



Universidade do Estado do Rio de Janeiro

Centro de Tecnologia e Ciências

Instituto de Matemática e Estatística

Fernanda Teixeira Novo

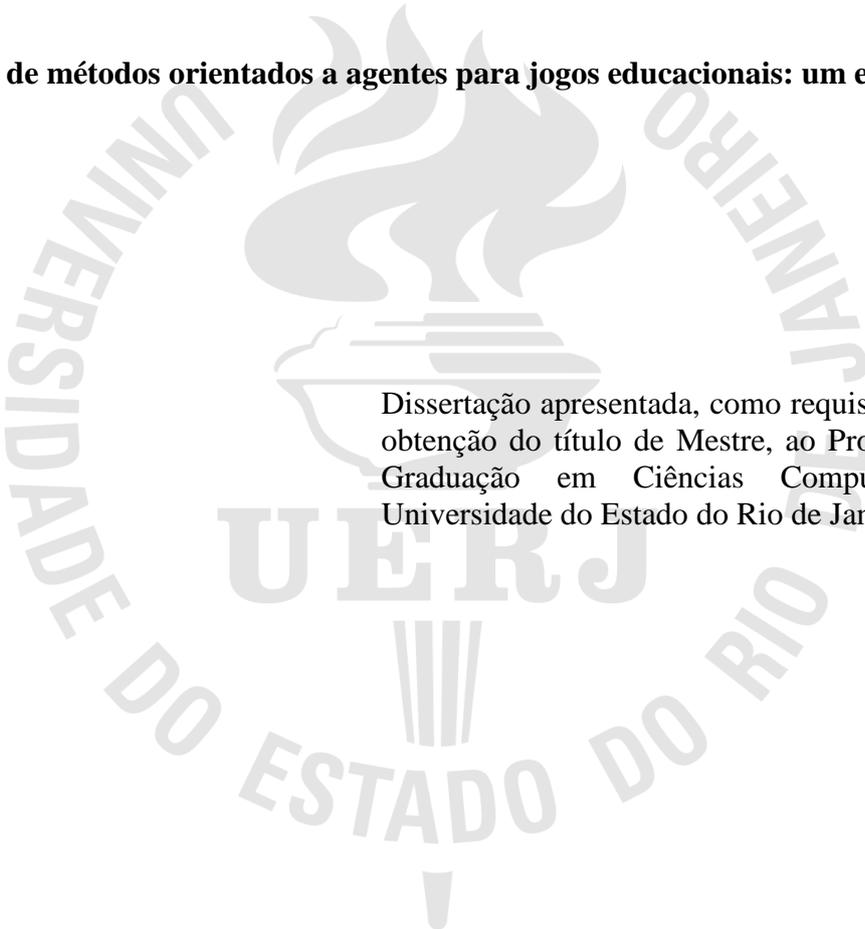
Aplicação de métodos orientados a agentes para jogos educacionais: um estudo de caso

Rio de Janeiro

2018

Fernanda Teixeira Novo

Aplicação de métodos orientados a agentes para jogos educacionais: um estudo de caso



Dissertação apresentada, como requisito parcial para obtenção do título de Mestre, ao Programa de Pós-Graduação em Ciências Computacionais, da Universidade do Estado do Rio de Janeiro.

Orientadora: Prof.^a Dra. Vera Maria Benjamim Werneck

Coorientadora: Prof.^a Dra. Rosa Maria E. Moreira da Costa

Rio de Janeiro

2018

CATALOGAÇÃO NA FONTE
UERJ / REDE SIRIRUS / BIBLIOTECA CTC-A

N944 Novo, Fernanda Teixeira.
Aplicação de métodos orientados a agentes para jogos educacionais: um estudo de caso / Fernanda Teixeira Novo. – 2018.
115f.: il.

Orientadora: Vera Maria Benjamim Werneck.
Coorientadora: Rosa Maria E. Moreira da Costa.
Dissertação (Mestrado Ciências Computacionais) - Universidade do Estado do Rio de Janeiro, Instituto de Matemática e Estatística.

1. Sistemas multiagentes - Teses. 2. Jogos educacionais - Teses. 3. Software - Desenvolvimento - Teses. I. Werneck, Vera Maria Benjamim. II. Costa, Rosa Maria E. Moreira da. III. Universidade do Estado do Rio de Janeiro. Instituto de Matemática e Estatística. IV. Título.

CDU 004.415.2

Rosalina Barros *CRB/7 - 4204* - Responsável pela elaboração da ficha catalográfica

Autorizo para fins acadêmicos e científicos, a reprodução total ou parcial desta dissertação, desde que citada a fonte.

Assinatura

Data

Fernanda Teixeira Novo

Aplicação de métodos orientados a agentes para jogos educacionais: um estudo de caso

Dissertação apresentada, como requisito parcial para obtenção do título de Mestre, ao Programa de Pós-Graduação em Ciências Computacionais, da Universidade do Estado do Rio de Janeiro.

Aprovada em 27 de agosto de 2018.

Banca Examinadora:

Prof.^a Dra. Vera Maria Benjamim Werneck (Orientadora)
Instituto de Matemática e Estatística - UERJ

Prof.^a Dra. Rosa Maria E. Moreira da Costa (Coorientadora)
Instituto de Matemática e Estatística - UERJ

Prof. Dr. Marcelo Schots de Oliveira
Instituto de Matemática e Estatística – UERJ

Prof. Dr. Eduardo Kinder Almentero
Universidade Federal Rural do Rio de Janeiro

Rio de Janeiro

2018

AGRADECIMENTOS

Agradeço, primeiramente, a Deus por colocar pessoas tão especiais em minha vida e sempre me proporcionar saúde e força para superar os obstáculos.

Aos meus pais, Ronaldo e Maria Christina, meu infinito agradecimento. Por sempre acreditarem na minha capacidade e estarem ao meu lado em todos os momentos.

Ao meu marido, Lucas, pelo incentivo e carinho nas horas difíceis, de desânimo e cansaço.

Ao meu irmão, Fillipe, pelos momentos de descontração entre as horas de estudo.

As minhas orientadoras Vera Werneck e Rosa Maria pela paciência e dedicação durante a elaboração da dissertação e do estudo de caso.

Aos queridos alunos da turma de inteligência artificial do segundo semestre de 2017 pela dedicação e determinação para alcançar excelência no estudo de caso.

Aos membros da banca examinadora por dedicarem parte de seu tempo em examinar e avaliar meu trabalho.

À Universidade do Estado do Rio de Janeiro por me proporcionar o aprendizado desde a graduação até o Mestrado. Pelo excelente ensino concedido a mim através de professores que fizeram grande diferença na minha formação.

Por vezes sentimos que aquilo que fazemos não é senão uma gota de água no mar. Mas o mar seria menor se lhe faltasse uma gota.

Madre Teresa de Calcutá

RESUMO

NOVO, Fernanda Teixeira. *Aplicação de métodos orientados a agentes para jogos educacionais: um estudo de caso*. 2018. 164 f. Dissertação (Mestrado em Ciências Computacionais) – Instituto de Matemática e Estatística, Universidade do Estado do Rio de Janeiro, Rio de Janeiro, 2018.

Métodos orientados a agentes têm sido utilizados no desenvolvimento de sistemas em diferentes áreas do conhecimento, apresentando vantagens e desvantagens. Entre algumas desvantagens destacam-se: grande esforço, inflexibilidade, alto custo, tempo e dificuldade para adaptar-se ao escopo dos problemas. O objetivo deste trabalho é conceituar e avaliar métodos orientados a agentes, que utilizam a proposta de desenvolvimento tradicional e ágil, no contexto de construção de um jogo médico educacional. Estes métodos foram utilizados na criação do jogo médico MEDEDUC por alunos do curso de Ciência da Computação, da turma de Inteligência Artificial de uma Universidade pública do Estado do Rio de Janeiro em 2017. Esse trabalho foi inspirado no estudo feito por um ex-aluno de mestrado, Ferreira (2016), que comparou outros métodos. Assim, agregaram-se novas evidências aos dados obtidos em seu estudo, formando uma ampla base para realizar comparações entre os métodos. Três métodos foram utilizados neste trabalho: i. INGENIAS, um método tradicional de desenvolvimento de sistemas multiagentes, e utiliza um processo de desenvolvimento tradicional; ii. INGENIAS Scrum, um método baseado no processo de desenvolvimento ágil para sistemas multiagentes e iii. AgilePASSI, um método baseado no processo de desenvolvimento ágil para sistemas multiagentes.

Palavras-chave: Sistemas Multiagentes. Métodos de Sistemas Multiagentes. Métodos Ágeis.

INGENIAS. AgilePASSI. INGENIAS Scrum.

ABSTRACT

NOVO, Fernanda Teixeira. *Application of agent-oriented methods for educational games: a case study*. 2018. 164 f. Dissertação (Mestrado em Ciências Computacionais) – Instituto de Matemática e Estatística, Universidade do Estado do Rio de Janeiro, Rio de Janeiro, 2018.

Agent-oriented methods have been used in the development of systems in different areas of knowledge presenting advantages and disadvantages. Some disadvantages include: great effort, inflexibility, high cost, time and difficulty to adapt to the scope of problems. The objective of this work is to conceptualize and evaluate agent-oriented methods that explore traditional and agile development methods to construct a medical educational game. These methods were used to implement the medical game MEDEDUC by Computer Science students of a class of Artificial Intelligence from the public university of the state of Rio de Janeiro in 2017. This work was inspired in a study done by a master's student, Ferreira (2016), who compared other methods. Thus, new evidences were added to the data obtained in his work, forming a broader basis for methods comparisons. Three methods were used in the present work: i. INGENIAS, is a traditional method of developing multi-agent systems and uses a traditional development process; ii. INGENIAS Scrum, is a method based on the agile development process for multi-agent systems and iii. AgilePASSI, is a method based on the agile development process for multi-agent systems.

Keywords: Multi-agent systems. Methods of Multi-agent systems. Agile Methods. INGENIAS. AgilePASSI. INGENIAS Scrum.

LISTA DE FIGURAS

Figura 1 –	Fases do Processo Ágil XP.....	27
Figura 2 –	Ciclo de vida de um projeto em Scrum.....	28
Figura 3 –	As interações de um agente genérico com seu ambiente.....	30
Figura 4 –	Fases do ciclo de vida para o método INGENIAS.....	33
Figura 5 –	Metamodelos do método INGENIAS.....	34
Figura 6 –	Elementos do modelo de organização.....	35
Figura 7 –	Elementos do modelo de organização (definição <i>Workflow</i>).....	36
Figura 8 –	Elementos do modelo de agentes.....	37
Figura 9 –	Elementos do modelo de metas e tarefas.....	38
Figura 10 –	Elementos do modelo de interação.....	39
Figura 11 –	Elementos do modelo de ambiente.....	40
Figura 12 –	<i>Preparation Phase workflow</i>	44
Figura 13 –	<i>Sprint Phase workflow</i>	45
Figura 14 –	Processo de desenvolvimento AgilePASSI.....	46
Figura 15 –	Grau de conhecimento por nível do jogo.....	91
Figura 16 –	Grupo 1 – Modelo de Organização – INGENIAS.....	112
Figura 17 –	Grupo 1 – Modelo de Agentes – INGENIAS.....	112
Figura 18 –	Grupo 1 – Modelo de Metas e Tarefas – INGENIAS.....	113
Figura 19 –	Grupo 1 – Modelo de Interação – INGENIAS.....	113
Figura 20 –	Grupo 1 – Modelo GRASIA- INGENIAS.....	114
Figura 21 –	Grupo 1 – Modelo de Ambiente – INGENIAS.....	114
Figura 22 –	Grupo 1 – Tela Inicial – INGENIAS.....	115
Figura 23 –	Grupo 1 – Tela de Cadastro de Usuário- INGENIAS.....	115
Figura 24 –	Grupo 1 – Parte do Banco de dados - Nivelamento – INGENIAS.....	116
Figura 25 –	Grupo 1 – Parte do Banco de dados – Nível 1 – INGENIAS.....	116
Figura 26 –	Grupo 1 – Questão – Nível 1 – INGENIAS.....	117
Figura 27 –	Grupo 1 – Questão com Imagem – Nível 1 – INGENIAS.....	117
Figura 28 –	Grupo 1 – Tela de Erro de Questão – Nível 1 – INGENIAS.....	118
Figura 29 –	Grupo 1 – Tela de Feedback– Nível 1 – INGENIAS.....	118

Figura 30 – Grupo 1 –Tela de Aprovação – Nível 2 – INGENIAS.....	119
Figura 31 – Grupo 1 – Tela de Reforço – Nível 1 – INGENIAS.....	119
Figura 32 – Grupo 1 – Tela do Consultório– Nível 1 – INGENIAS.....	120
Figura 33 – Grupo 1 – Tela Final – Nível 1 - INGENIAS.....	120
Figura 34 – Grupo 2 – Modelo de Organização – INGENIAS.....	121
Figura 35 – Grupo 2 – Modelo de Agentes - INGENIAS.....	121
Figura 36 – Grupo 2 – Modelo de Metas e Tarefas - INGENIAS.....	122
Figura 37 – Grupo 2 – Modelo de Interação - INGENIAS.....	122
Figura 38 – Grupo 2 – Modelo de Ambiente - INGENIAS.....	123
Figura 39 – Grupo 2 – Tela Inicial - Configurações - INGENIAS.....	123
Figura 40 – Grupo 2 – Tela Inicial – Configurações (2) - INGENIAS.....	124
Figura 41 – Grupo 2 – Tela de Nivelamento – INGENIAS.....	124
Figura 42 – Grupo 2 – Tela Atingir Nível – INGENIAS.....	125
Figura 43 – Grupo 2 – Questão – Nível 1 - INGENIAS.....	125
Figura 44 – Grupo 2 – Ganho de Equipamento – Nível 1 - INGENIAS.....	126
Figura 45 – Grupo 2 – Tela Reposta Correta – Nível 1 - INGENIAS.....	126
Figura 46 – Grupo 2 – Tela Reposta Errada – Nível 1 - INGENIAS.....	127
Figura 47 – Grupo 2 – Tela Reforço – Nível 1 - INGENIAS.....	127
Figura 48 – Grupo 2 – Tela Aprovação – Nível 1 - INGENIAS.....	128
Figura 49 – Grupo 2 – Tela Histórico – Nível 1 – INGENIAS.....	128
Figura 50 – Grupo 3 – AgilePASSI – Diagrama de caso de uso UML - Funcionalidades.....	129
Figura 51 – Grupo 3 – AgilePASSI – Diagrama de caso de uso UML - Jogabilidade.....	129
Figura 52 – Grupo 3 – AgilePASSI – Diagrama de identificação de agentes.....	132
Figura 53 – Grupo 3 – AgilePASSI – Tela Inicial.....	134
Figura 54 – Grupo 3 – AgilePASSI – Cadastro Usuário.....	134
Figura 55 – Grupo 3 – AgilePASSI – Visão Consultório - Nivelamento.....	135
Figura 56 – Grupo 3 – AgilePASSI – Nivelamento.....	135
Figura 57 – Grupo 3 – AgilePASSI – Histórico Nivelamento.....	136
Figura 58 – Grupo 3 – AgilePASSI – Ganho de equipamento.....	136
Figura 59 – Grupo 3 – AgilePASSI – Consultório Completo.....	137

Figura 60 – Grupo 3 – AgilePASSI – Tela de Reforço.....	137
Figura 61 – Grupo 4 – AgilePASSI – Diagrama de caso de uso UML.....	138
Figura 62 – Grupo 4 – AgilePASSI – Diagrama de identificação de agentes.....	140
Figura 63 – Grupo 4 – AgilePASSI – Tela Inicial.....	143
Figura 64 – Grupo 4 – AgilePASSI – Tela Inicial (2).....	143
Figura 65 – Grupo 4 – AgilePASSI – Tela Nível 1.....	144
Figura 66 – Grupo 4 – GILEPASSI – Histórico Usuário.....	144
Figura 67 – Grupo 4 – AgilePASSI – Tela de Pergunta.....	145
Figura 68 – Grupo 4 – AgilePASSI – Tela Consultório Completo.....	145
Figura 69 – Grupo 5 – INGENIAS Scrum – Modelo de Organização.....	146
Figura 70 – Grupo 5 – INGENIAS Scrum – Modelo de Agentes.....	146
Figura 71 – Grupo 5 – INGENIAS Scrum – Modelo de Metas e Tarefas.....	147
Figura 72 – Grupo 5 – INGENIAS Scrum – Modelo de Interação.....	147
Figura 73 – Grupo 5 – INGENIAS Scrum – Modelo de Ambiente.....	148
Figura 74 – Grupo 5 – INGENIAS Scrum – Tela Inicial.....	150
Figura 75 – Grupo 5 – INGENIAS Scrum – Tela Nível.....	150
Figura 76 – Grupo 5 – INGENIAS Scrum – Tela Passagem para Nível 2.....	151
Figura 77 – Grupo 5 – INGENIAS Scrum – Tela Ganho de Equipamento.....	151
Figura 78 – Grupo 5 – INGENIAS Scrum – Tela Pergunta Tipo Áudio.....	152
Figura 79 – Grupo 5 – INGENIAS Scrum – Tela Pergunta Tipo Imagem.....	152
Figura 80 – Grupo 5 – INGENIAS Scrum – Tela Reforço.....	153
Figura 81 – Grupo 5 – INGENIAS Scrum – Tela Final.....	153
Figura 82 – Grupo 6 – INGENIAS Scrum – Modelo de Organização.....	154
Figura 83 – Grupo 6 – INGENIAS Scrum – Modelo de Agentes.....	154
Figura 84 – Grupo 6 – INGENIAS Scrum – Modelo de Metas e Tarefas.....	155
Figura 85 – Grupo 6 – INGENIAS Scrum – Modelo de Interação.....	155
Figura 86 – Grupo 6 – INGENIAS Scrum – Modelo de Ambiente.....	156
Figura 87 – Grupo 6 – INGENIAS Scrum – Tela Inicial.....	158
Figura 88 – Grupo 6 – INGENIAS Scrum – Tela Principal – Nivelamento.....	158
Figura 89 – Grupo 6 – INGENIAS Scrum – Tela Nivelamento.....	159
Figura 90 – Grupo 6 – INGENIAS Scrum – Tela Feedback.....	159
Figura 91 – Grupo 6 – INGENIAS Scrum – Tela Reforço.....	160

Figura 92 – Grupo 6 – INGENIAS Scrum – Tela Histórico do Usuário no Nível.....	160
Figura 93 – Grupo 6 – INGENIAS Scrum – Tela Ganho de Equipamento.....	161
Figura 94 – Grupo 6 – INGENIAS Scrum – Tela Histórico de Ganho de Equipamentos.....	161
Figura 95 – Grupo 6 – INGENIAS Scrum – Tela Final.....	162
Figura 96 – Grupo 6 – INGENIAS Scrum – Tela Consultório Completo.....	162

LISTA DE QUADROS

Quadro 1 –	Princípios do processo ágil.....	25
Quadro 2 –	Resultado das fases de análise e design do processo de desenvolvimento com INGENIAS.....	41
Quadro 3 –	Comparação entre métodos INGENIAS, AGILEPASSI e INGENIAS Scrum.....	48
Quadro 4 –	Cronograma do estudo de caso.....	56
Quadro 5 –	Comparação dos envolvidos no estudo de caso, por método.....	60
Quadro 6 –	Artefatos requeridos para o grupo 1 – INGENIAS.....	61
Quadro 7 –	Artefatos requeridos para o grupo 2 – INGENIAS.....	61
Quadro 8 –	Artefatos requeridos para o grupo 3 – AgilePASSI.....	62
Quadro 9 –	Artefatos requeridos para o grupo 4 – AgilePASSI.....	62
Quadro 10 –	Artefatos requeridos para o grupo 5 – INGENIAS Scrum.....	63
Quadro 11 –	Artefatos requeridos para o grupo 6 – INGENIAS Scrum.....	63
Quadro 12 –	Total de dias por semana em atraso nas entregas dos artefatos.....	64
Quadro 13 –	Pontuação dos requisitos.....	65
Quadro 14 –	Notas do protótipo final do jogo implementado.....	69
Quadro 15 –	Resumo dos grupos para cálculo da ANOVA.....	71
Quadro 16 –	Cálculo da ANOVA em relação aos dias de atraso.....	71
Quadro 17 –	Cálculo da ANOVA em relação à análise dos requisitos.....	72
Quadro 18 –	Resumo dos grupos para cálculo da ANOVA.....	72
Quadro 19 –	Avaliação da satisfação das métricas.....	73
Quadro 20 –	Avaliação da satisfação das dimensões.....	74
Quadro 21 –	Comparativo dos métodos do estudo de caso.....	77
Quadro 22 –	Nível x Quantidade de questões x Grau de conhecimento.....	90
Quadro 23 –	Exemplo de funcionamento do nível 1 do jogo.....	93
Quadro 24 –	Exemplo de funcionamento do nível de reforço 1 do jogo.....	94
Quadro 25 –	Análise de variância.....	96
Quadro 26 –	Questionário de avaliação da qualidade.....	105
Quadro 27 –	Descrição das dimensões do questionário de avaliação da qualidade.....	109

Quadro 28 –	Descrição das métricas do questionário de avaliação da qualidade.....	111
Quadro 29 –	Grupo 3 – AgilePASSI – Detalhamento dos casos de uso (Funcionalidades).....	130
Quadro 30 –	Grupo 3 – AgilePASSI – Detalhamento dos casos de uso (Jogabilidade).....	131
Quadro 31 –	Grupo 3 – AgilePASSI – Detalhamento dos requisitos funcionais.....	133
Quadro 32 –	Grupo 3 – AgilePASSI – Detalhamento dos requisitos não funcionais...	133
Quadro 33 –	Grupo 4 – AgilePASSI – Detalhamento dos casos de uso.....	139
Quadro 34 –	Grupo 4 – AgilePASSI – Detalhamento dos requisitos funcionais.....	140
Quadro 35 –	Grupo 6 – AgilePASSI – Detalhamento dos requisitos não funcionais...	142
Quadro 36 –	Grupo 5 – INGENIAS Scrum – 1ª versão do <i>backlog</i>	148
Quadro 37 –	Grupo 5 – INGENIAS Scrum – 2ª versão do <i>backlog</i>	149
Quadro 38 –	Grupo 5 – INGENIAS Scrum – 3ª versão do <i>backlog</i>	149
Quadro 39 –	Grupo 5 – INGENIAS Scrum – 4ª versão do <i>backlog</i>	149
Quadro 40 –	Grupo 6 – INGENIAS Scrum – 1ª versão do <i>backlog</i>	156
Quadro 41 –	Grupo 6 – INGENIAS Scrum – 2ª versão do <i>backlog</i>	157
Quadro 42 –	Grupo 6 – INGENIAS Scrum – 3ª versão do <i>backlog</i>	157
Quadro 43 –	Grupo 6 – INGENIAS Scrum – 4ª versão do <i>backlog</i>	157

LISTA DE GRÁFICOS

Gráfico 1 –	Quantidade absoluta dos participantes em relação à faixa etária, por grupo.....	57
Gráfico 2 –	Quantidade absoluta dos participantes em relação a atuação no mercado, por grupo.....	58
Gráfico 3 –	Quantidade absoluta dos participantes em relação aos conhecimentos sobre agentes, por grupo.....	58
Gráfico 4 –	Quantidade absoluta dos participantes em relação aos conhecimentos sobre sistemas multiagentes, por grupo.....	59
Gráfico 5 –	Quantidade absoluta dos participantes em relação às expectativas, por grupo.....	80
Gráfico 6 –	Quantidade absoluta dos participantes em relação ao aprendizado do método, por grupo.....	81
Gráfico 7 –	Quantidade absoluta dos participantes em relação ao método proposto, por grupo.....	82
Gráfico 8 –	Quantidade dos participantes em relação a probabilidade do participante voltar a trabalhar com sistemas multiagentes, por grupo.....	82

LISTA DE TABELAS

Tabela 1 –	Resultado NÃO se rejeita a hipótese – Grupos 1 e 3.....	97
Tabela 2 –	Resultado NÃO se rejeita a hipótese – Grupos 1 e 4.....	97
Tabela 3 –	Resultado rejeita-se a hipótese – Grupos 1 e 5.....	98
Tabela 4 –	Resultado NÃO se rejeita a hipótese – Grupos 1 e 6.....	98
Tabela 5 –	Resultado rejeita-se a hipótese – Grupos 2 e 3.....	99
Tabela 6 –	Resultado rejeita-se a hipótese – Grupos 2 e 4.....	99
Tabela 7 –	Resultado rejeita-se a hipótese – Grupos 2 e 5.....	100
Tabela 8 –	Resultado rejeita-se a hipótese – Grupos 5 e 6.....	100
Tabela 9 –	Resultado NÃO se rejeita a hipótese – Grupos 1 e 3.....	101
Tabela 10 –	Resultado NÃO se rejeita a hipótese – Grupos 1 e 4.....	101
Tabela 11 –	Resultado NÃO se rejeita a hipótese – Grupos 1 e 5.....	102
Tabela 12 –	Resultado NÃO se rejeita a hipótese – Grupos 1 e 6.....	102
Tabela 13 –	Resultado NÃO se rejeita a hipótese – Grupos 2 e 3.....	103
Tabela 14 –	Resultado NÃO se rejeita a hipótese – Grupos 2 e 4.....	103
Tabela 15 –	Resultado NÃO se rejeita a hipótese – Grupos 2 e 5.....	104
Tabela 16 –	Resultado NÃO se rejeita a hipótese – Grupos 2 e 6.....	104

LISTA DE ABREVIATURAS E SIGLAS

ANOVA	Análise de variância
AOSEM	<i>Agent-Oriented Software Engineering Methodologies</i>
API	<i>Application Programming Interface</i>
CASE	<i>Computer-Aided Software Engineering</i>
CPU	Unidade central de Processamento
EPS	<i>Evolutionary Spiral Process</i>
FUP	<i>Follow-up</i>
IAF	INGENIAS <i>Agent Framework</i>
IDK	kit de desenvolvimento INGENIAS
IID	Independentes e identicamente distribuídos
JADE	<i>Java Agent Development Framework</i>
MaSE	Multiagent System Engineering
MDD	<i>Model-driven-development</i>
PASSI	<i>Process for Agent Societies Specification and Implementation</i>
QMRes	Quadrado médio dos resíduos
QMTrat	Quadrado médio dos tratamentos
RAD	<i>Rapid Application Development</i>
RUP	<i>Rational Unified Process</i>
SMA	Sistema multiagentes
SQRes	Soma dos quadrados dos resíduos
SQTotal	Soma dos quadrado total
SQTrat	Soma dos quadrados dos tratamentos
UDP	<i>Unified Development Process</i>
UML	<i>Unified Modeling Language</i>
XP	<i>Extremming Programming</i>

SUMÁRIO

	INTRODUÇÃO.....	18
1	DESENVOLVIMENTO DE SISTEMAS	23
1.1	Processos para o desenvolvimento de <i>software</i>.....	23
1.2	Modelos ágeis no desenvolvimento de <i>software</i>.....	24
1.3	XP - <i>Extreme Programming</i>.....	26
1.4	<i>Scrum</i>.....	27
2	ORIENTAÇÃO A AGENTES.....	29
2.1	Agentes de <i>software</i>.....	29
2.2	Sistemas multiagentes.....	31
2.3	Métodos para sistemas multiagentes.....	31
2.4	Método INGENIAS.....	32
2.4.1	<u>Metamodelos INGENIAS.....</u>	33
2.4.2	<u>Modelo de organização.....</u>	34
2.4.3	<u>Modelo de agentes.....</u>	36
2.4.4	<u>Modelo de metas e tarefas.....</u>	37
2.4.5	<u>Modelo de interação.....</u>	38
2.4.6	<u>Modelo de ambiente.....</u>	39
2.4.7	<u>Processo de desenvolvimento do INGENIAS.....</u>	40
2.5	Método INGENIAS Scrum.....	41
2.5.1	<u>A aplicação do método Scrum no método INGENIAS.....</u>	42
2.5.2	<u>O ciclo de vida.....</u>	43
2.6	Método AgilePASSI.....	45
2.7	Comparação dos métodos INGENIAS, AgilePASSI E INGENIAS Scrum.....	48
3	CONCEITOS SOBRE ESTUDO DE CASO.....	50
3.1	Principais definições.....	50
3.2	Processo para realização de um estudo de caso.....	51
4	ESTUDO DE CASO: DESENVOLVIMENTO DO JOGO MEDEDUC COM INGENIAS, INGENIAS SCRUM E AGILEPASSI.....	53
4.1	Visão geral do estudo de caso.....	53
4.2	Objetivo e hipótese.....	54

4.3	Contextualização do estudo de caso: MEDEDUC.....	54
4.4	Cronograma do Estudo de Caso.....	56
4.5	Preparação dos participantes.....	56
4.6	Coleta de dados do estudo de caso.....	60
4.7	Análise e interpretação dos dados estatísticos.....	70
4.8	Análise qualitativa dos resultados.....	75
4.9	Comparação das propostas de desenvolvimento do estudo de caso.....	77
4.10	Ameaças à validade.....	79
4.11	Pesquisa de satisfação do estudo de caso.....	79
	CONCLUSÕES E FUTUROS TRABALHOS.....	84
	REFERÊNCIAS.....	86
	APÊNDICE A – Descrição detalhada do MEDEDUC.....	90
	APÊNDICE B – Análise de Variância.....	95
	APÊNDICE C – Testes estatísticos ANOVA (atraso na entrega).....	97
	APÊNDICE D – Testes estatísticos ANOVA (requisitos).....	101
	APÊNDICE E – Questionário de avaliação da qualidade.....	105
	APÊNDICE F – Descrição das dimensões e métricas da avaliação da qualidade.....	108
	ANEXO A – Expectativas dos envolvidos no estudo de caso.....	111
	ANEXO B – Artefatos desenvolvidos pelo grupo 1 – INGENIAS.....	112
	ANEXO C – Artefatos desenvolvidos pelo grupo 2 – INGENIAS.....	121
	ANEXO D – Artefatos desenvolvidos pelo grupo 3 – AgilePASSI.....	129
	ANEXO E – Artefatos desenvolvidos pelo grupo 4 – AgilePASSI.....	138
	ANEXO F – Artefatos desenvolvidos pelo grupo 5 – INGENIAS Scrum.....	146
	ANEXO G – Artefatos desenvolvidos pelo grupo 6 – INGENIAS Scrum.....	154
	ANEXO H – Comentários sobre a experiência dos envolvidos no estudo de caso.....	163

INTRODUÇÃO

Nos dias atuais, os sistemas multiagentes (SMA) vêm sendo cada vez mais reconhecidos e explorados em diferentes contextos (STURM; SHEHORY, 2014). Esses sistemas são constituídos por agentes autônomos e inteligentes, que visam atingir um objetivo comum. Algumas características dos sistemas multiagentes podem ser vistas como vantagens sobre os sistemas comuns, como por exemplo: rapidez para solucionar problemas, maior flexibilidade, mais segurança e necessidade de menor comunicação (MOULIN; CHAIB-DRAA, 1996).

Nesta perspectiva, métodos orientados a agentes foram propostos (STURM; SHEHORY, 2014), (HENDERSON-SELLERS; GIORGINI, 2005). É importante que, diante de tantos métodos distintos, escolha-se a abordagem mais adequada. A prioridade dos serviços computacionais é satisfazer às necessidades dos envolvidos em sua concepção e uso e, por isso, deve-se conhecer os métodos existentes, considerando suas características, vantagens e desvantagens, para que as expectativas dos usuários e da equipe desenvolvedora possam ser contempladas.

Por outro lado, mais recentemente, os métodos ágeis de desenvolvimento de *software* vêm sendo utilizados com mais frequência. O objetivo de se utilizar estes métodos é a busca de estratégias de construção de sistemas de alta qualidade, de maneira rápida e que supram as necessidades dos usuários do sistema. Esses métodos ágeis possuem um comportamento regular e prático para o desenvolvimento de *software*, onde são elaborados planos de ação, que, por sua vez, consideram implicitamente as fases futuras, de modo a adaptá-las a mudanças, de acordo com a evolução do projeto.

No contexto dos métodos ágeis voltados para o desenvolvimento orientado a agentes, existem duas propostas mais difundidas: a AgilePASSI (CHELLA et al., 2004) e a INGENIAS Scrum (GONZÁLEZ-MORENO et al., 2014). Cada uma delas possui suas particularidades em relação aos domínios a serem trabalhados. Entretanto, a literatura da área carece de trabalhos comparativos, que ofereçam indicativos para apoiar os desenvolvedores na escolha do método ideal para cada projeto. Desta forma, torna-se importante compará-las entre si e em relação a abordagens tradicionais de desenvolvimento orientado a agentes.

Neste caso, no presente trabalho, são considerados os métodos AgilePASSI e INGENIAS Scrum, métodos de desenvolvimento ágil para sistemas multiagentes e o método

INGENIAS que apresenta uma proposta orientada a agentes com uso do Processo Unificado (*Unified Development Process*) (COSSENTINO et al., 2014).

O método INGENIAS (PAVÓN et al., 2005) foi criado a partir da necessidade de evoluir, ou mudar conceitos de tecnologias de agentes. Ele possui algumas características diferenciadas dos demais métodos tradicionais orientados a agentes, como por exemplo: possuir ferramentas *Computer-Aided Software Engineering* (CASE) abertas; dispor de uma notação simples e de fácil entendimento (os meta-modelos); possuir um processo de desenvolvimento detalhado, que possibilitam reduzir problemas de implementação.

O método AgilePASSI foi baseado no método PASSI (COSSENTINO; POTTS, 2002; CHELLA et al., 2004), que gerava SMAs com boa qualidade, mas era demorado e pouco flexível. O método AgilePASSI é uma versão ágil do PASSI e para sua criação foram aproveitados estudos que abordam os métodos orientados a agentes, que por sua vez, permitem a composição de uma nova tecnologia reutilizando fragmentos existentes.

O método INGENIAS Scrum (GONZÁLEZ-MORENO et al., 2014) aproveitou a abordagem de meta-modelos do método INGENIAS e agregou os conceitos do método ágil de gerenciamento de projetos Scrum. Baseia-se em equipes auto-organizadas e habilitadas para entregar os incrementos do produto e em clientes ou proprietários do produto, que devem fornecer à equipe de desenvolvimento uma lista dos recursos desejados, usando o valor comercial como o mecanismo de prioridade.

Em 2016, Ferreira avaliou alternativas de desenvolvimento aplicando três métodos baseados nos conceitos da Engenharia de *Software* (O-MaSE, AgilePASSI e Scrum) em um estudo de caso, onde foi modelado e implementado um jogo médico educacional (FERREIRA, 2016). Três métodos com distintas características foram escolhidos, entre eles, INGENIAS que é específico para sistemas multiagentes com um processo de desenvolvimento tradicional, INGENIAS Scrum que aplica o desenvolvimento ágil num método orientado a agentes e o AgilePASSI que é específico para desenvolvimento ágil. Embora o AgilePASSI seja uma proposta intermediária, os resultados do estudo de caso demonstraram que o uso do Scrum possibilitou uma melhor versão de protótipo, mas não considerou o uso de agentes. Nesse estudo, a melhor solução em relação à criação dos agentes acabou sendo as obtidas pelos grupos que usaram o método orientado a agentes sem desenvolvimento ágil.

Como as propostas de métodos adotados em Ferreira (2016) eram muito diferentes entre si, foi proposto um novo estudo de caso a fim de agregar evidências ao estudo de caso já realizado, para que se pudesse comprovar a hipótese de que abordagens de desenvolvimento ágil “são responsáveis por melhor definir e implementar os requisitos e têm entrega mais rápida

quando comparadas com produtos desenvolvidos usando uma abordagem de desenvolvimento tradicional” (FERREIRA, 2016, p. 66).

A finalidade deste trabalho é aplicar estes três métodos orientados a agentes em um estudo de caso direcionado à área médica – jogo médico educacional – e definir as principais vantagens e desvantagens de cada método, discutindo quais foram mais adequadas ao estudo de caso.

Objetivo geral

Este trabalho apresenta como objetivo geral a comparação, por meio de um estudo de caso, de métodos de desenvolvimento de sistemas multiagentes na elaboração de um jogo médico educacional voltado para estudantes de medicina.

Objetivos específicos

Os objetivos específicos desta dissertação são:

- Aplicar três diferentes métodos orientados a agentes, com distintas características para desenvolver um jogo médico educacional orientado a agentes (INGENIAS, método tradicional orientado a agentes, AgilePASSI e INGENIAS Scrum, métodos orientados a agentes ágeis);
- Comparar os resultados alcançados em um experimento realizado por grupos de estudantes de uma turma de Inteligência Artificial do segundo semestre de 2017, de uma universidade pública;
- Comparar os métodos e apontar as vantagens e desvantagens no uso de cada um deles;
- Comparar os métodos em relação aos resultados obtidos, utilizando técnicas de análise estatística e avaliação da qualidade das aplicações;

- Identificar um conjunto de diretrizes, para alcançar melhores resultados no desenvolvimento de sistemas orientados a agentes.

Contribuições esperadas

Esta dissertação visa fornecer informações necessárias para que outras pessoas que possuem interesse na área de construção de sistemas multiagentes tenham como referência uma coleção de métodos considerando suas vantagens e desvantagens.

Relevância

A contribuição deste trabalho, gerada por meio dos resultados de um estudo de caso, será a de fornecer estratégias de utilização de metodologias para sistemas multiagentes e identificar as vantagens e desvantagens associadas ao uso das metodologias em questão. A relevância do estudo é fundamentada na carência de trabalhos que integrem, de uma forma prática, assuntos como sistemas multiagentes, jogos educativos e metodologias tradicionais e ágeis.

Organização

Para atingir seu objetivo, este trabalho está organizado em quatro capítulos, seguidos pela conclusão, referências e anexos.

O capítulo 1 aborda de uma forma geral os conceitos teóricos de modelos ágeis de desenvolvimento de sistemas como *Extreme Programming* (XP) e Scrum. Além disso, apresenta os resultados da comparação entre modelos ágeis e tradicionais para desenvolvimento de sistemas multiagentes.

O capítulo 2 apresenta os conceitos de orientação a agentes, tais como: agentes de *software*, sistemas multiagentes, métodos para desenvolvimento de sistemas multiagentes e os

métodos utilizados neste estudo de caso, INGENIAS, INGENIAS Scrum e AgilePASSI. Além de comparar estes três métodos.

No capítulo 3, os conceitos relacionados a estudo de casos, como realizar o processo de um estudo de caso, conceitos sobre jogos médicos educacionais e métodos para o desenvolvimento de jogos médicos educacionais são apresentados.

O capítulo 4 descreve o desenvolvimento do jogo médico educacional que foi utilizado no estudo de caso – MEDEDUC – voltado para estudantes de medicina, utilizando os métodos INGENIAS, INGENIAS Scrum e AgilePASSI. Aborda, também, o objetivo e hipótese, cronograma, análise estatística e pesquisa de satisfação, entre outros.

Por fim, são apresentadas as considerações finais e futuros trabalhos.

1 DESENVOLVIMENTO DE SISTEMAS

Este capítulo tem por objetivo descrever de forma conceitual a definição de processos de desenvolvimentos de sistemas, retratando os modelos de processo ágeis, como o XP e Scrum.

1.1 Processos para o desenvolvimento de *software*

A engenharia de *software* é uma abordagem sistemática e disciplinada para o desenvolvimento de *software*. No desenvolvimento de um projeto de *software*, quanto mais complexo é o *software*, maior é o empenho que o engenheiro de *software* deve fazer para desenvolver e tem que ter maior gerenciamento (JALOTE, 2005). A engenharia de *software* é uma tecnologia em camadas e elas são distribuídas em: foco na qualidade, processo de desenvolvimento, métodos e ferramentas. A camada mais importante da engenharia de *software* é a de processo. O processo de engenharia de *software* é o elo que mantém unidas as camadas de tecnologias e permite o desenvolvimento racional e oportuno de *softwares* de computador.

Apesar de existirem vários processos para o desenvolvimento de *software*, existem atividades fundamentais comuns a todos eles, que são: especificação de *software*, projeto e implementação de *software*, validação de *software* e evolução de *software*.

Em 1970, o modelo clássico ou cascata, foi proposto por Royce. Este modelo também é conhecido por abordagem “*top-down*”. Este modelo foi o único com aceitação por todos até meados da década de 1980. Ele foi originário de modelos de atividade de engenharia com o objetivo de estabelecer ordem no desenvolvimento de grandes produtos de *software*. As fases deste modelo são: especificação, projeto, implementação, validação e manutenção.

O modelo espiral foi proposto o modelo *Evolutionary Spiral Process* (EPS) (BOEHM, 1988). Este modelo baseia-se em quatro principais atividades:

- i. determinação dos objetivos, das alternativas e restrições;
- ii. análise de risco e prototipação;
- iii. validação e verificação;
- iv. planejamento da fase seguinte.

O modelo EPS possui algumas desvantagens, como: a exigência de que se tenha gerentes e técnicos experientes e as tarefas gerenciais para acompanhamento e controle do projeto são mais difíceis.

O *Rational Unified Process* (RUP) surgiu no começo dos anos 80, pela *Rational Software Corporation*. Fundada por Paul Levy e Mike Devlin, a *Rational* se dedicou a desenvolver com sucesso sistemas largos e complexos (GIBBS, 2006). Este modelo foi criado com a finalidade de apoiar o desenvolvimento orientado a objetos, fornecendo uma forma sistemática para se obter vantagens no uso da *Unified Modeling Language* (UML). Possui como principal particularidade atender às necessidades dos usuários a fim de garantir uma produção de *software* de alta qualidade que cumpra um cronograma e um orçamento viável.

Entende-se que o RUP possui uma organização preestabelecida e atua como auxiliador para o desenvolvimento de *softwares*. Ele conta com regras importantes para redução de riscos e a ideia mais utilizada do RUP é o de conceitos de melhores práticas, isto é, ele define melhores práticas para tornar o desenvolvimento do *software* mais eficiente e com menos riscos. Algumas destas melhores práticas podem ser definidas por: investigação constante da qualidade, verificação das mudanças que ocorrem, gerenciamento dos requisitos, desenvolvimento de forma iterativa, adequação visual e uso de arquiteturas baseadas em componentes.

O modelo iterativo de desenvolvimento RUP identifica os ciclos do processo em quatro fases: início (*inception*) onde se determina o escopo do desenvolvimento do *software*, elaboração (*elaboration*) onde se planejam as atividades e os recursos que serão utilizados, construção (*construction*) em que se implementa o *software* e a transição (*transition*) onde o produto é entregue aos usuários.

1.2 Modelos ágeis no desenvolvimento de *software*

Bassi Filho (2008, p. 22) define o modo pelo qual os processos de desenvolvimento ágeis surgiram:

Durante a evolução dos processos de Engenharia de *Software*, a indústria se baseou nos métodos tradicionais de desenvolvimento de *software*, que definiram por muitos anos os padrões para criação de *software* nos meios acadêmico e empresarial. Porém, percebendo que a indústria apresentava um grande número de casos de fracasso, alguns líderes experientes adotaram modos de trabalho que se opunham aos principais conceitos dos métodos tradicionais. Aos poucos, foram percebendo que suas formas

de trabalho, apesar de não seguirem os padrões no mercado, eram bastante eficientes. Aplicando-as em vários projetos, elas foram aprimoradas e, em alguns casos, chegaram a se transformar em novos métodos de desenvolvimento de *software*. Esses métodos passaram a ser chamadas de leves por não utilizarem as formalidades que caracterizavam os processos tradicionais e por evitarem a burocracia imposta pela utilização excessiva de documentos. Com o tempo, algumas delas ganharam destaque nos ambientes empresarial e acadêmico, gerando grandes debates, principalmente relacionados à confiabilidade dos processos e à qualidade do *software*.

Os modelos de processos ágeis surgiram devido a insatisfação e a sobrecarga nos modelos de desenvolvimento de *software* tradicionais nas décadas de 1980 e 1990. As principais características dos modelos ágeis são: ter como foco principal o código, e não somente o projeto; são planejados para entregar o *software* em funcionamento e evoluí-lo de forma rápida e são baseados em abordagem interativa.

A ideia do manifesto ágil (BECK et al., 2001) é descobrir melhores maneiras de desenvolver *softwares*, valorizando mais os indivíduos e interações, *softwares* já existentes, adaptação a mudanças e colaboração do cliente.

Tomando como base estas definições, o manifesto ágil (BECK et al., 2001) descreveu doze fatores importantes dos processos ágeis, sendo os fatores mais importantes descritos no quadro 1: envolvimento do cliente, entrega incremental, pessoas (não processos), aceitar as mudanças e manter a simplicidade.

Quadro 1 – Princípios do processo ágil.

Princípios	Descrição
Envolvimento do cliente	Os clientes devem estar intimamente envolvidos no processo de desenvolvimento. Seu papel é fornecer e priorizar novos requisitos do sistema e avaliar suas iterações.
Entrega incremental	O software é desenvolvido em incrementos com o cliente, especificando os requisitos para serem incluídos em cada um.
Pessoas, não processos	As habilidades da equipe de desenvolvimento devem ser reconhecidas e exploradas. Membros da equipe devem desenvolver suas próprias maneiras de trabalhar, sem processos prescritivos.
Aceitar as mudanças	Deve-se ter em mente que os requisitos do sistema vão mudar. Por isso, projete o sistema de maneira a acomodar essas mudanças.
Manter a simplicidade	Focalize a simplicidade, tanto do software a ser desenvolvido quanto do processo de desenvolvimento. Sempre que possível, trabalhe ativamente para eliminar a complexidade do sistema.

Fonte: SOMMERVILLE, 2011.

As aplicabilidades dos processos ágeis são muitas. Entre elas destacam-se: desenvolvimento de um produto, desenvolvimento de sistema personalizado e correções de problemas de escalabilidade em sistemas grandes.

Alguns problemas enfrentados por métodos ágeis são: dificuldade em manter o interesse dos clientes envolvidos no processo, dificuldade em priorizar mudanças e trabalho extra ao tentar manter a simplicidade. Por estas razões, o sucesso da utilização dos métodos ágeis depende do engajamento da equipe e da cooperação em relação à adaptação a mudanças durante o processo de desenvolvimento.

Algumas propostas de desenvolvimento ágil surgiram, sendo que XP e Scrum são as mais conhecidas.

1.3 XP - Extreme Programming

Define-se o XP como um modelo de processo ágil e é utilizado para o desenvolvimento de projetos. Possui o objetivo de tornar o projeto em desenvolvimento mais flexível e possibilitar que o cliente receba o máximo de valor de cada dia de trabalho da equipe de desenvolvimento e, por isso, possui valores para assegurar isto. São eles: *feedback*, comunicação, simplicidade e coragem.

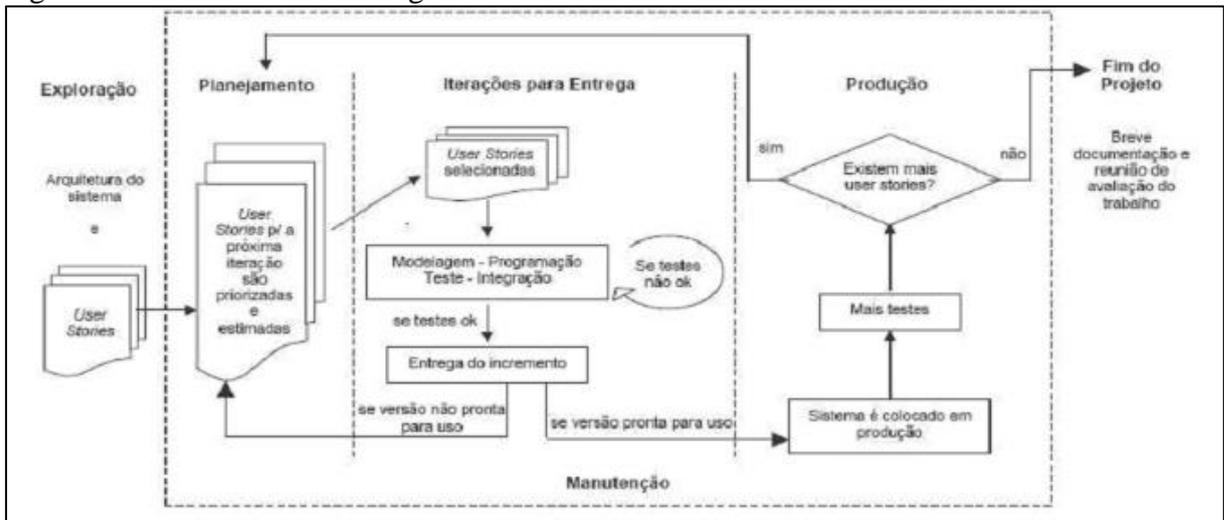
O modelo de XP (BECK, 1999) é um modelo de processo ágil para o desenvolvimento de projetos de tecnologia da informação cujas especificações são passíveis de alterações. As abordagens ágeis são parecidas ao RUP, porém possuem algumas mudanças que são importantes. O XP não é caracterizado em uma documentação no âmbito tradicional e sofre influência pela *Rapid Application Development (RAD)*. Pode-se citar como uma diferença considerável do XP, a sua ênfase nos testes desde o início do ciclo de desenvolvimento.

A XP é dividida em seis fases, estabelecidas na figura 1:

- i. Exploração: o cliente descreve uma funcionalidade desejada para o primeiro *release*.
- ii. Planejamento: definem-se as prioridades junto com o cliente.
- iii. Iterações para a fase de *release*: acontecem muitas iterações até o primeiro *release* ser completado.
- iv. Validação para produção: os testes e verificações são feitas para validação do *software*.

- v. Manutenção: após o primeiro *release* para produção, há novas edições do sistema com novas funcionalidades.
- vi. Fim do projeto: o cliente está satisfeito com o código.

Figura 1 – Fases do Processo Ágil XP.



Fonte: BECK, 1999.

1.4 Scrum

Segundo Schwaber e Beedle (2001), o Scrum é um modelo de gerenciamento de projetos ágeis que é usado para entregar iterativamente para o cliente, incrementos de *software* de alto valor. É um projeto simples usado para organizar equipes e obter trabalho de forma mais produtiva com maior qualidade. O Scrum depende da auto-organização, motivando equipes a entregar os incrementos do produto. Também depende de um cliente, do proprietário do produto (*product owner*), que fornece à equipe de desenvolvimento uma lista de recursos desejados usando o valor do negócio como o mecanismo de priorização.

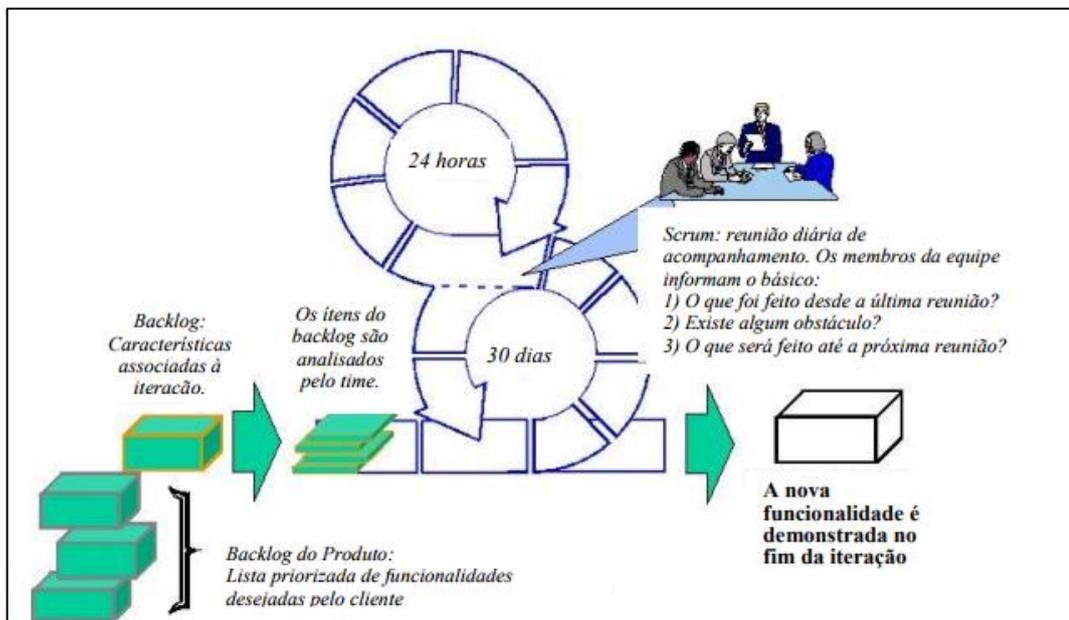
Este modelo de processo ágil baseia-se na teoria de controle de processos e, por esta razão, visa minimizar e controlar os riscos na execução de determinado projeto. Além disso, há transparência em todas as etapas de desenvolvimento de um projeto diminuindo as chances de problemas futuros.

É importante ressaltar que, no modelo Scrum, toda a parte de desenvolvimento do projeto ou produto é realizada em iterações. O esforço é orientado de uma maneira que seja

exibido um conjunto novo de aplicabilidades no fim de cada iteração. Sabe-se, também, que cada iteração tem um período estabelecido.

O ciclo de vida do processo Scrum é definido de acordo com a figura 2.

Figura 2 – Ciclo de vida de um projeto em Scrum.



Fonte: AGILLE, 2003.

Por meio do *backlog*, é possível que os clientes e os usuários estabeleçam as suas necessidades. Dentro do *backlog* são listados os itens de necessidade e ordenados por prioridade e, em seguida, são apresentados à equipe de desenvolvimento, que faz a avaliação para determinar o tempo necessário para atender a cada item da lista.

Após obter o resultado, define-se o conteúdo da iteração, e estes são chamados de *sprint*. Durante a iteração, a equipe desenvolve os itens do *backlog* pertencentes à iteração. O status do projeto é informado em reuniões rápidas, que podem ser realizadas diariamente/semanalmente, ou de acordo com a necessidade do time.

Ao final do *sprint*, a equipe submete uma nova versão do projeto com o novo conjunto de funcionalidades implementado. O resultado da iteração é avaliado junto ao cliente, e o conteúdo do próximo *sprint* é definido.

2 ORIENTAÇÃO A AGENTES

Este capítulo aborda de forma conceitual a definição de agentes de *software* e de SMAs. Os métodos orientados a agentes são identificados enfatizando suas características e são descritos em detalhes: o método orientado a agentes INGENIAS e os métodos de desenvolvimento ágil orientado a agentes INGENIAS Scrum e AgilePASSI.

2.1 Agentes de *software*

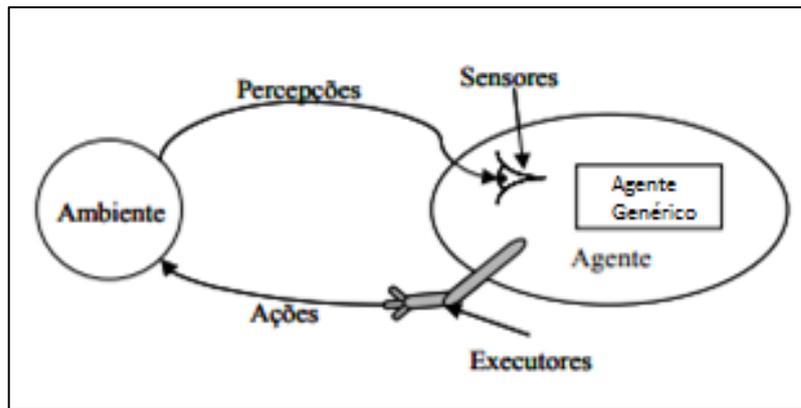
A abordagem de agentes de *software* é algo que ainda não possui uma definição única. Diversas definições podem ser encontradas na literatura, cada uma tentando adequar-se aos propósitos daqueles que a define. Pode-se citar alguns conceitos de agentes de *software*:

“Um agente é alguma coisa que pode ser vista como percebendo seu ambiente através de sensores e que age neste ambiente através de seus atuadores.” (RUSSEL; NORVIG, 1995, apud TURBAN et al., 2000, p. 497) (Figura 3).

“Agentes são sistemas computacionais situados em um ambiente, sendo capazes de realizar ações autônomas e flexíveis no intuito de alcançar seus objetivos.” (JENNINGS, 1998, p. 7-38).

Agentes compartilham características comuns como: a capacidade de agir no ambiente e comunicar-se com outros agentes, autonomia - ou seja, suas ações não sofrem interferência constante do usuário; possui uma representação parcial interna do ambiente em que está inserido e ser capaz de percebê-lo (FERBER; FITZGERALD, 1999, p. 9).

Figura 2 – As interações de um agente genérico com seu ambiente



Fonte: RUSSEL; NORVIG, 1995.

É importante ressaltar que um agente não precisa ser necessariamente um *software*. Um exemplo é o ser humano, que percebe o ambiente através de sensores como audição, olfato e paladar e interage com o mesmo através de atuadores, como: mãos, pernas, boca, nariz etc. Liang e Huang (2000) definem agente como sendo uma pessoa ou um negócio autorizado a agir no lugar de outra. Em desenvolvimento de *software*, um agente é um programa de computador que pode operar autonomamente e efetuar tarefas singulares sem a direta supervisão humana (HOFFMAN; NOVAK, 1996).

Apesar de não haver uma definição única sobre agentes, há uma concordância de que a autonomia é a característica principal dos agentes. Eles são capazes de atuar sem intervenção humana ou de outros sistemas através do controle que eles possuem do seu estado interno e do seu comportamento.

Para Wooldridge (1999) um agente inteligente é aquele que possui comportamento autônomo e flexível para alcançar seus objetivos, e flexibilidade significa três coisas:

- Reatividade – os agentes inteligentes percebem seu ambiente e respondem em tempo às mudanças que nele ocorrem para alcançar seus objetivos.
- Proatividade - os agentes inteligentes são capazes de exibir comportamento orientado a objetivos tomando a iniciativa de forma a alcançar seus objetivos.
- Habilidade social - os agentes inteligentes são capazes de interagir com outros agentes e, possivelmente, com humanos para alcançar seus objetivos.

O crescimento da importância de agentes de *software* faz com que a necessidade do uso de técnicas de decisão e negociação aumente. Os agentes, como já foi abordado, são padronizações do conhecimento humano, codificadas em linguagem de computadores, ou seja,

sem o auxílio de bons mecanismos de decisão, os agentes não são bons tomadores de decisão e, conseqüentemente, delegar-lhes tarefas não é uma boa prática.

2.2 Sistemas multiagentes

SMA são definidos como “componentes de *software* que se comunicam com seus colegas através de uma expressiva linguagem de comunicação de agentes” (GENESERETH, 1994, p. 48-53).

Um SMA pode ser definido como um grupo de agentes autônomos, interagindo entre si e compartilhando um mesmo ambiente, que é percebido através de sensores, onde eles agem realizando ações, e também podem fornecer alto grau de escalabilidade, através da inclusão de novos agentes quando necessário, e ainda fazer com que agentes assumam as atividades de outros agentes em casos de falha (BUSONI et al., 2008).

As razões para o aumento do interesse em pesquisas com SMA são: a capacidade de fornecer robustez e eficiência, de permitir interoperabilidade entre sistemas legados e de resolver problemas cujo dado, especialidade ou controle é distribuído.

O conceito de sociedade de agentes é também definido. Assim como os seres humanos, os agentes podem ser organizados em sociedades. Cada um dos agentes pode ter seu papel nessas comunidades. Respeitando-se as regras ou “leis” dessa sociedade, pode-se acrescentar, retirar ou substituir agentes.

2.3 Métodos para sistemas multiagentes

As técnicas de engenharia de *software* convencionais não se encaixam apropriadamente para projetar sistemas baseados em agentes devido às suas peculiaridades.

No contexto de engenharia de *software*, desenvolver *softwares* de qualidade de forma produtiva, é viabilizada por um processo sistemático com uso de métodos e ferramentas. Os métodos fornecem técnicas para serem utilizadas em todas as fases de desenvolvimento. As ferramentas apoiam a automatização dos métodos (PRESSMAN, 2011).

Na literatura são encontrados vários métodos orientados a agentes (HENDERSON-SELLERS; GIOGINI, 2005; STURM; SHEHOBY, 2014), que surgiram da engenharia de *software* e outros da inteligência artificial/engenharia do conhecimento.

Neste trabalho, são descritos alguns métodos orientados a agentes. São eles: INGENIAS (PAVÓN et al., 2005) e os métodos ágeis orientados a agentes: AgilePASSI (CHELLA et al., 2004) e INGENIAS Scrum (GÓMEZ-SANZ, 2008). O método tradicional para o desenvolvimento de sistemas multiagentes INGENIAS foi escolhido por possuir poucos trabalhos publicados em relação a ele e pela afinidade do pesquisador com o método. Já os métodos INGENIAS Scrum e AgilePASSI foram escolhidos pois são duas propostas de uso de agentes com desenvolvimento ágil com características distintas. No estudo de Ferreira (2016), o uso de um método ágil possibilitou uma melhor versão de protótipo, mas sem boa utilização de agentes. O método INGENIAS tem como base a engenharia de *software* e a PASSI a inteligência artificial/engenharia do conhecimento.

2.4 Método INGENIAS

O método INGENIAS (PAVÓN et al., 2005) é um método de engenharia de *software* orientado a agentes para o desenvolvimento de sistemas multiagentes. Combina resultados de pesquisa de agentes com conceitos e métodos estabelecidos em MESSAGE / UML.

Este método utiliza como processo uma adaptação do *Unified Development Process* (UDP) utilizado junto com UML em orientação a objetos. Ele abrange o ciclo completo de desenvolvimento de sistemas multiagentes. Inclui suporte para a obtenção de requisitos, análise, *design*, codificação e testes. É destinado ao uso geral, ou seja, sem restrições a determinados domínios de problemas. O método é suportado pelo kit de desenvolvimento INGENIAS (IDK), que contém um editor gráfico para especificações dos modelos e diagramas de sistemas multiagentes.

Além disso, a ferramenta *INGENIAS agent framework (IAF)* (GÓMEZ-SANZ, 2008) integrada no IDK, é um conjunto de bibliotecas que facilitam a implementação de agentes INGENIAS e são construídas com base no *Java Agent Development Framework (JADE)* (BELLIFEMINE et al., 2001).

Segundo Booch e colaboradores (1999), o UDP e sua versão estendida, o RUP, distribuem as tarefas de análise e *design* em quatro fases consecutivas (Figura 4): concepção,

elaboração, construção e transição. Cada fase pode ter várias iterações, onde cada iteração significa um ciclo de desenvolvimento completo. Uma iteração executa vários *workflows* (ou seja, conjunto de tarefas), por exemplo, requisitos, análise e *design*, e implementação. A realização da sequência de iterações de todas as fases leva ao sistema final.

Figura 4 – Fases do ciclo de vida para o método INGENIAS.



Fonte: GÓMEZ-RODRÍGUEZ et al., 2014.

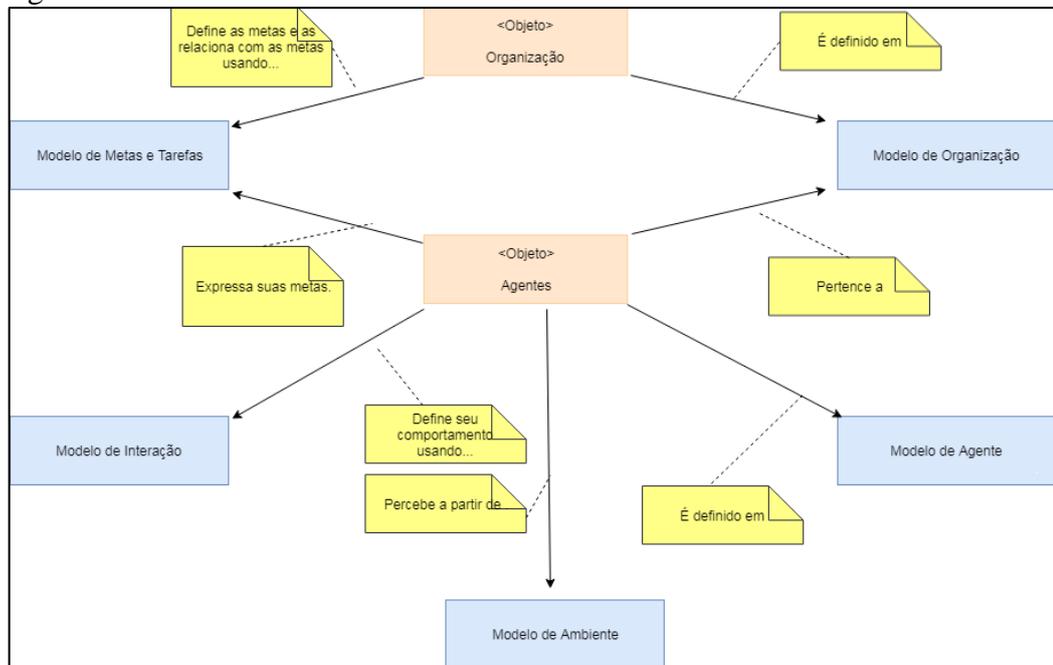
A abordagem orientada para modelos adotada em INGENIAS (PAVÓN et al., 2005) é adaptada do *model driven development* (MDD) (FRANCE; RUMPE, 1980) e promove o foco no desenvolvimento da especificação de sistemas e geração semiautomática deles a partir de outros artefatos, como documentação, código, testes e *scripts* de configuração. INGENIAS suporta esse tipo de desenvolvimento com ferramentas como o IDK (GÓMEZ-SANZ, 2008) e a IAF, que geram automaticamente código executável a partir das especificações do sistema. Esta abordagem é baseada na definição de metamodelo (GÓMEZ-SANZ, 2008), que descreve os elementos que constituem um sistema multiagentes de vários pontos de vista (ou seja, modelos). Estes pontos de vista são: (i) definição, controle e gerenciamento do estado mental de cada agente; (ii) interações do agente, (iii) organização do SMA, (iv) ambiente e (v) as tarefas e metas atribuídas a cada agente.

2.4.1 Metamodelos INGENIAS

Um metamodelo define as propriedades primitivas e sintáticas de um modelo. Os metamodelos INGENIAS são cinco (Figura 5): modelo de organização, modelo de agentes, modelo de metas e tarefas, modelo de interação e modelo de ambiente. Esses metamodelos correspondem às cinco visões diferentes do sistema, e são gerados durante a fase de concepção do processo.

Cada metamodelo especifica vários tipos de diagramas, que podem ser usados para descrever sua perspectiva do sistema. Assim, a especificação de um sistema em INGENIAS inclui cinco modelos (um para cada perspectiva), que, por sua vez, podem incluir diagramas diferentes. Cada um dos metamodelos considerados suporta vários níveis de abstração. Isso é devido a dois motivos principais. Primeiro, esses metamodelos pretendem ser uma definição precisa da linguagem de especificação. Em segundo lugar, eles apresentam todos os elementos de modelagem do método INGENIAS exigidos pelas ferramentas de MDD, em particular pelo IDK (GÓMEZ-SANZ, 2008), para orientar a geração de código, para sistemas multiagentes e ferramentas de suporte, conforme apresentado na figura 5:

Figura 3 – Metamodelos do método INGENIAS.



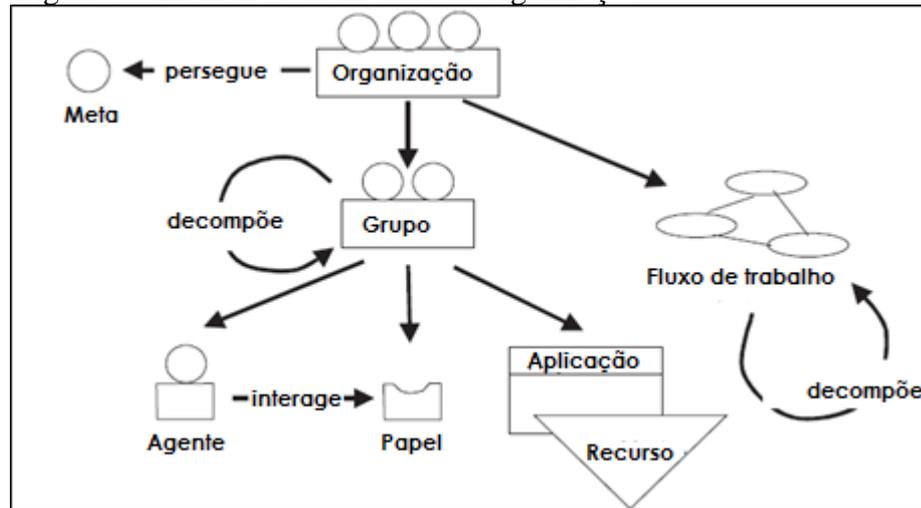
Fonte: GÓMEZ-RODRÍGUEZ et al., 2014.

2.4.2 Modelo de organização

O modelo de organização (Figura 6) descreve os agentes, recursos, suas tarefas e suas metas. É definido por sua estrutura, funcionalidade e relações sociais.

Do ponto de vista estrutural, a organização é um conjunto de entidades com relacionamentos de agregação e herança. A estrutura da organização define uma decomposição do sistema multiagentes em grupos e fluxos de trabalho (Figura 6).

Figura 6 – Elementos do modelo de organização.

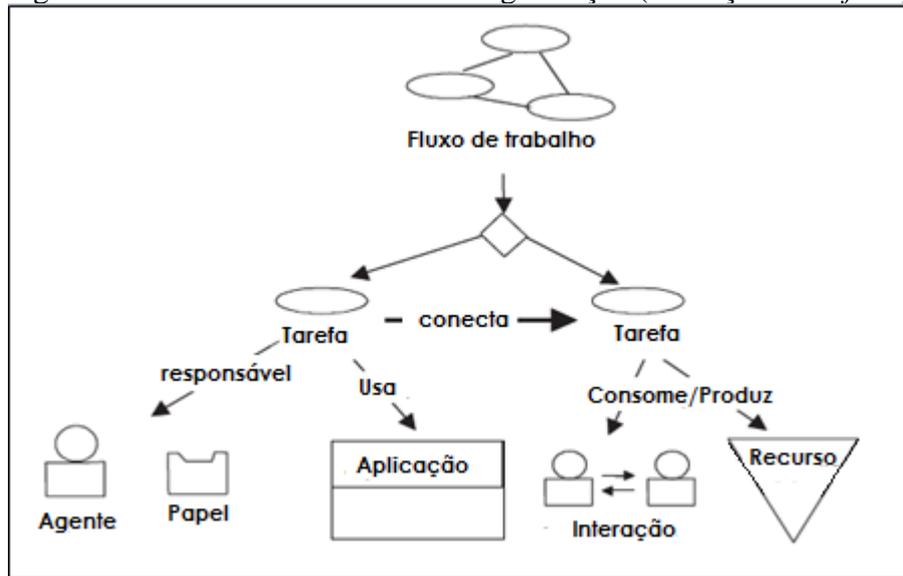


Fonte: GOMEZ-SANZ, 2008.

Os grupos podem conter agentes, papéis, recursos ou aplicativos. Os grupos são úteis quando o número de elementos em um sistema multiagentes aumenta. As atribuições de tais elementos para um grupo obedecem a algum propósito organizacional, porque o agrupamento facilita a definição de fluxos de trabalho (*workflows*), ou porque seus membros têm algumas características comuns.

A funcionalidade da organização é definida por sua finalidade e tarefas. A organização tem uma ou mais metas e depende de seus agentes para executar tarefas necessárias para alcançá-las. Os fluxos de trabalho (Figura 7) definem como essas tarefas estão relacionadas e quem é responsável pela sua execução.

Figura 7 – Elementos do modelo de organização (definição *Workflow*).



Fonte: GOMEZ-SANZ, 2008.

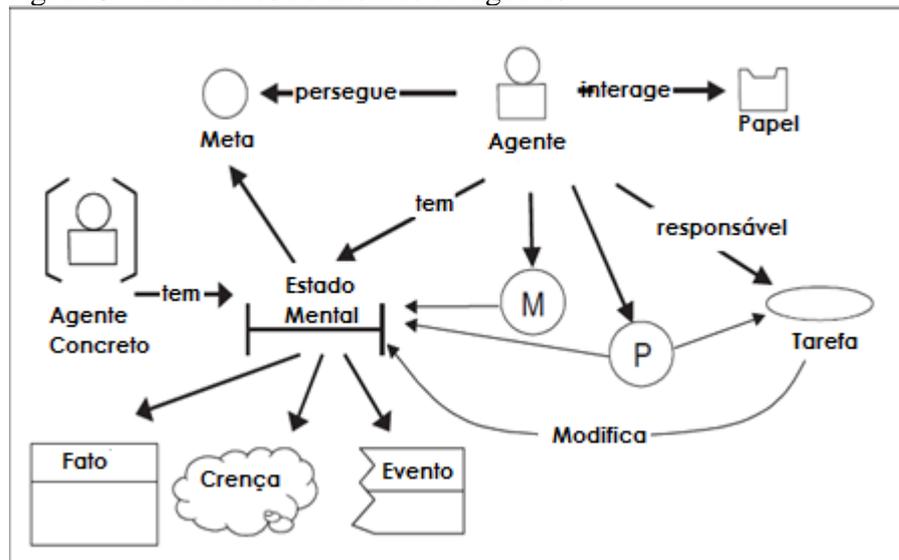
Os fluxos de trabalho (*workflows*) definem associações entre tarefas e informações gerais sobre suas execuções (Figura 7). Por exemplo, para cada tarefa, um *workflow* define seus resultados, o agente ou o papel responsável pela sua execução, e os recursos que são requeridos. Isso é útil para obter conhecimento sobre as relações entre agentes através de tarefas, e a atribuição e disponibilidade de recursos em uma organização

2.4.3 Modelo de agentes

Um agente é um programa que segue o princípio da racionalidade (NEWELL, 1982) e processa o conhecimento. O princípio da racionalidade diz que um agente executa as ações que o façam atingir suas metas.

O ponto de vista do modelo de agentes (Figura 8) diz respeito à funcionalidade de cada agente: propósito (quais metas um agente está comprometido em perseguir), as responsabilidades (quais tarefas ele precisa executar) e recursos (quais funções ele possui). O comportamento do agente é definido através de três componentes: estado mental, gerente do estado mental e processador do estado mental.

Figura 8 – Elementos do modelo de agentes.



Fonte: GOMEZ-SANZ, 2008.

Estado mental (*mental state*) é uma agregação de entidades mentais, como metas, crenças, fatos e compromissos. Cada agente tem um estado mental inicial representado por uma associação do agente a uma entidade de estado mental. Só pode haver uma dessas associações.

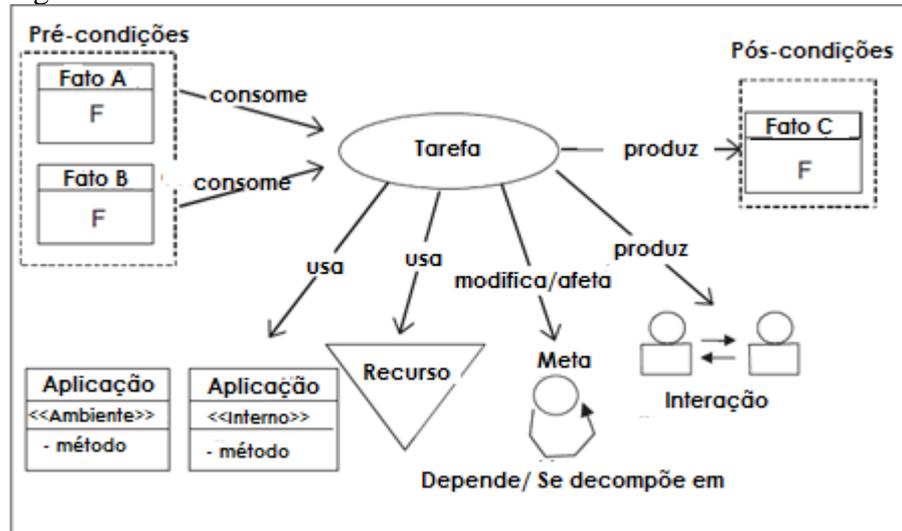
Gerente de estado mental (*mental state manager*) provê operações para criar, destruir, e modificar entidades mentais.

Processador de estado mental (*mental state processor*) determina como o estado mental evolui, descrito em termos de regras, planejamento e assim por diante.

2.4.4 Modelo de metas e tarefas

O modelo de metas e tarefas (Figura 9) considera a decomposição de metas e tarefas e descreve as consequências de realizar uma tarefa e o porquê de ela ser realizada (ou seja, justifica a execução de tarefas como forma de satisfazer as metas). Para cada tarefa, determinam-se quais os elementos necessários e quais saídas são esperadas. Para identificar quais metas são influenciadas pela execução de uma tarefa, existem relacionamentos de satisfação e falha. Finalmente, o modelo de metas e tarefas explica como uma meta solucionada afeta outras metas existentes usando a decomposição e relações de dependência. É útil saber que na resolução de uma submeta, uma super meta também pode ser resolvida.

Figura 9 – Elementos do modelo de metas e tarefas.



Fonte: GOMEZ-SANZ, 2008.

Diagramas desta visão podem ser usados para explicar como o processador do estado mental e o gerente de estado mental trabalham. Um desenvolvedor pode definir tarefas especiais, que expressem o que fazer quando novos elementos aparecem no estado mental, ou tarefas que produzem entidades especiais que representam a próxima tarefa a executar.

2.4.5 Modelo de interação

O modelo de interação aborda a troca de informações ou pedidos entre agentes, ou entre agentes e usuários humanos. A definição de uma interação requer: a identificação de atores na interação, especificação de interação, contexto da interação e natureza da interação.

Os atores na interação podem desempenhar o papel de iniciador ou de colaboradores. Deve haver pelo menos um colaborador e apenas um iniciador.

A especificação de interação deve detalhar como a interação é construída em tempo de execução. Isso se refere ao protocolo seguido ao longo da interação, mas também pode incluir as atitudes mentais que os agentes devem observar e ações desencadeadas da entrega ou recebimento de uma mensagem.

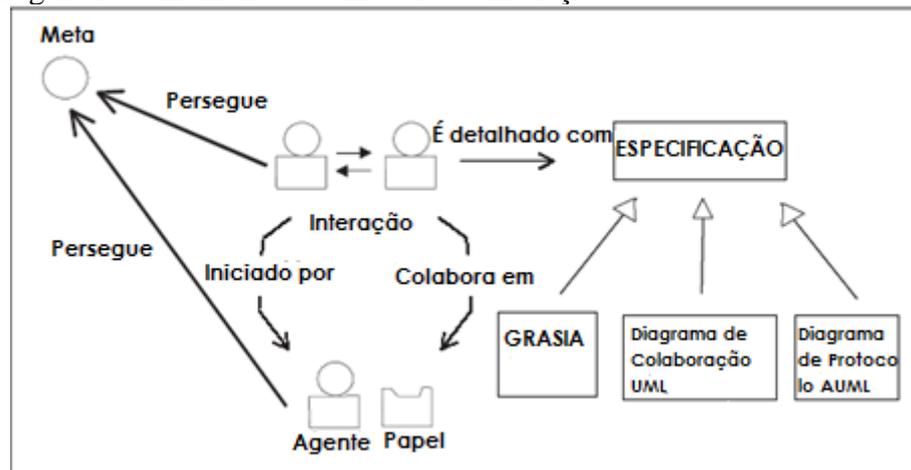
O contexto da interação deve definir as metas da interação e os estados mentais que seus participantes passam quando a interação começa, e durante a execução devem ter algo em

comum. Este contexto da interação pode ser representado por meio de um diagrama de metas e tarefas.

A natureza da interação refere-se à atitude de interação dos participantes: eles negociam, cooperam, realizam planejamento distribuído, e assim por diante. Esta natureza é detalhada na hierarquia definida por Huhns e Stephens (2000). Conhecer a natureza com antecedência pode ajudar a localizar um protocolo adequado, ou a identificar restrições de controle.

A figura 10 mostra os elementos que participam do modelo de interação. Para cada interação, existe uma especificação de interação associada, que pode ser especializada com as necessidades do desenvolvedor. INGENIAS não usa um único formalismo para especificação da interação. Admite diferentes notações para especificar interações, como diagramas de protocolo de agente UML.

Figura 10 – Elementos do modelo de interação.



Fonte: GOMEZ-SANZ, 2008.

2.4.6 Modelo de ambiente

O modelo de ambiente define as entidades com as quais o sistema multiagentes interage, que podem ser: recursos, outros agentes ou aplicações.

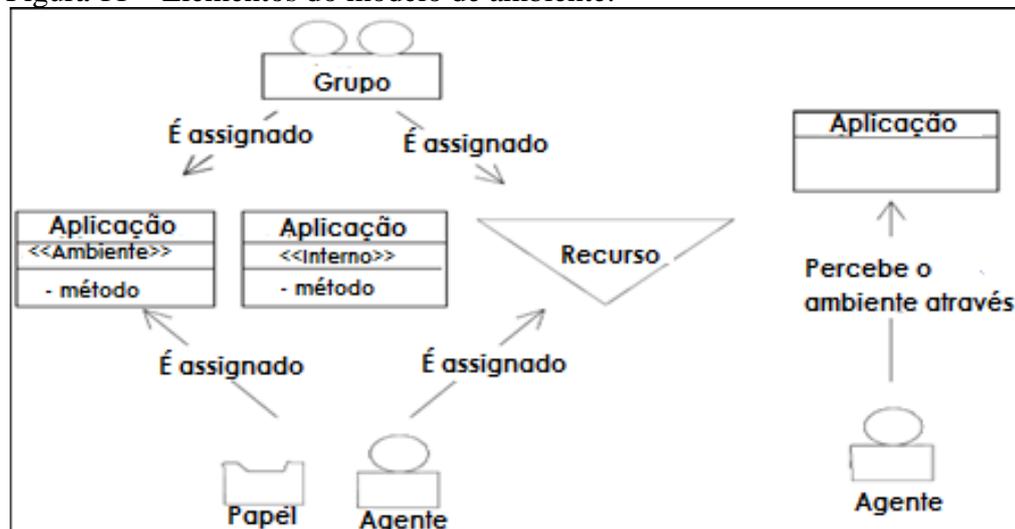
Os recursos são os elementos exigidos por tarefas que não fornecem uma *Application Programming Interface* (API) concreta. Exemplo de recursos: Unidade central de Processamento (CPU).

Outros agentes são agentes de outras organizações existentes e satisfazem o princípio racionalidade (NEWELL, 1982). Os agentes no SMA interagem com esses agentes para satisfazer as metas do sistema.

Normalmente, as aplicações oferecem algum API (local ou remoto). Seu principal uso é para expressar a percepção e a ação dos agentes; aplicações produzem eventos que podem ser observados.

A figura 11 mostra que um agente define sua percepção através de suas aplicações. A percepção pode ser concebida como a notificação de eventos ocorridos na aplicação, ou como a pesquisa do estado do aplicativo através de seus métodos, para procurar mudanças concretas. Sistemas mais complexos podem precisar de outros tipos de percepção. Aplicações e recursos podem ser tratados por grupos, papéis ou agentes. Quando um recurso ou uma aplicação é atribuída a um agente, grupo ou função, estes são responsáveis por controlar a sua utilização e acesso.

Figura 11 – Elementos do modelo de ambiente.



Fonte: GOMEZ-SANZ, 2008.

2.4.7 Processo de desenvolvimento do INGENIAS

No desenvolvimento de projetos pequenos não é necessária muita gerência, mas quando o projeto é maior, satisfazer as dependências pode consumir a maior parte do esforço de desenvolvimento. Para aliviar essas dependências, é proposto gerar a especificação em um

refinamento gradual seguindo um processo de desenvolvimento de engenharia de *software* convencional. Este processo foi personalizado para produzir os elementos requeridos numa especificação de sistema multiagentes.

Inicialmente, INGENIAS considera o UDP (BOOCH et al., 1999) como seu desenvolvimento de *software*. INGENIAS processa e define um conjunto de atividades inter-relacionadas (cerca de cem) que podem levar o desenvolvedor para a especificação final do sistema multiagentes.

O quadro 2 resume os resultados obtidos em cada fase do UDP para análise e atividades de *design*, aplicadas à especificação de sistemas multiagentes em INGENIAS.

Quadro 2 – Resultado das fases de análise e design do processo de desenvolvimento com INGENIAS.

		Fases			
		Concepção	Elaboração	Construção	
Workflows	Análise	Gerar casos de uso e identificar as ações deles com modelos de interação .	Refinar casos de uso.	Estudar os outros Casos de Uso e refiná-los.	
		Esboçar a arquitetura do sistema com um modelo de organização .	Criar modelo do agente que detalha elementos da arquitetura do sistema; <i>Workflows</i> e tarefas nos modelos de organização .		
		Gerar modelo de ambiente para representar os resultados da fase de coleta de requisitos.	Modelos de metas e tarefas a destacar restrições de controle. Refinar o modelo de ambiente para incluir novos elementos do ambiente.		
	Design	Gerar protótipos talvez com ferramenta de desenvolvimento rápido como ZEUS ou <i>Agent Tool</i> .		Refinamentos de <i>workflows</i> .	Gerar novos modelos de agentes e refinar os existentes.
				Modelos de interação que mostram como as tarefas são executadas.	
				Modelos de metas e tarefas que refletem dependências e necessidades identificadas nos <i>workflows</i> .	Gerar um modelo de implantação.
				Modelo de agente que mostra os padrões de estado mental necessários.	

Fonte: PAVÓN et al., 2005.

2.5 Método INGENIAS Scrum

Esta seção tem como objetivo apresentar o método INGENIAS Scrum que adota o método INGENIAS num processo de desenvolvimento para sistemas multiagentes seguindo a filosofia Scrum.

O método INGENIAS, que foi explicado no item 2.4, ressalta que é um método utilizado para casos gerais, ou seja, sem restrições de domínio (GÓMEZ-SANZ, 2008).

INGENIAS é baseado nos diagramas UML e no processo de desenvolvimento RUP, facilitando seu uso e aprendizado. Novos modelos são adicionados e os diagramas UML são enriquecidos com conceitos de introdução a agentes e organizacional.

De acordo com Pavón e colaboradores (2006), nos últimos anos, os metamodelos INGENIAS demonstraram sua capacidade e maturidade, dando suporte às especificações para o desenvolvimento dos sistemas multiagentes.

2.5.1 A aplicação do método Scrum no método INGENIAS

Nesta seção, o objetivo é apresentar como uma técnica de gerenciamento de projetos ágeis, como Scrum, pode ser integrada a um método como INGENIAS.

Ao tentar mapear um método com seu processo bem estabelecido em um novo processo, é necessário definir os passos a serem realizados para obter o produto desejado. Segundo Gómez-Rodríguez e González-Moreno (2008), há várias etapas que devem ser seguidas na definição de um novo desenvolvimento de modelos de processo para *Agent Oriented Software Engineering Methodologies* (AOSEM). Essas etapas são:

- i. Identificar o modelo de processo com um modelo de processo existente, se possível: se não for possível, definir o modelo de processo do zero, tomando como base os próximos passos.
- ii. Definir o ciclo de vida: identificar as fases, iterações e sequência de aplicações.
- iii. Definir as disciplinas: aqui, os recursos são o objeto das atividades definidas.
- iv. Definir o guia: este proporciona as informações sobre os elementos dos modelos.

Após aplicar os passos anteriores ao IAF e Scrum, os resultados podem ser sintetizados nos seguintes aspectos:

- i. Modelar os processos: existem vários sistemas multiagentes que podem ser construídos rapidamente com o INGENIAS, reutilizando os desenvolvimentos anteriores.
- ii. Definir ciclo de vida: no Scrum, cada lançamento (*release*) é produzido dentro de um número de iterações chamadas *Sprints*. Todo o processo é feito

em duas fases: a fase de preparação (antes do primeiro *sprint*) e a fase de *sprints* [*sprints* sucessivos que levam ao lançamento (*release*)].

- iii. Definir disciplinas: a equipe define as tarefas requeridas para desenvolver o *sprint*: *Daily Scrum*, *Initiate Product Backlog*, *Manage problems*, *Plan Release*, *Sprint Planning*, *Sprint Retrospective*, *Sprint Review* e *Update product backlog*.
- iv. Definir guia: o guia para a implementação do processo do INGENIAS pode ser encontrado na documentação do IAF.

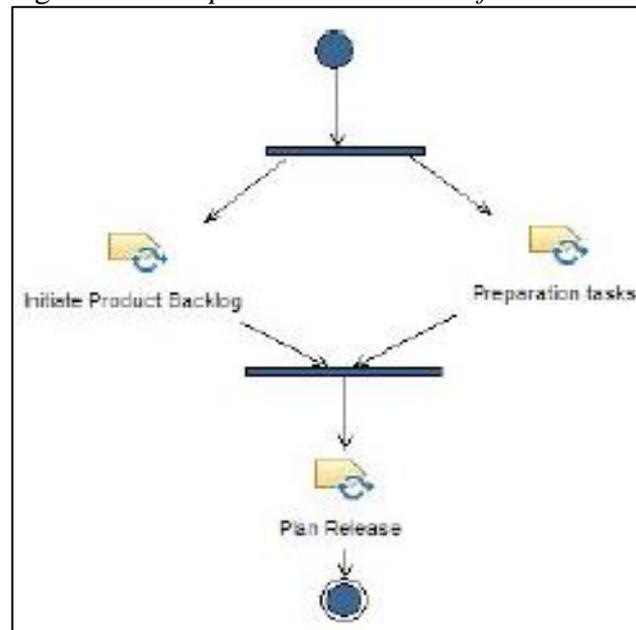
2.5.2 O ciclo de vida

Embora o Scrum não descreva atividades de engenharia para o desenvolvimento de produto, o processo do INGENIAS Scrum precisou descrever as atividades para que se ajustasse às recomendações do IAF. Alguns aspectos deste método são descritos a seguir:

- Agentes: são os principais blocos de construção do sistema.
- Interações: descrevem como os agentes se coordenam.
- Tarefas: são as unidades básicas do comportamento do agente.
- Desenvolvimento: expressa como os agentes são instanciados e inicializados no sistema final.
- Testes: são descritas as unidades de testes que o sistema multiagentes deve passar.
- Meta: os agentes possuem metas a serem alcançadas.
- Estado mental: as tarefas desempenhadas pelos agentes dependem de seus estados mentais e de suas metas perseguidas.

INGENIAS Agent Framework requer o uso do kit IDK. De acordo com o IDK, a definição da fase de preparação do Scrum (Figura 12) é composta pelas tarefas: *Initiate Product Backlog*, *Plan Release* e *Preparation Tasks*.

Figura 12 – *Preparation Phase workflow.*



Fonte: GÓMEZ-SANZ, 2008.

A priori é importante ressaltar os conceitos necessários das funções exercidas dentro do método Scrum. São eles:

- *Product Owner*: é a pessoa que representa o cliente e define os itens que compõem o *Product Backlog*.
- *Scrum Master*: Tem papel de facilitador, deve agir como um técnico para a equipe e para o *Product Owner*.
- *ScrumTeam*: São os principais responsáveis pela atividade de *Preparation Tasks* que é realizada junto com a *Initiate Product Backlog*.
- *Stakeholder*: Participação opcional, pode ser representado por qualquer pessoa que não participa diretamente do projeto, mas pode influenciar o desenvolvimento do produto.

O *backlog* do produto (*Initiate Product Backlog*) é uma lista prioritária de itens que contém descrições curtas de toda a funcionalidade desejada para o produto.

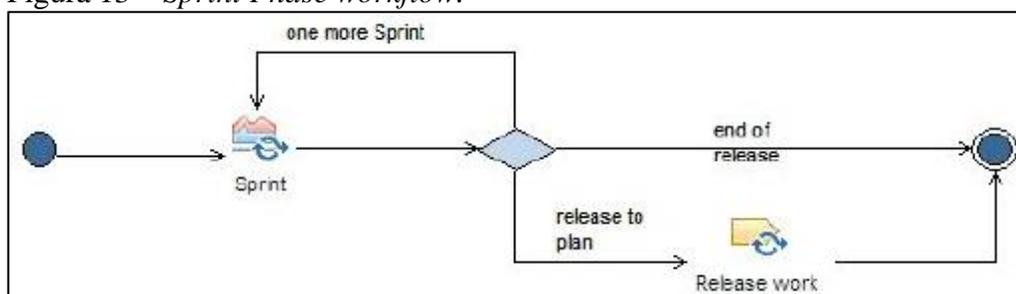
Na fase de *preparation tasks*, é criada uma constante especificação destinada a reorganizar, formalizar e estruturar as especificações preliminares no *product backlog*. Esta atividade é realizada pelo *scrumteam* a fim de preparar as próximas reuniões com o *productowner*. Neste estágio do processo, começa-se a criação dos modelos de organização e agentes.

Na etapa de *plan release*, definem-se os *sprints* requeridos para definir as metas.

Na fase dos *sprints*, algumas outras definições são importantes para o melhor entendimento do processo.

O projeto é realizado em iterações ou ciclos de até um mês de calendário chamado de *sprints* (Figura 13). São *timeboxed* (duração fixa) para que tenham sempre um início e fim, data fixa, e, geralmente, todos eles devem estar com a mesma duração. O *sprint backlog* é uma lista de tarefas que o *Scrum team* se compromete a fazer em um *sprint*. O *sprint retrospective* ocorre ao final de um *sprint* e serve para identificar o que funcionou bem, o que pode ser melhorado e que ações serão tomadas para melhorar. A *sprint review meeting* ocorre durante esta reunião, o *Scrum team* mostra o que foi alcançado durante o *sprint*. Por fim, durante o *sprint planning meeting*, o *product owner* descreve as funcionalidades (itens) de maior prioridade para a equipe. A equipe faz perguntas durante a reunião de modo que seja capaz de quebrar as funcionalidades em tarefas técnicas, após a reunião. Essas tarefas dão origem ao *sprint backlog*.

Figura 13 – *Sprint Phase workflow*.



Fonte: GÓMEZ-SANZ, 2008.

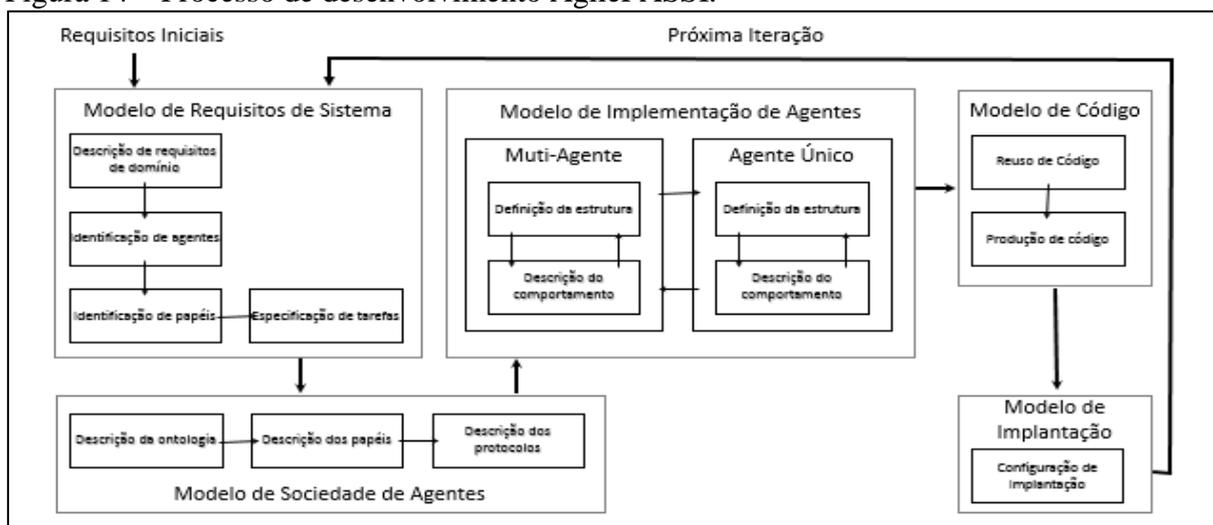
2.6 Método AgilePASSI

O método AgilePASSI (CHELLA et al., 2004) é oriundo do método *Process for Agent Societies Specification and Implementation* (PASSI) (COSSENTINO; POTTS, 2002).

Por se tratar de um método ágil, ele possui alguns aspectos fundamentais. São eles: simplicidade e velocidade, incrementabilidade e interação, liberação rápida de protótipos, promoção para trabalhos cooperativos (desenvolvedores e clientes colaborando com comunicação próxima) e facilidade para aprender, modificar e enfrentar mudanças.

A arquitetura do método AgilePASSI conta com quatro modelos principais (Figura 14): modelo de requisitos do sistema, modelo de sociedade de agentes, modelo de código e modelo de teste.

Figura 14 – Processo de desenvolvimento AgilePASSI.



Fonte: COSSENTINO; POTTS, 2002.

A fase de definição do modelo de requisitos de sistema é dividida em duas etapas, são elas: planejamento e descrição de requisitos de subdomínio. Na primeira etapa realiza-se o planejamento de uma ou mais iterações privilegiando as comunicações entre a equipe de desenvolvimento e os componentes, por meio de uma análise de riscos e requisitos. Na segunda etapa, os diagramas comuns de casos de uso UML são usados para representar uma descrição funcional do sistema. O termo sub refere-se à oportunidade de dividir o problema maior em subproblemas. O resultado é uma decomposição do problema em vários problemas menores e mais fáceis, que são resolvidos com iterações sequenciais.

Na identificação do modelo de requisitos se identificam os agentes, onde cada funcionalidade do agente é representada com um pacote de caso de uso. Também se identificam os papéis que são constituídos por diagramas de sequência que exploram as responsabilidades de cada agente através de cenários específicos da função. E, por fim, se especificam as tarefas onde o comportamento de cada agente é especificado em diagramas de atividades.

A fase de definição do modelo de sociedade de agentes é elaborada em duas etapas principais: a identificação do agente e a descrição de domínio de ontologia. Nesta primeira etapa obtêm-se como entrada os diagramas de casos de uso selecionados para a iteração atual,

desta forma, os casos de uso devem ser divididos em grupos de responsabilidades (os agentes do sistema). No diagrama, isso significa agrupar um ou mais casos de uso em pacotes estereotipados com rótulos de agente.

A atividade de descrição de domínio de ontologia visa capturar o domínio no qual o sistema representa, identificando entidades como classes UML. Utilizam-se diagramas de classe para descrever o conhecimento atribuído a agentes individuais e suas comunicações. Além disso, descrevem-se os papéis e protocolos utilizando diagrama de classes e de sequências, respectivamente.

O terceiro modelo é o de códigos, em que a sua fase de elaboração também inclui duas partes principais: reuso do código e produção de código. No primeiro, os desenvolvedores tentam reutilizar padrões de *design* de agentes, obtendo assim peças de código reutilizável, documentadas com uma visão estrutural e comportamental. A etapa de produção de código completa manualmente o código produzido anteriormente, colocando em prática todas as regras da programação extrema. Na geração de código a partir do modelo, usam-se as funcionalidades do PASSI *toolkit*.

A quarta e última fase de elaboração usa o modelo de testes. É planejado antes da fase de codificação e é realizado logo após a codificação. Nesta arquitetura de processos, o desenvolvedor deve preparar um ou mais testes, que deverão satisfazer o componente após a fase de codificação. Quando a fase de teste termina com sucesso, uma versão de trabalho dos agentes é liberada.

2.7 Comparação dos métodos INGENIAS, AgilePASSI E INGENIAS Scrum

O quadro 3 apresenta dados comparativos entre os métodos estudados nesta dissertação. Alguns critérios são observados nesta lista comparativa, como: abordagem (tradicional ou ágil), origem, orientação a agentes, nível de dificuldade na identificação de agentes, fases, necessidade de iteração, modelos, artefatos, flexibilidade, tipos de reunião, papéis e tamanho da equipe.

Quadro 3 – Comparação entre métodos INGENIAS, AgilePASSI e INGENIAS Scrum.

	INGENIAS	AgilePASSI	INGENIAS Scrum
Abordagem	Tradicional	Ágil	Ágil
Origem	MESSAGE/UML	PASSI	UML/RUP
Orientação do método	Orientado a projetos e a produção de produtos.	Orientado a pessoas e foco no desenvolvimento de agentes.	Orientado a pessoas com foco na motivação e comprometimento dos envolvidos.
Fases	Concepção, elaboração e construção.	Definição de requisitos, modelo de sociedade de agentes, plano de teste, codificação e testes.	Fase de preparação e fase de <i>sprints</i> .
Modelos desenvolvidos	Modelo de organização, modelo de agentes, modelo de metas e tarefas, modelo de ambiente e modelo de interação.	Modelo de requisitos do sistema, modelo de sociedade, modelo de código e modelo de teste.	Modelo de organização, modelo de agentes, modelo de metas e tarefas, modelo de ambiente e modelo de interação.
Artefatos	Diagrama de casos de uso, diagrama de organização, diagrama de agentes, diagrama de metas e tarefas, diagrama de ambiente, diagrama de interação e modelo de códigos.	Estórias de usuários, diagramas de caso de uso UML, diagramas de classes de agentes, plano de testes e código.	Diagrama de casos de uso, diagrama de organização, diagrama de agentes, diagrama de metas e tarefas, diagrama de ambiente, diagrama de interação, <i>product backlog</i> , <i>Sprint backlog</i> , <i>BurnDown Chart</i> , estórias de usuários e código.
Tipo de reunião	Não possui	Não possui	<i>Sprint planning</i> , <i>daily meeting</i> , <i>sprint review</i> , <i>sprint retrospective</i> , <i>FUP meeting</i> e <i>reestimate meetings</i>
Equipe	Não define	Não define	<i>Scrum Master</i> , <i>Product Owner</i> , e <i>Team Scrum</i> .
Tamanho da equipe	Não possui	Não possui	Máximo 15 participantes.

Legenda: FUP - *Follow-up*; UML - *Unified Modeling Language*; RUP - *Rational Unified Process*.

Fonte: A autora, 2018.

3 CONCEITOS SOBRE ESTUDO DE CASO

Este capítulo apresenta as principais definições a respeito do estudo de caso utilizado nesta dissertação. Além de detalhar os conceitos e métodos para desenvolvimento de jogos médicos educacionais.

3.1 Principais definições

A expressão "estudo de caso" é utilizada com frequência em artigos de engenharia de *software*. A característica principal deste método de pesquisa é o fato de estudar um caso específico, em contraste com uma amostra de uma população especificada (WHOLIN et al., 2012).

O estudo de caso é uma estratégia de pesquisa comumente usada em áreas como psicologia, sociologia, ciência política, trabalho social e negócios. Nessas áreas, estudos de caso são conduzidos com os objetivos de conhecimento não crescente (por exemplo, conhecimento sobre indivíduos, grupos, e organizações e sobre fenômenos sociais, políticos e relacionados), mas também provoca mudanças sobre o fenômeno em estudo (por exemplo, melhoria da educação ou da educação social). A pesquisa de engenharia de *software* tem objetivos semelhantes de alto nível, ou seja, para melhor compreender como e por que a engenharia de *software* deve ser realizada e, com este conhecimento, para tentar melhorar o processo de engenharia de *software* e os produtos de *software* resultantes.

Alguns autores definem de maneira simples e clara o conceito de estudo de caso.

A tendência central entre todos os tipos de estudo de caso é que ele tenta iluminar uma decisão ou um conjunto de decisões: por que elas são tomadas, como elas são implementadas e com que resultado (SCHRAMM, 1971).

O estudo de caso é um método de pesquisa que utiliza, geralmente, dados qualitativos, coletados a partir de eventos reais, com o objetivo de explicar, explorar ou descrever fenômenos atuais inseridos em seu próprio contexto. Caracteriza-se por ser um estudo detalhado e exaustivo de poucos, ou mesmo de um único objeto, fornecendo conhecimentos profundos (EISENHARDT, 1989; YIN, 2009).

Os estudos de caso oferecem uma abordagem que não exige um limite rigoroso entre o objeto de estudo e seu ambiente. Eles não geram os mesmos resultados sobre, por exemplo, relações causais, mas fornecem uma compreensão mais profunda dos fenômenos em estudo.

3.2 Processo para realização de um estudo de caso

Para realizar um estudo de caso é preciso organizá-lo em torno de suas necessidades, ou seja, deve-se pensar nas possíveis questões que precisam ser respondidas sobre o estudo em questão. De acordo com Gil (1995), o estudo de caso não requer um roteiro rígido para a sua delimitação, porém se torna possível definir quatro fases principais que determinam o seu delineamento:

- a) delimitação da unidade-caso;
- b) coleta de dados;
- c) seleção, análise e interpretação dos dados;
- d) elaboração do relatório.

A fase de delimitação da unidade-caso baseia-se em delimitar a unidade que constitui o caso. Esta fase requer que o pesquisador tenha aptidão para notar quais informações são necessárias e hábeis para compreender o objeto de maneira geral. Sabe-se que nem sempre os casos estudados são escolhidos usando critérios estatísticos, desta forma, recomendam-se algumas ações que devem ser aplicadas como: procurar casos típicos, optar por casos extremos, encontrar casos atípicos.

A etapa seguinte é denominada coleta de dados. Esta fase geralmente é executada com diversas técnicas quantitativas e qualitativas tais quais: observação, análise de documentos, entrevista formal ou entrevista informal, aplicação de questionário de caracterização com perguntas pré-estabelecidas, levantamentos de dados, análise de conteúdo, etc.

A terceira etapa é constituída pela seleção, análise e interpretação dos dados. A seleção dos dados analisa as metas da pesquisa, seus limites e um sistema com o objetivo de ponderar quais dados são úteis, ou podem ser descartados. O pesquisador precisa definir no início um plano de análise e precisar levar em consideração as possíveis limitações dos dados colhidos, principalmente no que se refere à qualidade da amostra, pois tendo uma boa amostra, há um pretexto para generalizar o resultado obtido pela amostra a toda população de dados. Se a amostra não for boa, deve-se analisar os resultados em relação aos cálculos de probabilidade. É

sempre importante optar por categorias de análise que sejam reconhecidas no campo do conhecimento, para que não haja interpretação dos dados que envolvam julgamentos implícitos, preconceitos etc.

A quarta e última etapa é apresentada pela preparação de relatórios parciais e finais. Ressalta-se que se deve especificar como os dados foram coletados, qual teoria apoiou a categorização destes dados e comprovar a validade dos dados obtidos. O relatório precisa ser breve, no entanto, em algumas situações pode ser solicitado o relatório detalhado.

4 ESTUDO DE CASO: DESENVOLVIMENTO DO JOGO MEDEDUC COM INGENIAS, INGENIAS SCRUM E AGILEPASSI

Este capítulo aponta, detalhadamente, a contextualização do desenvolvimento do estudo de caso. Apresenta pontos importantes como o cronograma, preparação dos participantes, coleta de dados e a análise estatística e qualitativa do estudo de caso.

4.1 Visão geral do estudo de caso

O estudo de caso foi inspirado no trabalho de Ferreira (2016) e tem como objetivo estudar os métodos orientados a agentes no desenvolvimento de jogos médicos educacionais, com ênfase no desenvolvimento ágil. Além disso, este estudo de caso é sobre o MEDEDUC, um jogo médico educacional que tem como objetivo propagar conhecimento e, ao mesmo tempo, divertir os usuários. O jogo tem como foco o público de alunos da área médica. Ao propagar conhecimento, ele ensina os conceitos de medicina, fazendo com que os futuros médicos absorvam conceitos médicos relacionados às suas áreas de estudo.

O jogo educacional MEDEDUC foi proposto a partir do método proposto em Carro e colaboradores (2002).

De uma forma geral, o jogo é estruturado da seguinte maneira: O jogador (usuário) é apresentado a perguntas sobre conceitos médicos que são separadas por 5 níveis de dificuldade. Conforme o participante é aprovado em cada nível, ganha um equipamento médico para montar seu consultório virtual. Este equipamento pode ser um computador, estetoscópio, raio X, etc. Ao finalizar os cinco níveis, o jogador terá o consultório preenchido pelos equipamentos ganhos durante o jogo. Antes de iniciar os níveis, o jogador passa por uma fase de nivelamento em que são testados seus conhecimentos prévios e ele é direcionado para o nível apropriado ao seu grau de conhecimento. Caso o jogador não tenha o rendimento esperado (mais de 70% de acertos), ele é direcionado à fase de reforço. Durante todo o jogo o usuário recebe *feedbacks* sobre suas ações. No APÊNDICE A é apresentada a descrição detalhada do jogo.

Este estudo de caso foi realizado com uma turma do curso de Ciência da Computação, na disciplina de Inteligência Artificial do segundo semestre de 2017, em uma universidade pública do estado do Rio de Janeiro.

4.2 Objetivo e hipótese

Este estudo de caso teve como objetivo averiguar a utilização de procedimentos eficientes para o desenvolvimento de jogos médicos educacionais orientados a agentes. Por esta razão, três diferentes métodos foram escolhidos com características distintas: (i) o INGENIAS que é um método de desenvolvimento tradicional para sistemas multiagentes; (ii) o INGENIAS Scrum e o AgilePASSI que são métodos de desenvolvimento ágil para sistemas multiagentes.

O objetivo específico deste estudo de caso é comparar a utilização dos métodos ágeis INGENIAS Scrum e AgilePASSI, para verificar se conseguem entregar os artefatos de forma mais rápida e atendem aos requisitos no jogo médico educacional, com maior eficiência e conformidade em relação ao método tradicional.

Hipótese: Como hipótese para o estudo de caso temos que jogos médicos educacionais orientados a agentes construídos utilizando uma abordagem de desenvolvimento ágil orientado a agentes (AgilePASSI e INGENIAS Scrum), são responsáveis por definir e implementar melhor os requisitos pedidos (maior conformidade) e tem entrega mais rápida (maior eficiência) quando comparado a jogos desenvolvidos através de uma abordagem de desenvolvimento tradicional orientado a agentes (INGENIAS). A variável independente para esta hipótese é a utilização de metodologias próprias para agentes com abordagem de desenvolvimento tradicional (INGENIAS) versus desenvolvimento ágil (AgilePASSI e INGENIAS Scrum).

4.3 Contextualização do estudo de caso: MEDEDUC

O trabalho foi realizado da seguinte maneira: a turma de inteligência artificial composta por 19 alunos foi dividida em seis grupos, sendo 2 grupos compostos por 4 participantes, 3 grupos compostos por 3 participantes e 1 grupo composto por 2 participantes. Os alunos puderam escolher seus colegas participantes, porém sem saber qual método seria aplicado aos grupos. Os métodos foram sorteados para os seis grupos onde: grupos 1 e 2 eram responsáveis por aplicar o método orientado a agentes INGENIAS; os grupos 3 e 4 utilizaram o método orientado a agentes AgilePASSI; os grupos 5 e 6 fizeram a aplicação do método ágil INGENIAS Scrum.

O método do jogo foi proposto a eles de acordo com a descrição na seção 4.1 deste trabalho. Os encontros foram realizados às segundas-feiras iniciando no dia 02/10/2017 e terminando no dia 11/12/2017, utilizando a carga horária da disciplina de inteligência artificial, sendo que os conceitos estudados nas aulas teóricas puderam ser utilizados nos encontros práticos do desenvolvimento do estudo de caso. Além disso, foi acordado que um representante de cada grupo ficaria como o responsável por realizar as entregas dos artefatos previstos no cronograma. No dia 02/10/2017, na primeira aula, os seis grupos tiveram uma aula teórica sobre agentes de *software*, sistemas multiagentes, modelagem em sistemas multiagentes, exemplos de casos de utilização de sistemas multiagentes e desenvolvimento de sistemas multiagentes.

Os seis grupos também tiveram uma aula específica sobre cada método com duração de 3 horas. Foram abordados os conteúdos principais do método INGENIAS como: aplicação, como realizar a modelagem e ferramenta para o desenvolvimento de meta-modelos. Para o método AgilePASSI abordou-se temas como: principais aspectos, descrição e aplicabilidade dos principais modelos. E, para o método INGENIAS Scrum, foi necessário ressaltar os principais tópicos do método ágil Scrum, tais quais: aplicabilidade, estrutura e passos que devem ser seguidos para obter o resultado esperado neste método e, além disso, foram apresentados os meta-modelos INGENIAS e os tópicos vistos pelos grupos do método INGENIAS.

Após estas aulas teóricas, os participantes iniciaram suas atividades em seus respectivos grupos, de acordo com o cronograma apresentado e distribuído para o representante de cada grupo.

No início do desenvolvimento do trabalho, foi distribuído aos alunos um formulário visando entender as expectativas dos alunos quanto ao desenvolvimento do trabalho e à aplicação dos métodos, mapeando o perfil do aluno participante. Ao término do trabalho, foi enviado um novo formulário onde os alunos informaram as partes que sentiram mais dificuldade, foi avaliada a satisfação quanto ao método e se as expectativas iniciais haviam sido alcançadas.

4.4 Cronograma do Estudo de Caso

No primeiro encontro com os alunos, foi entregue um cronograma com todas as datas de entregas requeridas para submissão dos artefatos do método que foi aplicado em cada grupo (Quadro 4).

Quadro 4 – Cronograma do estudo de caso.

Artefatos e atividades		Grupos 3 e 5 INGENIAS	Grupos 2 e 6 AgilePASSI	Grupos 1 e 4 INGENIAS SCRUM
1	Aula sobre Agentes	02/10/2017	02/10/2017	02/10/2017
2	Aula sobre sistemas Multiagentes	02/10/2017	02/10/2017	02/10/2017
3	Aula sobre a Metodologia + Jogo	02/10/2017	02/10/2017	02/10/2017
4	Formulário - Início do projeto	02/10/2017	02/10/2017	02/10/2017
5	Formulário - Término do projeto	11/12/2017	11/12/2017	11/12/2017
6	Entrega do Jogo Médico Educacional	11/12/2017	11/12/2017	11/12/2017
7	Apresentação do Jogo Médico Educacional	12/12/2017	12/12/2017	12/12/2017
8	Discussão e Planejamento	09/10/2017	09/10/2017	09/10/2017
9	Modelo de Casos de Uso	16/10/2017	-	-
10	Modelo de Organização	23/10/2017	-	-
11	Modelo de Agentes	30/10/2017	-	-
12	Modelo de Metas e Tarefas	06/11/2017	-	-
13	Modelo de Interação	13/11/2017	-	-
14	Criar o modelo de Ambiente	27/11/2017	-	-
15	Entrega do Protótipo	04/12/2017	-	-
14	Lista de atividades ordenadas 1	-	16/10/2017	-
15	Diagramas de caso de uso UML	-	23/10/2017	-
16	Diagramas detalhados das funcionalidades e identificação dos agentes	-	30/10/2017	-
17	Lista de atividades ordenadas 2	-	31/10/2017	-
18	Plano de testes que será executado	-	06/11/2017	-
19	Lista de atividades ordenadas 3	-	13/11/2017	-
20	Lista de atividades ordenadas 4	-	23/11/2017	-
21	Protótipo Final	-	04/12/2017	-
22	Relatório de Entregas Previstas Sprint 1	-	-	16/10/2017
23	Reunião de revisão Sprint 1	-	-	23/10/2017
24	Relatório de Entregas Previstas Sprint 2	-	-	30/10/2017
25	Reunião de revisão Sprint 2	-	-	31/10/2017
26	Relatório de Entregas Previstas Sprint 3	-	-	06/11/2017
27	Reunião de revisão Sprint 3	-	-	13/11/2017
28	Relatório de Entregas Previstas Sprint 4	-	-	27/11/2017
29	Reunião de revisão Sprint 4	-	-	04/12/2017

Fonte: A autora, 2018.

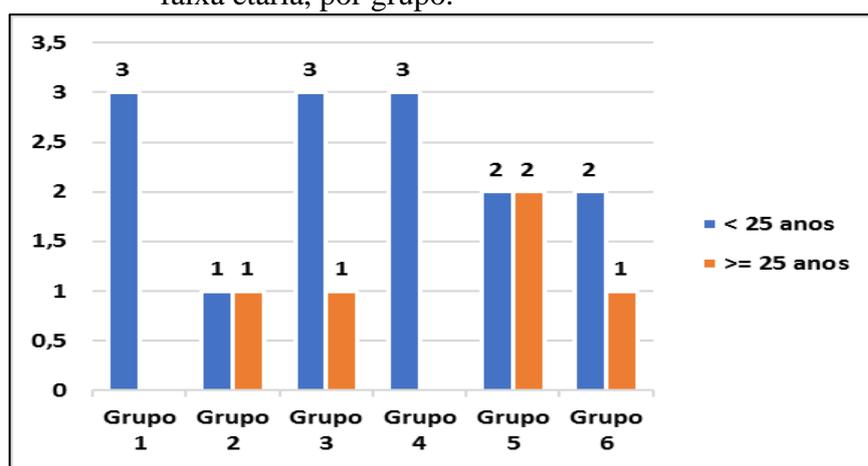
4.5 Preparação dos participantes

Ao iniciar o estudo de caso, os participantes envolvidos responderam um questionário de caracterização a fim de entender e determinar o perfil dos alunos. O questionário avaliava a faixa etária em que o participante estava incluído, o nível de experiência em sistemas

multiagentes (tanto no âmbito acadêmico, quanto no âmbito profissional). Outros quesitos, como: atuação no mercado de trabalho, nível de conhecimento sobre agentes, nível de conhecimento sobre sistemas multiagentes, conhecimento sobre o método proposto e participações anteriores no desenvolvimento de jogos e de soluções médicas. E, ao final, deixou-se um campo aberto para que os integrantes pudessem escrever as suas expectativas diante do trabalho desenvolvido. Para fins de avaliação deste questionário, a escala utilizada nas perguntas variava de 0 a 5, onde 0 representava baixo nível de conhecimento sobre o tema até 5 que representava alto nível de conhecimento sobre o tema.

Em relação a faixa etária dos envolvidos, o gráfico 1 apresenta, em relação aos grupos, a quantidade absoluta dos participantes menores de 25 anos ou que possuíam 25 anos ou mais.

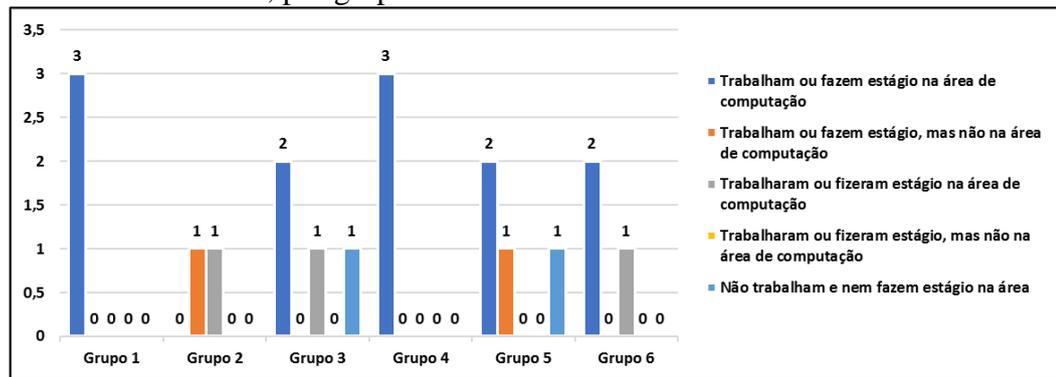
Gráfico 1 – Quantidade absoluta dos participantes em relação à faixa etária, por grupo.



Fonte: A autora, 2018.

Em relação a atuação no mercado de trabalho (Gráfico 2), constatou-se que os grupos foram bem distribuídos e em todos eles continham participantes que trabalham ou fazem estágio na área de computação ou trabalharam ou fizeram estágio na área de computação.

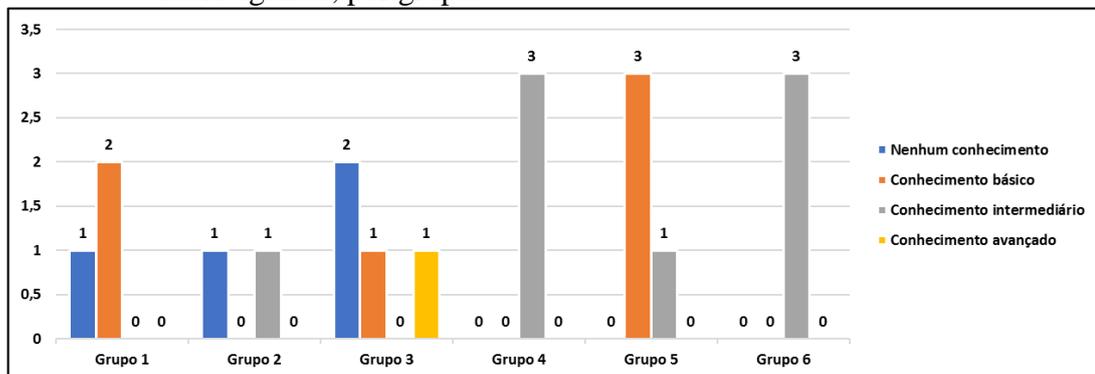
Gráfico 2 – Quantidade absoluta dos participantes em relação a atuação no mercado, por grupo.



Fonte: A autora, 2018.

Em relação aos conhecimentos sobre agentes (Gráfico 3) observou-se que os grupos estavam balanceados, pois apesar de 3 grupos conterem participantes sem nenhum conhecimento, também possuíam participantes com níveis de conhecimento entre básico e avançado.

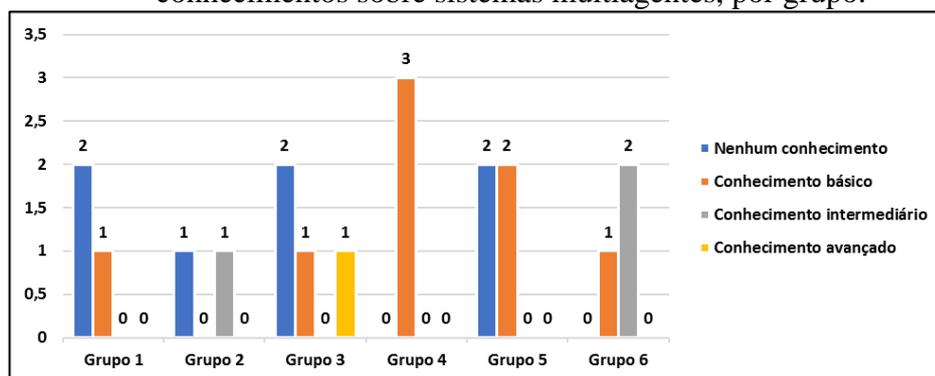
Gráfico 3 – Quantidade absoluta dos participantes em relação aos conhecimentos sobre agentes, por grupo.



Fonte: A autora, 2018.

Em relação à experiência na área de sistemas multiagentes (Gráfico 4) notou-se que todos os grupos estavam balanceados e que, somente, no grupo 2 não havia participante que possuía conhecimento básico, entretanto, 1 participante possuía conhecimento intermediário.

Gráfico 4 – Quantidade absoluta dos participantes em relação aos conhecimentos sobre sistemas multiagentes, por grupo.



Fonte: A autora, 2018.

Sobre o conhecimento do método proposto, 4 participantes no total (21,1%) responderam que possuíam algum conhecimento sobre ele.

Um resultado positivo foi verificado ao serem questionados quanto à participação em desenvolvimento de jogos, onde 10 participantes no total (52,6%) relataram já terem tido experiência nesta atividade.

Dentre os participantes, apenas 2 (10,5%) reportaram que tinham experiência em desenvolvimento de soluções médicas.

As principais expectativas quanto ao estudo de caso foram explicitadas pelos envolvidos e são apresentadas no ANEXO A – Expectativas dos envolvidos no estudo de caso.

O quadro 5 apresenta uma síntese entre os perfis dos participantes distribuídos em cada um dos métodos.

Desta maneira, observa-se que os participantes envolvidos no estudo de caso apresentam homogeneidade por estarem cursando o mesmo período na universidade, terem o mesmo grau de conhecimento sobre os assuntos apresentados e possuírem uma experiência semelhante o que proporciona um ambiente ideal para o estudo de caso.

Quadro 5 – Comparação dos envolvidos no estudo de caso, por método.

	INGENIAS	AgilePASSI	INGENIAS Scrum
Faixa etária	1 \geq 25 anos, enquanto 4 < 25 anos.	3 \geq 25 anos, enquanto 4 < 25 anos.	1 \geq 25 anos, enquanto 6 < 25 anos.
Atuação no mercado de trabalho	80% estão inseridos no mercado de trabalho.	71% estão inseridos no mercado de trabalho.	71% estão inseridos no mercado de trabalho.
Experiência em sistemas multiagentes	Nenhum aluno tem experiência.	Nenhum aluno tem experiência.	1 aluno tem experiência.
Nível de conhecimento em agentes	Conhecimento básico.	Conhecimento básico.	Conhecimento básico.
Nível de conhecimento em sistemas multiagentes	Conhecimento básico.	Conhecimento básico.	Conhecimento básico.
Conhecimento do método proposto	2 alunos.	2 alunos.	Nenhum aluno.
Participação em desenvolvimento de jogos	2 alunos.	2 alunos.	6 alunos.
Participação em desenvolvimento de soluções médicas	1 aluno.	Nenhum aluno.	1 aluno.

Fonte: A autora, 2018.

4.6 Coleta de dados do estudo de caso

Os dados que foram coletados no estudo de caso produziram pontuações alcançadas considerando os dias em atraso nas entregas dos artefatos e a conformidade dos requisitos requeridos para o desenvolvimento do jogo médico educacional. Estabeleceu-se a nota 0 para requisito não entregue, 1 para requisito parcialmente entregue e 2 para requisito entregue. Ademais, a versão final do jogo entregue foi observada e pontuada por um especialista em jogos médicos educacionais e por um cliente. Desta maneira, ao final do estudo de caso os dados coletados foram agrupados por grupos de trabalho e, por fim, pelos métodos aplicados no estudo de caso.

Durante o estudo de caso os alunos dos grupos 1 e 2 produziram os artefatos apresentados nos quadros 6 e 7 e ANEXOS B e C, respectivamente. Esses artefatos eram requeridos pelo método de desenvolvimento para sistemas multiagentes INGENIAS.

Quadro 6 – Artefatos requeridos para o grupo 1 - INGENIAS

	Artefato	Data prevista	Data entregue	Dias em atraso
1	Modelo de Casos de Uso	16/10/2017	16/10/2017	0
2	Modelo de Organização	23/10/2017	23/10/2017	0
3	Modelo de Agentes	30/10/2017	30/10/2017	0
4	Modelo de Metas e Tarefas	06/11/2017	06/11/2017	0
5	Modelo de Interação	13/11/2017	13/11/2017	0
6	Modelo de Ambiente	27/11/2017	27/11/2017	0
7	Entrega do Jogo	04/12/2017	08/12/2017	4
8	Apresentação do Jogo	12/12/2017	12/12/2017	0

Fonte: A autora, 2018.

Quadro 7 – Artefatos requeridos para o grupo 2 - INGENIAS

	Artefato	Data prevista	Data entregue	Dias em atraso
1	Modelo de Casos de Uso	16/10/2017	12/12/2017	57
2	Modelo de Organização	23/10/2017	12/12/2017	50
3	Modelo de Agentes	30/10/2017	12/12/2017	43
4	Modelo de Metas e Tarefas	06/11/2017	12/12/2017	36
5	Modelo de Interação	13/11/2017	12/12/2017	29
6	Modelo de Ambiente	27/11/2017	12/12/2017	15
7	Entrega do Jogo	04/12/2017	12/12/2017	8
8	Apresentação do Jogo	12/12/2017	12/12/2017	0

Fonte: A autora, 2018.

Os alunos dos grupos 3 e 4 produziram os artefatos apresentados nos quadros 8 e 9, e ANEXOS D e E, respectivamente. Esses artefatos utilizaram o método de desenvolvimento para sistemas multiagentes AgilePASSI.

Quadro 8 – Artefatos requeridos para o grupo 3 - AgilePASSI

	Artefato	Data prevista	Data entregue	Dias em atraso
1	Lista de atividades ordenadas 1	16/10/2017	23/10/2017	7
2	Diagramas de caso de uso UML	23/10/2017	06/11/2017	14
3	Diagramas detalhados das funcionalidades e identificação dos agentes	30/10/2017	06/11/2017	7
4	Lista de atividades ordenadas 2	31/10/2017	06/11/2017	6
5	Plano de testes executado	06/11/2017	06/11/2017	0
6	Lista de atividades ordenadas 3	13/11/2017	13/11/2017	0
7	Lista de atividades ordenadas 4	27/11/2017	27/11/2017	0
8	Entrega do Jogo	04/12/2017	04/12/2017	0
9	Apresentação do Jogo	12/12/2017	12/12/2017	0

Fonte: A autora, 2018.

Quadro 92 – Artefatos requeridos para o grupo 4 - AgilePASSI

	Artefato	Data prevista	Data entregue	Dias em atraso
1	Lista de atividades ordenadas 1	16/10/2017	16/10/2017	0
2	Diagramas de caso de uso UML	23/10/2017	23/10/2017	0
3	Diagramas detalhados das funcionalidades identificação dos agentes	30/10/2017	06/11/2017	7
4	Lista de atividades ordenadas 2	31/10/2017	13/11/2017	13
5	Plano de testes executado	06/11/2017	13/11/2017	7
6	Lista de atividades ordenadas 3	13/11/2017	27/11/2017	14
7	Lista de atividades ordenadas 4	27/11/2017	27/11/2017	0
8	Entrega do Jogo	04/12/2017	04/12/2017	0
9	Apresentação do Jogo	12/12/2017	12/12/2017	0

Fonte: A autora, 2018.

Os grupos 5 e 6 produziram os artefatos apresentados nos quadros 10 e 11 e ANEXOS F e G, respectivamente. Esses artefatos foram criados com o método de desenvolvimento para sistemas convencionais INGENIAS Scrum.

Quadro 10 – Artefatos requeridos para o grupo 5 – INGENIAS Scrum

	Artefato	Data prevista	Data entregue	Dias em atraso
1	Relatório de Entregas Previstas Sprint 1	16/10/2017	13/11/2017	28
2	Reunião de revisão Sprint 1	23/10/2017	13/11/2017	21
3	Relatório de Entregas Previstas Sprint 2	30/10/2017	30/10/2017	0
4	Reunião de revisão Sprint 2	31/10/2017	27/11/2017	27
5	Relatório de Entregas Previstas Sprint 3	06/11/2017	27/11/2017	21
6	Reunião de revisão Sprint 3	13/11/2017	27/11/2017	14
7	Relatório de Entregas Previstas Sprint 4	27/11/2017	27/11/2017	0
8	Entrega do Jogo	04/12/2017	04/12/2017	0
9	Apresentação do Jogo	12/12/2017	12/12/2017	0

Fonte: A autora, 2018.

Quadro 31 – Artefatos requeridos para o grupo 6 – INGENIAS Scrum

	Artefato	Data prevista	Data entregue	Dias em atraso
1	Relatório de Entregas Previstas Sprint 1	16/10/2017	16/10/2017	0
2	Reunião de revisão Sprint 1	23/10/2017	23/10/2017	0
3	Relatório de Entregas Previstas Sprint 2	30/10/2017	30/10/2017	0
4	Reunião de revisão Sprint 2	31/10/2017	31/10/2017	0
5	Relatório de Entregas Previstas Sprint 3	06/11/2017	06/11/2017	0
6	Reunião de revisão Sprint 3	13/11/2017	13/11/2017	0
7	Relatório de Entregas Previstas Sprint 4	27/11/2017	27/11/2017	0
8	Entrega do Jogo	04/12/2017	04/12/2017	0
9	Apresentação do Jogo	12/12/2017	12/12/2017	0

Fonte: A autora, 2018.

Os dias em atraso nas entregas dos artefatos de cada um dos grupos foram organizados no quadro 12.

Quadro 12 – Total de dias por semana em atraso nas entregas dos artefatos.

Semanas	DIAS EM ATRASO					
	Grupo 1 INGENIAS	Grupo 2 INGENIAS	Grupo 3 AgilePASSI	Grupo 4 AgilePASSI	Grupo 5 INGENIAS Scrum	Grupo 6 INGENIAS Scrum
Semana 1	0	57	7	0	28	0
Semana 2	0	50	14	0	21	0
Semana 3	0	43	7	7	0	0
Semana 4	0	36	6	13	27	0
Semana 5	0	29	0	7	21	0
Semana 6	0	15	0	14	14	0
Protótipo	4	8	0	0	0	0
Apresentação	0	0	0	0	0	0
Total	4	57	14	14	28	0
Máximo						

Fonte: A autora, 2018.

Ao final do estudo de caso, cada grupo recebeu uma pontuação variando de zero a dois para cada requisito (0, 1 ou 2), onde zero representa requisito não entregue, 1 requisito parcialmente entregue e 2 representa requisito entregue. Estes pontos podem ser vistos no quadro 13. Os requisitos avaliados como parcialmente entregues ocorreram em alguns casos, por exemplo para o requisito 15: Usuário pode interromper o nivelamento e deve iniciar o processo novamente e as respostas anteriores são desconsideradas. Consideraram-se parcialmente entregues os grupos que permitiram ao usuário interromper o nivelamento, porém as respostas anteriores não foram desconsideradas.

Quadro 13 – Pontuação dos requisitos (continua).

Código	Requisito	Grupo 1 INGENIAS	Grupo 2 INGENIAS	Grupo 3 AgilePASSI	Grupo 4 AgilePASSI	Grupo 5 INGENIAS Scrum	Grupo 6 INGENIAS Scrum
Req01	Ao entrar no jogo pela primeira vez, usuário é submetido a questões para definir seu nível e as demais funcionalidades estão indisponíveis.	2	2	2	2	2	2
Req02	A qualquer momento o usuário pode verificar módulos disponíveis para acesso.	2	2	2	2	2	2
Req03	Usuário visualiza apenas módulos disponíveis conforme sua evolução.	1	2	2	2	1	2
Req04	Ao selecionar o nível disponível na página principal, usuário é direcionado para a primeira questão aleatória do Nível em questão.	2	2	2	2	2	2
Req05	Os níveis mais altos têm questões com maior a complexidade. As questões disponibilizadas têm níveis de conhecimento de 1 a 5, sendo as questões de nível 1 são mais fáceis do que as questões de nível 5.	2	1	1	1	1	2
Req06	<i>Feedback</i> ao usuário do resultado ao final do módulo (positivo ou negativo).	2	2	2	2	2	2
Req07	As questões do nivelamento, níveis e reforço são mescladas nos formatos: áudio, vídeo, imagens e texto.	0	0	0	0	2	2
Req08	Cada questão do nivelamento, de nível e reforço é composta por 3 opções de respostas (A, B e C), onde haverá uma opção correta e duas incorretas.	2	2	2	2	2	2
Req09	Ao selecionar uma opção incorreta em uma questão, usuário é informado da opção correta.	2	2	2	2	2	2
Req10	Ao selecionar uma opção incorreta em uma questão, usuário recebe um retorno com uma justificativa relatando o porquê da opção escolhida estar incorreta. Este retorno pode ser em formato de áudio, vídeo, imagens e texto.	0	0	0	0	0	0
Req11	Usuário para cada pergunta pode rever a questão, resposta e retorno. Entretanto, usuário não consegue alterar a opção selecionada.	2	2	2	2	2	2

Quadro 13 – Pontuação dos requisitos (continuação).

Código	Requisito	Grupo 1 INGENIAS	Grupo 2 INGENIAS	Grupo 3 AgilePASSI	Grupo 4 AgilePASSI	Grupo 5 INGENIAS Scrum	Grupo 6 INGENIAS Scrum
Req12	O nivelamento, corresponde a 10 questões referente aos cinco níveis disponíveis. Para cada nível, existem duas questões distribuídas aleatoriamente.	2	0	0	0	0	2
Req13	Durante todo o nivelamento, níveis e reforço, usuário acompanha a sua evolução com uma barra de status. Para acertos, usuário vê um indicativo verde na barra de status. Enquanto para erros, usuário vê um indicativo vermelho.	2	2	2	2	2	2
Req14	Ao final do nivelamento, respondendo todas as questões, usuário é informado o nível alcançado: Nível 5, (acertar todas as questões do nível 1 ao 4), Nível 4 (acertar todas as questões do nível 1 ao 3), Nível 3 (acertar todas as questões do nível 1 e 2), Nível 2 (acertar todas as questões do nível 1) e Nível 1 (errar uma das duas perguntas do Nível 1).	2	2	2	2	2	2
Req15	Usuário pode interromper o nivelamento e deve iniciar o processo novamente e as respostas anteriores são desconsideradas.	1	2	1	2	1	2
Req16	Usuário deve realizar o Nível 5, mesmo que acerte todas as questões do nivelamento.	2	2	2	2	2	2
Req17	Após finalizar o nivelamento, usuário é direcionado para a página principal do jogo e poderá iniciar o nível alcançado.	2	2	2	2	2	2
Req18	Ao entrar no jogo após realização do nivelamento, usuário permanecerá no último nível alcançado.	2	2	2	2	2	2
Req19	Durante os níveis 1, 2, 3, 4 e 5, usuário responde 10 questões referente ao nível de conhecimento 1, 2, 3, 4 e 5, respectivamente.	2	0	0	0	0	2
Req20	Usuário deve poder conseguir visualizar uma questão, resposta e retorno. Entretanto, usuário não poderá alterar a opção selecionada.	2	2	2	2	2	2

Quadro 13 – Pontuação dos requisitos (conclusão).

Código	Requisito	Grupo 1 INGENIAS	Grupo 2 INGENIAS	Grupo 3 AgilePASSI	Grupo 4 AgilePASSI	Grupo 5 INGENIAS Scrum	Grupo 6 INGENIAS Scrum
Req21	Ao final de cada nível, usuário que obter 70% de acertos ou mais, está apto a avançar para o próximo nível.	2	2	2	2	2	2
Req22	Usuário obtém ao final de cada Nível um Equipamento para o seu consultório.	2	2	2	2	2	2
Total		38	35	34	35	35	42
Total de requisitos entregues		18	17	16	17	16	21
Total de requisitos parcialmente entregues		2	1	2	1	3	0
Total de requisitos não entregues		2	4	4	4	3	1

Fonte: A autora, 2018.

Além da avaliação por requisitos, também foi realizada uma análise qualitativa por duas especialistas em jogos médicos orientados a agentes e pelo cliente, representado pela aluna de mestrado em ciências computacionais, autora desta dissertação, simulado no estudo de caso gerando o quadro 14. Na apresentação final dos grupos foram observados:

- Grupo 1: A solução implementada foi considerada boa pelos especialistas. Um fator positivo foi que a identificação dos agentes e características do sistema aconteceu de maneira mais fácil com o auxílio dos metamodelos INGENIAS. O grupo identificou três agentes: jogador, examinador e randomizador.
- Grupo 2: A solução implementada foi considerada boa pelos especialistas. Um fator positivo foi que a identificação dos agentes e características do sistema aconteceu de maneira mais fácil com o auxílio dos metamodelos INGENIAS. O grupo identificou cinco agentes: gerenciador do jogo, gerenciador de questões, gerenciador de interface, gerenciador da pontuação e gerenciador de eventos. O protótipo deste grupo foi considerado inovador pela utilização dos conceitos de *Business Intelligence* na sua criação.
- Grupo 3: A solução implementada foi considerada muito boa pelos especialistas. Com a utilização do método AgilePASSI foi possível encontrar nove agentes, sendo eles: nivelador, randomizador, armazenador, analisador, guia, avaliador, recuperador e inicializador. O grupo apresentou o fluxo do sistema e a cada nível do jogo apresentava o novo equipamento ganho. Além disso, o consultório foi apresentado em todas as etapas do jogo.
- Grupo 4: A solução implementada foi considerada muito boa pelos especialistas. Com a utilização do método AgilePASSI foi possível encontrar sete agentes, sendo eles: nivelador, randomizador, gerenciador de persistência, renderizador, justificador, gerenciador de progresso e usuário. O grupo utilizou o ambiente *Ionic Framework*.
- Grupo 5: A solução implementada foi considerada boa pelos especialistas. Um fator positivo foi que a identificação dos agentes e características do sistema aconteceu de maneira mais fácil com o auxílio dos metamodelos INGENIAS. O grupo identificou três agentes: jogador, controlador e administrador, o método utilizado foi o INGENIAS Scrum e a ferramenta

Trello para gerenciamento de projetos foi utilizada a fim de facilitar a interação entre os desenvolvedores e o cliente.

- Grupo 6: A solução implementada foi considerada muito boa pelos especialistas. Um fator positivo foi que a identificação dos agentes e características do sistema aconteceu de maneira mais fácil com o auxílio dos metamodelos INGENIAS. O grupo identificou cinco agentes: *master*, usuário, gerenciador de banco de dados, gerenciador de telas e gerenciador de conhecimento. O método utilizado foi o INGENIAS Scrum e a ferramenta Trello para gerenciamento de projetos foi utilizada a fim de facilitar a interação entre os desenvolvedores e o cliente. Este grupo utilizou perguntas específicas da área da saúde em seu banco de questões. O consultório final foi apresentado em imagem 3D e, além disso, apresentou o sistema por inteiro em vídeo.

As notas do protótipo final do jogo foram organizadas no quadro 14, estando as notas dentro de uma escala de zero a quatro (0 a 4), onde 0 representa a menor nota e 4 a maior nota. Os critérios utilizados para a nota do especialista e do cliente foram: atraso das entregas dos artefatos, layout do protótipo e conformidade dos requisitos.

Quadro 14 – Notas do protótipo final do jogo implementado

	Especialista	Cliente	Média %
Grupo 1	3,5	3,5	87,5%
Grupo 2	3,5	3,5	87,5%
Grupo 3	4,0	3,5	93,7%
Grupo 4	4,0	3,5	93,7%
Grupo 5	3,5	3,0	81,2%
Grupo 6	4,0	4,0	100%

Fonte: A autora, 2018.

4.7 Análise e interpretação dos dados estatísticos

Para realizar a análise e interpretação dos dados estatísticos, foi utilizada a análise de variância (APÊNDICE B – ANOVA), a fim de obter dados mais significativos para esta aplicação de comparação de métodos orientados a agentes para jogos educacionais. A análise de variância foi necessária pelo fato de possuir somente dois dados por método.

Foram propostos fatores que levavam em consideração as variáveis quantitativas e qualitativas, enquanto que a variável dependente foi uma variável quantitativa.

Com a análise de variância, o objetivo foi analisar as diferenças entre as médias aritméticas dos grupos, a partir de uma análise na variação dos dados, entre os grupos. Em termos de análise estatística, foi considerada a variação total e subdividida na variação entre os grupos e na variação dentro dos grupos.

4.7.1 Análise estatística dos resultados

A análise estatística dos dados coletados neste estudo de caso foi realizada usando a análise de variância (ANOVA) a fim de determinar se as diferenças entre os métodos utilizados pelos grupos são estatisticamente significativas. Os testes estão dispostos nos APÊNDICES C e D.

O teste foi aplicado para a hipótese deste estudo de caso, dois a dois, e objetivou-se analisar se o desenvolvimento de jogos médicos educacionais orientados a agentes possui entrega mais rápida em relação aos artefatos em geral, quando são desenvolvidos por métodos ágeis orientados a agentes (AgilePASSI e INGENIAS Scrum), ou não. A hipótese foi rejeitada com nível de significância de 5% ($\alpha < 5\%$). Ou seja, confirma que jogos médicos educacionais orientados a agentes construídos utilizando uma abordagem de desenvolvimento ágil tem entrega mais rápida, quando comparado a jogos desenvolvidos usando uma abordagem de desenvolvimento tradicional orientado a agentes.

Em relação à hipótese de que os desenvolvimentos de jogos médicos educacionais orientados a agentes atendem melhor aos requisitos do projeto quando são desenvolvidos por métodos ágeis orientados a agentes (AgilePASSI e INGENIAS Scrum), ou não. Comparando-se o grupo 1 com os demais que utilizaram métodos ágeis e, o grupo 2 com os mesmos demais

grupos, a hipótese não foi rejeitada com nível de significância de 5% ($\alpha < 5\%$). Ou seja, confirma que jogos médicos educacionais orientados a agentes construídos utilizando uma abordagem de desenvolvimento ágil não atendem melhor aos requisitos do projeto quando comparado a jogos desenvolvidos através de uma abordagem de desenvolvimento tradicional orientado a agentes.

Os cálculos da ANOVA em relação aos dias de atraso são apresentados nos quadros 15 e 16.

Quadro 15 – Resumo dos grupos para cálculo da ANOVA.

Grupos	Quantidade de artefatos entregues	Média dos dias de atraso	Variância dos dias de atraso
INGENIAS	8	0,375	1,125
INGENIAS	8	29,75	420,5
AgilePASSI	9	3,778	25,194
AgilePASSI	9	4,556	34,528
INGENIAS SCRUM	9	12,333	152,75
INGENIAS SCRUM	9	0	0

Fonte: A autora, 2018.

Quadro 16 – Cálculo da ANOVA em relação aos dias de atraso.

Fontes de Variação	Soma dos quadrados	Grau de liberdade	Quadrado médio	Fcalc.
Entre grupos	1413,48	1	1413,48	8,22
Dentro dos grupos	1900,63	15	1013,67	
Total	3314,11	16	—	—

Fonte: A autora, 2018.

O valor crítico de F, ao nível de significância de 5%, é 4,54. Comparando o $F_{tab} = 4,54$ e o $F_{calc.} = 8,22$, e sendo $F_{calc.} > F_{tab}$, rejeita-se ao nível de 5%, a hipótese H_0 .

Os cálculos da ANOVA em relação à análise dos requisitos para os seis grupos são apresentados nos quadros 17 e 18.

Quadro 17 – Cálculo da ANOVA em relação à análise dos requisitos.

Grupos	Quantidade de requisitos/pontuação máxima	Soma da pontuação dos requisitos	Percentual de atendimentos dos requisitos	Média da pontuação dos requisitos	Variância da pontuação dos requisitos
INGENIAS	22/44	38	86,4%	1,727	0,398
INGENIAS	22/44	38	86,4%	1,727	0,398
AgilePASSI	22/44	34	77,3%	1,545	0,641
AgilePASSI	22/44	35	79,5%	1,591	0,634
INGENIAS SCRUM	22/44	35	79,5%	1,591	0,539
INGENIAS SCRUM	22/44	42	95,5%	1,909	0,182

Fonte: A autora, 2018.

Quadro 18 – Resumo dos grupos para cálculo da ANOVA.

Fontes de Variação	Soma dos quadrados	Grau de liberdade	Quadrado médio	Fcalc.
Entre grupos	0,284	1	0,284	0,7
Dentro dos grupos	18,84	42	0,449	
Total	19,124	43	—	—

Fonte: A autora, 2018.

O valor crítico de F, ao nível de significância de 5%, é 4,08. Comparando $F_{tab} = 4,08$ e o $F_{calc.} = 0,7$, e sendo $F_{calc.} > F_{tab}$, não se rejeita ao nível de 5% a hipótese H_0 .

Usando a análise de variância ANOVA foi possível verificar que o desenvolvimento de jogos médicos educacionais orientados a agentes possuem entrega mais rápida quando são desenvolvidos por métodos ágeis orientados a agentes (AgilePASSI e INGENIAS Scrum), e também que jogos médicos educacionais orientados a agentes construídos utilizando uma abordagem de desenvolvimento ágil (INGENIAS Scrum e AgilePASSI) não atendem melhor aos requisitos do projeto, quando comparado a jogos desenvolvidos usando uma abordagem de desenvolvimento tradicional orientado a agentes (INGENIAS).

4.7.2 Análise através da avaliação da qualidade

A análise do estudo de caso foi realizada com a ANOVA, como visto anteriormente. Além desta análise, também foi verificada a qualidade dos métodos utilizados pelos 6 grupos usando um questionário de avaliação de qualidade de modelos de sistemas multiagentes

(APÊNDICE E – Questionário de avaliação da qualidade), desenvolvido como proposta da dissertação de mestrado concluída pela ex-aluna do programa de mestrado em ciências computacionais da Universidade do Estado do Rio de Janeiro (MONTEIRO, 2018). O questionário possui como principal contribuição o apoio ao desenvolvedor para verificação de qualidade de seu modelo. Ele pode ser aplicado para modelos de sistemas multiagentes baseados em qualquer um dos métodos existentes. O objetivo, neste caso, é aplicar o questionário na modelagem já realizada baseada no estudo de caso MEDEDUC. Esta análise está relacionada a percepção dos usuários em relação as modelagens.

Os questionários foram enviados a todos os participantes do estudo de caso no dia 16/05/2018 e foram respondidos seis questionários em um total de 19. Entre estes questionários recebidos, um corresponde ao método INGENIAS do grupo 1, dois ao método AgilePASSI do grupo 4 e três ao método INGENIAS Scrum, todos do grupo 6. Apesar de termos poucas respostas, algumas análises puderam ser realizadas, conforme os quadros 19 e 20.

Quadro 19 – Avaliação da satisfação das métricas

Métricas	INGENIAS	AgilePASSI	INGENIAS Scrum
Acurácia	Atende	Atende	Atende
Adequação	Atende	Atende	Atende
Cobertura de Contexto	Atende	Atende	Atende
Completeza	Atende	Atende	Atende
Concisão	Atende	Supera	Atende
Conformidade	Atende	Atende	Atende
Corretude	Atende	Atende	Atende
Disponibilidade	Atende parcialmente	Atende com muitas restrições	Atende
Eficácia	Atende	Atende parcialmente	Atende
Eficiência	Atende parcialmente	Atende	Atende
Integridade	Atende	Atende	Atende
Maturidade	Atende	Atende parcialmente	Atende parcialmente
Modificabilidade	Atende parcialmente	Atende parcialmente	Atende parcialmente
Segurança	Atende parcialmente	Atende	Atende parcialmente

Fonte: A autora, 2018.

Quadro 204 – Avaliação da satisfação das dimensões.

Dimensões	INGENIAS	AgilePASSI	INGENIAS Scrum
Qualidade do Modelo	Atende	Atende	Atende
Qualidade do Desempenho	Atende parcialmente	Atende	Atende
Qualidade dos Recursos	Atende	Atende parcialmente	Atende
Confiabilidade	Atende	Atende parcialmente	Atende
Qualidade da Modelagem	Atende parcialmente	Atende	Atende
Durabilidade	Atende	Atende parcialmente	Atende parcialmente
Facilidade de Manutenção	Atende parcialmente	Atende parcialmente	Atende
Qualidade da Avaliação	Atende	Atende	Atende
Qualidade de Agentes	Atende	Atende	Atende

Fonte: A autora, 2018.

Através da avaliação da satisfação em relação às métricas (Quadro 20), é possível avaliar alguns fatores importantes aos sistemas multiagentes como: segurança, acurácia e conformidade, por exemplo. As especificações da dimensão e métricas estão dispostas no APÊNDICE F – Descrição das dimensões e métricas da avaliação da qualidade. Algumas destas métricas se adequam melhor a alguns métodos, como por exemplo, o método INGENIAS foi avaliado em “atende parcialmente” em relação à eficiência, enquanto os métodos ágeis atendem bem a esta métrica. Já para a métrica maturidade os métodos ágeis INGENIAS Scrum e AgilePASSI foram avaliados em “atende parcialmente” e o método tradicional INGENIAS foi avaliado como “atende”.

Por meio da avaliação da satisfação em relação às dimensões, constatou-se que todos os métodos atendem às dimensões em algum sentido. Algumas destas dimensões se adequam melhor a alguns métodos, como por exemplo o método INGENIAS foi avaliado em “atende parcialmente” em relação à qualidade do desempenho, enquanto os métodos ágeis atendem bem a esta dimensão. Para a dimensão durabilidade os métodos ágeis INGENIAS Scrum e AgilePASSI foram avaliados em “atende parcialmente” e o método tradicional INGENIAS foi avaliado como “atende”.

4.8 Análise qualitativa dos resultados

Em resumo, observou-se através dos dados estatísticos que, apesar de os métodos ágeis para desenvolvimento de sistemas multiagentes terem um desenvolvimento mais dinâmico do que os métodos tradicionais, este fato não influenciou na qualidade dos artefatos.

Avaliando separadamente cada grupo, notou-se que os grupos que utilizaram o método INGENIAS tinham artefatos fixos definidos pelos metamodelos. Os encontros com os participantes destes grupos foram feitos semanalmente e durante estes encontros, a comunicação entre os envolvidos e cliente acontecia, dúvidas referentes aos requisitos eram sanadas e validações eram realizadas.

Em relação aos grupos que desenvolveram o estudo de caso utilizando o método AgilePASSI, notou-se que a dinâmica para construção dos artefatos foi realizada de forma mais simples. Foram desenvolvidos diagramas de casos de uso e de agentes, além de lista de requisitos funcionais e não funcionais. Os encontros com os participantes destes grupos também foram feitos semanalmente e durante estes encontros, a comunicação entre os envolvidos e cliente acontecia, as dúvidas referentes aos requisitos eram sanadas e validações eram realizadas.

O método INGENIAS Scrum foi implementado por dois grupos e foi possível perceber que a comunicação era realizada de forma eficiente entre os participantes, *productowner* e *Scrum master*. Apesar do método requerer os tradicionais metamodelos INGENIAS como artefatos, ao utilizar-se a dinâmica de gerenciamento de projetos do Scrum a entrega dos artefatos foi realizada de forma mais produtiva para os envolvidos, devido a frequente comunicação entre o *Scrum team* e o *product owner*. Os encontros com os participantes destes grupos também foram realizados semanalmente e durante estes encontros, a comunicação entre os envolvidos e *productowner* acontecia, dúvidas referentes aos requisitos eram sanadas e validações eram realizadas.

Ao longo do estudo de caso, vinte e seis artefatos foram produzidos no total nos três métodos. Os métodos AgilePASSI e INGENIAS Scrum exigiram um total de nove artefatos, sendo que o AgilePASSI tinha 4 relatórios gerenciais (estórias de usuários), enquanto o INGENIAS Scrum 6 artefatos do método INGENIAS, e 4 relatórios gerenciais. Os artefatos comuns nos três métodos foram o código e o jogo.

Ao final do desenvolvimento, identificou-se que dentre os 22 requisitos propostos para o jogo médico educacional, todos os grupos foram capazes de entregar mais de 79% dos

requisitos solicitados. Alguns dos requisitos foram entregues parcialmente e uma parte destes requisitos não foi iniciada por todos os grupos.

Houve algumas diferenças entre os grupos que utilizaram o mesmo método. Como, por exemplo, o grupo 1, que aplicou o método INGENIAS, teve melhor desempenho em relação ao tempo de entrega dos artefatos. Já o grupo 2, que também utilizou o método INGENIAS, obteve mais facilidade na criação dos metamodelos e execução do jogo, implementando funções que não foram estipuladas como requisitos do sistema. Com relação aos grupos que utilizaram o método AgilePASSI, observou-se que ambos os grupos identificaram mais agentes comparados aos outros métodos. Enquanto os outros métodos identificaram 3 a 5 agentes, os grupos do AgilePASSI identificaram 7 e 9 agentes para o estudo de caso. Os grupos que utilizaram o método INGENIAS Scrum foram divergentes em relação à rapidez da entrega dos artefatos. O grupo 5 enfrentou dificuldade em apresentar os artefatos nos prazos estipulados. Em contrapartida, o grupo 6 não atrasou nenhuma entrega e foi o que entregou mais requisitos completos e obteve melhor pontuação.

A análise estatística através da ANOVA constatou que em relação aos requisitos que os grupos que utilizaram os métodos ágeis para a execução do estudo de caso para o desenvolvimento de jogos médicos educacionais orientados a agentes possuem entrega mais rápida quando são desenvolvidos por métodos ágeis orientados a agentes (AgilePASSI e INGENIAS Scrum).

Os grupos AgilePASSI e INGENIAS Scrum realizavam entregas periódicas dos jogos médicos educacionais. Desta forma, semanalmente, era possível acompanhar os resultados dos grupos que utilizaram INGENIAS Scrum e AgilePASSI. Nos grupos do método INGENIAS, os jogos só foram apresentados ao término do estudo de caso, isto é, após 2 meses de concepção, construção e elaboração. Ressalta-se que, em relação aos encontros entre cliente e o *product owner*, o grupo 1 obteve semanalmente, já o grupo 2 optou por desenvolver sem interação com o *product owner*, ou seja, não compareceu aos encontros.

Os jogos desenvolvidos tiveram características, soluções, *design* e tecnologias distintas em suas implementações. Durante o desenvolvimento, os grupos não utilizaram um *framework* de desenvolvimento próprio para sistemas multiagentes. Os artefatos do jogo, como suas telas e modelagens, podem ser vistos nos ANEXOS B, C, D, E, F e G.

O quadro 21 apresenta uma síntese das características analisadas qualitativamente desse estudo de caso.

Quadro 21 – Comparativo dos métodos do estudo de caso.

	INGENIAS	AgilePASSI	Ingenias Scrum
Comunicação cliente e envolvidos	Um encontro por semana/uma vez no início e na apresentação	Um encontro por semana	Um encontro por semana
Quantidade de modelos desenvolvidos	6	2	6
Quantidade de artefatos entregues	8	9	9
Artefatos entregues	Diagrama de casos de uso, diagrama de organização, diagrama de agentes, diagrama de metas e tarefas, diagrama de ambiente, diagrama de interação e modelo de códigos.	Estórias de usuários, diagramas de caso de uso UML, diagramas de classes de agentes, planos de testes e código.	Diagrama de casos de uso, diagrama de organização, diagrama de agentes, diagrama de metas e tarefas, diagrama de interação, product <i>backlog</i> , estórias de usuários e código.
Quantidade de requisitos entregues ao final do estudo de caso	18/17 requisitos entregues.	16/17 requisitos entregues; 2/1 requisitos parcialmente entregues.	16/21 requisitos entregues; 3/0 requisitos parcialmente entregues.
Quantidade de requisitos não entregues	2/4 requisitos não entregues.	4/4 requisitos não entregues.	3/1 requisitos não entregues.
Quantidade de agentes identificados	3/3 respectivamente.	9/7 respectivamente.	3/4 respectivamente.
Envolvimento do cliente	Comprometido durante todo tempo.	Comprometido durante todo tempo.	Comprometido durante todo tempo.
Tempo de entrega da implementação	2 meses.	2 em 2 semanas.	1 vez por semana.
Teste	Ao término do desenvolvimento.	A cada reunião semanal.	A cada reunião semanal.

Fonte: A autora, 2018.

4.9 Comparação das propostas de desenvolvimento do estudo de caso

O método INGENIAS Scrum foi o que apresentou diferenças estatisticamente significativas entre seus grupos, sendo que um deles foi a que teve mais conformidade nos requisitos e nenhum de atraso nas entregas dos artefatos. Em contrapartida no outro grupo que utilizou o mesmo método, atrasou as entregas dos artefatos e teve uma menor conformidade nos requisitos (79%). Assim, a aplicação deste método depende de fatores do grupo que pode ser devido ao interesse, conhecimento, habilidade, entre outros fatores. Os outros métodos não se diferenciaram tanto em relação aos requisitos, ou seja, estatisticamente a porcentagem de requisitos entregues foi similar. Em relação à rapidez na entrega dos artefatos 1 grupo que utilizou o método INGENIAS teve um grande atraso que pôde ser ocasionado pelo método não ter explicito a necessidade de encontros com cliente. E assim os mesmos fatores citados anteriormente, podem ter influenciado: interesse no estudo de caso, conhecimento e habilidade.

Uma diferença importante observada durante o estudo de caso entre os métodos ágeis (INGENIAS Scrum e AgilePASSI) e tradicional (INGENIAS) é que nos métodos ágeis, os grupos possuíam a capacidade de entregar resultados mais rápido. Os dois métodos ágeis

enfatazavam os grupos, o jogo, a colaboração do cliente, e resposta à mudança; enquanto no método tradicional, o foco estava no plano, processo, documentação e ferramenta.

Em relação ao método tradicional, os grupos realizavam reuniões semanais com o cliente a fim de obter todos os requisitos detalhados durante a fase inicial do estudo de caso e sanar as dúvidas ocasionais. Em seguida, os grupos iniciaram a fase de concepção, seguida por uma fase de codificação. A fase de teste só começou quando todo o processo de codificação foi concluído. Então, o produto final só foi apresentado após o término de todo o ciclo de desenvolvimento e teste.

O método ágil INGENIAS Scrum se mostrou capaz de entregar um produto final que atendia às necessidades do solicitante, independente do grupo, pois em um curto prazo das iterações, os módulos eram concluídos e apresentados ao solicitante para revisão. Estes módulos não eram de forma integrada como um produto completo e, portanto, qualquer retrabalho ou requisito adicional não aumentava significativamente o tempo de desenvolvimento, além da equipe estar sempre pronta para incluir todos os requisitos desejados pelo solicitante.

Embora os métodos ágeis tenham demonstrado vantagens, na maior parte, em relação ao tempo de atraso sobre o método tradicional INGENIAS, em alguns momentos existia a dificuldade em colocá-los em prática. Uma delas foi que o método ágil AgilePASSI obtinha menos modelos que auxiliavam na percepção do estudo de caso. Métodos ágeis são conhecidos por enfatizar a comunicação e o envolvimento do cliente. Entretanto, para problemas complexos, o uso de modelos facilita a comunicação da equipe e o *design* da solução. Para cada iteração finalizada, o grupo e o solicitante realizavam uma reunião, onde os membros do grupo resumiam a evolução do trabalho desenvolvido. No decorrer do estudo de caso, o grupo e o cliente se encontravam em reuniões semanais, estas eram feitas de forma proposital para entender como a equipe iria se comportar com as mudanças de requisitos.

A vantagem do método tradicional INGENIAS se deu por ter um modelo de exigência bem definido antes da execução e do processo de codificação começar, e este modelo funcionava como uma referência para os grupos durante a codificação. Os grupos faziam toda a modelagem e codificação de acordo com a documentação disponibilizada pelo solicitante até o jogo ser concluído e pudesse ser apresentado.

O método AgilePASSI pode ser considerado intermediário entre o INGENIAS e INGENIAS Scrum, tendo as vantagens da orientação de agentes e do desenvolvimento ágil. Uma desvantagem detectada foi a demora na identificação dos requisitos fundamentais para o desenvolvimento do jogo, postergando as entregas dos protótipos.

No método INGENIAS Scrum, a comunicação entre o cliente e Scrum *team* está a todo tempo disponível e possui a vantagem de contar com os metamodelos INGENIAS, a fim de facilitar o entendimento dos agentes, metas e organização, ambiente e interação. Outra vantagem foi que o produto final foi se adequando aos requisitos do cliente e puderam ser identificados novos requisitos.

4.10 Ameaças à validade

Algumas ameaças à validade foram detectadas, apesar de cuidados que foram tomados no estudo de caso em questão, segundo a taxonomia de Wholinet e colaboradores (2000):

Validade Interna: (i) as equipes não continham a mesma quantidade de participantes.

Validade Interna: (ii) alguns grupos demonstraram mais interesse no desenvolvimento do estudo de caso.

Validade externa: (i) os produtos não foram apresentados a um cliente da área da saúde e sim à especialistas em desenvolvimento de produtos de *software* para a área da saúde.

Algumas estratégias foram utilizadas a fim de minimizar as ameaças detectadas no estudo de caso, como:

- a. Foi estabelecido um plano de desenvolvimento de artefatos e, notou-se que entregar no início do desenvolvimento auxilia no processo de desenvolvimento;
- b. Foram desenvolvidos relatórios gerenciais que ajudaram no acompanhamento do projeto e das versões entregues;
- c. Foram mantidas reuniões periódicas para se discutir o produto, andamento do projeto e testes.

4.11 Pesquisa de satisfação do estudo de caso

Ao final do estudo de caso, os participantes envolvidos responderam um questionário de satisfação a fim de averiguar se as expectativas haviam sido superadas, atendidas ou não atendidas. Em uma escala entre opções como: (5) extremamente satisfeito, (4) moderadamente

satisfeito, (3) satisfeito, (2) nem satisfeito e nem insatisfeito, (1) moderadamente insatisfeito e (0) extremamente insatisfeito.

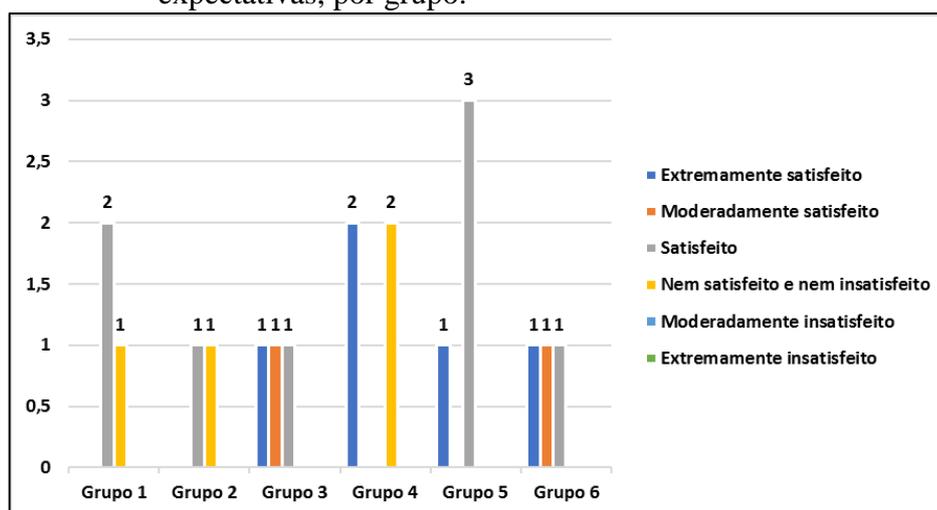
Os participantes votaram em relação ao aprendizado do método e aos resultados obtidos com o uso do método proposta. Sobre às suas expectativas ao início do estudo a escala utilizada foi: (4) superadas, (3) atingidas; (2) indiferente; (1) não foram atingidas; (0) não foram superadas.

E em relação à área de sistemas multiagentes a probabilidade de o aluno participar de outros estudos, utilizou-se as opções de resposta: (4) extremamente provável, (3) muito provável, (2) provável, (1) pouco provável, (0) nada provável.

Além disso, deixou-se um campo aberto para que os participantes pudessem responder sobre o que mais gostaram da experiência com métodos para o desenvolvimento de sistemas multiagentes.

Em relação ao aprendizado do método dos envolvidos, o gráfico 5 apresenta quantidade absoluta de participantes por grupo e o total de todos os grupos foi, 6 participantes, ou seja, 32% ficaram extremamente satisfeitos, 2 participantes (10%) optaram por moderadamente satisfeitos, 7 participantes (37%) se consideraram satisfeitos e 4 participantes (21%) ficaram nem satisfeitos e nem insatisfeitos.

Gráfico 5 – Quantidade absoluta dos participantes em relação às expectativas, por grupo.

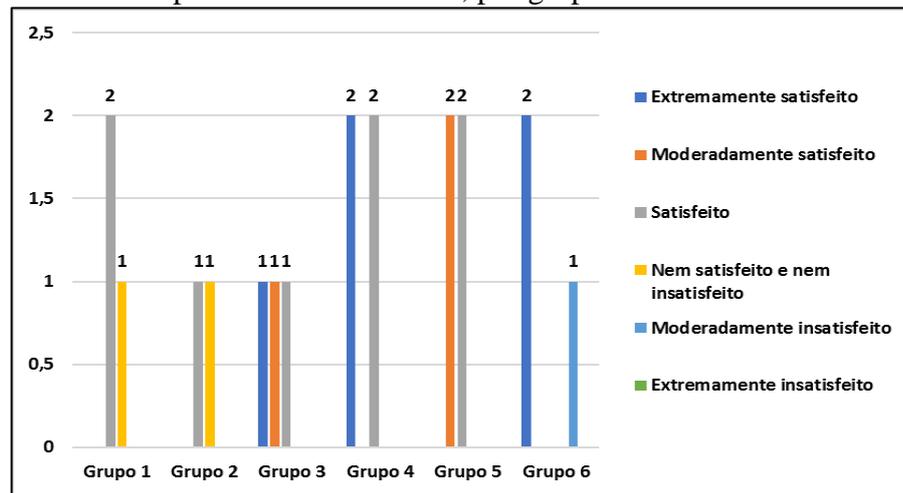


Fonte: A autora, 2018.

Em relação aos resultados obtidos com o uso do método proposto, o gráfico 6 apresenta quantidade absoluta por grupo e descrevem-se de forma geral para todos os grupos, 5 participantes, ou seja, 26% ficaram extremamente satisfeitos, 3 participantes (16%) optaram por moderadamente satisfeitos, 8 participantes (42%) se consideraram satisfeitos, 2

participantes (11%) ficaram nem satisfeitos e nem insatisfeitos e, apenas 1 participante (5%) se considerou moderadamente insatisfeito.

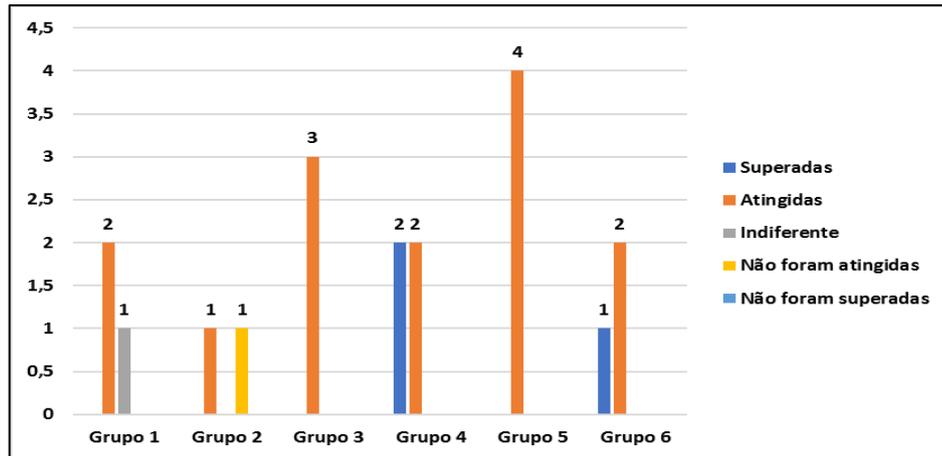
Gráfico 6 – Quantidade absoluta dos participantes em relação ao aprendizado do método, por grupo



Fonte: A autora, 2018.

Em relação às expectativas em relação ao início do trabalho, o gráfico 7 apresenta quantidade absoluta por grupo e para o total de todos os grupos e observou-se: 3 participantes, ou seja, 16% consideraram superadas, 14 participantes (74%) optaram por atingidas, 1 participante (5%) se considerou indiferente e 1 participante (5%) relatou que não foram atingidas.

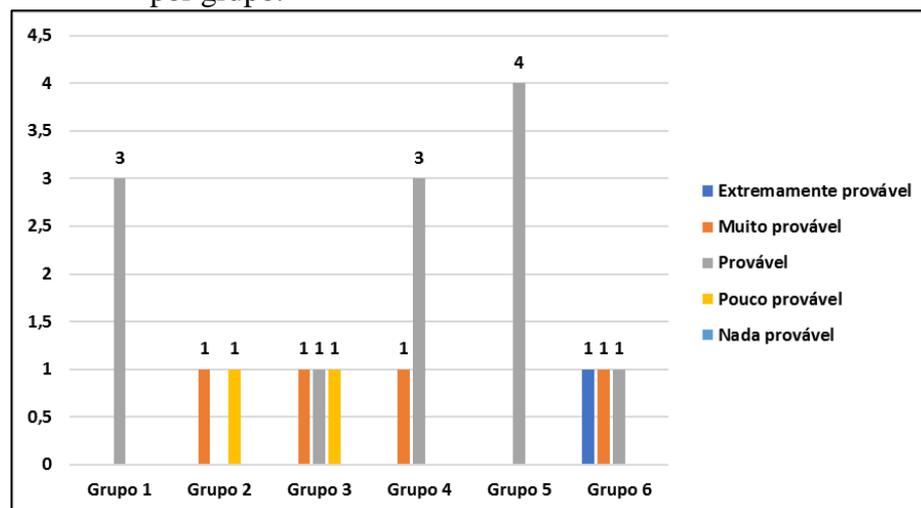
Gráfico 7 – Quantidade absoluta dos participantes em relação ao método proposto, por grupo.



Fonte: A autora, 2018.

Em relação à área de sistemas multiagentes, a probabilidade de o participante voltar a participar de outros estudos ou trabalhos é descrita no gráfico 8 que apresenta a quantidade absoluta por grupo, já em relação ao total: 1 participante, ou seja, 5% considerou extremamente provável, 4 participantes (22%) optaram por muito provável, 12 participantes (67%) escolheram a opção provável, 1 participante (6%) optou por pouco provável e 1 participante (5%) respondeu nada provável.

Gráfico 8 – Quantidade dos participantes em relação a probabilidade do participante voltar a trabalhar com sistemas multiagentes, por grupo.



Fonte: A autora, 2018.

As opiniões sobre o que mais gostaram da experiência com métodos para o desenvolvimento de sistemas multiagentes e jogos médicos educacionais são apresentadas no ANEXO J – Comentários sobre a experiência dos envolvidos no estudo de caso.

CONCLUSÕES E FUTUROS TRABALHOS

Essa dissertação teve como objetivo a comparação da aplicação de três métodos orientados a agentes (INGENIAS, AgilePASSI e INGENIAS Scrum) com diferentes características (ágeis e tradicionais) no desenvolvimento de um jogo médico educacional voltado para estudantes de medicina. Nos capítulos iniciais foi introduzido os conceitos de sistemas multiagentes, e os métodos escolhidos para o desenvolvimento do estudo de caso. Observou-se que os desenvolvimentos de jogos orientados a agentes têm resultados satisfatórios quando direcionados por um método específico para sistemas multiagentes e que tenha uma vertente ágil.

Todos os métodos utilizados tiveram vantagens e desvantagens durante a elaboração do estudo de caso. Algumas medidas foram tomadas a fim de reduzir riscos na aplicação do estudo de caso, como por exemplo: a elaboração de um cronograma detalhado sobre quais artefatos cada grupo deveria entregar durante os encontros e manter encontros semanais com os participantes dos grupos independente de qual método foi utilizado.

Durante a aplicação do método INGENIAS, os grupos relataram como vantagem de o método ter metamodelos pré-estabelecidos que auxiliam na identificação dos requisitos e no desenvolvimento do estudo de caso. Entretanto, diferente dos métodos ágeis, a comunicação com o cliente se torna mais distante pelo fato de que não houve a evolução do jogo ao decorrer das semanas, e sim, somente ao final do desenvolvimento dos metamodelos e do estudo de caso. Neste método é importante que o cliente esteja totalmente envolvido no processo de concepção, elaboração e construção do jogo médico educacional.

Ao se tratar dos métodos com processos ágeis, a maior vantagem foi a comunicação frequente com o cliente sobre o desenvolvimento do jogo. Semanalmente, era possível compreender as possíveis limitações dos grupos e sanar dúvidas a respeito do jogo. Porém, o maior desafio para os envolvidos se deu na identificação dos agentes necessários para o jogo, principalmente para o método AgilePASSI em que ambos os 2 grupos identificaram um número considerável de agentes a mais do que os outros dois métodos. Para o método INGENIAS Scrum, o desenvolvimento foi mais claro para os grupos, pois contava com os metamodelos INGENIAS para auxiliar na visualização de metas, agentes, organização, ambiente e interação e era possível ir modelando o jogo simultaneamente. Para estes métodos ágeis é importante a identificação dos agentes e a comunicação clara e frequente com o cliente.

Através da análise estatística dos dados notou-se que os métodos com processos ágeis foram mais rápidos, entretanto isso não comprometeu a qualidade dos artefatos apresentados. Observou-se também que o protótipo gerado por esses métodos foram os melhores avaliados, tanto pelo cliente, quanto pelos especialistas. Entretanto esses resultados não podem ser generalizados por ser um estudo de caso limitado, sendo avaliado por professores e um cliente que não são da área médica.

A partir desta dissertação, novos trabalhos e pesquisas poderão ser desenvolvidos em assuntos relacionados à engenharia de *software*, jogos médicos educacionais orientados a agentes e, principalmente, sobre métodos de desenvolvimento tradicional e ágil para sistemas multiagentes.

Outro trabalho futuro é implementar o jogo MEDEDUC adicionando inteligência aos agentes e sua avaliação pode ser ampliada para professores e alunos de medicina.

Da mesma forma que o estudo de caso realizado por Ferreira (2016) encorajou o desenvolvimento deste estudo de caso, os resultados deste estudo de caso podem encorajar o surgimento de novos esforços em um futuro próximo, no intuito de validar se os conceitos ágeis aplicados ao desenvolvimento de jogos médicos educacionais com processos de desenvolvimento tradicionais trarão benefícios no resultado final do jogo.

REFERÊNCIAS

- AGILLE, A. “*What is Scrum?*”. Advanced Development Methods. 2003. Disponível em: <<http://www.controlchaos.com/>>. Acesso em: 18 set. de 2017.
- BASSI FILHO, D. L.: *Experiências com desenvolvimento ágil*. 2008. 149 f. Dissertação (Mestrado em Ciências) – Instituto de Matemática e Estatística, Universidade de São Paulo, São Paulo, 2008.
- BECK, K. *Embrace Changewith Extreme Programming*. IEEE Computer Magazine [S.1]. 1999. p. 70-77.
- BECK, K. et al. *Manifesto for agile software development*, 2001. Disponível em: <https://moodle2016-17.ua.es/moodle/pluginfile.php/80324/mod_resource/content/2/agile-manifesto.pdf>. Acesso em: 20 out. de 2017.
- BELLIFEMINE, F.; POGGI, A.; RIMASSA, G. *Compliant agent development environment and mobile devices in cultural heritagrobotics*. Proceedings of the Robotics Workshop in. In: Fifth International Conference on Autonomous Agents. 2001. p. 16-217.
- BOEHM, B. *A Spiral Model of Software Development and Enhancement Computer*. 1988. p.61-72.
- BOOCH, G.; JACOBSON, I.; RUMBAUGH, J. *The Unified Software Development Process*. Addison-Wesley, Reading, 1999.
- BUSONI, L.; BABUSKA, R.; DE SCHUTTER, B. *A comprehensive survey of multiagent reinforcement learning*. In: IEEE Transactions on Systems, Man, and Cybernetics—Part C: Applications and Reviews – Part C. v. 38, n. 2, p. 156–72, 2008.
- CARRO, M. et al. *A methodology for developing adaptive educational-game environments*. In: Adaptive hypermedia and adaptive Web-based systems. 2nd international conference, AH 2002, Málaga, Spain, May 29–31, 2002. Proceedings p.90-99.
- CHELLA, A. et al. *From passi to agile passi: tailoring a design process to meet new needs*. In: IEEE/WIC/ACM International Joint Conference on Intelligent Agent Technology (IAT-04), Beijing, China. 2004.
- COSENTINO, M.; HILAIRE, V.; MOLESINI, A. *Handbook on Agent-Oriented Design Processes*. Springer, 2014. 580 p.
- COSENTINO, M.; POTTS, C. *A case tool supported methodology for the design of multi-agent systems*. In: The 2002 International Conference on Software Engineering Research and Practice. 2002, SERP’02, Las Vegas.
- EISENHARDT, K. M. *Building theories form case study research*. Academy of Management Review. New York, v. 14, n. 4, p. 532-50, 1989.

FERBER, J.; FITZGERALD, K. Multi-Agent Systems Instructional methods: selection, use, and evaluation. In: BASTABLE, S. B. (Eds). *Nurse as Educator: Principles of Teaching and Learning*. Sudbury, MA: Jones and Bartlett, 1999, p. 261–86.

FERREIRA, V. *Comparação de Desenvolvimento Orientado a Agentes para Jogos Educacionais: Um estudo de caso*. 2016. 165 f. Dissertação (Mestrado em Ciências Computacionais) – Instituto de Matemática e Estatística, Universidade do Estado do Rio de Janeiro, Rio de Janeiro. Brazil. 2016.

FRANCE, R.; RUMPE, B. *Model-driven development of complex software: a research roadmap*. Academic gaming in nurse education. v. 5, n. 6, p. 601-13, 1980.

GENESERETH, M. R. *Software agents*. Communications of the ACM, v. 37, n. 7, p. 48-53, 1994.

GIBBS, G. E. *Managerial Communication: Strategies and Applications*. SAGE Publications. 2006.

GIL, A. C. Como elaborar projetos e pesquisa. 3. ed. São Paulo: Atlas, 1995.

GÓMEZ-RODRÍGUEZ, A. et al. *INGENIAS with the Unified Development Process*. Springer, Berlin, Heidelberg. 2014.

GÓMEZ-RODRÍGUEZ, A.; GONZÁLEZ-MORENO, J. C. *Definition of Process Models for Agent-based Development*. In: 9th International Workshop on AOSE, 2008.

GÓMEZ-SANZ, J. *Ingenias Agent Framework*. Development Guide V. 1.0. Universidad Complutense de Madrid, 2008.

GONZÁLEZ-MORENO, J. C. et al. Ingenias-Scrum. In: COSSENTINO, M. et al. *Handbook on Agent-Oriented Design Processes*. Springer, 2014. p. 219-51.

HENDERSON-SELLERS, B.; GIORGINI, P. *Agent-oriented methodologies*. IGI Global, 2005.

HOFFMAN, D. L.; NOVAK, T.P. *Marketing in hyper-media computer-mediated environments: conceptual foundations*. Journal of Marketing, v. 60, n. 2, p. 50-68, 1996.

HUHNS, M. N.; STEPHENS, L. M. Multiagent systems and societies of agents. In: WEISS, G. *Multiagent systems*. Cambridge, MA: MIT Press, 2000. p. 79-120.

JALOTE, P. *An Integrated Approach to Software Engineering*. 3. ed. New York: Springer, 2005. 566 p.

JENNINGS, R.; SYCARA, K.; WOOLDRIDGE, J. *A roadmap of agent research and development*. Journal of Autonomous Agents and Multi-Agent Systems, v. 1, n. 1, p. 7–38, 1998.

LIANG, P.; HUANG, S. *A framework for applying intelligent agents to support electronic trading*. Decision Support Systems, v. 28, p. 305-17, 2000.

MONTEIRO, L. *Diretrizes para Avaliação de Modelos de Sistemas Multiagentes: Uma Proposta Baseada em Métricas*. Rio de Janeiro. Brazil. 2018.

MOULIN, B.; CHAIB-DRAA, B. An Overview of Distributed Artificial Intelligence. In: O'HARE, G.; JENNINGS, N. R. (Eds.). *Foundations of distributed artificial intelligence*. [S.l.]: John Wiley and Sons, 1996. cap.1.

NEWELL, A. *The knowledge level*. Artificial Intelligence, v. 18, n. 1, p. 87–127, 1982.

PAVÓN, J.; GÓMEZ-SANZ, J.; FUENTES-FERNÁNDEZ, R. The INGENIAS methodology and tools. In: HENDERSON-SELLERS, B.; GIORGINI, P. (eds) *Agent-Oriented Methodologies*, Hershey: IGP, 2005, cap. IX, p. 236–276.

_____. *Model Driven Development of Multi-Agent Systems*. ECMDA-FA, Proceedings of Model Driven Architecture- Foundations and Applications, Second European Conference, ECMDA-FA 2006, Bilbao, Espanha, v. 4066, p. 284–298. 2006.

PRESSMAN, R. S. *Engenharia de Software*. 7. ed. Porto Alegre: Pearson Makron Books, 2011.

ROYCE, W. *Managing the Development of Large Software Systems*. Proc. 9th. Intern. Conf. Software Engineering, IEEE Computer Society, 1987, 328-338 Originally published in Proc. WESCON. 1970.

RUSSEL, J.; NORVIG, P. *Artificial intelligence: a modern approach*. New Jersey: Prentice Hall, 1995 apud TURBAN et al. *Electronic commerce: a managerial perspective*. 1. ed. New Jersey: Prentice Hall, 2000.

SCHRAMM, W. *Notes on case studies of instructional media projects*. USA: Washington, 1971. 43 p.

SCHWABER, K.; M. BEEDLE. *Agile Software Development with Scrum 1st*. USA: Prentice Hall PTR, Upper Saddle River, N, 2001. 158 p.

SOMMERVILLE, I. *Engenharia de Software*. 9. ed. São Paulo: Pearson, 2011.

STURM, A.; SHEHORY, O. The landscape of agent-oriented methodologies. In: *Agent-Oriented Software Engineering: Reflections on architectures, methodologies, languages, and frameworks*. Springer Berlin Heidelberg, 2014. pp. 137-154.

TURBAN et al. *Electronic commerce: a managerial perspective*. 1. ed. New Jersey: Prentice Hall, 2000.

WHOLIN, C. et al. *Experimentation in Software Engineering: an introduction*. Springer, 2012. 249 p.

_____. *Experimentation in Software Engineering: an Introduction*. Massachusetts: Kluwer Academic Publishers, 2000.

WOOLDRIDGE, M. Intelligent Agents. In: WEISS, G. *Multiagents systems: A Modern Approach to Distributed Artificial Intelligence*. The MIT Press, 1999.

YIN, R. K. *Case study research*. Design and methods (applied social research methods). 2. ed. Sage Publications, 2009.

APÊNDICE A – Descrição detalhada do MEDEDUC

O MEDEDUC é detalhado da seguinte forma: Ele é formado por um banco de dados de 110 questões e com 3 opções de respostas, estas questões são distribuídas por níveis de conhecimento, são eles: Nivelamento, Nível 1 ao Nível 5 e Reforço do Nível 1 ao Nível 5, e cada um dos níveis é formado por 10 questões relativas a conhecimentos. O quadro 22 mostrará essa divisão de questões por nível de dificuldade.

Quadro 22 – Nível x Quantidade de questões x Grau de conhecimento.

Nível	Quantidade de questões	Grau de conhecimento
Nivelamento	10 questões aleatórias	5 níveis
Nível 1	10 questões	Nível 1
Nível 2	10 questões	Nível 2
Nível 3	10 questões	Nível 3
Nível 4	10 questões	Nível 4
Nível 5	10 questões	Nível 5
Nível 1 - Reforço	10 questões	Nível 1
Nível 2 - Reforço	10 questões	Nível 2
Nível 3 - Reforço	10 questões	Nível 3
Nível 4 – Reforço	10 questões	Nível 4
Nível 5 - Reforço	10 questões	Nível 5

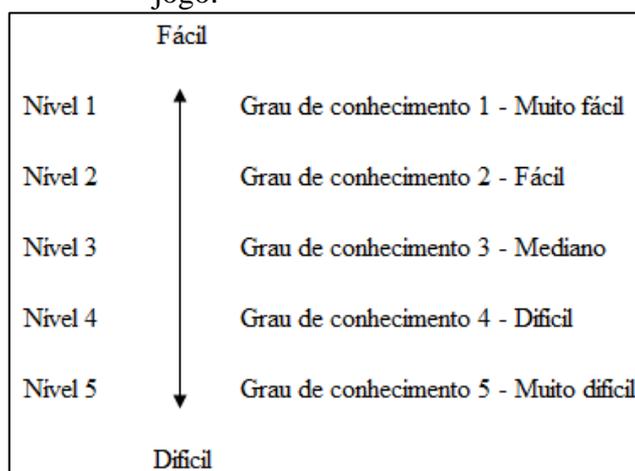
Fonte: A autora, 2018.

O jogo médico educacional em que estamos estudando possui alguns requisitos básicos no que se refere ao comportamento geral do jogo, são eles:

O jogador deve conseguir verificar na página principal quais módulos estão disponíveis para acesso, não deve ter disponível todos os módulos (Nivelamento, Níveis 1,2,3,4 e 5 e Reforço) paralelamente. Cada módulo estará disponível para o jogador, de acordo com a sua evolução no jogo, ele deve ser informado quando for aprovado ou reprovado em algum nível, a complexidade das questões deve crescer juntamente com o nível, as questões disponibilizadas

têm níveis de conhecimento de 1 a 5, sendo as questões de nível 1 mais fáceis comparado as questões de nível 5 (Figura 15).

Figura 15 – Grau de conhecimento por nível do jogo.



Fonte: FERREIRA, 2016.

O nivelamento do jogo médico educacional MEDEDUC:

Na fase de nivelamento existem alguns requisitos básicos que são sinalados no jogo médico educacional, são eles:

As questões são apresentadas nos formatos: áudio, vídeo, imagens e texto, cada uma é constituída por 3 opções de respostas (A, B e C), em que deverá ter uma opção correta e duas incorretas, ao escolher uma opção incorreta, o jogador deve ser notificado da opção correta, ao escolher uma opção incorreta, o jogador deve receber o feedback com uma justificativa relatando o motivo da opção escolhida estar incorreta. Este feedback pode ser em formato de áudio, vídeo, imagens e texto.

No primeiro ingresso no jogo, o jogador deve realizar o nivelamento e as demais funcionalidades não deverão estar disponíveis.

Durante o nivelamento, o jogador deve responder 10 questões relacionado aos cinco níveis disponíveis. Desta forma existem, 2 questões de nível 1, 2 questões de nível 2, 2 questões de nível 3, 2 questões de nível 4 e 2 questões de nível 5. Ele deve ser capaz de acompanhar a sua evolução com uma barra de status. Para questões corretas, o usuário vê a barra da questão pintada de verde na barra de status. E para questões incorretas, o usuário vê a barra da questão pintada de vermelho. O jogador, também, deve ser capaz de clicar sobre a posição indicativa de

cada pergunta na barra de status e rever a questão, resposta e retorno. Contudo, o jogador não deve conseguir alterar a opção selecionada, além disso, mesmo que acerte todas as questões, o jogador deve realizar o Nível 5.

No final do nivelamento, o jogador é considerado apto ao Nível 5, se acertar todas as questões do nível 1, 2, 3 e 4, é considerado apto ao Nível 4, se acertar todas as questões do nível 1, 2, 3, é considerado apto ao Nível 3, se acertar todas as questões do nível 1 e 2, é considerado apto ao Nível 2, se acertar todas as questões do nível 1, se ele errar uma das duas perguntas do Nível 1, ele deve obrigatoriamente iniciar no Nível 1, ao finalizar as questões do nivelamento, o jogador deve receber um retorno informando o nível alcançado, se ficar no Nível 1 durante o nivelamento, ele recebe o Equipamento1 para o seu consultório, se ficar no Nível 2 durante o nivelamento, ele recebe o Equipamento2 para o seu consultório, se ficar no Nível 3 durante o nivelamento, ele recebe o Equipamento3 para o seu consultório, se ficar no Nível 4 durante o nivelamento, ele recebe o Equipamento4 para o seu consultório, o jogador deve ser direcionado para a página principal do jogo e poderá iniciar o nível alcançado no nivelamento, ao selecionar o nível disponível na página principal, o jogador é direcionado para a primeira questão aleatória do Nível que foi alocado, se ele interromper o nivelamento, ele deve iniciar o processo novamente e as respostas anteriores são desconsideradas.

Nos níveis do jogo médico educacional existem alguns requisitos básicos que são sinalados, são eles: As questões são divididas e apresentadas nos formatos: áudio, vídeo, imagens e texto, são compostas por 3 opções de respostas (A, B e C), em que existe uma opção correta e duas incorretas, ao escolher uma opção incorreta o jogador deve ser informado da opção correta, ao escolher uma opção incorreta em uma questão, o jogador deve receber um *feedback* relatando o motivo da opção escolhida estar incorreta. Este feedback pode ser em formato de áudio, vídeo, imagens e texto.

Durante os níveis do jogo, pós realização do nivelamento, o jogador deve iniciar no último nível alcançado, durante o nível 1 o jogador deve responder 10 questões referente ao nível de conhecimento 1, durante o nível 2 o jogador deve responder 10 questões referente ao nível de conhecimento 2, e assim sucessivamente até o nível 5 onde o jogador deve responder 10 questões referente ao nível de conhecimento 5, durante os níveis 1, 2, 3, 4 e 5 o jogador deve ser capaz de acompanhar a sua evolução com uma barra de status. Para questões corretas, ele vê a barra da questão pintada de verde na barra de status. E para questões incorretas, o usuário vê a barra da questão pintada de vermelho, o jogador deve ser capaz de clicar sobre a posição indicativa de cada pergunta na barra de status e rever a questão, resposta e retorno. Contudo, ele não deve conseguir alterar a opção selecionada;

Em relação à pontuação necessária, ao fim de cada nível, caso o jogador deve obter 70% de acertos ou mais, para estar apto a avançar para o próximo nível, ao fim de cada nível, caso o jogador obtenha resultado inferior de 70% de acertos, ele é direcionado para a página principal e precisa realizar o nível de reforço referente ao nível em que foi reprovado, ao finalizar as questões dos níveis 1, 2, 3, 4 ou 5, o jogador deve receber um *feedback* informando a aprovação no nível e para qual próximo nível ele é direcionado; se o jogador realizar um dos níveis 1, 2, 3, 4 ou 5 mais de duas vezes e neste tempo não obtenha mais de 70% de acertos, ele deve ser direcionado para a página principal e iniciar no nível 1 do jogo.

Ao finalizar os níveis, ao concluir com aprovação o Nível 1 (acertos acima de 70%), ele recebe o Equipamento1 para o seu consultório, ao concluir com aprovação o Nível 2 (acertos acima de 70%), ele recebe o Equipamento2 para o seu consultório e assim por diante, até que ao fim dos níveis, o jogador é conduzido para a página principal do jogo e está apto para avançar para o próximo nível, além disso, ao selecionar o nível disponível na página principal, o jogador é direcionado para a primeira questão aleatória do Nível em questão, se o jogador interromper o jogo em qualquer um dos níveis disponíveis, ele deve iniciar o processo novamente e as respostas anteriores serão armazenadas, ao fim do jogo o jogador recebe a pontuação obtida em cada nível e sua média dos níveis.

O quadro 23 exemplifica o funcionamento do nível 1 do MEDEDUC.

Quadro 5 – Exemplo de funcionamento do nível 1 do jogo.

Nível 1	Nível 1	Nível 1	Nível 1
1. Questão 01	2. Questão 02	...	10. Questão 10
a. Opção a	a. Opção a		a. Opção a
b. Opção b	b. Opção b		b. Opção b
c. Opção c	c. Opção c		c. Opção c

Fonte: FERREIRA, 2016.

Sobre o reforço do jogo médico educacional MEDEDUC, só é solicitado se, por exemplo, ao fim do nível 1 o jogador não obtiver 70% ou mais de aproveitamento, será necessário realizar o reforço do nível reprovado, ou seja, do nível 1. Se o jogador obtiver 70% ou mais de acertos no reforço, o jogador está apto a realizar novamente o nível 1. Se o jogador obtiver nota abaixo de 70%, ele deve permanecer no nível de reforço 1 até obter uma média

acima de 70%. O quadro 24 exemplifica a fase de reforço no nível 1 que é composto por 10 questões.

Quadro 6 – Exemplo de funcionamento do nível de reforço 1 do jogo.

Nível Reforço 1	Nível Reforço 1	Nível Reforço 1	Nível Reforço 1
1. Questão 01	2. Questão 02	...	10. Questão 10
a. Opção a	a. Opção a		a. Opção a
b. Opção b	b. Opção b		b. Opção b
c. Opção c	c. Opção c		c. Opção c

Fonte: FERREIRA, 2016.

O detalhamento dos níveis de 2 a 5 seguem a mesma estrutura e requisitos apresentados no nível 1.

APÊNDICE B – Análise de Variância

A análise de variância possui como objetivo principal comparar três ou mais tratamentos, sendo esses tratamentos definidos por uma condição imposta ou objeto que se deseja medir ou avaliar em um experimento. Em um experimento, cada observação Y_{ij} pode ser decomposta conforme o modelo a seguir:

$$Y_{ij} = \mu + \tau_i + \epsilon_{ij} \quad i = 1, \dots, I \text{ e } j = 1, \dots, J \quad (1)$$

Sabe-se que:

Y_{ij} é a observação do i -ésimo tratamento na j -ésima unidade experimental ou parcela;

μ é o efeito constante (média geral);

τ_i é o efeito do i -ésimo tratamento;

ϵ_{ij} é o erro associado ao i -ésimo tratamento na j -ésima unidade experimental ou parcela assumido como: $\epsilon_{ij} \sim \text{IID } N(0, \sigma^2)$. Aqui, IID significa que os erros devem ser independentes e identicamente distribuídos.

A análise de variância, baseia-se na decomposição da variação total da variável resposta em partes que podem ser atribuídas aos tratamentos (variância entre) e ao erro experimental (variância dentro). Essa variação pode ser medida por meio das somas de quadrados definidas para cada um dos seguintes componentes:

$$SQTotal = \sum_{i=1}^I \sum_{j=1}^J y_{ij}^2 - C \quad (2)$$

em que

$$C = \frac{(\sum_{i=1}^I \sum_{j=1}^J y_{ij})^2}{IJ}, \quad (3)$$

$$SQTrat = \frac{\sum_{i=1}^I y_i^2}{J} - C \quad (4)$$

e a soma de quadrados dos resíduos pode ser obtida pela diferença:

$$SQRes = SQTotal - SQTrat. \quad (5)$$

A $SQTrat$ também é chamada de variação entre, que é a variação existente entre os diferentes tratamentos e a $SQRes$ é chamada de variação dentro que é função das diferenças existentes entre as repetições de um mesmo tratamento.

Essas somas de quadrados podem ser organizadas em uma tabela e para testar a hipótese H_0 , utiliza-se o teste F apresentado no quadro da análise de variância (Quadro 25), em que $QMTrat = SQTrat / (I - 1)$ e $QMRes = SQRes / (I(J - 1))$.

Quadro 7 – Análise de variância.

Causas de Variação	Graus de Liberdade	Soma de Quadrados	Quadrados Médios	F calculado
Tratamentos	I-1	SQTrat	QMTrat	QMTrat/QMRes
Resíduo	I(J-1)	SQRes	QMRes	
Total	IJ-1	SQTotal		

Fonte: FERREIRA, 2016.

Se $F_{calculado} > F_{tabelado}$, rejeitamos a hipótese de nulidade H_0 , ou seja, existem evidências de diferença significativa entre pelo menos um par de médias de tratamentos, ao nível α de significância escolhido. Caso contrário, não se rejeitamos a hipótese de nulidade H_0 , ou seja, não há evidências de diferença significativa entre tratamentos, ao nível α de significância escolhido.

APÊNDICE C – Testes estatísticos ANOVA (atraso na entrega)

Tabela 1 – Resultado NÃO se rejeita a hipótese – Grupos 1 e 3.

Anova: Single Factor

SUMMARY

<i>Groups</i>	<i>Count</i>	<i>Sum</i>	<i>Average</i>	<i>Variance</i>
G1	8	3	0,375	1,125
G3	9	34	3,777778	25,1944444

ANOVA

<i>Source of Variation</i>	<i>SS</i>	<i>df</i>	<i>MS</i>	<i>F</i>	<i>P-value</i>
Between Groups	49,04003	1	49,04003	3,5123838	0,080523791
Within Groups	209,4306	15	13,96204		
Total	258,4706	16			

Fonte: A autora, 2018.

Tabela 2 – Resultado NÃO se rejeita a hipótese – Grupos 1 e 4.

Anova: Single Factor

SUMMARY

<i>Groups</i>	<i>Count</i>	<i>Sum</i>	<i>Average</i>	<i>Variance</i>
G1	8	3	0,375	1,125
G4	9	41	4,555556	34,5277778

ANOVA

<i>Source of Variation</i>	<i>SS</i>	<i>df</i>	<i>MS</i>	<i>F</i>	<i>P-value</i>	<i>F crit</i>
Between Groups	74,02042	1	74,02042	3,90819158	0,066732384	4,543077
Within Groups	284,0972	15	18,93981			
Total	358,1176	16				

Fonte: A autora, 2018.

Tabela 34 – Resultado rejeita-se a hipótese – Grupos 1 e 5.

Anova: Single Factor

SUMMARY

<i>Groups</i>	<i>Count</i>	<i>Sum</i>	<i>Average</i>	<i>Variance</i>
G1	8	3	0,375	1,125
G5	9	111	12,33333	152,75

ANOVA

<i>Source of Variation</i>	<i>SS</i>	<i>df</i>	<i>MS</i>	<i>F</i>	<i>P-value</i>	<i>F crit</i>
Between Groups	605,6544	1	605,6544	7,38678	0,015879	4,543077
Within Groups	1229,875	15	81,99167			
Total	1835,529	16				

Fonte: A autora, 2018.

Tabela 4 – Resultado NÃO se rejeita a hipótese – Grupos 1 e 6.

Anova: Single Factor

SUMMARY

<i>Groups</i>	<i>Count</i>	<i>Sum</i>	<i>Average</i>	<i>Variance</i>
G1	8	3	0,375	1,125
G6	9	0	0	0

ANOVA

<i>Source of Variation</i>	<i>SS</i>	<i>df</i>	<i>MS</i>	<i>F</i>	<i>P-value</i>	<i>F crit</i>
Between Groups	0,595588	1	0,595588	1,134454	0,303673	4,543077
Within Groups	7,875	15	0,525			
Total	8,470588	16				

Fonte: A autora, 2018.

Tabela 55 – Resultado rejeita-se a hipótese – Grupos 2 e 3.

Anova: Single Factor

SUMMARY

<i>Groups</i>	<i>Count</i>	<i>Sum</i>	<i>Average</i>	<i>Variance</i>
G2	8	238	29,75	420,5
G3	9	34	3,777778	25,1944444

ANOVA

<i>Source of Variation</i>	<i>SS</i>	<i>df</i>	<i>MS</i>	<i>F</i>	<i>P-value</i>	<i>F crit</i>
Between Groups	2856,944	1	2856,944	13,6258854	0,002177435	4,543077
Within Groups	3145,056	15	209,6704			
Total	6002	16				

Fonte: A autora, 2018.

Tabela 6 – Resultado rejeita-se a hipótese – Grupos 2 e 4.

Anova: Single Factor

SUMMARY

<i>Groups</i>	<i>Count</i>	<i>Sum</i>	<i>Average</i>	<i>Variance</i>
G2	8	238	29,75	420,5
G4	9	41	4,555556	34,5277778

ANOVA

<i>Source of Variation</i>	<i>SS</i>	<i>df</i>	<i>MS</i>	<i>F</i>	<i>P-value</i>	<i>F crit</i>
Between Groups	2688,395	1	2688,395	12,5246616	0,00297551	4,543077
Within Groups	3219,722	15	214,6481			
Total	5908,118	16				

Fonte: A autora, 2018.

Tabela 7 – Resultado rejeita-se a hipótese – Grupos 2 e 5.

Anova: Single Factor

SUMMARY

<i>Groups</i>	<i>Count</i>	<i>Sum</i>	<i>Average</i>	<i>Variance</i>
G2	8	238	29,75	420,5
G5	9	111	12,33333	152,75

ANOVA

<i>Source of Variation</i>	<i>SS</i>	<i>df</i>	<i>MS</i>	<i>F</i>	<i>P-value</i>	<i>F crit</i>
Between Groups	1284,735	1	1284,735	4,626342	0,048193	4,543077
Within Groups	4165,5	15	277,7			
Total	5450,235	16				

Fonte: A autora, 2018.

Tabela 8 – Resultado rejeita-se a hipótese – Grupos 5 e 6.

Anova: Single Factor

SUMMARY

<i>Groups</i>	<i>Count</i>	<i>Sum</i>	<i>Average</i>	<i>Variance</i>
G5	8	238	29,75	420,5
G6	9	0	0	0

ANOVA

<i>Source of Variation</i>	<i>SS</i>	<i>df</i>	<i>MS</i>	<i>F</i>	<i>P-value</i>	<i>F crit</i>
Between Groups	3748,5	1	3748,5	19,10226	0,000548	4,543077
Within Groups	2943,5	15	196,2333			
Total	6692	16				

Fonte: A autora, 2018.

Tabela 9 – Resultado NÃO se rejeita a hipótese – Grupos 1 e 3.

Anova: Single Factor

SUMMARY

<i>Groups</i>	<i>Count</i>	<i>Sum</i>	<i>Average</i>	<i>Variance</i>
G1	22	38	1,727273	0,398268
G3	22	34	1,545455	0,640693

ANOVA

<i>Source of Variation</i>	<i>SS</i>	<i>df</i>	<i>MS</i>	<i>F</i>	<i>P-value</i>	<i>F crit</i>
Between Groups	0,363636	1	0,363636	0,7	0,407519	4,072654
Within Groups	21,81818	42	0,519481			
Total	22,18182	43				

Fonte: A autora, 2018.

Tabela 10 – Resultado NÃO se rejeita a hipótese – Grupos 1 e 4.

Anova: Single Factor

SUMMARY

<i>Groups</i>	<i>Count</i>	<i>Sum</i>	<i>Average</i>	<i>Variance</i>
G1	22	38	1,727273	0,398268
G4	22	35	1,590909	0,634199

ANOVA

<i>Source of Variation</i>	<i>SS</i>	<i>df</i>	<i>MS</i>	<i>F</i>	<i>P-value</i>	<i>F crit</i>
Between Groups	0,204545	1	0,204545	0,396226	0,532454	4,072654
Within Groups	21,68182	42	0,516234			
Total	21,88636	43				

Fonte: A autora, 2018.

Tabela 11 – Resultado NÃO se rejeita a hipótese – Grupos 1 e 5.

Anova: Single Factor

SUMMARY

<i>Groups</i>	<i>Count</i>	<i>Sum</i>	<i>Average</i>	<i>Variance</i>
G1	22	38	1,727273	0,398268
G5	22	35	1,590909	0,538961

ANOVA

<i>Source of Variation</i>	<i>SS</i>	<i>df</i>	<i>MS</i>	<i>F</i>	<i>P-value</i>	<i>F crit</i>
Between Groups	0,204545	1	0,204545	0,43649	0,512429	4,072654
Within Groups	19,68182	42	0,468615			
Total	19,88636	43				

Fonte: A autora, 2018.

Tabela 12 – Resultado NÃO se rejeita a hipótese – Grupos 1 e 6.

Anova: Single Factor

SUMMARY

<i>Groups</i>	<i>Count</i>	<i>Sum</i>	<i>Average</i>	<i>Variance</i>
G1	22	38	1,727273	0,398268
G6	22	42	1,909091	0,181818

ANOVA

<i>Source of Variation</i>	<i>SS</i>	<i>df</i>	<i>MS</i>	<i>F</i>	<i>P-value</i>	<i>F crit</i>
Between Groups	0,363636	1	0,363636	1,253731	0,269204	4,072654
Within Groups	12,18182	42	0,290043			
Total	12,54545	43				

Fonte: A autora, 2018.

Tabela 13 – Resultado NÃO se rejeita a hipótese – Grupos 2 e 3.

Anova: Single Factor

SUMMARY

<i>Groups</i>	<i>Count</i>	<i>Sum</i>	<i>Average</i>	<i>Variance</i>
G2	22	38	1,727273	0,398268
G3	22	34	1,545455	0,640693

ANOVA

<i>Source of Variation</i>	<i>SS</i>	<i>df</i>	<i>MS</i>	<i>F</i>	<i>P-value</i>	<i>F crit</i>
Between Groups	0,363636	1	0,363636	0,7	0,407519	4,072654
Within Groups	21,81818	42	0,519481			
Total	22,18182	43				

Fonte: A autora, 2018.

Tabela 14 – Resultado NÃO se rejeita a hipótese – Grupos 2 e 4.

Anova: Single Factor

SUMMARY

<i>Groups</i>	<i>Count</i>	<i>Sum</i>	<i>Average</i>	<i>Variance</i>
G2	22	38	1,727273	0,398268
G4	22	35	1,590909	0,634199

ANOVA

<i>Source of Variation</i>	<i>SS</i>	<i>df</i>	<i>MS</i>	<i>F</i>	<i>P-value</i>	<i>F crit</i>
Between Groups	0,204545	1	0,204545	0,396226	0,532454	4,072654
Within Groups	21,68182	42	0,516234			
Total	21,88636	43				

Fonte: A autora, 2018.

Tabela 15 – Resultado NÃO se rejeita a hipótese – Grupos 2 e 5.

Anova: Single Factor

SUMMARY

<i>Groups</i>	<i>Count</i>	<i>Sum</i>	<i>Average</i>	<i>Variance</i>
G2	22	38	1,727273	0,398268
G5	22	35	1,590909	0,538961

ANOVA

<i>Source of Variation</i>	<i>SS</i>	<i>df</i>	<i>MS</i>	<i>F</i>	<i>P-value</i>	<i>F crit</i>
Between Groups	0,204545	1	0,204545	0,43649	0,512429	4,072654
Within Groups	19,68182	42	0,468615			
Total	19,88636	43				

Fonte: A autora, 2018.

Tabela 16 – Resultado NÃO se rejeita a hipótese – Grupos 2 e 6.

Anova: Single Factor

SUMMARY

<i>Groups</i>	<i>Count</i>	<i>Sum</i>	<i>Average</i>	<i>Variance</i>
G2	22	38	1,727273	0,398268
G6	22	42	1,909091	0,181818

ANOVA

<i>Source of Variation</i>	<i>SS</i>	<i>df</i>	<i>MS</i>	<i>F</i>	<i>P-value</i>	<i>F crit</i>
Between Groups	0,363636	1	0,363636	1,253731	0,269204	4,072654
Within Groups	12,18182	42	0,290043			
Total	12,54545	43				

Fonte: A autora, 2018.

APÊNDICE E – Questionário de avaliação da qualidade

Quadro 26 – Questionário de avaliação da qualidade (continua).

DESENVOLVEDOR		<nome do desenvolvedor>		
DIMENSÃO: QUALIDADE DO MODELO				
O modelo deve representar corretamente o Domínio do Problema e possuir seus principais conceitos.				
Questão	Componente	Métrica	Resposta	Pontuação
Q1: De uma forma geral, o modelo representa corretamente o Domínio do Problema?	Representação	Corretude		
		Conformidade		
Q2: O modelo engloba os principais conceitos do Domínio do Problema?	Conceitos	Completeza		
		Concisão		
		Cobertura de contexto		
Q3: O modelo atingiu às expectativas e finalidades iniciais?	Expectativas	Conformidade		
		Adequação		
	Objetivos	Conformidade		
		Adequação		
DIMENSÃO: QUALIDADE DO DESEMPENHO				
Cobertura do modelo de todos os recursos, funções e resultados existentes no Domínio do Problema. Bom relacionamento dos agentes de forma a garantir os objetivos da modelagem como um todo.				
Questão	Componente	Métrica	Resposta	Pontuação
Q1: O modelo possui todos os recursos, que o Domínio do Problema modelado apresenta?	Recursos	Completeza		
Q2: O modelo possui todas as funções, que o Domínio do Problema modelado apresenta?	Funções	Completeza		
Q3: O modelo possui todos os resultados, que o Domínio do Problema modelado apresenta?	Resultados	Completeza		
Q4: Os agentes possuem um bom relacionamento de forma a garantir os objetivos da modelagem?	Relacionamento dos agentes	Eficácia		
		Conformidade		
Q5: Os agentes possuem uma boa comunicação de forma a garantir os objetivos da modelagem?	Comunicação dos agentes	Eficácia		
		Conformidade		
DIMENSÃO: QUALIDADE DOS RECURSOS				
Os agentes do modelo devem representar/gerenciar adequadamente os recursos do Domínio do Problema. Todos os atributos dos recursos devem estar representados no modelo.				

Quadro 26 – Questionário de avaliação da qualidade (continuação)

Questão	Componente	Métrica	Resposta	Pontuação
Q1: O modelo fornece agentes que representam adequadamente os recursos do Domínio do Problema?	Representação de Recursos	Adequação		
Q2: O modelo fornece agentes que gerenciam adequadamente os recursos do Domínio do Problema?	Gerência de atributos dos recursos	Adequação		
DIMENSÃO: CONFIABILIDADE				
Os agentes do modelo devem representar/gerenciar os recursos, capacidades e funcionalidades sem a ocorrência de erros. As alterações ocorridas nos parâmetros do Domínio do Problema, desde que estejam corretas, não devem influenciar na qualidade de representação do modelo.				
Questão	Componente	Métrica	Resposta	Pontuação
Q1: Os agentes do modelo que representam os recursos o fazem sem a ocorrência de erros?	Representação de Recursos	Corretude		
Q2: Os agentes do modelo que gerenciam os recursos o fazem sem a ocorrência de erros?	Gerência de Recursos	Corretude		
Q3: Os agentes do modelo que representam as capacidades o fazem sem a ocorrência de erros?	Representação de capacidades	Corretude		
Q4: Os agentes do modelo que gerenciam as capacidades o fazem sem a ocorrência de erros?	Gerência de capacidades	Corretude		
Q5: Os agentes do modelo que representam as funcionalidades o fazem sem a ocorrência de erros?	Representação de funcionalidades	Corretude		
Q6: Os agentes do modelo que gerenciam as funcionalidades o fazem sem a ocorrência de erros?	Gerência de funcionalidades	Corretude		
Q7: A alteração dos parâmetros do Domínio do Problema influencia na qualidade da representação do modelo?	Alteração de parâmetros	Corretude Eficácia		
Q8: O modelo é seguro o suficiente para proteger dados de partes interessadas?	Proteção de Dados	Segurança		
DIMENSÃO: QUALIDADE DA MODELAGEM				
A modelagem de Sistemas Multiagentes deve estar conforme os padrões propostos na metodologia. A metodologia utilizada deve ser a metodologia mais adequada para modelar o Domínio do Problema.				
Questão	Componente	Métrica	Resposta	Pontuação
Q1: A modelagem de Sistemas Multiagentes do Domínio do Problema está conforme os padrões propostos na metodologia?	Metodologia	Conformidade		
Q2: A metodologia de Sistemas Multiagentes é a mais adequada para a modelagem do Domínio do Problema?	Metodologia	Adequação		
Q3: A ferramenta utilizada para a modelagem é a mais adequada para a metodologia aplicada?	Ferramenta	Adequação		

Quadro 26 – Questionário de avaliação da qualidade (continuação)

DIMENSÃO: DURABILIDADE				
Correções/ modificações no Sistema Multiagente gerado na modelagem devem ser realizadas sem a geração de efeitos colaterais indesejados no modelo. A confiabilidade do modelo deve aumentar no decorrer do tempo com estas correções/ modificações.				
Questão	Componente	Métrica	Resposta	Pontuação
Q1: O Sistema Multiagente gerado na modelagem pode ser modificado sem a geração de efeitos colaterais indesejados no modelo?	Sistema Multiagente	Modificabilidade		
Q2: As mudanças farão com que a confiabilidade aumente no decorrer do tempo?	Mudanças	Maturidade		
DIMENSÃO: FACILIDADE DE MANUTENÇÃO				
O tempo de correção do Sistema Multiagente gerado na modelagem deve ser razoável (não muito longo). Todas as documentações e informações necessárias para correção de defeitos devem estar disponíveis para a equipe de suporte.				
Questão	Componente	Métrica	Resposta	Pontuação
Q1: O Sistema Multiagente gerado na modelagem pode ser corrigido em um período de tempo aceitável?	Sistema Multiagente	Eficiência		
		Modificabilidade		
Q2: As informações e documentações necessárias para correção de defeitos estão disponíveis para a equipe de suporte do sistema?	Documentações	Disponibilidade		
DIMENSÃO: QUALIDADE DE AVALIAÇÃO				
Os resultados obtidos na avaliação do modelo devem estar de acordo com os resultados esperados para o Domínio do Problema. Os testes realizados para avaliar o modelo devem ser suficientes para garantir sua integridade.				
Questão	Componente	Métrica	Resposta	Pontuação
Q1: Os resultados da avaliação do modelo estão de acordo com os resultados esperados para o Domínio do Problema?	Resultados da avaliação	Conformidade		
Q2: Os testes realizados para avaliar o modelo foram suficientes para garantir sua integridade?	Testes	Completeza		
	Modelo	Integridade		
Q3: Todos os componentes do modelo foram testados?	Componentes do modelo	Completeza		
Q4: Os resultados da avaliação estão dentro da faixa comum?	Resultados dentro do intervalo esperado	Acurácia		

Quadro 26 – Questionário de avaliação da qualidade (conclusão).

DIMENSÃO: QUALIDADE DE AGENTES				
Todos os papéis e responsabilidades do Domínio do Problema devem estar representados no modelo. As habilidades e tarefas dos agentes devem ser suficientes para o Domínio do Problema.				
Questão	Componente	Métrica	Resposta	Pontuação
Q1: Todos os papéis presentes no Domínio do Problema estão representados no modelo?	Papéis	Completeza		
Q2: Todas as responsabilidades dos papéis presentes no Domínio do Problema estão representados no modelo?	Responsabilidades dos Papéis	Completeza		
Q3: As habilidades realizadas pelos agentes são suficientes para o Domínio do Problema?	Habilidades dos agentes	Cobertura de contexto		
Q4: As tarefas realizadas pelos agentes são suficientes para o Domínio do Problema?	Tarefas dos agentes	Cobertura de contexto		
Q5: Os agentes possuem boa percepção do ambiente de forma a reagir a estímulos para atingir aos objetivos de design?	Percepção do ambiente	Conformidade		
		Eficiência		
	Agentes reativos	Conformidade		
		Eficiência		
Q6: Os agentes são proativos no momento necessário para atingir aos objetivos de design?	Agentes proativos	Conformidade		
		Eficiência		
INFORME ABAIXO A METODOLOGIA UTILIZADA EM SUA MODELAGEM:				
<nome da metodologia>				

Fonte: A autora, 2018.

APÊNDICE F – Descrição das dimensões e métricas da avaliação da qualidade

Quadro 27 – Descrição das dimensões do questionário de avaliação da qualidade (continua).

Dimensão	Descrição
Qualidade do modelo	O modelo deve representar corretamente o Domínio do Problema e possuir seus principais conceitos.
Qualidade do desempenho	Cobertura do modelo de todos os recursos, funções e resultados existentes no Domínio do Problema. Bom relacionamento dos agentes de forma a garantir os objetivos da modelagem como um todo.
Qualidade dos recursos	Os agentes do modelo devem representar/ gerenciar adequadamente os recursos do Domínio do Problema. Todos os atributos dos recursos devem estar representados no modelo.
Confiabilidade	Os agentes do modelo devem representar/ gerenciar os recursos, capacidades e funcionalidades sem a ocorrência de erros. As alterações ocorridas nos parâmetros do Domínio do Problema, desde que estejam corretas, não devem influenciar na qualidade de representação do modelo.
Qualidade da modelagem	A modelagem de Sistemas Multiagentes deve estar conforme os padrões propostos na metodologia. A metodologia utilizada deve ser a metodologia mais adequada para modelar o Domínio do Problema.
Durabilidade	Correções/ modificações no Sistema Multiagente gerado na modelagem devem ser realizadas sem a geração de efeitos colaterais indesejados no modelo. A confiabilidade do modelo deve aumentar no decorrer do tempo com estas correções/ modificações.
Facilidade de manutenção	O tempo de correção do Sistema Multiagente gerado na modelagem deve ser razoável (não muito longo). Todas as documentações e informações necessárias para correção de defeitos devem estar disponíveis para a equipe de suporte.
Qualidade da avaliação	Os resultados obtidos na avaliação do modelo devem estar de acordo com os resultados esperados para o Domínio do Problema. Os testes realizados para avaliar o modelo devem ser suficientes para garantir sua integridade.
Qualidade de agentes	Todos os papéis e responsabilidades do Domínio do Problema devem estar representados no modelo. As habilidades e tarefas dos agentes devem ser suficientes para o Domínio do Problema.
Acurácia	Capacidade de prover, com o grau de precisão necessário, resultados ou efeitos corretos conforme acordados.

Quadro 27 – Descrição das dimensões do questionário de avaliação da qualidade (conclusão).

Métrica	Descrição
Adequação	Capacidade de fornecer funções que atendam às necessidades declaradas e implícitas quando usadas sob condições especificadas.
Cobertura de contexto	Capacidade de ser usado com eficácia, eficiência, liberdade de risco e satisfação tanto em contextos específicos de uso como em outros contextos.
Completeza	Capacidade de não faltar nada do que é exigido.
Concisão	Capacidade de utilizar o estritamente necessário.
Conformidade	Capacidade de estar de acordo com normas, convenções, regulamentações ou prescrições similares.
Corretude	Capacidade de ser isento de erros ou faltas.
Disponibilidade	Capacidade de ser operacional e acessível para uso, quando necessário.
Eficácia	Capacidade de permitir que usuários atinjam metas especificadas com acurácia e completitude, em um contexto de uso especificado.
Eficiência	Capacidade de apresentar desempenho apropriado, relativo à quantidade de recursos usados, sob condições especificadas.
Integridade	Capacidade de estar inteiro (no sentido de não faltar nenhuma das partes).
Maturidade	Capacidade de atender às necessidades de confiabilidade em operação normal.
Modificabilidade	Capacidade de ser efetivamente e eficientemente modificado sem apresentar defeitos ou degradar a qualidade do produto existente.
Segurança	Capacidade de proteger informações e dados para que pessoas, produtos ou sistemas tenham o apropriado grau de acesso a dados, com seus tipos e níveis de autorização.

Fonte: A autora, 2018.

ANEXO A – Expectativas dos envolvidos no estudo de caso.

Quadro 28 – Descrição das métricas do questionário de avaliação da qualidade.

Aprender sobre sistemas multiagentes e desenvolver um jogo que tenha uma boa aplicação na área médica, sendo útil.

Expandir meus conhecimentos sobre agentes e sistemas multiagentes, tema que tenho bastante interesse.

Entender melhor sobre agentes e sistemas multiagentes e desenvolver um bom jogo relacionado a esse assunto, que possa ajudar muitas pessoas.

Aprender sobre a matéria, ter conhecimento sobre o jogo proposto e das metodologias.

Aprender sobre o assunto proposto.

Ter conhecimento e experiência prática de uso de uma ferramenta e metodologia que tenho apenas um conhecimento superficial.

Adquirir conhecimentos que possam ser úteis, não só na universidade, quanto no mercado de trabalho.

Aplicar todo conhecimento adquirido na matéria em um sistema que possa ajudar pessoas.

Aprender sobre sistemas agentes e multiagentes, assim sobre as metodologias.

Ter a possibilidade de trabalhar na prática com Inteligência Artificial é algo que desperta meu interesse. Pensar nos agentes, elaborar diagramas e ver seu trabalho tendo frutos é algo muito gratificante. Estou com boas expectativas para o trabalho proposto.

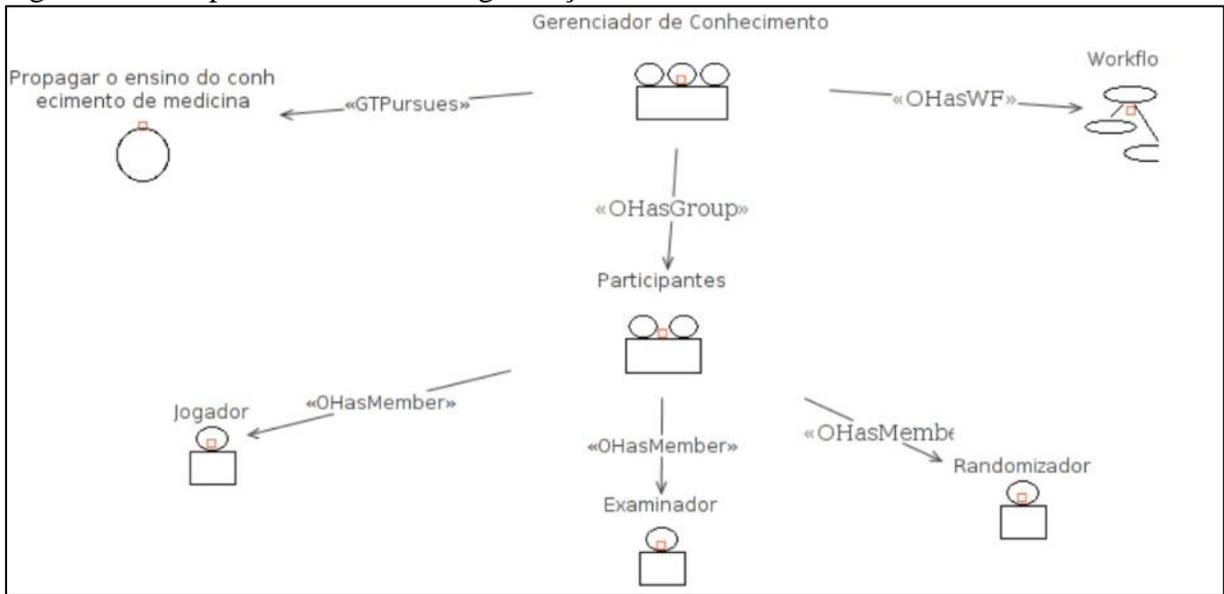
Aprofundar o conhecimento em desenvolvimento de agentes/multiagentes e adquirir conhecimento sobre as possíveis aplicações.

Adquirir e agregar conhecimento na área e sobre as ferramentas que serão utilizadas.

ANEXO B - Artefatos desenvolvidos pelo grupo 1 – INGENIAS.

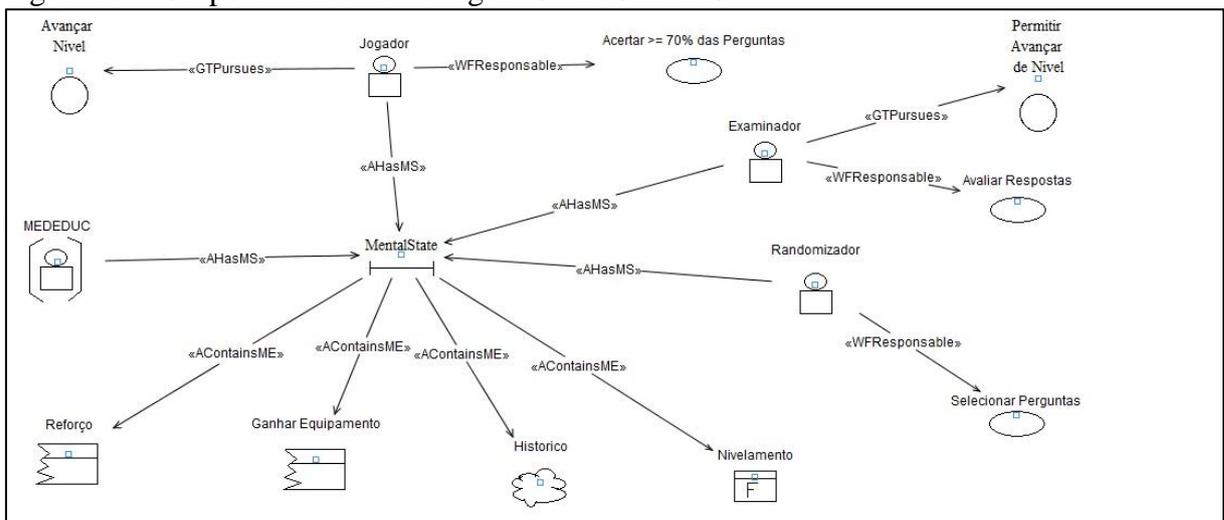
- Modelo de organização:

Figura 16 – Grupo 1 – Modelo de Organização - INGENIAS.



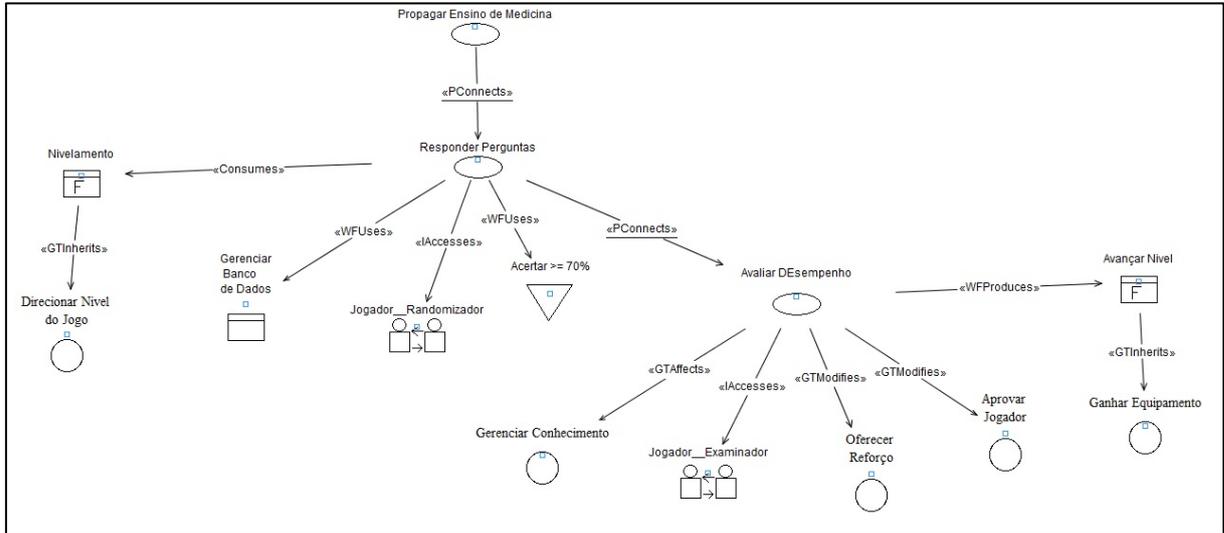
- Modelo de agentes:

Figura 17 – Grupo 1 – Modelo de Agentes - INGENIAS.



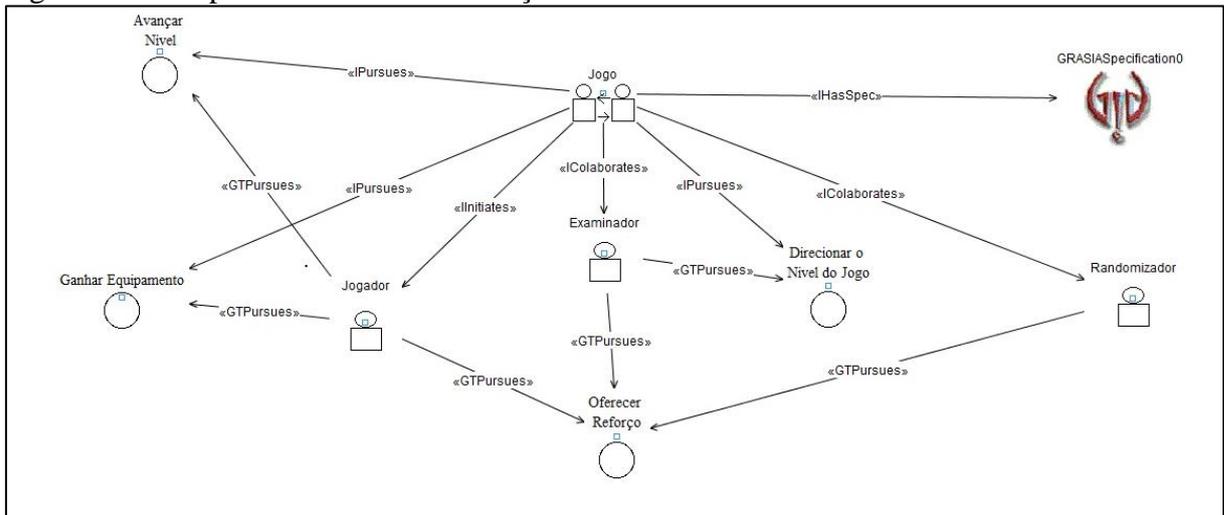
- Modelo de metas e tarefas:

Figura 18 – Grupo 1 – Modelo de Metas e Tarefas - INGENIAS.



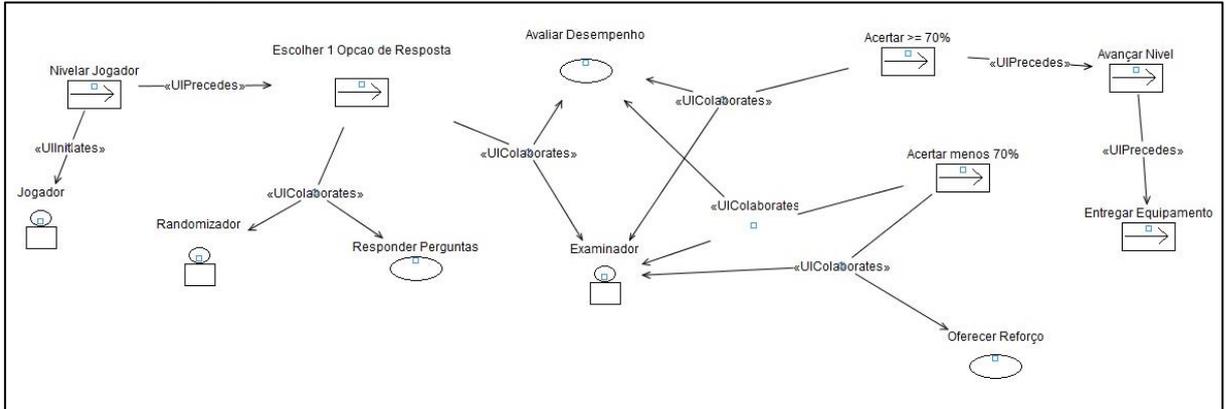
- Modelo de interação:

Figura 19 – Grupo 1 – Modelo de Interação - INGENIAS.



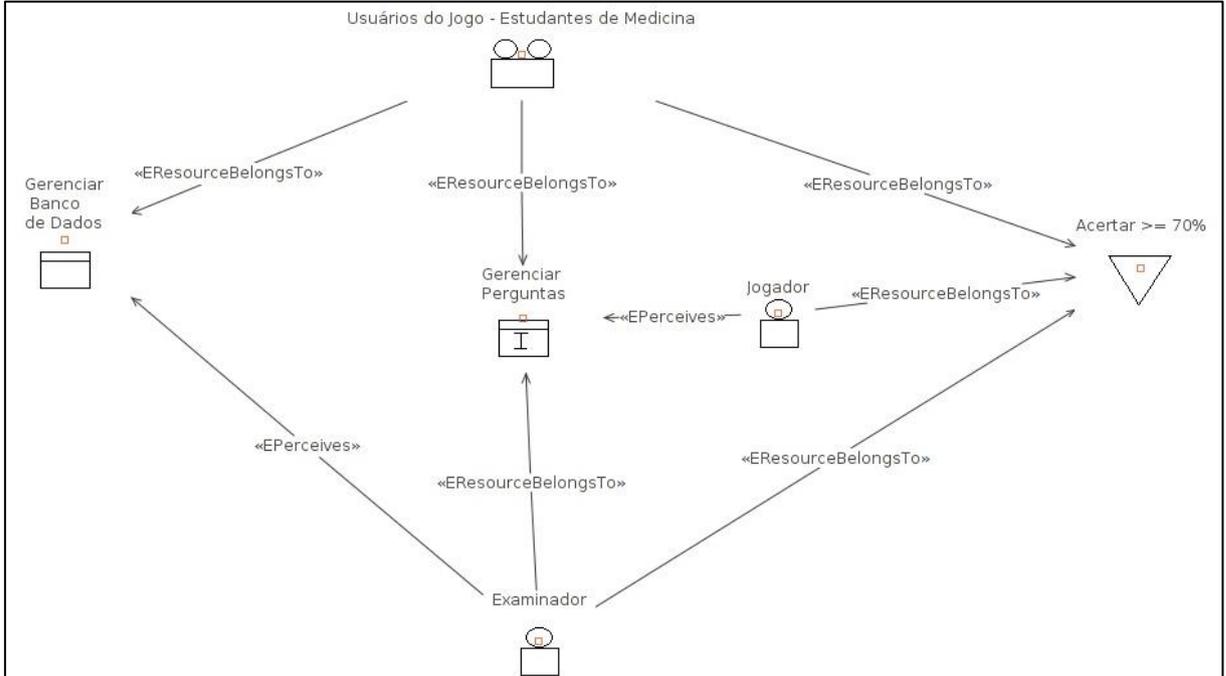
- Modelo GRASIA:

Figura 20 – Grupo 1 – Modelo GRASIA- INGENIAS.



- Modelo de ambiente:

Figura 21 – Grupo 1 – Modelo de Ambiente - INGENIAS.



- Telas do jogo do Grupo 3 – INGENIAS

Figura 22 – Grupo 1 - Tela Inicial - INGENIAS.

The screenshot shows the login interface for MEDEDUC. At the top left is the logo 'MEDEDUC' and at the top right are the links 'Entre' and 'Cadastre-se'. A light blue banner contains the text 'Para continuar, efetue login ou registre-se.'. Below this is the 'Sign in' section, which includes an 'Email' field with the text 'Annyj', a 'Senha' field with masked characters '*****', a checkbox for 'Lembrar de mim', a blue 'Sign in' button, and a link for 'Esqueceu sua senha?'.

Figura 23 – Grupo 1 –Tela de Cadastro de Usuário- INGENIAS.

The screenshot shows the user registration interface for MEDEDUC. At the top left is the logo 'MEDEDUC' and at the top right are the links 'Entre' and 'Cadastre-se'. The form includes three input fields: 'Email', 'Senha', and 'Repetir senha'. A blue 'Sign up' button is located at the bottom left of the form.

Figura 24 – Grupo 1 – Parte do Banco de dados - Nivelamento - INGENIAS

MEDEDUC									Editar seu cadastro	Editar perguntas	Sair
Nivelamento											
Id	Pergunta	Opção 1	Opção 2	Opção 3	Justificativa 1	Justificativa 2	Justificativa 3	Resposta			
103	Um menino com 1 mês de vida, raça negra, morador de área urbana, com condições de vida adequadas, é l...	admita que a suplementação pode ser dispensada enquanto a criança estiver em aleitamento materno excl...	esclareça que morar em área urbana e ser de raça negra são considerados fatores de risco para raquit...	admita que a suplementação é facultativa, desde que programem a coleta de cálcio e fósforo no sexto m...	A quantidade de vitamina D do leite humano é insuficiente, pois tem habitualmente uma concentração d...	Resposta Correta - A prevenção do raquitismo através da exposição solar está na dependência do tempo...	Não há indicação de aferição de níveis séricos de cálcio e fósforo, como afirma a opção 3, assim com...	1	Editar		
104	Um menino, com 3 anos e meio de idade, morador de zona rural, é levado ao médico pelos pais, que se q...	a principal hipótese é de anemia ferropriva, uma vez que, além da microcitose e hipocromia, observa...	a presença de microcitose e hipocromia em criança bem nutrida impõe a o- talassemia como principal di...	o diagnóstico de anemia falciforme deve ser excluído com a realização da prova de falcização, uma vez...	alternativa correta: anemia ferropriva apresenta-se com microcitose, hipocromia e aumento do elevada...	criança bem nutrida não significa aporte de ferro adequado.	anemia falciforme não tem alta capacidade de ligação do ferro.	1	Editar		
106	O médico da Unidade Básica de Saúde é acionado pela enfermeira, por solicitação dos pais, para averig...	tétano	poliomielite	coqueluche	Tétano - Pode provocar reações locais, como dor, eritema, edema ou induração local por até dez dias	Poliomielite - O principal risco associado (vacina oral) é a paralisia infantil associada à vacina. ...	Resposta correta - Coqueluche - É o componente da vacina DPT que mais frequentemente causa efeitos C...	1	Editar		
110	Qual a especialidade médica estudada para se cuidar do ouvido, do nariz, seios paranasais, faringe, l...	otorrinolaringologista	cardiologista	psiquiatra	Resposta Correta: Otorrinolaringologista, essa especialidade médica é a mais completa no ramo da medi...	Especialista cardíaco.	Psiquiatria é uma especialidade da Medicina que lida com a prevenção, atendimento, diagnóstico, trata...	1	Editar		
111	Qual a especialidade que deve ser usada para o	pediatra	ortopedista	clínico geral	Simplificando, a pediatria é a especialidade médica que cuida da	Resposta correta, a ortopedia trata dos	é a especialidade médica que trata de	2	Editar		

Figura 25 – Grupo 1 – Parte do Banco de dados – Nível 1 - INGENIAS.

Perguntas											
Nível 1											
Id	Pergunta	Opção 1	Opção 2	Opção 3	Justificativa 1	Justificativa 2	Justificativa 3	Resposta			
1	Qual a especialidade médica usada para estudar doenças causadas por vírus, bactérias, protozoários e ...	Infectologista	Dermatologista	Dermatologista	resposta correta	é o médico especialista no diagnóstico, tratamento e prevenção de doenças da pele, pelos, mucosas, ca...	Simplificando, a pediatria é a especialidade médica que cuida da manutenção da saúde de crianças e ad...	1	Editar		
4	Qual é o nome da especialidade que faz diagnóstico por imagem? (RAIO-X)?	Radiologia	Mamografia	Ginecologia	Resposta correta	A mamografia ou mastografia é um exame de diagnóstico por imagem, que tem como finalidade estudar o t...	A ginecologia é a prática da medicina que lida diretamente com a saúde do aparelho reprodutor feminin...	1	Editar		
5	Qual é o instrumento usado pelos cirurgiões para fazer incisões?	Seringa	Gase	Bisturi	Uma seringa é um equipamento de bombeamento, provido de uma agulha, usado por profissionais da área d...	tecido leve, ordinariamente de algodão, muito poroso, estéril ou esterilizável, de tamanho variável ...	resposta correta. Ele é um instrumento muito afiado, caro e perigoso.	3	Editar		
6	Qual é a	Ortopedia	Geriatria	Gastroenterologia	A ortopedia é a	Medicina geriátrica	resposta correta.	3	Editar		

Figura 26 – Grupo 1 – Questão – Nível 1 - INGENIAS.

MEDEDUC Editar seu cadastro Sair

Respostum was successfully updated.

Nível: 1 Pergunta 2 de 10

1 2 3 4 5 6 7 8 9 10

↑

Qual é o nome da especialidade que faz diagnostico por imagem? (RAIO-X)?

Radiologia +

Mamografia -

Ginecologia -

Parabéns, você acertou

Anterior Próxima

Figura 27 – Grupo 1 – Questão com Imagem – Nível 1 - INGENIAS.

Nível: 1 Pergunta 1 de 10

1 2 3 4 5 6 7 8 9 10

↑

Qual a especialidade médica usada para estudar doenças causadas por vírus, bactérias, protozoários e outros?

Vitamina D
Reumatologista Cristiano Zerbini dá dicas sobre o nutriente

Onde conseguir?

antes das 10h da manhã

É recomendado usar protetor solar antes de sair para evitar a absorção excessiva de radiação UVB.

No mínimo 3 vezes por semana

Não se deve usar filtro solar antes de sair para evitar a absorção excessiva de radiação UVB.

15 a 20 minutos

Pele saudável, que é fundamental para a absorção (depende por tipo de pele). O ideal é expor braços, pernas, pescoço e rosto.

Os alimentos são responsáveis pelos outros 10% de vitamina D. Leite, ovos, margarina e peixes de água fria, como salmão, são ótimas fontes.

Suplementos vitamínicos, indicados para idosos e pessoas que não podem se expor ao sol com frequência.

Que benefícios a vitamina D traz?

Ajudar a absorver o cálcio do intestino e evitar a osteoporose (doença que enfraquece os ossos e pode causar fraturas principalmente na coluna, bacia, nos punhos e nos cotovelos).

Mantém o equilíbrio e evita quedas. Cuidado com a desidratação.

Como detectar a falta?

Em testes laboratoriais, pela análise do cálcio na urina (se houver pouco, pode indicar falta de cálcio e também de vitamina D).

Exames de sangue

Em crianças, pode se manifestar como o raquitismo (doença que inclui fraqueza e perda óssea).

Nos adultos, não costuma ter sintomas, exceto por uma eventual falta de equilíbrio.

Alerta

Novo: false

Infectologista

Dermatologista

Dermatologista

Figura 28 – Grupo 1 –Tela de Erro de Questão – Nível 1 - INGENIAS.

MEDEDUC Editar seu cadastro Sair

Respostum was successfully updated.

Nível: 1 Pergunta 3 de 10

1 2 3 4 5 6 7 8 9 10

Qual é o instrumento usado pelos cirurgiões para fazer incisões?

Seringa

Gase

Bisturi

Você errou

tecido leve, ordinariamente de algodão, muito poroso, estéril ou esterilizável, de tamanho variável conforme o uso a que se destina, de largo emprego em curativos, intervenções cirúrgicas etc., impregnado ou não de substâncias várias, como anti-sépticos;

Anterior Próxima

Figura 29 – Grupo 1 –Tela de Feedback– Nível 1 - INGENIAS.

MEDEDUC Editar seu cadastro Sair

Respostum was successfully updated.

Nível: 1 Pergunta 3

1 2 3 4 5 6 7

Qual é o instrumento usado pelos cirurgiões para fazer incisões?

Seringa

Gase

Bisturi

Você errou

tecido leve, ordinariamente de algodão, muito poroso, estéril ou esterilizável, de tamanho variável conforme o uso a que se destina, de largo emprego em curativos, intervenções cirúrgicas etc., impregnado ou não de substâncias várias, como anti-sépticos;

Justificativa da resposta 1

Uma seringa é um equipamento de bombeamento, provido de uma agulha, usado por profissionais da área da saúde (ou eventualmente por usuários de drogas) para: inserir substâncias líquidas por via intravenosa, intramuscular, intracardíaca, subcutânea, intradérmica, intra-articular; retirar sangue; ou, ainda, realizar uma punção aspirativa em um paciente.

Close

Anterior Próxima

Figura 306 – Grupo 1 – Tela de Aprovação – Nível 2 - INGENIAS.

MEDEDUC Editar seu cadastro Sair

Nível: 2 Pergunta 1 de 10

1 2 3 4 5 6 7 8 9 10

↑

 Parabéns. Você acertou mais de 70% das perguntas e foi aprovado
 Você ganhou um prêmio : **Escada**

O que é leucemia?

Neoplasia dos órgãos produtores de sangue 

Neoplasia dos órgãos 

Neoplasia do SNC 

 **Parabéns, você acertou**

[Próxima](#)

Figura 31 – Grupo 1 –Tela de Reforço – Nível 1 - INGENIAS.

MEDEDUC Editar seu cadastro Editar perguntas Sair

Respostas were successfully updated.

Nível: 1 Pergunta 8 de 10

1 2 3 4 5 6 7 8 9 10

↑

 Infelizmente você não foi aprovado.
 Verifique o manual para fazer o reforço: [Manual](#)

A amenorreia, é um dos sintomas característicos e específicos da hemocrematose cuja definição é o acúmulo de ferro e cobre no fígado causando Hepatite por Hereditariedade. Sobre esse sintoma é correto afirmar que?

Amenorreia células do tipo rítopodes atuam na locomoção e aquisição de alimentos no fígado 

Processo onde o sistema imune ataca as células do próprio fígado 

Ausência de menstruação 

 **Você errou**

A causa da amenorreia nem sempre é uma doença. Alguns fenômenos naturais, como gravidez, amamentação e menopausa, também podem interromper os ciclos menstruais.

[Anterior](#)

[Próxima](#)

Figura 32 – Grupo 1 –Tela do Consultório– Nível 1 - INGENIAS.

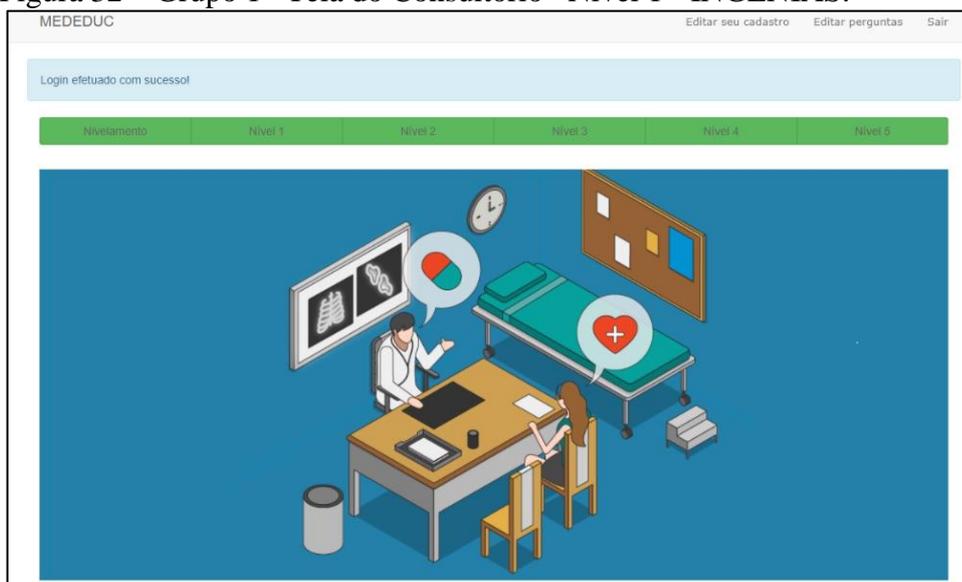


Figura 33 – Grupo 1 –Tela Final – Nível 1 - INGENIAS

Você conquistou 5 de 5 equipamentos

Parabéns, você concluiu o jogo.

Você acertou 48 das 60 pergunta(s) respondidas

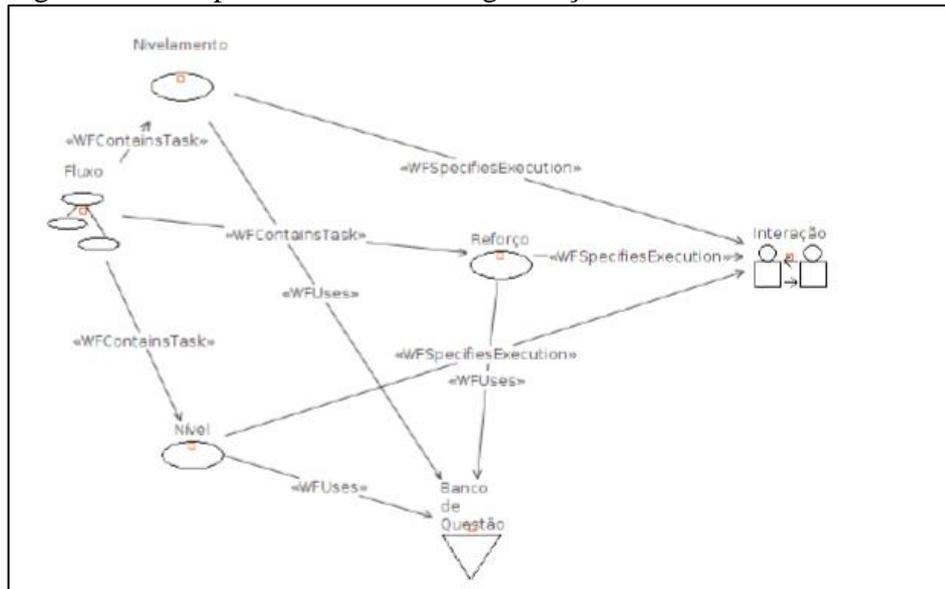
Nível	Pontuação
Nivelamento	5/10
Nível 1	9/10
Nível 2	8/10
Nível 3	8/10
Nível 4	8/10
Nível 5	10/10

Média: 80.0%

ANEXO C - Artefatos desenvolvidos pelo grupo 2 – INGENIAS.

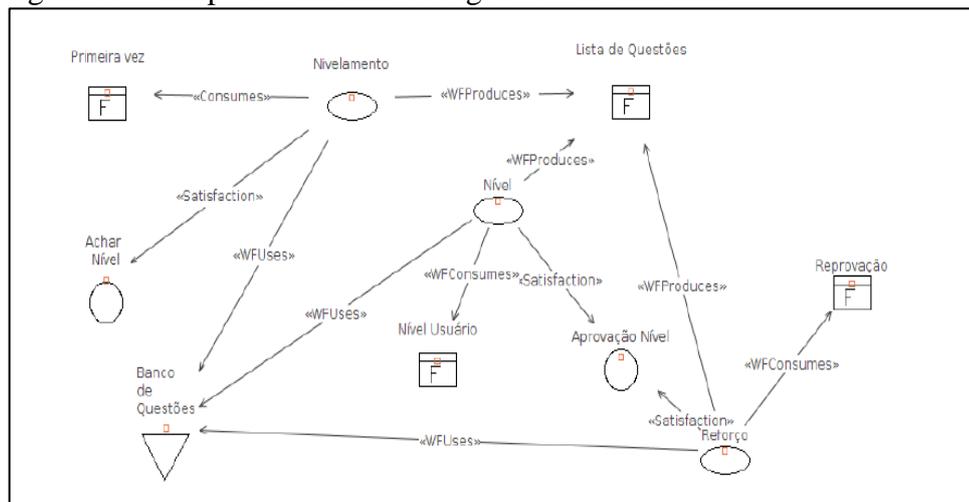
- Modelo de organização:

Figura 34 – Grupo 2 – Modelo de Organização - INGENIAS.



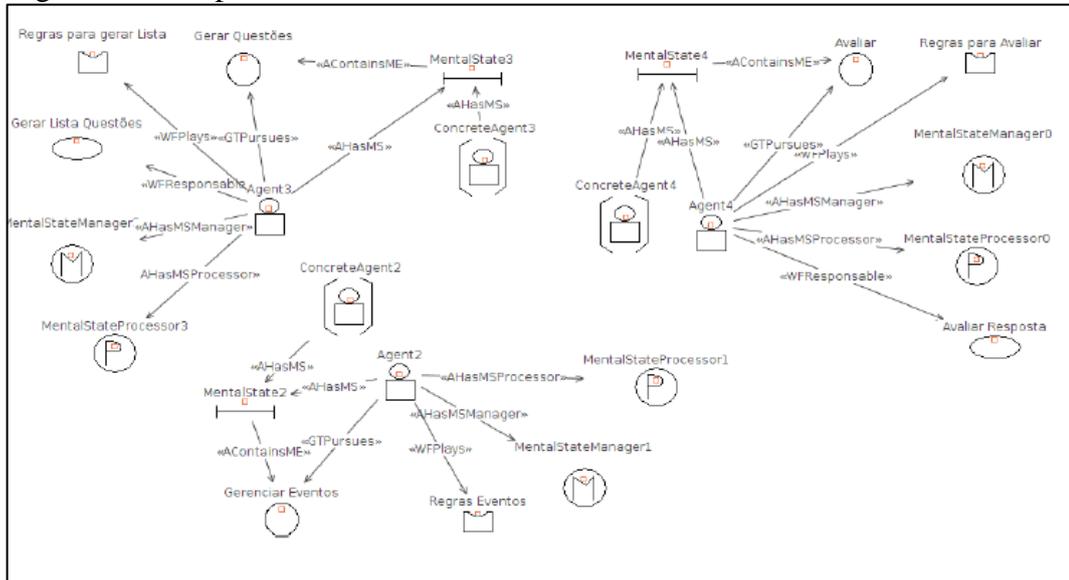
- Modelo de agentes:

Figura 35 – Grupo 2 – Modelo de Agentes - INGENIAS.



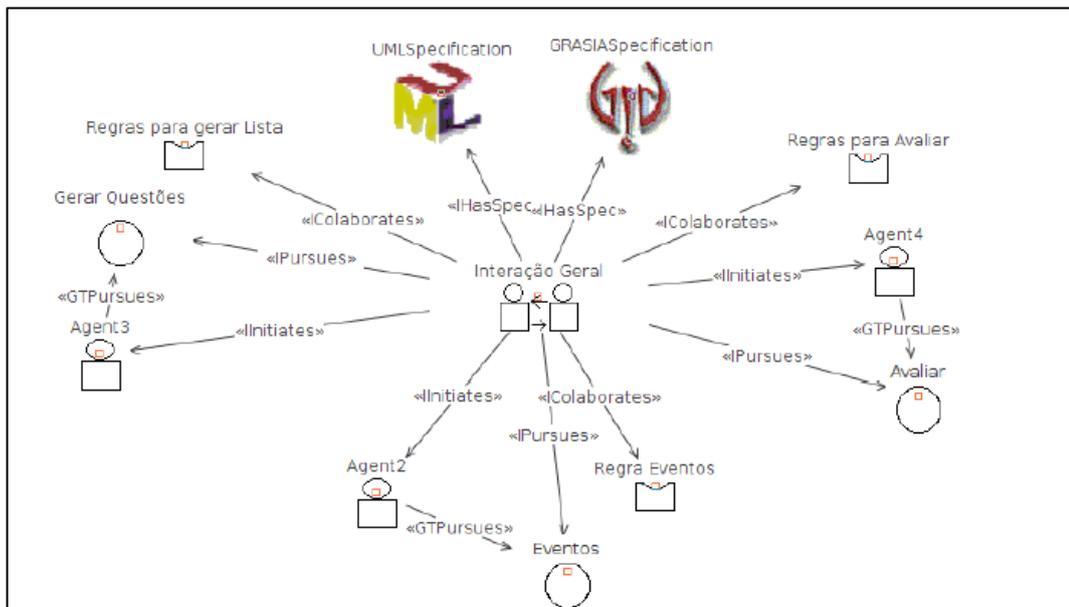
- Modelo de metas e tarefas:

Figura 36 – Grupo 2 – Modelo de Metas e Tarefas - INGENIAS.



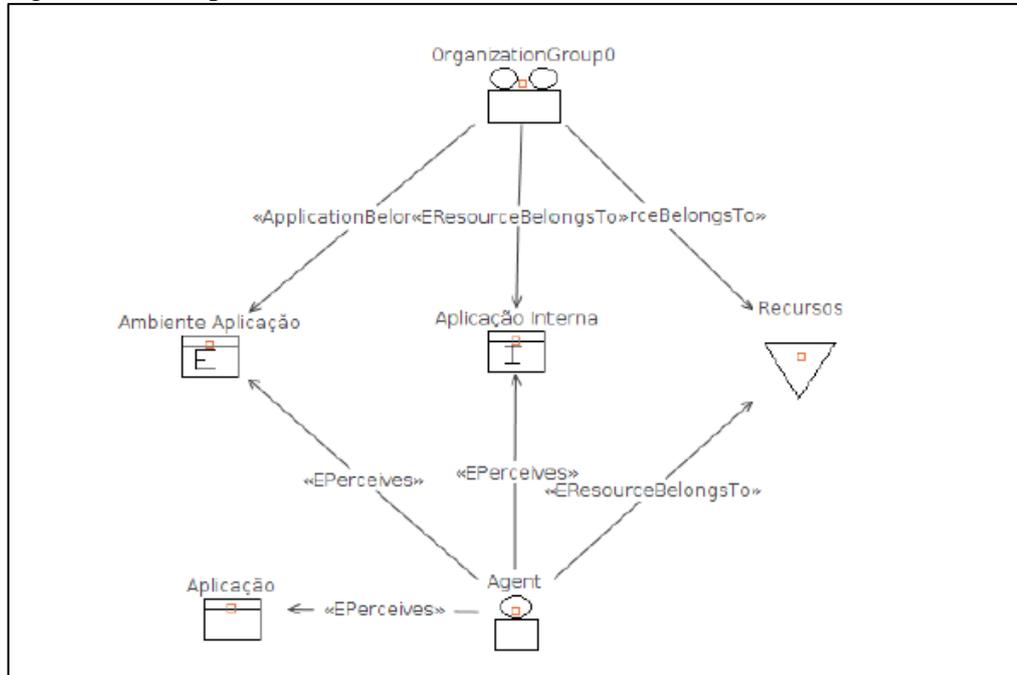
- Modelo de interação:

Figura 37 – Grupo 2 – Modelo de Interação - INGENIAS.



- Modelo de ambiente:

Figura 38 – Grupo 2 – Modelo de Ambiente - INGENIAS.



- Telas do jogo do Grupo 5 – INGENIAS

Figura 39 – Grupo 2 – Tela Inicial - Configurações - INGENIAS.

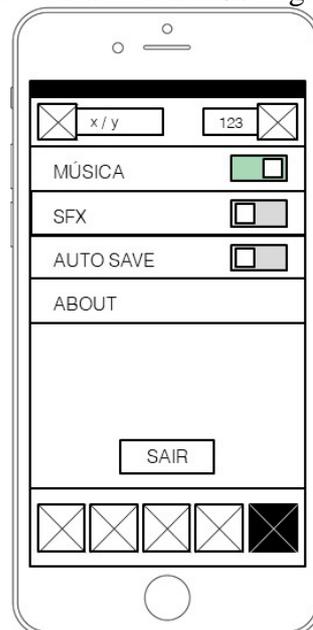


Figura 40 – Grupo 2 – Tela Inicial – Configurações (2) - INGENIAS.



Figura 41 – Grupo 2 – Tela de Nivelamento – INGENIAS.

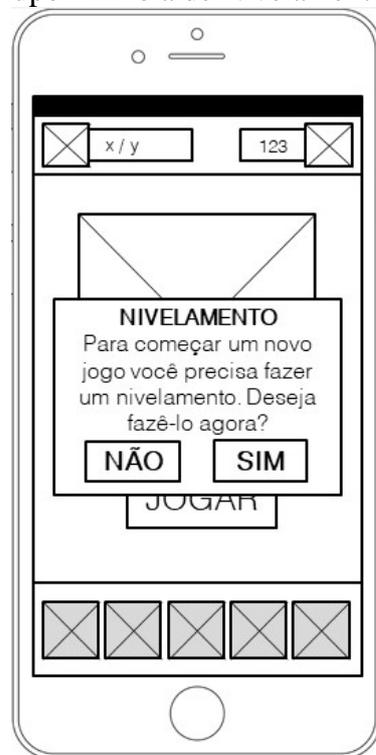


Figura 42 – Grupo 2 – Tela Atingir Nível – INGENIAS.

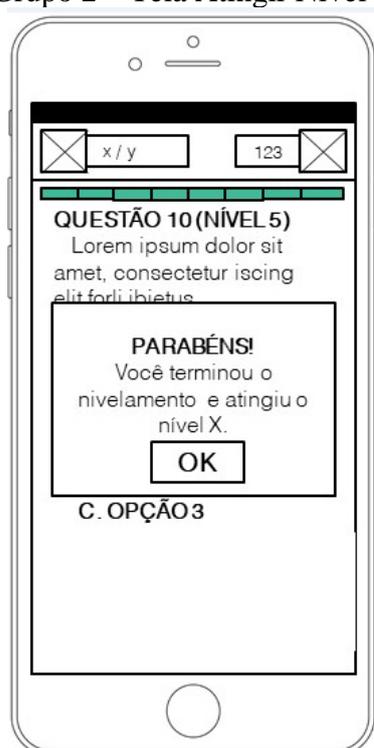


Figura 43 – Grupo 2 – Questão – Nível 1 - INGENIAS.

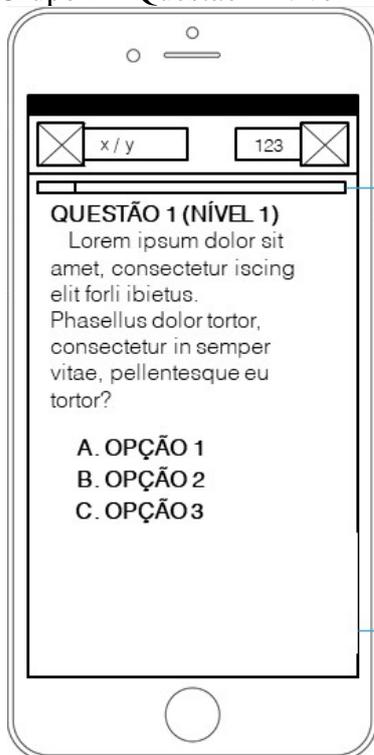


Figura 44 – Grupo 2 – Ganho de Equipamento – Nível 1 - INGENIAS.

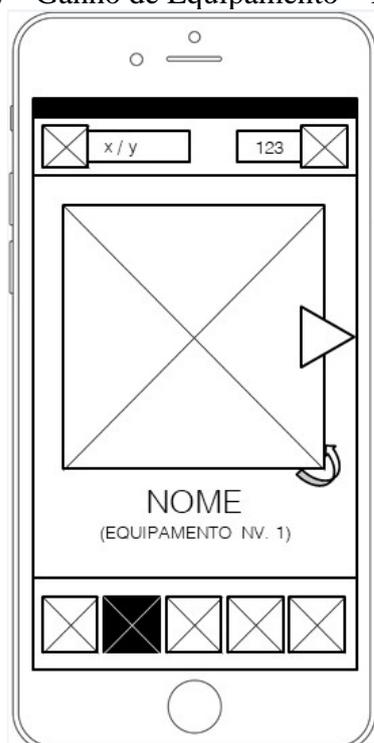


Figura 45 – Grupo 2 – Tela Reposta Correta – Nível 1 - INGENIAS.



Figura 46 – Grupo 2 – Tela Reposta Errada – Nível 1 - INGENIAS.

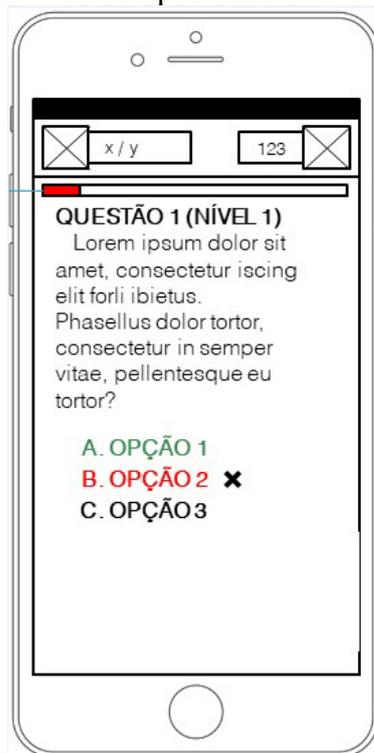


Figura 47 – Grupo 2 – Tela Reforço – Nível 1 - INGENIAS.



Figura 48 – Grupo 2 – Tela Aprovação– Nível 1 - INGENIAS.

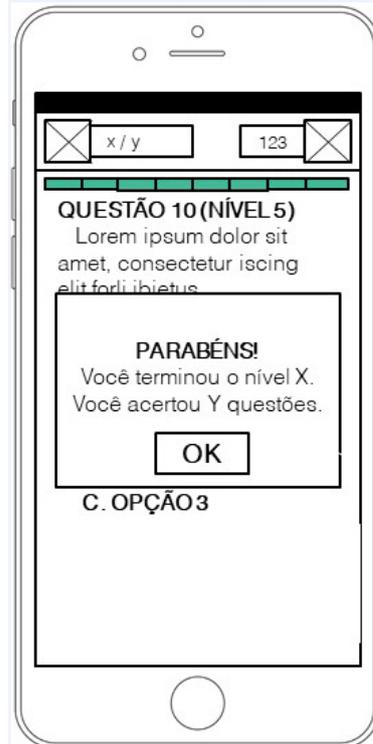


Figura 49 – Grupo 2 – Tela Histórico – Nível 1 – INGENIAS.



ANEXO D - Artefatos desenvolvidos pelo grupo 3 – AgilePASSI

- Diagramas de caso de uso UML (Funcionalidades):

Figura 50 – Grupo 3 – AgilePASSI – Diagrama de caso de uso UML - Funcionalidades.

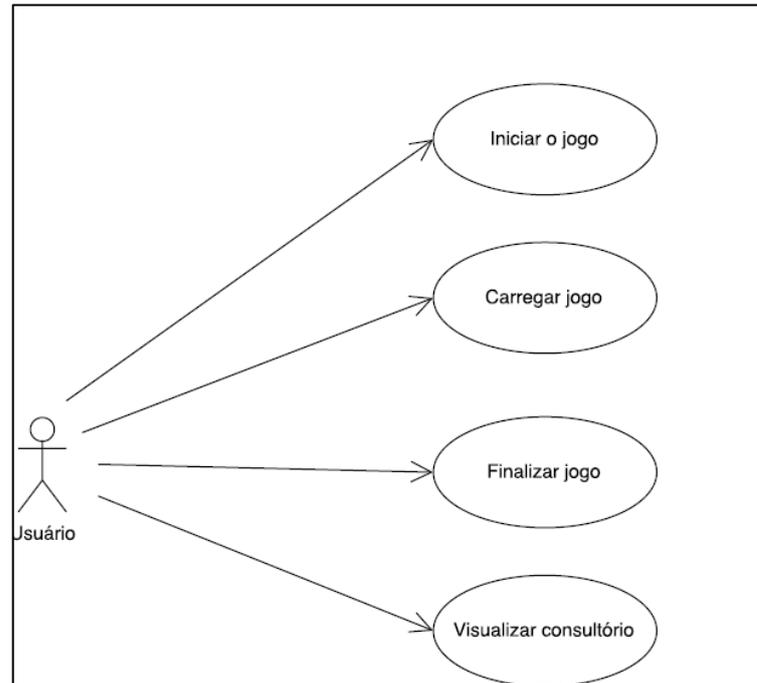
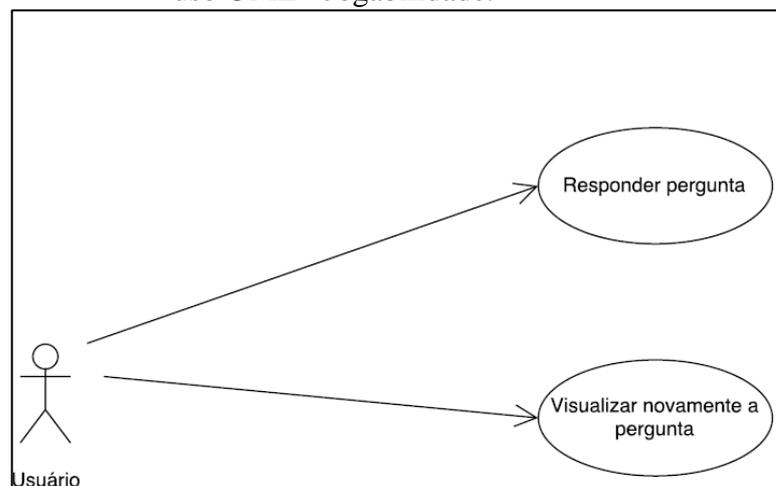


Figura 51 – Grupo 3 – AgilePASSI – Diagrama de caso de uso UML - Jogabilidade.



- Casos de Uso (Funcionalidades):

Quadro 29 – Grupo 3 – AgilePASSI – Detalhamento dos casos de uso (Funcionalidades).

UC01 - Iniciar o jogo / Objetivo: Iniciar um novo jogo. / Atores: Usuário / Pré-condições: - /

Fluxo principal:

1. Usuário seleciona a opção “Iniciar novo jogo” no menu principal.
2. Usuário preenche um formulário para definir seu perfil no jogo.
3. Usuário envia o formulário.
4. Usuário inicia o jogo.

Fluxo alternativo: - / Fluxo de exceção:

1. Algum campo do formulário é preenchido de maneira inválida.
2. Sistema informa o campo inválido para o usuário.
3. Usuário corrige o erro.
4. Usuário envia o formulário.
5. Usuário inicia o jogo.

UC02 - Carregar o jogo / Objetivo: Salvar o progresso de um usuário. / Atores: Usuário. / Pré-condições: Ter jogo salvo. / Fluxo principal:

1. Usuário seleciona a opção “Carregar jogo” no menu.
2. Usuário seleciona o respectivo arquivo de save.
3. Usuário carrega o jogo.

Fluxo alternativo:

1. Usuário seleciona um arquivo salvo.
2. Usuário seleciona a opção “Excluir”.
3. Usuário confirma o desejo de exclusão do arquivo.
4. Usuário exclui o save.

Fluxo de exceção: -

UC03 - Finalizar o jogo / Objetivo: Finalizar a sessão do jogo. / Atores: Usuário

Pré-condições: Iniciar uma sessão do jogo. / Fluxo principal:

1. Usuário seleciona a opção “Sair” no menu.
2. Usuário confirma que deseja finalizar a sessão do jogo.
3. Usuário finaliza a sessão do jogo.

Fluxo alternativo:

1. Usuário seleciona a opção “Sair” no menu.
2. Usuário não confirma o desejo de finalizar o jogo.
3. Usuário continua com a sessão do jogo.

Fluxo de exceção: -

UC04 - Visualizar consultório / Objetivo: Visualizar o consultório. / Atores: Usuário

Pré-condições: Iniciar uma sessão do jogo. / Fluxo principal:

1. Usuário seleciona a opção “Visualizar consultório” no menu.
2. Sistema exibe o consultório com seus respectivos itens para o usuário.
3. Usuário seleciona a opção “Voltar” para retornar ao menu principal.
4. Usuário retorna ao menu principal.

Fluxo alternativo:

1. Usuário seleciona a opção “Itens”.
2. Sistema exibe uma lista com todos os itens disponíveis (por nível) para o usuário.
3. Usuário seleciona a opção “Voltar” para retornar ao consultório.
4. Usuário retorna ao consultório.

Fluxo de exceção: -

- Casos de Uso (Jogabilidade):

Quadro 30 – Grupo 3 – AgilePASSI – Detalhamento dos casos de uso (Jogabilidade).

UC01 - Responder pergunta / Objetivo: Responder uma pergunta. / Atores: Usuário
Pré-condições: Iniciar uma sessão do jogo.

Fluxo principal:

1. Usuário seleciona uma das opções de resposta.
2. Usuário envia a resposta.
3. Sistema exibe feedback sobre a resposta.
4. Usuário seleciona a opção “Próxima”.
5. Sistema exibe próxima pergunta para o usuário.

Fluxo alternativo: -

Fluxo de exceção: -

UC02 - Visualizar novamente a pergunta / Objetivo: Visualizar novamente uma pergunta. / Atores: Usuário

Pré-condições: Iniciar uma sessão do jogo.

Fluxo principal:

1. Usuário seleciona um nível já completado.
2. Sistema exibe lista de perguntas respondidas.
3. Usuário seleciona uma pergunta.
4. Sistema exibe a pergunta e seu gabarito.
5. Usuário visualiza a pergunta e seu gabarito.
6. Usuário seleciona a opção “Voltar” para retornar a lista de perguntas.

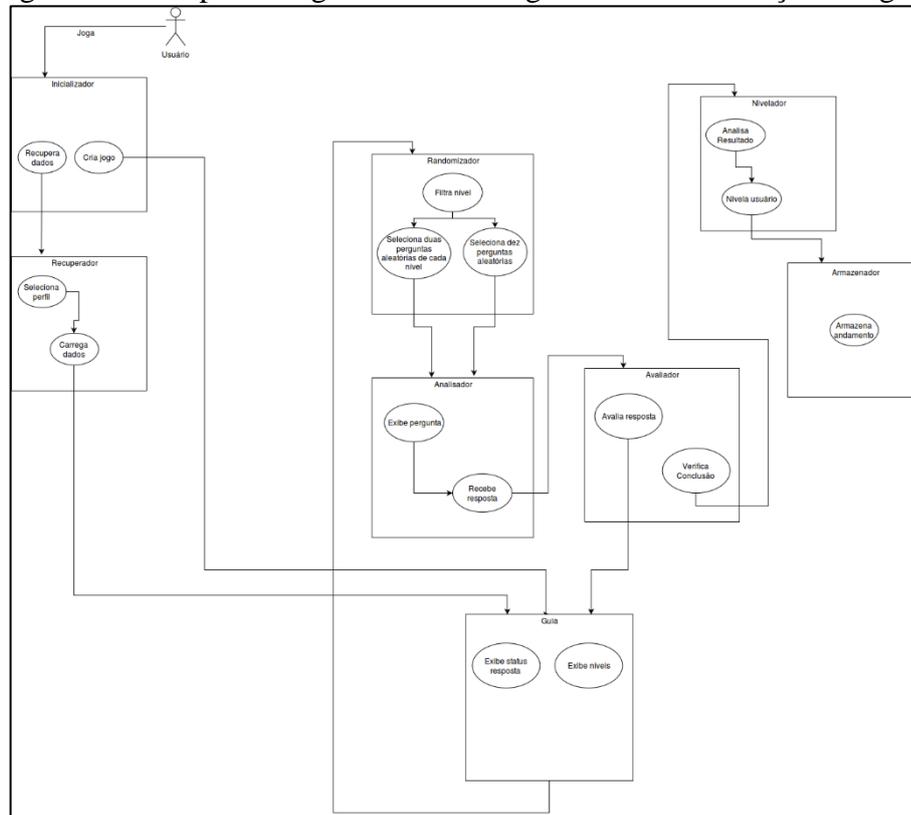
Fluxo alternativo: -

Fluxo de exceção:

1. Usuário seleciona um nível ainda não completado.
2. Sistema exibe um erro para o usuário

- Diagrama de identificação de agentes:

Figura 52 – Grupo 3 – AgilePASSI – Diagrama de identificação de agentes.



- Lista de Requisitos Funcionais:

Quadro 31 – Grupo 3 – AgilePASSI – Detalhamento dos requisitos funcionais.

1	O sistema deve apresentar perguntas ao usuário.
2	O sistema deve dar três opções de resposta para cada pergunta.
3	O sistema deve informar quais módulos estão disponíveis para acesso até o momento presente.
4	O sistema deve indicar ao usuário se a resposta dele está correta.
5	O sistema deve indicar ao usuário qual a opção correta em caso de erro.
6	O sistema deve apresentar ao usuário uma justificativa para a resposta está incorreta.
7	O sistema deve apresentar uma barra de status para as respostas durante os níveis.
8	O sistema deve indicar com a cor verde as respostas certas durante o nível atual.
9	O sistema deve indicar com a cor vermelha as respostas erradas durante o nível atual.
10	O sistema deve liberar o nível seguinte caso o usuário obtenha 100% de acerto de um nível, durante o nivelamento.
11	O sistema deve informar a situação do usuário ao final do nivelamento.
12	O sistema deve apresentar 10 questões para cada nível.
13	O sistema deve liberar o nível seguinte caso o usuário obtenha pelo menos 70% de acerto de um nível.
14	O sistema deve enviar o Equipamento de mesma numeração do nível concluído para o consultório, após conclusão do nível.
15	O sistema deve reiniciar o nivelamento caso o usuário o interrompa.
16	O sistema deve informar ao usuário, ao final do jogo, sua pontuação obtida em cada nível e sua média dos níveis.
17	O sistema deve iniciar o reforço relativo ao nível, caso o usuário obtenha menos de 70% de acertos no mesmo.
18	O sistema deve apresentar 10 perguntas referentes ao nível durante o reforço do mesmo.
19	O sistema deve repetir o reforço 1 até que o usuário obtenha pelo menos 70% de acertos.
20	O sistema deve permitir que o usuário salve o progresso do jogo.
21	O sistema deve retornar ao nível 1 caso o usuário atinja menos de 70% mais de duas vezes em qualquer nível.

- Lista de Requisitos Não Funcionais:

Quadro 328 – Grupo 3 – AgilePASSI – Detalhamento dos requisitos não funcionais.

1	O sistema deve exibir as questões de forma mesclada: áudio, vídeo, imagem e texto.
2	O sistema deve ter um limite de 5 minutos para que o usuário entre com a sua resposta, caso o tempo acabe a questão será considerada errada.

- Telas do jogo do grupo 3 – AgilePASSI

Figura 53 – Grupo 3 – AgilePASSI – Tela Inicial.

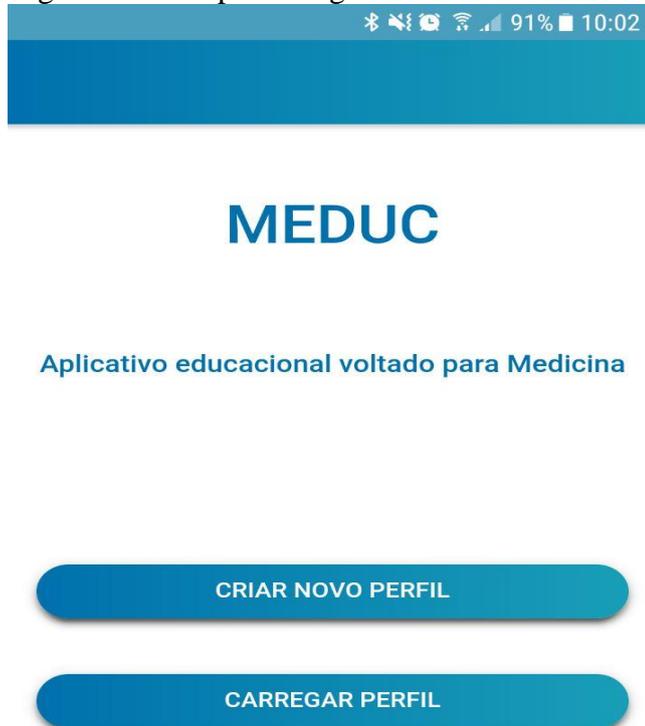


Figura 547 – Grupo 3 – AgilePASSI – Cadastro Usuário.

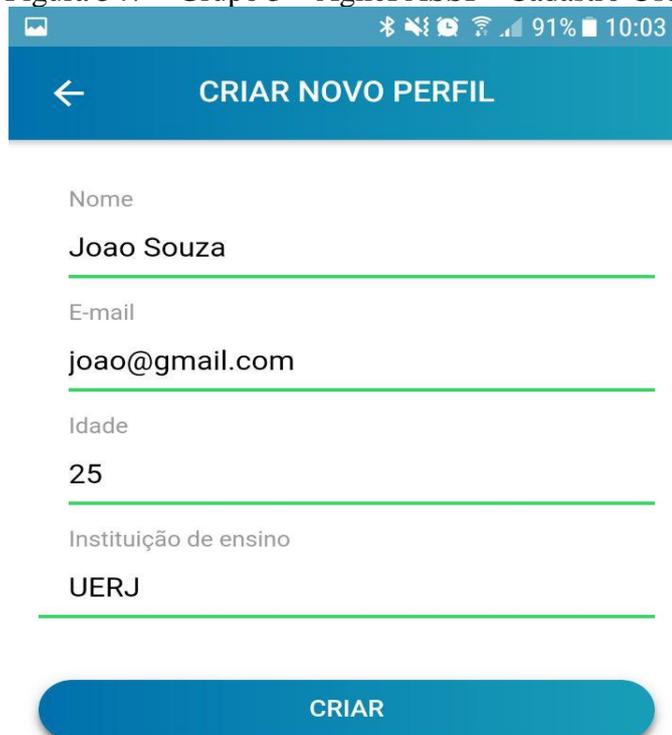


Figura 558 – Grupo 3 – AgilePASSI – Visão Consultório - Nivelamento.



Figura 56 – Grupo 3 – AgilePASSI – Nivelamento.

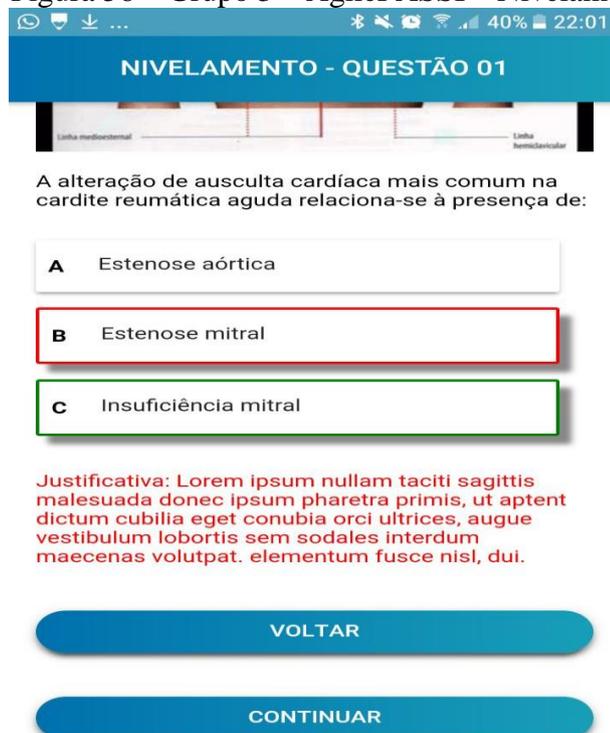


Figura 57 – Grupo 3 – AgilePASSI – Histórico Nivelamento.



Figura 58 – Grupo 3 – AgilePASSI – Ganho de equipamento.

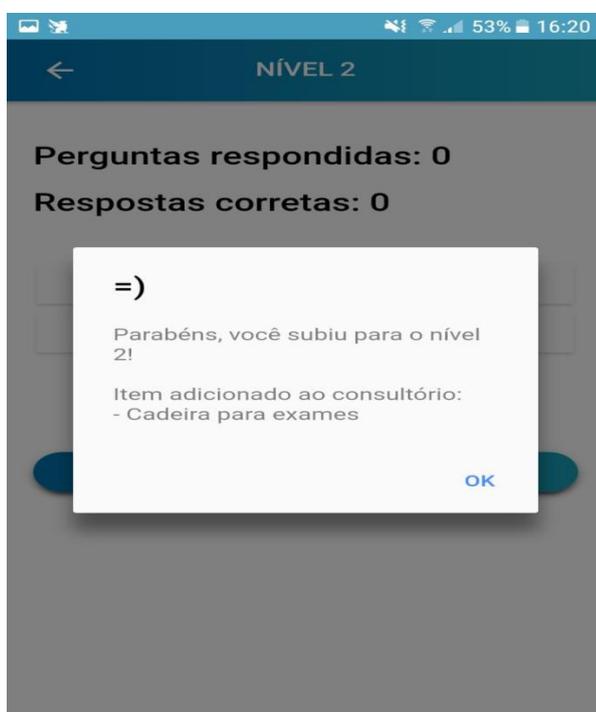
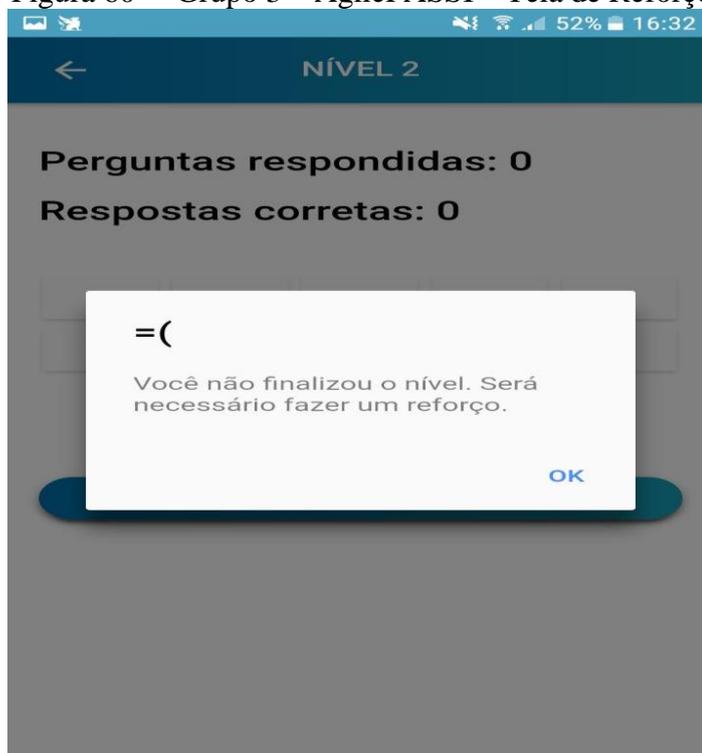


Figura 59 – Grupo 3 – AgilePASSI – Consultório Completo.



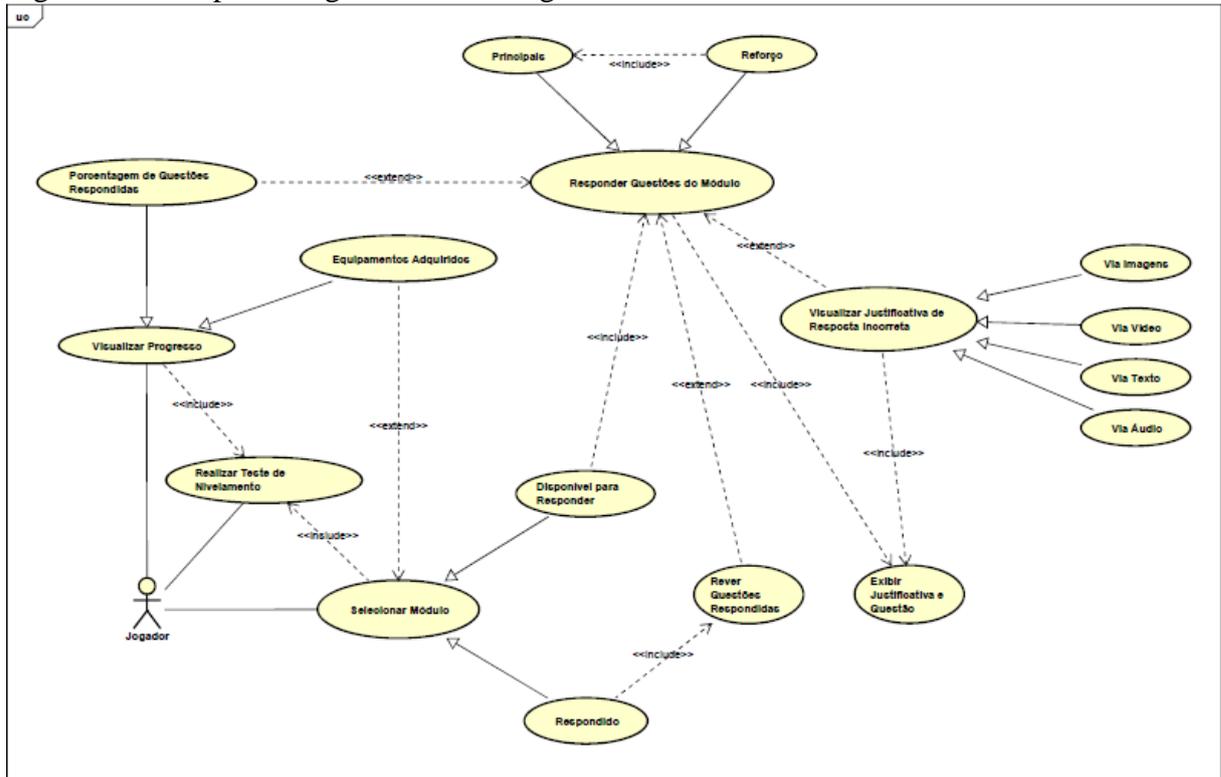
Figura 60 – Grupo 3 – AgilePASSI – Tela de Reforço



ANEXO E - Artefatos desenvolvidos pelo grupo 4 – AgilePASSI.

- Diagramas de caso de uso UML:

Figura 61 – Grupo 4 – AgilePASSI – Diagrama de caso de uso UML.

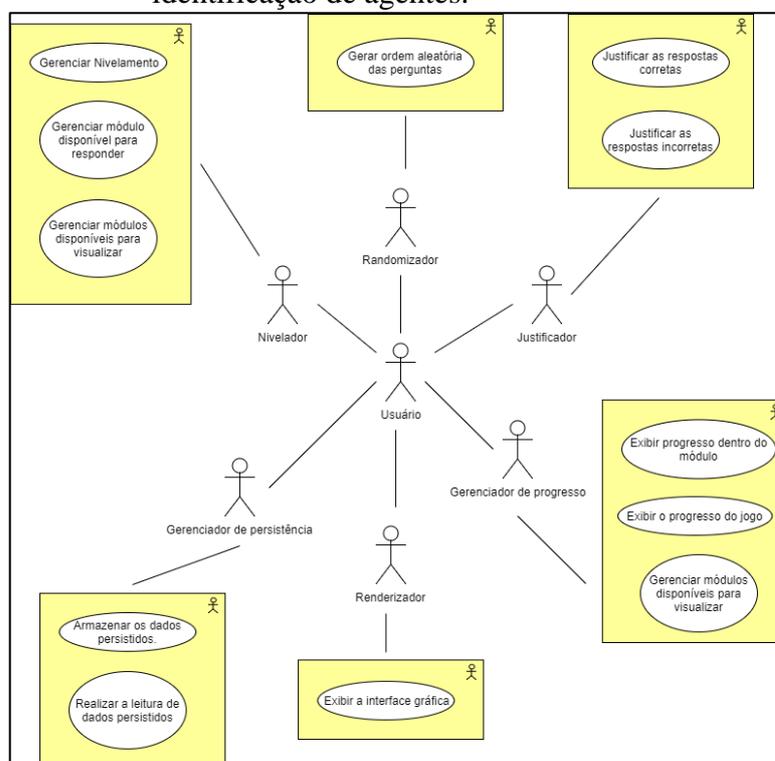


Quadro 33 – Grupo 4 – AgilePASSI – Detalhamento dos casos de uso.

<ul style="list-style-type: none">● Realizar teste de nivelamento<p>Ao iniciar o jogo, o jogador será levado a realizar o teste de nivelamento, que após terminado dirá em que módulo ele estará apto a começar. A partir disso, ele será capaz de visualizar seu progresso ou selecionar o próximo módulo de perguntas.</p>● Visualizar progresso<p>O jogador poderá visualizar seu progresso a partir do momento em que tiver o teste de nivelamento realizado. O progresso pode ser visto de duas maneiras: dentro de um módulo, através das porcentagens de questões daquele módulo que já foram respondidas, ou através dos equipamentos adquiridos, cuja quantidade é a mesma de módulos concluídos. Somente será possível verificar a porcentagem das questões respondidas quando o jogador tiver selecionado um módulo</p>● Selecionar módulo<p>O jogador poderá selecionar um módulo a partir do momento em que tiver o teste de nivelamento realizado. A qualquer momento, na tela inicial, ele será capaz de observar seu progresso nos módulos do jogo através da quantidade de equipamentos adquiridos que estiver sendo exibida. Um módulo pode estar disponível para responder ou respondido.</p>● Responder questões do módulo<p>Tendo selecionado o módulo disponível para ser respondido, o jogador irá responder às questões principais. Se não obtiver a pontuação necessária para avançar para o próximo módulo, deverá responder às questões de reforço.</p>● Rever questões respondidas<p>Tendo selecionado um módulo previamente respondido, o jogador poderá rever as questões respondidas e o seu resultado nelas.</p>● Visualizar justificativa de resposta incorreta<p>Após responder uma questão, se a resposta estiver incorreta, o usuário irá visualizar a justificativa da opção marcada estar incorreta. Essa justificativa pode ser dada via texto, via imagens, via vídeo ou via áudio.</p>● Exibir justificativa e questão<p>Após responder uma questão, a justificativa da resposta correta será exibida para o jogador, independente dele ter acertado ou errado a mesma.</p>

- Diagrama de identificação de agentes:

Figura 62 – Grupo 4 – AgilePASSI – Diagrama de identificação de agentes.



- Lista de Requisitos Funcionais:

Quadro 349 – Grupo 4 – AgilePASSI – Detalhamento dos requisitos funcionais (continua).

1	O jogador deve iniciar o jogo em um consultório com mobília básica.
2	O jogo deve ter 11 módulos: 5 níveis principais com 1 nível de reforço para cada e 1 nivelamento.
3	Cada nível deve ter 10 questões (incluindo nivelamento e reforço).
4	Cada questão deve ter 3 opções de resposta, em que apenas uma é correta.
5	O usuário deve conseguir verificar na página principal quais módulos estão disponíveis para acesso.
6	O usuário deve ter disponível somente um módulo.
7	O usuário deve ser informado quando o resultado de um módulo for positivo ou negativo.
8	As questões devem utilizar os seguintes formatos: áudio, vídeo, imagens e texto.
9	Ao escolher uma opção incorreta, o usuário deve ser informado da opção correta.

Quadro 34 – Grupo 4 – AgilePASSI – Detalhamento dos requisitos funcionais (continuação).

10	Ao escolher uma opção incorreta, o usuário deve receber um retorno com uma justificativa relatando o porquê da opção escolhida estar incorreta em um dos seguintes formatos: áudio, vídeo, imagens e texto.
11	O jogo deve recuperar as perguntas mais fáceis para o primeiro nível, tanto no nível principal quanto no reforço, aumentando a dificuldade a cada nível.
12	As questões disponibilizadas têm níveis de conhecimento de 1 a 5, sendo as de nível 1 as mais fáceis e as de nível 5 as mais difíceis.
13	O jogo deve permitir que o usuário acompanhe a sua evolução com uma barra de status. Para acertos, o usuário vê um indicativo verde na barra de status. Enquanto que para erros, o usuário vê um indicativo vermelho.
14	O jogo deve permitir que o usuário consiga rever questões que já foram respondidas, a resposta e o retorno. Entretanto, o usuário não consegue alterar a opção selecionada.
15	Ao final do jogo o usuário deve receber a pontuação obtida em cada nível e sua média dos níveis.
16	Ao entrar no jogo pela primeira vez, o usuário deve realizar o nivelamento e as demais funcionalidades estarão indisponíveis.
17	O nivelamento deve possuir 10 questões com todos os níveis disponíveis no jogo, sendo 2 de cada nível distribuídas aleatoriamente.
18	O nivelamento deve permitir ao jogador iniciar em um nível adequado ao seu conhecimento. O nível redirecionado será o nível da pergunta que houve o primeiro erro no nivelamento. Por exemplo, caso o usuário acerte as questões de nível 1, 2 e 4, mas somente acertou uma do nível 3, o jogo deverá habilitar o nível 3 para o jogador.
19	O usuário deve realizar o Nível 5, mesmo que acerte todas as questões.
20	Ao fim do nivelamento o jogador deve receber um retorno do nível alcançado.
21	Ao fim do nivelamento o jogador deve receber todos os equipamentos correspondentes aos níveis anteriores e ao nível atingido.
22	Após finalizar o nivelamento, o usuário é direcionado para a página principal do jogo e poderá iniciar o nível alcançado no nivelamento.
23	Caso o usuário interrompa o nivelamento, ele deve iniciar o processo novamente e as respostas anteriores serão desconsideradas.
24	Durante os níveis 1, 2, 3, 4 e 5 o usuário deve responder 10 questões referente ao nível de conhecimento 1, 2, 3, 4 e 5, respectivamente.
25	A cada nível concluído, o consultório do jogador deve receber um novo equipamento.
26	Ao final de cada nível, caso o usuário obtenha 70% de acertos ou mais, ele está apto a avançar para o próximo nível.
27	Caso o usuário obtenha resultado inferior de 70% de acertos, o nível em questão é bloqueado e será necessário realizar o reforço a fim de realizar o desbloqueio.
28	Ao fim de cada nível concluído, o usuário deve receber um retorno informando o sucesso na conclusão do nível e para qual próximo nível ele é direcionado.
29	Caso o usuário realize um dos níveis mais de duas vezes e neste tempo não obtenha mais de 70% de acertos, o mesmo será direcionado novamente para o nível 1.
30	Ao fim de cada nível o jogador deve receber um equipamento correspondente ao nível concluído.
31	Após finalizar os níveis 1, 2, 3 ou 4, o usuário é direcionado para a página principal do jogo e está apto para avançar para o próximo nível.

Quadro 34 – Grupo 4 – AgilePASSI – Detalhamento dos requisitos funcionais (conclusão).

32	No início de cada nível o usuário é direcionado para a primeira questão aleatória do nível em questão.
33	Caso o usuário interrompa o jogo em qualquer um dos níveis disponíveis, ele deve iniciar o processo novamente e as respostas anteriores serão armazenadas.
34	O jogo deve terminar quando todos os níveis estiverem concluídos e, consequentemente, o consultório estiver completamente equipado.
35	O jogo deve possuir um reforço para cada nível.
36	O reforço possui 10 perguntas com grau de conhecimento do nível do reforço.
37	Para concluir o reforço o jogador deve obter 70% de acertos e assim o nível referente a aquele nível de reforço é novamente liberado. Caso obtenha abaixo de 70%, ele deve realizar o reforço novamente.

Fonte: A autora, 2018.

- Lista de Requisitos Não Funcionais:

Quadro 35 – Grupo 6 – AgilePASSI – Detalhamento dos requisitos não funcionais.

1	O sistema deve garantir a privacidade dos dados, ou seja, não devem ser acessíveis por terceiros.
2	Todas as senhas devem ser salvas no banco de dados criptografadas.
3	O sistema deve ser simples e intuitivo. O usuário deve ser capaz de ter um bom entendimento de forma rápida do que é possível ser realizado em cada cenário da aplicação.
4	O sistema deve estar preparado para lidar com eventos inesperados.
5	O sistema deve, sempre, manter a integridade dos dados. Ou seja, os dados devem ser consistentes, mesmo em situações inesperadas.
6	O sistema deve atender às normas da ABNT.
7	O sistema não deve demorar mais de X segundos para processar uma requisição interna, considerando um dispositivo em condições adequadas para uso da aplicação.
8	O sistema deve ser desenvolvido na linguagem X.
9	O sistema deve se comunicar com banco de dados X.
10	O sistema deve ser executado em dispositivos Android ou IOS.

- Telas do jogo do grupo 4 – AgilePASSI

Figura 63 – Grupo 4 – AgilePASSI – Tela Inicial.

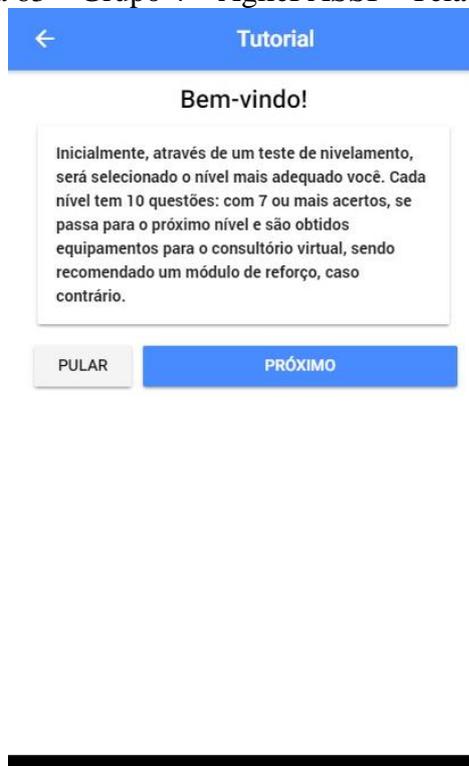


Figura 649 – Grupo 4 – AgilePASSI – Tela Inicial (2).



Figura 65 – Grupo 4 – AgilePASSI – Tela Nível 1.

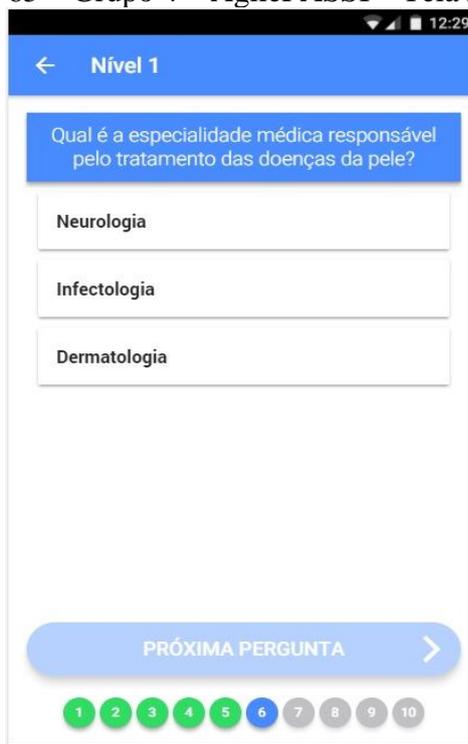


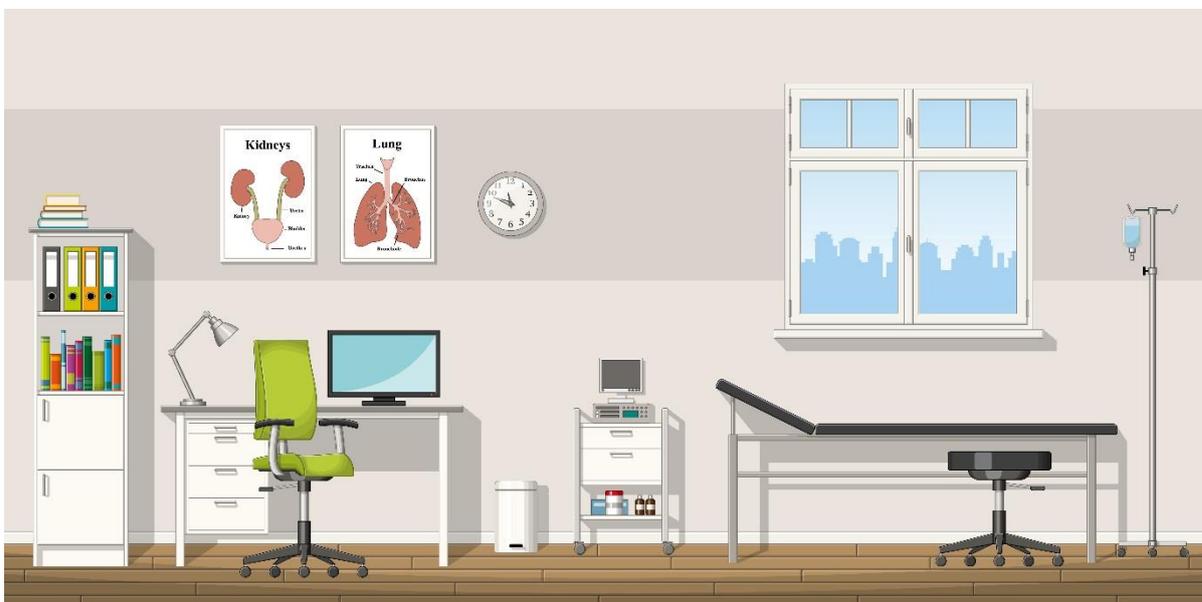
Figura 66 – Grupo 4 – AgilePASSI – Histórico Usuário



Figura 67 – Grupo 4 – AgilePASSI – Tela de Pergunta.



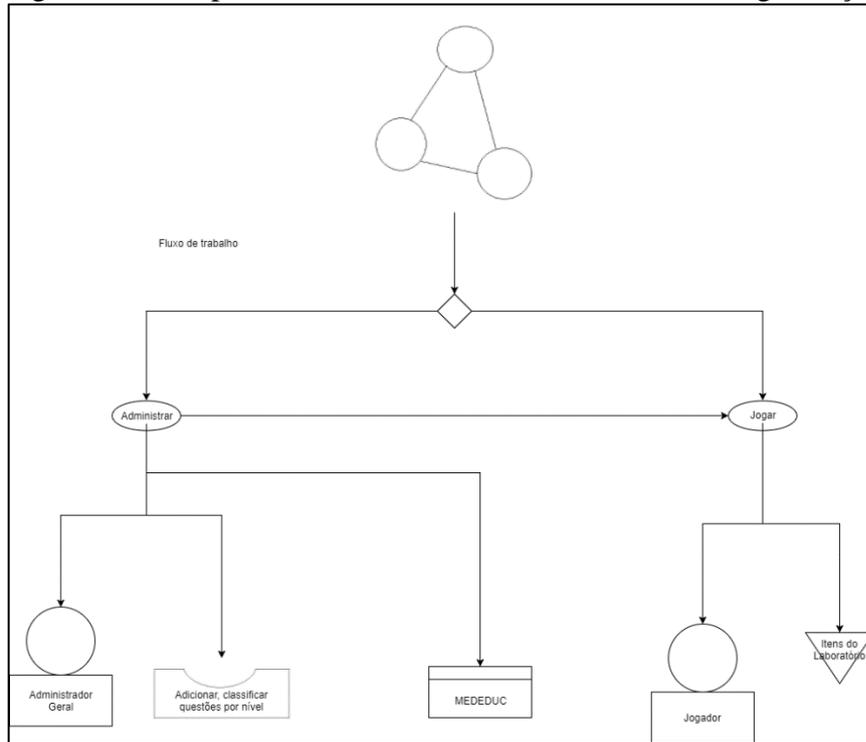
Figura 68 – Grupo 4 – AgilePASSI – Tela Consultório Completo.



ANEXO F- Artefatos desenvolvidos pelo grupo 5 – INGENIAS Scrum.

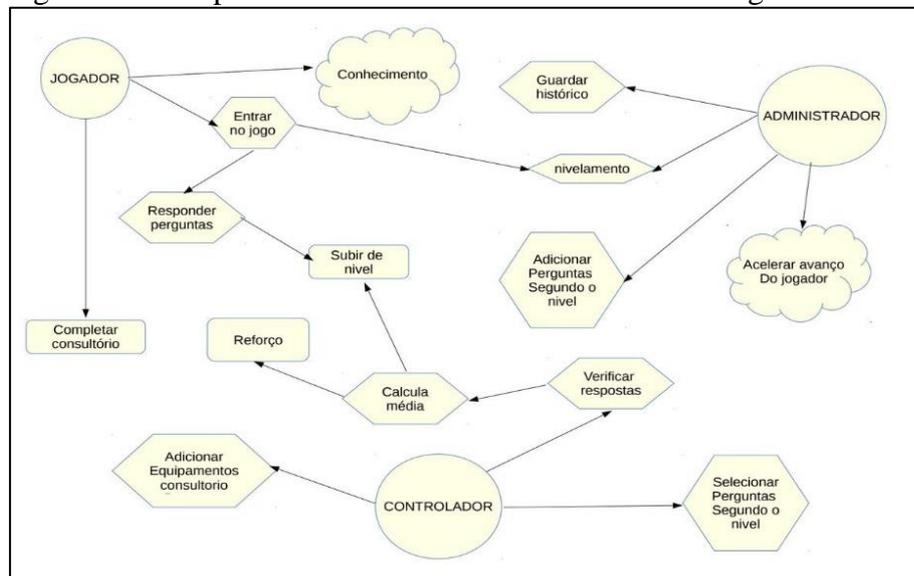
- Modelo de organização:

Figura 69 – Grupo 5 – INGENIAS Scrum – Modelo de Organização.



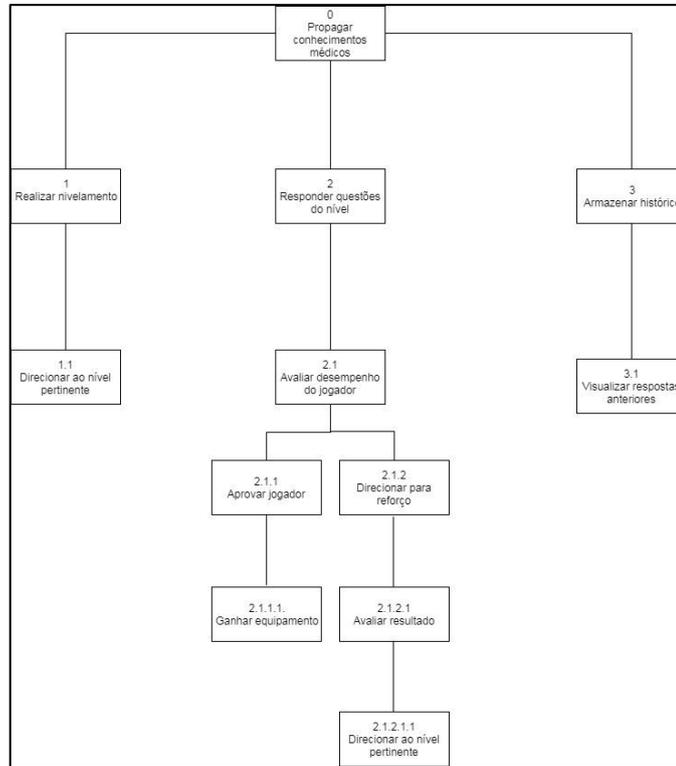
- Modelo de agentes:

Figura 70 – Grupo 5 – INGENIAS Scrum – Modelo de Agentes.



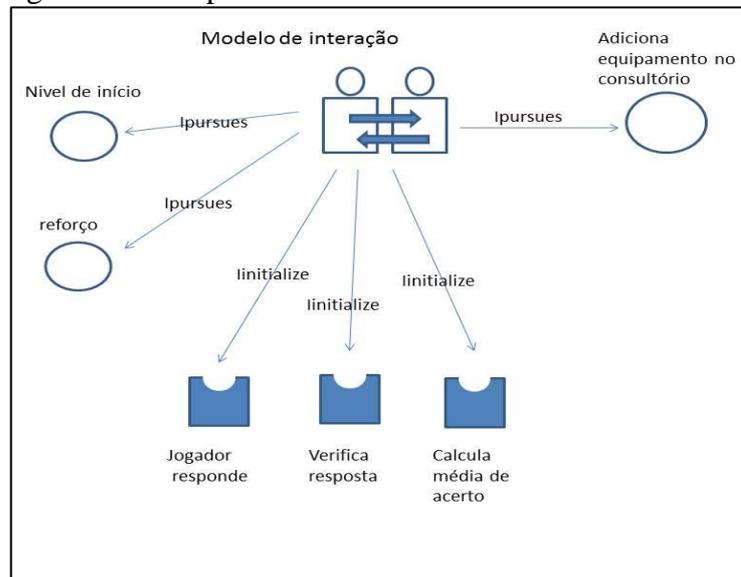
- Modelo de metas e tarefas:

Figura 71 – Grupo 5 – INGENIAS Scrum – Modelo de Metas e Tarefas.



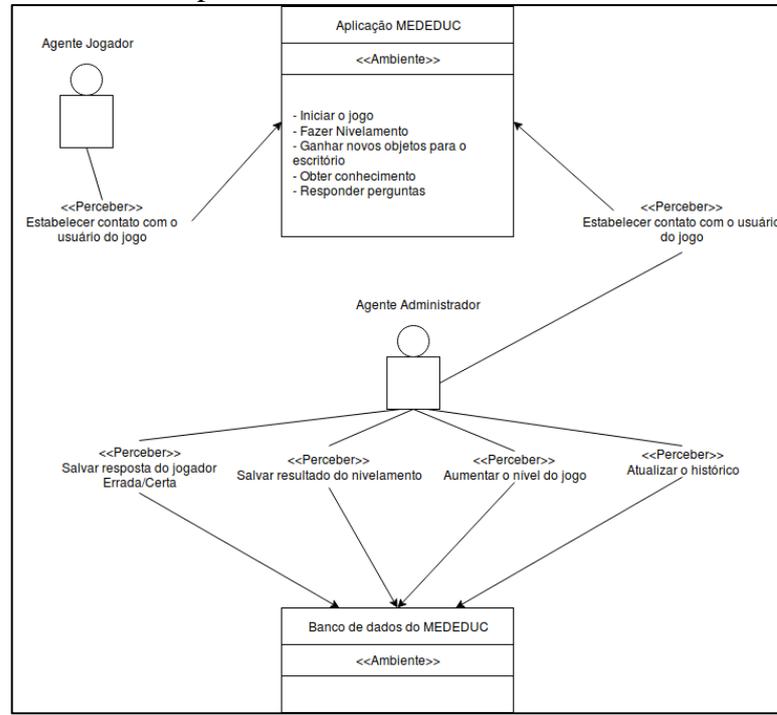
- Modelo de interação:

Figura 72 – Grupo 5 – INGENIAS Scrum – Modelo de Interação.



- Modelo de ambiente:

Figura 7310 – Grupo 5 – INGENIAS Scrum – Modelo de Ambiente.



- 1ª versão do *backlog*:

Quadro 36 – Grupo 5 – INGENIAS Scrum – 1ª versão do *backlog*.

1	Definir plataforma
2	Criar tela de perguntas
3	Criar metamodelo de organização
4	Criar modelo de casos de uso

- 2ª versão do *backlog*:

Quadro 37 – Grupo 5 – INGENIAS Scrum – 2ª versão do *backlog*.

1	Criar tela de home
2	Criar tela de introdução
3	Criar tela de nivelamento
4	Criar tela de perguntas (níveis)
5	Criar tela de reforço
6	Criar tela de prêmio por mudança de nível
7	Criar banco de dados das perguntas
8	Criar metamodelo de Agentes
9	Criar metamodelo de metas e tarefas

- 3ª versão do *backlog*:

Quadro 3810 – Grupo 5 – INGENIAS Scrum – 3ª versão do *backlog*

1	Criar perguntas em áudio, imagem e texto
2	Criar tela de introdução
3	Criar tela de histórico do usuário
4	Adequar barra de status para cada nível
5	Criar metamodelo de interação

- 4ª versão do *backlog*:

Quadro 39 – Grupo 5 – INGENIAS Scrum – 4ª versão do *backlog*.

1	Criar tela final do consultório completo
2	Manter telas e metamodelos atualizados na plataforma Trello.
3	Criar metamodelo de ambiente

- Telas do jogo do grupo 5 – INGENIAS Scrum

Figura 74 – Grupo 5 – INGENIAS Scrum – Tela Inicial.

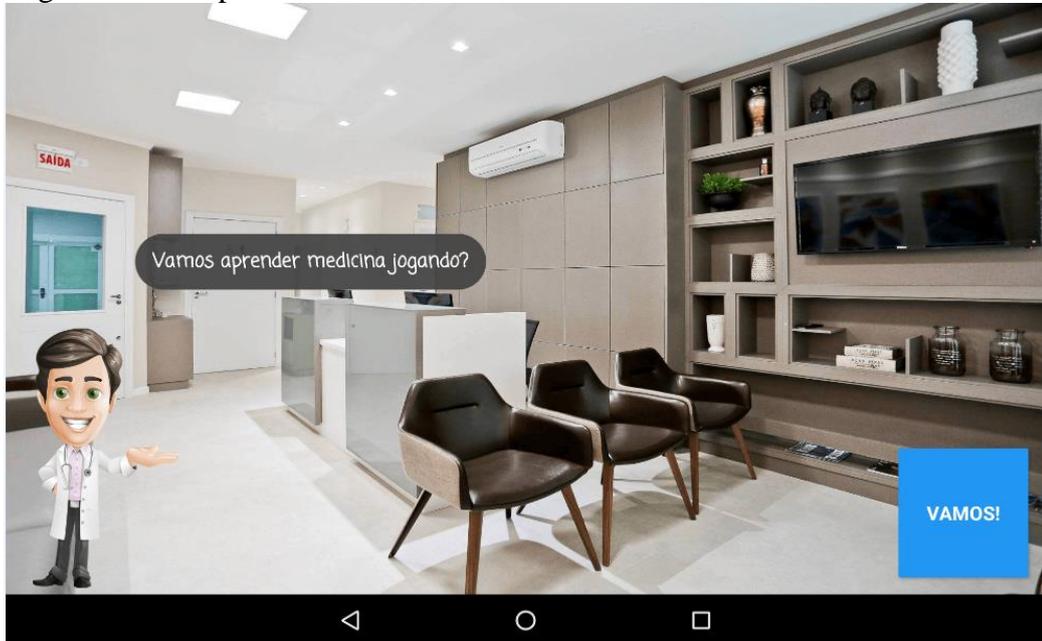


Figura 75 – Grupo 5 – INGENIAS Scrum – Tela Nível.



Figura 76 – Grupo 5 – INGENIAS Scrum – Tela Passagem para Nível 2.

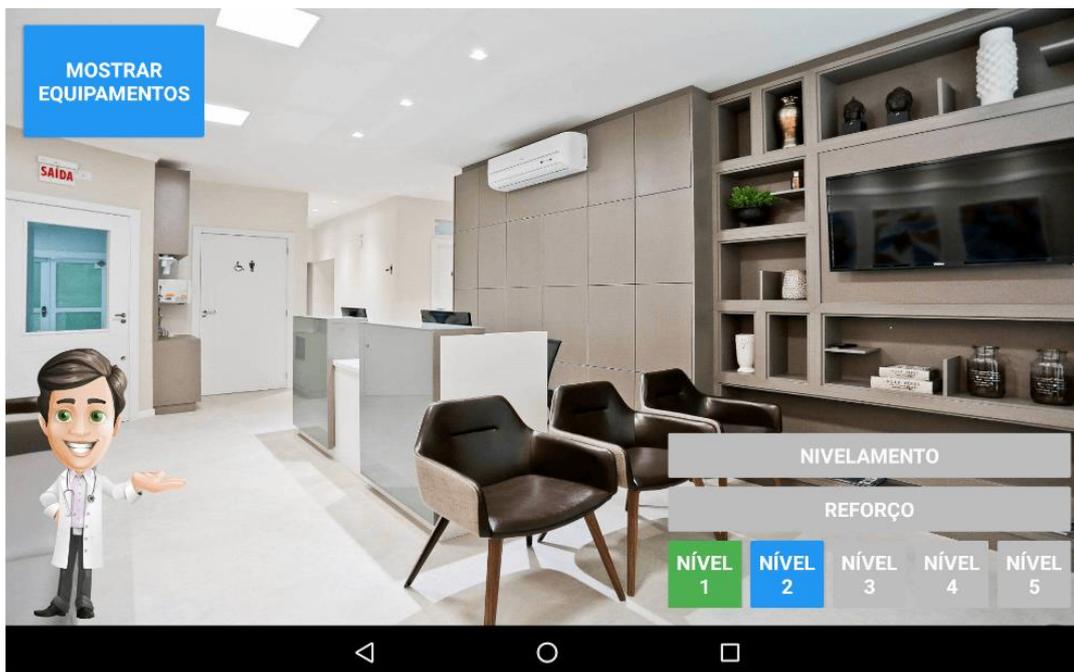


Figura 77 – Grupo 5 – INGENIAS Scrum – Tela Ganho de Equipamento.



Figura 78 – Grupo 5 – INGENIAS Scrum – Tela Pergunta Tipo Áudio.

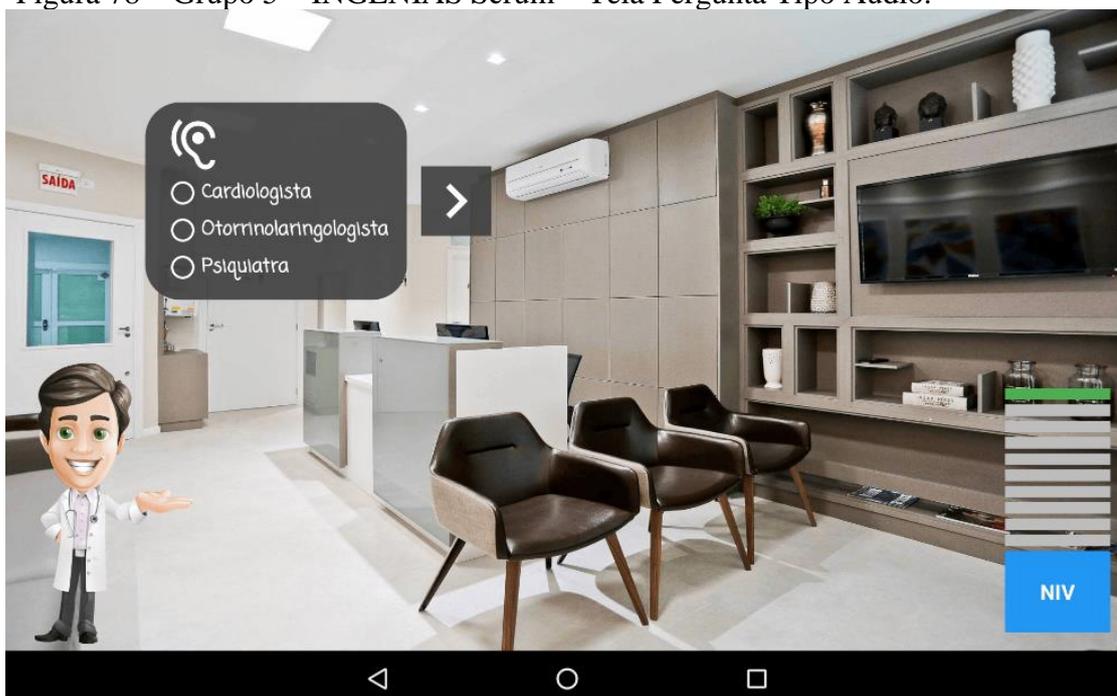


Figura 79 – Grupo 5 – INGENIAS Scrum – Tela Pergunta Tipo Imagem.

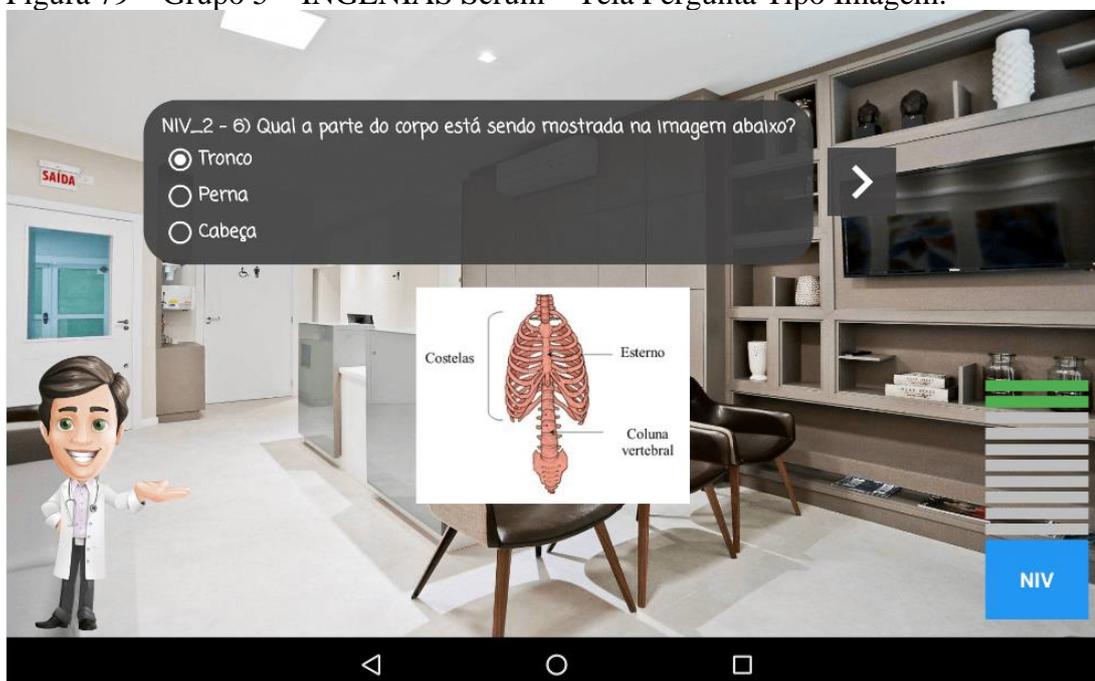


Figura 80 – Grupo 5 – INGENIAS Scrum – Tela Reforço.

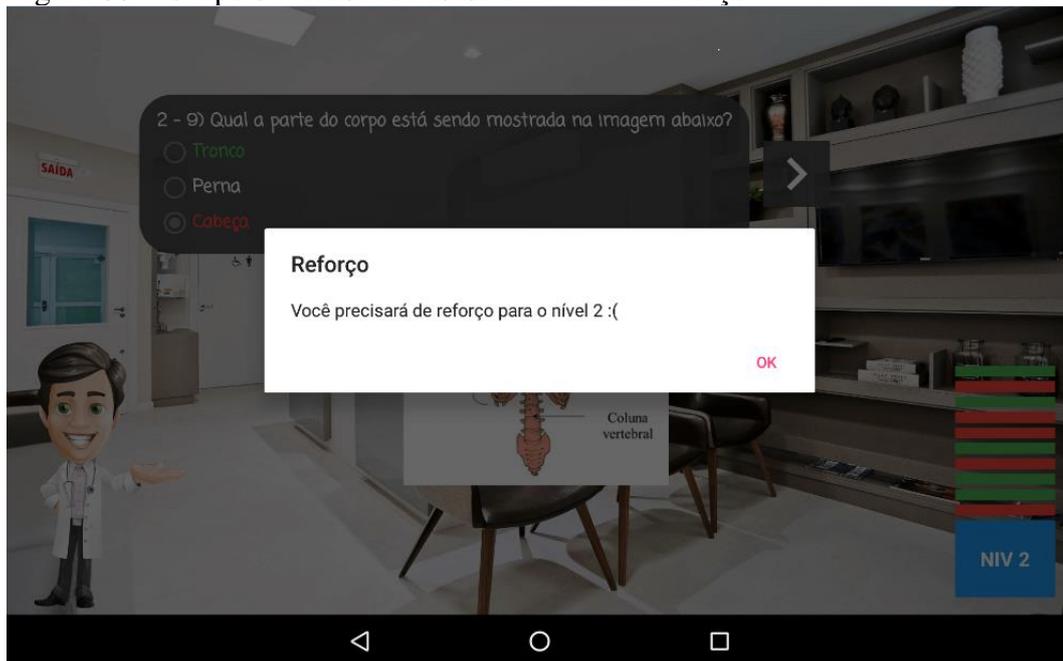
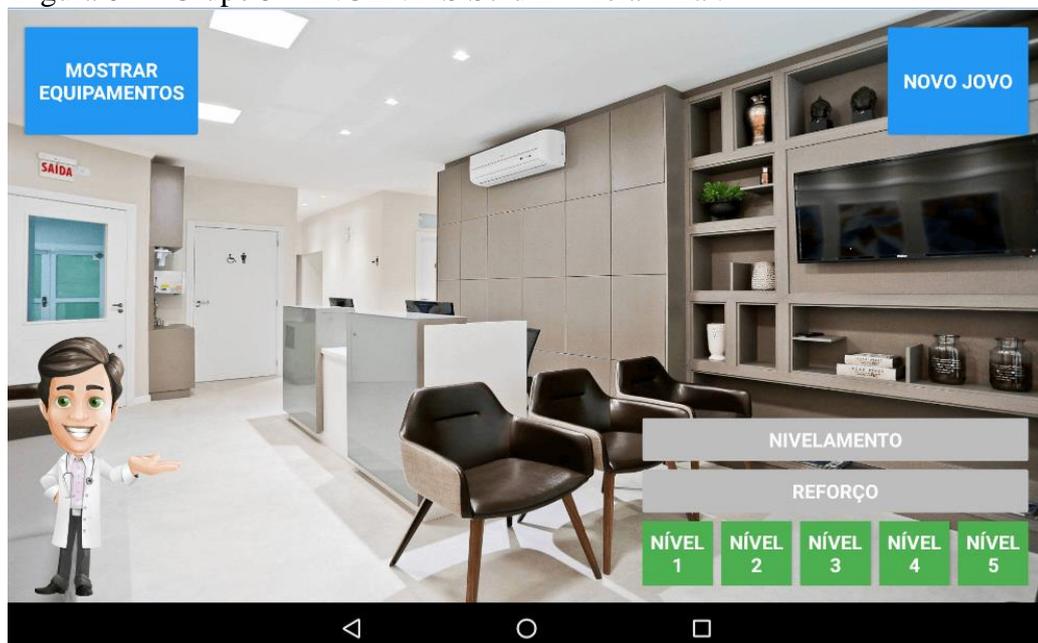


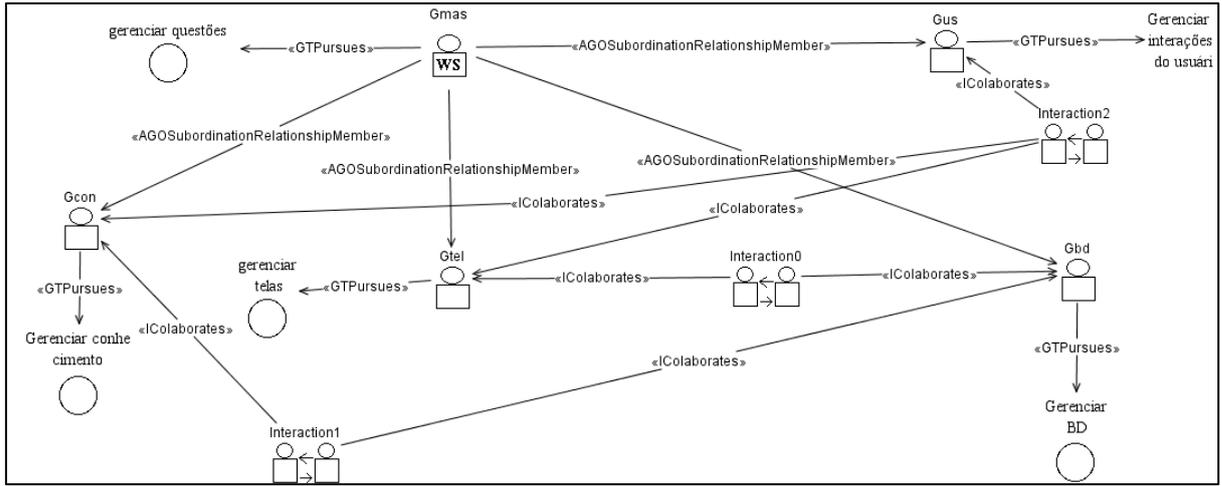
Figura 81 – Grupo 5 – INGENIAS Scrum – Tela Final.



ANEXO G – Artefatos desenvolvidos pelo grupo 6 – INGENIAS Scrum

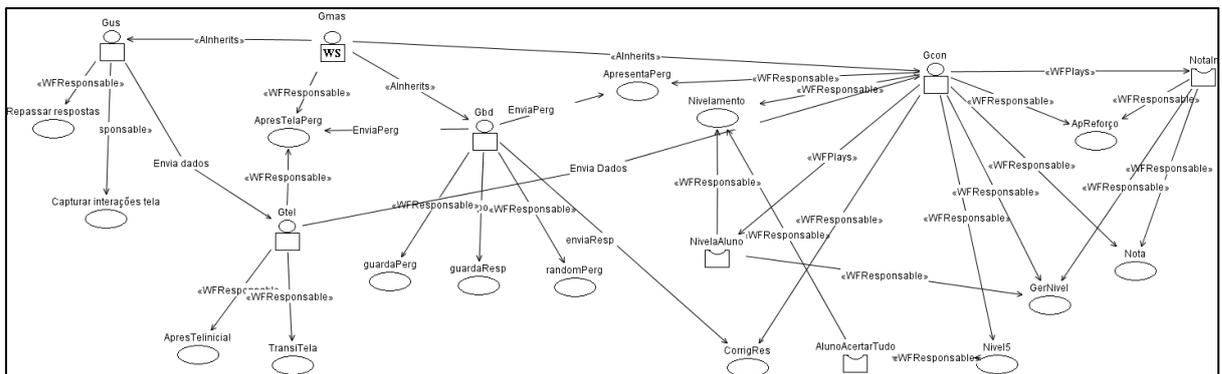
- Modelo de organização:

Figura 82 – Grupo 6 – INGENIAS Scrum – Modelo de Organização.



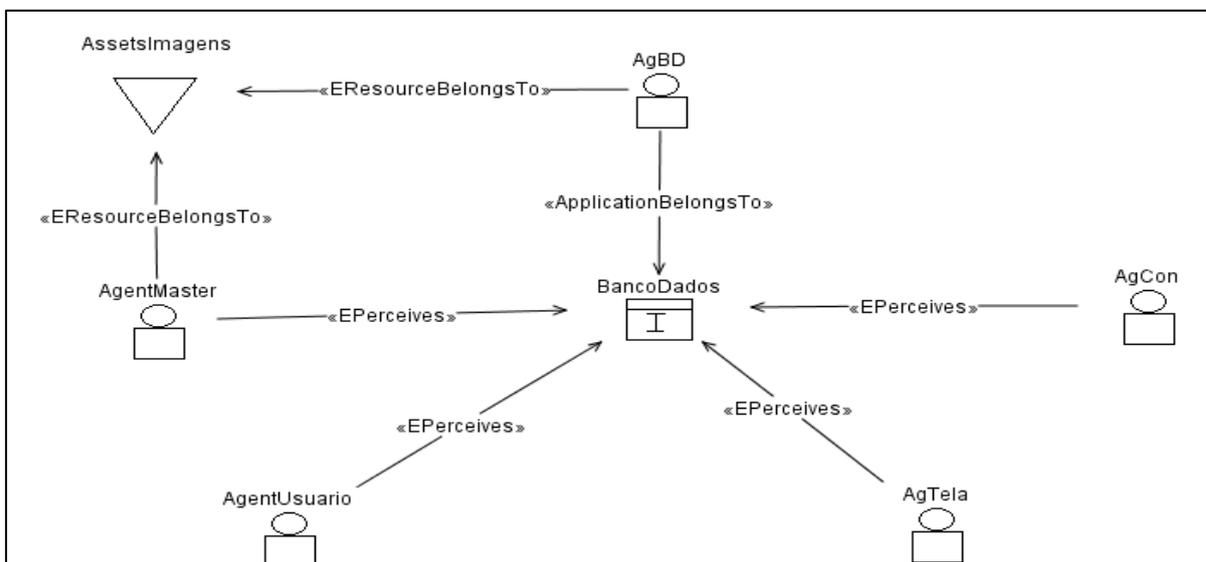
- Modelo de agentes:

Figura 83 – Grupo 6 – INGENIAS Scrum – Modelo de Agentes.



- Modelo de ambiente:

Figura 86 – Grupo 6 – INGENIAS Scrum – Modelo de Ambiente.



- 1ª versão do *backlog*:

Quadro 40 – Grupo 6 – INGENIAS Scrum – 1ª versão do *backlog*.

1	Definir a tela inicial com os componentes: Novo, Sair, Layout inicial e Imagem da tela
2	Unir os agentes na modelagem
3	Criar tela intermediária
4	Conectar com BD
5	Criar metamodelo de Organização
6	Criar modelo de casos de uso

- 2ª versão do *backlog*:

Quadro 41 – Grupo 6 – INGENIAS Scrum – 2ª versão do *backlog*.

1	Organizar o <i>Backlog</i>
2	Buscar perguntas para o jogo
3	Criar tela telas de perguntas tipo som
4	Criar tela telas de perguntas tipo imagem
5	Ler as perguntas do BD e aparecer na tela
6	Criar metamodelo de Agentes
7	Criar metamodelo de metas e tarefas

- 3ª versão do *backlog*:

Quadro 4211 – Grupo 6 – INGENIAS Scrum – 3ª versão do *backlog*.

1	Criar tela simulando o ganho do equipamento
2	Criar tela simulando o reforço
3	Criar regra na modelagem caso a pessoa acerte tudo, ela necessariamente começa do nível 5
4	Inserir perguntas no BD
5	Destacar resposta selecionada
6	Demonstração visual (no texto) de acerto ou erro
7	Criar metamodelo de interação

- 4ª versão do *backlog*:

Quadro 43 – Grupo 6 – INGENIAS Scrum – 4ª versão do *backlog*.

1	Criar justificativa para respostas incorretas
2	Criar forma de salvar histórico de respostas
3	Criar tela final do jogo
4	Inserir equipamentos que serão os prêmios
5	Cancelar progresso quando o usuário sair no meio do nível
6	Sincronizar a modelagem com o código desenvolvido pelo protótipo
7	Criar metamodelo de ambiente

- Telas do jogo do grupo 6 – INGENIAS Scrum

Figura 87 – Grupo 6 – INGENIAS Scrum – Tela Inicial.

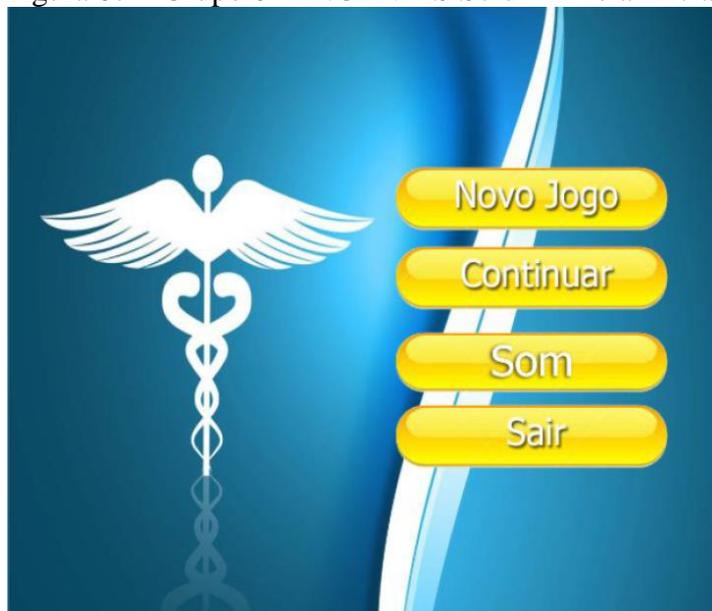


Figura 88 – Grupo 6 – INGENIAS Scrum – Tela principal - Nivelamento.

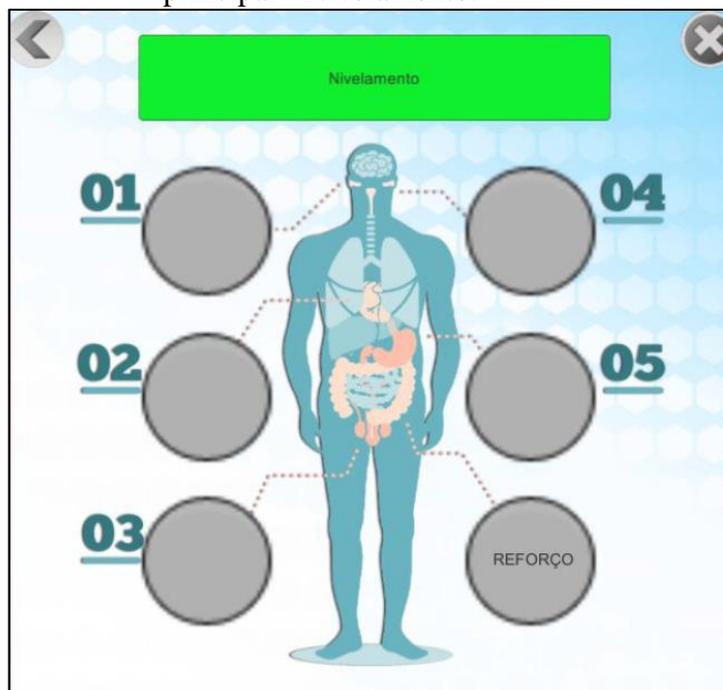


Figura 89 – Grupo 6 – INGENIAS Scrum – Tela Nivelamento.



Figura 90 – Grupo 6 – INGENIAS Scrum – Tela Feedback.

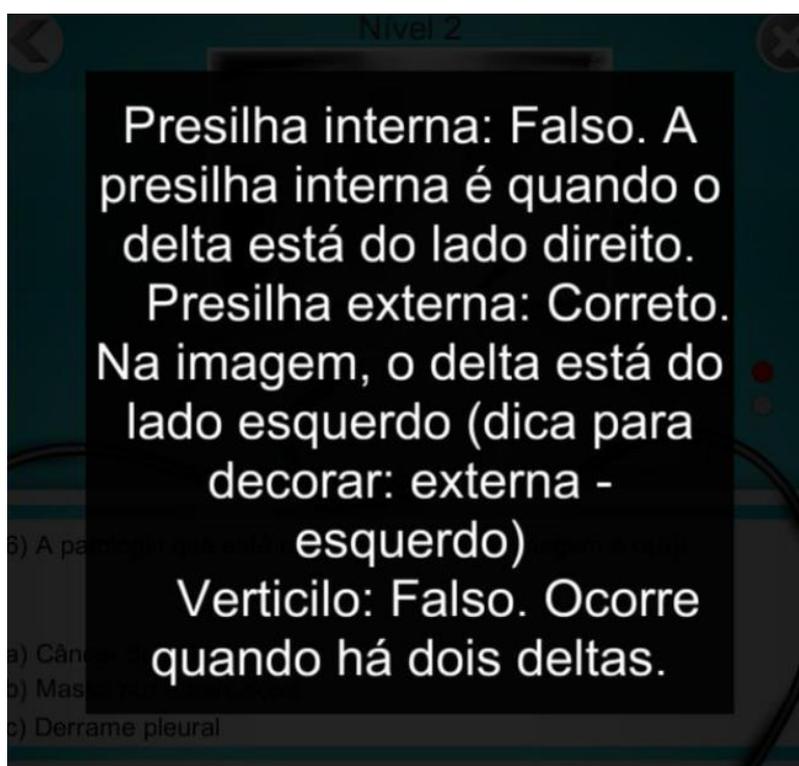


Figura 91 – Grupo 6 – INGENIAS Scrum – Tela Reforço.



Figura 92 – Grupo 6 – INGENIAS Scrum – Tela Histórico do Usuário no Nível.

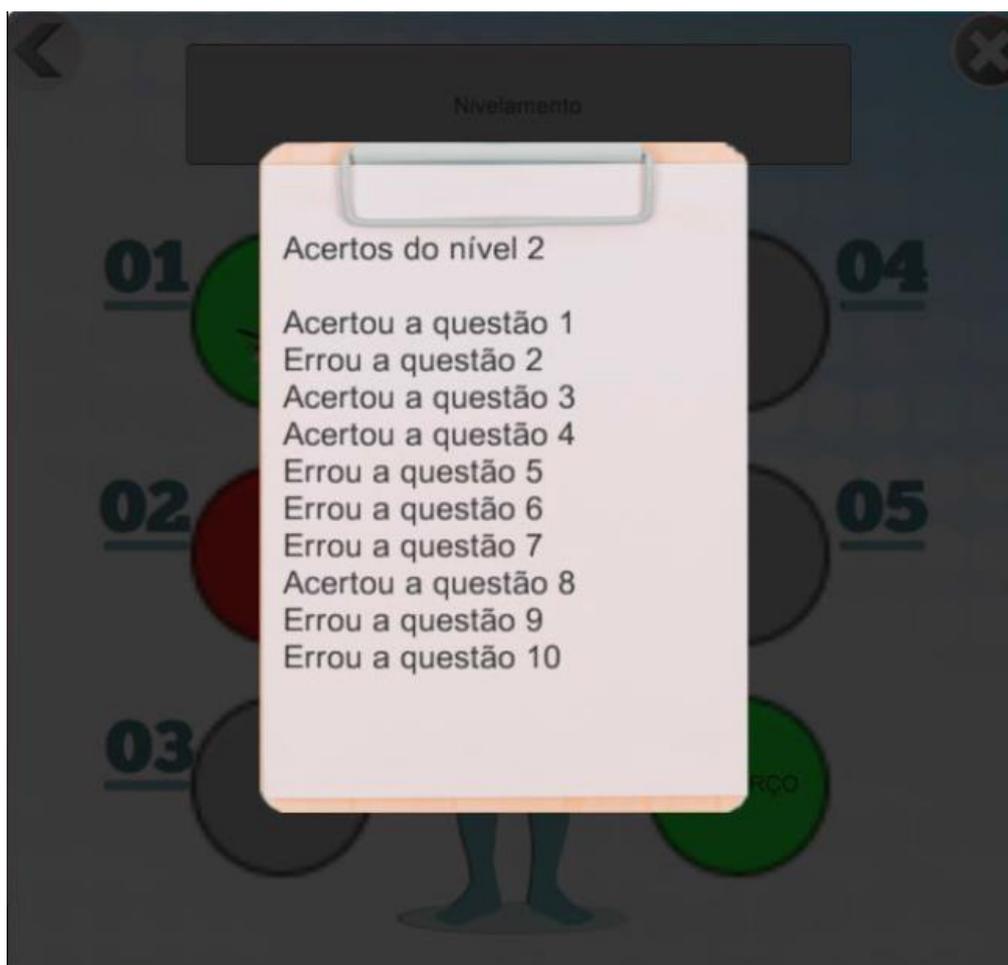


Figura 9311 – Grupo 6 – INGENIAS Scrum – Tela Ganho de Equipamento.



Figura 94 – Grupo 6 – INGENIAS Scrum – Tela Histórico de Ganho de Equipamentos.

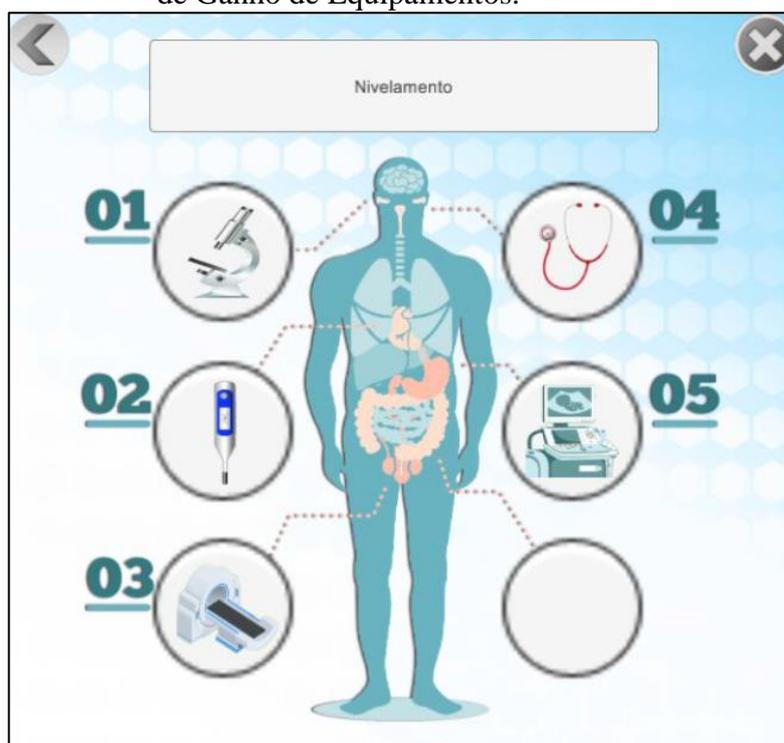


Figura 95 – Grupo 6 – INGENIAS Scrum – Tela Final.

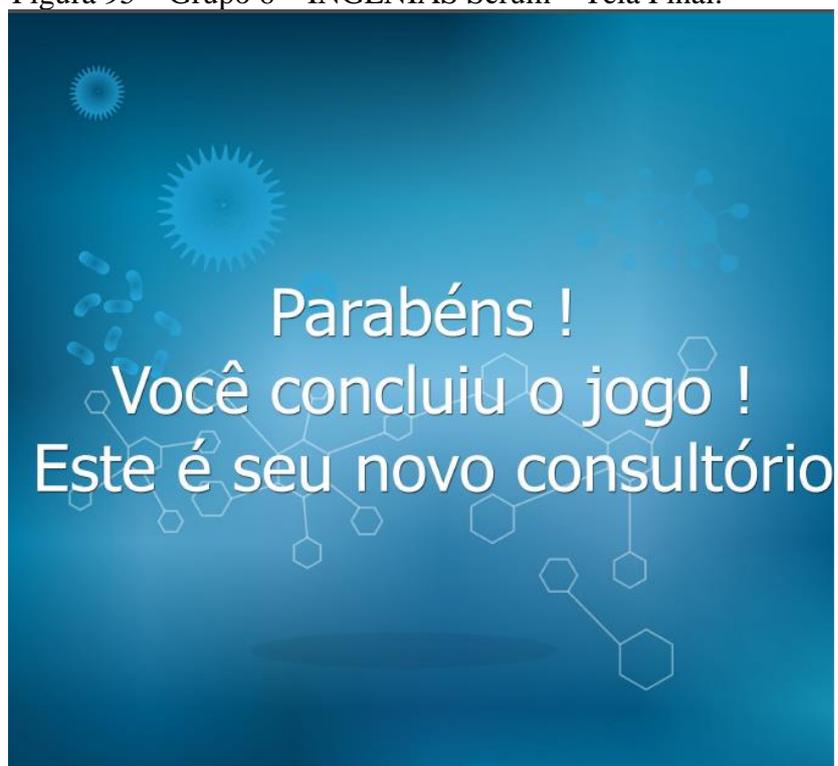
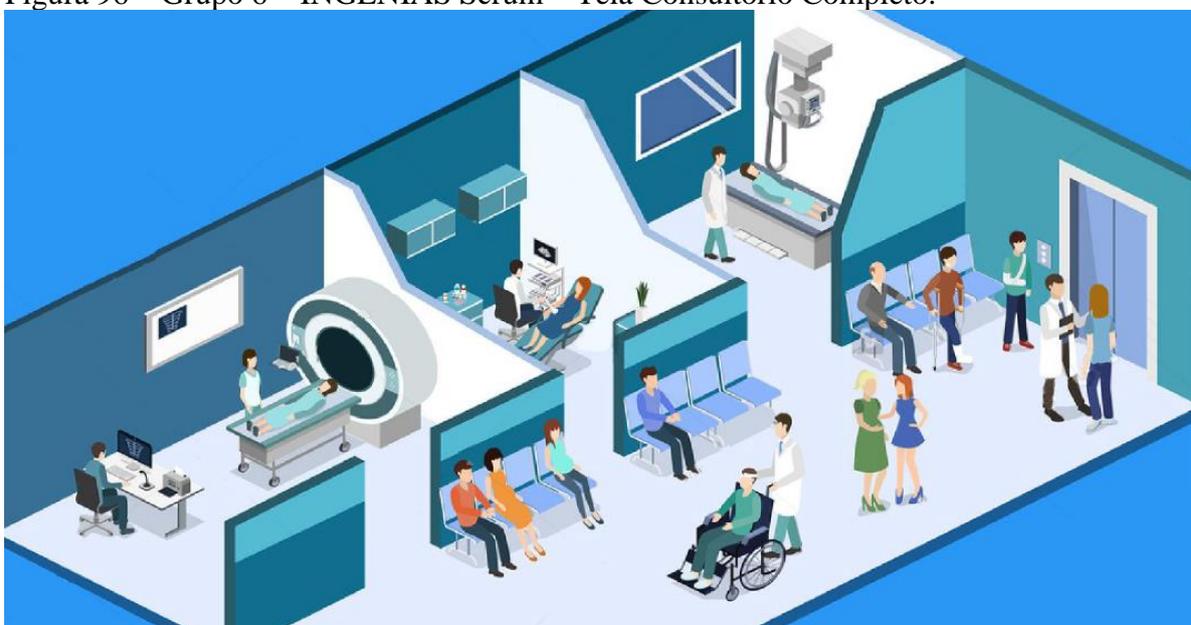


Figura 96 – Grupo 6 – INGENIAS Scrum – Tela Consultório Completo.



ANEXO H – Comentários sobre a experiência dos envolvidos no estudo de caso.

Como a metodologia foi aplicada de forma prática, era mais compreensível entender a abordagem. Outro ponto positivo foi o aprendizado maior em questões médicas através da pesquisa de questões. Foi um trabalho longo, mas extremamente positivo.

Como todo projeto que utiliza uma metodologia ágil, acredito que a melhor parte é a entrega de um sprint finalizado. Você começar a ver o projeto tomar forma é muito gratificante.

Ter contato com uma nova metodologia e trabalhar no desenvolvimento de um jogo, experiências inéditas para mim.

O aumento do conhecimento do desenvolvimento de jogos, e interação de programas com banco de dados.

O que mais gostei foi da organização e eficiência para desenvolver e documentar o projeto.

Conhecer um tipo de metodologia que não conhecia e o modo de relacionar a metodologia com o desenvolvimento de sistema.

Ter uma experiência prática na qual eu pude praticar todo o conteúdo dado até hoje.

O tema foi algo diferente e o SCRUM ajudou a nos organizar melhor para validar o novo progresso durante o desenvolvimento.

O aprendizado induzido pela experiência, foi extremamente engrandecedor, dando uma nova perspectiva no desenvolvimento de software.

A implementação de agentes na prática e ter um novo método para desenvolver projetos.

O desenvolvimento dos diagramas e da programação do aplicativo.

Para mim, a melhor parte foi modificar meu modo de pensar enquanto estava desenvolvendo, passando a pensar nas responsabilidades de cada agente.

Aprender e trabalhar com algo novo.

A experiência foi interessante para vermos como realmente funciona um sistema multiagente.

Desenvolver um aplicativo educacional.

A experiência de poder desenvolver um jogo do zero, com livre escolha de linguagem.

O contato com uma área nova para mim.

A facilidade de desenvolver com a metodologia e o resultado final.

Ter estudado um assunto diferente do que é visto em sala de aula.