não dar continuidade ao objetivo do princípio, levando ao estado *m3*;
- Variável *m5 - Espera Party Disponível*: busca a disponibilidade do jogador passivo, em outras palavras, verificar-se a ajuda para *Party* estará disponível ou não. Caso esteja indisponível, irá para a transição *Party não Disponibilizada*. Já, estando disponível, passa-se para a transição *Solicitar Party*;
- Variável *m7 - Party*: ocorre quando os membros envolvidos a ajudar estão em grupo e utilizam a ferramenta *Party*. Dependendo da conduta, pode-se prosseguir até a transição *Ajuda em Andamento ou Cancelar Party*;
- Variável *m8 - Reputação Negativa ao Solicitado*: este estado final aponta ao jogadores passivos que durante as Partys não completaram o objetivo;
- Variável *m6 - Ajuda Realizada*: trata-se do fim do processo de ajuda e ocorre quando se atinge o objetivo do grupo, prosseguindo-se para a transição *Pontuar Reputação*;
- Variável *m9 - Reputação Pontuada por Ajuda*: reputação pontuada pelo fato da ajuda;
- Variável *m10 - Reputação Pontuada ao Ajudado*: reputação pontuada pelo fato de utilizar o princípio;

- Variável *m11 - Objetivo da Empatia Atingido*: completando a empatia mesmo sem gerar o movimento cíclico. Em seguida, existe a transição *Ativando a Empatia*, que aborda a escolha do jogador ativo em ser consistente ou não;
- Variável *m12- Espera Escolha do Solicitante*: este estado aguarda a ação do jogador para com as transições *Escolher não Retribuir* ou *Escolher Retribuir*;
- Variável *m13 - Reputação Negativa ao Solicitante*: é o estado final da rede e ocorre quando o jogador ativo não escolhe ser consistente a Consistência, pontuado, assim, negativamente a sua reputação;
- Variável *m14 - Reputação ao Solicitante*: na transição *Escolher Retribuir*, existem duas rotas em paralelo a seguir. Uma das rotas é a *Reputação ao Solicitante*, em que o jogador ativo é pontuado positivamente em sua reputação, favorecendo-o quando desejar ativar novamente a Persuasão. A outra rota vai para o estado *Espera para Completar o Clico da Empatia*, a seguir explicado;
- Variável *m16 - Em Espera para Completar o Clico da Empatia*: trata-se apenas de um estado que aguarda que alguém o ative que, no caso, é o jogador passivo ou o ativo, seguindo, então, para a transição *Completa Ciclo da Empatia*;
- Variável *m27 - Espera do Início da Escassez*: trata-se do estado onde o jogador ativa o princípio seguindo para os estados *m1* e *m2*;

5.7. Conclusão do Capítulo

No capítulo 5, foi explicado o funcionamento de todo o modelo proposto, que é nada mais, que a união das Redes de Petris, Teoria dos Jogos e os Princípios da Persuasão aplicada ao ambiente MMORPG. Demonstrou-se o funcionamento do ambiente com todos os artifícios acima mencionados, indicando que tais instrumentos permitem ser utilizados para proporcionar uma melhor tomada de decisão e de estratégia na busca do conhecimento e, assim, possibilitam a implementação de um modelo lúdico satisfatório, no qual detenha ferramentas de apoio a persuadir outros jogadores na busca do conhecimento.

6 ESPECIFICAÇÃO DE UM MODELO DE TOMADA DE DECISÃO

No presente capítulo, será abordado detalhadamente todo o funcionamento do protótipo. Para explicar o funcionamento são utilizados os diagramas da UML, formalismo este pertencente à engenharia de *software* que ajustará no entendimento da atuação da Teoria da Persuasão no modelo de tomada de decisão proposto pelo autor.

6.1. Descrição do ambiente

O ambiente lúdico proposto abriga dois tipos de agentes que atuam ao longo do jogo: aqueles formados pelos jogadores que assumem o papel de um personagem da história; e aqueles formados pelos personagens controlados pelo computador (NPC's⁸). O modelo da topologia do modelo está exposto na Figura 8. Logo abaixo será apresentada a estrutura topológica do ambiente educacional e explicado mais detalhadamente os seus agentes (LORENZI *et. al*, 2001) (HAYDEN *et. al*, 1999).

- Agentes Reativos: são aqueles que dependem de um estímulo externo para se ativarem, ou seja, respondem em reação a um determinado estímulo. Este trabalho propõe dois agentes reativos para o jogo. O primeiro representa o Ambiente (mestre do jogo), que trabalha como âncora do sistema, permite a interação entre todos os mapas do cenário válidos para a história, reage em resposta ao servidor de jogo e a determinados estímulos dos personagens, possuindo ainda objetos livres que são considerados aqui como a posição no mapa e a descrição das características dos objetos que estão distribuídos pelo cenário. O segundo é formado pelo conjunto de jogadores que habitam este ambiente, quer sejam os jogadores ou aqueles controlados pelo computador, isto é, os NPC's.
- Agente Jogador humano ou Agentes Cognitivos: são aqueles que não dependem de estímulo para desempenharem suas ações, além disso, são capazes de decidir que ações e papéis vão desempenhar, atuando no ambiente de forma mais independente. Todavia, isso não impede que em

⁸ NPC – Non-Player Character.

determinados momentos eles atuam meramente em resposta a estímulos, assumindo neste momento uma postura reativa. O agente cognitivo proposto aqui é o agente jogador (jogador humano), responsável pelo controle das ações de um personagem, tal qual em um ambiente de RPG convencional, assumindo, portanto, o papel de intermediário na interação de um personagem com o ambiente e conseqüentemente com a história do jogo, e com os demais personagens. Características essas que conduzem a uma abordagem cognitiva.

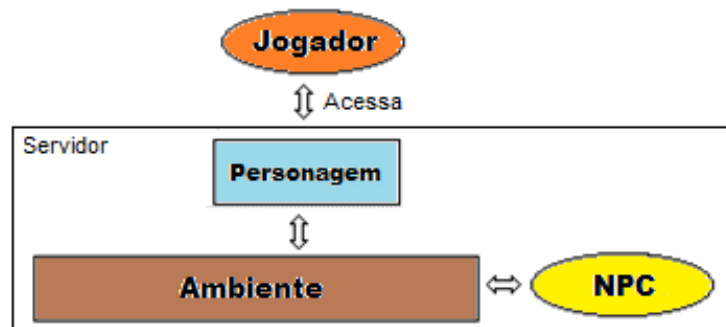


Figura 8: Topologia do modelo.

O ambiente foi projetado de forma distribuída permitindo ser executado em várias estações de trabalho distintas. Isso, além de facilitar o uso do ambiente pelos jogadores, permite que os agentes reativos sejam distribuídos em diversas máquinas, uma vez que os NPCs podem ser criados/instanciados pela aplicação em execução em cada estação de trabalho, cabendo ao mestre apenas manter a consistência de informação entre as estações a fim de evitar incoerências na representação/ação de um mesmo NPC representado em diferentes estações, enquanto que os jogadores como agentes cognitivos serão individualmente distribuídos em diferentes máquinas. Isto alivia o peso computacional e acelera o tempo de resposta da cognição. Para possibilitar essa distribuição, o sistema foi dividido em dois módulos principais: servidor e cliente. Conforme na Figura 9.

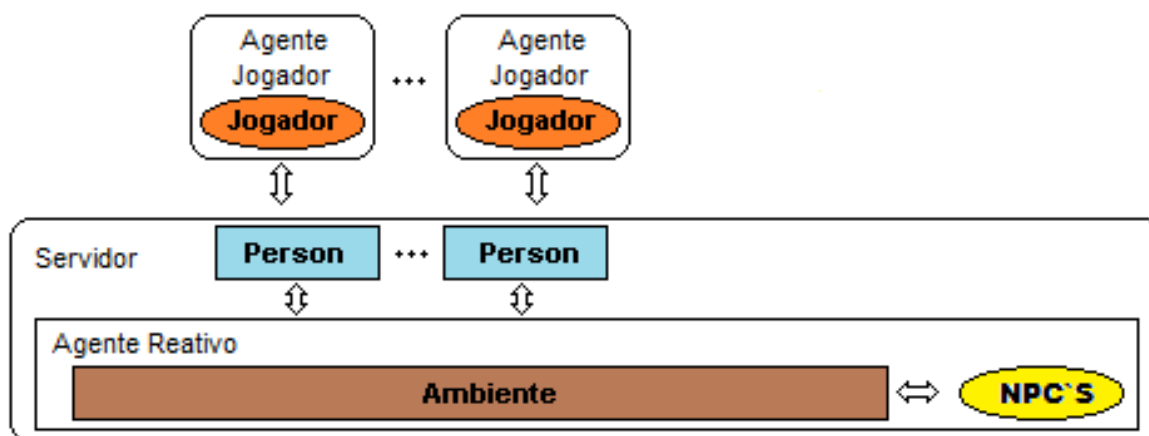


Figura 9: Topologia dos Agentes.

O Servidor é o módulo principal do sistema e é responsável por carregar o contexto da história, inicializar o jogo e garantir a comunicação entre os módulos clientes. Os agentes Personagens e o Ambiente ficam localizados no módulo Servidor junto com os personagens NPC's. O módulo Agente Jogador representa cada personagem do jogo controlando os agentes reativos.

Sendo assim, não se pode deixar de mencionar como ocorre essa troca de informação/conhecimento, com o mapa conceitual da troca do Conhecimento no ambiente, explanado na Figura 10;

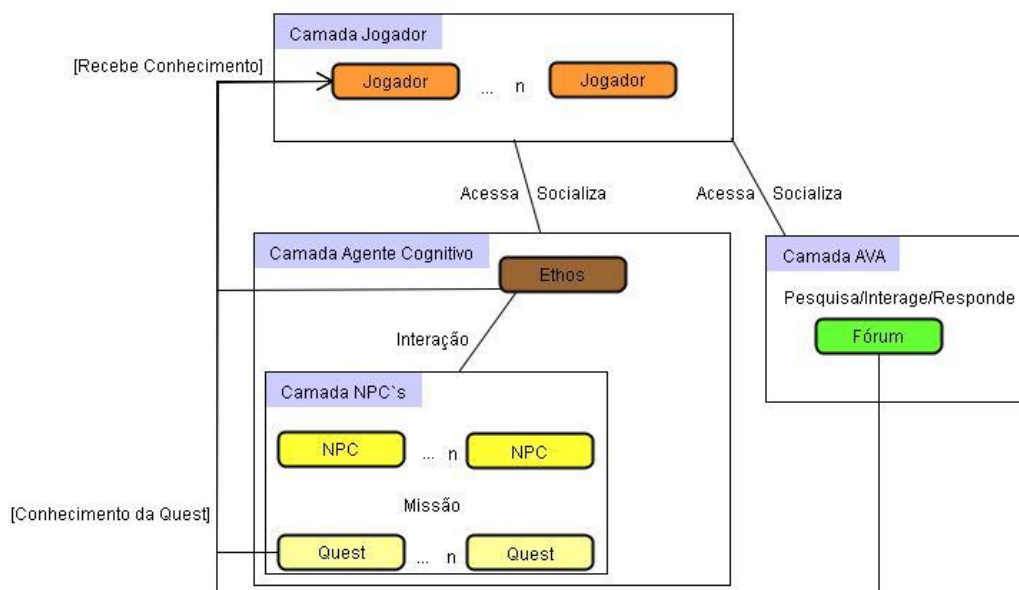


Figura 10: Mapa do Conhecimento.

- O conhecimento trafega nas quatro diferentes camadas existentes no ambiente lúdico, tais como: Camada Estudante (o jogador que irá acessar tanto ao jogo quanto aos fóruns na plataforma do ambiente virtual), Camada do ambiente virtual (responsável pelo armazenamento das informações postadas), Camada Agente Cognitivo (o ambiente lúdico educacional CAI⁹), Camada NPC`s (máquina instrutora do conhecimento que irá passar as *Quest's*, a fim de que os jogadores possam solucioná-las);
- Os estudantes irão acessar a plataforma e, assim, socializar-se com diferentes jogadores, interagindo com o ambiente educacional: monstros, itens, geografia dos mapas, *NPC's* e etc. Há a possibilidade da interação dos jogadores com os NPC's. Tais estudantes terão permissão de solucionar inúmeras *Quest's*;
- Existindo uma grande probabilidade de eventuais problemas ou de dúvidas no decorrer do jogo, o jogador acessará a plataforma do ambiente virtual, assim postará suas dúvidas ou problemas, permitindo também visualizar outros fóruns já postados – caso for a sua dúvida – ou então propor soluções para devidas dificuldades de outros jogadores;
- Através de todo o ciclo do conhecimento, o estudante irá receber toda a informação/Conhecimento por meio do acesso ao ambiente do jogo e ambiente virtual.

A modelagem e implementação deste ambiente lúdico seguem o padrão UML¹⁰ de modelagem de software, implementado com o auxílio da ferramenta de modelagem JUDE¹¹, obtendo assim um conjunto de abordagens que oferece o suporte necessário ao sucesso do tema proposto.

⁹ *Computer-Aided Instruction* ou *Computer-Assisted Instruction* significa “Instrução Assistida ou Auxiliada por Computador” e é um dos componentes do CBL (*Computer Based-Learning* - Aprendizagem baseada no computador) (RICKEL, 1989).

¹⁰ Linguagem padrão para especificar, visualizar, documentar e construir artefatos de um sistema e pode ser utilizada com todos os processos ao longo do ciclo de desenvolvimento e através de diferentes tecnologias de implementação (BOOCH, 2010).

¹¹ *Java and UML Developers Environment* (Ambiente para Desenvolvedores UML e Java) - software livre para modelagem UML (JUDE DESIGN & COMMUNICATION, 2009).

Para o tema sugerido, os fatores primordiais para se propor o modelo no padrão UML foram (BOOCH, 2010):





- Visualização (comunicação de modelos conceituais e alguns aspectos de um sistema que não são deduzidos apenas na leitura do código);
- Especificação (construir modelos: precisos, não-ambíguos e completos);
- Construção (não é uma linguagem de programação, mas seus modelos podem ser ligados diretamente a uma);
- Documentação (arquitetura e todos os detalhes documentados, definição de requisitos e testes).

Seguindo a proposta do padrão UML, a seguir serão apresentados os atores do modelo, os casos de uso, os diagramas de caso de uso, diagramas de classes, diagrama de estados, diagramas de subsistemas, os diagramas de classe com suas respectivas interfaces e o diagrama de componente com implantação.

No entanto, um fator conflitante para se construir o modelo proposto, deu-se em consequência do padrão UML ser destinado a modelagem de softwares de acordo com os diagramas anteriormente mencionados, acrescentando ainda diagramas que não serão abordados aqui por não fazerem parte do escopo desejado para este modelo. Dentre tais diagramas estão os de seqüência, o de colaboração e o diagrama de atividades. Além disso, também não constam do escopo da proposta a elaboração de regras de negócios ou mesmo a elaboração dos casos de uso detalhados. Portanto, não serão apresentados pelo autor quaisquer diagramas que necessitem da elaboração de regras de negócios. Esta abordagem foi tomada em virtude desta proposta apresentar um modelo para um jogo MMORPG e aprendizado social, não sendo, portanto interessante fixar regras que prendam autores ou desenvolvedores de jogos a quaisquer eventuais regras propostas, uma vez que estas devem ser adaptadas de acordo com o contexto abordado pelo jogo. A omissão destes elementos no modelo que se propõem aqui, objetiva assim a concepção de um modelo genérico que pode ser aplicado a diferentes propostas de jogos. Todavia, uma vez que se deseje aplicar este modelo ao projeto de um jogo, caso seja necessária a criação dos diagramas não abordados neste trabalho a fim de se obter uma modelagem adequada do software, será necessária a criação das regras de negócio pertinentes ao jogo.

6.2. Atores

Atores são usuários e/ou outros meios externos que desenvolvem algum papel em relação ao sistema. Os meios externos são hardwares e/ou softwares que, assim como os usuários, geram informações para o sistema ou necessitam de informações geradas a partir do sistema (BOOCH, 2010).


Atores	Descrição
 Jogador	Pessoa responsável por: <ul style="list-style-type: none"> -Criar conta e personagem; -Completar <i>Quest's</i>; -Comprar itens e <i>Skill's</i>¹²; -Formar <i>Party</i>¹³ e <i>Guild</i>¹⁴; -Vender itens aos NPC's; -Falar com outros jogadores; -Atacar monstros ou outros jogadores; -Responder e Criar Fóruns; -Dar lances nos leilões; -Consultar fóruns e leilões; -Movimentar-se nas áreas permitidas;
 Jogador_Vendedor	Pessoa responsável por: <ul style="list-style-type: none"> -Vender itens ou objetos por meio de leilão; -Vender itens ou objetos nos fóruns de vendas; -Vender informações para eventuais dúvidas de outros jogadores; -Vender força de trabalho para completar <i>Quest's</i>;
 Monstro	Pessoa responsável por: <ul style="list-style-type: none"> -Atacar outros jogadores por "opção"¹⁵; -Deixar cair itens ou dinheiro ao morrer; -Ganhar experiência na morte do jogador; -Dar experiência quando morrer; -Movimentar-se nas áreas permitidas;
 NPC	Pessoa responsável por: <ul style="list-style-type: none"> -Criar as <i>Guild's</i> para os personagens; -Repassar as <i>Quest's</i> e itens ou dinheiro da respectiva <i>Quest</i> completada; -Vender <i>Skill's</i> ou itens; -Comprar itens ou objetos; -Falar com os jogadores;

¹² Termo técnico do RPG que em Inglês significa habilidade a qual pertence ao personagem exemplo: auto-cura.

¹³ Termo técnico do RPG que em Inglês significa grupo de caça ou grupo de missão.

¹⁴ Termo técnico do RPG que em inglês significa corporação ou associação a qual o jogador pertence.

¹⁵ O caso da "opção" e o fato de tornar o jogo mais real, exemplo: um siri ataca apenas para se defender, porém um leão ataca por instinto.

	Pessoa responsável pelo: -Salvamento das informações do banco de dados do sistema.
---	---

Quadro 1: Atores.

É importante enfatizar que a diferença existente entre os atores *Jogador* e *Jogador_Vendedor* é que o *Jogador_Vendedor* realiza diferentes ações excepcionalmente atribuída ao vendedor do “produto” que será futuramente esclarecida. Enquanto, o *Jogador_Vendedor* tem a permissão de atuar como um simples *Jogador*. Outro fator que necessita ser esclarecido é o caso dos atores *Monstro* e *NPC*, pois, apesar de estarem internamente na plataforma, são apenas *softwares* dotados de inteligência artificial, desenvolvidos exclusivamente para operar suas respectivas funções no ambiente, contribuindo assim com a interatividade da plataforma.

6.3. Caso de Uso

A descrição dos serviços a serem oferecidos pelo sistema é designada aos Casos de Usos do sistema. Cada Caso de Uso descreve uma aplicação ou uso completo do *software*. O conceito de caso de uso não deve ser confundido como um módulo do sistema, já que Caso de Uso não é um componente do sistema, mas sim um dos seus empregos possíveis. Também não se deve confundir o conceito de Caso de Uso com o de função que possui um escopo muito mais limitado. Um Caso de Uso é muito mais abrangente, pois envolve todo um conjunto de transações que juntas constituem um serviço completo oferecido pelo sistema (BOOCH, 2010).

A especificação das funcionalidades do sistema na forma de casos de uso permite um olhar mais abrangente das aplicações do sistema, valorizando um levantamento mais completo e preciso de suas atribuições (BOOCH, 2010).

Logo abaixo, serão apenas apresentados os principais Casos de Uso do modelo proposto, pois se pretende somente trabalhar em cima da proposta de escopo do trabalho, sabendo-se que o funcionamento genérico de um MMORPG é de conhecimento dos jogadores e profissionais da área.

6.3.1. Jogador

- Manter¹⁶ Conta;
- Manter Personagem;
- Criar *Party*;
- Cancelar *Party*;
- Manter *Guild*;
- Completar *Quest*;
 - Escolher *Quest*;
 - Cancelar *Quest*;
 - Atacar;
 - Cancelar Ataque;
 - *Dropar*¹⁷ Experiência;
 - Passar de Level¹⁸;
 - Comprar Itens;
 - Comprar Objetos;
 - Cancelar Compra Itens;
 - Cancelar Compra Objetos;
 - Cancelar Venda Itens NPC;
 - Cancelar Venda Objetos NPC;
 - Vender Itens NPC;
 - Vender Objetos NPC;
 - Comprar *Skill*;
 - Cancelar Venda *Skill*;
 - Dialogar *Chat*¹⁹;
 - Responder Leilão;
 - Visualizar Leilão;

¹⁶ Função que contém: cadastrar, excluir, editar e cancelar.

¹⁷ Termo técnico do RPG que vem do Inglês *Drop* significa cair, soltar, largar.

¹⁸ Termo técnico do RPG que significa que o personagem ficou mais forte.

¹⁹ Termo do Inglês que significa bate-papo.

- Visualizar Item Venda;
- Responder Item Venda;
- Criar Fóruns;
- Responder Fóruns;
- Visualizar Fóruns;

6.3.2. Jogador_Vendedor

- Criar Leilão;
- Cancela Leilão;
- Cria Item Venda;
- Cancela Item Venda;
- Abrir o *Trade*²⁰;
- Cancelar *Trade*;
- Responder Leilão;
- Responder Item Venda;

6.3.3. Monstro

- *Dropar* Item;
- *Dropar* Dinheiro;
- *Dropar* Experiência;
- Passar de Level;
- Atacar;
- Cancela Ataque;

6.3.4. NPC

- Vender Item;
- Vender Objeto;
- Vender *Skill*;

²⁰ Ferramenta típica dos jogos do gênero MMORPG, possibilita a venda ou troca de produtos entre dois jogadores.

- Criar *Guild* para Jogador;
- Confirmar *Quest* Completa;
- Repassar *Quest*;
- Dar Item Bônus *Quest*;
- Dar Dinheiro Bônus *Quest*;
- Responder ao Jogador;
- Comprar Item do Jogador;
- Comprar Objeto do Jogador;

6.3.5. Sist. de Backup

- Efetuar o Backup;

6.4. Diagrama de caso de uso

O diagrama de caso de uso é usado para se identificar como o sistema se comporta em várias situações que podem ocorrer durante a operação. Descrevendo o sistema, seu ambiente é a relação entre os dois (BOOCH, 2010).

No entanto, um projeto de *software*, normalmente, contém um único Diagrama de Casos de Uso o qual descreve todo o conjunto de serviços oferecidos pelo sistema. Mas, como vêm ao caso, por se tratar de um sistema grande e complexo, foi possível construir vários Diagramas de Caso de Uso a partir da decomposição da plataforma principal, tornando melhor o entendimento do leitor.

A decomposição do Diagrama de Casos de Uso foi feita utilizando o padrão UML de Pacotes, demonstrada na Figura 11. Um pacote é um encapsulador que não possui uma semântica específica dentro dos projetos. É usado essencialmente para separar os elementos de Caso de Uso dos diferentes projetos, agrupando os elementos de cada projeto sem criar necessariamente um vínculo entre os elementos (BOOCH, 2010).

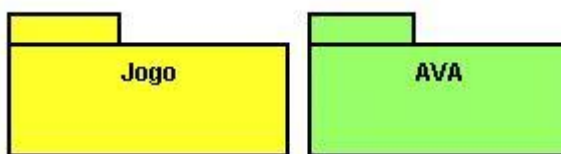


Figura 11: Pacotes do Modelo.

Em sendo assim, os atores situados à direita do pacote são os atores primários, ou seja, os principais atores para que o ambiente educacional funcione. Já os atores situados à esquerda do pacote são os atores secundários, atores estes importantes, mas, não tão quanto os primários. Já que o ambiente se trata de um jogo MMORPG, existe, dessa forma, a possibilidade de acesso para N estudantes/jogadores. Conseqüentemente, imagine-se que, no surgimento dos atores *Jogador* e *Jogador_Vendedor*, existe a probabilidade de ser N *Jogador* ou N *Jogador_Vendedor*, contudo não será exposto inúmeras vezes atores mencionados, simplificando em apenas um a cada de Caso de Uso.

6.4.1. Jogador e NPC

Nesse diagrama, ocorre a interação entre o *Jogador* como ator primário e o *NPC* como ator secundário, demonstrada na Figura 12. O Caso de Uso em todos os diagramas é representado pelo símbolo semelhante a uma elipse, proposto pelo padrão UML, onde internamente estará escrito o caso.

Existem algumas observações a serem feitas para o melhor entendimento do diagrama, tais como:

- O Caso de Uso *Completar_Quest*: será ativado no momento em que o *Jogador* se comunicar com o *NPC*, comprovado o cumprimento do(s) objetivo(s) da *Quest* e, se ocorre o fato da mesma tiver como prêmio algum tipo de bônus, é ativado o caso *Dar_Bonus*;
- O Caso de Uso *Comprar_NPC* e *Vender_NPC*: fica subentendido caso o *Jogador* for realizar uma compra ou uma venda ao *NPC*;
- O Caso de Uso *Guild*: é ativado apenas quando o *Jogador* solicitar ao *NPC*;
- O Caso de Uso *Falar_NPC*: trata-se da comunicação entre *Jogador* e *NPC*, abordado no jogo em forma de textos pré-definidos;

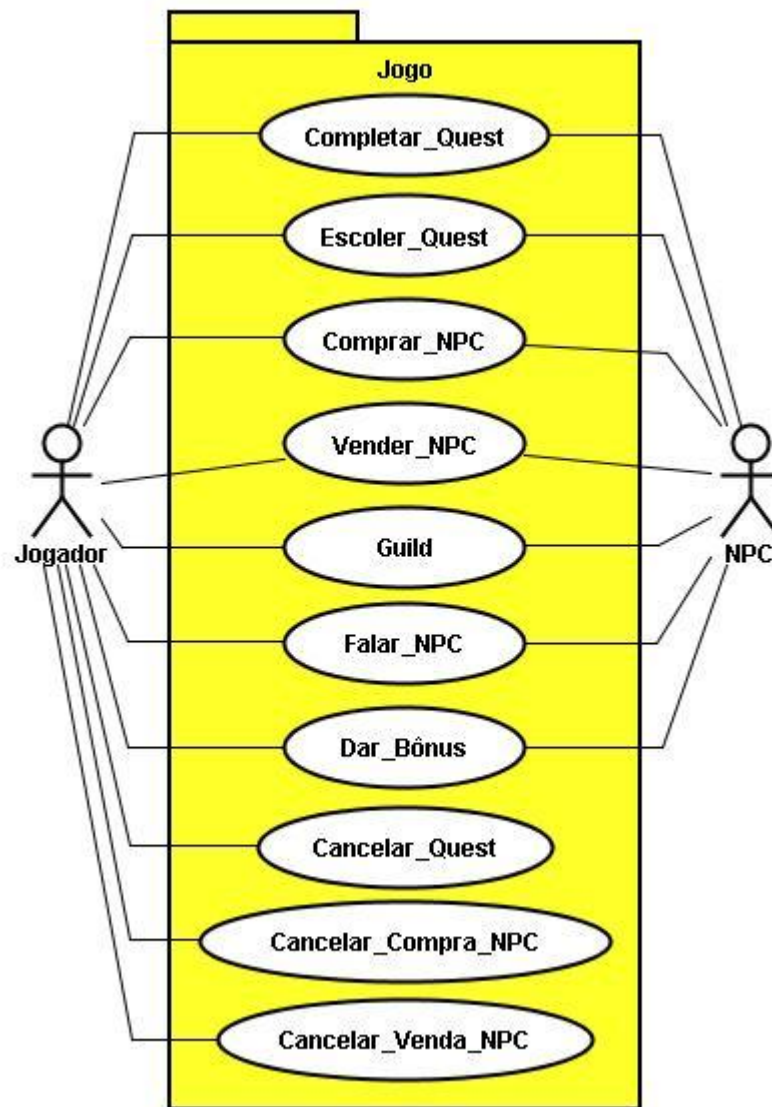


Figura 12: Diagrama de Caso de Uso Jogador e NPC.

6.4.2. Jogador e Jogador

No diagrama, fica explícito a interação entre *Jogador* e *Jogador*, Figura 13. Nesse diagrama, ocorre a possibilidade anteriormente mencionada, N Jogadores interagindo com o ambiente educacional. Tal ator atua como personagem primário e secundário, tal como pode ser visualizada na Figura 13.

O Caso de Uso *Criar_Party* será ativado apenas quando dois ou mais jogadores desejarem dividir a experiência, a fim de que possam subir de nível de forma cooperativa ou tenham algum objetivo em comum e, assim, queiram se ajudar.

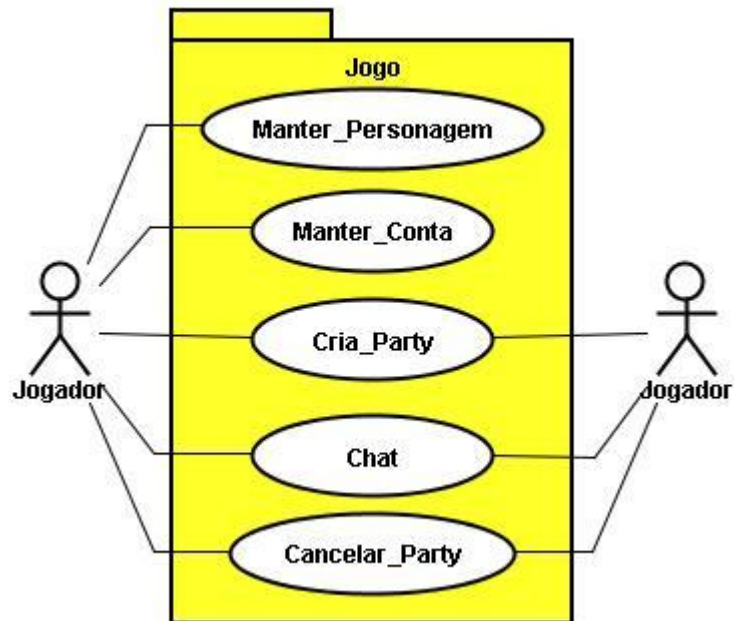


Figura 13: Diagrama de Caso de Uso Jogador e Jogador.

6.4.3. Jogador e Monstro

Para este diagrama trata-se do relacionamento entre *Jogador* e *Monstro*, Figura 14. Da mesma forma que o diagrama anterior, ocorre a possibilidade de ser *N Jogador* e *N Monstro*.

Ao Caso de Uso *Atacar*, poderá ser criada regras dedicadas à matemática do ataque, da mesma forma regras destinada aos casos *Dropa_Experiencia*, *Passar_Level*, *Dropar_Item* e *Dropar_Dinheiro*.

Já ao caso *Cancelar_Ataque*, nada mais é que o caso do *Monstro* ou o *Jogador* entrarem em modo de fuga para não morrer ou o *Jogador* não deseje mais atacar o *Monstro*.

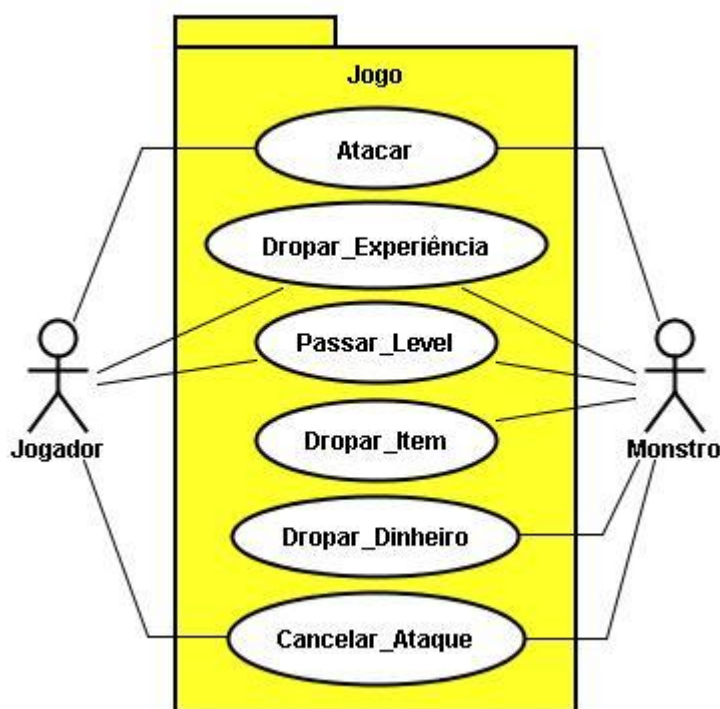


Figura 14: Diagrama de Caso de Uso Jogador e Monstro.

6.4.4. Jogador e Jogador Vendedor

Ao diagrama *Jogador* e *Jogador_Vendedor*, foi modelado seguindo o modelo de Ueyama (UEYAMA *et al.*, 2001) ao qual propõe um modelo de negócio eletrônico automatizado entre comprador e vendedor, adequando esse modelo, foi criado um modelo de negócio para um ambiente MMORPG, que visa haver a interação social, visualizado na Figura 15.

O principal meio de negociação do ambiente fica na plataforma do ambiente virtual, onde estão cadastrados os itens a venda e os leilões. No entanto na plataforma é onde ocorre a finalização do negócio no caso *Chat* e por fim o caso *Trade*.

Existem algumas observações a serem feitas aos respectivos Casos de Uso:

- Ler_Leilão: se trata da opção do jogador de visualizar todos os leilões correntes, onde mesmo possa realizar seu lance no caso *Responder_Leilão*;
- Ler_Item_Venda: caso que lista todos os itens a venda, assim, jogador poderá solicitar a sua compra no caso *Responder_Item_Venda*;

- Cria_Fóruns, Responde_Fóruns e Ler_Fóruns: são os casos onde o jogador relata dúvidas, soluções, maus vendedores e etc.

Fica-se claro que a escolha de participação na plataforma do ambiente virtual e de livre arbítrio jogador.

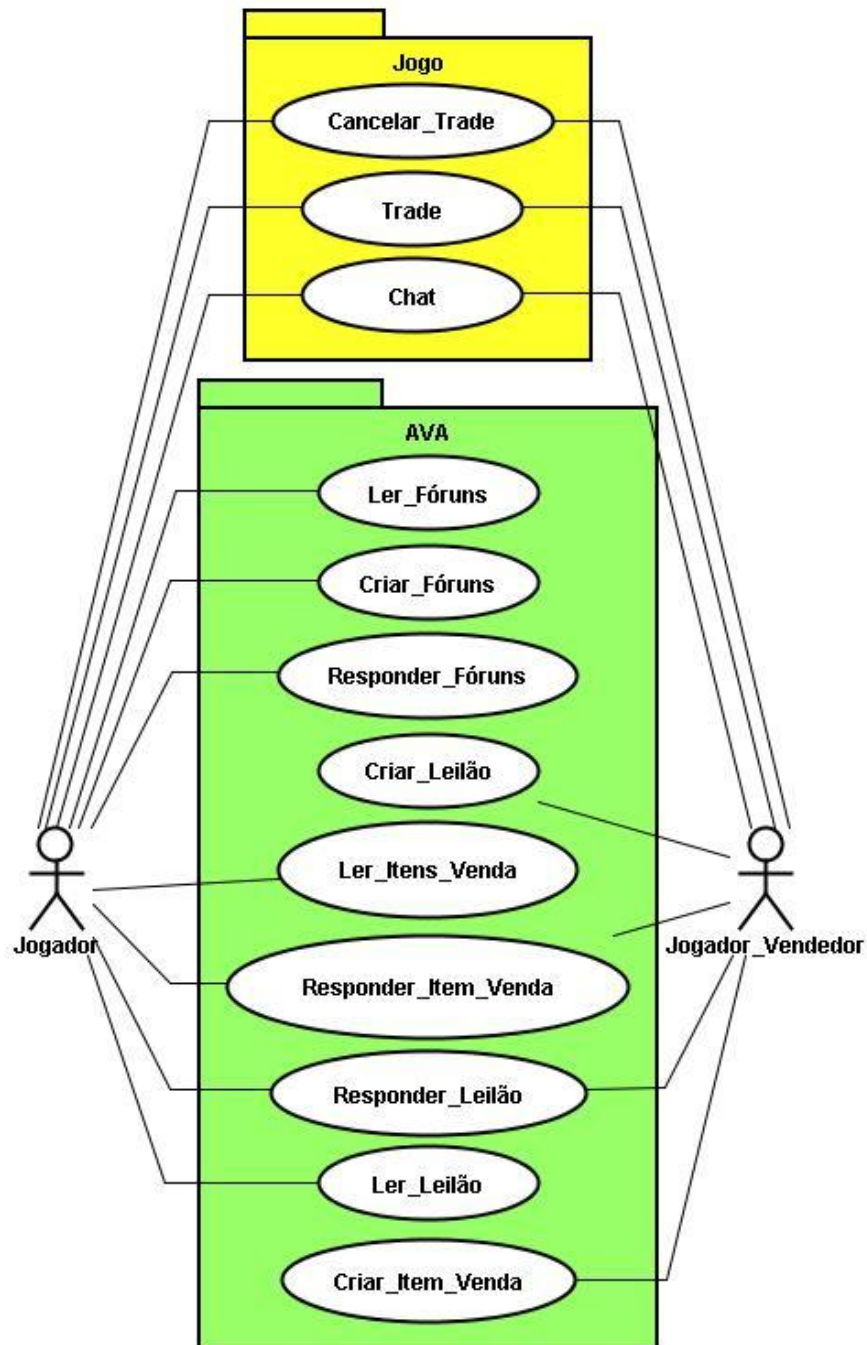


Figura 15: Diagrama de Caso de Uso Jogador e Jogador Vendedor.

6.4.5. Sistema de Backup

Caso de Uso este, que visa solucionar futuros problemas que possa ocorrer durante o processo de funcionamento do ambiente educacional, exibido na Figura 16. O *software* realizará eventualmente *backup* de todo o sistema ou de informações da escolha do administrador.

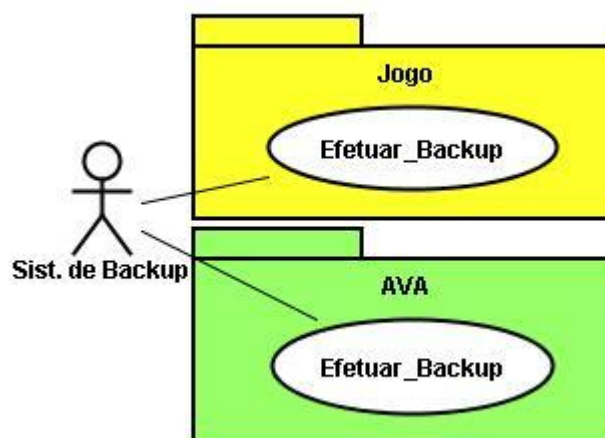


Figura 16: Diagrama de Caso de Uso Backup.

6.5. Diagrama de Classes

Para ter um entendimento melhor dos Diagramas de Classe, é necessário ter uma abordagem rápida nos conceitos de classe e objeto. Sendo assim, as classes e objetos são as principais primitivas ou elementos de composição de *softwares* orientados a objetos (BOOCH, 2010).

Um objeto pode ser visto ou entendido segundo uma visão conceitual e segundo uma visão de implementação (BOOCH, 2010):

- Visão conceitual: nesta visão, um objeto é um elemento componente de um sistema computacional que representa, dentro do sistema, alguma entidade ou coisa do mundo real ou que define uma porção do sistema com limites e atribuições bem definidos. De tal forma, um objeto possui três características, quais sejam: identidade, cada objeto possui um nome ou identificador que o distingue dos demais objetos e que permite que os outros objetos o reconheçam e o enderecem; atributos, cada objeto pode possuir informações que são registradas na forma de um conjunto de atributos; e comportamento,

cada objeto pode possuir um conjunto de habilidades de processamento que juntas descrevem seu comportamento.

- Visão de implementação: um objeto pode ser executado como qualquer módulo de *software*, receber dados de entrada e produzir resultados. Na orientação a objetos, procura-se construir estes módulos de *software* (objetos) de forma que eles tenham um alto grau de modularidade (pequena dependência entre um módulo e outro) e que cada módulo possua atribuições ou responsabilidades próprias que o caracterize. As três características de um objeto se traduzem da seguinte forma na implementação: identidade (objeto será determinado por um nome ou identificador inventado pelo programador para cada objeto); atributos (um objeto é traduzido na forma de variáveis internas do objeto); e comportamento (descrito por uma ou mais funções ou procedimentos).

Já, uma classe é a descrição de um tipo de objeto. Todos os objetos são instâncias de classes. A classe, por sua vez, descreve as propriedades e comportamentos daquele objeto. Os objetos só podem ser instanciados de classes. Ou seja, usam-se classes para classificar os objetos que são identificados no mundo real (BOOCH, 2010).

Compreendidos os conceitos básicos de classe e de objeto, segue-se adiante demonstrando os Diagramas de Classe do modelo. Não menos importante, são as observações ilustrativas para o entendimento do leitor no Quadro 2, em que se abordam as diferentes tonalidades dos *Forms* para o respectivo modelo. O *Form*, também chamado de fronteira, nada mais é que uma interface gráfica entre o jogador e o gerenciador de informações da plataforma.

Cor	Legenda
Amarelo	<i>Form principal do jogo;</i>
Cinza	Gerenciador de Informações do jogo;
Vermelho	<i>Form de exibição das Quest's;</i>
Verde	<i>Form do Trade;</i>
Azul	<i>Form de compra;</i>
Marrão	<i>Form de criar conta;</i>
Laranja	<i>Form de criar personagem;</i>
Ciano	<i>Form de criar Guild;</i>
Verde-oliva	Gerenciador de Informações ambiente virtual;
Rosa	<i>Form relacionado a vendas de itens;</i>
Amarelo-pálido	<i>Form relacionado aos fóruns;</i>
Laranja-avermelhado	<i>Form relacionado aos leilões;</i>

Quadro 2: Cores dos *Forms*.

A seguir será apresentado os principais Diagramas de Classe para se modelar um MMORPG com aprendizado social.


6.5.1. Diagrama de Classe Cria Conta e Cria Personagem Jogador

O diagrama de classe criador de conta e personagem trata os casos de uso *Manter_Conta* e *Manter_Personagem*. Para existir os referidos casos, necessita-se da existência das classes *Personagem_Jogador_Class*, *Conta_Class*, *Raça_Class* e *Sexo_Class*, possibilitando, assim, a existência dos objetos *Conta*, *Personagem*, *Raça* e *Sexo*.

O relacionamento entre classes ocorre da seguinte maneira:

- O jogador acessa a plataforma pela interface *Form_Jogo*, criando a sua conta pelo *Form_Cria_Conta*, seu personagem no *Form_Cria_Personagem*, iniciando, por fim, a sua jornada na plataforma;
- Internamente, o relacionamento entre as classes ocorre seqüencialmente na medida em que o jogador solicita as ações mencionadas nos casos de uso. No entanto, existem outras classes internas que se comunicam, solicitando outras informações para o seus funcionamentos (parâmetros para o

funcionamento dos métodos), tal como: o método *Novo_Personagem* que tem como uns dos parâmetros os atributos Raça e Sexo;

- Entre as classes, existe um único tipo de associação, seguindo a proposta UML. O modelo aborda a relação por composição, representada pelo símbolo () , que é chamado de agregação composta. Por exemplo, o relacionamento entre as classes *Personagem_Jogador_Class* e *Conta_Class* é dado pelo fato simples fato de que a existência do Personagem está condicionada à prévia criação da Conta, Raça e Sexo;
- Para cada tipo de associação entre classes, existe o fator da multiplicidade, que indica o número de instâncias de uma classe relacionado com uma instância de outra classe. Exemplo: o relacionamento entre as classes *Personagem_Jogador_Class* e *Raça_Class*. Para um Personagem, o jogador pode escolher uma dentre várias Raças. Por obrigação, o personagem deve ter uma Raça. Já, na relação entre *Conta_Class* e *Personagem_Jogador_Class*, pode existir, para uma Conta, um ou muitos Personagens, continuando da mesma forma as multiplicidades em todas as associações;
- Na proposta do padrão UML, existem alguns métodos padrões, tais como: *Get's*²¹ e *Set's*²². Por exemplo, no método *Get_Raça() : Raça_Class*, disponibiliza-se o objeto Raça da classe *Raça_Class*. Já, o método *Set_Raça() : void* modifica o objeto Raça na classe *Raça_Class*, atribuindo apenas os parâmetros do método, para que assim possa modificar o objeto;
- Em cada classe, existem métodos, como pode ser visualizados no *Apêndice A – Diagramas de Classes*, que contém a Figura 35. Estes métodos são as funções recorrentes aos objetos que a classe representa. Exemplo: para se criar uma Conta, utilizar-se o método *Cria_Conta(Nome_Conta : String, Email_Conta : String, Senha_Conta : String) : Conta_Class*, que tem os parâmetros a serem inseridos. Por fim, cria-se a conta solicitada pelo

²¹ Utiliza este método para solicitar alguma informação desejada do objeto na classe.

²² Utiliza este método para modificar alguma informação desejada do objeto na classe.

jogador, ocorrendo da mesma forma em todas as classes apresentada no modelo.

6.5.2. Diagrama de Classe Caçar

O diagrama de classe caçar aborda os casos de uso *Atacar*, *Cancelar Ataque*, *Dropar Experiência*, *Passar de Level*, *Dropar Item*, *Dropar Dinheiro*, *Dropar Experiência* e *Passar de Level*. Para existir os casos de uso, necessita-se da existência das classes *Matematica_Dano_Defesa_Class*, *Personagem_Jogador_Class*, *Level_Personagem_Class*, *Level_Monstro_Class*, *Monstro_Class*, e *Item_Class*, possibilitando, assim, a existência dos objetos Matemática do Dano e Defesa, Monstro, *Level* dos respectivos Monstro e Personagem e o Item.

O padrão UML da seção 6.5.1 aplica-se ao diagrama de Classe Caçar, acrescentando-se a esta, porém, algumas especificidades a seguir expostas, pelo fato de se tratar de um diagrama diferente da seção anterior:

- O jogador, ao acessar e selecionar o personagem na plataforma, por meio da interface *Form_Jogo*, interage com as diversas funcionalidades do jogo presente no *Form_Jogo*.
- O relacionamento das classes ocorre à medida que o jogador e o monstro atuam com o ambiente, havendo a possibilidade de ambos passarem de *level* (monstro e jogador), *dropar* itens;
- Foram criadas duas classes diferentes para passagem de *level* (*Level_Personagem_Class* e *Level_Monstro_Class*), pois, o padrão do jogo segue diferentes estruturas a depender dos atores, sendo as regras criadas de acordo com as necessidades dos autores ou desenvolvedores de jogos que utilizarem esse modelo. Por exemplo, pode-se criar a seguinte regra: em relação à passagem de *level* de personagem (tabela adequada a jogador), este ganhará experiência ao derrotar o monstro e não retornará ao nível padrão, quando derrotado; em relação ao *level* de monstro (tabela adequada a monstro), o monstro passará de *level* até o momento em que algum personagem ou *party* o derrote, regredindo o monstro ao nível padrão;

- Na Figura 36, representada no *Apêndice A – Diagramas de Classes*, pode-se observar que os métodos da classe *Personagem_Jogador_Class* foram ocultados para solucionar a grandeza do diagrama, possibilitando a existência de *N* classe *Personagem_Jogador_Class*.
- Caso exista alguma eventual dúvida em relação aos métodos da classe, checa-se a Figura 35.

6.5.3. Diagrama de Classe Adquirir Quest

O diagrama de classe adquirir *Quest* aborda os casos de uso *Completar Quest*, *Escolher Quest*, *Cancelar Quest*, *Repassar Quest*, *Dar Item Bônus Quest*, *Responder ao Jogador* e *Dar Dinheiro Bônus Quest*. Para existir os casos de uso, necessita-se da existência das classes *NCP_Class*, *Quest_Class*, *Item_Class* e *Personagem_Jogador_Class*, possibilitando, assim, a existência dos objetos *Quest*, *NPC*.

O padrão UML da seção 6.5.1 aplica-se ao diagrama de classe adquirir *Quest*, acrescentando-se a esta, porém, algumas especificidades a seguir expostas, pelo fato de se tratar de um diagrama diferente da seção anterior:

- O jogador, utilizando-se do personagem na plataforma e interface *Form_Jogo*, interage com o NPC, solicitando as *Quest's* disponíveis e exibindo o *Form_Quest*, que permite ao jogador selecionar a *Quest* desejada. Há também a possibilidade de cancelar e completar a *Quest*, e, após a sua conclusão, o jogador é bonificado;
- As classes *Sprite_Class*, *Valor_Item_Class* e *Atributo_Item_Class*, são classes complementares a plataforma, uma vez que *sprite* é a aparência do personagem, monstro, item e etc (interface gráfica). Já, as classes relacionadas ao item são responsáveis para gerar o item propriamente dito, quando solicitado pela classe *Item_Class*;
- No *Apêndice A – Diagramas de Classes* encontra-se Figura 37;

6.5.4. Diagrama de Classe Compra Item do Jogador

O diagrama de classe compra item do jogador aborda os casos de uso *Comprar Itens*, *Comprar Objetos*, *Cancelar Compra Itens*, *Cancelar Compra Objetos*, *Dialogar Chat*, *Abrir o Trade* e *Cancelar Trade*. Para existir os casos de uso, necessita-se da existência das classes *Item_Class*, *Personagem_Jogador_Class* (duas classes), *Money_Class*, e assim possibilitando a existência dos objetos *Money*²³, *Venda*.

Esse diagrama aborda o comércio entre dois jogadores, criando um modelo de negócio para MMORPG. Comércio este que pode ser iniciado na plataforma do ambiente virtual ou no próprio ambiente do jogo.

Para entender melhor a diagrama, necessita-se expor algumas observações:

- O jogador, utilizando-se do personagem na plataforma e interface *Form_Jogo*, interage com outro jogador, utilizando-se como ferramenta o *Chat*. Após fechado negócio na plataforma do ambiente virtual (leilão ou compra de item), solicita-se ao outro jogador a abertura do Trade para a venda ou troca de item na plataforma. Método este encontrado na classe *Personagem_Jogador_Class* (+ *Trade(Item : Item_Class, Personagem_Jogador : Personagem_Jogador_Class) : void*). Depois de confirmado os valores da venda ou troca no *Form_Trade*, ambos os jogadores precisam clicar “OK” para efetuar a venda, caso contrário, clica-se em cancela o *Trade*;
- Caso ocorra um escambo, o *Trade* apenas receberá objeto(s) ou item(s), da classe *Item_Class*. Ocorrendo o evento da venda, com a interveniência de moeda, utiliza-se a classe *Money_Class*;
- No *Apêndice A – Diagramas de Classes*, encontra-se Figura 38;

6.5.5. Diagrama de Classe Compra ou Vende Item do NPC

O Diagrama de Classe Compra ou Vende Item do NPC aborda os casos de uso: *Comprar Itens*; *Comprar Objetos*; *Cancelar Compra Itens*; *Cancelar Compra Objetos*; *Cancelar Venda Itens NPC*, *Cancelar Venda Objetos NPC*; *Vender Itens NPC*;

²³ Do inglês dinheiro.

Vender Objetos NPC; Vender Item; Vender Objeto; Responder ao Jogador; Comprar Item do Jogador e Comprar Objeto do Jogador. Para existir os casos de uso, necessita-se da existência das classes *Item_Class*, *Personagem_Jogador_Class*, *Money_Class*, *Atributo_Item_Class*, *Valor_Item_Class* e *NCP_Class*, possibilitando, assim, a existência dos objetos Money, Venda, Item.

Esse diagrama aborda o comércio entre jogador e NPC, criando um pseudo modelo de negócio para MMORPG. O único problema é o fator de que o jogador não pode negociar o valor do objeto ou item, pois, o NPC apenas tem os valores do produto definidos pela classe *Valor_Item_Class*. Por exemplo, caso o item seja usado e o jogador desejar vender, o próprio gerenciador do jogo verificará o valor do item, ocorrendo o mesmo para comprar itens novos.

Para entender melhor a diagrama, necessita-se expor algumas observações:

- O jogador, utilizando-se do personagem na plataforma e da interface *Form_Jogo*, interage com NPC, aproveitando-se de falas predefinidas. Solicitando a compra ou venda de item(s) e/ou objeto(s), abre-se o *Form_Compra*. E por fim, realiza-se a venda ou compra;
- No *Apêndice A – Diagramas de Classes*, encontra-se a Figura 39;

6.5.6. Diagrama de Classe Compra Skill

O Diagrama de Classe Compra *Skill* aborda os casos de uso *Comprar Skill*, *Cancelar Venda Skill*, *Vender Skill* e *Responder ao Jogador*. Para existir os casos de uso, necessita-se da existência das classes *Personagem_Jogador_Class*, *Money_Class*, *Skill_Class* e *NCP_Class*, possibilitando, assim, a existência dos objetos Money, Venda, Item.

Esse diagrama trabalha semelhantemente ao diagrama da seção 6.5.5, com a única diferença de que o jogador apenas pode comprar as *Skill's* do NPC.

Para entender melhor a diagrama, necessita-se expor algumas observações:

- O jogador, utilizando-se do personagem na plataforma e interface *Form_Jogo*, interage com NPC, aproveitando-se de falas predefinidas. Solicitando a compra da *Skill*, abre-se o *Form_Compra*. E por fim, realiza-se sua compra ou a cancela;

- A classe *Dano_Defesa_Skill_Class* dá suporte à classe *Skill_Class*, em relação a danos e defesa da *Skill*;
- No *Apêndice A – Diagramas de Classes*, encontra-se Figura 40;

6.5.7. Diagrama de Classe Cria *Guild*

O Diagrama de Classe Cria *Guild* aborda os casos de uso *Manter Guild* e *Criar Guild para Jogador*. Para existir os casos de uso, necessita-se da existência das classes *Personagem_Jogador_Class* e *NCP_Class*, possibilitando, assim, a existência do objeto *Guild*.

Esse diagrama aborda a criação de facções entre jogadores, possibilitando que estes expressem suas filosofias, hierarquia, organização e interação social. O jogador (chefe da *Guild*) solicita a criação ao NPC, podendo também acrescentar novos membros ou desistir da facção.

Para entender melhor a diagrama, necessita-se expor algumas observações:

- O jogador, utilizando-se do personagem na plataforma e interface *Form_Jogo*, interage com NPC, aproveitando-se de falas predefinidas, solicitando a criação da *Guild* no *Form_Cria_Guild*. Neste *Form*, é possível acrescentar novos membros ou desfazer a facção;
- No *Apêndice A – Diagramas de Classes*, encontra-se Figura 41;

6.5.8. Diagrama de Classe Cria *Party*

O Diagrama de Classe Cria *Party* aborda os casos de uso *Criar Party* e *Cancelar Party*. Para existir os casos de uso, necessita-se da existência das classes *Party_Class*, *Personagem_Jogador_Class* (N classes), *Distribuir_Experiencia_Class*, *Monstro_Class*, *Level_Personagem_Class* e *Level_Monstro_Class*, possibilitando, assim, a existência do objeto *Party*. Tal diagrama aborda a interação entre jogadores com objetivo(s) em comum, criando um ambiente cooperativo e colaborativo.

O Diagrama de Classe Cria *Party* é semelhante ao diagrama da seção 6.5.2, existindo, porém, a probabilidade do Monstro derrotar os membros da *Party* e aumentar o seu *level*, ao mesmo tempo em possibilita os membros de passarem de *level*,

ajudando-os a derrotar monstros de nível mais alto. Da mesma forma, a *Party* pode ser formada quando se tem *Quest's* em comum.

Para entender melhor a diagrama, necessita-se expor algumas observações:

- O jogador, utilizando-se do personagem na plataforma e interface *Form_Jogo*, interage com outro jogador, formando a *Party* e utilizando-se como ferramenta o *Chat* para comunicar-se;
- Caso os membros da *Party* derrotem algum monstro, ativa-se a classe *Distribuir_Experiencia_Class*, que tem por função dividir toda a experiência para o grupo. O mesmo ocorre quando o monstro derrote algum membro da *Party*;
- O caso de uso *Party* só existe apenas no ato de ter dois ou mais jogadores em grupo;
- No *Apêndice A – Diagramas de Classes*, encontra-se Figura 42.

6.5.9. Diagrama de Classe Fóruns, Leilão e Itens a Venda

O Diagrama de Classe Fóruns, Leilão e Itens a Venda aborda os casos de uso *Responder Leilão*, *Visualizar Leilão*, *Visualizar Item Venda*, *Responder Item Venda*, *Criar Fóruns*, *Responder Fóruns*, *Visualizar Fóruns*, *Criar Leilão*, *Cancela Leilão*, *Cria Item Venda* e *Cancela Item Venda*. Para existir os casos de uso, necessita-se da existência das classes *Forum_Class*, *Itens_Venda_Class*, *Conta_Forum_Class* e *Leilao_Class*, possibilitando, assim, a existência dos objetos Fórum, Venda, Leilão, Conta Fórum.

O Diagrama inicial do modelo de comércio proposto ao MMORPG aborda o comércio entre vários jogadores, na plataforma ambiente virtual, finalizado no ambiente do jogo. Há também a interação entre os jogadores nos fóruns, existindo a troca de informação para solucionar algum problema, *Quest's* e etc.

Para entender melhor a diagrama, necessita-se expor algumas observações:

- O jogador, utilizando-se de sua conta na plataforma do ambiente virtual, acessa o Form desejado, a interface *Form_Ler_Foruns*, *Form_Responde_Foruns* e *Form_Cria_Foruns*. Esses *Forms* são responsáveis pelos fóruns. Já, nos *Form_Cria_Item_Venda*,

Form_Ler_Item_Venda e *Form_Responde_Item_Venda*, ocorre o comércio em geral, existindo, porém, uma variação de venda, que é o leilão representado pelos *Form_Cria_Leilao*, *Form_Responde_Leilao* e *Form_Ler_Leilao*;

- Para cada conta, existe um atributo chamado reputação do jogador (- *Reputacao_Conta : int*). Esses pontos são atribuídos a cada venda finalizada ou informações postadas nos fóruns, atribuídas como positivas ou negativas pelos outros jogadores;
- No entanto, os fóruns têm sua reputação adquirida de quem cria ou de quem responde, dando a escolha para o jogador acreditar ou não;
- Já as vendas, funcionam semelhantemente ao comércio eletrônico. Caso seja uma simples venda de item, este é negociado e solicitado na plataforma ambiente virtual e finalizado no ambiente do jogo, sendo, por fim, atribuído os pontos de reputação ao comprador e vendedor. Os leilões nada mais são que uma variação de venda, tendo como únicas diferenças a existência de uma disputa para o produto desejado e de um prazo para finalizar o leilão;
- No *Apêndice A – Diagramas de Classes*, encontra-se Figura 43;

6.6. Diagrama de Estado

Da mesma maneira que foi apresentado no sessão 6.5, para melhor entendimento do modelo, é necessário apresentar os Diagramas de Estado.

O comportamento de uma Classe de Objetos é representado através de um Diagrama de Transição de Estado ou simplesmente chamado de Diagrama de Estado, que descreve o ciclo de vida de uma dada classe, os eventos que causam a transição de um estado para outro e as ações resultantes da mudança de estado (BOOCH, 2010).

O espaço amostral dos estados de uma dada Classe corresponde à enumeração de todos os estados possíveis de um objeto. O estado de um objeto é uma das possíveis condições na qual o objeto pode existir. O estado compreende todas as propriedades dos objetos (estáticas) associadas aos valores correntes (dinâmico) de cada uma dessas propriedades.

Um diagrama de estados é um grafo dirigido, cujos nodos representam estados e cujos arcos representam transições entre estados. Estados são representados graficamente por elipses ou retângulos com bordas arredondadas e transições são representadas por segmentos de retas dirigidos (com uma seta em uma das extremidades). Descrevem-se a seguir os conceitos de estado e transição de estado (BOOCH, 2010):

- Estado de um Objeto: um estado pode ser entendido como um momento na vida de um objeto. Neste momento (ou estado), o objeto encontra-se em uma certa situação. Dito de forma análoga, um objeto ao longo de sua vida passa por um conjunto de diferentes estados. Nos diagramas de estados, procura-se descrever este conjunto de estados e seu encadeamento. Existem duas notações gráficas especiais para dois estados particulares existentes em diagramas de estados. Utiliza-se a notação de um círculo preenchido para indicar o estado inicial do diagrama de estados. O estado inicial é um estado de partida do objeto. Pode-se entender que ele representa o momento de criação ou alocação do objeto. Utiliza-se a notação de dois círculos concêntricos para representar o estado final de um objeto. O estado final representa o momento de destruição ou deslocação do objeto;
- Transição de Estado: os objetos tendem a avançar de um certo estado para algum outro à medida que executem seus processamentos. Este avanço de uma situação (estado) para outra denomina-se uma transição de estado. Assim, em um diagrama de estados, descreve-se o conjunto de estados de um objeto e também o conjunto de transições de estados existentes. No entanto, existem algumas condições na transição de estado, são elas:
 - / Ação: indica uma ação (cálculo, atribuição, envio de mensagem, etc) que é executada durante a transição de um estado a outro;
 - [Condição]: é uma expressão lógica que é avaliada quando o evento (se houver), associado a uma transição, ocorrer. Uma transição só ocorre se o evento acontecer e a condição associada for verdadeira;
 - Argumentos: são valores recebidos junto com o evento;

- Evento: indica o nome de um sinal, mensagem ou notificação recebida pelo objeto e que torna a transição habilitada podendo, por consequência, ocorrer, levando o objeto a um novo estado.

Após ter analisado os conceitos do Diagrama de Estado, dar-se continuidade ao modelo MMORPG. A seguir, serão apresentados os principais Diagramas de Estado para se modelar um MMORPG com aprendizado social.

6.6.1. Diagrama de Estado Cria Conta

O Diagrama de Estado Cria Conta trata o Caso de Uso Criar Conta. A Figura 17 exhibe os estados, as condições, os eventos, as ações e respectivamente os argumentos da transição do objeto Conta.

Para um melhor entendimento, exigem-se algumas observações:

- A condição inicial do objeto apenas passa para o estado *Solicita_Cria_Conta* com a condição de o ambiente do jogo estar aberto;
- No momento que o evento *when(Janela_Conta_On)* ocorrer, passa-se para o estado *Espera_Dados_Conta*. Este estado é quando o jogador coloca os dados solicitados na classe conta;
- Em qualquer momento, o jogador pode solicitar o evento *when(Cancela_Conta)*. Quando este não deseja se cadastrar no ambiente do jogo, passando para o estado final *Cancela_Conta*, finalizar-se o objeto conta;
- O evento *when(Espera_Dados_Conta)* é onde ocorre o *looping* do objeto, que aguarda as informações até ativado o evento *when(Janela_Conta_Ok)*;
- No momento que o jogador inserir todos os dados cadastrais, ativa-se o evento *when(Janela_Conta_Ok) [Nome_Conta<>Nome_Conta_Cadastrado]*. Tal evento traz uma condição, qual seja, que o jogador apenas pode cadastrar a conta quando o nome da conta não estiver cadastrado;
- Depois de satisfeita a condição de transição do estado *Espera_Dados_Conta*, passa-se para o estado final *Conta_Finalizada*, ativa-se a abertura da janela para criar o personagem, e finaliza-se o objeto conta.

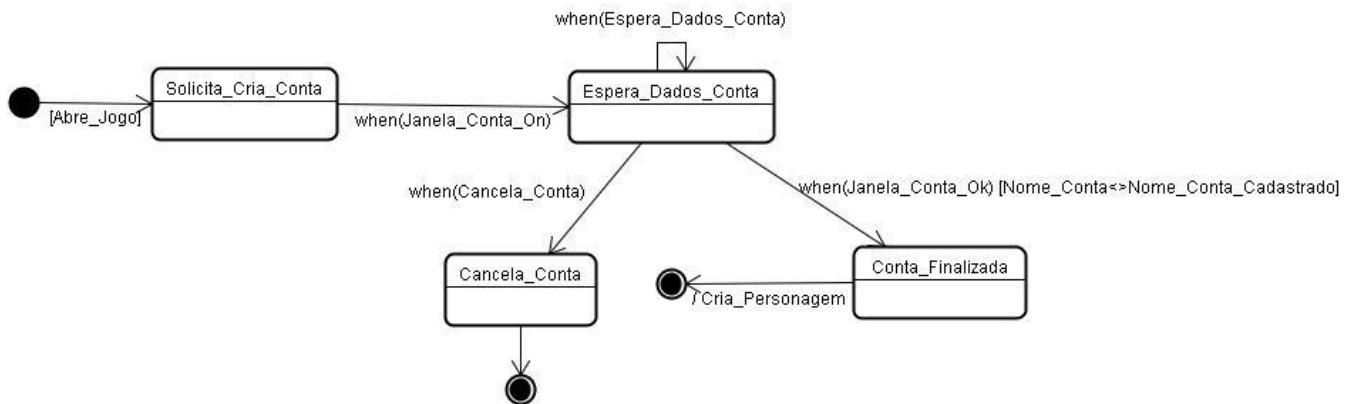


Figura 17: Diagrama Estado Cria Conta.

6.6.2. Diagrama de Estado Cria Personagem

O Diagrama de Estado Cria Personagem é semelhante ao da seção 6.6.1 e trata o Caso de Uso Cria Criar Personagem. A Figura 18 exibe os estados, as condições, os eventos e respectivamente argumentos da transição do objeto Personagem.

Para um melhor entendimento, exigem-se algumas observações:

- A condição inicial do objeto apenas passa para o estado *Solicita_Cria_Personagem* com a condição de que a conta seja criada;
- No momento que o evento *when(Janela_Personagem_On)* ocorrer, passa-se para o estado *Espera_Dados_Personagem*. Este estado é quando o jogador coloca os dados solicitados na classe personagem;
- O evento *when(Espera_Dados_Personagem)* é onde ocorre o *looping* do objeto, que aguarda as informações até ativado o evento *when(Janela_Personagem_Ok)*;
- Em qualquer momento, o jogador pode solicitar o evento *when(Cancela_Personagem)*. Quando o mesmo não deseja criar o personagem, passando para o estado final *Cancela_Conta*, finalizar-se o objeto conta;
- No momento que o jogador inserir todos os dados do personagem, ativa-se o evento *when(Janela_Personagem_Ok) [Nome_Personagem<=>Nome_Personagem_Cadastrado]*. Tal evento traz

uma condição, qual seja, que o jogador apenas pode cadastrar o personagem quando o nome do personagem não estiver cadastrado;

- Depois de satisfeita a condição de transição do estado *Espera_Dados_Personagem*, passa-se para o estado final *Personagem_Finalizado*, ativa-se a abertura da janela jogar, e finaliza-se o objeto conta.

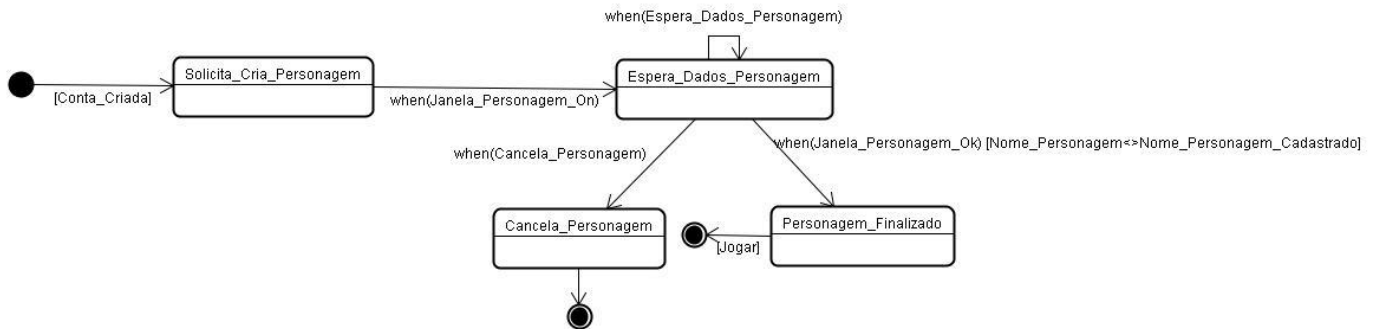


Figura 18: Diagrama Estado Cria Personagem.

6.6.3. Diagrama de Estado Caçar

O Diagrama de Estado Caçar trata o Caso de Caçar. A Figura 19 exibe os estados, as condições, os eventos e respectivamente os argumentos da transição do objeto Caçar.

Para um melhor entendimento, exigem-se algumas observações:

- O estado inicial *Combate_Iniciado* ocorre quando o jogador ou o monstro se aproxima do alvo, ativando o evento *when(Ataque)*. Satisfazendo a condição *[Primeiro_Hit]*, passa-se para o estado *Combate_Andamento*;
- No estado *Combate_Andamento*, existem três possibilidades de mudança de estado ou estabelece-se um círculo no mesmo estado, ativando o evento *when(Ataque)* sempre atendendo a condição *[Ataque_On]* de ao menos um dos confrontantes;
- A ativação da condição *[Cancelar_Combate]* é o momento em que o jogador entra em modo de fuga, e passa-se para o estado *Cancela_Combate*. Depois de satisfeita a condição *[Fuga_Personagem,]* a caça é cancelada e finaliza-se o objeto;

- A ativação da condição *[Morte_Monstro_Ok]* é o momento em que o jogador derrota o monstro e passa para o estado *Combate_Finalizado*. Ligando a ação */ Item_Drop_Monstro, Experiencia_Personagem, Money_Drop*, ocorrem ganhos de pontos de experiência, queda de itens do monstro e dinheiro, finalizando-se o objeto;
- A ocorrência da condição *[Morte_Personagem_Ok]* é o momento em que o monstro derrota o jogador, passando para o estado *Morte_Personagem*. Acionando-se */ Experiencia_Monstro*, o monstro garante seus pontos de experiência e, assim, finaliza-se o objeto caçar.

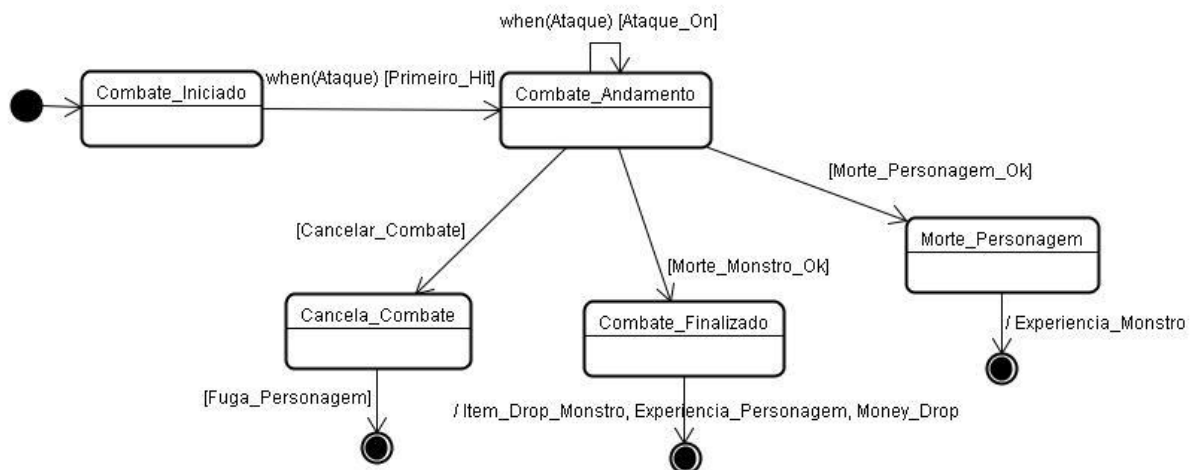


Figura 19: Diagrama Estado Caçar.

6.6.4. Diagrama de Estado Compra Item do Jogador

O Diagrama de Estado Compra Item do Jogador aborda o Caso de Uso Compra Item Jogador na plataforma, onde ocorre a transferência do comércio desejado. Este comércio pode ser inicializado no ambiente virtual, que será mencionado na seção 6.6.12. A Figura 20 exhibe os estados, as condições, os eventos e respectivamente os argumentos da transição do objeto Compra.

Para um melhor entendimento, exigem-se algumas observações:

- O estado inicial *Solicita_Trade* ocorre quando um jogador solicita a abertura do Trade;
- Passa-se para o estado *Trade_Aberto*, quando o evento *when(Trade_Aceito)* e a condição *[2_Jogadores]* forem respectivamente aceita e satisfeita –

lembrando-se que o Trade só funciona quando se tem dois jogadores (um solicita e o outro aceita);

- No estado *Trade_Aberto*, é onde o negociação final ocorre, satisfazendo a condição *[Trade_On]* e gerando um ciclo permanente do estado;
- Para passar ao estado *Cancela_Trade*, necessita-se que ao menos um dos jogadores acione o evento *when(Cancela_Trade)*, cancelando a venda e finalizando-se o objeto venda;
- Confirmado o negócio, ativa-se o evento *when(Janela_Trade_Ok)*. Satisfazendo o critério que os dois negociantes confirmarão, com a condição *[2_Jogadores_Ok]*, passa-se para o estado *Trade_Finalizado* e ativa-se a ação */ Item_Negociado_Ganho* para ambos os negociantes, finalizando o objeto.

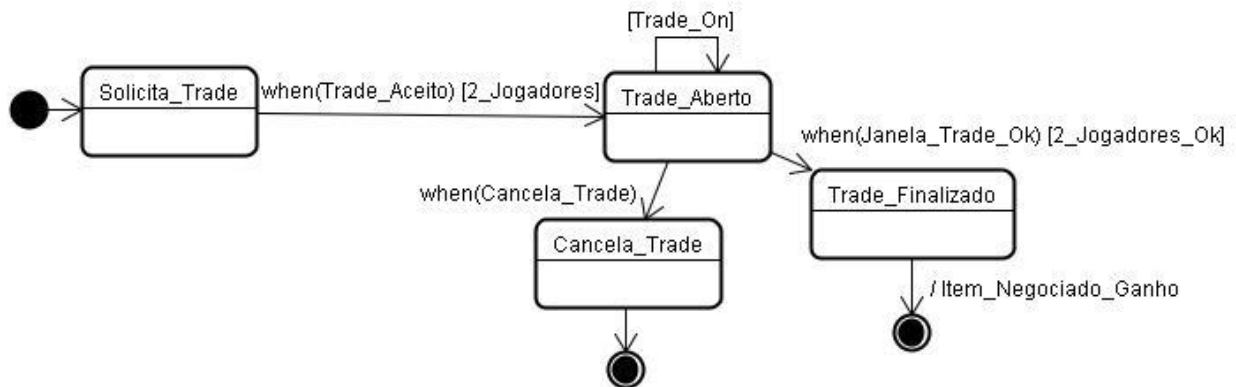


Figura 20: Diagrama Estado Compra Item Jogador.

6.6.5. Diagrama de Estado Compra Item do NPC

O Diagrama de Estado Compra Item do NPC aborda o Caso de Uso Compra Item NPC na plataforma. A Figura 21 exibe os estados, as condições, os eventos e respectivamente os argumentos da transição do objeto Compra.

Para um melhor entendimento, exigem-se algumas observações:

- Para o objeto mudar para o seu estado inicial *Solicita_Compra*, o jogador necessita ativar a condição *[Falar_NPC]*. Satisfeita esta condição, o objeto muda de estado;

- Automaticamente, ocorre a ação / *Janela_Compra_Aberta*, passando para o estado *Busca_Item*. Este estado é onde o jogador seleciona os itens a serem comprados, gerando o movimento cíclico da condição [*Janela_Compra_On*];
- Para passar ao estado *Cancela_Busca_Item*, necessita-se que o jogador acione o evento *when(Janela_Compra_Cancela)*, cancelando a compra e finalizando o objeto venda;
- Confirmada a compra o jogador, ativa-se o evento *when(Janela_Compra_Ok)*, satisfazendo o critério que o valor total da compra seja menor ou igual ao valor total de dinheiro do personagem. Pela condição [*Valor_Total_Item<=Money_Persongagem*], passa-se para o estado *Compra_Finalizada* e ativa-se a ação / *Item_Comprado_Ganho* para o jogador, finalizando o objeto;

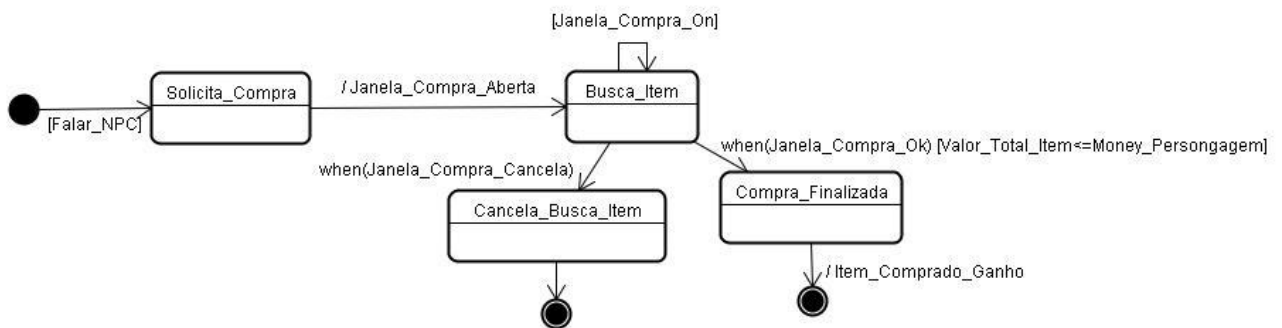


Figura 21: Diagrama Estado Compra Item NPC.

6.6.6. Diagrama de Estado Compra Skill

O Diagrama de Estado Compra Skill aborda o Caso de Uso Compra Skill na plataforma, que é semelhante ao estado da seção 6.6.5. A Figura 22 exhibe os estados, as condições, os eventos e respectivamente os argumentos da transição do objeto Compra.

Para um melhor entendimento, exigem-se algumas observações:

- Para o objeto mudar para o seu estado inicial *Solicita_Compra_Skill*, o jogador necessita ativar a condição [*Falar_NPC*]. Satisfeita esta condição, o objeto muda de estado;

- Automaticamente, ocorre a ação / *Janela_Compra_Aberta*, passando para o estado *Busca_Skill*. Este estado é onde o jogador seleciona as *Skill's* a serem compradas, gerando o movimento cíclico da condição [*Janela_Compra_On*];
- Para passar ao estado *Cancela_Busca_Skill*, necessita-se que o jogadores acione o evento *when(Janela_Compra_Cancela)*, cancelando a compra e finalizando o objeto venda;
- Confirmada a compra o jogador, ativa-se o evento *when(Janela_Compra_Ok)*, satisfazendo o critério que o valor total da compra seja menor ou igual ao valor total de dinheiro do personagem. Pela condição [*Valor_Total_Skill<=Money_Persongagem*], passa-se para o estado *Compra_Finalizada* e ativa-se a ação / *Skill_Comprado_Ganho* para o jogador, finalizando o objeto;

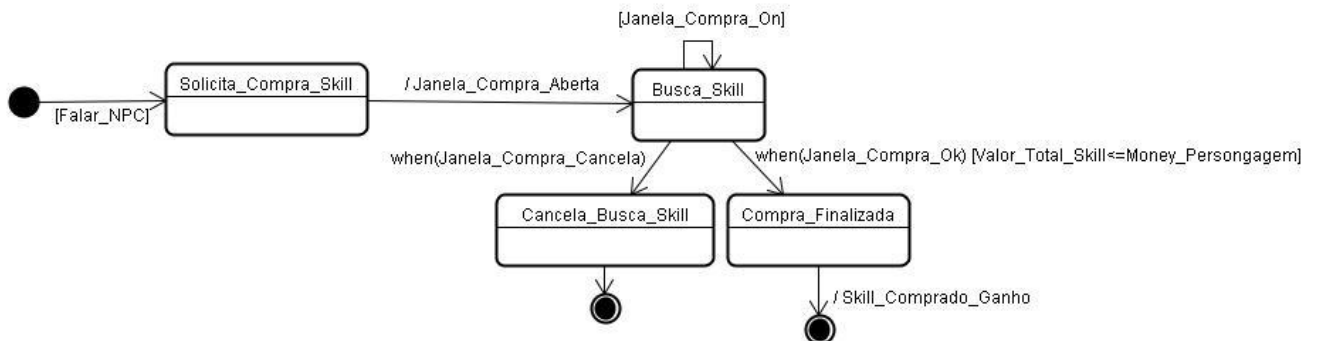


Figura 22: Diagrama Estado Compra Skill.

6.6.7. Diagrama de Estado Cria Guild

O Diagrama de Estado Cria Guild aborda o Caso de Uso Cria Guild, que é semelhante ao estado da seção 6.6.5. A Figura 23 exhibe os estados, as condições, os eventos e respectivamente os argumentos da transição do objeto *Guild*. Para um melhor entendimento, exigem-se algumas observações:

- Para o objeto mudar para o seu estado inicial *Solicita_Guild*, o jogador necessita ativar a condição [*Falar_NPC*]. Satisfeita esta condição o objeto, muda-se de estado;

- Satisfazendo a condição de que o personagem deve ter o *level* maior ou igual ao *level* permitido para formar uma Guild, $[Lv_Personagem \geq Lv_Permitido]$, automaticamente ocorre a ação / *Janela_Guild_On*, passando para o estado *Espera_Dados_Guild*. Este estado é onde o jogador insere todos os dados e integrantes da Guild, gerando o movimento cíclico da condição $[Dados_Guild]$;
- Para passar ao estado *Cancela_Cria_Guild*, necessita-se que o jogadores acione o evento $when(Cancela_Guild)$, cancelando a criação da Guild e finaliza-se o objeto Guild;
- Confirmada a criação da Guild, o jogador ativa o evento $when(Janela_Guild_Ok)$. Este evento possui a condição de guarda, $[Dados_Guild == Dados_Ok \& \& Nome_Guild \neq Nome_Guild_Cadastrado]$, que aborda os critérios de todos os dados da Guild, onde o criador e membros estão qualificados e não deverá existir nenhuma Guild com o nome igual. Depois de satisfeita a condição, passa-se para o estado final *Guild_Finaliza*, ativando a ação / *Personagem_ADD_Guild*, que cria e adiciona todos os membros do grupo, finalizando o objeto.

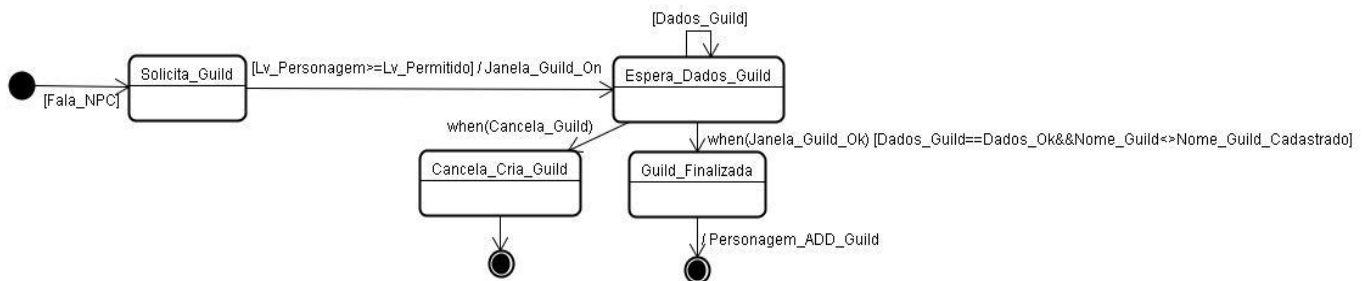


Figura 23: Diagrama Estado Cria Guild.

6.6.8. Diagrama de Estado Cria Party

O Diagrama de Estado Cria Party Diagrama aborda o Caso de Uso Cria Party. A Figura 24 exibe os estados, as condições, os eventos e respectivamente os argumentos da transição do objeto Party.

Para um melhor entendimento, exigem-se algumas observações:

- O *Solicita_Membro_Party* trata-se de um envio de convite a outro jogador para que permita a formação da Party;
- Quando o jogador confirmar o evento convite de formação da Party e satisfazer a condição *level* do convidado de ser maior ou igual ao *level* permitido para formação de party, passa-se para o estado *Party_On*;
- No estado *Party_On*, existe um ciclo de mesmo estado que é ativado pela condição $[Membros_Party \geq 1]$, onde a Party, para continuar, deve possuir ao menos dois membros (criador não é contado);
- Satisfeita a condição $[Membros_Party < 1]$, qual seja, os membros da Party deve ser zero, ativa-se a ação / *Cancela_Party* e passa-se para o estado *Cancela_Party*, finalizando o objeto.

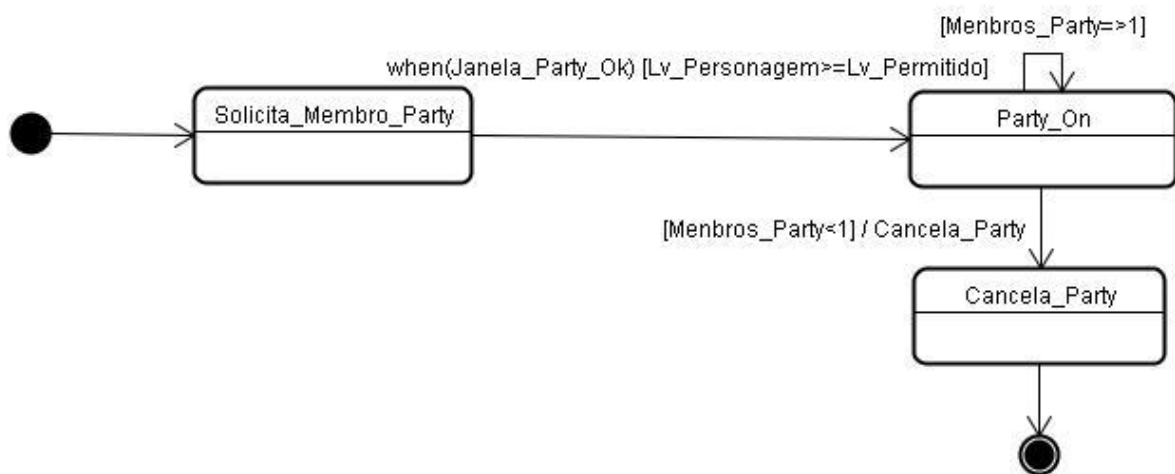


Figura 24: Diagrama Estado Cria Party.

6.6.9. Diagrama de Estado Quest

O Diagrama de Estado Quest aborda o Caso de Uso Quest. A Figura 25 exibe os estados, as condições, os eventos e respectivamente os argumentos da transição do objeto Quest.

Para um melhor entendimento, exigem-se algumas observações:

- Para o objeto mudar para o seu estado inicial *Solicita_Quest_NPC*, o jogador necessita ativar a condição $[Falar_NPC]$. Satisfeita a condição, o objeto muda de estado;

- No estado *Solicita_Quest_NPC*, ocorre a lista de *Quest's* que são habilitadas ao personagem. Optando a *Quest* desejada, usa-se o evento *when(Escolhe_Quest)*, passando para o estado *Quest_Incompleta*. No mesmo estado, existem também a opção de escolher outras *Quest's*, criando um ciclo com o evento *when(Escolhe_Nova_Quest)*. A janela que solicita a *Quest* continuará aberta pela condição [*Janela_Quest_On*]. Caso seja solicitada a(s) *Quest(s)* desejada, há a possibilidade de se cancelar a janela. Solicitando o evento *when(Cancela_Janela_Quest)*, o personagem continuará com as *Quest's* requisitadas no estado *Quest_Incompleta*, passando o objeto janela para o estado final *Cancela_Janela*;
- No estado *Quest_Incompleta*, o jogador pode optar no cancelamento da *Quest*, usando o evento *when(Cancelar_Quest)*, quando a *Quest* selecionada para cancelamento passa para o estado *Cancela_Quest*. No entanto, quando o jogador obtêm êxito ao cumprir o objetivo da *Quest*, acionado a condição [*Ter_Item or Kill_Monstro*], a referida *Quest* passa para o estado *Quest_Em_Andamento*;
- Para o estado *Quest_Em_Andamento*, existe também a possibilidade do cancelamento da *Quest*, caso o jogador não deseje completá-la, selecionando o mesmo o evento *when(Cancelar_Quest)*. Caso o jogador deseje completar a *Quest*, fala com o NPC da *Quest*, satisfazendo a condição [*Fala_NPC*], entregando o objetivo da *Quest*, e utilizando o evento *when(Entrega_Item or Pova_Kill_Monstro)*. A *Quest* passa para o estado *Quest_Completa*, acionando / *Ganha_Item or Skill or Experiencia* e finalizando o objeto.

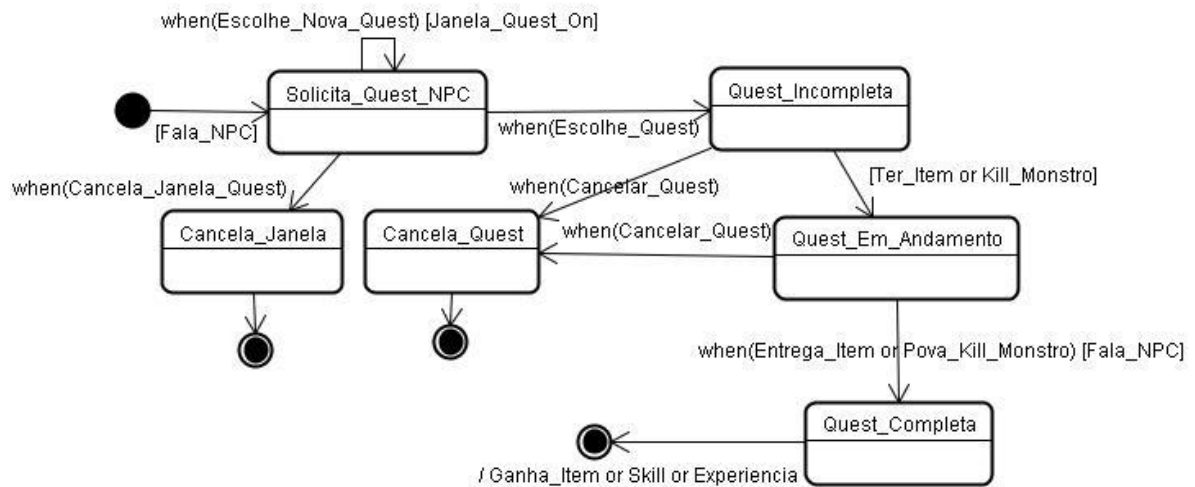


Figura 25: Diagrama Estado Quest.

6.6.10. Diagrama de Estado Vende Item para NPC

O Diagrama de Estado Vende Item para NPC aborda o Caso de Uso Vende Item NPC na plataforma, que é semelhante ao da seção 6.6.5. A Figura 26 exibe os estados, as condições, os eventos e respectivamente argumentos da transição do objeto Venda. Para um melhor entendimento, exigem-se algumas observações:

- Para o objeto mudar para o seu estado inicial *Solicita_Venda*, o jogador necessita ativar a condição *[Falar_NPC]*. Satisfeita esta condição, o objeto muda de estado;
- Automaticamente, ocorre a ação */ Janela_Venda_Aberta*, passando para o estado *Busca_Item_Venda*. Este estado é onde o jogador seleciona os itens a serem vendidos, gerando o movimento cíclico da condição *[Janela_Venda_On]*;
- Para passar ao estado *Cancela_Item_Venda*, necessita-se que o jogadores acione o evento *when(Janela_Venda_Cancela)*, cancelando a venda e finaliza-se o objeto venda;
- Confirmada a venda, o jogador ativa o evento *when(Janela_Venda_Ok)* e passa para o estado *Venda_Finalizada*, ativando a ação */ Item_Vendido && Money_Ganho* para o jogador, que finaliza o objeto.

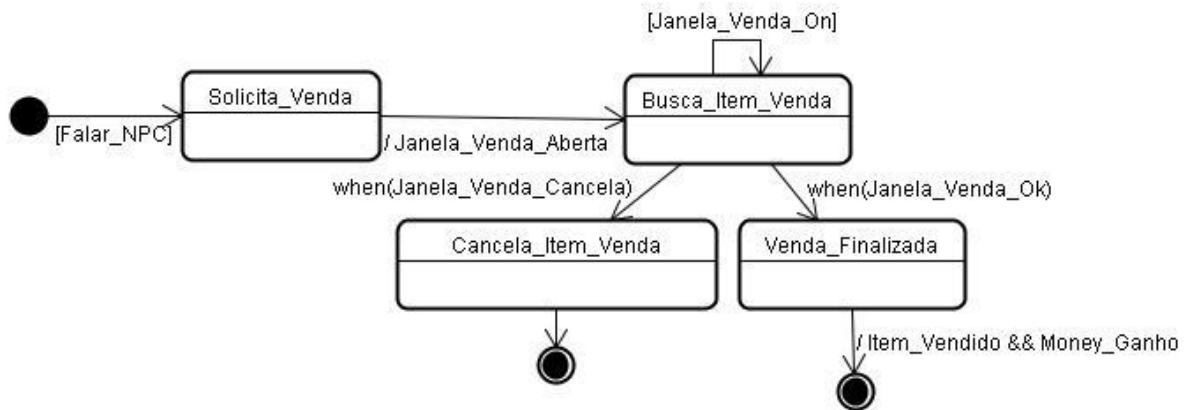


Figura 26: Diagrama Estado Venda Item NPC.

6.6.11. Diagrama de Estado Fórum

O Diagrama de Estado Fórum aborda o Caso de Uso Ler, Responde e Cria Fórum na plataforma do ambiente virtual. A Figura 27 exibe os estados, as condições, os eventos e respectivamente os argumentos da transição do objeto Fórum.

Para um melhor entendimento, exigem-se algumas observações:

- Para o objeto mudar para o seu estado inicial *Solicita_Cria_Forum*, o jogador necessita acessar a plataforma do ambiente virtual para ativar a condição *[Acessa_AVA]*. Satisfeita esta condição, o objeto muda de estado;
- Automaticamente, ocorre a ação */ Forum_Criado*, passando para o estado *Forum_Aberto*. Este estado é onde o jogador poderá responder os fóruns abertos, utilizando o evento *when(Responde_Forum)*, ou visualizar os fóruns com o evento *when(Ler_Forum)*;
- O criador do fórum poderá selecionar o evento *when(Cancela_Forum)*, para cancelar o fórum, passando-se para o estado *Cancela_Forum*. No estado *Cancela_Forum*, ocorre a ação */ Forum_Excluido*, finalizando-se o objeto fórum.

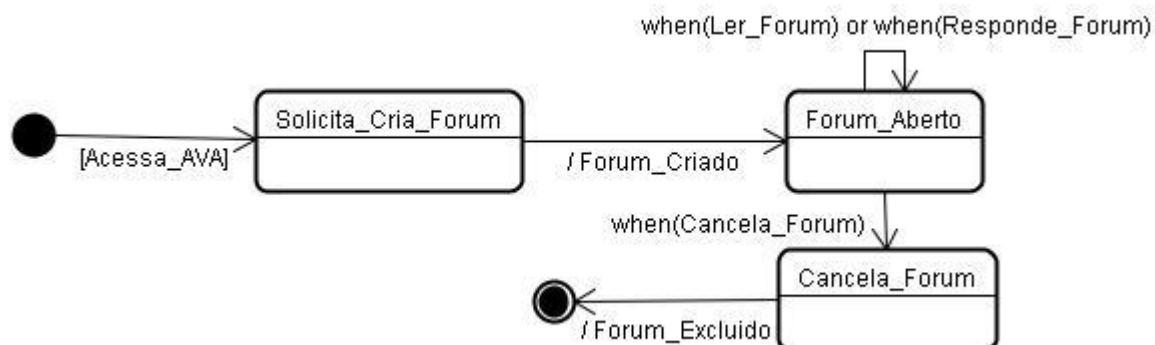


Figura 27: Diagrama Estado Fórum.

6.6.12. Diagrama de Estado Itens à Venda

O Diagrama de Estado Itens à Venda Diagrama aborda o Caso de Uso Ler, Responde e Cria Itens a Venda na plataforma do ambiente virtual. Essa é a parte inicial do modelo de negócio abordado para o MMORPG, que é finalizado pelo Caso de Uso Venda Item Jogador, visualizado na seção 6.6.4. A Figura 28 exhibe os estados, as condições, os eventos e respectivamente os argumentos da transição do objeto Venda.

Para um melhor entendimento, exigem-se algumas observações:

- Para o objeto mudar para o seu estado inicial *Solicita_Cria_Venda_Item*, o jogador necessita acessar a plataforma do ambiente virtual para ativar a condição *[Acessa_AVA]*. Satisfeita esta condição, o objeto muda de estado;
- Automaticamente, ocorre a ação */ Item_Venda_Criado*, passando para o estado *Venda_Item_Aberto*. Este estado é onde o jogador poderá responder as propostas das vendas abertas, utilizando o evento *when(Responde_Item_Venda)*, ou visualizar os itens a venda com o evento *when(Ler_Item_Venda)*;
- O criador do item a venda poderá selecionar o evento *when(Cancela_Item_Venda)* para cancelar a venda do item, passando-se para o estado *Cancela_Item_Venda*. No estado *Cancela_Item_Venda*, ocorre a ação */ Item_Venda_Excluido*, finalizando-se o objeto fórum;
- No momento em que o criador do item a venda fechar negócio, seleciona-se o evento *when(Venda_Fim)*. Nesse estágio, o objeto passa para o estado *Venda_Finalizado*, não permitindo mais receber outra proposta ou

cancelamento de venda, acionando-se / *Venda_Fechada* e finalizando o objeto.

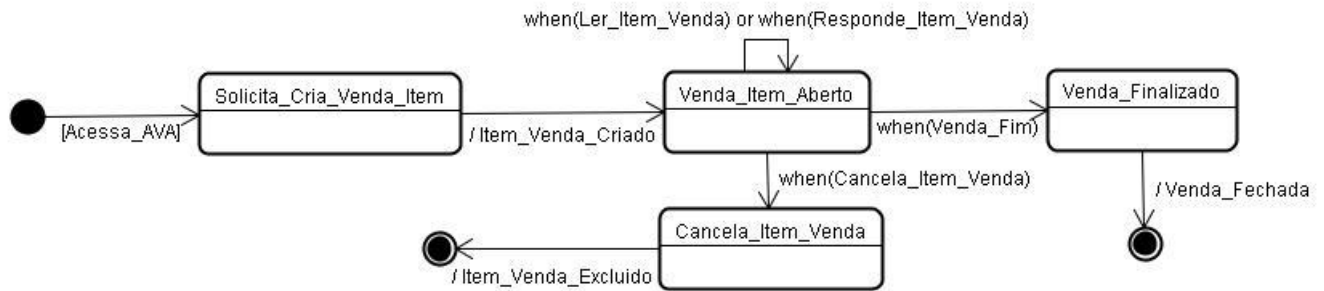


Figura 28: Diagrama Estado Itens a Venda.

6.6.13. Diagrama de Estado Leilão

O Diagrama de Estado Leilão Diagrama aborda o Caso de Uso Ler, Responde e Cria Leilão na plataforma do ambiente virtual. Essa é a parte inicial do modelo de negócio abordado para o MMORPG, que semelhante ao caso mencionado na seção 6.6.12, diferenciando-se apenas quanto ao tipo de modalidade de venda (leilão tem um curto tempo de propostas). A Figura 29 exhibe os estados, as condições, os eventos e respectivamente os argumentos da transição do objeto Leilão.

Para um melhor entendimento, exigem-se algumas observações:

- Para o objeto mudar para o seu estado inicial *Solicita_Cria_Leilao*, o jogador necessita acessar a plataforma do ambiente virtual para ativar a condição *[Acessa_AVA]*. Satisfeita esta condição, o objeto muda de estado;
- Automaticamente, ocorre a ação */ Leilao_Criado*, passando para o estado *Leilao_Aberto*. Este estado é onde o jogador poderá responder as propostas de valor de compra, utilizando o evento *when(Responde_Leilao)*, ou visualizar os leilões com o evento *when(Ler_Leilao)*;
- O criador do leilão poderá selecionar o evento *when(Cancela_Leilao)* para cancelar o leilão, passando-se para o estado *Cancela_Leilao*. No estado *Cancela_Leilao*, ocorre a ação */ Leilao_Excluido* e finaliza-se o objeto fórum;
- Quando esgotar o tempo de duração do leilão, satisfaz-se a condição *[Tempo_Esgotado]* e aciona-se */ Leilao_Fim*, mudando o objeto para o

estado *Leilao_Finalizado*. Neste estado, será exibido o ganhador do leilão e será ativada ação / *Leilao_Fechado*, finalizando o objeto.

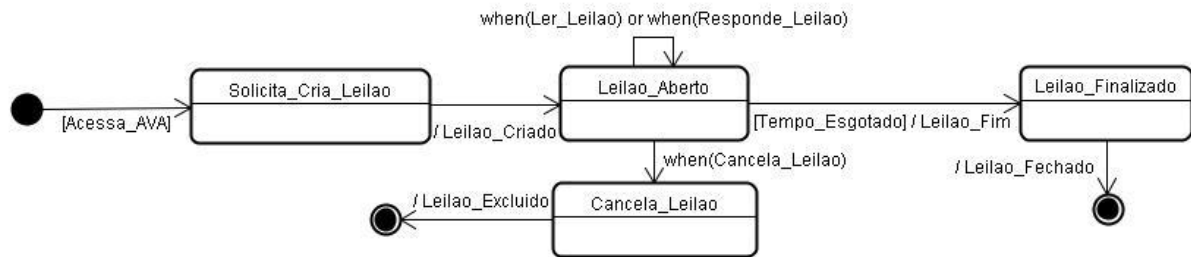


Figura 29: Diagrama Estado Leilão.

6.7. Subsistemas

Uma classe pode ser dividida em mais de uma classe ou transformada em um subsistema. Por outro lado, várias classes muito acopladas podem ser unificadas em uma única classe ou agrupadas em um subsistema (BOOCH, 2010).

As classes mais complexas geralmente são quebradas em outras classes mais simples ou transformadas em subsistemas. Subsistemas oferecem serviços bem definidos, possuindo uma interface pública estável para acesso a esses serviços. Na Figura 30, pode-se observar exemplo de um subsistema conta personagem, com sua interface pública *I_Dados_Personagem*, que contém as classes: *Personagem_Jogador_Class*, *Sexo_Class* e *Raça_Class*, criando, assim, um sistema independente, que realiza todas as suas funções atribuída ao subsistema.

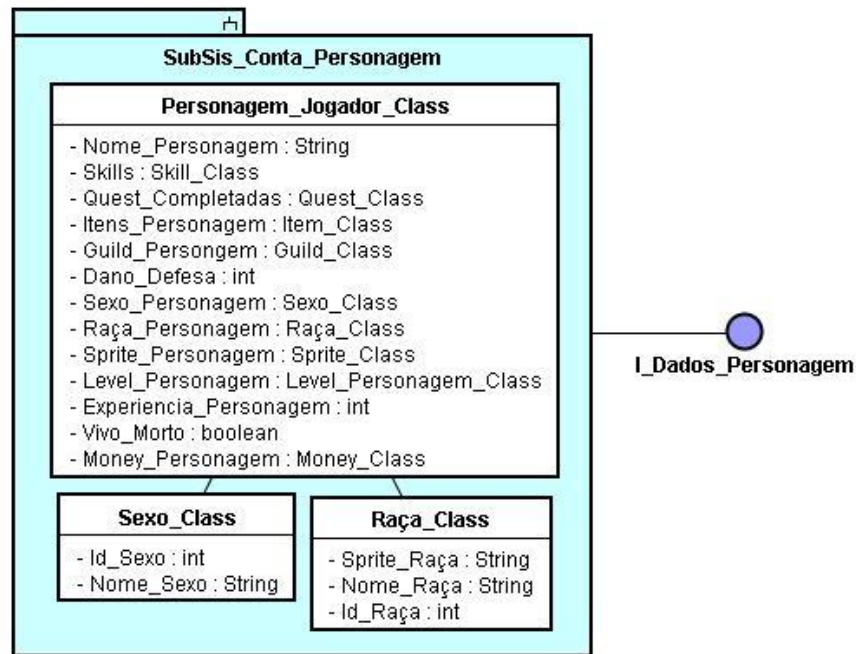


Figura 30: Exemplo de Subsistema.

Classes de fronteira (*I_Dados_Personagem*) representam a interface de comunicação entre a aplicação (ou um subsistema da aplicação) e os atores, que podem ser usuários ou outros sistemas. As classes de fronteira normalmente dão origem a interfaces, que definem os serviços oferecidos pela aplicação. Futuramente, na seção 6.8, poder-se-á visualizar tais classes fronteiras.

6.7.1. Diagrama de Subsistemas do ambiente

Os Subsistemas do ambiente do ambiente foram modelados agrupando os objetos em pequenos subsistemas, onde permite uma “certa” funcionalidade independente dos métodos, que estão contidos em cada subsistema. No entanto, existe apenas a restrição de dependência (representada pelo símbolo: `----->`, quem está do lado direito está cedendo alguma informação para quem vem buscar do sentido esquerdo), como exemplo: o *SubSis_Conta_Personagem* solicita à classe fronteira *I_Dados_Sprite* o *Sprite*, o qual está ligado ao objeto personagem; o *Sprite* é cedido pela classe *Sprite_Class*, que contém o método `+ Get_Sprite() : Sprite_Class`. Da mesma forma ocorre em todos os relacionamentos dos subsistemas do ambiente.

Antes de analisar a Figura 31, existem algumas observações a serem mencionadas, abordando todos os subsistemas e suas respectivas dependências:

- Os métodos contidos nas classes dos subsistemas foram ocultados, a fim de que possa solucionar o problema de espaçamento de página. Caso ocorra alguma eventual dúvida, deve-se checar a seção 6.5, onde foram abordados todos os diagramas de classe;
- Não foram exibidos todos os subsistemas e todas as classes que possam compor um MMORPG, pelo fato de que formaria um diagrama complexo para o entendimento. Em contra proposta do autor do respectivo modelo, abordou-se apenas os principais casos para atingir a meta do modelo;

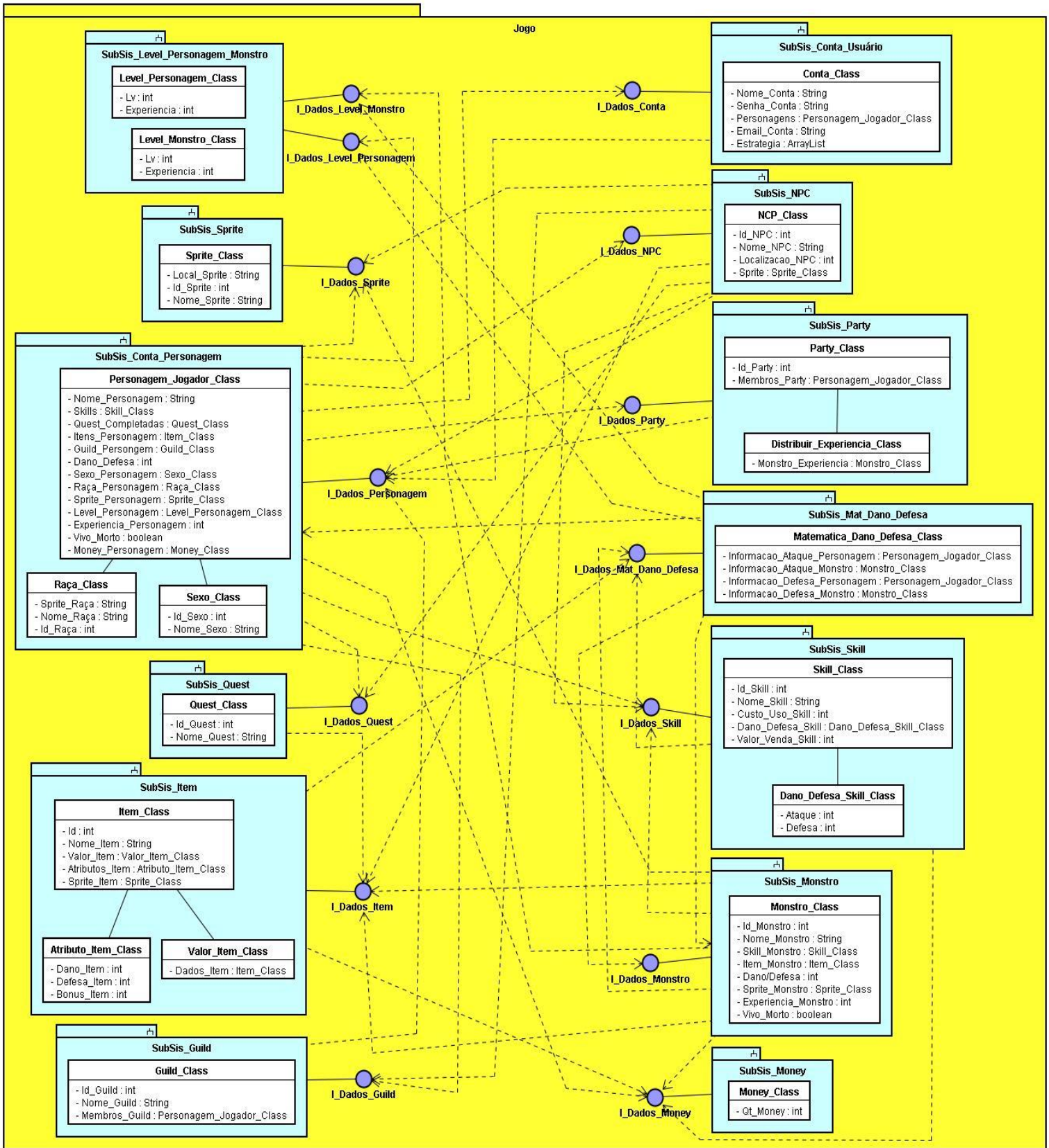


Figura 31: Diagrama Subsistemas do Ambiente.

6.7.2. Diagrama de Subsistemas do ambiente virtual

Os Subsistemas do ambiente do ambiente virtual foi modelado agrupando os objetos em pequenos subsistemas, seguindo o mesmo modelo da seção 6.7.1. O *SubSis_Conta_Forum* consegue trabalhar independente, já os subsistemas *SubSis_Forum* e *SubSis_Venda_Item* dependem das informações das contas jogadores cedidas pelo método `+ Get_Reputacao() : void` e `+ Get_Conta_Forum() : Conta_Class` do subsistema *SubSis_Conta_Forum* para funcionarem.

Antes de analisar a Figura 32, há algumas observações a serem mencionadas, abordando todos os subsistemas e suas respectivas dependências:

- Os métodos contidos nas classes dos subsistemas foram ocultados, a fim de que possa solucionar o problema de espaçamento de página. Caso ocorra alguma eventual dúvida, deve-se checar a seção 6.5 onde foram abordados todos os diagramas de classe;
- Não foram exibidos todos os subsistemas e todas as classes que possam compor um MMORPG, pelo fato de que formaria um diagrama complexo para o entendimento. Em contra proposta do autor do respectivo modelo, abordou-se apenas os principais casos para atingir a meta do modelo;
- O subsistema *SubSis_Venda_Item* tem como classe principal *Itens_Venda_Class*, pelo fato de que a classe *Leilao_Class* é uma venda de modalidade diferente.

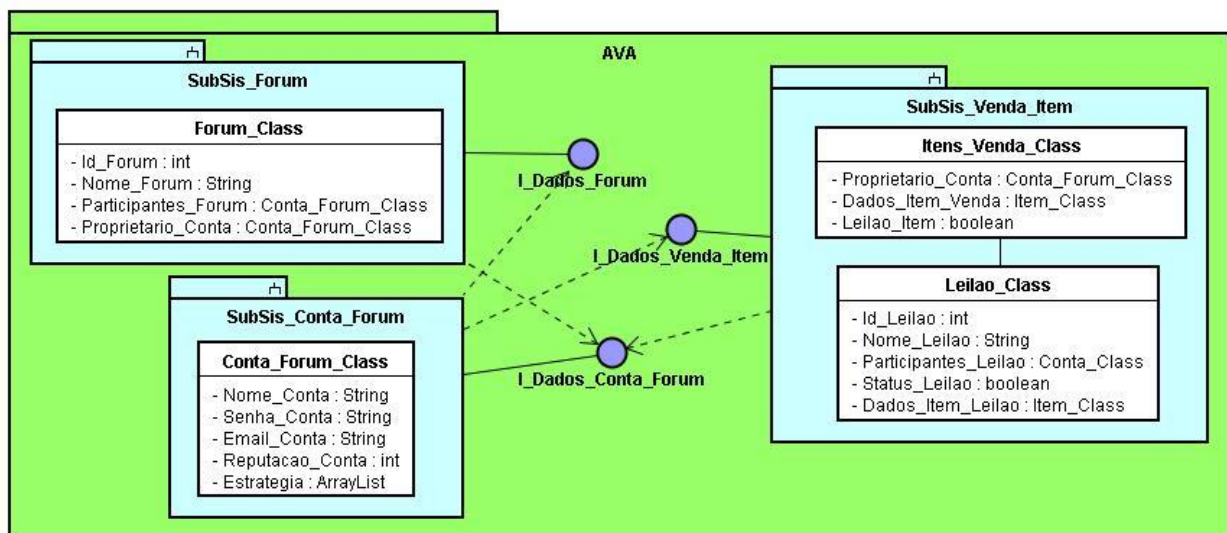


Figura 32: Diagrama Subsistemas do ambiente virtual.

6.8. Diagrama De Classes Com Interface

A proposta do respectivo capítulo assemelha-se ao do seção 6.5, com o acréscimo, no entanto, da classe estereotipada como interface, que tem como função assegurar uma melhor comunicação interna entre as classes, exibindo apenas os métodos que outras classes possam solicitar, conforme demonstrado no diagrama de classe.

Seguindo a proposta do padrão UML, o círculo ligado a uma classe representa a interface, ou seja, outra classe estereotipada que já foi exibida em outro diagrama. A seta tracejada fechada (-----▷) aborda a classe apontadora, que é proprietária da classe interface. Já, a seta aberta tracejada (----->) aponta para exclusivamente a uma classe interface e trabalha da seguinte forma: a classe apontadora solicita informação(ões) pertencentes à proprietária da classe interface.

A seguir será apresentado, os principais Diagramas de Classe com Interface para se modelar um MMORPG com aprendizado social. Observa-se que, devido por motivos de espaço, os métodos da classe *Personagem_Jogador_Class* foram ocultados. Então, caso exista alguma dúvida em relação aos métodos, deve-se checar o *APÊNDICE A – Diagramas de Classes*, que contém a Figura 35.

6.8.1. Diagrama de Classe com Interface Cria Conta e Cria Personagem Jogador

O Diagrama de Classe com Interface Cria Conta e Cria Personagem Jogador tem funcionamento semelhante ao do diagrama da seção 6.5.1, com o acréscimo das classes interfaces *Visualiza_Personagem_Jogador*, *Visualiza_Raça*, *Visualiza_Sexo* e *Visualiza_Conta*, que disponibilizam os dados da classe proprietária para o funcionamento dos métodos da classe solicitante. Para uma melhor compreensão do diagrama, remete-se ao *APÊNDICE B – Diagrama de Classes com Interface*, Figura 44.

6.8.2. Diagrama de Classe com Interface Caçar

O Diagrama de Classe com Interface Caçar tem funcionamento parecido com o do diagrama da seção 6.5.2, com acréscimo das classes interfaces *Visualiza_Matematica_Dano_Defesa*, *Visualiza_Personagem_Jogador*, *Visualiza_Monstro*, *Visualiza_Level_Monstro*, *Visualiza_Item* e

Visualiza_Level_Personagem, que atuam disponibilizando os dados da classe proprietária para o funcionamento dos métodos da classe solicitante. Para uma melhor compreensão do diagrama, remete-se ao *APÊNDICE B – Diagrama de Classes com Interface*, Figura 45.

6.8.3. Diagrama de Classe com Interface Adquirir Quest

O Diagrama de Classe com Interface Adquirir Quest tem funcionamento semelhante ao do diagrama da seção 6.5.3, com acréscimo das classes interfaces *Visualiza_NPC*, *Visualiza_Personagem_Jogador*, *Visualiza_Sprite*, *Visualiza_Quest*, *Visualiza_Item*, *Visualiza_Valor_Item* e *Visualiza_Atributo_Item*, que disponibilizam os dados da classe proprietária para o funcionamento dos métodos da classe solicitante. Para uma melhor compreensão do diagrama, remete-se ao *APÊNDICE B – Diagrama de Classes com Interface*, Figura 46.

6.8.4. Diagrama de Classe com Interface Compra Item do Jogador

O Diagrama de Classe com Interface Compra Item do Jogador tem funcionamento semelhante ao do diagrama da seção 6.5.4, com acréscimo das classes interfaces *Visualiza_Personagem_Jogador*, *Visualiza_Item* e *Visualiza_Money*, que trabalham disponibilizando os dados da classe proprietária para o funcionamento dos métodos da classe solicitante. Para uma melhor compreensão do diagrama, remete-se ao *APÊNDICE B – Diagrama de Classes com Interface*, Figura 47.

6.8.5. Diagrama de Classe com Interface Compra ou Vende Item do NPC

O Diagrama de Classe com Interface Compra ou Vende Item do NPC tem funcionamento semelhante ao diagrama da seção 6.5.5, com acréscimo das classes interfaces *Visualiza_NPC*, *Visualiza_Personagem_Jogador*, *Visualiza_Item*, *Visualiza_Money*, *Visualiza_Atributo_Item* e *Visualiza_Valor_Item*, que disponibilizam os dados da classe proprietária para o funcionamento dos métodos da classe solicitante. Para uma melhor compreensão do diagrama, remete-se ao *APÊNDICE B – Diagrama de Classes com Interface*, Figura 48.

6.8.6. Diagrama de Classe com Interface Compra Skill

O Diagrama de Classe com Interface Compra Skill tem funcionamento semelhante ao do diagrama da seção 6.5.6, com acréscimo das classes interfaces *Visualiza_NPC*, *Visualiza_Personagem_Jogador*, *Visualiza_Dano_Defesa_Skill*, *Visualiza_Money* e *Visualiza_Skille*, que atuam fornecendo dados da classe proprietária para o funcionamento dos métodos da classe solicitante. Para uma melhor compreensão do diagrama, remete-se ao *APÊNDICE B – Diagrama de Classes com Interface*, Figura 49.

6.8.7. Diagrama de Classe com Interface Cria Guild

O Diagrama de Classe com Interface Cria Guild tem funcionamento semelhante ao do diagrama da seção 6.5.7, com acréscimo das classes interfaces *Visualiza_NPC*, *Visualiza_Personagem_Jogador* e *Visualiza_Guild*, que transferem dados da classe proprietária para o funcionamento dos métodos da classe solicitante. Para uma melhor compreensão do diagrama, remete-se ao *APÊNDICE B – Diagrama de Classes com Interface*, Figura 50.

6.8.8. Diagrama de Classe com Interface Cria Party

O Diagrama de Classe com Interface Cria Party tem funcionamento semelhante ao do diagrama da seção 6.5.8, com acréscimo das classes interfaces *Visualiza_Party*, *Visualiza_Personagem_Jogador*, *Visualiza_Level_Personagem*, *Visualiza_Distribuir_Experiencia*, *Visualiza_Monstro* e *Visualiza_Level_Monstro*, que trabalham disponibilizando os dados da classe proprietária para o funcionamento dos métodos da classe solicitante. Para uma melhor compreensão do diagrama, remete-se ao *APÊNDICE B – Diagrama de Classes com Interface*, Figura 51.

6.8.9. Diagrama de Classe com Interface Fóruns, Leilão e Itens à Venda

O Diagrama de Classe com Interface Fóruns, Leilão e Itens à Venda tem funcionamento semelhante ao do diagrama da seção 6.5.9, com acréscimo das classes interfaces *Visualiza_NPC*, *Visualiza_Personagem_Jogador*, *Visualiza_Dano_Defesa_Skill*, *Visualiza_Money* e *Visualiza_Skille*, que atuam

fornecendo dados da classe proprietária para o funcionamento dos métodos da classe solicitante. Para uma melhor compreensão do diagrama, remete-se ao *APÊNDICE B – Diagrama de Classes com Interface*, Figura 52.

6.9. Diagrama de Componentes com Implantação

Segundo Booch (2010), um componente é qualquer arquivo que contenha uma parte necessária à consecução de um software. O componente pode ser uma página HTML, um arquivo txt, dll, jar e etc. A Figura 33 exemplifica como se é representado um componente na UML. Para se armazenar os componentes, que possibilitam o funcionamento, existem os nós. Esses nós são entidades que representam *hardware* ou *software*, que recebem os componentes, tal como se verifica na Figura 34.

Componentes são tipos, mas apenas componentes executáveis podem ter instâncias. Um diagrama de componente mostra apenas componentes como tipos. Para mostrar instâncias de componentes, deve ser usado um diagrama de implantação, onde as instâncias executáveis são alocadas em nós.



Figura 33: Exemplo de Componente.



Figura 34: Exemplo de Nó.

No entanto, o diagrama de componente e o de implantação são diagramas que mostram o sistema por um lado funcional, expondo as relações entre seus componentes e a organização de seus módulos durante sua execução.

Os Componentes podem definir interfaces que são visíveis para outros componentes. Uma interface é mostrada com um círculo no interior do componente. Dependências entre componentes são semelhantes ao funcionamento do Diagrama de Classe com Interface. Os componentes podem apontar para a interface de outros

componentes para permitir a busca de informações necessária para o funcionamento, atentando-se que a seta de decência é tracejada.

Na implantação, mostra-se a arquitetura física do hardware e do software no sistema, os atuais computadores e periféricos, juntamente com as conexões que eles estabelecem entre si, podendo-se mostrar também os tipos de conexões entre esses computadores e periféricos.

A seguir, será apresentado os diagramas que envolvem as plataformas.

6.9.1. Diagrama de Componentes com Implantação do jogo

O Diagrama de Componentes com Implantação do jogo que exhibe todos os componentes da plataforma e seus nós, possibilitando a sua implantação.

Para um melhor entendimento do diagrama exposto no *APÊNDICE C – Diagrama de Componentes com Implantação* (Figura 53), apresentam-se as seguintes observações:

- O nó *:Estacao_Jogador_1* representa um jogador em casa. Porém, o nó *:Estacao_Jogador_N* é uma possibilidade que N jogadores podem acessar a plataforma. Para acessar a plataforma, utiliza-se a Internet, demonstrado no ligamento dos nós *:Estacao_Jogador_1* e *:Form_Jogador*;
- No *:Form_Jogador*, está toda a interface gráfica da plataforma e todos os componentes gráficos, comunicando-se com o nó principal da plataforma *:Servidor_Aplicacao_Jogo* (*Internet*, *WLAN*²⁴, *LAN*²⁵). No momento em que a interface necessita de informações para o funcionamento do ambiente, ocorre o fator dependência, ligando o componente à interface de um outro componente, ocorrendo o mesmo quando um componente do *:Servidor_Aplicacao_Jogo* necessitar de outra(s) informação(ões) para realizar a sua atividade solicitada pela interface gráfica;
- Para realizar o salvamento das informações da plataforma, existe um nó chamado *:Banco_de_Dados_Jogo*, que contém dois componentes: *Persistencia* (responsável pela validação de todas as informações que será

²⁴ Wireless Local Area Network.

²⁵ Local Network.

salva no Banco de Dados) e *SGBD* (Banco de Dados). A comunicação ente nós (:*Servidor_Aplicacao_Jogo*, :*Banco_de_Dados_Jogo*) é dada por ODBC²⁶;

- Por medida de segurança, existe o nó :*Backup*, no qual salvam-se todas as informações do Banco de Dados jogo e do Servidor de Aplicação do jogo;

6.9.2. Diagrama de Componentes com Implantação do ambiente virtual

O Diagrama de Componentes com Implantação do ambiente virtual demonstra todos os componentes da plataforma do ambiente virtual e seus nós, que permitem a sua implantação.

Para melhor compreensão do diagrama exposto no *APÊNDICE C – Diagrama de Componentes com Implantação* (Figura 54), apresentam-se as seguintes observações:

- Os nós :*Estacao_Jogador_1* e :*Estacao_Jogador_N* funcionam da mesma forma da seção 6.9.1;
- Os :*Form_AVA*, :*Servidor_Aplicacao_AVA* e :*Banco_de_Dados_AVA* assemelham-se ao funcionamento, respectivamente, dos :*Form_Jogador*, :*Servidor_Aplicacao_Jogo* e :*Banco_de_Dados_Jogo* da seção 6.9.1;
- O mesmo serviço de *backup* da seção anterior também é responsável pelo salvamento das informações do Banco de Dados do ambiente virtual e do Servidor de Aplicação do ambiente virtual.

6.10. Conclusão do Capítulo

No capítulo 6, foi explicado o funcionamento de todo o protótipo, junto com as suas definições formais da engenharia de *software* (diagramas UML). Demonstrou-se o funcionamento e suas ferramentas que auxiliam na persuasão, que proporciona criar um ambiente persuasivo para a tomada de decisões e escolha da melhor estratégia, favorecendo o jogador atingir o seu objetivo.

²⁶ Open Data Base Connectivity.

CONCLUSÃO

Atualmente, poucos são aqueles que têm consciência da importância dos Princípios da Persuasão aplicada à tomada de decisão. Tanto que não existe um modelo de tomada de decisão que se utilize dos artifícios da Teoria da Persuasão. Tal realidade foi comprovada ao longo deste trabalho, quando demonstrou que os jogos mostram-se ainda muito incipientes em relação ao uso desta teoria.

Com a presente pesquisa, ficou claro que a implementação da Teoria da Persuasão como ferramenta de tomada de decisão no ambiente MMORPG mudará o modo de lidar com a informação (conhecimento), de criar a melhor estratégia e de relacioná-los com os outros jogadores. Alguns estudiosos como Pavão, Lopes, Klimick & Casanova, Vygotsky, entre outros, principalmente pesquisadores da área acadêmica, vêm desenvolvendo pesquisas no sentido de se criar mundos virtuais que envolvam o usuário, trabalhando com elementos lúdicos que estimulem a fantasia e, por consequência, a criatividade. Assim, cria-se atividade lúdica, a fim de fazer com que os jogadores participem de maneira espontânea e criativa, tornando a construção do conhecimento mais proveitosa (PAVÃO, 1996) (LOPES, KLIMICK & CASANOVA, 2002) (VYGOTSKY, 1994).

Contudo, há de se considerar que ainda há muito a se fazer. Pois, enquanto não houver um verdadeiro interesse comercial no desenvolvimento destes ambientes na área educacional ou em outras áreas, esta iniciativa não passará de meros projetos acadêmicos e talvez jamais chegarão a se concretizar.

Tal realidade poderá ser mudada a partir do momento em que os menos atentos aos benefícios trazidos pelos Princípios da Persuasão passarem a trabalhar em ambientes capazes de entender as necessidades dos jogadores, a fim de favorecer a criação de novas estratégias na obtenção do conhecimento, e de passarem a aprender com outros jogadores, tornando o ambiente um verdadeiro parceiro do processo de aprendizagem.

Não obstante, ficou demonstrado que o objetivo principal dos Princípios da Persuasão não é forçar outras pessoas a acreditarem em determinado assunto ou aliciar a fazer algo que o outro não queria, mas sim usar a comunicação para alterar

atitudes, crenças ou comportamentos de outras pessoas, de modo que essa mudança ocorra voluntariamente. Em outras palavras, o indivíduo que persuade leva o seu interlocutor à aceitação de uma determinada idéia. Então, busca-se o desenvolvimento de um modelo que permita uma melhor tomada de decisão, inserindo elementos que possibilitem a implantação da persuasão entre os jogadores.

Essa dissertação visou justamente atender para os reais benefícios que os Princípios da Persuasão podem trazer para o desenvolvimento de ambiente de tomada de decisão, demonstrando as vantagens da utilização deste princípio em um ambiente de jogo já existente, a fim de se modelar uma proposta de arquitetura apta a atender as reais necessidades dos jogadores - problemas estes apresentados no ambientes jogáveis convencionais.

Como objeto de pesquisa, foi escolhido o desenvolvimento de um modelo que possibilite a criação de ferramentas de apoio a persuasão para jogadores na busca do conhecimento, em razão da carência de ambientes que detenham estas ferramentas e que satisfaçam as necessidades dos jogadores. Para isto, o modelo foi aplicado ao MMORPG, modalidade de jogo já solidificada na sociedade.

A fim de alcançar o mencionado objetivo, mesclou-se à prática a análise teórica da literatura sobre os Princípios da Educação, Teoria dos Jogos, RPG e a Rede de Petri, para depois, através daquele embasamento teórico, ter subsídios suficientes para apontar os pontos fortes no desenvolvimento do modelo citado.

Dessa forma, acredita-se que os sistemas de tomada de decisão, aplicados ao ambiente MMORPG e unidos aos princípios e teorias anteriormente mencionados, atuarão em conformidade com as condições propostas do modelo da persuasão, dando suporte aos jogos, uma vez que permitirá aos jogadores uma melhor construção do conhecimento. A persuasão, então, contribuirá para os sistemas de tomada de decisão, pois transformará as decisões de condições de incerteza para as condições de certeza por fatores persuasivos demonstrados nos princípios.

Portanto, propôs-se a criação de um ambiente MMORPG que possa ser utilizado pelos professores em suas salas de aulas, como uma ferramenta que possibilite os jogadores aprenderem e fixarem a matéria de uma forma descontraída e, principalmente, por meio do trabalho em equipe.

6.1. Trabalhos Futuros

Para trabalhos futuros, pretende-se aprofundar os estudos relacionados às técnicas de tomada de decisões aplicadas aos jogos, na busca de uma melhor estratégia para o jogo.

Pretende-se implementar o modelo, afim de que possa experimentá-lo em um grupo de jogadores de MMORPG, e assim, analisar os resultados com o intuito de validar o funcionamento do ambiente.

Do ponto de vista educacional, há a necessidade de implementar este modelo proposto e aplicá-lo na comunidade, a fim de que possa estudar os resultados obtidos na busca do melhoramento do modelo.

Aproveitando o fator genérico de atuação do modelo proposto, pretende-se adaptá-lo a outras áreas de conhecimento carentes em estratégias de persuasão, por exemplo, comércio eletrônico, simulação de cenários para economia, jogos de tabuleiros e etc.

REFERÊNCIAS

BARANAUSKAS, M. C. C., ROCHA, H. V., MARTINS, M. C., D'ABREU, J. V. **Uma taxonomia para ambientes de aprendizado baseados no computador**. In: VALENTE, J. A. (Org). O Computador na Sociedade do Conhecimento. Brasília: Proinfo-SED Ministério da Educação, Governo Federal, p. 45-69, 1999.

BATISTA, Emerson de Oliveira. **Sistema de Informação: o uso consciente da tecnologia para o gerenciamento**. São Paulo: Saraiva, 2004.

BIOHAZARD. Disponível em <<http://cms.mit.edu/games/education/Biohazard/Intro.htm>> Acesso em: 03 fev. 2010.

BIOWARE CORP. Baldur's Gate – Official Page. 2010. Disponível em: <http://www.bioware.com/games/baldurs_gate/>. Acesso em: 16 dez. 2010.

_____. Neverwinter Nights – Official Page. 2010. Disponível em: <<http://nwn.bioware.com>>. Acesso em: 16 dez. 2010.

BLIZZARD ENTERTAINMENT. Diablo – Official Page. 2010. Disponível em: <<http://www.blizzard.com/diablo/>>. Acesso em: 16 dez. 2010.

_____. World of Warcraft – Official Page. 2010. Disponível em: <<http://us.battle.net/wow/en/>>. Acesso em: 16 dez. 2010.

BOOCH, Grady. **UML - Guia Do Usuário**. 2ª Edição. São Paulo, 2010, 500 pp.

BREHM, J. W. **A Theory of Psychological Reactance**. New York: Academic Press, 1966.

CALE, C. **The real truth about dungeons & dragons**. 2002. Disponível em: <<http://www.cale.com/paper.htm>>. Acesso em: 16 dez. 2010.

CARDWELL Jr, P. The attacks on role-playing games. **Skeptical Inquirer**, 157-165, *Studies About Fantasy Role-Playing Games*, 1994. Disponível em: <<http://www.rpgstudies.net/cardwell/attacks.html>>. Acesso em: 16 dez. 2010.

CHAN, T. W. Learning Companion Systems, Social Learning Systems, and Global Social Learning Club. **Journal of Artificial Intelligence in Education**, vol. 7, n. 2, p. 125-159, 1996. Disponível em: <<http://chan.lst.ncu.edu.tw/publications/1996-Chan-lcs.pdf>>. Acesso em: 16 nov. 2010.

CIALDINI, Robert B. **Influence The Psychology of Persuasuion**. Collins, 1998.

CITELLI, Adilson. **Linguagem e persuasão**. 2 ed. São Paulo: Ática, série Princípios, 1986.

CONCEIÇÃO, K.; GONÇALVES, M. B. Contribuição para o ensino de matemática nos cursos de engenharia. In: **XXIV Encontro Nac. de Eng. de Produção**, Florianópolis, 2004. Disponível em: <http://www.abepro.org.br/biblioteca/ENEGEP2004_Enegep1101_0228.pdf>. Acesso em: 16 nov. 2010.

CONWAY, J. All Games Brigh and Beautiful. **The American Mathematical Monthly**, pp. 417–434, 1977.

_____. **A Gamut of Game and Theories**. *Mathematics Magazine*, pp. 5–12, 1978.

COSTA, E. B.; BITTENCOURT, I. I. **Construção de Ambientes Interativos de Aprendizagem usando Agentes Inteligentes**. 1 ed. Brasília, 2006.

DEBBIO, Marcelo D. **Arkanun**. 2. ed. Local : Demon, 1998.

DEITEL, H. M.; DEITEL, P. J.; STEINBUHLER, K. **E-Business e e-commerce para administradores**. Tradução Maurício Stocco, Mônica Rosemberg, José Carlos Barbosa dos Santos. São Paulo: Pearson Education do Brasil, 2004. Cap XIII, p.269.

DAVENPORT, Thomas H; PRUSAK, Laurence. **Conhecimento empresarial**. Tradução de Lenke Peres. Rio de Janeiro: Campus; São Paulo: Publifolha, 1999.

DRATH, R., **Visual Object Net ++**. Technical University of Ilmenau, Alemanha, 2004. Disponível em: <http://www.systemtechnik.tuilmenu.de/~drath/visual_E.htm>. Acesso em: 16 nov. 2010.

ELETRONIC ARTS, Inc. Ultima Online Visitor Center. 2010. Disponível em: <<http://www.ou.com/visitor>>. Acesso em: 16 dez. 2010.

FADEL, Ricardo. **O que está rolando em torno de World of Warcraft?** Baixaki Jogos. 2009. Disponível em: <<http://www.baixakijogos.com.br/noticias/6092>>. Acesso em: 16 dez. 2010.

FERNANDES, L. D. *et al.* Jogos no computador e a formação de recursos humanos na indústria. **SIMPÓSIO BRASILEIRO DE INFORMÁTICA NA EDUCAÇÃO**, VI. Florianópolis, 1995.

FERREIRA, L. de F. **Ambiente de aprendizagem construtivista**. Disponível em: <<http://www.penta.ufrgs.br/~luis/Ativ1/Construt.html>>. Acesso em: 16 nov. 2010.

FERREIRA, M. S.; PALAZZO, L. A. M. Sistemas Tutoriais Inteligentes na Formação de Professores. In: **VII Oficina de Inteligência Artificial**. Pelotas: EDUCAT, 2003.

FISCHETTI, E.; GISOLFI, A. From Computer-Aided Instruction to Intelligent Tutoring Systems. In: **Educational Technology**, vol. 30, n. 8, p. 7-17, 1990.

FOINA, Paulo Sérgio. **Tecnologia de Informação**: planejamento e gestão. São Paulo: Atlas, 2001.

FREIRE, Paulo. **Pedagogia do Oprimido**. 28.ed. Rio de Janeiro: Paz e Terra, 1987, 184pp.

GALVIS, A. H. **Ingeniería de software educativo**. Santa Fé, Bogotá: Ediciones Uniandes, 1992.

GAMES-TO-TEACH PROJECT. Disponível em <http://cms.mit.edu/games/education/>
> Acesso em: 03 fev. 2010.

GARCIA, E.; GARCIA, O. P. A importância do sistema de informação gerencial para a gestão empresarial. **Revista Ciências Sociais em Perspectiva, do Centro de Ciências Sociais Aplicadas de Cascavel**, Cascavel, v.2 , n.1, p. 21-32, 1 sem. 2003

GIL, Antônio de Loureiro. **Sistema de Informações Contábil/Financeiros**. 3. ed. São Paulo: Atlas, 1999.

GIRAFFA, L. M. M. **Uma arquitetura de tutor utilizando estados mentais**. Tese de Doutorado em Ciência da Computação. Universidade Federal do Rio Grande do Sul. Porto Alegre - RS, 1999.

GOODSON-ESPY, Tracy.; ESPY, Samuel.; CIFARELLI, Victor Warrick's Secrets: Teaching Middle School Mathematics through an Internet based 3D Massively Multiplayer Role Playing Game. In: **Society for Information Technology and Teacher Education International Conference**. 2002, n. 1, 1080-1084p.

GOULDNER, A. W. The Norm of Reciprocity: A Preliminary Statement. **American Sociological Review** 25, 1960: 161–78.

HAYDEN, S.; CARRICK, C.; YANG, Q. **Architectural Design Patterns for Multiagent Coordination** - Proc. of the International Conference on Agent Systems, Agents'99 Seattle, WA, 1999.

HEPHAESTUS. Disponível em <[http://cms.mit.edu/games/education/Hephaestus/Intro .htm](http://cms.mit.edu/games/education/Hephaestus/Intro.htm)> Acesso em: 03 fev. 2010.

HUGHES, J. **Therapy is fantasy: Roleplaying, healing and the construction of symbolic order**. Studies About Fantasy Role-Playing Games,1988. Disponível em: <http://www.rpgstudies.net/hughes/therapy_is_fantasy.html>. Acesso em: 16 dez. 2010.

HUNTER, William. **The dot Eaters – Videogame History 101**. 2003. Disponível em: <<http://www.emuunlim.com/doteaters/>>. Acesso em: 16 dez. 2010.

IGDA – International Game Developers Association. Alternate Reality Games White Paper. 2006. Disponível em: < <http://archives.igda.org/arg/resources/IGDA-AlternateRealityGames-Whitepaper-2006.pdf> >. Acesso em: 16 dez. 2010.

JACKSON, S. **Módulo básico RPG.** GURPS (2a ed.). São Paulo: Devir, 1994.

JENSEN, K., Coloured Petri Nets: Basic concepts, analysis methods and practical use. **EATCS monographs on Theoretical Computer Science**, Springer-Verlag, Berlim, 1996.

JUDE DESIGN & COMMUNICATION. **Jude/Community.** 31, Dezembro, 2009. Disponível em <<http://jude.change-vision.com/jude-web/product/community.html>>. Acesso em: 22 jun. 2010.

KLIMICK, Carlos. **Revista Dragão Brasil**, Rio de Janeiro, n.64, 1992.

KLIMICK, C.; LIMA, M.; VELOSO, M. A., **Esferas: RPG.** Rio de Janeiro: Akritó, 2001.

KROGH, G. V.; ICHIJO, K.; NONAKA, I. **Facilitando a criação do conhecimento: reinventando a empresa com o poder da inovação contínua.** Rio de Janeiro: Campus, 2001.

LANDOWISKI, E. Sobre el contagio. In: **Semiótica, estesis, estética.** São Paulo: Educ, 1999.

LAUDON, Kenneth C.; LAUDON, Jane Price. **Sistemas de informação.** 4. ed. LTC: Rio de Janeiro, 1999.

LEAKEY, R. **People of the Lake.** New York: Anchor Press/Doubleday, 1978.

LEVY, P. **O que é o virtual.** SP: Editora 34, 1996.

_____. **Cibercultura.** São Paulo: Ed. 34, 1999, 260 pp.

LOPES, Laura M. C.; KLIMICK, Carlos; CASANOVA, Marco A. Relato de uma Experiência de Sistema Híbrido no Ensino Fundamental: Projeto Aulativa. In: **Revista Brasileira de Aprendizagem Aberta e a Distância**. Associação Brasileira de Educação à Distância, 2002, vol.1 n. 2. Disponível em <<http://www.abed.org.br>> Acesso em: 03 fev. 2010.

LORENZI, F, BAZZAN, A. & PAPROCKI, L. **Labirinto**: uma ferramenta para apoio no processo de ensino e aprendizagem via internet. Fortaleza, WIE, SBC, 2001.

MACIEL, P. R.; LINS, R. D.; CUNHA, P. **Introdução às Redes de Petri e Aplicações**, Campinas: X Escola de Computação, 1996.

MARCATO, A. **Saindo do quadro**: uma metodologia educacional lúdica e participativa baseada no Role Playing Games. São Paulo: A. Marcato, 1996.

MATURANA, H. R. e VARELA, F. J. **Autopoiesis and Cognition**: the Realization of Living, Reidel, Dordrecht,1980.

MENESES, L. J. S. A. **Formalização da interação colaborativa no âmbito do SIANALCO. Florianópolis**. Dissertação de mestrado em Ciência da Computação, Universidade Federal de Santa Catarina, Florianópolis - SC. 2001.

MURATA, T., **Petri Nets: properties, analysis, and applications**, Proc. IEEE pp. 541-580, 1989.

O'BRIEN, James A. **Sistemas de informação e as decisões gerenciais na era da internet**. Tradução de Cid Knipel Moreira. São Paulo: Saraiva, 2002.

OLIVEIRA, Djalma de Pinho Rebouças de. **Sistemas de informação gerenciais: estratégias, táticas, operacionais**. 8. ed., São Paulo: Atlas,1992.

_____. **Sistemas, organizações e métodos: uma abordagem gerencial**. 13. ed. São Paulo, 2002.

OTTE, Henrique. **Jogador e mestre de RPG**, 2002.

PADOVEZE, Clóvis Luís. **Sistemas de Informações Contábeis: fundamentos e análise**. 2. ed., São Paulo: Atlas, 2000.

PAES, Rodrigo B. **Regulando a Interação de Agentes em Sistemas Abertos – uma abordagem de Leis**. Dissertação (Mestrado) – Departamento de Informática, Pontifícia Universidade Católica do Rio de Janeiro, Rio de Janeiro, 2005.

PALLOFF, R. e PRATT, K. **Building Learning Communities in Cyberspace**. San Francisco: Jossey-Bass Publishers, 1999.

PAPERT, S. **A Máquina das Crianças: repensando a escola na era da informática**. Porto Alegre: Artes Médicas 1994.

PARAGUAÇU, F. **Vygotsky: Un Environnement d'Apprentissage Social pour la programmation fondé sur la collaboration entre agents d'aide à la conception par cas**. Tese de Doutorado. Université d'Aix Marseille III. Marseille – France, 1997.

PAVÃO, A. **A Aventura da Leitura e da Escrita entre Mestres de RolePlaying Game (RPG)**. Brochura, São Paulo, 1996, 231pp.

PEREIRA, Maria José Lara de Bretãs; FONSECA, João Gabriel Marques. **Faces da Decisão: as mudanças de paradigmas e o poder da decisão**. São Paulo: Makron Books, 1997.

PESSÔA NETO, A. C. A. **Um modelo híbrido baseado em ontologias e RBC para a concepção de um ambiente de descoberta que proporcione a aprendizagem de conceitos na formação de teorias por intermédio da metáfora de contos infantis**. Dissertação de Mestrado em Modelagem Computacional de Conhecimento, Universidade Federal de Alagoas, Maceió - AL, 2006.

REVOLUTION. Disponível em:

<<http://cms.mit.edu/games/education/revolution/index.html>> Acesso em: 03 fev. 2010.

REZENDE, Denis Alcides; ABREU, Aline França de. **Tecnologia da informação aplicada a sistemas de informação empresariais**: o papel estratégico da informação e dos sistemas de informação nas empresas. São Paulo: Atlas, 2000.

RIBEIRO, V. S. **Ambiente de aprendizagem Web**: um olhar a partir de um curso de especialização do Laboratório de Ensino a Distância (LED/UFSC). Dissertação de mestrado, UFSC/PPGEP, Florianópolis, 2001.

RICKEL, Jeff W. Intelligent Computer-Aided Instruction: A Survey Organized Around System Components. Vol. 19, No. 1. **IEEE Transactions on Systems and Cybernetics**, 1989.

ROSENZWEIG, Phil. **The Halo Effect**: ... and the Eight Other Business Delusions That Deceive Managers. Free Press, 2007.

SANTAELLA, Lúcia. **A crítica das mídias na entrada do século XXI**. In: Crítica das práticas midiáticas: da sociedade de massa às ciberculturas / org. José Luiz^a Prado. São Paulo: Hackers Editores, 2002.

SANTOS, Edméa Oliveira. **Ambientes virtuais de aprendizagem**: por autorias livre, plurais e gratuitas. In: Revista FAEBA, v.12, no. 18. 2003.

SARTINI, B. A., *et al.* Uma Introdução a Teoria dos Jogos. **II Bienal da SBM**. Universidade Federal da Bahia. Outubro, 2004.

SILVA NETO, H. C. ; CARVALHO, L. F. B. S. ; PARAGUACU, F. ; LOPES, R. V. V. . Um modelo de aprendizado social no jogo de MMORPG. In: **IX Simpósio Brasileiro de Jogos e Entretenimento Digital – SBGames 2010**, Florianópolis. Anais do Simpósio Brasileiro de Jogos de Computador e Entretenimento Digita - Trilha de Games & Cultura, 2010.

SILVA, F. de M. **Concepção e realização de um modelo computacional de jogos interativos no contexto da aprendizagem colaborativa**. Dissertação de Mestrado em Modelagem Computacional de Conhecimento, Universidade Federal de Alagoas, Maceió - AL, 2008.

SILVA, V. T. **Módulo Pedagógico para um ambiente hipermídia de aprendizagem.** Dissertação de Mestrado em Engenharia de Produção, Universidade Federal de Santa Catarina, Florianópolis – SC, 2000.

SKINNER, Burrhus Frederic. **Ciência e comportamento humano.** São Paulo: Martins Fontes, 2003. 489p.

SRINIVASAN, S.S., ANDERSON, R., PONNAVOLU, K. Customer Loyalty in E-commerce: An Exploration of Its Antecedents and Consequences, **Journal of Retailing**, 2002, pp 41-50.

STAIR, Ralph M. **Princípios de sistemas de informação.** Rio de Janeiro: LTC, 1998.

STEPHEN P. Robbins.; COULTER, Mary. **Administração.** 5.ed. Prentice. Hall Interamericana, 1996.

TRACY, A. L., *et al.* Pluralistic ignorance and hooking up. **Journal of Sex Research**, maio de 2003. Disponível em:
<http://findarticles.com/p/articles/mi_m2372/is_2_40/ai_105518215/>. Acesso em: 16 dez. 2010.

UEYAMA, Jó. MADEIRA, Edmundo R. M. Um Modelo de Negociação, Automatizada para Comércio Eletrônico. **19 Simpósio Brasileiro de Redes de Computadores.** Florianópolis, Santa Catarina, Maio de 2001. Disponível em
<<http://www.gta.ufrj.br/sbrc2001/>>. Acesso em: 22 jun. 2010.

VALENTE, J. A. Diferentes Usos do Computador na Educação. In: **Computadores e Conhecimento: repensando a educação**, p. 1-23, Campinas-SP: Gráfica Central UNICAMP. 1993. Disponível em:
<<http://nied.unicamp.br/publicacoes/separatas/Sep1.pdf>>. Acesso em: 20 nov. de 2010.

VASCONCELOS, C.; PRAIA, J. F.; ALMEIDA, L. S. Teorias de aprendizagem e o ensino/aprendizagem das ciências: da instrução à aprendizagem. **Psicol. esc. educ.** vol.7, n.1, p.11-19, 2003. Disponível em:
<http://pepsic.bvsi.org.br/scielo.php?script=sci_arttext&pid=S141385572003000100002&lng=pt&nrm=iso>. Acesso em: 20 nov. de 2010.

VYGOTSKY, Lev S. **A formação social da mente**. 5.ed. São Paulo: Martins Fontes, 1994, 191pp.

APÊNDICE

APÊNDICE A – Diagramas de Classes

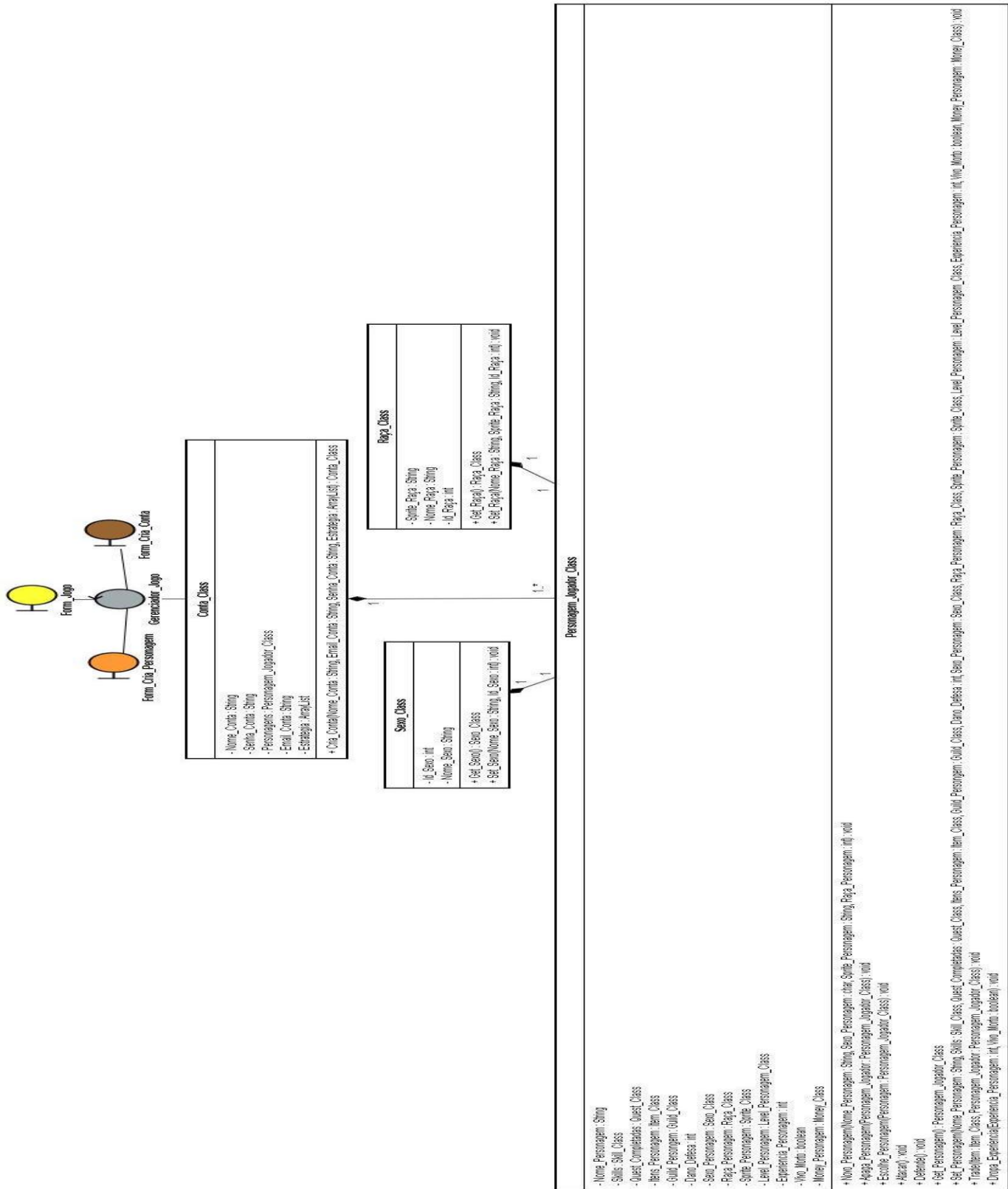


Figura 35: Diagrama de Classe cria conta e cria personagem.

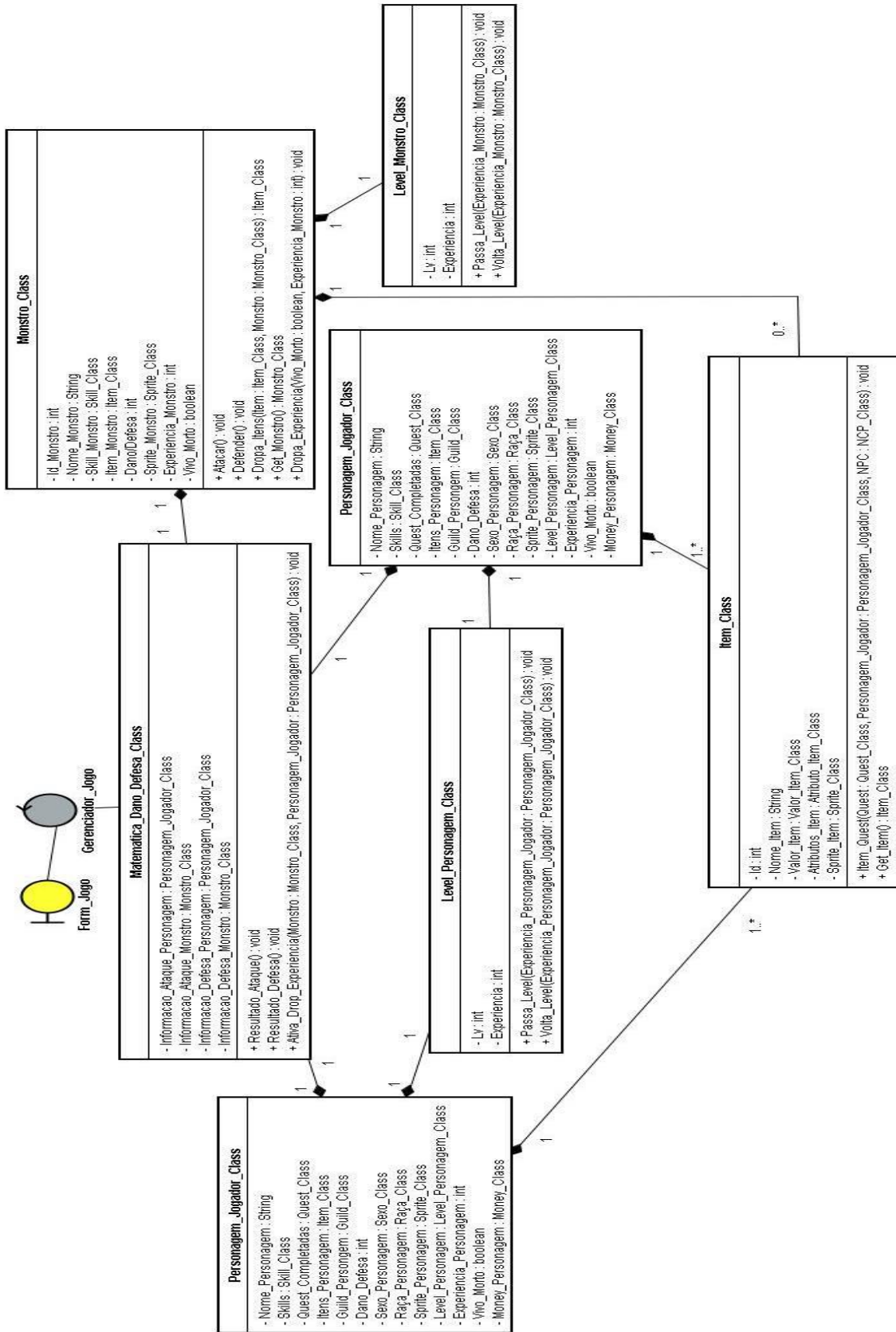


Figura 36: Diagrama Classe Caçar.

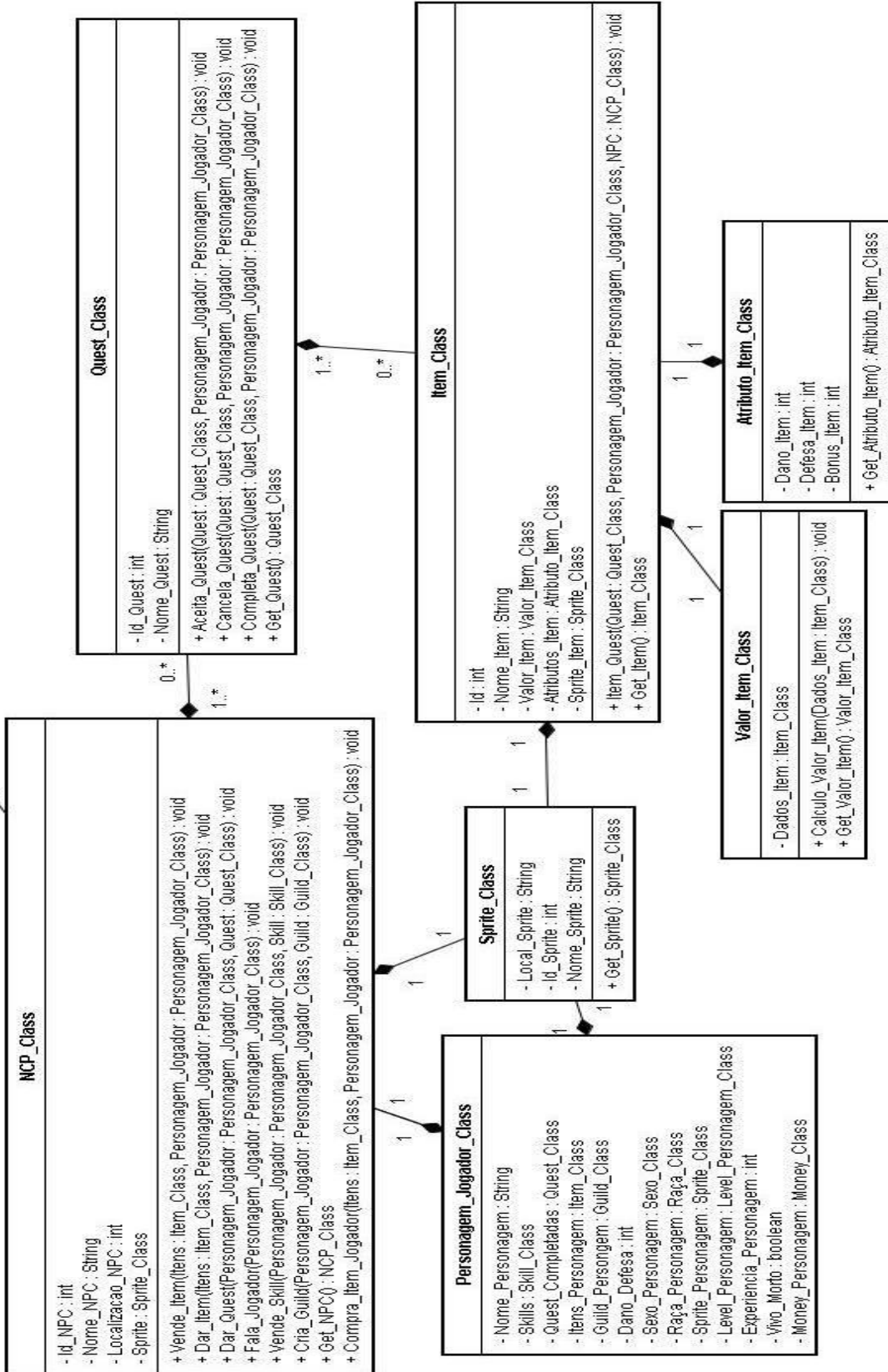
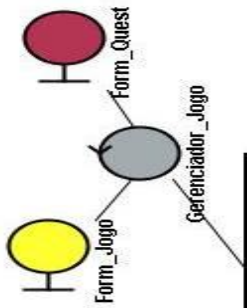


Figura 37: Diagrama Classe Adquirir Quest.

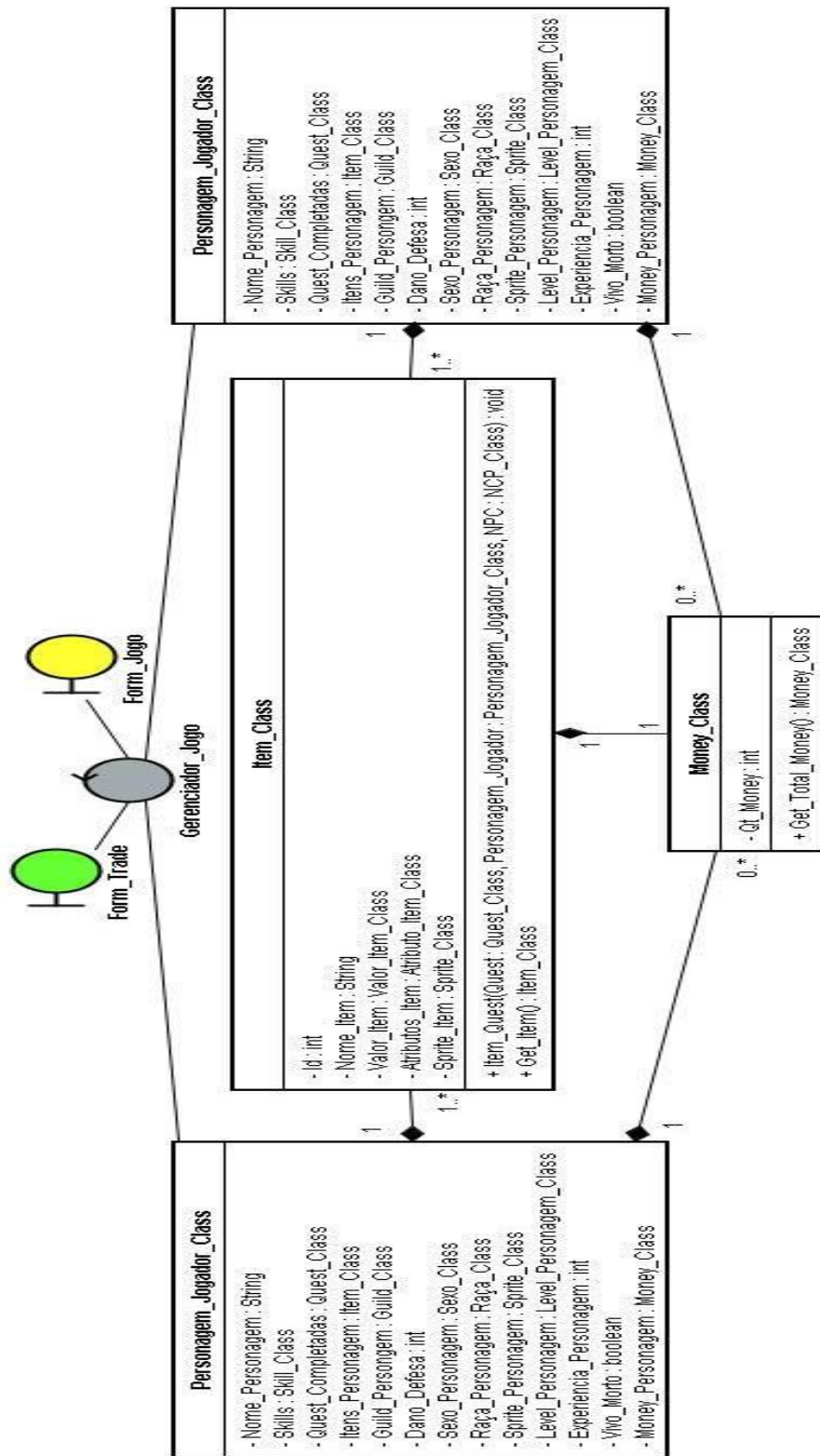


Figura 38: Diagrama Classe Compra Item Jogador.

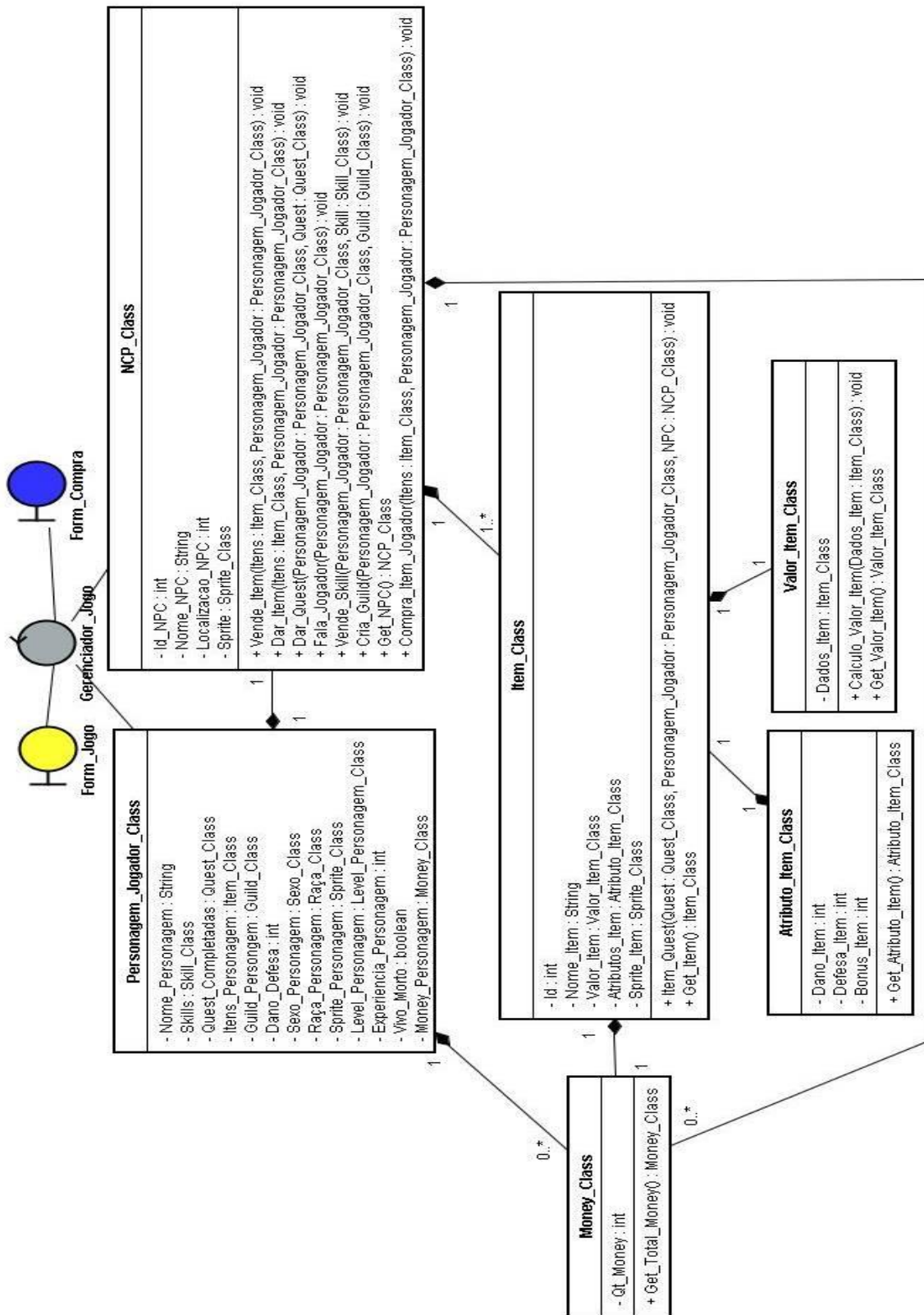


Figura 39: Diagrama Classe Compra ou Vende Item NPC.

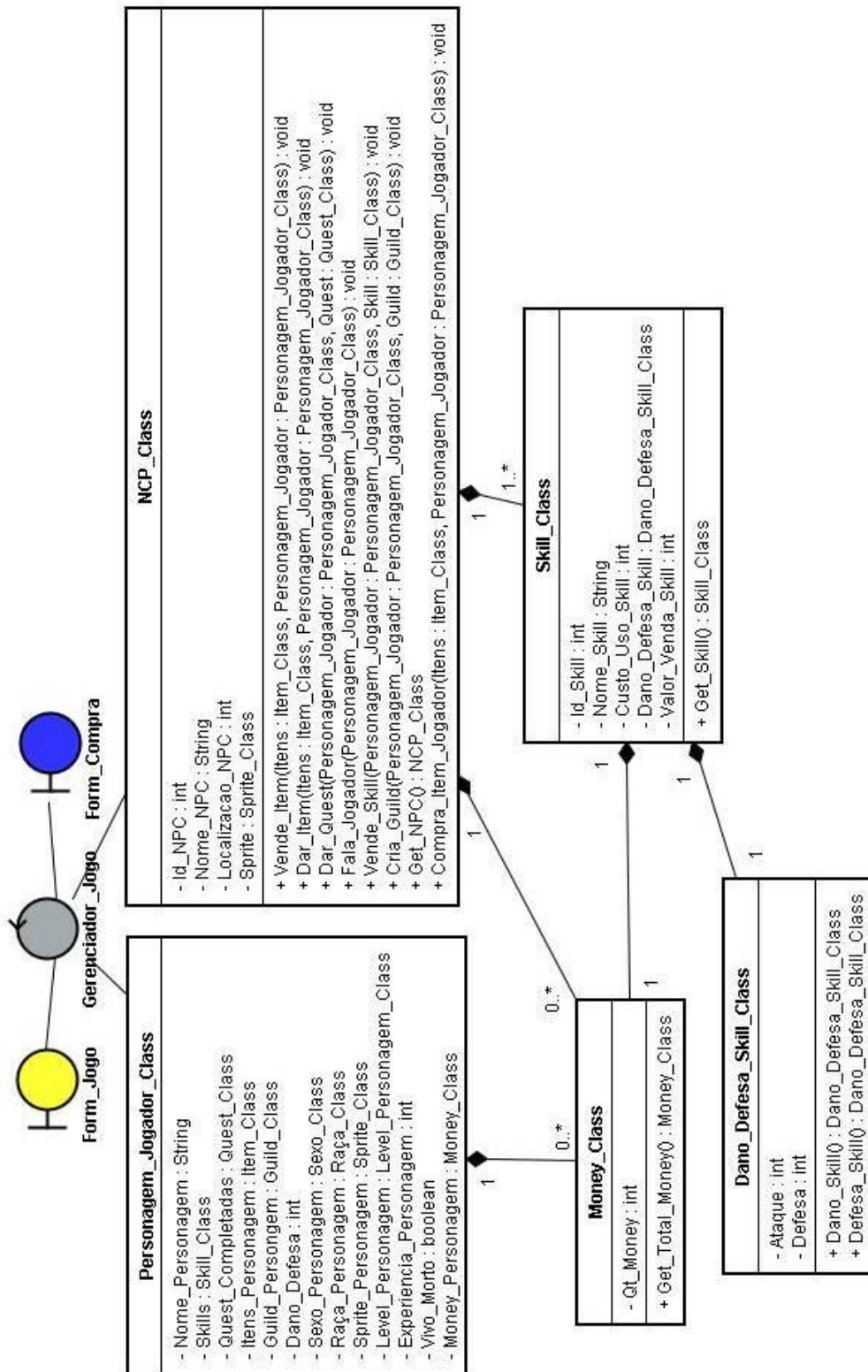


Figura 40: Diagrama Classe Compra Skill.

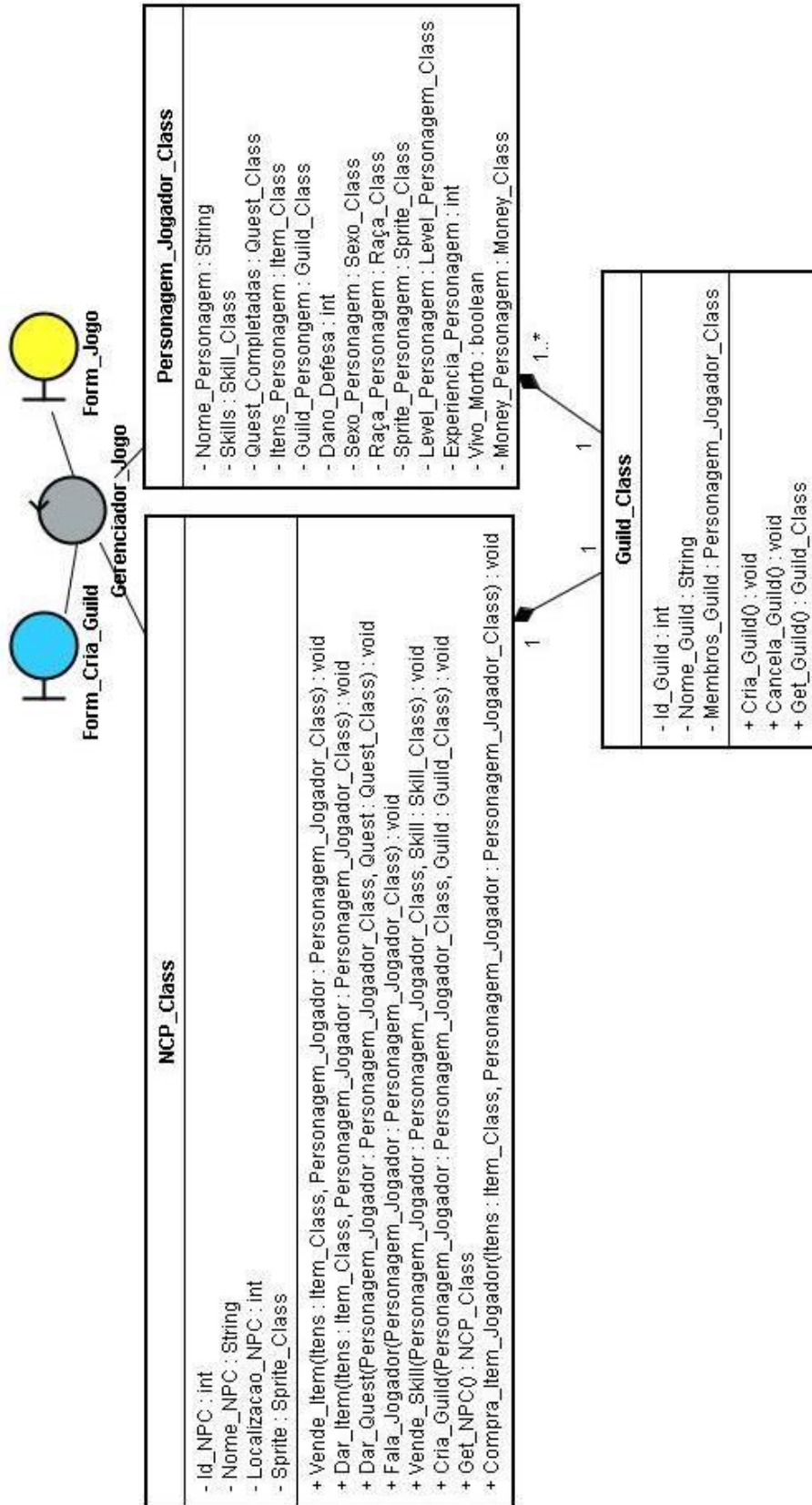


Figura 41: Diagrama Classe Cria Guild.

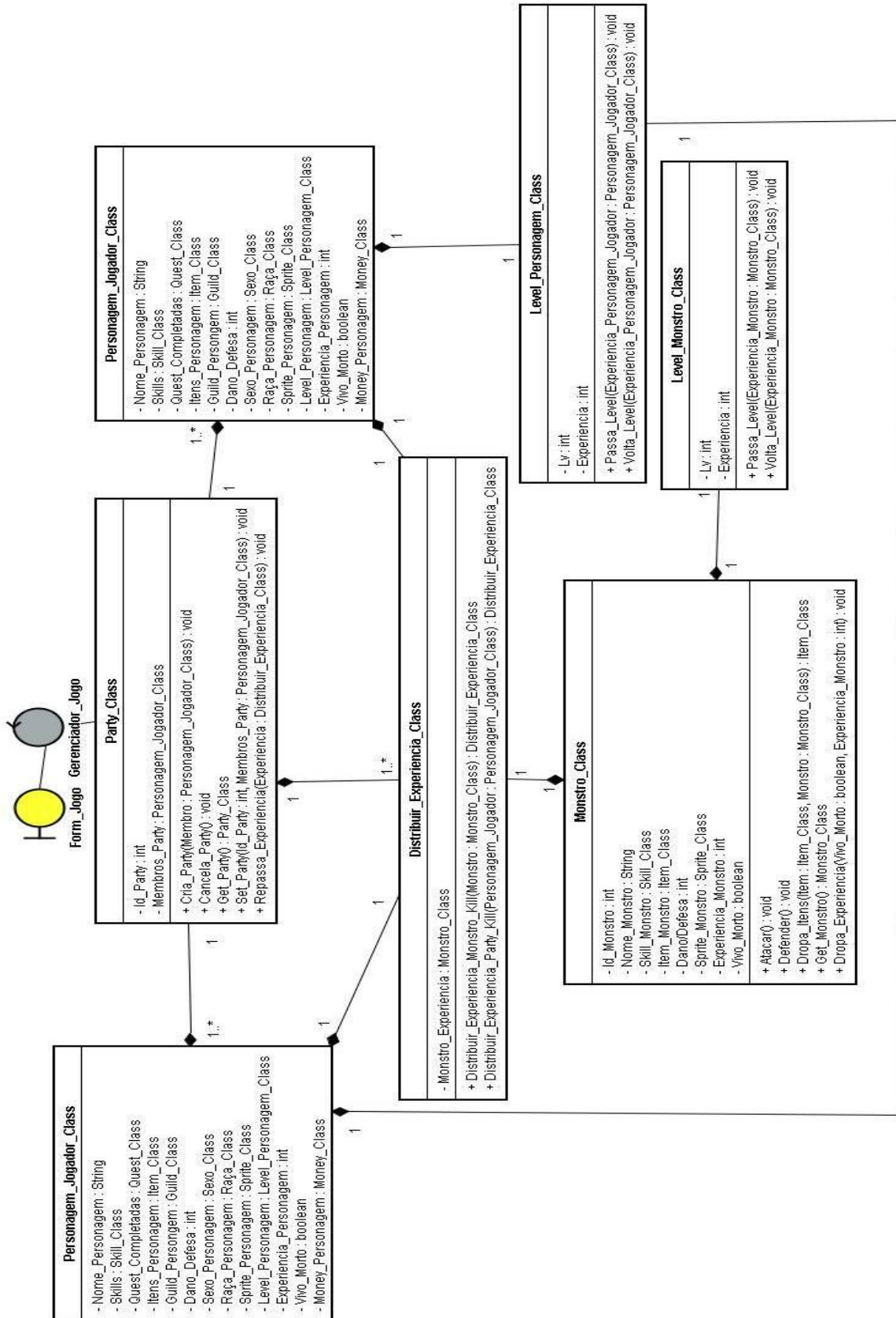


Figura 42: Diagrama Classe Cria Party.

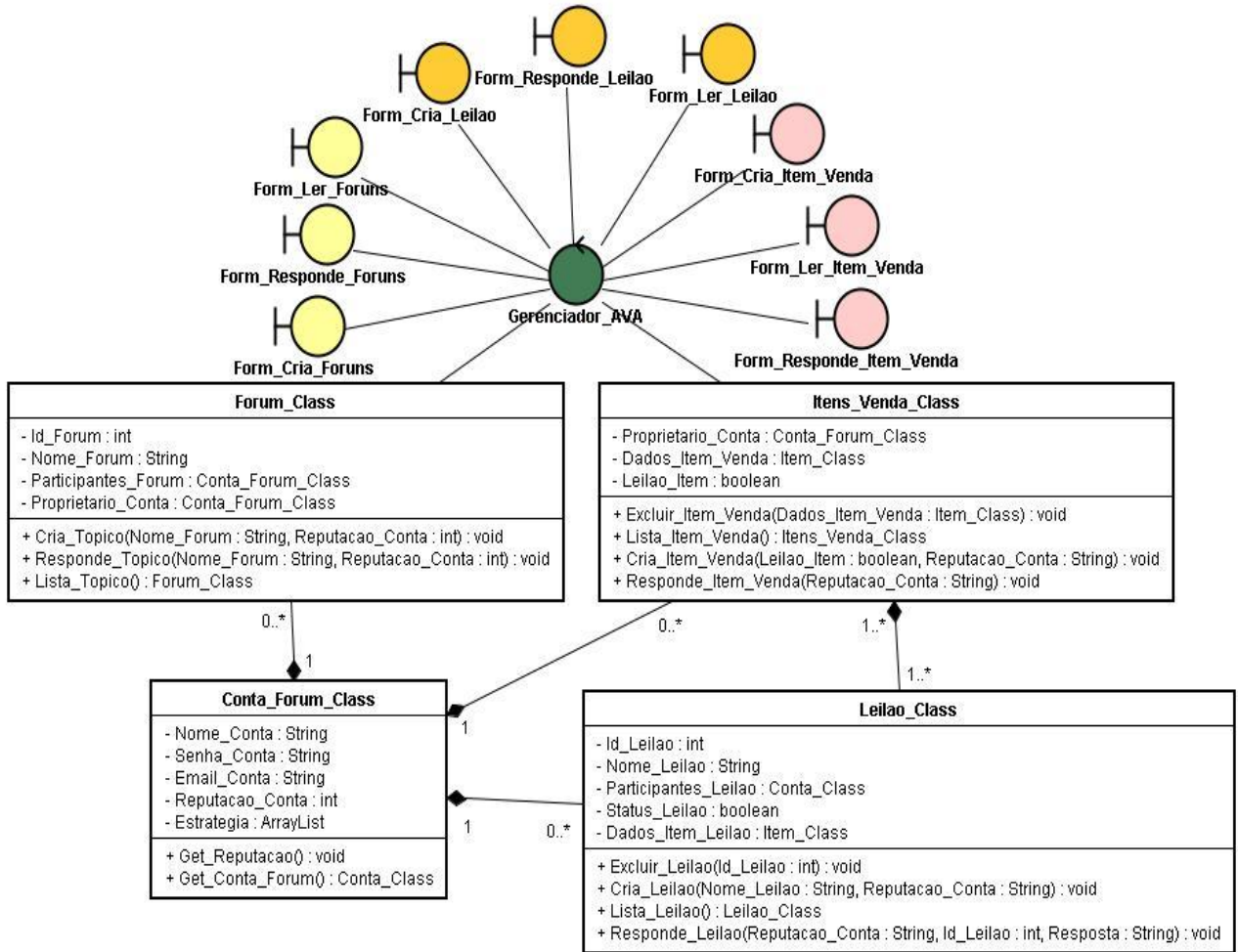


Figura 43: Diagrama Classe Fóruns, Leilão e Itens a Venda.

APÊNDICE B – Diagrama de Classes com Interface

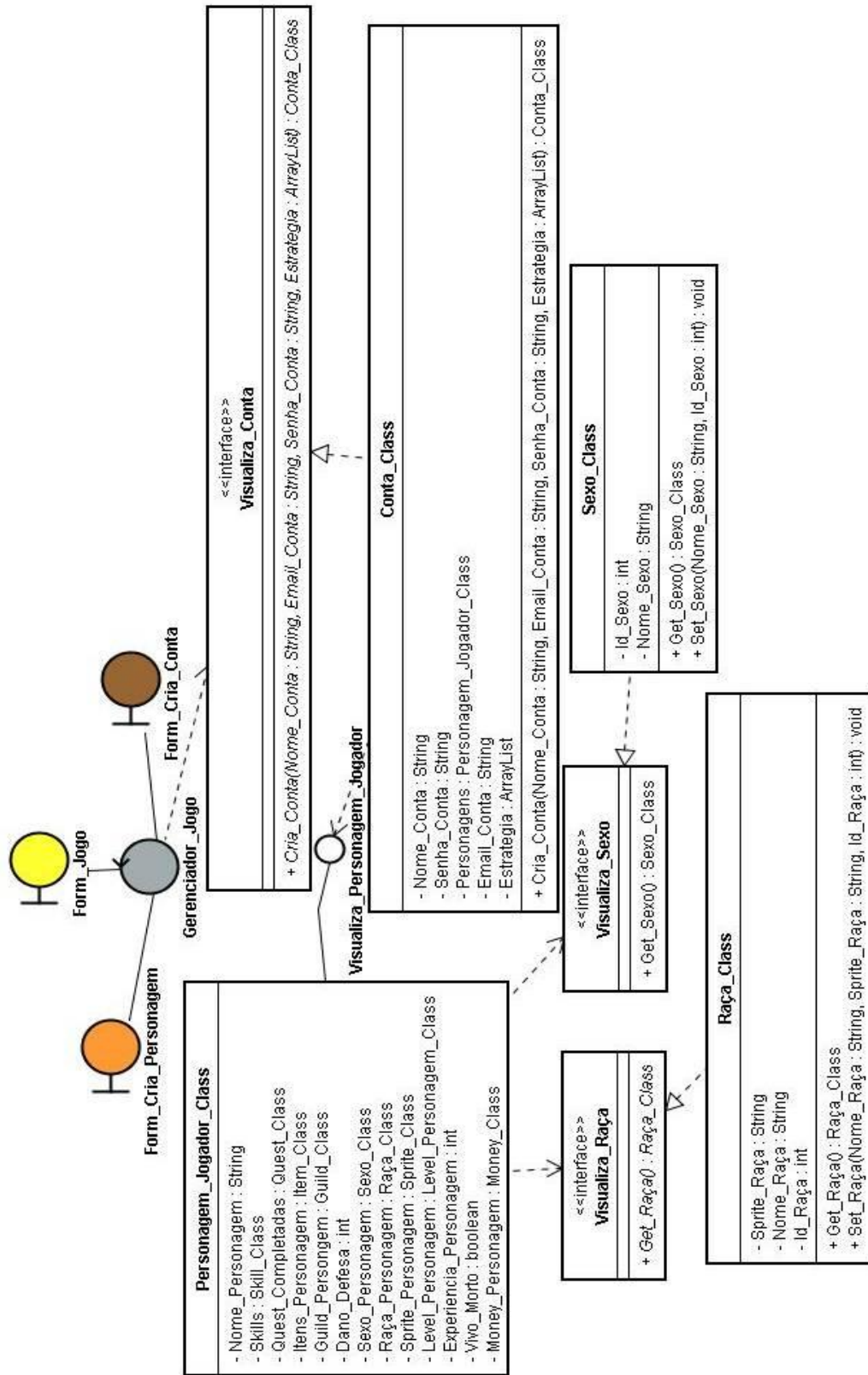


Figura 44: Diagrama Classe com Interface Cria Conta e Personagem Jogador.

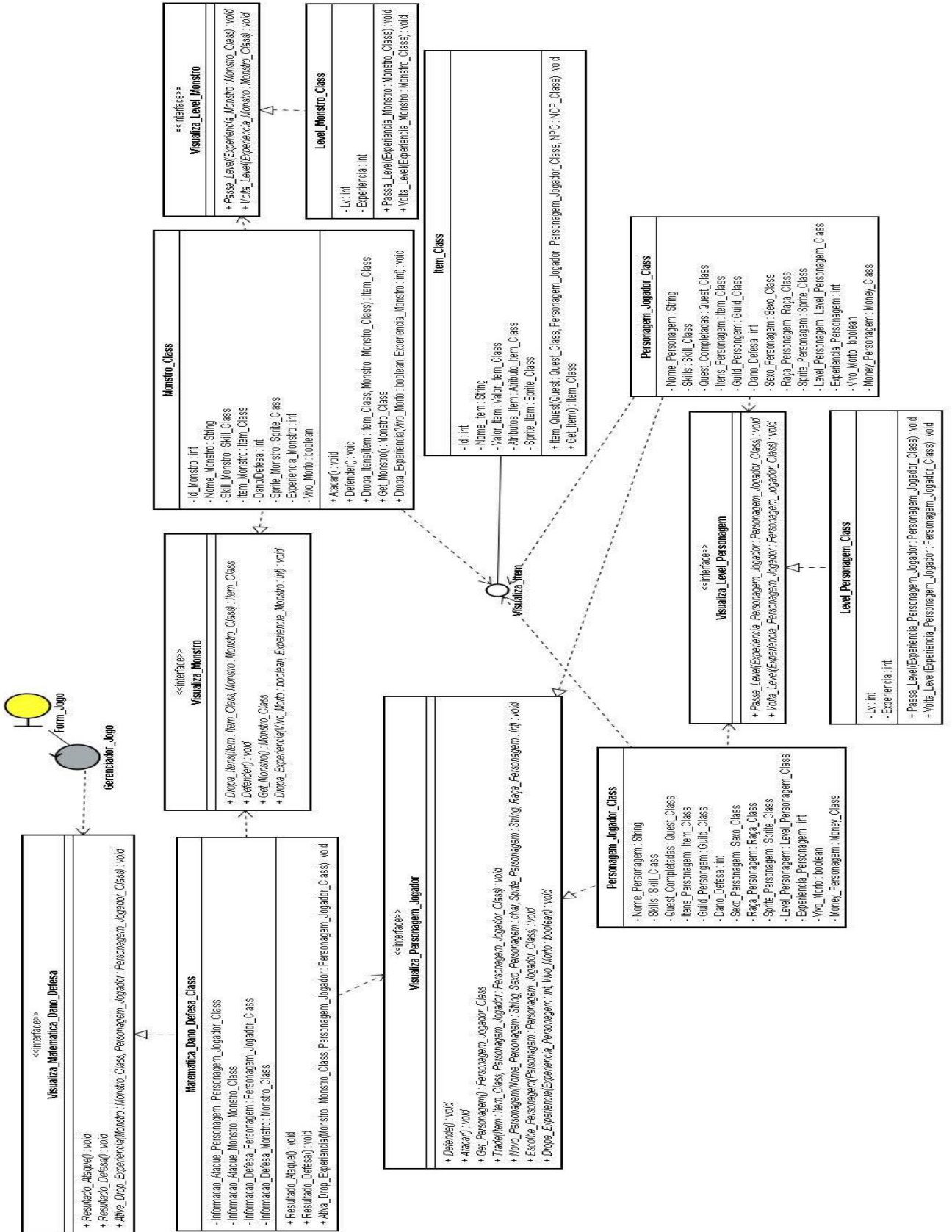


Figura 45: Diagrama Classe com Interface Caçar.

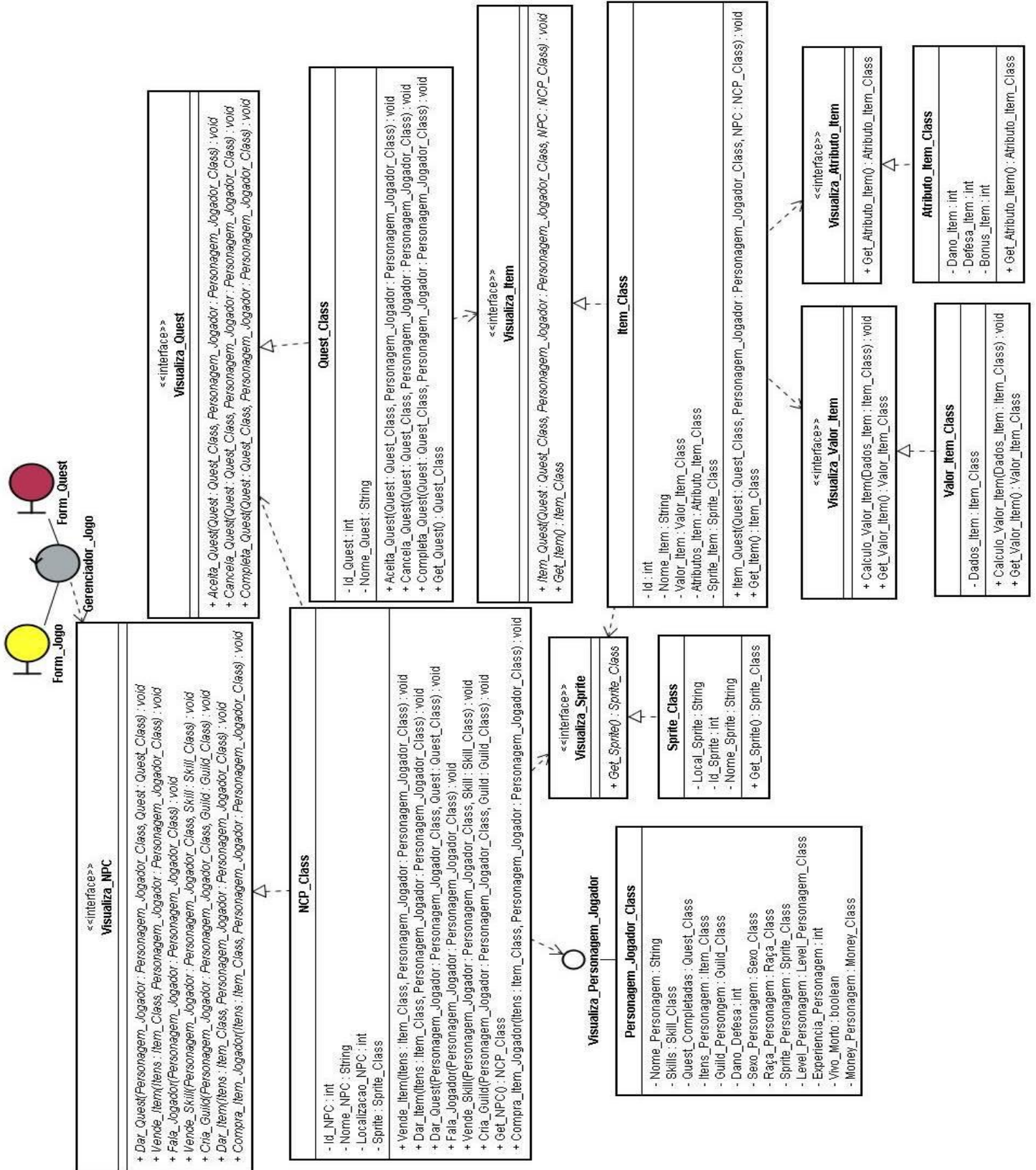


Figura 46: Diagrama Classe com Interface Adquirir Quest.

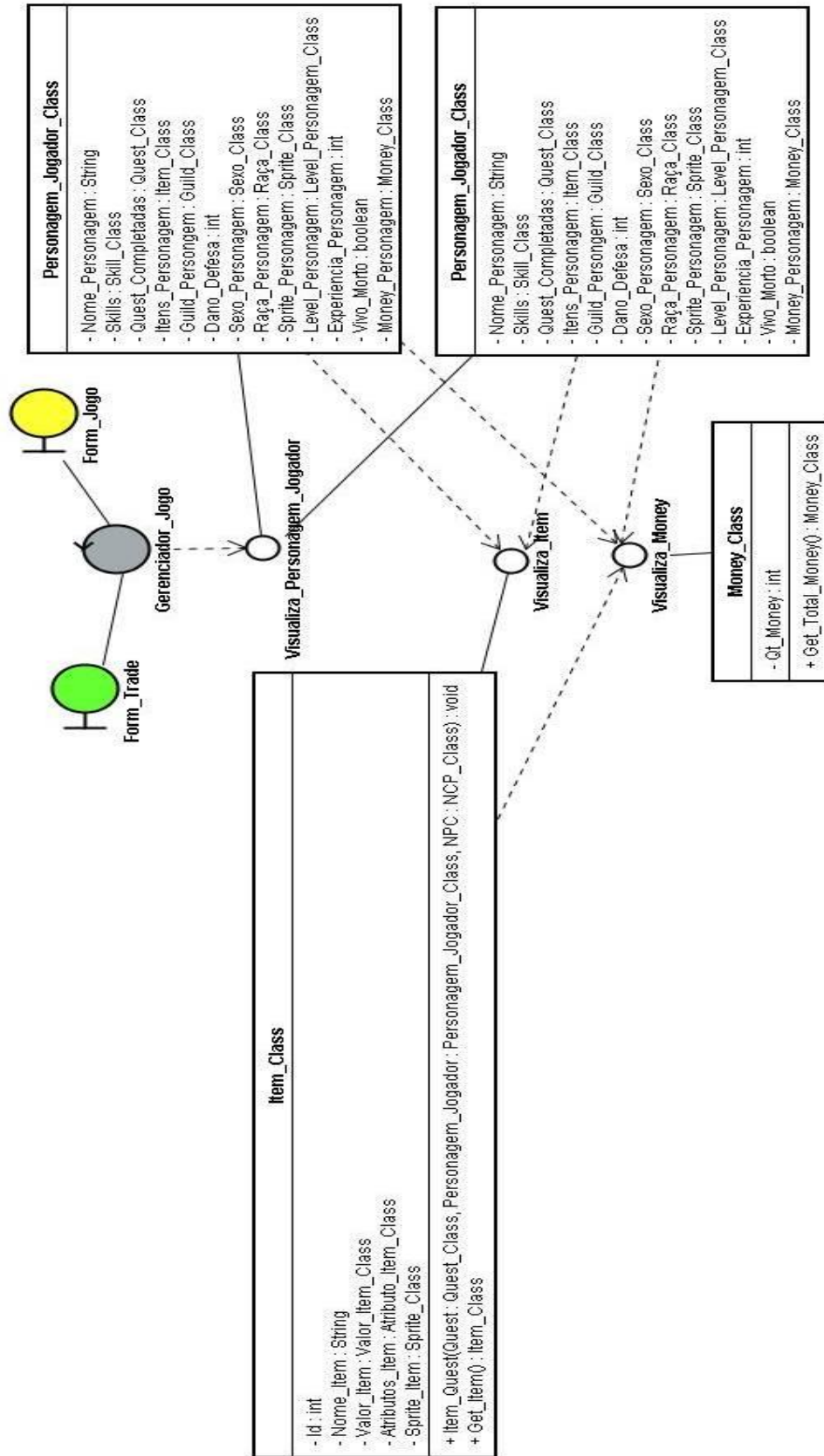


Figura 47: Diagrama Classe com Interface Compra Item Jogador.

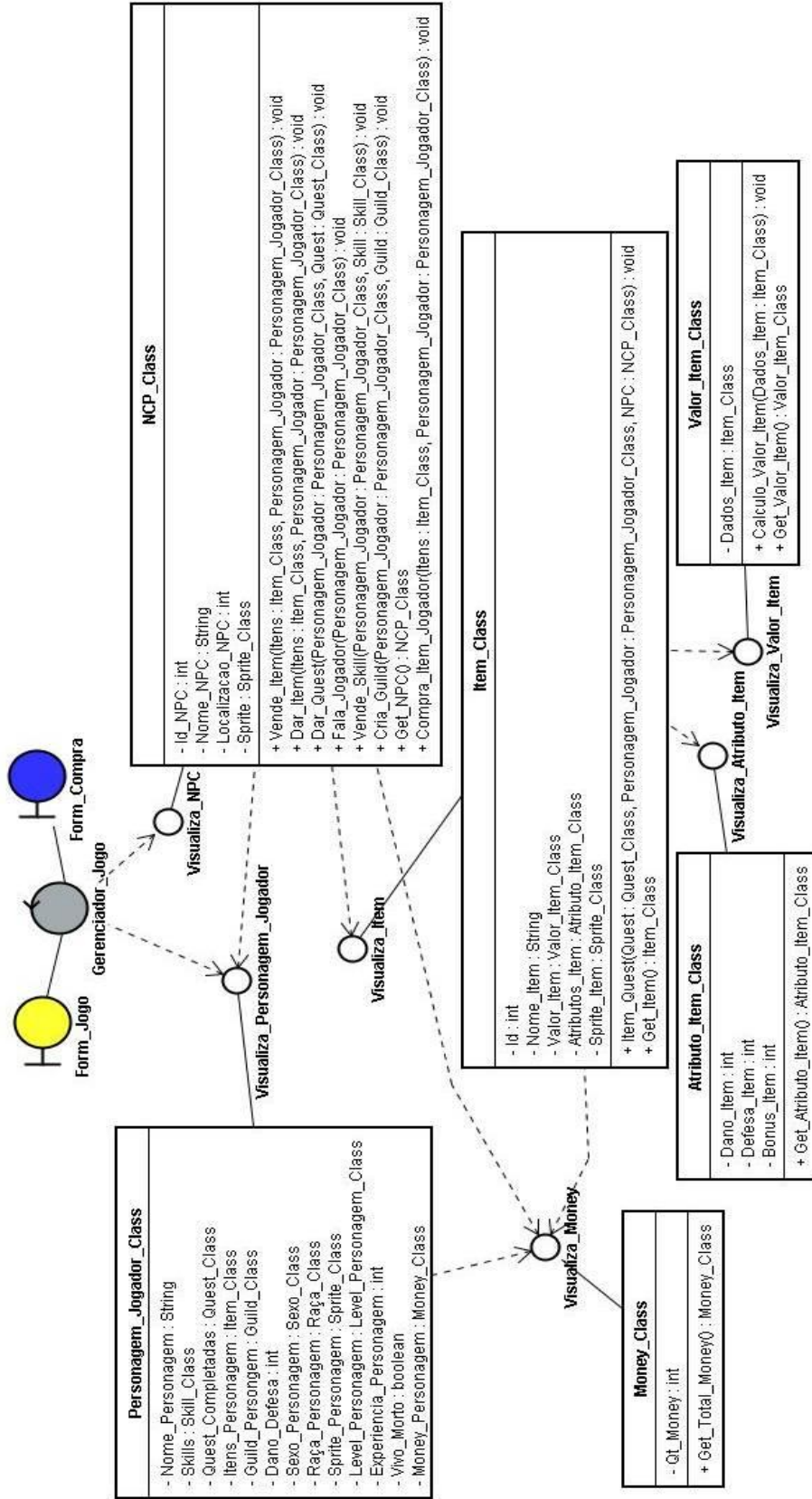


Figura 48: Diagrama Classe com Interface Compra ou Venda Item NPC.

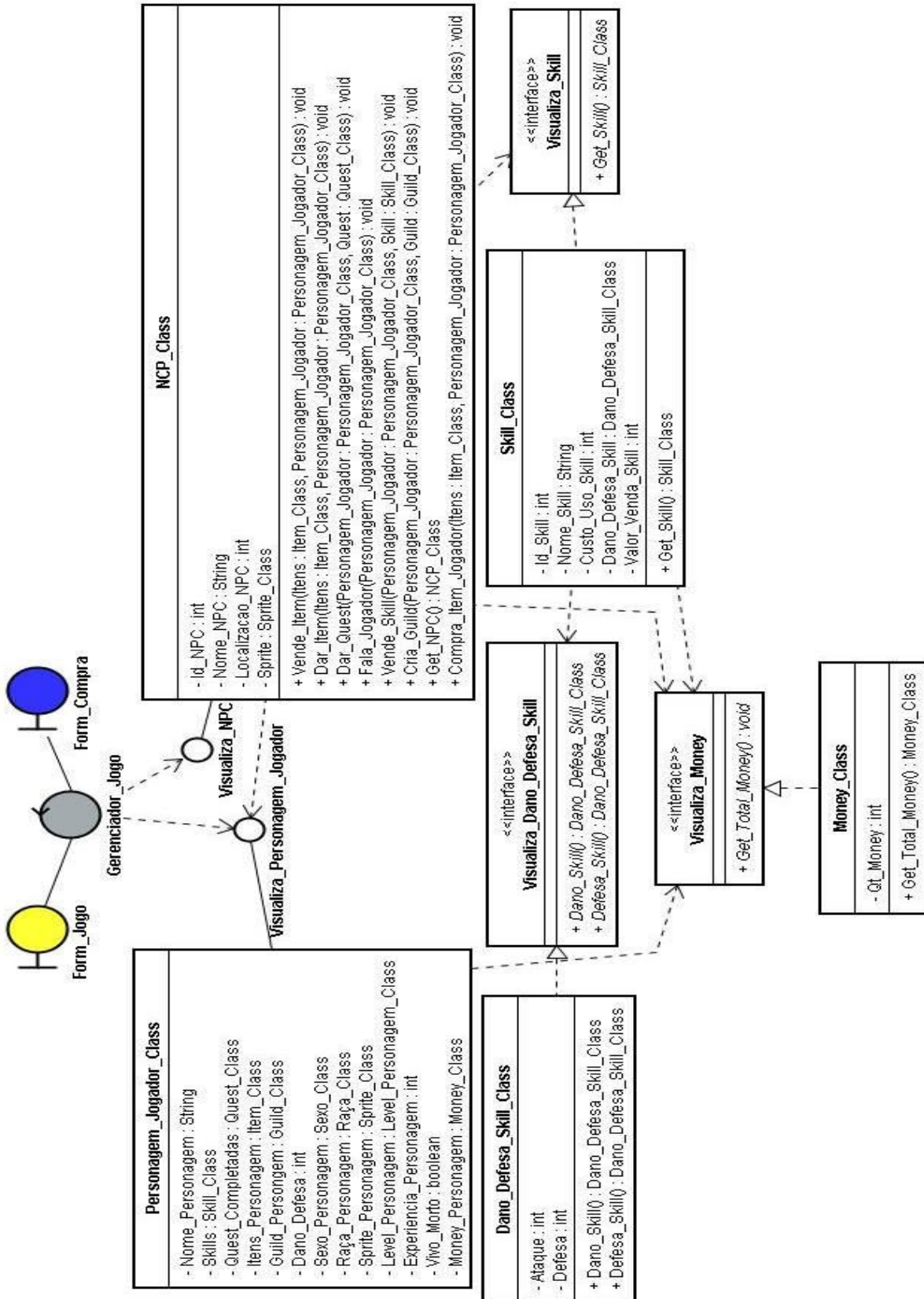


Figura 49: Diagrama Classe com Interface Compra Skill.

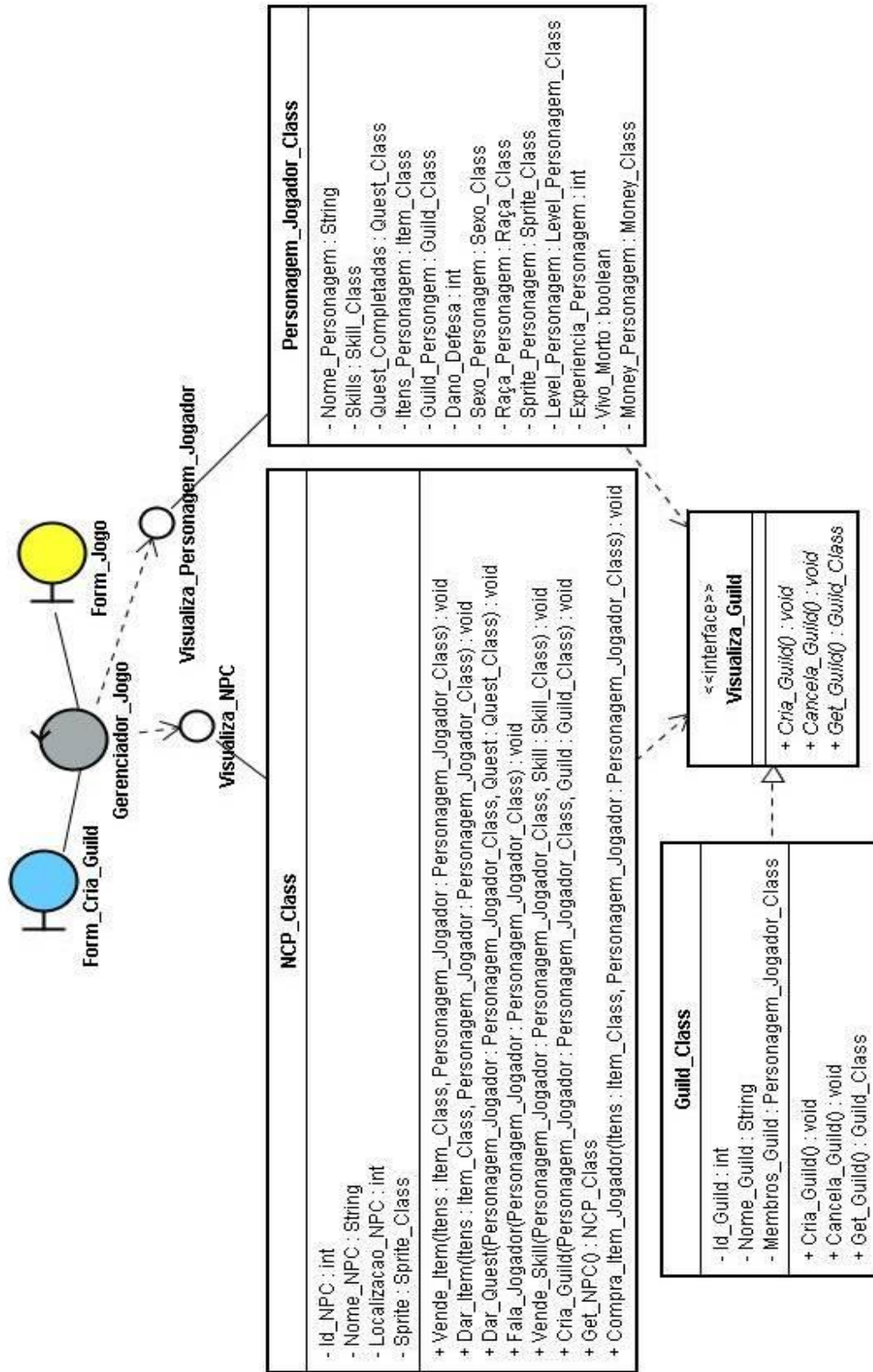


Figura 50: Diagrama Classe com Interface Cria Guild.

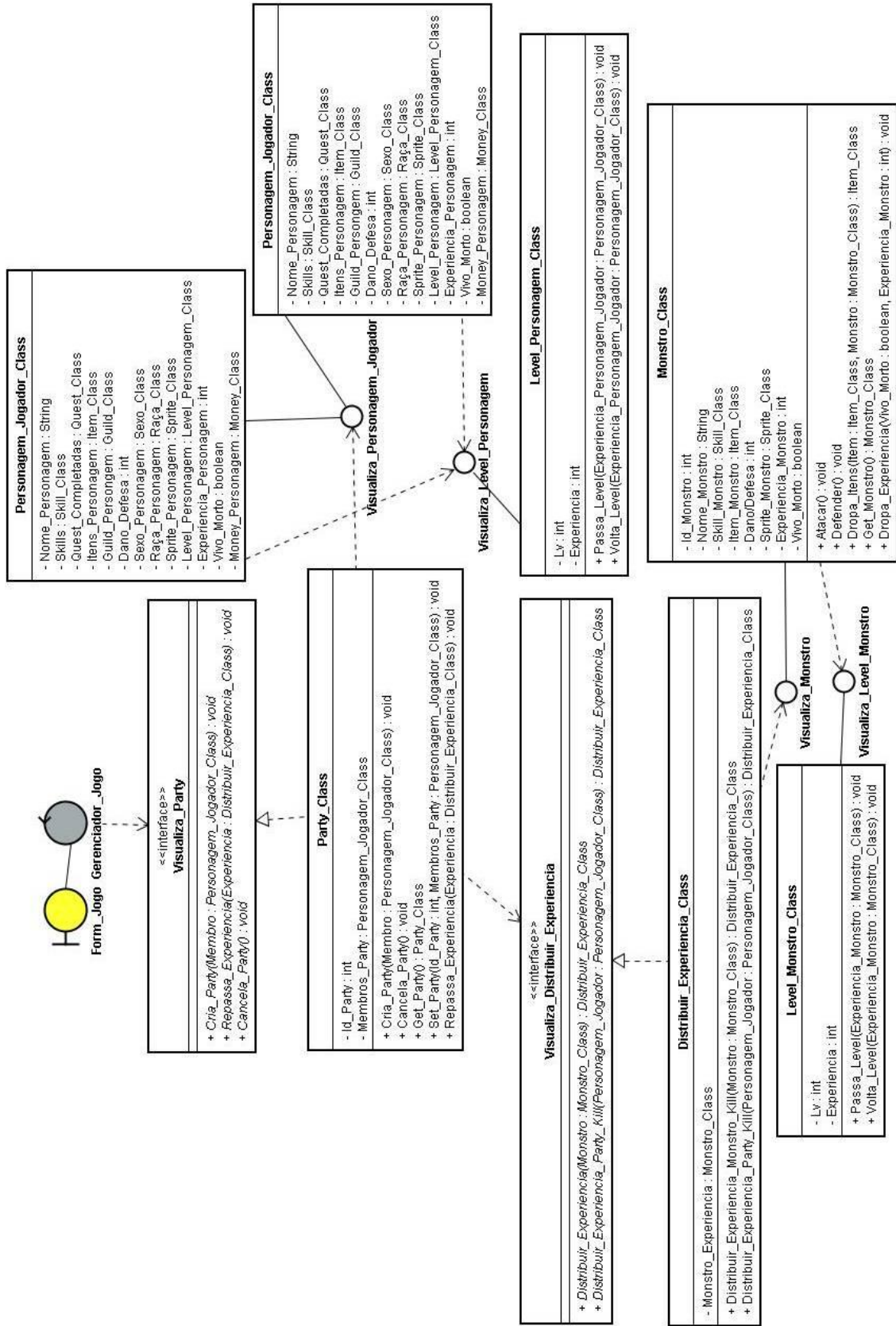


Figura 51: Diagrama Classe com Interface Cria Party.

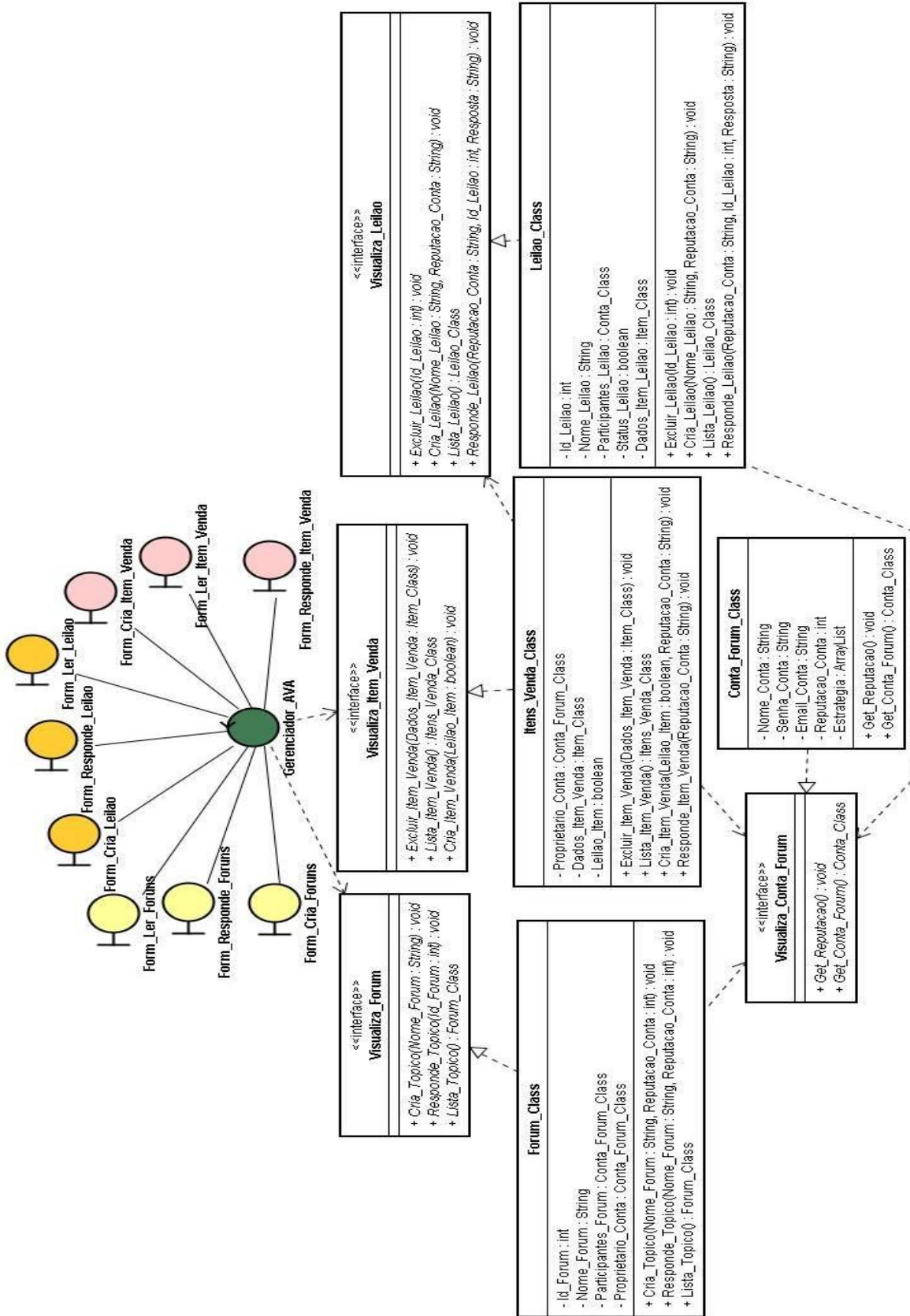


Figura 52: Diagrama Classe com Interface Fóruns, Leilão e Itens a Venda.

APÊNDICE C – Diagrama de Componentes com Implantação

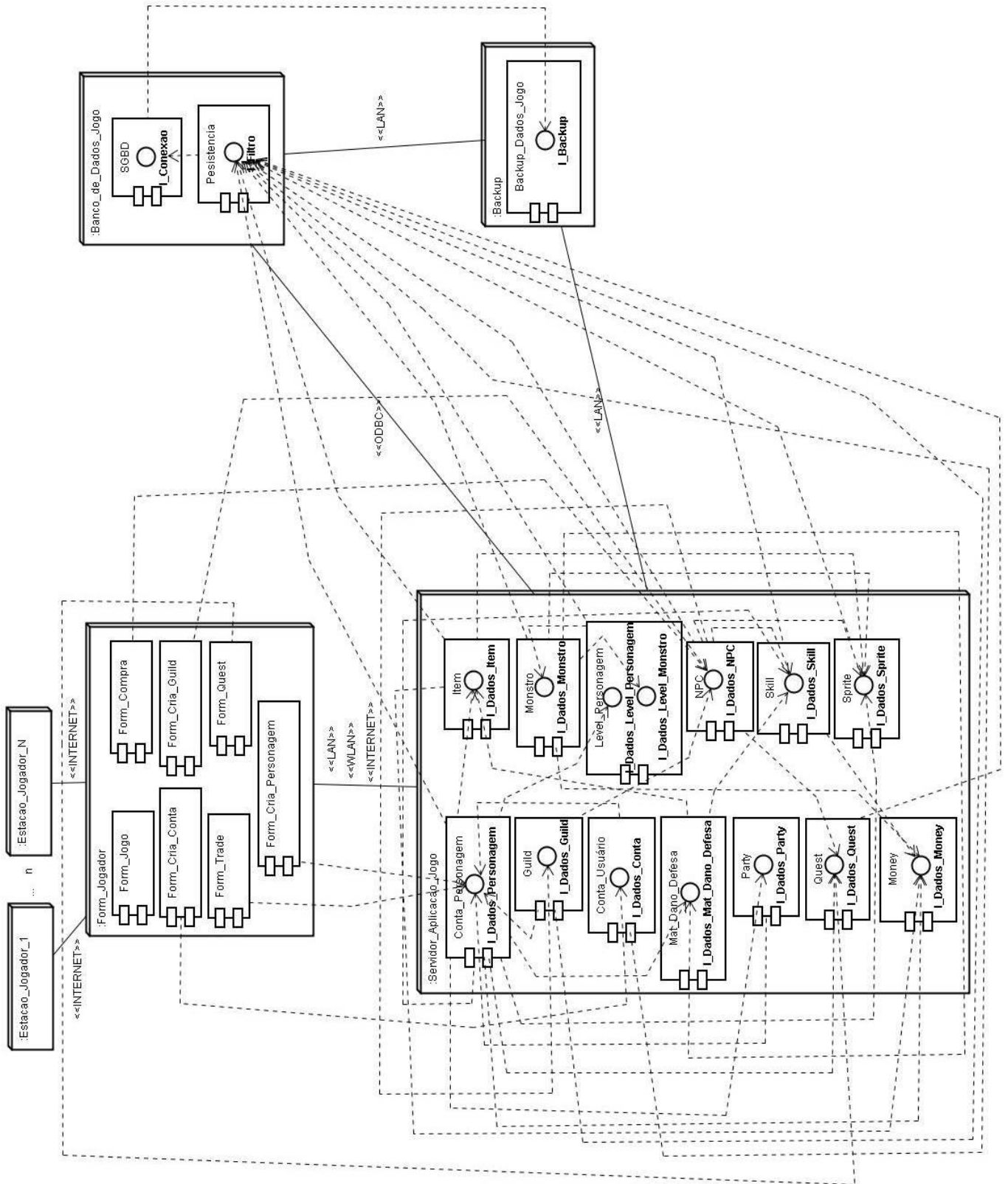


Figura 53: Diagrama Componentes com Implantação do jogo.

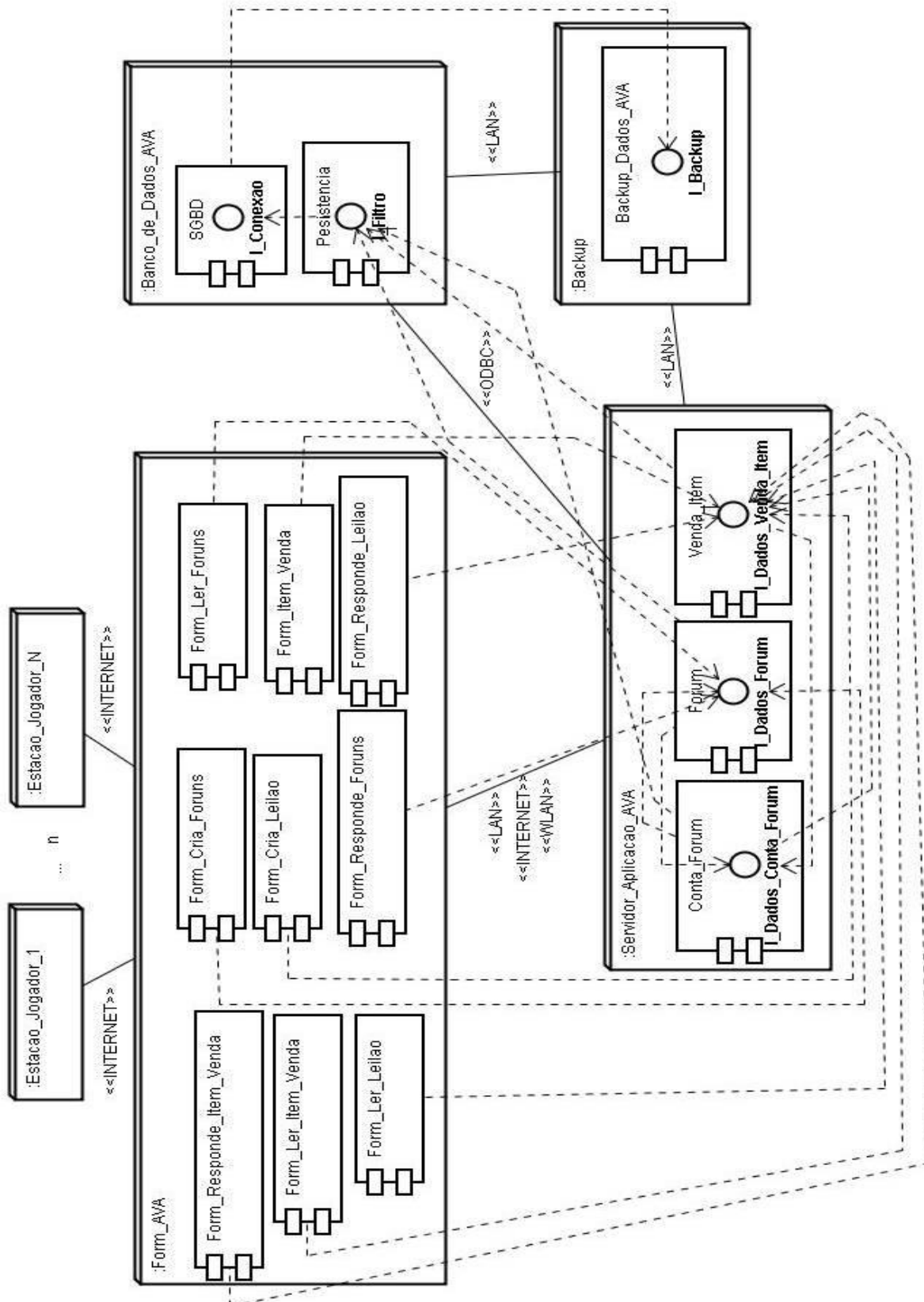


Figura 54: Diagrama Componentes com Implantação do ambiente virtual.

APÊNDICE D – Princípios da Persuasão aplicados ao MMORPG utilizando Redes de Petri

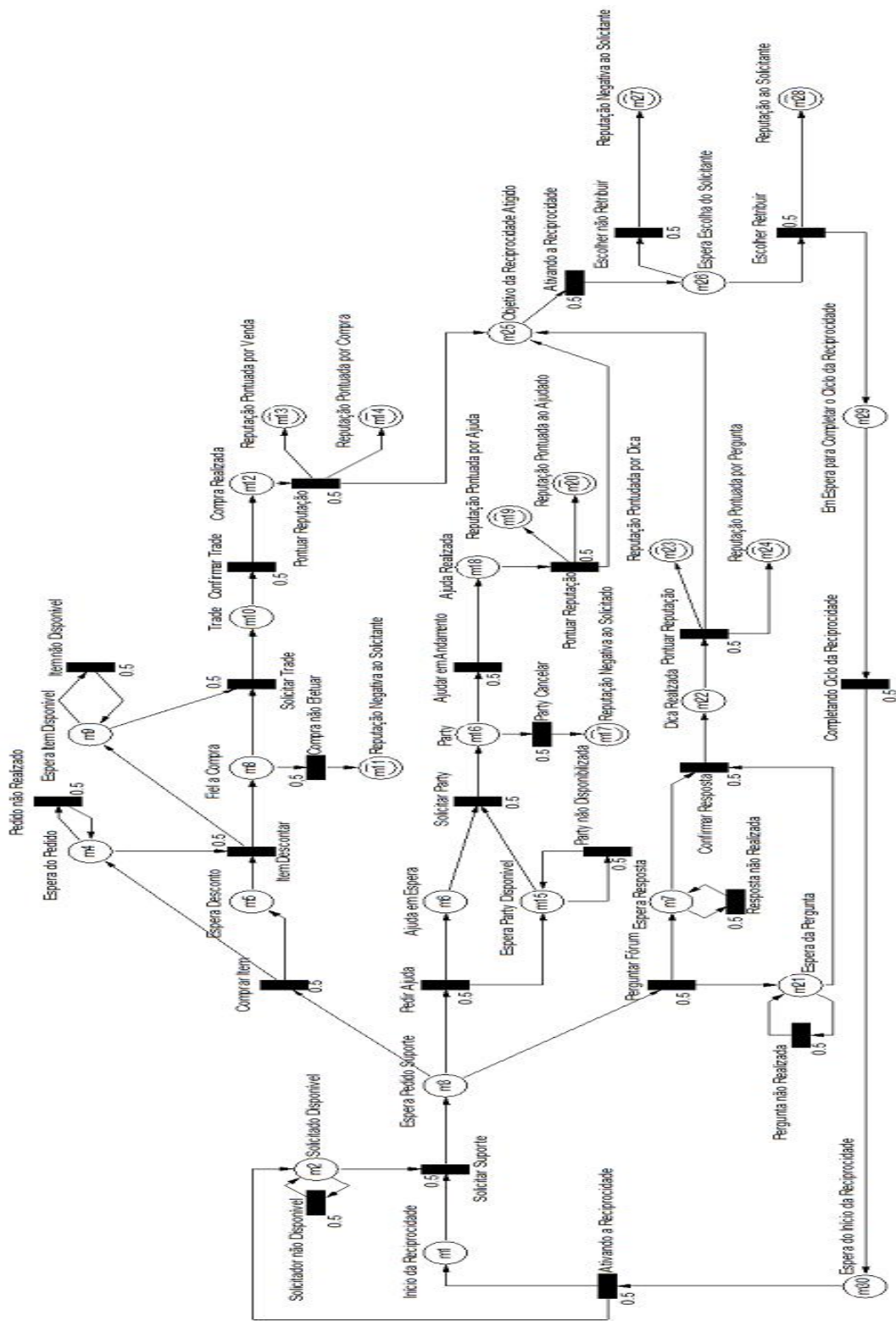


Figura 55: Reciprocidade na Rede de Petri.

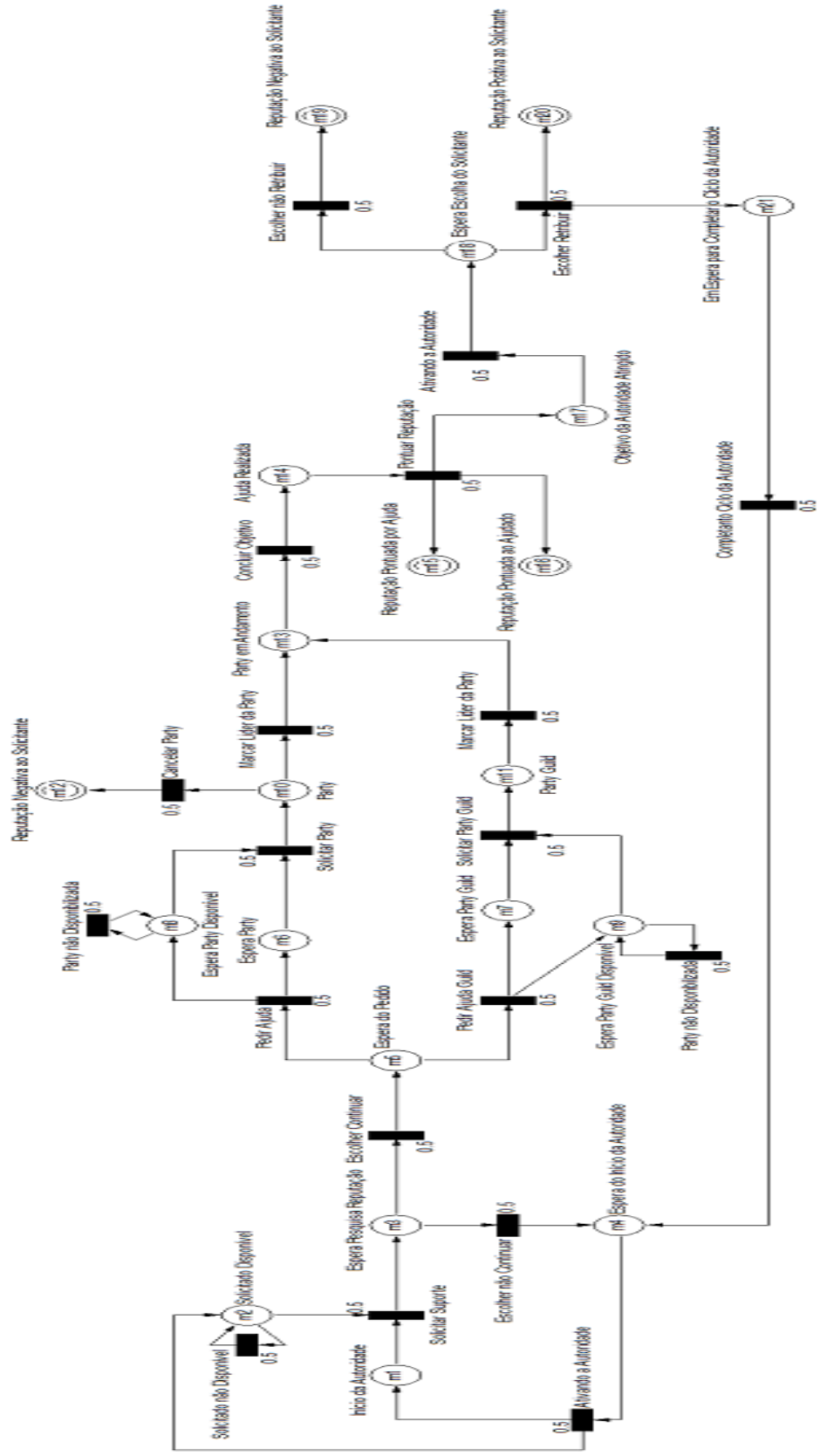


Figura 57: Autoridade na Rede de Petri.

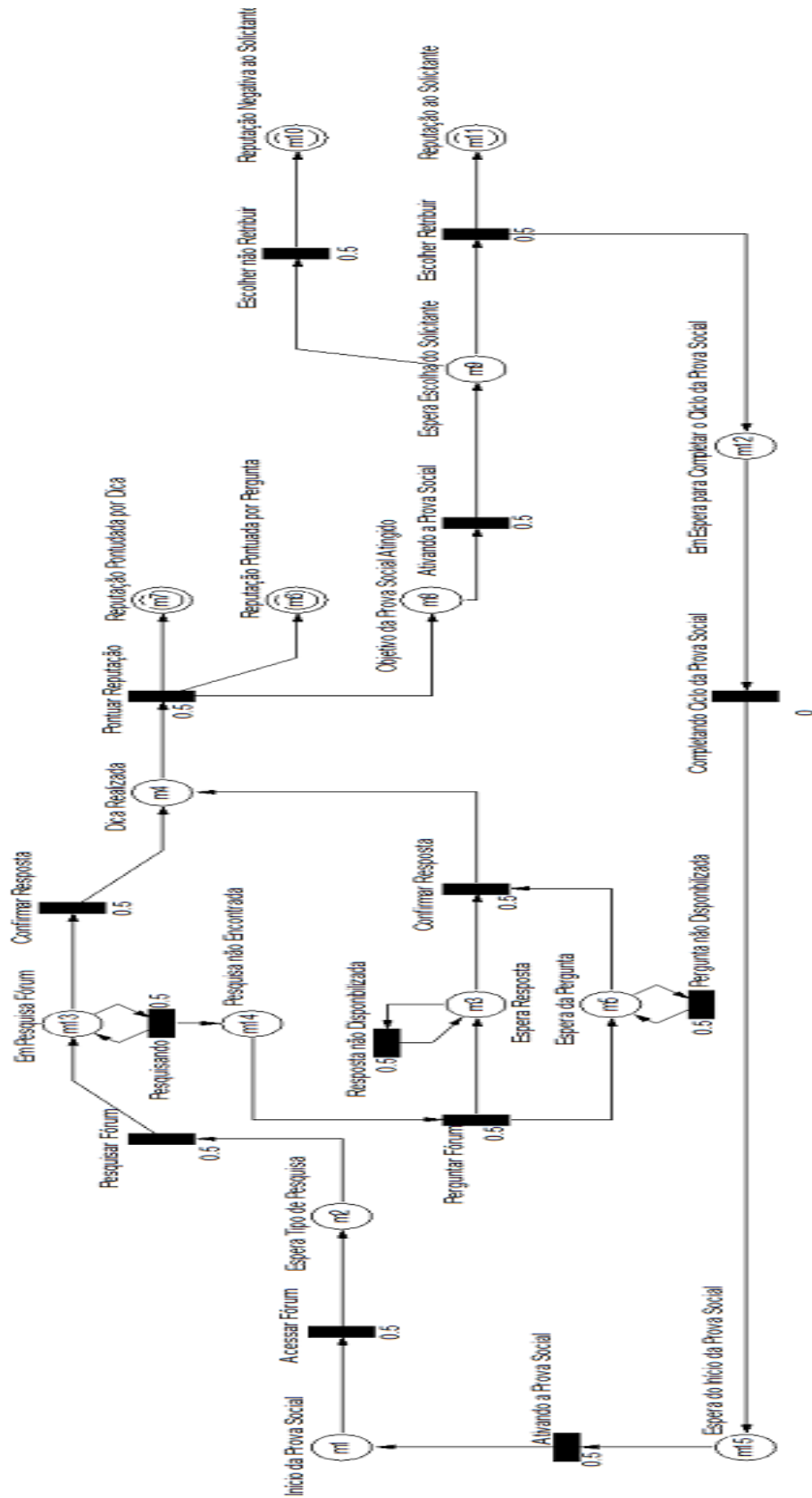


Figura 58: Prova Social na Rede de Petri.

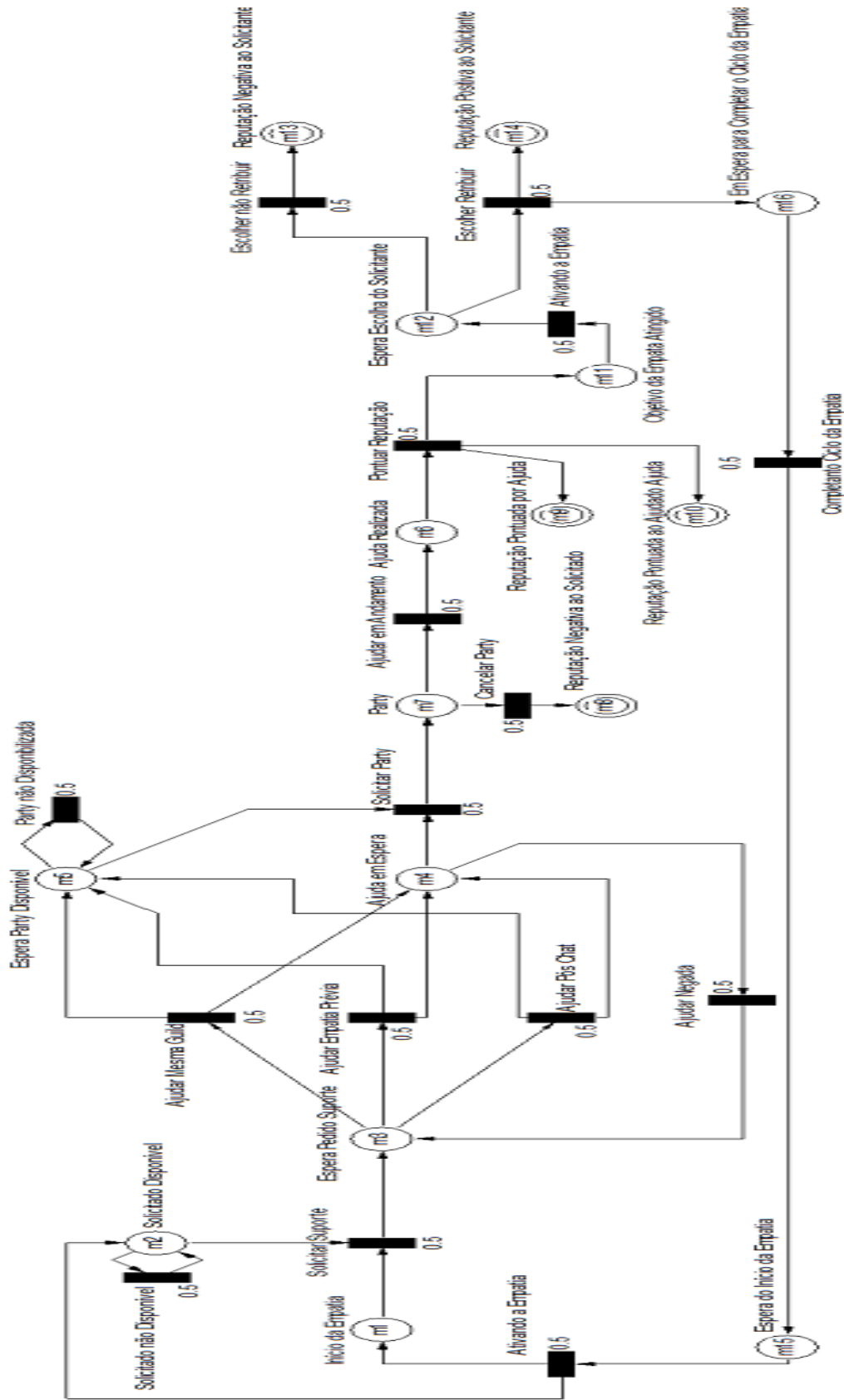


Figura 60: Empatia na Rede de Petri.

APÊNDICE E – Rede de Petri

O modelo proposto foi desenvolvido na ferramenta *Visual Objekt Net++*, baseado no formalismo matemático da Rede de Petri. Logo faz-se necessário compreender como um sistema computacional é modelado nesta ferramenta. Este capítulo apresenta um breve resumo deste formalismo e sua implementação na ferramenta.

As redes de Petri (MACIEL *et al.*, 1996) foram inicialmente definidas em 1962 por Carl Adam Petri, em seu trabalho de doutorado *Kommunikation Mit Automaten* na Universidade de Dramsdadt, na antiga Alemanha Ocidental. No seu trabalho, abordava-se um modelo matemático capaz de modelar sistemas paralelos, concorrentes, assíncronos e não-determinísticos. Pode-se considerar que as redes são uma evolução da teoria dos autômatos finitos (PAES, 2005), uma vez que essas redes, diferentemente destes últimos, são capazes de modelar o paralelismo dos sistemas. Por ser gráfica, facilita a compreensão e interpretação do funcionamento de sistemas. Uma rede, como é vista na Figura 61, é formada por dois elementos principais:

- Os lugares representam os elementos passivos da rede. Os lugares concebem os estados que um sistema pode assumir durante o seu fluxo de execução. Para tal, os lugares apresentam certos elementos denominados marcas ou *tokens*, os quais indicam a presença ou disponibilidade de algum recurso. Nas disposições destas marcas, definem-se o estado que se encontra a rede. Graficamente, os lugares são representados por círculos ou elipses e as marcas por pontos dentro dos lugares;
- As transições correspondem os elementos ativos da rede, pois são as responsáveis por dar a característica do dinamismo da mesma. Quando certas pré-condições são satisfeitas, uma transição passa do estado de desabilitada para habilitada e pode disparar. Já, o disparo das transições é responsável pela criação e pela destruição dos

recursos (*tokens*) da rede. Graficamente, são representadas por retângulos dispostos horizontalmente ou verticalmente;

- Os Arcos são responsáveis por estabelecer as conexões entre lugares e transições ou vice-versa, mas nunca de lugar para lugar ou de transição para transição.

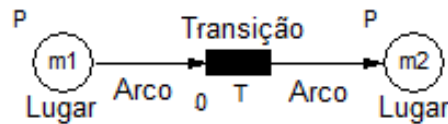


Figura 61: Elementos básicos de uma rede de Petri.

Em outras palavras, uma Rede de Petri é um grafo bipartido contendo dois tipos de nós (lugares e transições) e, que contém um estado inicial chamado de marcação inicial, M_0 . Os arcos partem de um lugar para uma transição, ou de uma transição para um lugar. Arcos orientados são identificados com seus pesos (inteiros positivos), onde um arco de peso K pode ser interpretado como um conjunto de K arcos em paralelo. A identificação para o peso unitário é usualmente omitido. A marcação (estado) relacionada a cada lugar é definido por um número inteiro não negativo de fichas neste lugar.

Formalmente, uma Rede de Petri é uma *quintuple* $PN = (P, T, F, W, M_0)$

(MURATA, 1989) onde:

- $P = \{p_1, p_2, \dots, p_m\}$ é um conjunto finito de lugares;
- $T = \{t_1, t_2, \dots, t_n\}$ é um conjunto finito de transições;
- $F \subseteq (P \times T) \cup T \times P$ é um conjunto de arcos orientados;
- $W : F \rightarrow \{1, 2, 3, \dots\}$ é a função peso;
- $M_0 : P \rightarrow \{1, 2, 3, \dots\}$ é a marcação inicial;
- $P \cap T = \emptyset$ e $P \cup T \neq \emptyset$.

Uma estrutura de Rede de Petri sem qualquer especificação de marcação inicial é dada por N , onde $N = (P, T, F, W)$. Uma Rede de Petri, com a marcação inicial definida, é dada pelo par (N, M_0) . Para a Rede de Petri simular o

comportamento dinâmico de um sistema, junto com seus estados e marcações, a rede necessita das regras de transição ou disparo:

- Uma transição t é dita habilitada se todos os lugares de entrada p de t estão marcados com no mínimo $W(p,t)$ fichas, onde $W(p,t)$ é o peso dos respectivos arcos de p para t ;
- Uma transição habilitada pode ou não disparar;
- Um disparo de uma transição habilitada t remove $W(p,t)$ fichas de cada lugar de entrada p de t , e adiciona $W(p,t)$ fichas para cada lugar de saída p de t , onde $W(p,t)$ é o peso do arco de t para p .

Um par de arcos entre um lugar p e uma transição t pode ser chamado de laço, se p é duplamente um lugar de entrada e um lugar de saída de t . Uma Rede de Petri é dita pura se não contém laços. Uma Rede de Petri é dita ordinária se todos os pesos dos arcos são iguais a 1.

Uma Rede de Petri (N, M_0) é dita limitada (ou de capacidade finita) se nenhum lugar p não ultrapassa um número máximo de fichas, isto é, se $M(p) \leq k$, para todo p . Neste caso, para uma transição t ser habilitada, é necessário também que o número de fichas de cada lugar de saída p de t não exceda a capacidade $K(p)$ após o disparo de t . Esta regra de transição é chamada de regra de transição estrita.

1.1.1. Redes de Petri Estendidas

As Redes de Petri Estendidas, por outro lado, correspondem a modelos para os quais as regras de transição sofrem algumas variações com a finalidade de aumentar a capacidade de representação do modelo. Aqui se podem considerar três tipos de subclasses (MURATA, 1989):

- As que têm o poder de representação de máquinas Turing (Redes de Petri com arcos inibidores);
- As que permitem a modelagem de Redes de Petri Híbridas e Redes de Petri contínuas;

- As que correspondem a modelos que descrevem o funcionamento de sistemas cuja evolução vai depender de eventos externos e/ou da variável tempo (Redes de Petri sincronizadas, Redes de Petri temporizadas e Redes de Petri estocásticas).

1.1.2. Propriedades da Rede de Petri

Existem dois tipos de propriedades que podem ser estudadas pelas Redes de Petri: as propriedades comportamentais (que dependem da marcação inicial) e as propriedades estruturais (não dependem da marcação inicial).

As propriedades comportamentais mais comumente usadas são:

- Alcançabilidade: uma marcação M_n é dita alcançável a partir de M_0 se existe uma seqüência de disparos que transforma M_0 em M_n . Neste caso, M_n é alcançável a partir de M_0 pela seqüência de disparos σ e denota-se $M_0 [\sigma > M_n$;

- Limitação: uma Rede de Petri é dita k-limitada ou simplesmente limitada se o número de fichas em cada lugar nunca excede um número finito k para qualquer marcação alcançável a partir de M_0 . Uma Rede de Petri (N, M_0) é dita segura se é 1-limitada.

- Vivacidade: uma Rede de Petri é dita viva se apresenta a ausência completa de bloqueios na operação do sistema. Ou seja, uma rede (N, M_0) é dita viva se, independentemente da marcação alcançável a partir de M_0 , qualquer transição da rede for possível de ser disparada, partindo desta marcação através de uma determinada seqüência de disparos. Porém, esta propriedade é muito forte para ser verificada. Por este motivo, em (MURATA, 1989), define-se diferentes níveis de vivacidade. Assim, dada uma Rede de Petri (N, M_0) , uma transição $t \in T$ é dita:

- Morta (L0-viva), se t não aparece em nenhuma seqüência de disparo de $L(M_0)$;

- Potencialmente disparável (L1-viva), se t aparece ao menos uma vez em alguma seqüência de disparo de $L(M_0)$;
- L2-viva, se, dado um número $k \in \mathbb{N}_+, k > 1$, t aparece ao menos k vezes em alguma seqüência de disparo de $L(M_0)$;
- L3-viva, se t aparece infinitas vezes em alguma seqüência de disparo de $L(M_0)$;
- L4-viva ou simplesmente viva se t é L1 viva para cada marcação M em $R(M_0)$. Este é o nível de vivacidade mais forte e corresponde ao conceito de vivacidade expresso inicialmente;
- Reversibilidade: uma Rede de Petri (N, M_0) é dita reversível se, para cada marcação M em $R(M_0)$, M_0 é alcançável a partir de M ;
- Distância Síncrona: define-se a distância síncrona entre duas transições t_1 e t_2 de uma Rede de Petri (N, M_0) por $d_{12} = \max|\sigma(t_1) - \sigma(t_2)|$, onde σ é uma seqüência de disparo partindo de uma marcação qualquer M em $R(M_0)$ e $\sigma(t_i)$ é o número de vezes que a transição t_i dispara em σ .

São ditas propriedades estruturais das Redes de Petri as propriedades que não dependem da marcação inicial, mas somente da estrutura topológica da rede. As propriedades estruturais são:

- Vivacidade Estrutural: uma Rede de Petri N é dita estruturalmente viva se existe uma marcação inicial viva para N ;
- Controlabilidade: uma Rede de Petri é dita completamente controlável se qualquer marcação é atingível a partir de uma dada marcação;
- Limitação Estrutural: uma Rede de Petri N é dita limitada estruturalmente se é limitada para toda marcação inicial M_0 ;
- Conservabilidade: uma Rede de Petri N é dita parcialmente conservativa, se existe um inteiro positivo $y(p)$ para algum lugar p , tal que a soma ponderada de fichas (*tokens*) seja constante, isto é, $M^T y = M_0^T y = cte$;

de forma análoga, uma Rede de Petri N é dita totalmente conservativa, se existe um inteiro positivo $y(p)$ para cada lugar p , tal que a soma ponderada de fichas (*tokens*) seja constante, isto é, $M^T y = M_0^T y = cte$;

- **Consistência:** uma Rede de petri é dita parcialmente consistente se existe uma marcação inicial M_0 e uma seqüência de disparos σ que leva ciclicamente a M_0 de modo que (alguma) cada transição ocorre pelo menos uma vez em σ .

O comportamento dinâmico de grande parte dos sistemas estudados pela engenharia pode ser descrito usando equações diferenciais. No caso de sistemas modelados por Rede de Petri, este comportamento dinâmico também é regido por sua equação de estado. Na próxima seção, será apresentada a representação algébrica da Rede de Petri, ou seja, a equação de estado.

1.1.3. Equação de Estado

Uma parte fundamental da equação de estado é o que se denomina como matriz de incidência (MURATA, 1989). A matriz de incidência representa a estrutura de uma rede. Para uma Rede de Petri N com n transições e m lugares, a matriz de incidência $A = [a_{ij}]$ é uma matriz $n \times m$ dada por $A = A^+ - A^-$.

A matriz A^+ de ordem $n \times m$ representa a quantidade de fichas adicionadas aos lugares após o disparo de uma transição, sendo que, $A^+ = [a_{ij}^+]$, onde, $a_{ij}^+ = w(i, j)$ é o peso do arco que leva da transição i ao lugar j .

A matriz A^- de ordem $n \times m$ representa a quantidade de fichas retiradas dos lugares após o disparo de uma transição, sendo que, $A^- = [a_{ij}^-]$, onde, $a_{ij}^- = w(i, j)$ é o peso do arco que leva do lugar j a transição i .

A equação de estado de uma rede é dada pela seguinte equação:

$$M_{k+1} = M_k + A^t v_k$$

Onde:

- M_{k+1} e M_k são respectivamente vetores colunas de $m \times 1$, que representam as marcações sucessora e atual, respectivamente;
- v_k é o vetor coluna de $n \times 1$ que representa o vetor de habilitação ou disparo.

Se considerarmos que uma marcação M_d é alcançável a partir de M_0 através da seqüência de disparo $\{v_1, v_2, \dots, v_d\}$, pode-se escrever a equação de estado para:

$$M_d = M_0 + A^t \sum_{k=1}^d v_k$$

Que pode ser reescrita da seguinte forma:

$$A^t x = \Delta M$$

Onde $\Delta M = M_d - M_0$ e $x = \sum_{k=1}^d v_k$.

O i -ésimo componente do vetor x denota a quantidade de vezes que a i -ésima transição deve ser disparada até alcançar a marcação M_d .

Existem outras propriedades nas redes que podem ser calculadas a partir da matriz de incidência. A solução da equação homogênea $Ay = 0$ é chamada de invariantes de lugar ou p -invariantes. Da mesma forma, a solução da equação homogênea $A^t x = 0$ é chamada de invariantes de transição ou t -invariantes. As soluções p -invariantes e t -invariantes são usadas na análise estrutural das Redes de Petri para a verificação das propriedades: limite estrutural, conservatividade, repetitividade e consistência.

1.1.4. Refinamento na Rede de Petri

A técnica de refinamento consiste em substituir uma transição ou um lugar por uma rede. Um conceito relacionado com o refinamento é a redução que consiste em substituir um subconjunto de elementos por um lugar ou uma transição, (MURATA, 1989), e apresenta um conjunto de regras de redução que

preservam propriedades como vivacidade, segurança e limitação. A Figura 62 ilustra estas transformações.

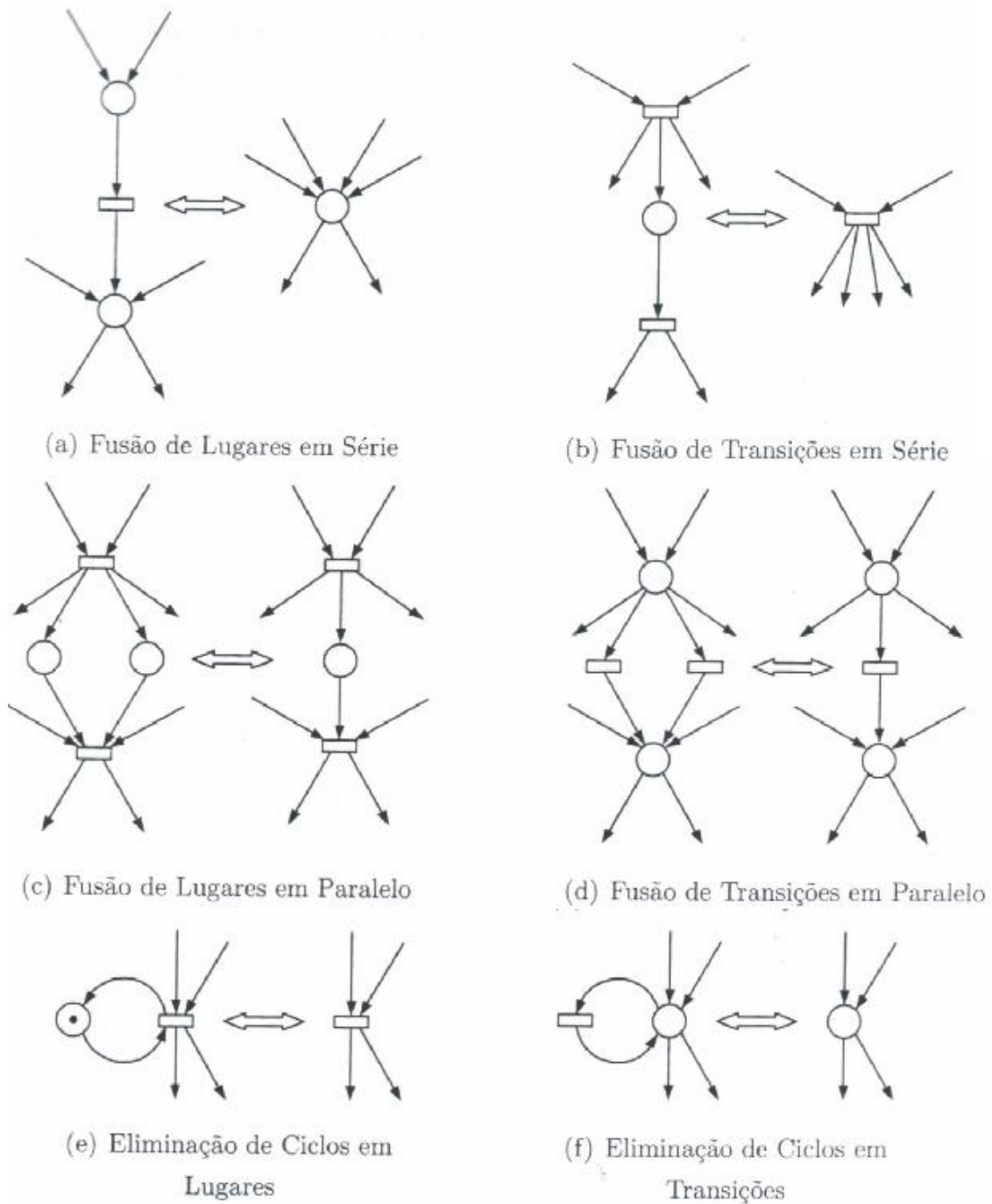


Figura 62: Transformações preservando vivacidade, segurança e limitação (MURATA, 1989).

Outra maneira que facilita o refinamento da Rede de Petri é a simulação. Com a simulação, pode-se mencionar as seguintes vantagens:

- Efeitos das animações e gráficos ajudam no aprendizado e entendimento do sistema simulado;
- Permite manter um maior controle sobre o experimento, o que muitas vezes não é possível no sistema real;
- Permite estudar o sistema durante o longo período de tempo simulado.

No tópico a seguir, será descrito o simulador *Visual Object Net ++*, desenvolvido por um grupo de pesquisadores alemães, haja vista este simulador ter sido utilizado para modelar as redes que serão apresentadas no Capítulo 5.

1.1.5. Ferramenta de Simulação *Visual Object Net ++*

Atualmente, é muito difícil encontrar uma ferramenta de simulação de Redes de Petri que dê suporte a todas as variações encontradas na Rede de Petri. Algumas ferramentas se limitam a simular apenas Rede de Petris Ordinárias, outras apenas Rede de Petris Estocásticas e etc.

O simulador *Visual Object Net ++* é bastante simples e oferece uma interface fácil de usar. Suas características básicas são:

- Suporta Redes de Petri Temporizadas;
- Suporta elementos discretos e contínuos;
- Não suporta Redes de Petri Coloridas e Estocásticas;
- Análise simples de desempenho;
- Configuração de velocidade de simulação;
- Possui um editor gráfico;
- Animação de *tokens*.

Este simulador foi desenvolvido para máquinas com sistema operacional Windows. Para instalar, basta descompactar o arquivo obtido através da *homepage* do simulador (DRATH, 2004).

Hoje em dia, é possível encontrar simuladores mais completos, porém, apenas oferecidos no comércio, não estando disponibilizados gratuitamente.

1.1.5.1. Utilizando o Visual Object Net ++

O principal objetivo deste tópico é apresentar o *Visual Object Net ++* (DRATH, 2004), algumas de suas interfaces e como simular uma Rede de Petri.

Este simulador apresenta suporte a elementos discretos e contínuos. Não menos importante, mas para o melhor entendimento do simulador, exigem-se algumas observações a respeito:

- Transição contínua: Seu disparo pode estar associado a um valor constante ou a uma função;
- *Place* contínuo: Possui *tokens* com valores reais.

Na Figura 63, ilustra-se a interface principal o *Visual Object Net ++*, dividida em quatro módulos:

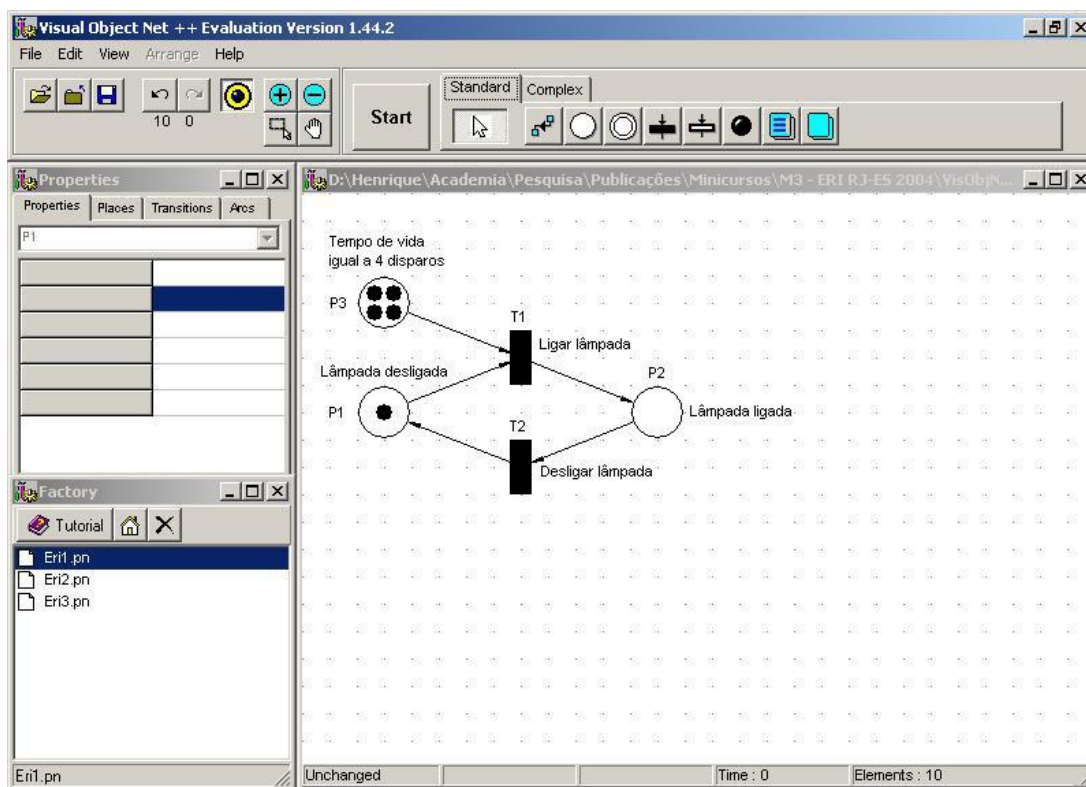


Figura 63: Interface Visual Object Net ++.

A Figura 64 e Figura 65 ilustram o módulo *Properties*. Marcando um elemento no módulo de edição, as propriedades deste aparecerão conforme ilustrado na Figura 64. Se forem necessárias informações a respeito de todos os

elementos da Rede de Petri, basta escolher uma das opções relativas a cada elemento, conforme é demonstrado na Figura 65.

Através do módulo de propriedades, permite-se alterar os seguintes parâmetros para cada elemento:

- *Places*: Nome, variável (pode ser usada em funções), tipo (discreto ou contínuo), quantidade de *tokens* iniciais, visibilidade do rótulo e tamanho;
- *Transições*: Nome, atraso (temporização ou função no caso de elemento contínuo), tipo (discreto ou contínuo), cor, visibilidade do rótulo, tamanho, prioridade e *Reservation*;
- *Arcos*: Peso, visibilidade do rótulo e tipo (*normal*, *inhibitor* e *static test arc*).

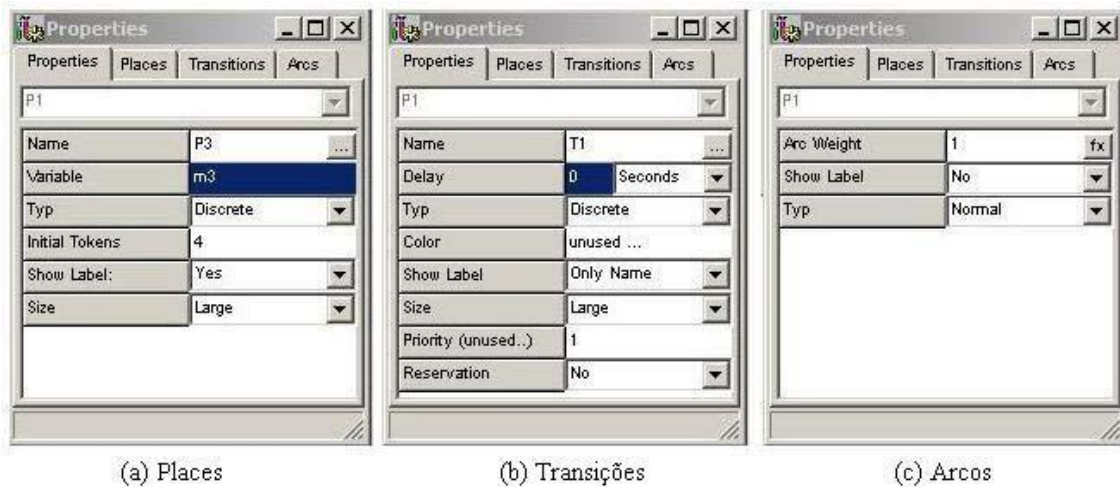
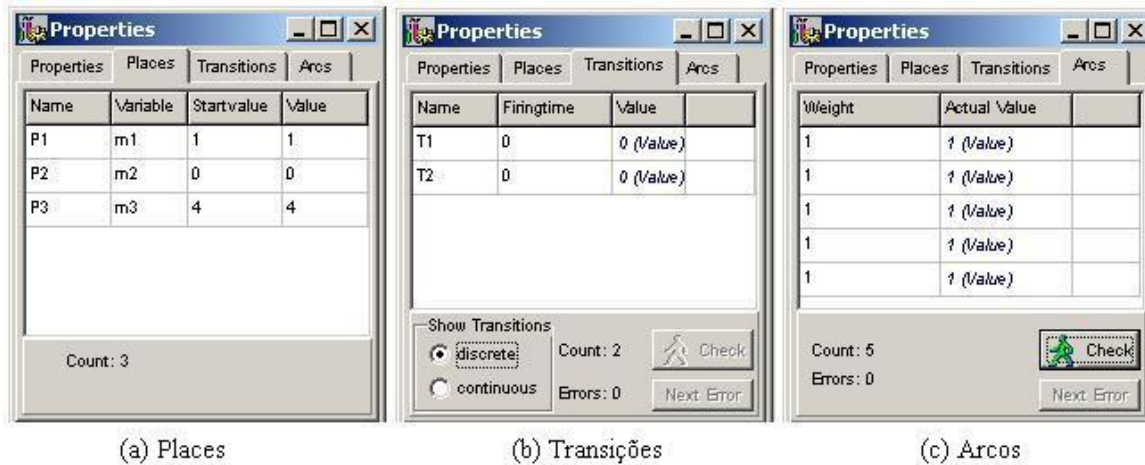


Figura 64: Interface das propriedades dos places, transições e arcos.



(a) Places

(b) Transições

(c) Arcos

Figura 65: Interface de informações relativas aos places, transições e arcos.

A Figura 66 apresenta a opção para início de uma simulação.

Entre os botões de acesso rápido que existem no primeiro módulo de interface, existe o botão *Start*. Este botão abre uma nova interface de botões para gerenciar a simulação. A função de cada botão é a seguinte:

- *Step*: Usando este botão, a Rede de Petri será simulada através de passos. Será sempre necessária a intervenção do usuário apertando este botão. Somente as transições habilitadas serão disparadas a cada passo;
- *Run*: A Rede de Petri será simulada sem prazo de tempo definido para parar. Enquanto houver uma transição habilitada para disparo a rede continuará sendo simulada;
- *Run to event*: É necessário um evento provocado pelo usuário. A inserção de um *token* em um determinado *place* provoca o início da simulação;
- *Stop*: Pára a simulação. Este botão não impede a continuação da simulação a partir do estado em que foi finalizada.

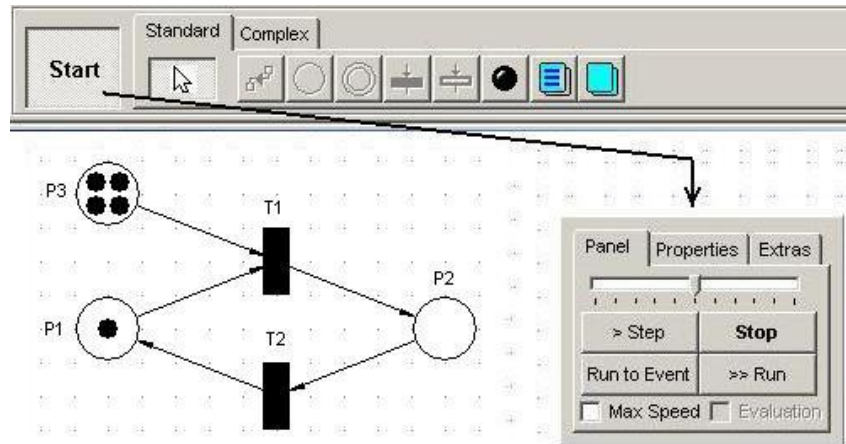


Figura 66: Simulando a execução de uma Rede de Petri.

Na Figura 67, explana-se o módulo de interface de opções. Neste módulo, é possível encontrar campos de configuração (os dois últimos campos ainda não foram projetados) referentes aos seguintes itens:

- Representação: Tipo de *grid*;
- Simulação: Tempo de simulação para mudança de um estado e tipo de animação;
- Performance: Análise simplificada de desempenho.

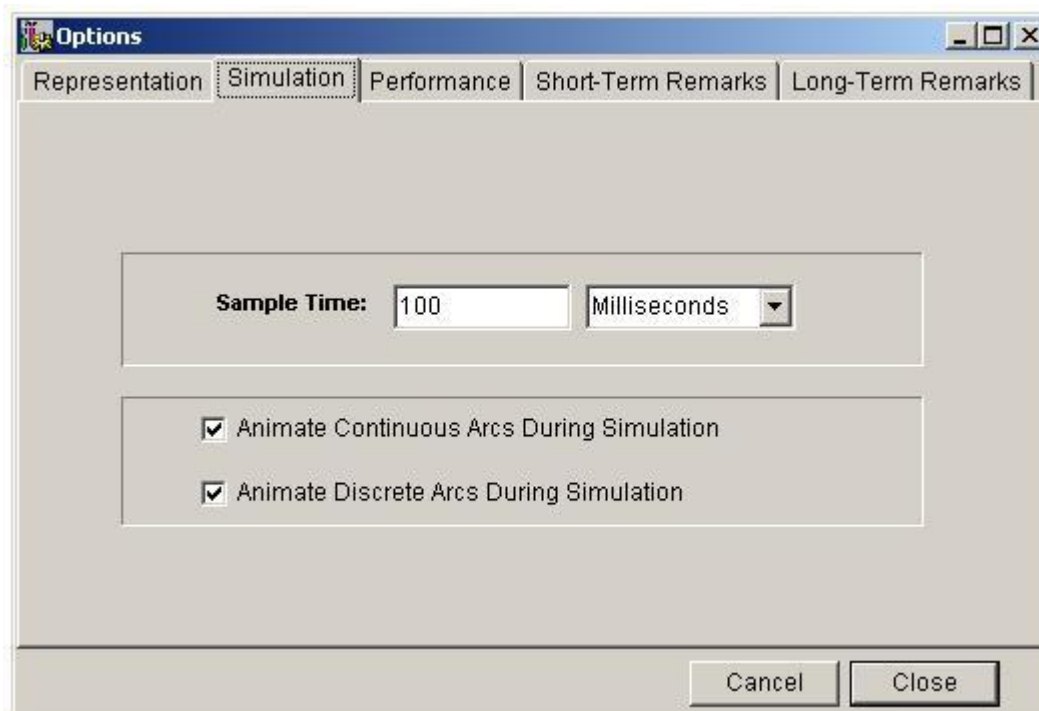


Figura 67: Interface de opções.

Prosseguido com a proposta de apresentação da ferramenta, será exemplificado o funcionamento da Rede de Petri, simulando o funcionamento de uma lâmpada, demonstrado na Figura 68:

- A rede é composta pelos estados *Ligada* e *Desligada*;
- A rede é composta pelas transições *Ligar* e *Desligar*;
- O estado inicial da rede é *Desligada*, no entanto, a rede apresenta um movimento cíclico, que, no caso, não apresenta estado final;
- Para cada transição ativada a lâmpada muda o seu estado;

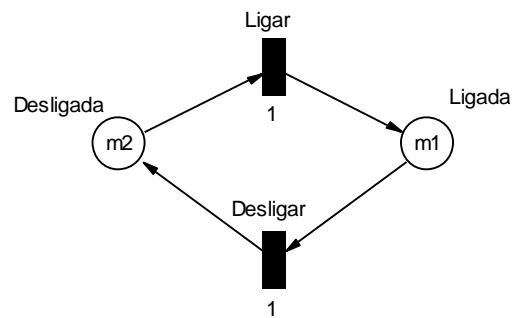


Figura 68: Rede de Petri Lâmpada.

1.1.6. Conclusão do Apêndice E

O apêndice E explanou os princípios matemáticos das Redes de Petris e a importância do uso destas redes para modelar sistemas computacionais. Com o intuito de simular estas redes, criou-se um simples tutorial do funcionamento da ferramenta utilizada, usando, por fim, um exemplo prático para explicar o funcionamento da ferramenta.