

UNIVERSIDADE DE PASSO FUNDO
INSTITUTO DE CIÊNCIAS EXATAS E GEOCIÊNCIAS
PROGRAMA DE PÓS-GRADUAÇÃO EM
COMPUTAÇÃO APLICADA

AGENTES DE SOFTWARE EM UM SISTEMA TUTOR
INTELIGENTE DE APOIO À PREPARAÇÃO
PARA A PROVA DE RESIDÊNCIA MÉDICA

André Luís Stefanello

Passo Fundo

2017

UNIVERSIDADE DE PASSO FUNDO
INSTITUTO DE CIÊNCIAS EXATAS E GEOCIÊNCIAS
PROGRAMA DE PÓS-GRADUAÇÃO EM COMPUTAÇÃO APLICADA

AGENTES DE SOFTWARE EM UM SISTEMA TUTOR INTELIGENTE
DE APOIO À PREPARAÇÃO
PARA A PROVA DE RESIDÊNCIA MÉDICA

André Luís Stefanello

Dissertação apresentada como requisito parcial
à obtenção do grau de Mestre em Computação
Aplicada na Universidade de Passo Fundo.

Orientador: Roberto dos Santos Rabello
Coorientador: Cassiano Mateus Forcelini

Passo Fundo
2017

CIP – Catalogação na Publicação

F816a Stefanello, André Luís
Agentes de software em um sistema tutor inteligente
de apoio à preparação para a prova de residência médica /
André Luís Stefanello. – 2017.
137 f. : il. color. ; 30 cm.

Orientador: Prof. Dr. Roberto dos Santos Rabello.
Coorientador: Prof. Dr. Cassiano Mateus Forcelini.
Dissertação (Mestrado em Computação Aplicada)
Universidade de Passo Fundo, 2017.

1. Sistemas tutoriais inteligentes. 2. Residentes
(Medicina). 3. Agentes inteligentes (Software). 4. Testes.
I. Rabello, Roberto dos Santos, orientador. II. Forcelini,
Cassiano Mateus, coorientador. III. Título.


CDU: 004:37

Catalogação: Bibliotecária Juliana Langaro Silveira - CRB 10/2427


**ATA DE DEFESA DO
TRABALHO DE CONCLUSÃO DE CURSO DO ACADÊMICO**

ANDRÉ LUÍS STEFANELLO

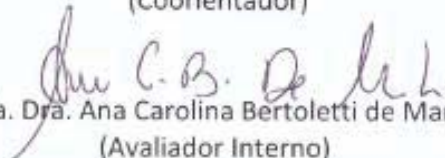
Aos sete dias do mês de julho do ano de dois mil e dezessete, às 14 horas, realizou-se, no Instituto de Ciências Exatas e Geociências, prédio B5, da Universidade de Passo Fundo, a sessão pública de defesa do Trabalho de Conclusão de Curso “**Um sistema tutor inteligente de apoio à preparação para a prova de Residência Médica**”, de autoria de André Luís Stefanello, acadêmico do Curso de Mestrado em Computação Aplicada do Programa de Pós-Graduação em Computação Aplicada – PPGCA/UPF. Segundo as informações prestadas pelo Conselho de Pós-Graduação e constantes nos arquivos da Secretaria do PPGCA, o aluno preencheu os requisitos necessários para submeter seu trabalho à avaliação. A banca examinadora foi composta pelos doutores Roberto dos Santos Rabello, Cassiano Mateus Forcelini, Ana Carolina Bertolotti de Marchi e Rejane Frozza. Concluídos os trabalhos de apresentação e arguição, a banca examinadora considerou o candidato APROVADO. Foi concedido o prazo de até quarenta e cinco (45) dias, conforme Regimento do PPGCA, para o acadêmico apresentar ao Conselho de Pós-Graduação o trabalho em sua redação definitiva, a fim de que sejam feitos os encaminhamentos necessários à emissão do Diploma de Mestre em Computação Aplicada. Para constar, foi lavrada a presente ata, que vai assinada pelos membros da banca examinadora e pela Coordenação do PPGCA.



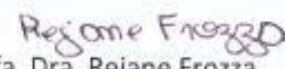
Prof. Dr. Roberto dos Santos Rabello
Presidente da Banca Examinadora
(Orientador)




Prof. Dr. Cassiano Mateus Forcelini
(Coorientador)



Profa. Dra. Ana Carolina Bertolotti de Marchi
(Avaliador Interno)



Profa. Dra. Rejane Frozza
(Avaliador Externo)



Prof. Dr. Rafael Rieder
Coordenador do PPGCA

Dedico este trabalho a todos os meus
familiares pelo apoio incondicional, durante
esta caminhada.

AGRADECIMENTOS

Agradeço, inicialmente, a Deus, por me proporcionar saúde e paz durante esta caminhada, e a chance de poder melhorar cada vez mais pessoal e profissionalmente.

Agradeço, em especial, à minha querida e amada esposa Liana Maria Basso Stefanello, pelo apoio incondicional nestes dois anos, pela paciência e compreensão, para com as horas difíceis, pela compreensão de minha ausência, muitas vezes tendo que desempenhar o papel de Pai e Mãe, muito obrigado, meu AMOR.

À minha filha querida, Maria Luísa Basso Stefanello, pelos beijos de boa noite na noite anterior às minhas viagens a Passo Fundo e pela pergunta incessante, antes de todas as viagens: “- Pai, amanhã você vai estar em casa né?”, agradeço também pela compreensão quando da minha ausência.

Ao meu “pequeno” João Pedro Basso Stefanello, apesar de criança, pela serenidade com que encarava minhas viagens, também com a boa e velha pergunta antes de dormir, na noite anterior a minhas viagens: “- Pai, amanhã você vai estar em casa né?”. Agradeço também pela compreensão quando da minha ausência.

Não há formas de agradecer tudo e todo o apoio, carinho, compreensão, conselhos, vivências compartilhadas com essas três pessoas maravilhosas, o sentimento é de puro amor, pois são a força, a esperança e a razão de meu viver.

Agradeço à minha Mãe, Leda Maria Rubin Stefanello, pelos ensinamentos sempre sábios, e pela frase sempre dita: “- Meu filho, estudo ninguém nunca irá te tirar.” Obrigado.

A meu pai, Dacir José Stefanello, por me ensinar a ser paciente e nunca desistir de meus objetivos, “Não importa o quanto você é forte, mas sim o quanto você consegue suportar.”.

Ao meu irmão Marcos Ângelo Stefanello, pelas caronas e ajuda quando precisei enquanto estava fora de casa nos afazeres do Mestrado.

À Família de minha Esposa, Ordevino Domingos Basso, Maria Bandeira Basso, Lucas Antônio Basso, também pelo apoio incondicional, nos momentos difíceis.

Agradeço a meu Orientador Dr. Roberto dos Santos Rabello, inicialmente por, há dois anos, confiar no meu trabalho e me selecionar como seu orientando, foi um prazer trabalhar com você.

Agradeço, também, a meu Coorientador Dr. Cassiano Mateus Forcelini, pelo apoio incondicional, pelas trocas de ideias e contribuições. Agradeço pelas palavras de apoio e confiança em meu trabalho.

Agradeço a Gustavo Hirt, aluno do curso de Medicina e apoiador do trabalho, agradeço também ao Dr. Rubens Rodriguez e estendo o agradecimento a todos os Professores e funcionários do PPGCA da Universidade de Passo Fundo.

Agradeço à URI – Universidade Regional Integrada, pelo apoio dispendido durante esta caminhada, sem este, esta etapa de minha formação profissional não teria sido concluída.

“O homem científico não almeja resultados imediatos. Ele não espera que suas ideias mais avançadas sejam rapidamente retomadas. Seu trabalho é como o de um agricultor para o futuro. Seu dever é estabelecer bases para aqueles que estão por vir e apontar o caminho a ser seguido.”

Nikola Tesla

AGENTES DE SOFTWARE EM UM SISTEMA TUTOR INTELIGENTE DE APOIO À PREPARAÇÃO PARA A PROVA DE RESIDÊNCIA MÉDICA

RESUMO

A utilização de provas de seleção vem se tornando algo cada vez mais presente no dia a dia das pessoas, tanto no meio profissional como acadêmico. Neste contexto, pode-se citar a prova de residência médica como um desafio para os alunos do curso de medicina. De acordo com dados estatísticos, tem-se a ciência da dificuldade de resolução da mesma, bem como a busca por soluções que possam auxiliá-los na preparação. Buscou-se neste sentido, a criação de um sistema dotado de agentes de *software* que possa contribuir ou auxiliar estes alunos em seus estudos. Para tanto, inserem-se neste contexto os STIs (Sistemas Tutores Inteligentes), que são sistemas capazes de apoiar o processo de aprendizagem, através de interações autônomas por seus agentes de *software*. Nessa perspectiva, este trabalho surgiu da necessidade de demonstrar, de forma diferenciada, o conteúdo para o estudo de disciplinas que envolvam as cinco grandes áreas da medicina: Clínica Médica, Pediatria, Ginecologia/Obstetrícia, Cirurgia Geral e Medicina Social, utilizando conceitos de agentes de *software* e STIs. Para tanto, foi desenvolvida a documentação de requisitos de *software*, bem como de um STI voltado para a área da medicina, dotado de quatro agentes de *software*. Como teste funcional de *software*, dados sobre o funcionamento dos agentes de *software*, foram coletados, tabulados e posteriormente apresentados em forma de gráficos. Este estudo demonstrou bons resultados quanto ao funcionamento dos agentes de *software*, respectivamente: no Nivelamento, Alocação, Tutoria e Duelo. Desse modo, o uso do STI desenvolvido pode facilitar e auxiliar na avaliação de desempenho dos alunos e jovens médicos, além de dar aporte aos professores, para que possam utilizar uma nova forma de comunicação e interação com os alunos.

Palavras-chave: Agentes de *software*, Medicina, Sistemas Tutores Inteligentes.

A TUTOR SYSTEM PREPARING TO SUPPORT SMART FOR RESIDENCY PROOF

ABSTRACT

The use of selection tests is becoming more and more present in people's daily lives, both in the professional and academic environments. In this context, one can cite proof of medical residency as a challenge for medical students. According to statistical data, one has the science of the difficulty of solving it, as well as the search for solutions that can help them in the preparation. In this sense, we have sought to create a system with software agents that can contribute or help these students in their studies. To this end, the STIs (Intelligent Tutors Systems) are inserted in this context, which are systems capable of supporting the learning process through autonomous interactions by its software agents. In this perspective, this work arose from the need to demonstrate, in a differentiated way, the content for the study of disciplines involving the five major areas of medicine: Medical Clinic, Pediatrics, Gynecology / Obstetrics, General Surgery and Social Medicine, using agent concepts Software and STIs. For that, the documentation of software requirements was developed, as well as an STI focused on the area of medicine, with four software agents. As a functional software test, data on the functioning of the software agents were collected, tabulated and later presented as graphs. This study demonstrated good results regarding the functioning of software agents, respectively: in Leveling, Allocation, Tutoring and Dueling. Thus, the use of STI developed can facilitate and assist in the evaluation of the performance of students and young physicians, as well as giving teachers a new way of communicating and interacting with students.

Keywords: Software Agents, Medicine, Intelligent Tutors.

LISTA DE FIGURAS

Figura 1: Interação de um agente adaptado de RUSSELL e NORVIG [15].....	24
Figura 2: Estrutura de Categorias de Agentes Inteligentes, adaptado de BRENNER, W; WITTIG, H.; ZARNEKOW, R. [20]).	25
Figura 3: Programa do agente reativo simples Agente-Aspirador. Adaptado de RUSSELL [22]	28
Figura 4: Diagrama esquemático agente reativo simples. Fonte: Adaptado de RUSSELL [22]	28
Figura 5: Um agente reativo simples. Fonte: Adaptado de RUSSELL [22]	28
Figura 6: Arquitetura Clássica de um STI. Adaptado de RUSSELL e NORVIG [15].	30
Figura 7: Fluxo de Atividade de Agentes. Fonte: Adaptado de LYRA [33].....	34
Figura 8: Fluxo de funcionamento do ciclo interno. Fonte: Adaptado de KAUTZMANN e JAQUES [34].....	36
Figura 9. Etapas propostas para a execução do trabalho. Fonte: Elaborado pelo autor.	39
Figura 10 Processo de Engenharia de <i>Software</i> – Desenvolvimento Incremental. Fonte: Adaptado de SOMMERVILLE [37].....	41
Figura 11: Arquitetura do STI desenvolvido. Fonte: Elaborado pelo autor.	46
Figura 12: Agente reativo simples Duelo. Fonte: Elaborado pelo autor.	49
Figura 13: Ferramentas e tecnologias, utilizadas para a implementação dos sistemas. Fonte: Elaborado pelo autor.....	52
Figura 14: Tela do <i>menu</i> Cadastrar. Fonte: elaborada pelo autor.....	54
Figura 15: Tela de Cadastro de Tipo de Usuário. Fonte: Elaborada pelo autor.	55
Figura 16: Tela de Cadastro de Sub Áreas. Fonte: Elaborada pelo autor.....	56
Figura 17: Tela de Cadastro de Configuração de Níveis de Dificuldade. Fonte: Elaborada pelo autor.	57
Figura 18: Tela de cadastro de questões. Fonte: Elaborada pelo autor.	59
Figura 19: Tela de cadastro de questões. Fonte: Elaborada pelo autor.	59
Figura 20: Tela de cadastro de questões. Fonte: Elaborada pelo autor.	60
Figura 21: Tela de cadastro de dicas de nível. Fonte: elaborada pelo autor.....	60
Figura 22: Telas de cadastro de desafio. Fonte: Elaborada pelo autor.	61
Figura 23: Tela de <i>login</i> do Sistema. Fonte: elaborada pelo autor.	62
Figura 24: Fluxo de dados <i>login</i> do Sistema. Fonte: elaborada pelo autor.	63

Figura 25: Gráfico de níveis completos. Fonte: Elaborado pelo autor.	64
Figura 26: Tela do Sistema MedicalGame. Fonte: elaborada pelo autor.	64
Figura 27: Ranking top 10. Fonte: Elaborado pelo autor.	65
Figura 28: Fluxo de dados MedicalGame. Fonte: elaborado pelo autor.	66
Figura 29: Agente de Nivelamento. Fonte: Elaborada pelo autor.	68
Figura 30: Fluxo de dados processo de nivelamento. Fonte: elaborada pelo autor.	69
Figura 31: Fluxo de dados processo de alocação. Fonte: Elaborada pelo autor.	71
Figura 32: Representação gráfica da primeira tutoria. Fonte: Elaborado pelo autor.	73
Figura 33: Representação gráfica da segunda tutoria. Fonte: Elaborado pelo autor.	74
Figura 34: Representação gráfica, erro última tentativa de resposta. Fonte: Elaborado pelo autor.	75
Figura 35: Caso de uso – Agente Tutoria. Fonte: Elaborado pelo autor.	75
Figura 36: Tela de Duelo. Fonte: elaborada pelo autor.	78
Figura 37: Fluxo de Dados do Agente Duelo. Fonte: elaborada pelo autor.	79
Figura 38: Laboratório de Informática da Faculdade de Medicina UPF.	82
Figura 39: Dados Agentes Nivelamento. Fonte: elaborado pelo Autor.	84
Figura 40: Dados do Agente de Alocação. Fonte: Elaborado pelo autor.	86
Figura 41: Número Acertos e Erros do Nível 1. Fonte: Elaborado pelo autor.	87
Figura 42: Número Acertos e Erros do Nível 2. Fonte: Elaborado pelo Autor.	88
Figura 43: Número Acertos e Erros do Nível 3. Fonte: Elaborado pelo Autor.	88
Figura 44: Número Acertos e Erros do Nível 4. Fonte: Elaborado pelo Autor.	89
Figura 45: Número Acertos e Erros do Nível 5. Fonte: Elaborado pelo Autor.	90
Figura 46: Página de Apresentação do Sistema MedicalGame. Fonte: Elaborado pelo autor.	98

LISTA DE TABELAS

Tabela 1: Quadro comparativo, trabalhos relacionados. Fonte: Elaborada pelo autor.....	38
Tabela 2: Descrição do caso de uso Nivelar Aluno. Fonte: Elaborado pelo autor.....	69
Tabela 3: Caso de uso – Agente de Alocação. Fonte: Elaborado pelo autor.....	71
Tabela 4: Descrição do caso de uso Agente Tutoria. Fonte: Elaborado pelo autor.....	76
Tabela 5: Descrição do caso de uso Agente Duelo. Fonte: Elaborado pelo autor.....	79

LISTA DE SIGLAS

ASP – *Active Server Pages*

CAI – Instrução Assistida por computador

CNRM – Comissão Nacional de Residência Médica

Cremsp – Conselho Regional de Medicina do Estado de São Paulo

CSS – *Cascading Style Sheets*

DARP – *Dynamic Analysis and Replanning Tool*

FTP – *File Transfer Protocol*

HTML – *Hypertext Markup Language*

HTML5 – *Hypertext Markup Language*, versão 5

IA – Inteligência Artificial

IBM – *International Business Machines*

PHP – acrônimo recursivo para PHP: *Hypertext Preprocessor*

POO – Programação Orientada a Objetos

SGBD – Sistema de Gerenciamento de Banco de Dados

STI – Sistemas Tutores inteligentes

TIC – Tecnologia da Informação e Comunicação

UPF – Universidade de Passo Fundo

XML – *eXtensible Markup Language*

SUMÁRIO

1. INTRODUÇÃO	19
2. FUNDAMENTAÇÃO TEÓRICA.....	23
2.1. AGENTES	23
2.1.1. Categorias de Agentes Inteligentes.....	24
2.1.1.1. Agentes de interface	25
2.1.1.2. Agentes móveis.....	26
2.1.1.3. Agentes Reativos Simples	27
2.1.1.4. Agentes de Informação	29
2.2. SISTEMAS TUTORES INTELIGENTES – STI.....	29
2.2.1. Arquitetura dos Sistemas Tutores Inteligentes	30
2.2.1.1. Modelo do Aluno.....	31
2.2.1.2. Modelo do domínio.....	31
2.2.1.3. Modelo Pedagógico	32
2.2.1.4. Modelo da Interface.....	32
2.3. TRABALHOS RELACIONADOS	33
2.3.1. Agentes de <i>Software</i> no Monitoramento de Alunos em Educação a Distância.....	33
2.3.2. Um modelo de treinamento adaptativo da habilidade metacognitiva de monitoramento do conhecimento.	35
3. METODOLOGIA DE TRABALHO	39
3.1. PROVA DE RESIDÊNCIA MÉDICA	40
3.2. 1ª ETAPA - REVISÃO.....	40
3.3. 2ª ETAPA – INVESTIGAÇÃO E DESENVOLVIMENTO	40
3.4. 3ª ETAPA – AVALIAÇÃO.....	42
4. ABORDAGEM PROPOSTA – MEDICALGAME.....	45
4.1. PROPOSTA DA ARQUITETURA DO STI – MEDICALGAME	45
4.1.1. Estratégias e táticas do modelo pedagógico.....	47
4.2. PROPOSTA E MODELAGEM DA ARQUITETURA BASEADA EM AGENTES DE <i>SOFTWARE</i>	48
4.3. DEFINIÇÃO DA UTILIZAÇÃO DE FRAMEWORKS/TÉCNICAS E FERRAMENTAS PARA O DESENVOLVIMENTO DE SISTEMAS BASEADOS EM AGENTES DE <i>SOFTWARE</i>	49
4.4. IMPLEMENTAÇÃO DO PROTÓTIPO DE STI.....	52
4.4.1. Página Administrativa.....	53

4.4.1.1.	Cadastrar tipo de usuário	54
4.4.1.2.	Cadastrar instituição	55
4.4.1.3.	Cadastrar áreas das questões.....	55
4.4.1.4.	Cadastrar subáreas das questões	56
4.4.1.5.	Cadastrar nível de dificuldade de questões.....	56
4.4.1.6.	Cadastro de questões.....	57
4.4.1.7.	Cadastro de dicas para os níveis	60
4.4.2.	Página do Sistema MedicalGame	61
4.4.2.1.	Cadastro de usuário/alterações de senha.....	61
4.4.3.	Tela do Sistema MedicalGame	63
4.5.	AGENTES DESENVOLVIDOS NO SISTEMA MEDICALGAME	66
4.5.1.	Agente de nivelamento	67
4.5.2.	Agente de Alocação.....	70
4.5.3.	Agente Tutoria	72
4.5.4.	Agente Duelo	77
5.	ESTUDO DE CASO	81
5.1.	CONSIDERAÇÕES ÉTICAS	81
5.2.	CARACTERIZAÇÃO DOS PARTICIPANTES	81
5.3.	EXPERIMENTO	81
5.4.	TESTE DE <i>REALEASE</i> E DISCUÇÃO DOS RESULTADOS.....	82
5.4.1.	Agente de Nivelamento.....	83
5.4.2.	Agente de Alocação.....	84
5.4.3.	Agente de Tutoria	86
5.4.4.	Agente Duelo	90
6.	CONCLUSÃO.....	91
6.1.	DISSEMINAÇÃO DO CONHECIMENTO.....	93
6.2.	TRABALHOS FUTUROS	93
	REFERÊNCIAS	95
	APÊNDICE A - TERMO DE CONSENTIMENTO	97
	APÊNDICE B - PÁGINA INSTITUCIONAL	98
	APÊNDICE C - DOCUMENTO DE REQUISITO DE SOFTWARE – PÁGINA	
	ADMINISTRATIVA MEDICAL GAME E SISTEMA MEDICALGAME..	99
	APÊNDICE D - ESTÓRIA CENÁRIO DE TESTES MEDICALGAME.....	137

1. INTRODUÇÃO

Segundo o MINISTÉRIO DA EDUCAÇÃO [1], a residência médica é uma modalidade de ensino de pós-graduação caracterizada por treinamento em serviço, destinada a médicos sob a forma de curso de especialização. Foi instituída pelo Decreto nº 80.281, de 5 de setembro de 1977. Funciona em instituições de saúde, sob a orientação de profissionais médicos, de elevada qualificação ética e profissional, sendo considerado o “padrão ouro” da especialização médica. CHAVES [2] cita que, com este mesmo decreto, foi criada, também, a Comissão Nacional de Residência Médica (CNRM), que tem como premissa considerar a necessidade de médicos especialistas, indicada pelo perfil socioepidemiológico da população, em consonância com os princípios e as diretrizes do Sistema Único de Saúde (SUS).

O MINISTÉRIO DA EDUCAÇÃO [1] ainda pondera que o Programa de Residência Médica, cumprido integralmente dentro de uma determinada especialidade, confere ao médico residente o título de especialista. Conforme dados do CREMERS [3], em torno de 900 a 1000 novos médicos são formados anualmente no estado do Rio Grande do Sul. Se levada em consideração a região Sul do Brasil, Rio Grande do Sul, Santa Catarina e Paraná, esse total se aproxima de 2.700 médicos formados por ano.

Em contrapartida, CHAVES [2] traz, em seu estudo, dados de 362 instituições de ensino distribuídas nas cinco regiões do Brasil. Estes dados são relevantes e devem ser considerados. Dentre eles, o número de vagas de residência médica por região, sendo que a região Sul tem um percentual considerável destas vagas, cerca de 15,2%, o que perfaz um total de 1.202 vagas do total geral de 7.931 vagas disponíveis no Brasil.

Em uma comparação entre os dados apresentados pelo CREMERS [3] e CHAVES [2], verifica-se que a quantidade de vagas de residência médica é deficitária, e o número aproximado de alunos formados anualmente na região Sul também é deficiente, apresentando 2.700 alunos formados por ano, para um total de 1.202 vagas. Esse número de vaga supre apenas 44% das requisições ou solicitações geradas.

Outro ponto a ser considerado é a quantidade de alunos que são reprovados nesta avaliação, sabe-se que dados Nacionais sobre esta prova não estão disponíveis de forma compilada, pois, segundo CHAVES [2], a definição dos parâmetros de pontuação e organização da prova de residência médica é de responsabilidade de cada coordenação de programa. Usa-se, aqui, como base, o MINISTÉRIO DA EDUCAÇÃO [1], EBC [4] e SALEM [5], os quais afirmam que mais da metade dos recém-formados em medicina, no

estado de São Paulo, foram reprovados no exame do Conselho Regional de Medicina do Estado (CREMESP), uma prova equivalente à prova de residência médica.

Desde o fim da década de 90, a perspectiva de uso das mídias eletrônicas e de recursos cibernéticos no ensino médico vem sendo entendida como uma via alternativa para diversificação e facilitação do aprendizado: KARM QAYUMI [6], FARRIMOND [7], MNGUNI [8]. Os resultados iniciais desta metodologia são promissores e permitem a sugestão de sua utilização de forma mais ampla no processo de ensino e aprendizagem médico.

Uma alternativa para os alunos tentarem melhorar seu desempenho nas provas de residência médica é a utilização de cursos preparatórios, disponíveis na *internet*, três destes cursos preparatórios são citados por MEDCEL [9], SJT [10] e SANTACASABH [11]. Estes três cursos pré-prova de residência médica tem como proposta disponibilizar uma pequena parte dos conteúdos de seus portais. Para que o aluno tenha um aprofundamento, ou possa utilizar todos os módulos dos cursos, deve efetuar pagamentos por demanda.

Dada à natureza complexa dos sistemas educacionais, novas abordagens tornam-se cada vez mais relevantes. Nessa perspectiva, SAKOWSKI e TÓVOLLI [4] tratam o aprendizado personalizado como uma importante forma de não somente reconhecer, mas, também, incorporar a heterogeneidade dos alunos, bem como aprimorar o desempenho educacional, estimulando a participação e a aprendizagem.

Neste sentido, os STIs (Sistemas Tutores Inteligentes) são amplamente utilizados em aplicações na área de ensino, pois, impulsionados pela massiva utilização de equipamentos eletrônicos, trazem novas oportunidades de uso, dentre elas o acesso à informação de qualquer lugar e a qualquer hora. Com tais oportunidades, o aluno tem a seu favor maior emponderamento e participação nos processos de ensino e aprendizagem.

Assim sendo, observa-se que no mercado a grande maioria dos *softwares* ou aplicações não se utiliza de agentes de *software*. BERCHT [12], VICCARI E GIRAFFA [13] citam que já há evidências sucintas de que *softwares* educacionais que não utilizem agentes de *software* para conceber certa adaptação ao perfil do aluno não conseguem atingir completamente os objetivos propostos.

Diante disso, o problema desta pesquisa constitui-se em como utilizar agentes de *software* em um STI para auxiliar os alunos de medicina na preparação à prova de residência médica. As várias formas de coleta, organização e apresentação de conteúdos, com a utilização de agentes de *software*, são desafios a serem estudados.

Imbuído deste desafio, o presente trabalho visa desenvolver um Sistema Tutor Inteligente de apoio à preparação para a prova de residência médica, utilizando agentes de *software*. Desta forma, constituem-se como objetivos específicos do trabalho – a proposta e modelagem da arquitetura do STI; a proposta e modelagem da arquitetura de Agentes de *Software*; a definição da utilização de *Frameworks/Técnicas* e Ferramentas para o desenvolvimento de sistemas baseados em Agentes de *Software* e, por fim, a implementação de um protótipo de STI utilizando o conceito de Agentes de *Software* voltado à área de medicina, bem como o teste funcional dos Agentes de *Software* criados durante o desenvolvimento do trabalho.

Denominado MedicalGame, o STI foi desenvolvido utilizando o conceito de agentes de *software* e busca disponibilizar uma opção de *software* que possa ser utilizada por alunos de medicina de qualquer nível de graduação, com a finalidade principal de ajudar na melhoria de seu desempenho acadêmico, assim como no desempenho na prova de residência médica.

A fim de melhor descrever a modelagem do STI, a proposta e modelagem da arquitetura de Agentes de *Software*, as ferramentas que foram utilizadas para configuração e implementação, além da implementação do STI, o presente estudo encontra-se organizado conforme segue: No capítulo 2 apresenta-se a Fundamentação teórica do trabalho, que discorre sobre a prova de residência médica, trata e discute sobre os temas agentes de *software*, categorias de agentes, bem como do contexto de sistemas tutores inteligentes. Traz, também, a conceituação e apresentação das principais características de dois dos trabalhos estudados, realizando-se uma comparação com o trabalho aqui desenvolvido. Na sequência, o Capítulo 3, intitulado “Metodologia de trabalho”, descreve a forma de trabalho, subdividida em três etapas, sendo elas: a etapa de revisão, etapa de investigação e desenvolvimento, e a etapa de avaliação.

O capítulo 4, intitulado “Abordagem proposta – MedicalGame”, está subdividido em quatro seções, que tratam especificamente da proposta da arquitetura do STI desenvolvida. Na seção terceira deste capítulo foram descritas as ferramentas utilizadas para o projeto e implementação do STI e dos agentes de *software*.

Na seção intitulada implementação do protótipo de STI efetivou-se uma subdivisão em três subseções: Página Administrativa, Página da Aplicação – MedicalGame e a descrição de quais as estratégias e táticas foram implementadas no STI.

No Capítulo 5, intitulado “Estudo de caso - teste funcional de *software* para os agentes de *software*”, são apresentadas as considerações éticas, a caracterização dos

participantes e o experimento que foi efetivado com os agente de *software* da aplicação. Como subseção deste capítulo são apresentados os resultados dos testes sobre os quatro agentes de *software* desenvolvidos no trabalho, sendo eles: Agente de Nivelamento; Agente de Alocação; Agente de Tutoria e Agente Duelo. Por fim, no capítulo 6 foram apresentadas as conclusões e apresentação dos trabalhos futuros. Outrossim, material adicional ao trabalho está disponível na seção de Apêndices.

2. FUNDAMENTAÇÃO TEÓRICA

Este capítulo compõe a revisão bibliográfica referente ao trabalho proposto. Serão tratados aqui assuntos como a Prova de Residência Médica, em que são levantados dados referentes à sua criação.

Na seção Agentes trata-se, de forma compilada, das diferentes categorias e estruturas dos agentes de *software*, foco deste trabalho. Na seção intitulada Sistemas Tutores Inteligentes – STI realiza-se uma conceituação desta expressão. Na seção intitulada Trabalhos Relacionados, há a conceituação e descrição das principais características de dois trabalhos que têm como proposta o desenvolvimento de agentes de *software*.

2.1. AGENTES

Segundo RUSSELL e NORVIG [14], os pesquisadores, encorajados pelo processo na resolução de problemas da IA, começaram a examinar, mais uma vez, o problema do “agente como um todo”. Desta forma, o movimento tem como objetivo entender o funcionamento interno dos agentes incorporados a ambientes reais com entradas de sensores contínuas.

Compreende-se que a *Internet* acabou se tornando um dos mais importantes ambientes para o trabalho com agentes. Ainda segundo RUSSELL e NORVIG [14], na atualidade, o sufixo “-bot”, passou a fazer parte da linguagem cotidiana, pois os sistemas de IA tornaram-se muito comuns na *Internet*. Então, a conceituação de agente, descrita por RUSSEL e NORVIG [15] como: “um agente é tudo o que pode ser considerado capaz de perceber seu ambiente por meio de sensores e de agir sobre esse ambiente por intermédio de atuadores”, mostra-se pertinente.

Uma analogia muito simples e eficaz para o entendimento de agentes é a de que um agente humano possui olhos, ouvidos como sensores, mãos, pernas, boca e algumas outras partes do corpo como atuadores. Já um agente robótico, para RUSSELL e NORVIG [14], poderia ter câmeras e vários motores e atuadores em funcionamento. A referida analogia pode ser visualizada, graficamente, da Figura 1.

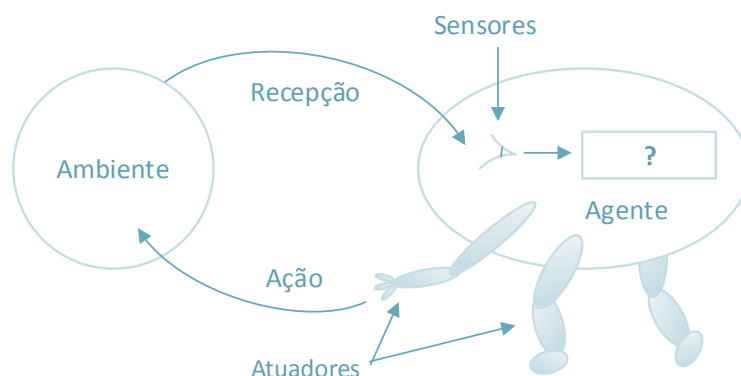


Figura 1: Interação de um agente adaptado de RUSSELL e NORVIG [14].

Do ponto de vista de COPPIN [16], um agente pode ser descrito como uma entidade capaz de realizar alguma tarefa, geralmente para auxiliar um usuário humano. Esses agentes podem ser de tipos diferentes, dentre eles: agentes biológicos, robóticos ou computacionais. Desta forma, há vários meios pelos quais agentes de *software* podem ser construídos, e uma série de propriedades atreladas a eles. Algumas das propriedades e características dos agentes são: autonomia, capacidade de colaborar entre si, além da capacidade de aprender.

Ademais das propriedades e arquiteturas, os agentes podem ser divididos ou classificados em: Agentes reativos; Agentes de Interface e Agentes de Informação. Neste contexto, WOOLDRIDGE [17] especifica que: “agentes são sistemas computacionais capazes de executarem ações autônomas em algum ambiente, a fim de alcançar os objetivos aos quais foram desenvolvidos ou projetados”.

Trabalhando-se sob o contexto de sistemas Multiagentes, em que, geralmente, os Agentes cooperam uns com os outros, sofrendo abordagens diferenciadas, WOOLDRIDGE [17] afirma que as abordagens para a comunicação entre agentes variam em: **Não-comunicação:** Consegue concluir racionalmente a questão; **Comunicação Primitiva:** A comunicação é restrita a alguns conjuntos finitos de sinais; **Alta Comunicação:** Existe um diálogo entre os agentes.

2.1.1. Categorias de Agentes Inteligentes

Esta seção discorrerá sobre as definições visualizadas no terceiro nível de hierarquia da Figura 2, todas as estruturas deste nível serão definidas e descritas. Não serão tratados aqui os níveis de hierarquia um e dois, haja vista que o escopo do trabalho debaterá

somente o item agentes de *software*, em específico, Agentes Reativos Simples, baseados nas características descritas por RUSSELL e NORVIG [14].

Sendo assim, serão tratadas e definidas as seguintes categorias de Agentes: Agente de Interface; Agente Móvel; Agente Reativo e Agentes de Informação. A Figura 2 traz a definição da hierarquia das categorias existentes na arquitetura de agentes.

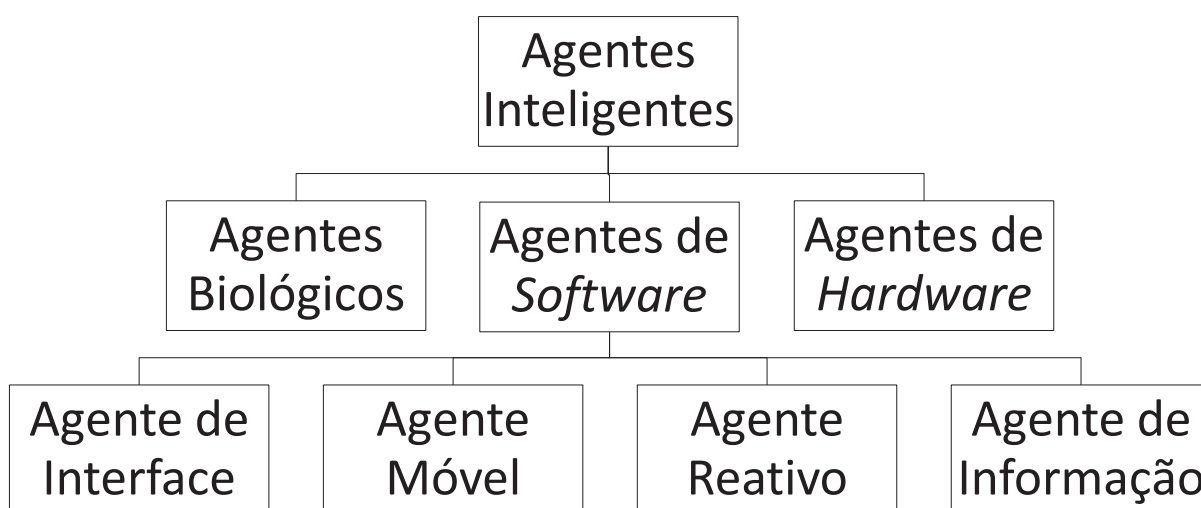


Figura 2: Estrutura de Categorias de Agentes Inteligentes, adaptado de BRENNER, W; WITTIG, H.; ZARNEKOW, R. [18]).

2.1.1.1. Agentes de interface

Um agente de interface, de acordo com COPPIN [16], pode ser considerado um assistente pessoal. Esses são, geralmente, autônomos, capazes de aprender tarefas a fim de realizá-las em nome de um usuário humano. Uma característica interessante nos agentes de interface é a colaboração com usuários, mas os mesmos não precisam colaborar com outros agentes, mesmo que, conforme WOOLDRIDGE [17], em alguns casos, agentes inteligentes possam aprender, buscando recomendações provenientes de outros agentes. Para exemplificar agente de interface, COPPIN [16] faz o uso do exemplo de uma nova ferramenta de *software*, em que o agente tem a capacidade de observar o que o usuário faz e, então, oferecer sugestões, para uma melhor realização de tais tarefas. WOOLDRIDGE [17] diz que agentes de interface podem, portanto, receber informações ou instruções de usuários, também aprendendo a partir de uma realimentação oriunda dos usuários.

Pelas palavras de COPPIN [16], em geral, é de grande importância que tarefas repetitivas sejam delegadas a um agente de interface, pois ele é capaz de aprender observando

a tarefa que o usuário procede, a fim de realizá-la de forma autônoma no futuro. Na explicação de PATTIE, M. e ROBYN, K. [19], um agente é capaz de auxiliar determinado usuário a organizar reuniões em um calendário, marcando, desmarcando, reagendando essas reuniões com outras pessoas em seu nome. Observando o comportamento do usuário, ele é capaz de aprender em quais horários o mesmo não gosta que sejam marcadas reuniões, evitando, por seu turno, tais marcações.

2.1.1.2. Agentes móveis

Na definição sucinta de COPPIN [16], agentes móveis são aqueles capazes de, literalmente, moverem-se, ou seja, terem uma movimentação física de um local para outro. No caso de agentes móveis, pode-se exemplificar com robôs móveis que, realmente, têm alguma movimentação física. WOOLDRIDGE [17] trata do caso de agentes móveis de *software*, quando a mobilidade, geralmente, está atrelada à *Internet*, ou ainda a outra rede. Nesta definição, verifica-se a consistência da exemplificação de COPPIN [16], uma vez que a afirmativa ocupa-se de que agentes móveis viajam de um computador para o outro, coletando informações e realizando ações, conforme necessário.

No exemplo de COPPIN [16] é evidenciado que um vírus de computador pode ser considerado um agente móvel, mesmo que a grande maioria dos vírus não seja inteligente, mas, sim, meramente autônomo. Dessa forma, os mesmos desenvolvem a capacidade de agir sem receberem instruções diretas de um usuário, todavia não têm a autonomia de adaptação aos ambientes ao seu entorno.

WOOLDRIDGE [17] trata da evidência que chama atenção, haja vista que para os agentes móveis serem executados remotamente em computadores hospedeiros, devem fornecer um ambiente adequado, serem executados ou se autoexecutarem.

Controvérsias e dúvidas surgem quanto à segurança perante as ideias de que agentes podem ser enviados a partir de um computador para a *Internet*, a fim de que possa ser executado em um computador remoto. Nesse contexto, questões de segurança e privacidade passam a ser levantadas e discutidas, contudo, para COPPIN [16], existem vantagens na utilização de agentes móveis:

- Eficiência: No caso de um agente ter a necessidade de se comunicar com uma série de servidores remotos e solicitar grandes quantidades de informação para tomar uma decisão, no caso de uso de agentes móveis, todo esse encargo de solicitação e transferência fica dispensado, pois o agente irá ser executado, localmente, no servidor remoto.

- Utilização para a geração de arquiteturas distribuídas de computação: Nessa perspectiva ocorre computação em múltiplos computadores em locais arbitrários.

- Realização de tarefas de maneira assíncrona: O agente pode ser desligado, arbitrariamente pelo usuário, quando o mesmo estiver pronto para processar ou receber o resultado, poderá ser invocado ou chamado novamente.

2.1.1.3. Agentes Reativos Simples

RUSSELL e NORVIG [14] tratam esta categoria de agentes como sendo uma das mais simples. Estes selecionam ações como base na percepção do que está ocorrendo atualmente, em que o restante do histórico de percepção é ignorado. Segundo RUSSELL [20] comportamentos reativos simples ocorrem, também, em ambientes mais complexos.

Um Agente Reativo Simples, consoante COPPIN [16], também é conhecido como agente de reflexo, e pode ser definido como um sistema de produção onde entradas geradas pelo ambiente são comparadas com regras para determinar que ação deve ser executada. Como WOOLDRIDGE [17] trata, em outras palavras, agentes reativos simplesmente reagem a eventos em seu ambiente, de acordo com regras predeterminadas.

Segundo COOPIN [16], agentes podem ter diversas outras propriedades, como: a versatilidade, ou seja, sendo capazes de realizar muitos tipos de tarefas; a não cooperação também é característica presente em agentes; antagonista ou altruísta; ou, ainda, ter uma grande mobilidade.

Ainda em conformidade com RUSSELL [20], o pseudocódigo da Figura 3, sendo especificado por um ambiente, neste exemplo, um aspirador de pó, a função recebe a posição e situação, efetivando determinada ação. Esta ação também pode ser chamada de regra situação-ação, regra de produção, ou regra de “se então”.

Se a situação for igual a sujo então ele retorna Aspirar, posteriormente testa, se a posição for igual a ‘A’ retorna para o *software* a instrução de posicionamento à direita; se não, testa novamente, se a posição for igual a ‘B’, retorna para o *software* a instrução de posicionamento à esquerda.

1	Função Agente-Aspirador (posição, situação) retorna uma ação
2	se situação = Sujo então retorna <i>Aspirar</i>

- | | |
|---|--|
| 3 | senão se posição = A então retorna <i>Direita</i> |
| 4 | senão se posição = B então retorna <i>Esquerda</i> |

Figura 3: Programa do agente reativo simples Agente-Aspirador. Adaptado de RUSSELL [20]

A Figura 4 fornece a estrutura do *software*, gerando a estrutura deste programa geral em forma esquemática, mostrando como as regras e condições ou as regras “se então” permitem ao agente fazer a percepção e a ação.

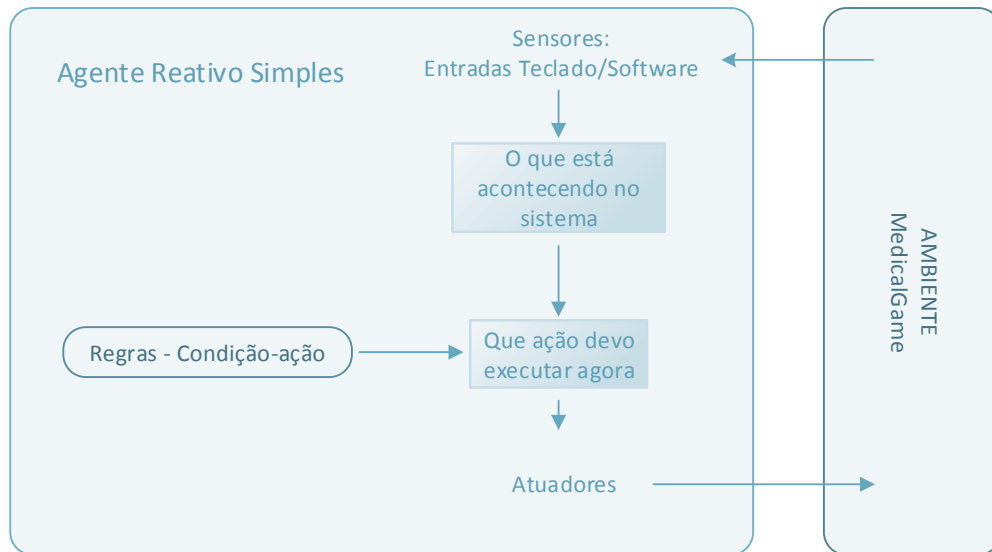


Figura 4: Diagrama esquemático agente reativo simples. Fonte: Adaptado de RUSSELL [20]

Na Figura 5, a função interpreta a entrada, gera uma decisão abstrata do estado atual a partir da percepção; enquanto a função regra-correspondente retorna à primeira regra no conjunto de regras que corresponde à descrição do estado dado. Desta forma, observa-se que a descrição em termos de “regras” e “correspondências” é puramente conceitual. As implementações reais podem ser tão simples quanto uma coleção de portas lógicas que implementam um circuito booleano.

- | | |
|---|---|
| 1 | função AGENTE-REATIVO-SIMPLES (percepção) retorna uma ação |
| 2 | variáveis estáticas: <i>regras</i> , um conjunto de regras condição-ação |
| 3 | <i>estado</i> – INTERPRETAR-ENTRADA (<i>percepção</i>) |
| 4 | <i>regra</i> – REGRA- CORRESPONDENTE (<i>estado, regras</i>) |
| 5 | <i>ação</i> – AÇÃO DA REGRA [<i>regra</i>] |
| 6 | retorna <i>ação</i> |

Figura 5: Um agente reativo simples. Fonte: Adaptado de RUSSELL [20]

2.1.1.4. Agentes de Informação

Na concepção de WOOLDRIDGE [17], agentes de informação, também conhecidos como agentes de coleta de informações, são, comumente, utilizados para a coleta de informações na *Internet*, vindo a ser chamados ou designados ainda de agentes de *Internet*. COPPIN [16] assevera que um agente de informação é usado para prestar ajuda ao usuário, encontrar, filtrar e classificar informações oriundas de diversificadas e vastas fontes disponíveis na *Internet*. Nesta afirmação, COPPIN [16] ainda trata de algumas características peculiares dos agentes de coleta de informação estes:

- Podem ser subdivididos em móveis e estáticos;
- Alguns agentes de informação são capazes de aprender, enquanto o comportamento de outros agentes é fixado;
- Agentes podem ser colaborativos com outros agentes ou trabalharem independentemente;
- A característica que distingue um agente de informação é a função que ele fornece e não o modo como ele funciona.

COPPIN [16] garante que os agentes de informação sabem como realizar buscas na *Internet*, em geral, utilizam-se de uma grande gama de ferramentas de busca, maximizando, assim, sua recuperação, porém, neste caso, um problema fica enfatizado, a precisão, para que esta seja melhorada, os agentes dependem de treinamento efetivado pelo usuário.

2.2. SISTEMAS TUTORES INTELIGENTES – STI

A incorporação das tecnologias aos processos pedagógicos é o que se pode chamar de informática educacional. Sendo assim, a importância do uso de computadores e das novas tecnologias no ensino, deve-se não somente aos impactos dessas ferramentas na sociedade, mas, também, às novas exigências culturais e sociais que se impõem. Segundo a UNESCO [21], as TICs exercem um papel cada vez mais importante na forma de comunicar, aprender e viver de todos. O grande desafio é equipar essas tecnologias efetivamente, de forma a atender aos interesses dos aprendizes e da grande comunidade de ensino-aprendizagem.

Nesse sentido, é importante citar os Sistemas Tutores Inteligentes (STIs), que são *softwares* que auxiliam o aluno em sua aprendizagem. Segundo BERCHT [12], os STIs são

sistemas capazes de, através da interação com o usuário, atualizar sua própria base de conhecimento. Assim sendo, são **capazes** de reconhecer o estado atual do aluno, diagnosticar lacunas em seu conhecimento e, com o desenrolar do diálogo, aprender, adaptar e aplicar diferentes tipos de estratégias de aprendizagem.

Posteriormente, elencar-se-á a definição da Arquitetura dos Sistemas Tutores Inteligentes, que fazem parte deste escopo. Em seguida será definida a arquitetura clássica do STI, com os modelos do: Aluno, Pedagógico, Domínio e Interface.

2.2.1. Arquitetura dos Sistemas Tutores Inteligentes

A arquitetura tratada como clássica é também conhecida como função tripartida ou arquitetura tradicional de um STI. Pela definição de WENGER [22], esta proposta trouxe grandes avanços à modelagem de ambientes educacionais, permitindo que estratégias de ensino fossem associadas em função das informações oriundas do Modelo do Aluno.

Em sua grande maioria, as arquiteturas, propostas para STIs, possuem quatro componentes essenciais, definidos por FREEMAN [23], como: Modelo do Aluno; Modelo do Domínio; Modelo Pedagógico e Modelo da Interface. Outro ponto importante a ser tratado é de que RUSSELL e NORVIG [14] definem que a adição de mais módulos a essa arquitetura básica dependerá, basicamente, do domínio em que o STI está sendo modelado.

Diante desta perspectiva, a arquitetura apresentada na Figura 6 é conhecida tanto como função tripartida quanto arquitetura tradicional de um STI. A designação de função tripartida dar-se-á pela associação do termo tripartido aos modelos pedagógico, aluno e do domínio. Dessa forma, o principal objetivo do STI é proporcionar ensino adaptado a cada aluno.

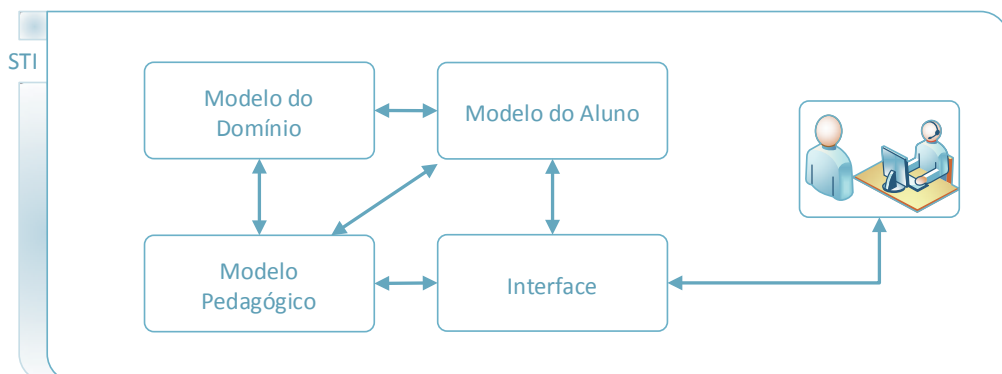


Figura 6: Arquitetura Clássica de um STI. Adaptado de RUSSELL e NORVIG [14].

Conforme citado anteriormente, esta arquitetura é composta, basicamente, pelos seguintes modelos:

- **Modelo do Aluno:** Como característica intrínseca este modelo armazena e modela as características individuais de cada aluno.
- **Modelo do Domínio:** Este modelo contém e detém o conhecimento sobre o conteúdo a ser tratado, no formato de regras de produção bem definidas.
- **Modelo do Pedagógico:** Neste modelo são implementadas as estratégias e táticas para serem selecionadas em função das características de cada aluno, estas representadas no Modelo do Aluno.
- **Modelo da Interface:** Caracteriza-se por fazer o intermédio das interações entre o Modelo Pedagógico e o Modelo do Aluno.

Com estas definições, pretendeu-se demonstrar as atividades desempenhadas por cada modelo dentro da arquitetura clássica de um STI, que serão descritas com mais detalhes nas seções posteriores.

2.2.1.1. Modelo do Aluno

Tratando-se do Modelo do Aluno, GREENO ET. AL. [24] afirma que os STIs são capazes de responderem ao estilo individual de aprendizado de cada aluno, para a distribuição de instruções sob medida para determinados momentos.

Segundo MITCHELL [25], um STI deve modelar o ambiente, o aprendiz e a interação professor/aluno. Destarte, as representações das habilidades cognitivas e de conhecimento do aluno estão representadas neste módulo. Conforme ponderações de VICCARI [26], será de fundamental importância para o tutor comprovar as hipóteses a respeito do aluno em questão, este ambiente ou modelo contém uma representação do estado de conhecimento no momento em que o aluno interage com o STI.

Na concatenação e cruzamento de informações oriundas do modelo e do conteúdo a ser tratado, o sistema deve ser capaz de inferir estratégias em seguida. Um modelo realista do aluno implica em atualização dinâmica à medida que o sistema avalia e trata o desempenho individual de cada aluno.

2.2.1.2. Modelo do domínio

Inicia-se, aqui, a definição do modelo de domínio do STI, que, para WOOLDRIDGE [17], é o componente especialista do tutor, constituído pelo material de estudos da área em foco, pela sistemática de geração de exemplos, pela formulação de diagnósticos e pelos processos de simulação. Contém estruturas concisas sobre o domínio que se deseja ensinar ao aluno.

Diversos modelos de representação de conhecimento podem ser utilizados dentro deste contexto, RUSSELL e NORVIG [14] citam alguns destes modelos de representação do conhecimento, como: redes semânticas, frames, scripts, regras de produção e programação orientada a objetos (OOP).

2.2.1.3. Modelo Pedagógico

Na afirmação de VICCARI [26], os modelos pedagógicos contêm as estratégias e as táticas de ensino. As estratégias constituem conhecimento sobre como ensinar, ou seja, como gerar, a partir das informações de diagnóstico, monitoramento e análise, uma sequência de táticas de ensino capazes de apresentar, com sucesso, um determinado tópico a um determinado aluno.

Segundo RUSSELL e NORVIG [14], assim como na interpretação de VICCARI [26], uma estratégia de ensino deve definir ou responder os dois próximos questionamentos: Quando efetivar a interrupção do raciocínio do aluno? Como interromper?

Um dos métodos muito utilizado pelos STIs é o chamado método socrático, que consiste em partir do conhecimento e domínio que o aluno detém, em que o STI efetiva uma troca de informações, levando o aluno a tomar as suas próprias decisões.

2.2.1.4. Modelo da Interface

A definição de modelo da interface é dada por OREY [27], RUSSELL e NORVIG [14], onde é, de comum acordo, que uma boa interface é vital para o sucesso de qualquer sistema interativo, inclusive os STIs. A importância neste modelo do sistema cresce bastante, uma vez que, além de suas atribuições, representa o material institucional da empresa ou até mesmo a instituição.

PÁDUA [28] define alguns objetivos comuns aos STIs, dentre eles a necessidade de evitar que o aluno se entedie diante da resolução de exercícios, ou seja, é necessário ter um

material bem elaborado; é desejável que o aluno possa intervir na tutoria; o tempo de resposta deve permanecer dentro do esperado pelo usuário; o monitoramento deve ser efetivado em sua grande maioria em *background*. Sabe-se que o usuário aprende a relação da interface junto ao conteúdo, conforme PÁDUA [28], a carga cognitiva adicional nestes tipos de *softwares* deve ser mínima.

Dentro deste contexto, observa-se que os STIs exploram a autonomia do sistema na autorrelação com o aluno, que, em grande parte, a interação é definida pelo sistema sem a necessidade de intervenção humana. Logo, os STIs têm a capacidade de interagir e podem ser utilizados em qualquer área e nível de conhecimento, tendo como base o sistema de apoio no processo de ensino aprendizagem.

2.3. TRABALHOS RELACIONADOS

Nesta seção serão apresentados outros trabalhos que utilizam agentes de *software*, em sistemas aplicados a área educacional. Para tanto, primeiramente fez-se a pré-seleção de trabalhos nas bases de periódicos científicos, em que foram efetivadas pesquisas utilizando-se os seguintes termos, respectivamente: (agentes de *software* no ensino, sti + agentes de *software*, agentes inteligentes, agentes de *software* + medicina). Neste estudo inicial foram encontrados aproximadamente 70 trabalhos que continham os termos de pesquisa supracitados. Buscou-se, a partir da leitura e de uma análise mais criteriosa, a identificação de elementos que aproximassem esses trabalhos com o proposto nesta pesquisa. Os trabalhos relacionados para serem destacados nesta seção foram o de LYRA [29] e KAUTZMANN e JAQUES [30] especificamente.

O trabalho de LYRA [29] trouxe um importante elemento como contribuição para o desenvolvimento de agentes, que diz respeito à utilização da linguagem de programação PHP para a estruturação dos mesmos, enquanto a grande maioria dos trabalhos estudados durante este processo faz menção e utilizam *frameworks* para este tipo de implementação.

2.3.1. Agentes de *Software* no Monitoramento de Alunos em Educação a Distância

A proposta de LYRA [29], em seu trabalho, é a utilização de agentes de *softwares* para desempenharem papéis importantes dentro de ambientes educacionais, por exemplo: as monitorias de entrega de atividades, e assiduidade de seus participantes.

LYRA [29] utilizou-se de uma plataforma pronta e gratuita, denominada *Moodle*, por possuir *plugins* que facilitaram a implantação dos agentes propostos, e, também, por utilizar a mesma linguagem de programação em que os agentes foram criados, ou seja, PHP. Foi utilizada, ainda, a estrutura de armazenamento de dados do sistema *Moodle* para consulta de dados, referentes a acessos e utilização de determinado curso, por seus alunos. Esses dados geram as informações necessárias para que os agentes conseguissem realizar suas tarefas, na forma que foram programados.

Na Figura 7 estão representados os dois agentes, o Agente Coletor e Agente Executor, criado por LYRA [29], descritos respectivamente: O Agente Coletor: este é responsável por monitorar a estrutura do banco de dados onde são armazenados os dados referentes à utilização do sistema. Este agente, em sua estrutura interna, utiliza *queries* com critérios pré-configurados, estas *queries* interagem com a camada de banco de dados, retornando registros com informações relevantes para a criação das mensagens que posteriormente serão enviadas; O Agente Executor, segundo agente da implementação, é responsável por monitorar a estrutura do banco de dados que contem as informações geradas pelo Agente Gerador, se houverem requisições de envio de mensagens, o mesmo trabalha neste envio.

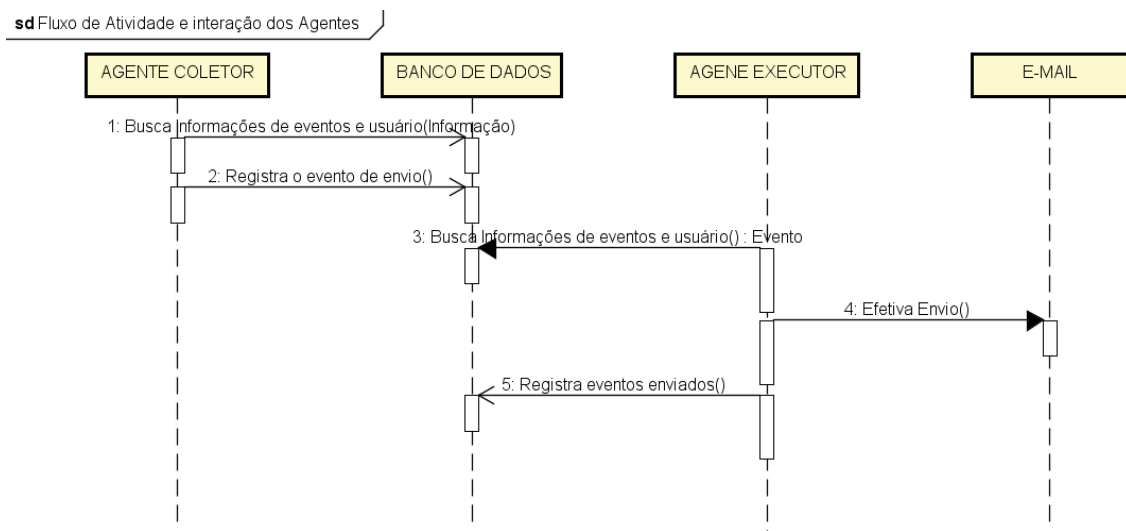


Figura 7: Fluxo de Atividade de Agentes. Fonte: Adaptado de LYRA [29].

LYRA [29] conclui seu trabalho declarando que a tecnologia de agentes mostrou-se adequada para o cumprimento dos objetivos iniciais do projeto, surgiram dificuldades quanto à programação e configuração dos mesmos, com características de *plugins* que, posteriormente, foram utilizados na plataforma *Moodle*.

Descreve que a escrita de um *plugin* utilizando a estrutura de agentes, teve como proposta a solução de duas lacunas em plataformas EAD, contribuiu para superar aspectos críticos destas plataformas, deixando claro que os agentes desenvolvidos podem auxiliar na aproximação de professores e alunos, estimulando a participação frequente destes alunos em atividades *on-line*.

2.3.2. Um modelo de treinamento adaptativo da habilidade metacognitiva de monitoramento do conhecimento.

O trabalho de KAUTZMANN e JAQUES [30] apresenta o modelo e a implementação de um agente que treina a habilidade de monitoramento do conhecimento através de uma instrução que se adapta às características do aprendiz e aos históricos de resolução de tarefas. O agente criado foi integrado a um STI e, na sequência, efetivada uma avaliação experimental. Nesta avaliação, evidências positivas foram apresentadas com relação aos benefícios do uso de agentes metacognitivos.

KAUTZMANN e JAQUES [30] utilizaram-se do STI PAT2Math para incorporar e efetivar avaliações, citam ainda que seu agente pode ser incorporado em qualquer outro STI. Dentro deste contexto, os autores descrevem que um agente que treina a habilidade de monitoramento do conhecimento de forma explícita, através de reflexões, e que adapta suas ações as características do aluno, faz com que ele reflita mais sobre seu conhecimento e melhora, assim, suas habilidades.

O agente desenvolvido por KAUTZMANN e JAQUES [30] tem a ação de reflexão, também chamada de *scanffolding*, pois prestam assistência, e essa é reduzida conforme o aprendiz melhora sua habilidade metacognitiva. O *scanffolding* é adaptado segundo o nível metacognitivo corrente do aluno, o histórico de solução de tarefas, bem como o conhecimento do aluno no domínio que está sendo envolvido.

Este agente emprega três tipos de *scanffolding*, sendo eles: *prompts*, *feedbacks* e *self-explanations*. Sendo que os *prompts* ocorrem antes que o aluno tente resolver um novo passo de tarefa, e incitam o aluno a refletir sobre seu conhecimento; Os *feedbacks* são mensagens textuais que notificam o aluno sobre seu nível corrente da habilidade de monitoramento de conhecimento, ou ainda sobre algum comportamento inadequado; E o *self-explanations* incita o aluno a escrever, com suas palavras, sobre como ele monitorou seu conhecimento.

KAUTZMANN e JAQUES [30] descrevem que, no agente desenvolvido, há dois ciclos sendo eles: o ciclo interno e o ciclo externo; O ciclo externo é o responsável pela ativação do ciclo interno. O ciclo interno tem a responsabilidade do treinamento metacognitivo, entrando em funcionamento antes de qualquer ação do aluno.

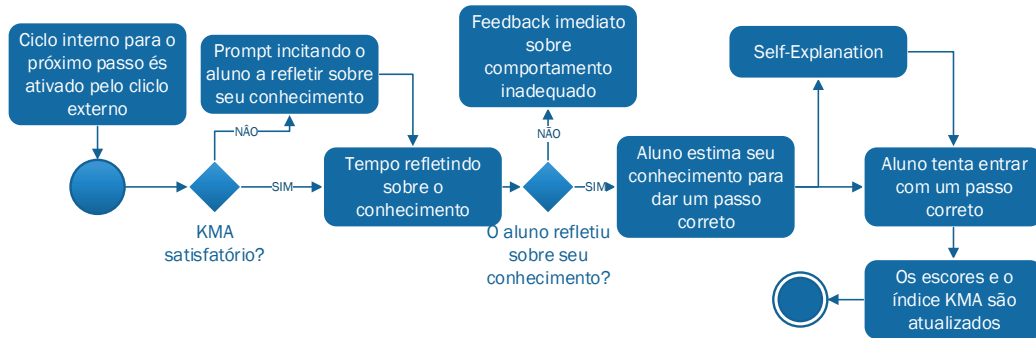


Figura 8: Fluxo de funcionamento do ciclo interno. Fonte: Adaptado de KAUTZMANN e JAQUES [30]

Na Figura 8 é apresentado o fluxo de funcionamento interno do agente metacognitivo desenvolvido. No fluxo demonstrado, o mecanismo verifica o índice KMA, que é um índice calculado internamente pelo sistema. Caso este índice seja insatisfatório, é selecionado um *prompt* que incita o aluno a refletir sobre seu conhecimento para avançar nos passos de resolução de problemas. Outra característica descrita pelo autor é a tentativa do aluno efetivar, ou tentar entrar em passos, muito rapidamente, sem utilização do passo chamado reflexão, um *feedback* imediato é exibido na interface, informando o comportamento inadequado.

Posteriormente, o aluno deve estimar seu conhecimento, iniciando um novo passo de resolução de tarefas. Neste momento o agente compara e estima o desempenho e o índice KMA do aluno. Como próximo passo, o agente poderá entregar atividades *self-explanation*, exibir *feedbacks* e diminuir, gradativamente, as interações com o *prompt* metacognitivo.

No trabalho, o ciclo externo é responsável pela ativação do ciclo interno, para o próximo passo da tarefa, este é executado sempre antes do aluno entrar com um passo. Duas decisões são utilizadas para a tomada de decisão quanto à ativação do ciclo interno – Decisão por nível metacognitivo, baseada em cálculos e índices internos do sistema, e decisão por conhecimento do aluno no domínio que ele está trabalhando.

KAUTZMANN e JAQUES [30] buscam treinar a habilidade de monitoramento do conhecimento através de uma instrução que incita o aluno na reflexão sobre seu conhecimento demonstrando, assim, a importância da habilidade metacognitiva.

Discutem-se, agora, algumas das principais características entre os trabalhos relacionados e o trabalho desenvolvido por esta pesquisa. Esta comparação pode ser visualizada em forma tabular na Tabela 1.

A primeira característica a ser destacada é a de utilização de sistema base, que tem o intuito de rotular qual STI foi utilizado para o acoplamento dos agentes criados. No trabalho de LYRA [29], a plataforma *Moodle* foi utilizada como base para a criação e acoplamento dos agentes desenvolvidos. No trabalho de KAUTZMANN e JAQUES [30], o agente por eles desenvolvido trabalhava integrado com o STI PAT2Math. E no trabalho aqui desenvolvido, criou-se um STI, denominado *MedicalGame*, no qual os agentes foram desenvolvidos e devidamente acoplados.

A segunda característica levada em consideração, na comparação dos trabalhos, foi a dos agentes possuírem a possibilidade de serem configurados através da aplicação desenvolvida. No trabalho de LYRA [29] somente textos de respostas dos agentes podem ser configurados. Já no trabalho de KAUTZMANN e JAQUES [30] nenhum tipo de configuração foi criado para ser utilizada pelos administradores do sistema. Em contrapartida, o trabalho proposto e desenvolvido na presente investigação possibilita a configuração da totalidade dos agentes implantados no sistema.

A terceira característica é a de qual tipo de agente cada autor utilizou em seu estudo, sendo que LYRA [29] trabalha especificamente com agente reativo simples; KAUTZMANN e JAQUES [30] trabalham com agentes metacognitivos e, neste trabalho, foram desenvolvidos agentes reativos simples.

Como quarta característica, a quantidade de agentes utilizadas. LYRA [29] trabalha com dois agentes, KAUTZMANN e JAQUES [30] trabalham com um único agente, enquanto neste trabalho foram desenvolvidos quatro agentes, denominados: Agente de nivelamento, Agente de alocação, Agente de tutoria, Agente duelo.

Como quinto item a ser tratado – uma comparação sobre o domínio das aplicações apresentadas na Tabela 1. Em comparação: LYRA [29] apresenta uma solução que pode ser utilizada para qualquer domínio; já KAUTZMANN e JAQUES [30] têm uma redução significativa no domínio em que a aplicação executa, pois a mesma trabalha única e exclusivamente com a resolução de funções do segundo grau; por sua vez, o estudo desenvolvido trabalha no domínio da área de medicina, especificamente.

Como sexta característica, as estratégias de ensino disponíveis dentro de cada trabalho. Quanto a esta propriedade: para LYRA [29] seus agentes efetivam somente o envio de *e-mails*, que servem principalmente para a organização particular dos alunos, e para atingir

maior engajamento dentro do curso a que este aluno está participando. Já no trabalho de KAUTZMANN e JAQUES [30] são utilizados *feedbacks*, tal como interação de escrita do aluno com o *software*. Todavia, no trabalho aqui apresentado, têm-se algumas estratégias de ensino embutidas na aplicação, sendo elas: *Feedback*, tutorias em todos os níveis da aplicação, vídeo embutidos em perguntas e tutorias, áudio embutido em perguntas e tutorias, reforço. E se acaso o aluno não conseguir efetivar pontuação suficiente para avançar de nível, o sistema continua trabalhando com questões daquele nível. Por fim o ranqueamento dos 10 primeiros colocados em cada nível.

Tabela 1: Quadro comparativo, trabalhos relacionados. Fonte: Elaborada pelo autor.

Autores	LYRA [29]	KAUTZMANN e JAQUES [30]	André Luís Stefanello
Utilização de Sistema Base.	SIM	SIM	NÃO
Sistema de Configuração dos Agentes.	SIM	NÃO	SIM
Tipos de Agentes Utilizados.	Reativo Simples	Metacognitivo	Reativo Simples
Quantidade de Agentes.	2	1	4
Domínio da Aplicação.	Qualquer área	Equações do Segundo Grau	Medicina
Estratégias de Ensino Disponíveis.	Mandar e-mail	<i>Feedbacks</i> /escrita aluno	<i>Feedback</i> , tutoria, vídeo, áudio, reforço e ranqueamento

3. METODOLOGIA DE TRABALHO

Este trabalho se enquadra, quanto à metodologia, como sendo uma pesquisa tecnológica, que condiz com o tipo de pesquisa científica aplicada, obtendo como produto final, neste caso, um *software*. Quanto à forma de abordagem, esta pesquisa é qualitativa.

Em relação aos procedimentos, esta pesquisa se classifica como bibliográfica. Inicialmente, neste trabalho, parte-se de pesquisas já efetivas e produzidas, como no caso livros e revistas científicas. Esse é o tipo de pesquisa para consolidar o conhecimento, e é a primeira parte da pesquisa que deve ser efetivada em um processo científico MEDEIROS [31].

Esta pesquisa, outrossim, na sequência, se enquadra como estudo de caso, visto que são provocadas alterações no ambiente alvo de pesquisa, e as interações realizadas são observadas em busca de identificar se estão produzindo os resultados esperados WAZLAWICK [32].

O desenvolvimento do trabalho foi organizado em etapas, que podem ser visualizadas na Figura 9. Cabe aqui ressaltar que foi elaborado um termo de consentimento disponível para consulta no Apêndice A - Termo de Consentimento. Outro documento importante está disposto no Apêndice D - A estória do cenário de testes do sistema MedicalGame, utilizada para repassar aos alunos quais passos deveriam ser seguidos na utilização do *software* MedicalGame.

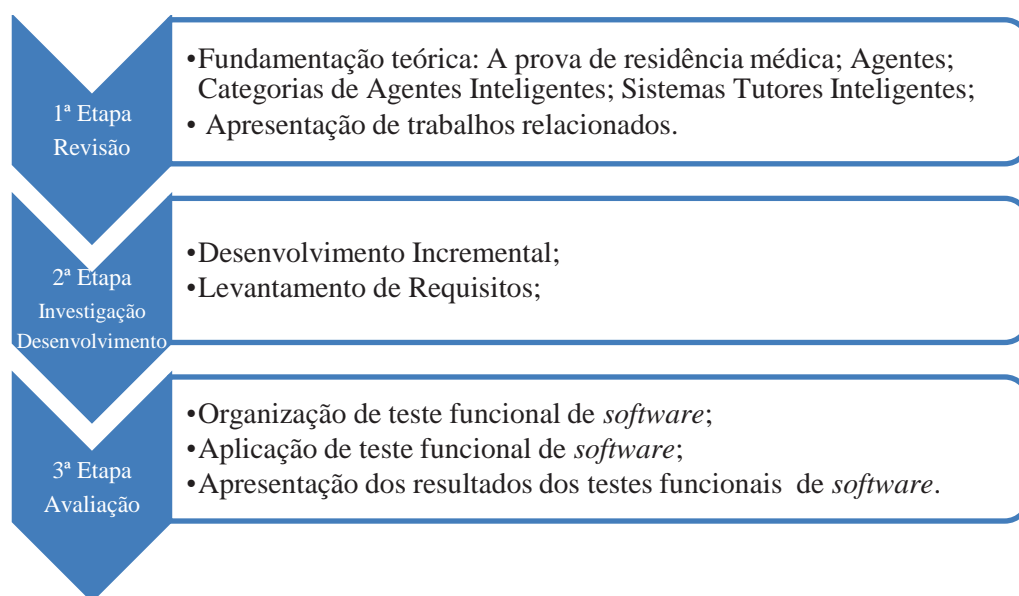


Figura 9. Etapas propostas para a execução do trabalho. Fonte: Elaborado pelo autor.

3.1. PROVA DE RESIDÊNCIA MÉDICA

O objetivo desta seção é descrever o surgimento da prova de residência médica, da mesma maneira de onde é aplicada.

Segundo o MINISTÉRIO DA EDUCAÇÃO [1], instituída pelo Decreto nº 80.281, de 5, de setembro de 1977, a residência médica é uma modalidade de ensino de Pós-Graduação, destinada a médicos, sob a forma de curso de especialização. Funciona em instituições de saúde, sob a orientação de profissionais médicos, de elevada qualificação ética e profissional. O mesmo decreto criou a Comissão Nacional de Residência Médica (CNRM). O Ministério da Educação considera que, o Programa de Residência Médica, cumprido integralmente dentro de uma determinada especialidade, confere ao médico residente o título de especialista.

CHAVES [2] cita que o decreto nº 7.562/2011 trata da composição da prova de residência médica, que foi modificada pela Resolução 03/2011, estabelecendo uma primeira fase obrigatória, consistindo em um exame objetivo, com peso mínimo de 50%. A segunda fase optativa deverá ser constituída de prova prática, com peso de 40% a 50% da nota total, destinando, ainda, a critério da instituição, a análise curricular no valor de 10% da nota total. A definição dos critérios de pontuação na análise curricular é critério de cada órgão coordenador da residência médica em cada Estado.

3.2. 1ª ETAPA - REVISÃO

A etapa de fundamentação teórica consistiu na revisão de textos, artigos, livros, revistas, que tangem os assuntos: A prova de residência médica; Agentes; Categorias de Agentes Inteligentes; e Sistemas Tutores Inteligentes. Esta etapa foi efetivada e completada dentro do capítulo 2 deste trabalho.

3.3. 2ª ETAPA – INVESTIGAÇÃO E DESENVOLVIMENTO

A segunda etapa, denominada Desenvolvimento Incremental e Levantamento de Requisitos, será descrita em dois passos, sendo: o primeiro passo denominado desenvolvimento de *software*, este obedeceu à arquitetura de construções de *software*,

baseado em métricas de Engenharia de *Software*¹, que têm como proposta e padrão o que é apresentado na Figura 10.

O desenvolvimento incremental tem como premissa o desenvolvimento de uma estrutura inicial, baseada na descrição de um esboço oriundo de um cliente, expô-la aos comentários dos usuários/solicitantes e dar continuidade com a criação de várias versões, até que um sistema adequado seja desenvolvido. Atividades de especificação, desenvolvimento e validação são intercaladas e separadas por pequenos *feedbacks* entre todas as atividades.

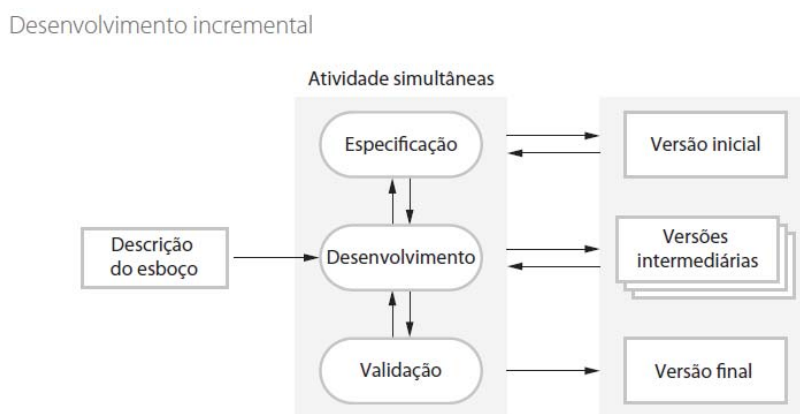


Figura 10 Processo de Engenharia de *Software* – Desenvolvimento Incremental. Fonte: Adaptado de SOMMERVILLE [33].

O desenvolvimento incremental de *software* é o que melhor reflete a maneira como se podem resolver problemas, muito raramente consegue-se elaborar uma completa solução do problema com antecedência, na maioria das vezes resolve-se passo a passo. Em um processo de entrega incremental, segundo SOMMERVILLE [33], os clientes identificam, em linhas gerais, os serviços a serem fornecidos pelo sistema. Identificam também os serviços que são mais importantes e a prioridade de entrega destas solicitações.

Dessa forma, iniciou-se o processo de elaboração dos *softwares*, em uma primeira reunião organizada, que ocorreu no dia 23 de Abril de 2015, na UPF de Passo Fundo. nela participaram dois médicos da UPF, um aluno do curso de Medicina da UPF, o orientador deste trabalho e o pesquisador.

Seguindo o escopo tratado no processo de desenvolvimento de *software* proposto, principiou-se o esboço dos *softwares* a serem desenvolvidos, bem como o agendamento de reuniões mensais, com duração média de 50 minutos, presididas pelo orientador do trabalho. As reuniões sempre contaram a participação de, ao menos, um membro do curso de Medicina,

¹ Engenharia de *Software* é uma área da computação voltada à especificação, desenvolvimento, manutenção e criação de sistemas de *software*, com aplicação de tecnologias e práticas de gerência de projetos e outras disciplinas, visando organização, produtividade e qualidade. Conforme SOMMERVILLE [33]

orientador e pesquisador. As reuniões objetivavam a organização e cumprimento dos fluxos das atividades de desenvolvimento, propostas no modelo de desenvolvimento incremental, apresentadas na Figura 10.

A cada nova reunião a descrição do esboço foi sendo melhorada, tal como o desenvolvimento, a especificação do *software*, sua validação, pela constante participação de um aluno e um professor do curso de Medicina da UPF.

Posterior ao processo de especificação, desenvolvimento e validação, novas funcionalidades eram incrementadas aos *softwares* desenvolvidos neste trabalho, chegando-se a uma versão estável da aplicação de configuração do STI, ou seja, a aplicação administrativa, que pôde receber carga de dados suficientes para seu propósito, uma versão estável da aplicação MedicalGame, além da página institucional do trabalho.

Como segundo passo do processo de investigação, descreveu-se o documento de requisitos do *software* MedicalGame (disposto no Apêndice C -), que de acordo com SOMMERVILLE [33], normalmente é chamado de especificação de requisitos de *software*. É uma declaração oficial do que os desenvolvedores do sistema devem programar. Este documento deve incluir tanto requisitos de usuário para um sistema, quanto uma especificação detalhada dos requisitos de sistema. Como documento final do passo de implementação, chegou-se a um documento de requisitos, que contempla as solicitações efetivadas pelos envolvidos, que é o documento final do processo ou documento de requisitos, assim denominado no processo de engenharia de *software*, (disponível no Apêndice C -).

3.4. 3ª ETAPA – AVALIAÇÃO

Na etapa definida como avaliação, houve o envolvimento dos professores, orientador e coorientador trabalho, o autor e alunos vinculados ao curso de Medicina da UPF - FAMED.

Esta etapa, designada avaliação, ou na literatura referente à engenharia de *Software*, teste, para SOMMERVILLE [33], é destinado a mostrar que um programa faz o que se propõe a fazer e descobrir os defeitos do programa antes do uso. Os resultados do teste são verificados à procura de erros, anomalias ou informações sobre os atributos não funcionais do programa.

Dentro do escopo de testes de *software* existem quatro subdivisões – testes de desenvolvimento, desenvolvimento dirigido a testes, testes de *release* e testes de usuário. Discute-se, aqui, o teste de *release*, utilizado no trabalho desenvolvido, e que, segundo

SOMMERVILE [33], é o processo de testar um *realease* em particular de um sistema que se destina para uso fora da equipe de desenvolvimento, geralmente, o *realease* é para uso dos clientes e usuários. Testes de *realease* costumam ser um teste de caixa-preta, cujo comportamento só pode ser denominado por meio das entradas e saídas relacionadas. Outro nome para isso é “teste funcional”, assim chamado porque o testador só está preocupado com a funcionalidade, e não com a implementação do *software*.

Dentro da proposta de testes de *realease* descritos anteriormente, existem subdivisões distintas: Testes baseados em requisitos; Testes de cenário; Testes de desempenho. No trabalho desenvolvido, utilizou-se o Teste Baseado em Cenário, que SOMMERVILE [33] trata como uma abordagem de testes de *realease* em que são imaginados cenários típicos de uso e os usa para desenvolver casos de teste para o sistema, ou seja, um cenário é uma estória² que descreve a maneira de usar o sistema.

Por conseguinte, a etapa de avaliação foi subdividida em **três processos**, sendo eles: a organização de teste de *realease* ou teste funcional de *software*; Aplicação do teste de *realease* ou teste funcional de *software*; Avaliação e organização dos resultados do teste de *realease* ou teste funcional de *software*.

O **primeiro passo** – organização de teste de *realease* ou teste funcional de *software* – foi desenvolvido somente para a aplicação MedicalGame, que tem em sua implementação os agentes de *software*, item importante para este trabalho. A partir disso, escreveu-se uma estória de como o sistema MedicalGame pode ser utilizado normalmente, documento encontra-se disponível no Apêndice D - ESTÓRIA CENÁRIO DE TESTES MEDICALGAME. Como descrito no apêndice citado anteriormente, muitas funcionalidades do sistema podem ser testadas a partir da descrição deste cenário, mas apenas 4 funcionalidades tiveram dados tabulados e apresentados: Funcionamento do Agente de Nivelamento; Funcionamento do Agente Desafio; Funcionamento do Agentes de Alocação; Funcionamento do Agentes de Tutoria.

Como **segundo passo** – a aplicação do teste de *realease* ou teste funcional de *software* – ocorreu na FAMED, Faculdade de Medicina da UPF, com o acompanhamento do orientador e coorientador do trabalho, bem como o aluno pesquisador.

E como **terceiro passo** – a avaliação e organização dos resultados do teste de *realease* ou teste funcional de *software*. Os dados de entrada e saída do sistema foram

² Estória: Forma de definir e especificar um processo de *software* em uma lista de funções que o sistema deve satisfazer. SOMMERVILE [33].

organizados e tabulados utilizando-se *scripts* SQL, no banco de dados da aplicação, também uma planilha do *Excel* para a geração e organização de gráficos e estatísticas.

O Estudo de caso – teste funcional de *software* para os agentes de *software* – proposto é do tipo descritivo, em que o foco está em descrever determinadas intervenções e o contexto em que elas ocorrem, desenvolvendo e interpretando a avaliação realizada.

Como proposta do próximo capítulo descreve-se e desenvolve-se a arquitetura do STI e de seus agentes de *software*.

4. ABORDAGEM PROPOSTA – MEDICALGAME

Este capítulo tem como finalidade principal a demonstração dos modelos e arquiteturas propostas durante a execução e desenvolvimento deste trabalho. Na seção 4.1, abordar-se-á e se discutirá a proposta e o modelo da arquitetura do STI; na seção 4.2 será discutida e apresentada a modelagem baseada nos conceitos de agentes de *software*; na seção 4.3 serão descritas as ferramentas que foram utilizadas para o desenvolvimento do Sistema Tutor Inteligente; na seção 4.4 descrever-se-á a implementação do STI, utilizando os conceitos de agentes de *softwares*; e na seção 4.5 são apresentados e descritos os agentes de *software* desenvolvidos na aplicação.

4.1. PROPOSTA DA ARQUITETURA DO STI – MEDICALGAME

Inicia-se esta seção com o intuito de demonstrar e exemplificar a arquitetura do STI construída neste trabalho, que foi baseada na arquitetura denominada tripartida, citada na seção 2.2.1. A arquitetura aqui proposta pode ser visualizada na Figura 11, que traz uma visão geral do STI. Este conta com um banco de dados, onde os dados, tanto do sistema administrativo do STI, quanto os do sistema aqui denominado MedicalGame, podem ser inseridos, consultados e tratados.

O STI é composto por quatro agentes, cada qual com sua função, a arquitetura destes agentes e o domínio das aplicações podem ser visualizada na **Erro! Fonte de referência não encontrada.**, estando subdividida em camadas, sendo elas: A camada WEB; A camada de banco de dados, A camada do STI, bem como uma subdivisão entre sistema administrativo e sistema MedicalGame, essas interligadas e efetivando interações, através de dados oriundos do banco de dado dos sistemas.

Na camada descrita como *web*, o STI está disponível para o acesso pelos mais diversos dispositivos, como, por exemplo: *Tabletes, Smartphones, Computadores pessoais, Notebooks*.

Na camada intitulada banco de dados, parte mais interna da Figura 11, que demonstra o domínio do sistema, são inseridos, alterados e consultados os dados gerados pela camada STI. Na camada denominada STI, as duas aplicações internas denominadas, respectivamente, como: Entrada e Configuração, responsável pela configuração do STI, cadastro e configuração de todas as variáveis responsáveis pelo funcionamento do sistema

MedicalGame. E MedicalGame, que busca informações que foram cadastradas, via camada denominada Entrada e Configuração, assim como pela utilização dos agentes: Agente de Nivelamento, Agente de Alocação, Agente de tutoria e Agente Duelo.

Por meio dos sistemas administrativo e MedicalGame, os modelos básicos que compõem um STI são construídos, sendo eles: o Modelo de Domínio; o Modelo Aluno; o Modelo Pedagógico; e o Modelo de Interface. Composto estes modelos, têm-se, implementados, os seguintes agentes: o Agente de Nivelamento; o Agente de Alocação; o Agente de Tutoria; E o Agente de Interface, todos estes agentes geram dados, que são inseridos, consultados ou alterados no banco de dados.

Os quatro modelos propostos na Figura 11 serão discutidos, da mesma maneira que as interações efetivadas entre eles e os agentes. A descrição e exemplificação dos quatro agentes aqui citados serão efetivadas, especificamente na seção 4.5.1.

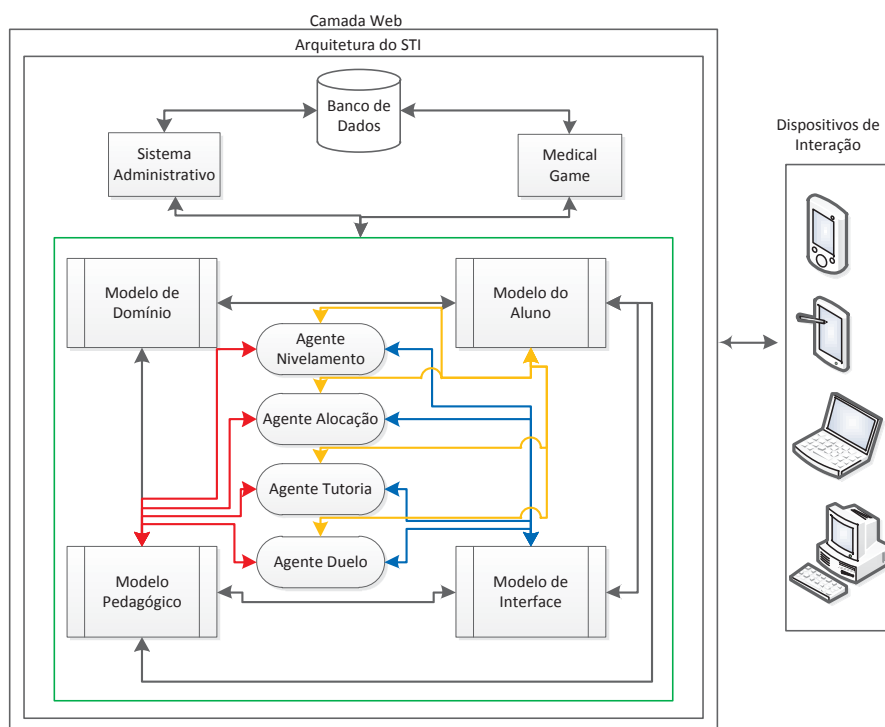


Figura 11: Arquitetura do STI desenvolvido. Fonte: Elaborado pelo autor.

Inicia-se a descrição da Figura 11, a partir o Modelo do Aluno, que tem como premissa a modelagem de características individuais de cada aluno. Neste contexto, o modelo aluno efetiva troca, envia e recebe informações oriundas de mais dois modelos, o modelo pedagógico e o modelo de domínio. Outra característica a ser tratada é a interconexão do modelo do aluno, como os agentes: Nivelamento, Alocação, Tutoria e Duelo.

Como segundo modelo a ser tratado, cita-se o Modelo de Domínio, este modelo detém o conhecimento dos conteúdos a serem tratados. Interagindo diretamente com o modelo pedagógico e o modelo do aluno. No que diz respeito a interações com os agentes construídos neste trabalho, o Modelo Pedagógico troca dados diretamente com todos os agentes disponíveis, sendo eles: Nivelamento, Alocação, Tutoria e Duelo.

O Modelo Pedagógico, modelo onde são implementadas as estratégias e táticas a serem selecionadas em função das características de cada aluno, interage com o modelo de domínio, onde o conhecimento sobre os conteúdos é tratado, e também com o modelo do aluno, para selecionar as características de cada aluno individualmente. O modelo pedagógico também troca dados diretamente com os seguintes agentes: Nivelamento, Alocação, Tutoria e Duelo.

Já, o Modelo de Interface, último modelo a ser descrito, faz o intermédio dos dados coletados, gerados e processados pelos modelos Pedagógico e Aluno. Estes dados, depois, são apresentados para a interface da aplicação. O modelo de interface, também necessita da utilização e troca de informações da totalidade dos agentes criados dentro da aplicação.

Cita-se, em seguida, algumas estratégias e prática que foram implementadas no Modelo Pedagógico do STI.

4.1.1. Estratégias e táticas do modelo pedagógico

No modelo pedagógico do STI desenvolvido são dispostas algumas estratégias e táticas, estas selecionadas em função das características de cada aluno, dentre elas: a utilização de uma etapa de nivelamento, responsável por alocar o aluno em seu primeiro contato com a aplicação, em um nível de dificuldade dentro STI. A segunda tática é a alocação do aluno em fases de dificuldade distintas, esta alocação ocorre em virtude de acertos efetivados durante o uso do STI.

Outras estratégias, dentro deste modelo, que não se enquadra individualmente a cada aluno, é a apresentação de tutorias, referentes à resolução de questões, disponível para o aluno. Estes tutorias podem ser oferecidos na forma de imagens, com explicações referentes ao problema proposto, áudio ou vídeo. Todas as questões do sistema MedicalGame contam com três tentativas de resolução, logo, se o aluno utilizar das três tentativas poderá receber até dois tutorias, quanto à resolução da referida questão.

Outra opção disposta neste contexto é a de utilização de materiais adicionais na apresentação das questões, como imagens, vídeos e áudios, estas também são dispostas durante a apresentação da questão que está sendo discutida.

Item importante a ser discutido dentro do contexto do modelo pedagógico é a opção de utilização do que se chama “dicas de níveis”, ou seja, materiais adicionais disponíveis para os alunos, que dizem respeito àquela etapa ou nível de dificuldade que o aluno está.

Outra opção que o sistema disponibiliza é a visualização da quantidade de níveis, assim como a porcentagem de perguntas respondidas pelo aluno, este recurso tenta buscar engajamento do aluno quanto à resolução das questões.

4.2. PROPOSTA E MODELAGEM DA ARQUITETURA BASEADA EM AGENTES DE *SOFTWARE*

Com embasamento nos estudos desenvolvidos, optou-se pela utilização dos conceitos definidos ou tratados sobre agentes reativos simples, que têm como sua premissa a relação de atuação, amparada em estímulos e repostas, verificando regras específicas para essas ações e desencadeando uma reação.

a construção dos agentes, então, foi efetivada utilizando os conceitos de agentes reativos simples, esses desenvolvidos em linguagem de programação, sem a utilização de frameworks específicos.

Para exemplificar o funcionamento dos agentes reativos simples, desenvolvidos neste trabalho, apresenta-se a Figura 12, que demonstra o fluxo de dados do agente reativo simples denominado duelo.

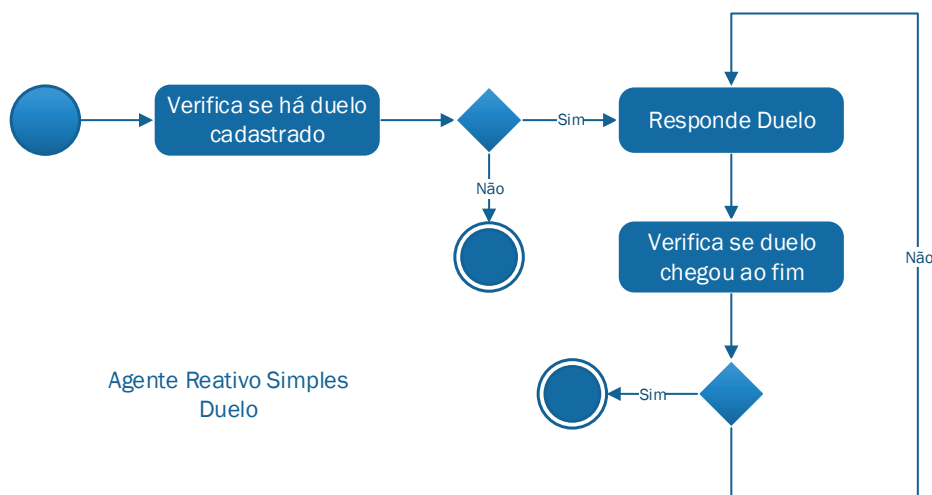


Figura 12: Agente reativo simples Duelo. Fonte: Elaborado pelo autor.

O agente reativo simples duelo inicia seu fluxo de informações, efetivando a verificação da existência de duelo cadastrado. Se não houver um duelo cadastrado, o fluxo passa para outro agente.

Se houver duelo cadastrado, o sistema apresenta as questões vinculadas a esse duelo, e o aluno inicia a resolução das mesmas. Dentro deste processo verifica-se se a quantidade de questões do duelo é igual a quantidade de questões respondidas dentro do agente duelo. Se esta verificação retornar verdadeiro, o fluxo de dados passa para o próximo agente, caso contrário, o fluxo é retornado e o aluno continua respondendo questões do duelo, este fluxo pode ser observado na Figura 12.

4.3. DEFINIÇÃO DA UTILIZAÇÃO DE FRAMEWORKS/TÉCNICAS E FERRAMENTAS PARA O DESENVOLVIMENTO DE SISTEMAS BASEADOS EM AGENTES DE *SOFTWARE*

Conforme RUSSELL [20], agentes reativos simples podem ser construídos utilizando-se estruturas de programação primárias, conforme já apresentado na seção 2.1.1.3, Figura 3, que apresenta um pseudocódigo com blocos de código, se, senão, então, e uma estrutura de função que envia alguma informação e recebe um retorno.

Dentro deste estudo constatou-se que para o desenvolvimento deste trabalho, poderiam ser utilizadas as seguintes tecnologias e ferramentas: CSS, HTML5, BootStrap, PHP e MySql, por serem tecnologias e *softwares* livres, ademais de disponibilizarem atualização constante, o que é característica importante na escolha de *softwares* para serem utilizados em implementações. Isso possibilita a expansão e o uso de novas ferramentas

dentro do contexto do *software* que foi desenvolvido, sem a preocupação da descontinuidade da linguagem ou *software* que foi utilizado para a sua construção, além de serem *softwares* livre³, e contarem com ampla gama de documentação para consulta e pesquisa.

As ferramentas de tecnologias utilizadas para o desenvolvimento do trabalho serão descritas e conceituadas brevemente:

CSS, *Cascading Style Sheets* é uma linguagem de folhas de estilo utilizada para definir a apresentação de documentos escritos em uma linguagem de marcação, como HTML ou XML. O seu principal benefício é a separação entre o formato e o conteúdo de um documento. - <https://www.w3schools.com/css/>

HTML5, (*Hypertext Markup Language*, versão 5) é uma linguagem para estruturação e apresentação de conteúdo para a *World Wide Web* e é uma tecnologia chave da *Internet* originalmente proposto por *Opera Software*. É a quinta versão da linguagem HTML. - www.w3schools.com/html/html5_intro.asp

BootStrap, *Bootstrap*, em Português (Brasil), agradável, intuitivo e poderoso *framework front-end* para criar facilmente, de forma ágil, projetos *web* responsivos e *mobile-first*. - getbootstrap.com.br/

PHP é uma linguagem de *script* de servidor e uma ferramenta poderosa para fazer páginas da *Web* dinâmicas e interativas. PHP é uma alternativa amplamente utilizada, livre e eficiente para concorrentes como o ASP da *Microsoft*. - https://php.net/manual/pt_BR/index.php

MySql é o SGBD de código aberto mais conhecido no mundo. Com comprovado desempenho, confiabilidade e facilidade de uso, o MySQL tornou-se a principal opção de banco de dados para aplicativos baseados na Web - <https://www.mysql.com>

Principia-se a discussão sobre a Figura 13 com a designação de cada camada nela criada, conforme segue: Hospedagem da Plataforma; Camada de Banco de Dados; Camada de Aplicação e Camada Sistema Tutor Inteligente. Trata-se, separadamente, de cada uma delas, designando as ferramentas e tecnologias utilizadas para concepção e construção de cada camada.

Inicialmente, a **Camada Hospedagem da Plataforma**, requisito importante para o bom funcionamento da aplicação. Nesta camada, são representados os itens para o funcionamento e manutenção dos serviços da aplicação, três tipos de serviços em um servidor, são necessários, sendo eles: Hospedagem da aplicação, Sistema gerenciador de Banco de

³ *Software* livre é uma expressão utilizada para designar qualquer programa de computador que pode ser executado, copiado, modificado e redistribuído pelos usuários gratuitamente.

dados, para a inserção e consultas, das informações oriundas do STI e da Aplicação, e um serviço FTP habilitado. Não necessariamente cada um destes serviços deve ser contratado em servidores diferentes, mas se houver necessidade, é plenamente viável esta opção.

Na camada intitulada **Camada de Banco de Dados**, optou-se pela utilização do SGBD MySQL, como proposta inicial para a utilização e criação do banco de dados necessário para a aplicação.

Na **Camada de Aplicações**, que é subdividida em Aplicação Administrativa e Módulo perguntas e Respostas, tem-se a utilização das mesmas tecnologias e ferramentas para as duas ocasiões. Deste modo, descreve-se a utilização destas ferramentas e tecnologias uma única vez neste texto. Nesta perspectiva, a Camada de Aplicações utiliza, explicitamente, PHP, folhas de estilo, para a configuração visual das páginas da aplicação, HTML5 e *Bootstrap* para a construção do sistema proposto.

Na camada intitulada **Sistemas Tutores Inteligentes**, são construídos e modelados: O Modelo do domínio; O Modelo do Aluno; O Modelo Pedagógico e O Modelo da Interface, faz-se o uso das seguintes ferramentas e tecnologia, PHP, HTML5, *Bootstrap* e folhas de Estilo.

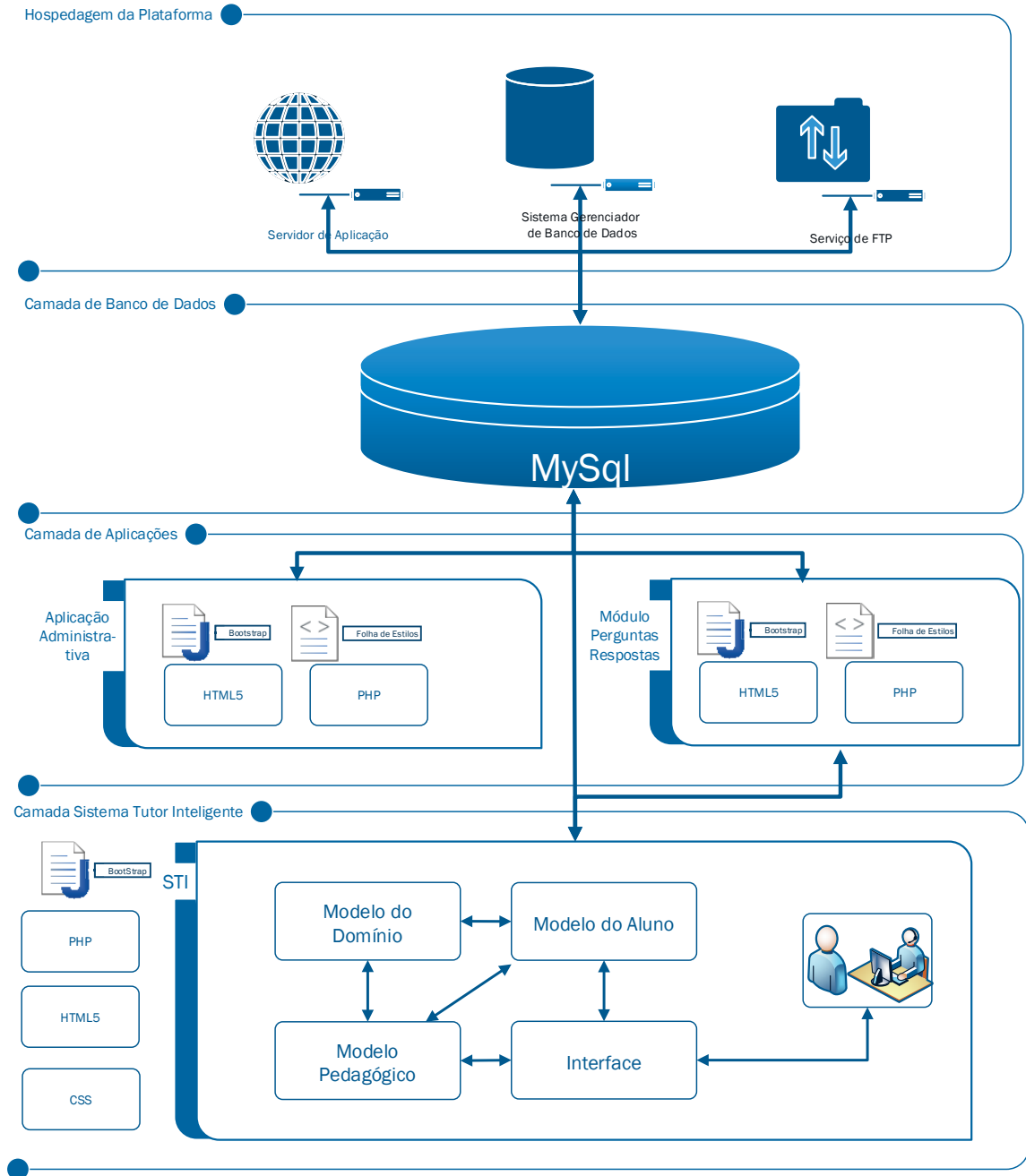


Figura 13: Ferramentas e tecnologias, utilizadas para a implementação dos sistemas. Fonte: Elaborado pelo autor.

4.4. IMPLEMENTAÇÃO DO PROTÓTIPO DE STI

Nesta seção serão apresentadas três partes da implementação efetivadas, quais sejam: A página Institucional, com sua descrição e apresentação no Apêndice B - Página institucional; A página de Administração, apresentada na seção 4.4.1 Página Administrativa, e a página do sistema MedicalGame, apresentada da seção 4.4.2 Página do Sistema

MedicalGame, bem como a apresentação das estratégias e táticas implementadas no STI, presentes na seção 4.5 – Agentes desenvolvidos no Sistema MedicalGame.

Apresenta-se uma breve descrição da página institucional do trabalho, esta tem o intuito de divulgar e descrever o trabalho que foi desenvolvido. Foi construída utilizando-se os *softwares* citados na metodologia do trabalho e tem como premissa a utilização de *layout* responsivo. A apresentação completa da página institucional, tal como a sua descrição estão disponíveis no Apêndice B - Página institucional.

O processo de implementação do protótipo de STI começou pelo estudo dos conceitos referentes à prova de residência médica, descritos na seção 3.1 - Prova de Residência Médica.

O processo de desenvolvimento de *software* obedeceu à arquitetura de construções de *software*, baseado em métricas de Engenharia de *Software*, que têm como proposta e padrão o que é apresentado na Figura 10, conforme descrito na metodologia do trabalho.

O documento de requisitos foi construído tanto para a página administrativa do sistema MedicalGame, quanto para a página do sistema MedicalGame (documentação presente no Apêndice C -Documento de Requisito de *software* – página administrativa medical game e sistema MedicalGame.), este documento foi apresentado em forma de apêndice por ser um documento estritamente técnico, apenas quatro casos de uso e suas descrições estão apresentados durante o texto na seção 4.4.2, para que um melhor entendimento dos agentes de *software* construídos seja possível.

4.4.1. Página Administrativa

A implementação da página administrativa será apresentada de forma a demonstrar a estrutura de configuração de dois modelos, que são essenciais, para um STI, neste caso, a configuração do **modelo de domínio** e a configuração do **modelo pedagógico**, envolvidos no contexto geral deste trabalho, sendo que a tela principal do sistema pode ser visualizada na Figura 14.

No *menu* principal da página administrativa do STI, são apresentados os seguintes itens de *menu*: Cadastrar, neste *menu* são implementadas oito funções na seguinte ordem: Tipo de Usuário; Instituição; Área de questões; Subárea de questões; Nível de Dificuldade das questões; Questões; Dicas de Níveis de Dificuldade; Cadastro de Desafios. Estes itens serão discutidos e apresentados na mesma ordem que aqui dispostos, alguns deles, com apenas descrição textual, por serem simples e terem fácil entendimento, e outros, com a

demonstração da tela e descrição textual para um melhor entendimento de suas funcionalidades.

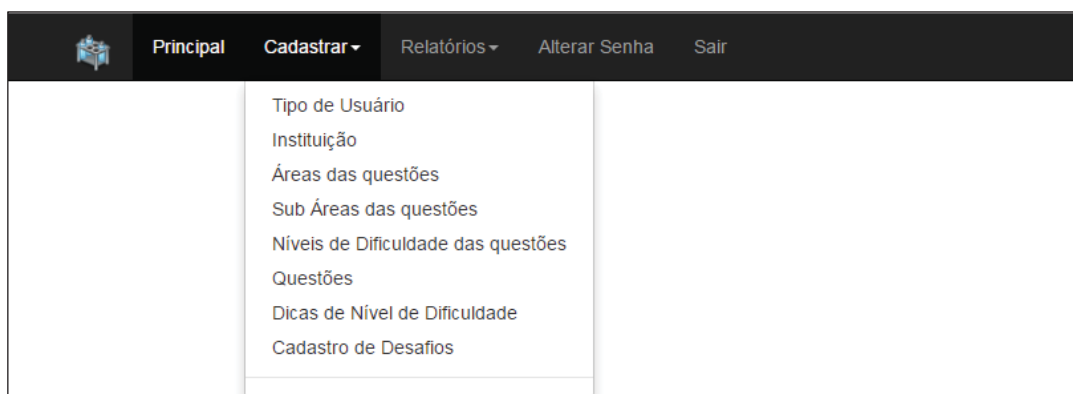


Figura 14: Tela do *menu* Cadastrar. Fonte: elaborada pelo autor.

Igualmente, serão demonstradas e exemplificadas as funcionalidades do módulo administrativo do sistema MedicalGame, este é padrão para todo o sistema. A Figura 15 apresenta a funcionalidade do sistema, intitulada cadastro de tipo de usuário; optou-se pela utilização e explicação desta tela como padrão, por ser uma das mais simples e de fácil entendimento. O objetivo da sua apresentação é a explicação das funcionalidades de inserção, denominada no sistema, cadastrar e a função de edição, assim respectivamente.

Divide-se a Figura 15 em três partes, a primeira, onde se tem o destaque de um retângulo, denominado **campos de cadastro**, nestes campos são inseridos os dados referentes às informações que estão sendo cadastrados na base de dados do sistema.

Na segunda parte da Figura 15, denominada **conjunto de botões**, apresentam-se dois botões, um denominado cancelar, que cancela o que está sendo efetivado na tela, e um segundo botão, com duas funções implementadas, sendo elas: a função de edição e a função de inserção de registros, estas são dependentes do contexto que está sendo tratado.

Na terceira parte da Figura 15, denominada apresentação dos últimos dados cadastrados, em um grid são apresentados os últimos dez registros cadastrados no banco de dados, com a opção de edição, à direita de cada registro, para que, se for necessário alteração de algum dado, o administrador do sistema efetive esta operação.

Após, serão apresentadas telas dentro do mesmo contexto de *menu*. Desta forma, para que não se torne repetitivo, as próximas telas terão o *menu* principal retirado da imagem, ficando visível somente o que está sendo contextualizado, exemplificado e discutido.

4.4.1.1. Cadastrar tipo de usuário

Esta função tem como intuito o cadastro de tipos distintos de usuários para a utilização dos sistemas apresentados durante este trabalho. A tela deste cadastro pode ser visualizada na Figura 15.

No contexto do trabalho aqui desenvolvido foram necessários o cadastro de dois tipos de usuários. O tipo aluno, com acesso ao módulo STI, e o tipo administrador, que tem acesso ao módulo administrativo e também ao módulo STI.

Código	Descrição	Editar
1	Administrador	
2	Alunos	

Figura 15: Tela de Cadastro de Tipo de Usuário. Fonte: Elaborada pelo autor.

4.4.1.2. Cadastrar instituição

A tela cadastro e instituição foi desenvolvida com o objetivo do cadastro e edição de mais de uma instituição para a utilização do sistema, tendo por objetivo a expansão do trabalho proposto, para a participação futura de mais instituições de ensino.

Estas várias instituições poderão utilizar as questões desta base de dados, assim como a troca de informações referentes a questões, alunos, gerando informação e valor agregado para as universidades e faculdades envolvidas no processo.

4.4.1.3. Cadastrar áreas das questões

O cadastro de áreas das questões foi implementado com o mesmo intuito apresentado no cadastro de instituições. Optou-se pela criação da estrutura da base de dados, em que se contempla o cadastro de várias áreas, aqui designada medicina, para transformar o STI utilizável por qualquer curso ou área que venha a ser cadastrado no sistema. No contexto do

estudo, trabalhou-se somente com a área de medicina, que é propósito inicial, e que foi discutido até o presente momento.

4.4.1.4. Cadastrar subáreas das questões

Neste contexto, tem-se a implementação da funcionalidade responsável pelo cadastro e edição de subáreas das questões.

Na pesquisa desenvolvida, foram trabalhadas e cadastradas cinco subáreas estipuladas pela equipe de medicina envolvida, sendo elas: Clínica Médica; Pediatria; Ginecologia/Obstetrícia; Cirurgia Geral; Medicina Social, que podem ser visualizadas na Figura 16.

Cod_sub_area	Desc_sub_area	Desc_area	Editar
1	Clínica Médica	MEDICINA	
2	Pediatria	MEDICINA	
3	Ginecologia/Obstetrícia	MEDICINA	
4	Cirurgia Geral	MEDICINA	
5	Medicina Social	MEDICINA	

Figura 16: Tela de Cadastro de Sub Áreas. Fonte: Elaborada pelo autor.

4.4.1.5. Cadastrar nível de dificuldade de questões

O cadastro de níveis de dificuldade de questões tem como objetivo a organização e separação das diversas questões que podem ser cadastradas no STI. A tela deste cadastro pode ser visualizada na Figura 17, onde são apresentados os seguintes itens de cadastro: Descrição do Nível; Quantidade de Questões a serem respondidas no Nível; Mínimo de Acertos a serem efetivados no Nível; Quantidade de Questões do Nivelamento; Quantidade de Questões do Duelo, que serão descritos e discutidos a seguir.

O campo **Descrição do Nível** é utilizado pelo administrador do STI para descrever, de forma sucinta, o nível das questões que está se referindo ou configurando.

No campo Quantidade de Questões a serem respondidas no Nível, o administrador tem a opção de informar a quantidade de questões que devem ser respondidas no nível de dificuldade que está sendo editado ou cadastrado.

O campo **Mínimo de Acertos a serem efetivados no Nível** possibilita ao administrador a configuração da quantidade mínima de acertos que o aluno deve realizar para que o STI efetive a alteração de nível de forma automática.

O campo designado **Quantidade de Questões do Nivelamento** possibilita ao administrador a efetivação e configuração da quantidade de questões que devem ser respondidas para o nivelamento⁴.

No campo designado **Quantidade de questões do Duelo**, o administrador tem a opção de efetivar a configuração da quantidade de questões a serem respondidas no item designado Duelo⁵.

Medical Game -- Níveis de Dificuldade das Questões Cadastrados

Descrição do Nível

Descrição do Nível

Quantidade de Questões a serem respondidas no Nível

Mínimo de Acertos a serem efetivados

Mínimo de Acertos a serem efetivados no Nível

Mim. Acertos Niv.

Quantidade de Questões do Nivelamento

Quantidade de Questões do Nivelamen

Quantidade de questões do Duelo

Quantidade de questões do Duelo

Cadastrar Voltar

Figura 17: Tela de Cadastro de Configuração de Níveis de Dificuldade. Fonte: Elaborada pelo autor.

4.4.1.6. Cadastro de questões

⁴ Nivelamento – Especificamente no sistema proposto, define-se nivelamento como um processo inicial do sistema, para com o aluno/jogador, para que o mesmo responda algumas questões dos mais diversos níveis, a fim de elencar determinado nível a este usuário.

⁵ Duelo – Especificamente no sistema, duelo tem o sentido de disputa ou seja os dois primeiros alunos de cada nível podem participar de uma rodada de perguntas a serem respondidas, o que acertar mais questões será o vencedor do duelo proposto.

A função de cadastro de questões é uma das mais extensas a serem discutidas, esta funcionalidade é apresentada na Figura 18, Figura 19 e Figura 20. Esta tela foi separada deste modo para que se possa explicar e exemplificar, de forma clara e sucinta, o funcionamento de toda a tela e funções nela implementadas.

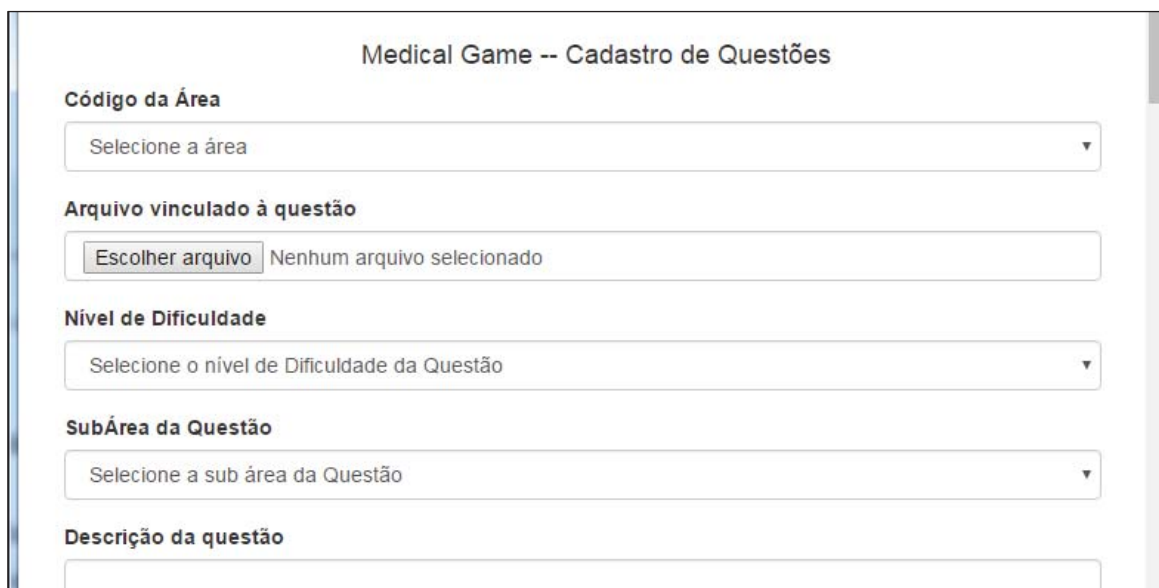
Na Figura 18, dispõe-se um comobox, com título Código da Área, com as opções de áreas que estão cadastradas no banco de dados da aplicação, neste caso (medicina), no campo Arquivo vinculado à questão, o administrador pode efetivar o carregamento de qualquer tipo de arquivo, que tenha disponível ou que seja essencial para o entendimento da questão. Este arquivo será salvo em um diretório na nuvem⁶, para posterior busca e apresentação, se houver necessidade.

Como próximo item de tela está disposto um comobox que apresenta os níveis de dificuldades pré-cadastrados e configurados no banco de dados do sistema, neste caso os níveis de dificuldade são: Nível 1; Nível 2; Nível 3; Nível 4 e Nível 5.

Em sequência, outro comobox traz do banco de dados a informação de todas as subáreas possíveis, para a área selecionada. Neste contexto, a área é medicina e as subáreas deste comobox são: Clínica Médica; Pediatria; Ginecologia/Obstetrícia; Cirurgia Geral e Medicina Social.

Como último campo apresentado na Figura 18, apresenta-se o cadastro da descrição da questão, que é efetivada em um campo de texto, onde o administrador do STI pode descrever, de forma detalhada, o enunciado da questão que está cadastrando ou editando.

⁶ O conceito de computação em nuvem refere-se à utilização da memória, capacidade de armazenamento, cálculo de computadores e servidores compartilhados e interligados por meio da *Internet*. Segundo CHEE [35]



Medical Game -- Cadastro de Questões

Código da Área
 Seleccione a área

Arquivo vinculado à questão
 Escolher arquivo Nenhum arquivo selecionado

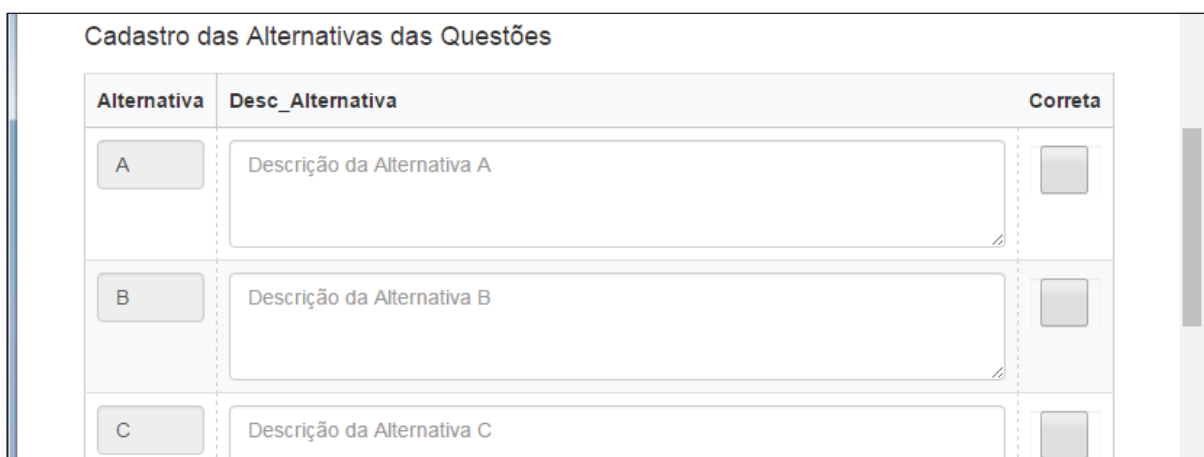
Nível de Dificuldade
 Seleccione o nível de Dificuldade da Questão

SubÁrea da Questão
 Seleccione a sub área da Questão

Descrição da questão

Figura 18: Tela de cadastro de questões. Fonte: Elaborada pelo autor.

Como continuação da apresentação e explicação da tela de cadastro de questões, apresenta-se a Figura 19, que demonstra um recorte da tela, onde são apresentados campos texto para a descrição das alternativas, bem como uma opção de checagem para a alternativa que estiver correta.



Alternativa	Desc_Alternativa	Correta
A	Descrição da Alternativa A	<input type="checkbox"/>
B	Descrição da Alternativa B	<input type="checkbox"/>
C	Descrição da Alternativa C	<input type="checkbox"/>

Figura 19: Tela de cadastro de questões. Fonte: Elaborada pelo autor.

Como terceira parte e, não menos importante, da implementação da tela de cadastro de questões, apresenta-se a Figura 20. Nela é demonstrada a implementação de um campo texto que tem a função de receber dados de um *feedback* para quando o usuário responder a questão pela primeira vez.

Como último campo, há a opção de cadastro de tutoria. Neste campo, o administrador pode efetivar o carregamento de qualquer tipo de arquivo que julgar conveniente para a tutoria da questão que está sendo cadastrada.

Cadastro de feedback de erro da questão / VAI APARECER NO MOMENTO EM QUE O ALUNO ERRAR A QUESTÃO PELA PRIMEIRA VEZ

Cadastro de TUTORIA / VAI APARECER PARA O USUÁRIO NO SEGUNDO ERRO DA MESMA QUESTÃO

Escolher arquivo Nenhum arquivo selecionado

Figura 20: Tela de cadastro de questões. Fonte: Elaborada pelo autor.

4.4.1.7. Cadastro de dicas para os níveis

Perante o contexto estudado e discutido durante o desenvolvimento do trabalho proposto, a função apresentada nesta tela foi desenvolvida com o objetivo de cadastro de dicas para cada nível.

A tela é composta por um comobox que apresenta todos os níveis de dificuldade cadastrados, um campo de texto para que o administrador possa efetivar uma descrição sucinta de uma dica, e outro campo de texto, onde o administrador pode efetivar o cadastro de um link, referente a qualquer tipo de material, que seja condizente com o nível de dificuldade que está sendo configurado ou tratado.

Medical Game -- Dicas para os Níveis de Dificuldade das Questões.

Nível de Dificuldade
Selecione o nível de Dificuldade da

descrição da dica do Nível de Dificuldade
Descrição da Dica do Nível de Dificuldade

Link da dica do Nível de Dificuldade
Link da Dica do Nível de Dificuldade

Cadastrar Voltar

Figura 21: Tela de cadastro de dicas de nível. Fonte: elaborada pelo autor.

Outra estrutura pensada, estudada e desenvolvida é a designada Cadastro de Duelos, na Figura 22. Mesmo sendo simplista em sua concepção, tem sua funcionalidade garantida. Nela o administrador deve somente selecionar o nível de dificuldade do qual quer cadastrar o duelo.

As informações dos níveis de dificuldade estão dispostas em um comobox, selecionando o nível neste comobox, o sistema busca as informações de quem são os dois primeiros colocados deste nível, cadastrando um duelo entre estes dois alunos. A apresentação e descrição da funcionalidade do duelo serão apresentadas na seção 4.5.4.

O grande objetivo do desenvolvimento desta funcionalidade foi o de melhorar o engajamento dos alunos perante o STI, em sua concepção final.

Cod. Des.	Id. Des.	Nom. Usu.	Cod. Niv. Des.	Pontos Des.	Dt. Respondido? Cad.
27	2	usuario1@usuario14			
28	2	usuario2@usuario24			

Figura 22: Telas de cadastro de desafio. Fonte: Elaborada pelo autor.

4.4.2. Página do Sistema MedicalGame

Posteriormente, serão definidos e explicados em forma de diagrama de fluxo e com as referidas telas do *software*, os fluxos das interações definidas e implementadas no STI.

4.4.2.1. Cadastro de usuário/alterações de senha

Nesta seção, a tela de *login* ao sistema MedicalGame, Figura 23, esta possui dois campos para preenchimento, um de *e-mail*, que é o que o aluno cadastrou no sistema, e um campo de senha, para que o mesmo preencha com a senha já cadastrada previamente.



Figura 23: Tela de *login* do Sistema. Fonte: elaborada pelo autor.

Buscando-se um melhor entendimento e compreensão do que foi desenvolvido na tela de *login* do sistema, apresenta-se um diagrama de fluxo de dados, que pode ser visualizado na Figura 24.

O procedimento de *login* pode ser descrito como o início do processo. O preenchimento dos campos citados anteriormente, referentes à Figura 23, e ao clicar no botão Entrar, o *e-mail* e senha do usuário são validados junto ao banco de dados da aplicação, se a validação for negativa o sistema retorna uma mensagem de alerta dizendo que o usuário ou senha estão incorretos e o usuário pode efetuar nova tentativa de *login*.

Seguindo o fluxo de informações referentes ao item Registrar, disposto na Figura 24, se o usuário não for registrado no sistema, este deve se registrar, seguindo o fluxo registrar. Ele será endereçado para uma página de cadastro onde são solicitados os seguintes dados: *E-mail*, Nome, Senha, o nível de graduação do aluno que está se cadastrando, e se já efetuou a prova de residência médica.

Efetivando seu cadastro, o aluno está apto a efetuar *login* no sistema. Como terceiro e último fluxo de dados deste diagrama, está representado o fluxo de alteração de senha, onde o usuário é endereçado para uma página de alteração de senha, sendo solicitado o *e-mail* que o mesmo cadastrou no sistema. Efetivando o preenchimento deste, o sistema, automaticamente, gera uma chave de validação de *e-mail*, com validade de 24 horas e envia esta chave ao usuário, em seu *e-mail*, com um *link*. Clicando neste *link* o usuário é reportado

para a página de alteração de senha. Após essa alteração, o usuário poderá efetuar *login* no sistema novamente.

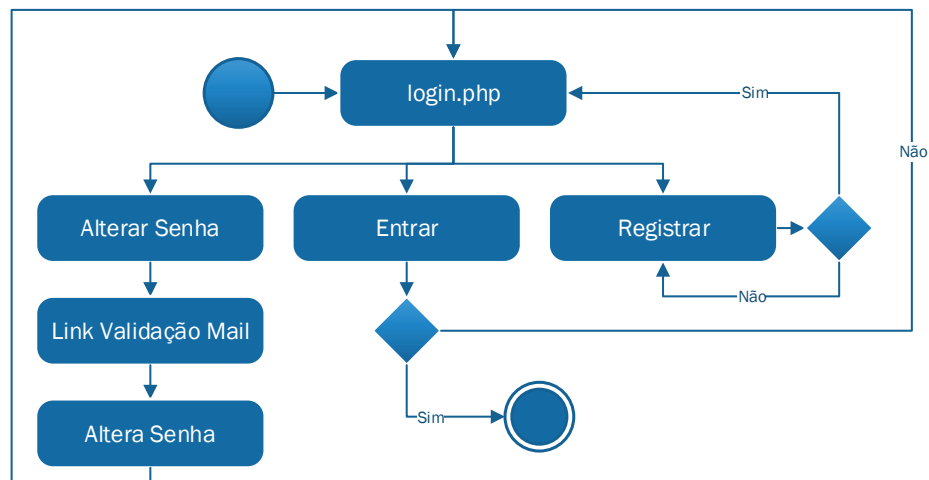


Figura 24: Fluxo de dados *login* do Sistema. Fonte: elaborada pelo autor.

4.4.3. Tela do Sistema MedicalGame

O painel de demonstração de informações referentes ao STI é apresentado na Figura 26 e será detalhado, inicialmente, pelas funcionalidades implementadas em seus *menus*.

Na tela representada na Figura 26, alguns *menus* são apresentados, estes, serão descritos sequencialmente, da direita para a esquerda. Primeiro, é apresentado o ícone de *logout*, que tem a função de efetivar a saída do usuário do sistema.

Como segundo *menu*, encontra-se um gráfico, que é atualizado em tempo real, baseado na resolução das questões de cada nível, sendo que a representação deste se dá pela resolução das questões de cada usuário. Desta forma, o aluno consegue saber o quanto de questões de cada nível respondeu, do mesmo que quantos níveis foram completados. Esta apresentação pode ser visualizada na Figura 25.

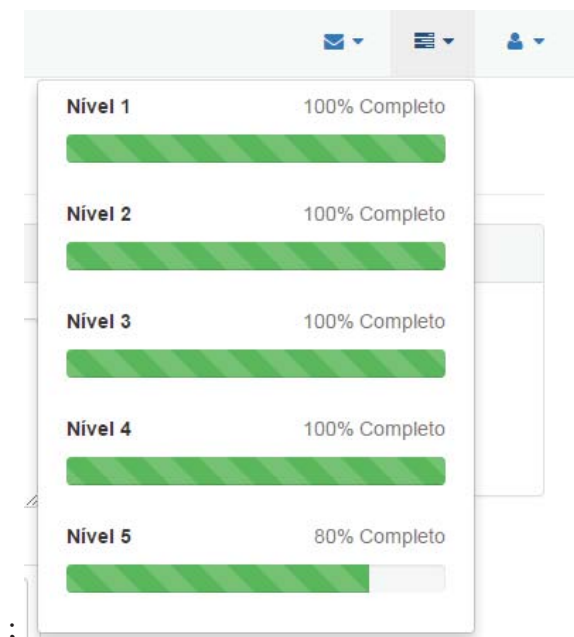


Figura 25: Gráfico de níveis completos. Fonte: Elaborado pelo autor.

Como terceiro ícone, são apresentadas as dicas do nível ao qual o usuário está vinculado. Estas dicas podem ser desde uma página *web*, vídeos, vídeo aulas, enfim, qualquer conteúdo que esteja disponível na *Internet*.

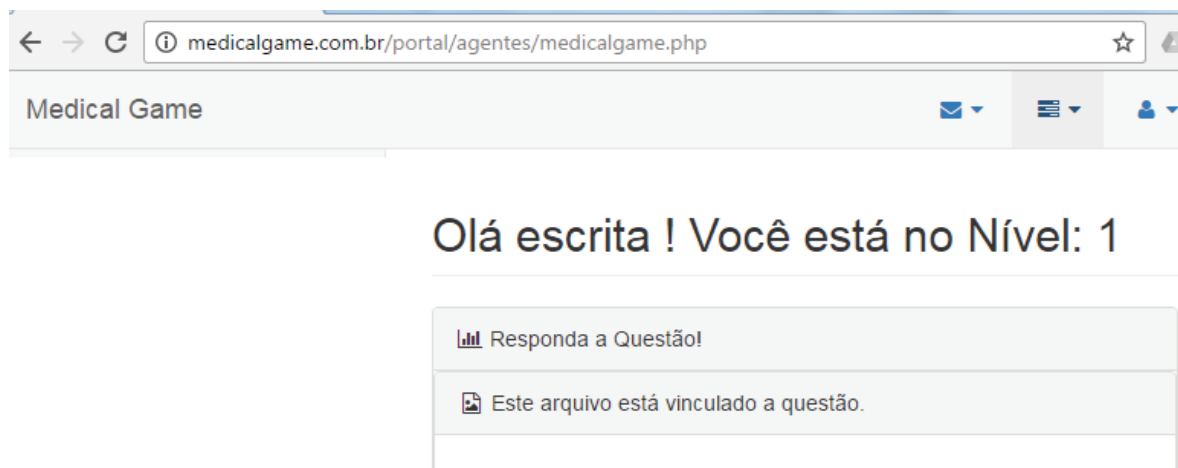


Figura 26: Tela do Sistema MedicalGame. Fonte: elaborada pelo autor.

Outro item importante e implementado é o *ranking* dos alunos, dentro de cada nível. Os 10 primeiros alunos melhor pontuados ficam visíveis, como demonstrado na Figura 27. A atualização desta funcionalidade dá-se em tempo real, conforme os usuários respondem as questões, a pontuação e *ranking* são atualizados automaticamente.

Em caso de empate na pontuação, o usuário que atingiu a pontuação primeiro aparece como primeiro colocado no *ranking*, como é o caso do usuario4@usuario4 e

asdf@asdf, ambos têm 4 pontos, mas como o usuario4@usuario4 teve a pontuação adquirida por primeiro, ele tem preferência no *ranking*. Para a apresentação desta funcionalidade foram utilizados usuários fictícios, a fim de que a identidade dos alunos que participaram dos testes fosse preservada.



Figura 27: Ranking top 10. Fonte: Elaborado pelo autor

A proposta da Figura 28 é de demonstrar e situar o leitor dos passos e processos desempenhados, desde seu simples cadastro de um aluno, até a finalização do fluxo das informações dentro do sistema desenvolvido.



Figura 28: Fluxo de dados MedicalGame. Fonte: elaborado pelo autor.

4.5.

AGENTES DESENVOLVIDOS NO SISTEMA MEDICALGAME

Nesta seção são apresentados os quatro agentes desenvolvidos durante o trabalho, sendo eles intitulados, Agente de nivelamento, Agente de alocação, Agente de tutoria, Agente de duelo.

4.5.1. Agente de nivelamento

O agente de nivelamento, primeiro agente a ser descrito e apresentado, tem como objetivo a verificação e validação do nível de dificuldade a que o aluno será atrelado dentro do sistema MedicalGame. Este agente é baseado na arquitetura de agentes reativos simples, sendo que este solicita e envia informações à base de dados da aplicação para que os agentes de Duelo, Agente Alocação e Agente Tutoria utilizem estes dados posteriormente.

O agente de nivelamento, na **Erro! Fonte de referência não encontrada.**, pode ser acessado, tanto do módulo administrativo, onde são efetivadas as configurações de quantidade de questões a serem respondidas no processo de nivelamento, quanto no sistema MedicalGame, que efetiva o nivelamento, ou calcula em qual nível de dificuldade o aluno será enquadrado dentro da aplicação.

Inicia-se, em tal caso, a demonstração da tela que implementa o Agente de Nivelamento, na Figura 29, tendo dispostas as seguintes informações: uma mensagem descrevendo o que esta tela representa, na sequência são apresentadas a descrição e as alternativas de uma questão, para que o aluno possa efetivar sua resposta.

Bem vindo ao Medicalgame, você deverá responder algumas questões para que consigamos te alocar em um nível de dificuldade condizente com as respostas que você efetivou durante este processo.

Bom estudo!

Medical Game -- Agentes de Nivelamento

Descrição da questão 57 -- 1

Mulher de 20 anos de idade, dirigindo sem cinto de segurança, perde o controle em uma curva e colide frontalmente com outro veículo. Trazida à emergência, chega agitada, sudorética, com palidez importante e extremidades frias. Após o atendimento inicial previsto no ATLS (Advanced Trauma Life Support), o FAST (Focused Assessment with Sonography for Trauma) mostra líquido livre na goteira parietocólica esquerda e na pelve, com volume de aproximadamente 800 ml. Após a reposição volêmica inicial, segue instável hemodinamicamente. Considerando as circunstâncias, a próxima conduta deve ser:

Alternativas da Questão

Alternativa	Resposta	Correta
A	Aguardar a evolução, com monitoramento invasivo da paciente	<input type="checkbox"/>
B	Indicar laparotomia exploradora de imediato	<input type="checkbox"/>
C	Realizar punção abdominal para a caracterização do líquido detectado	<input type="checkbox"/>
D	Realizar ressonância magnética para avaliar com maior precisão o volume de líquido intra-abdominal	<input type="checkbox"/>
E	Realizar novo ultrassom para melhor investigar a causa da instabilidade	<input type="checkbox"/>

Responder

Figura 29: Agente de Nivelamento. Fonte: Elaborada pelo autor.

Dentro do fluxo de dados apresentado na Figura 30 demonstra-se o que é efetivado, internamente, na página niv1.php, apresentada na Figura 29, em que o Agente de Nivelamento é implementado.

Nesta perspectiva, começa-se o processo de tomada de decisão. Sabe-se que o aluno que foi conduzido ao Agente de Nivelamento não tem um nível de dificuldade de questões atrelado a ele dentro de sua configuração, seja por não ter efetivado parte ou a totalidade deste nivelamento. Por isso, o Agente de Nivelamento inicia a efetivação de várias

etapas: A primeira delas é a verificação da quantidade de níveis de dificuldades cadastrados no sistema; na segunda etapa o Agente de Nivelamento concretiza a seleção de uma questão, embasado na informação de que o usuário deve responder questões que sejam sorteadas aleatoriamente, em princípio, do nível de dificuldade 1, como terceira etapa, a questão sorteada é apresentada. Esta apresentação pode ser visualizada na Figura 29. Como quarta etapa, o sistema verifica se a questão está certa ou errada, insere essas informações no banco de dados; na quinta etapa do processo há uma verificação se a quantidade de níveis chegou ao fim; na sexta etapa a verificação da quantidade de questões a serem respondidas naquele nível é efetivada.

Se o nível de dificuldade for o último, e não houver mais questões do nível a serem respondidas, a pontuação do aluno é calculada, esta pontuação é embasada em seus acertos e erros. Após este cálculo o aluno é alocado em determinado nível de dificuldade de questões.

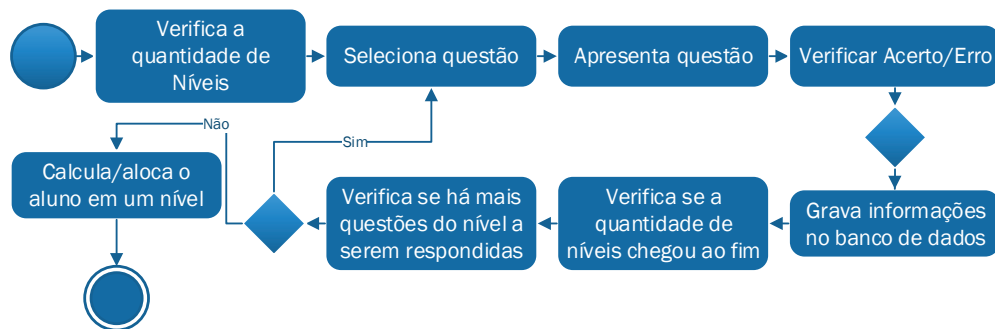


Figura 30: Fluxo de dados processo de nivelamento. Fonte: elaborada pelo autor.

Apresenta-se, aqui, o caso de uso referente ao Agente de Nivelamento, que pode ser visualizado na Tabela 2, características específicas são apresentadas de forma detalhada, e, também, os processos efetivados pelo Agente Nivelamento.

Tabela 2: Descrição do caso de uso Nivelar Aluno. Fonte: Elaborado pelo autor.

Manter Nivelar Aluno	
Cadastrar Resposta Questão - Nivelar Aluno	
Item	Descrição
Caso de Uso	Nivelar Aluno
Resumo	O sistema verifica a quantidade de níveis, seleciona questões embasado nos níveis de dificuldade, o sistema apresenta a questão para o aluno, o sistema requisita respostas do aluno, o aluno procede com a leitura, interpretação e devida resposta. São gerados logs desse caso de uso, que são inseridos em um banco de dados.
Ator	Agente Nivelamento

	Pré-condições	O Agente Nivelamento deve ter sido pré-configurado no sistema administrativo.
	Pós-Condições	Um novo registro de resposta da questão deve ser cadastrado no sistema O Aluno deve ser alocado em determinado nível de dificuldade dentro do sistema.
	Fluxo Principal	<ol style="list-style-type: none"> 1. O sistema verifica a quantidade de níveis disponíveis; 2. O sistema sorteia uma questão; 3. O sistema apresenta uma questão a ser respondida; 4. O aluno responde a uma questão; 5. O sistema retorna uma mensagem de que a questão foi respondida. 6. O sistema cadastra a resposta da questão e o caso de uso termina.
	Fluxo Alternativo	<p>Fluxo Alternativo (2): Verifica de qual nível de dificuldade a questão deve ser sorteada.</p> <p>Fluxo Alternativo (5 a): verifica quantos níveis estão configurados no sistema, se respostas estiverem faltando em questões para o nivelamento, o nível em que o usuário está é enviado para o Fluxo Principal (2).</p> <p>Fluxo Alternativo (5 b): Se o Aluno respondeu todas as questões solicitadas no nivelamento, o sistema apresenta uma mensagem que ele será alocado em determinado nível, embasado nas respostas efetivadas de forma correta.</p>
	Fluxo de Exceção	Não se aplica
Alterar Resposta Questão - Nivelar Aluno		
	Item	Descrição
	Caso de Uso	Nivelar Aluno – Não se aplica
Excluir Resposta Questão - Nivelar Aluno		
	Item	Descrição
	Caso de Uso	Nivelar Aluno – Não se aplica
Consultar Resposta Questão - Nivelar Aluno		
	Item	Descrição
	Caso de Uso	Nivelar Aluno – Não se aplica

4.5.2. Agente de Alocação

O agente de alocação visa alocar o aluno em um nível específico dentro do sistema MedicalGame. Este agente é baseado na arquitetura de agente reativo simples, capaz de receber determinada informação, processá-la e gerar uma saída baseada no que foi recebido. Este agente pode receber e gravar informações no banco de dados do sistema MedicalGame. Ele também recebe e trabalha com dados gerados pelo Agente de

Nivelamento, que estão no banco de dados da aplicação, ademais de dados oriundos do *software* MedicalGame, para que a troca de níveis seja efetivada, conforme a configuração efetivada previamente pelo administrador do sistema.

Na Figura 31, pode-se verificar de forma clara as várias etapas de todo o fluxo da informação dentro do agente alocação, onde: A primeira etapa verifica se a quantidade de questões respondidas no nível é igual à quantidade de questões a serem respondidas, se o número de questões respondidas no nível em que o aluno está é menor do que o número de questões a serem respondidas dentro do nível, uma nova pergunta é apresentada, caso contrário, o sistema verifica se o total de acertos efetivados dentro do nível que o usuário está, é maior ou igual ao número de acertos necessários para o avanço de nível, se for o aluno, tem seu nível avançado, caso contrário, o aluno continua respondendo questões do nível que está.

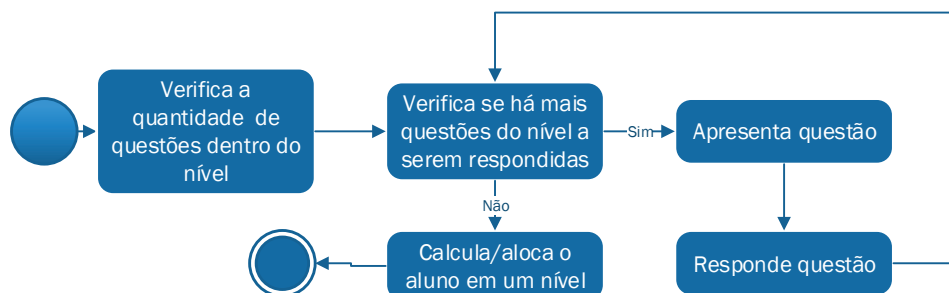


Figura 31: Fluxo de dados processo de alocação. Fonte: Elaborada pelo autor.

Na Tabela 3 apresenta-se a descrição do caso de uso Alocar Aluno, que traduz e descreve as atividades e características do Agente Alocação.

Tabela 3: Caso de uso – Agente de Alocação. Fonte: Elaborado pelo autor.

Manter Alocar Aluno	
Cadastrar Alocar Aluno – Alocar Aluno	
Item	Descrição
Caso de Uso	Alocar Aluno - <i>não se aplica para cadastrar, pois o cadastro é efetivado pelo agente de nivelamento.</i>
Alterar Alocar Aluno – Alocar Aluno	
Item	Descrição
Caso de Uso	Alocar Aluno – <i>não se aplica</i>
Resumo	O sistema solicita informações inseridas no banco de dados pelo (Agente Alocação), para verificar em que nível este Aluno está. Se for necessária nova alocação, esta é efetivada. São gerados logs deste caso de uso, que são inseridos em um banco de dados.

Ator	Agente Alocação.
Pré-condições	1 - Níveis de dificuldade das questões devem ser pré-cadastrados no sistema administrativo. 2 – A quantidade de questões por nível deve ser pré-configuradas no sistema administrativo. 3 – A quantidade de acertos por nível deve ser pré-configuradas no sistema administrativo.
Pós-Condições	O Aluno deve ser alocado em nível de dificuldade dentro da aplicação.
Fluxo Principal	1 – O sistema, verifica se a quantidade de questões respondidas no nível é igual à quantidade de questões a serem respondidas (pré-configuradas no sistema administrativo). 2 – O sistema, verifica se, o total de acertos efetivados dentro do nível que o usuário está, é maior ou igual ao número de acertos necessários para o avanço de nível, este já (pré-configurado no sistema administrativo). 3 – O sistema altera a informação referente ao nível de dificuldade vinculado ao usuário e o caso de uso termina.
Fluxo Alternativo	Fluxo Alternativo (1): Se o número de questões respondidas no nível em que o aluno está/é menor do que o número de questões a serem respondidas dentro do nível, (pré-configurado no sistema administrativo), uma nova questão é apresentada para o aluno. Fluxo Alternativo (2a): Se a quantidade de acertos efetivadas no nível for maior ou igual, o aluno tem seu nível acrescido em 1, ou seja tem o nível de dificuldade alterado dentro do sistema. Fluxo Alternativo (2b): Se a quantidade de acertos efetivadas no nível for menor do que foi no sistema administrativo, uma mensagem de aviso dizendo que o aluno não atingiu pontuação suficiente para alteração de nível é apresentada. Todos os registros de respostas referentes a aquele nível são transferidos para um log e o aluno inicia uma nova tentativa de troca de nível. Esse fluxo termina somente quando o aluno conseguir alcançar o mínimo de acertos daquele nível.
Deletar Alocar Aluno – Alocar Aluno	
Item	Descrição
Caso de Uso	<i>Alocar Aluno - Não se aplica</i>
Consultar Alocar Aluno – Alocar Aluno	
Item	Descrição
Caso de Uso	<i>Alocar Aluno - Não se aplica</i>

4.5.3. Agente Tutoria

Como terceiro agente, descreve-se o Agente Tutoria, que interage no momento em que o aluno efetiva a resolução de questões. Sabe-se que cada questão disponível no sistema tem três tentativas de respostas, somente a primeira resposta se efetivada de forma correta, computa pontos para a troca de níveis, nas outras tentativas, dicas e tutorias podem ser apresentadas.

O funcionamento do Agente Tutoria terá sua explicação baseada no fluxo de informações representadas na Figura 35. Como primeiro passo do processo, o Agente Tutoria recebe a resposta da questão do aluno, com esta variável há a verificação de acerto ou erro da questão, neste momento, dois fluxos são possíveis, ou seja: o primeiro fluxo é percorrido se o aluno efetivar de forma correta a resolução da questão, um registro de acerto é gravado na base de dados, e uma mensagem é apresentada para o aluno.

O segundo fluxo com possibilidade de ser percorrido, inicia-se também pela resposta do aluno, mas, nesse caso, a resposta à questão foi efetivada de forma errada. O Agente de Tutoria efetiva a gravação de um registro de erro na base de dados e uma mensagem de erro é apresentada para o aluno. Como passo seguinte, há uma verificação de qual é a tentativa de resposta da questão. Se for a primeira, o Agente Tutoria busca, na base de dados, a tutoria a ser apresentada, se esta questão não tiver primeira tutoria cadastrada, uma mensagem de que não existe tutoria é apresentada ao aluno.

Na Figura 32, a descrição referente a uma tutoria está sendo representada graficamente. Neste caso, o Agente Tutor apresentou um texto para que o aluno pudesse ter maior embasamento para a nova tentativa de resolução do problema ou da questão a ele apresentada.

Após a apresentação da tutoria, o sistema aguarda uma nova resposta vinda do usuário.

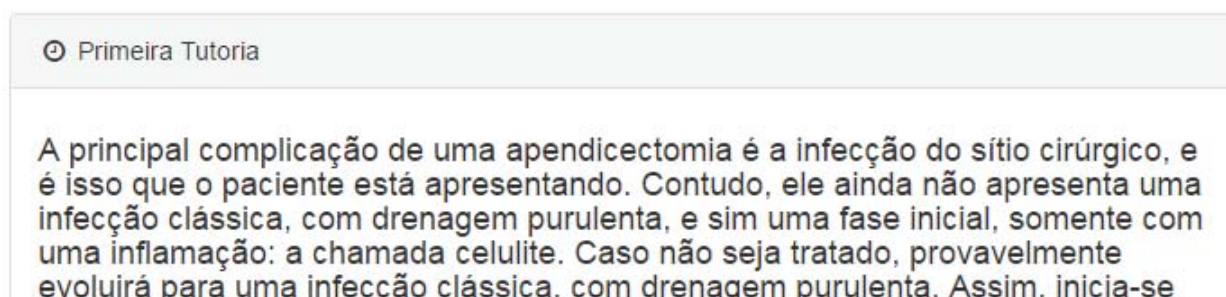


Figura 32: Representação gráfica da primeira tutoria. Fonte: Elaborado pelo autor.

Com a segunda tentativa de resposta tem-se dois fluxos distintos de informação onde: Se houve acerto, um log de acerto é gravado na base de dados e uma mensagem é

apresentada para o aluno. Se houver erro, o sistema verifica se é a primeira interação (neste caso é a segunda), se não for a primeira, verifica se é a segunda interação de resposta (nesse caso é), então o Agente Tutoria busca as informações da segunda tutoria e as apresenta ao aluno, a saber que esta tutoria pode ser composta por áudio, vídeo ou imagens, se esta questão não tiver tutoria uma mensagem de que não há tutoria é apresentada conforme Figura 33.

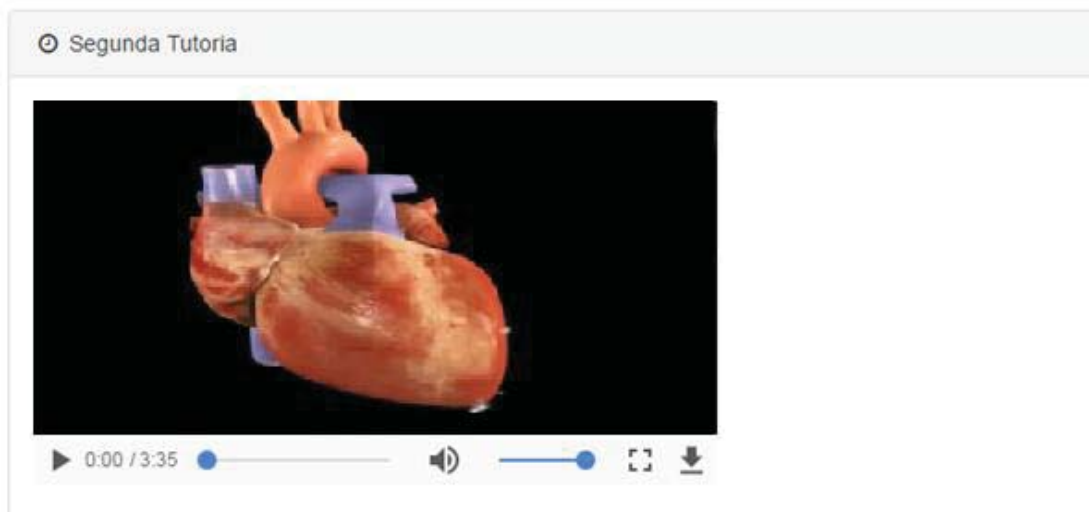


Figura 33: Representação gráfica da segunda tutoria. Fonte: Elaborado pelo autor.

Após a apresentação da segunda tutoria o sistema aguarda uma nova resposta vinda do usuário. Com a apresentação de uma nova resposta, dois fluxos distintos de informação são possíveis, onde: Se houve acerto, um log de acerto é gravado na base de dados e uma mensagem é apresentada para o aluno. Se houver erro, o sistema verifica se é a primeira tentativa de resposta (neste caso não é), se não for a primeira tentativa, verifica se é a segunda tentativa de resposta (nesse caso não é), se não for a segunda tentativa, verifica se é a terceira tentativa de resposta (que é o caso), nessa situação, o sistema retorna a mensagem de erro, que pode ser visualizada na Figura 34.

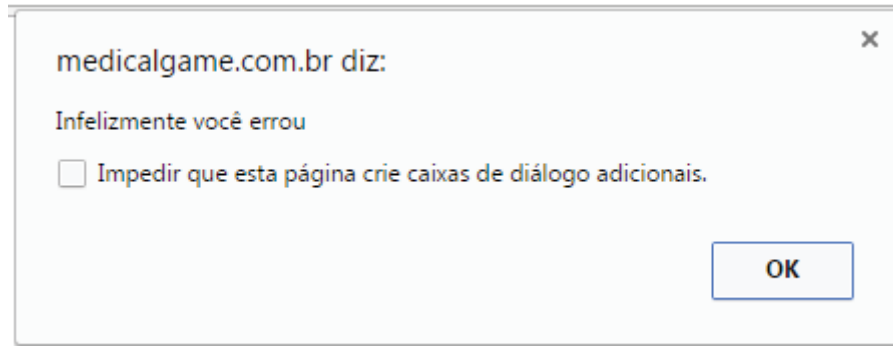


Figura 34: Representação gráfica, erro última tentativa de resposta. Fonte: Elaborado pelo autor.

Após a apresentação da última mensagem de acerto ou erro de questões para o aluno, o processo é finalizado e uma nova questão é apresentada, como pode ser visualizado no diagrama de fluxo de dados apresentado na Figura 35.

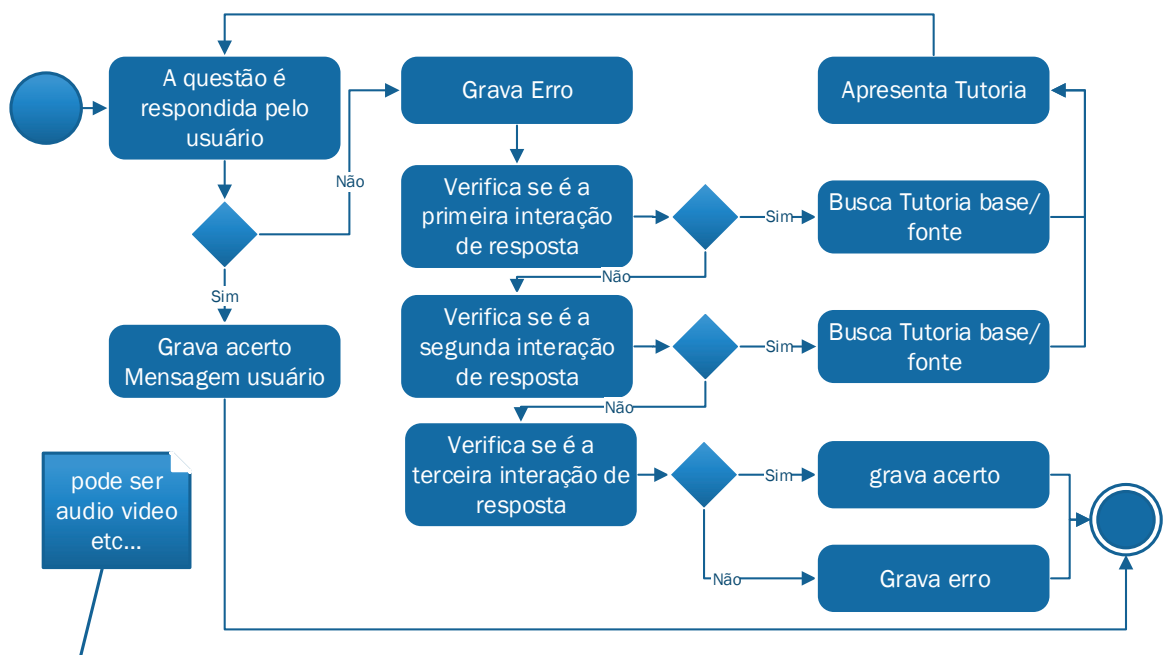


Figura 35: Caso de uso – Agente Tutoria. Fonte: Elaborado pelo autor.

Na Tabela 4 tem-se a descrição do caso de uso Consultar Tutoria, que traduz e descreve as atividades e características do Agente Tutoria.]

Tabela 4: Descrição do caso de uso Agente Tutoria. Fonte: Elaborado pelo autor.

Manter Tutoria	
Cadastrar Tutoria – Inserir tutoria	
Item	Descrição
Caso de Uso	<i>Cadastrar Tutoria – não se aplica – é efetivado no sistema administrativo – insert.</i>
Alterar Tutoria – Alterar tutoria	
Item	Descrição
Caso de Uso	<i>Alterar Tutoria – não se aplica – é efetivado no sistema administrativo – update.</i>
Deletar Tutoria – Excluir Tutoria	
Item	Descrição
Caso de Uso	<i>Excluir Tutoria - Não se aplica – é efetivado no sistema administrativo – delete.</i>
Consultar Tutoria – Consultar Tutoria	
Item	Descrição
Caso de Uso	Consultar Tutoria.
Resumo	A questão é respondida pelo aluno dentro do sistema e mensagens de tutoria podem ser apresentadas.
Ator	Agente Tutoria.
Pré-condições	1 - Níveis de dificuldade das questões devem ser pré-cadastrados no sistema administrativo. 2 - Quantidade de questões por nível devem ser pré-configuradas no sistema administrativo. 3 - Quantidade de acertos por nível devem ser pré-configuradas no sistema administrativo. 4 – Tutorias devem ser pré-configuradas no sistema administrativo. 5 – Questões devem ser respondidas pelo aluno.
Pós-Condições	Uma tutoria deve ser apresentada para o aluno.
Fluxo Principal	1 – O sistema aguarda uma resposta de uma questão. 2 – A questão é respondida pelo aluno. 3 – O sistema grava a informação de acerto ou erro. 4 – O sistema apresenta mensagem de erro ou acerto ao usuário e o caso de uso termina.
Fluxo Alternativo	Fluxo Alternativo (3a): Se a resposta do Aluno for correta é gravado um log desta informação e posteriormente o sistema apresenta uma mensagem de acerto. Fluxo Alternativo (3b): Se a resposta do aluno for errada, é gravado um log desta informação e posteriormente o sistema verifica se é a primeira interação (primeira tentativa de resposta), se for, o sistema efetiva um busca na base de dados, ou na fonte, por uma tutoria e posteriormente a apresenta para o aluno. Retornando o fluxo para o fluxo principal (1). Fluxo Alternativo (3c): Se a resposta do aluno for errada, é gravado um log desta informação e posteriormente o

		<p>sistema verifica se é a segunda interação (segunda tentativa de resposta), se for, o sistema efetiva um busca na base de dados, ou na fonte, por uma tutoria e posteriormente a apresenta para o aluno. Retornando o fluxo para o fluxo principal (1).</p> <p>Fluxo Alternativo (3d): Se a resposta do aluno for errada, é gravado um log desta informação e posteriormente o sistema verifica se é a terceira interação (terceira tentativa de resposta), se for o sistema retorna uma mensagem, informando que o aluno respondeu de forma incorreta a questão.</p> <p>Fluxo Alternativo (3): Se a resposta do aluno for correta, é gravado um log desta informação e posteriormente o sistema verifica se é a terceira interação (terceira tentativa de resposta), se for o sistema retorna uma mensagem, informando que o aluno acertou a resposta da questão.</p>
--	--	--

4.5.4. Agente Duelo

Apresenta-se a tela em que o agente duelo foi desenvolvido, e pode ser visualizado na Figura 36. Neste contexto, a implementação desta funcionalidade do sistema tem como característica o envolvimento do aluno na tarefa que está desenvolvendo.

O processo começa com a apresentação da tela intitulada Duelo, que se parece em muito com a tela de nivelamento. As grandes diferenças são a descrição da localização do usuário no STI e uma breve explicação do que deve ser executado. Estão dispostos o enunciado da questão e cinco alternativas, além de um botão denominado duelar.

Ao clicar neste botão, o aluno recebe uma mensagem, Figura 36, dizendo que a resposta foi guardada no banco de dados e desejando bons estudos ao aluno. O resultado deste duelo pode ser visualizado pelos usuários no sistema MedicalGame no *menu* relatórios/duelos, a partir do momento em que o usuário “desafiante” também terminar de responder as questões do duelo.

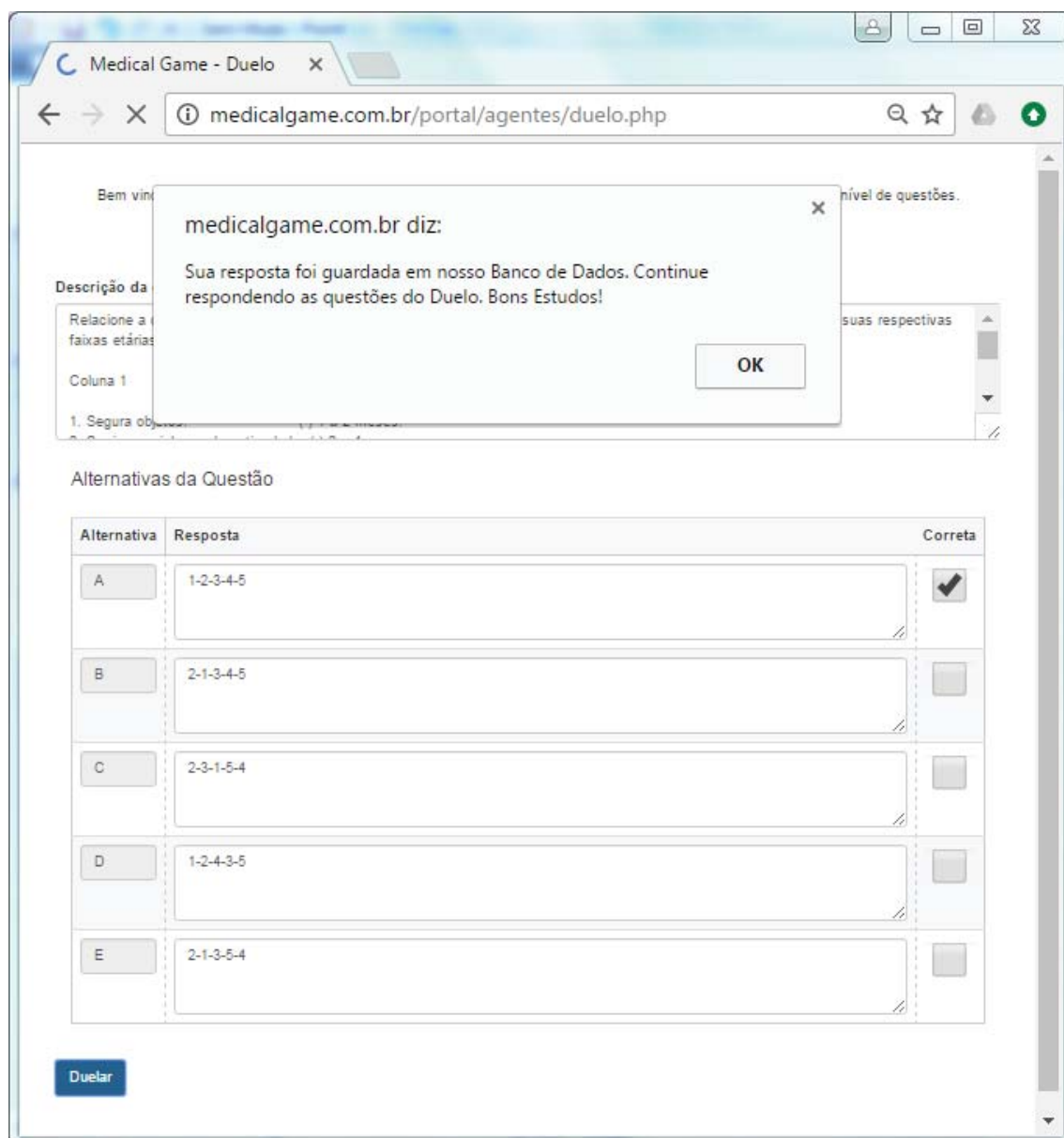


Figura 36: Tela de Duelo. Fonte: elaborada pelo autor.

A partir de então é apresentado o fluxo de dados referente ao agente intitulado Duelo. Esse fluxo pode ser visualizado na Figura 37, iniciando com a verificação de duelos cadastrados para o aluno em questão. Se existir, é efetivada uma verificação de qual nível o aluno está. Desta forma, solicitando ao banco de dados uma questão referente àquele nível especificamente, esta questão e suas alternativas são apresentadas na tela e posteriormente respondidas pelo aluno.

Na sequência, o agente duelo insere estas informações no banco de dados para computarem o score do desafio, apresentando uma mensagem sobre o que foi efetivado com a resposta e desejando bons estudos. Após este processo, faz-se uma verificação, se o número

de questões for maior que a quantidade de questões respondidas, o agente encaminha o fluxo de informações para a seleção e apresentação de questões referentes ao nível em discussão.

A partir do momento em que o número de questões respondidas do duelo for igual ao que foi configurado no módulo administrativo, o processo do agente se encerra.

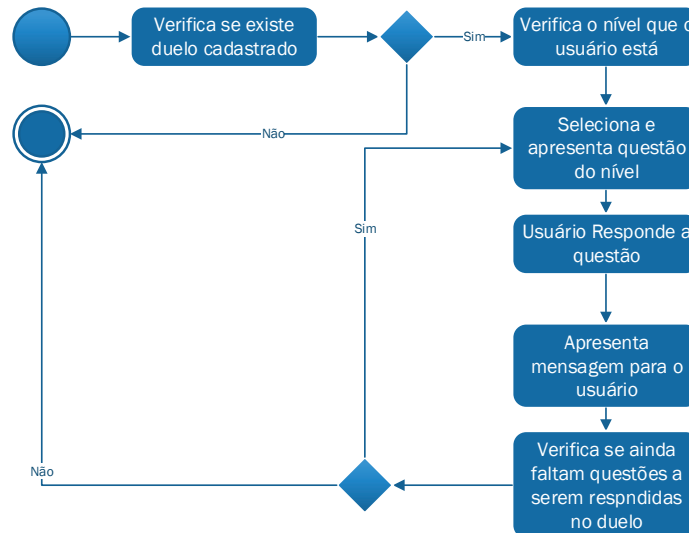


Figura 37: Fluxo de Dados do Agente Duelo. Fonte: elaborada pelo autor.

No **Erro! Fonte de referência não encontrada.**, referente ao caso de uso apresentar Duelo, está descrito o resumo de cada caso de uso, os atores envolvidos, as pré-condições, as pós-condições, o fluxo principal.

Tabela 5: Descrição do caso de uso Agente Duelo. Fonte: Elaborado pelo autor.

Manter Duelo	
Cadastrar Duelo – Inserir Duelo	
Item	Descrição
Caso de Uso	<i>Cadastrar Duelo – não se aplica – é efetivado no sistema administrativo – insert.</i>
Alterar Duelo – Alterar Duelo	
Item	Descrição
Caso de Uso	<i>Alterar Duelo – não se aplica – é efetivado no sistema administrativo – update.</i>
Deletar Duelo – Excluir Desafio	
Item	Descrição
Caso de Uso	<i>Excluir Duelo - Não se aplica – Não há opção no sistema.</i>
Apresentar Duelo – Apresentar Duelo	
Item	Descrição
Caso de Uso	Apresentar Duelo
Resumo	Questões referentes ao nível em que o aluno está são

		apresentadas, para que o mesmo as responda na forma de desafio, onde os dois primeiros colocados de cada nível de dificuldade de questões irão participar.
Ator		Agente Duelo
Pré-condições		1 – Níveis de dificuldade das questões devem ser pré-cadastrados no sistema administrativo; 2 – Duelos devem estar pré-configurados no sistema administrativo; 5 – Questões devem ser respondidas pelo aluno.
Pós-Condições		Uma mensagem informando que o duelo foi concluído deve ser apresentada.
Fluxo Principal		1 – O sistema apresenta uma questão do Duelo; 2 – O sistema aguarda uma resposta de uma questão; 3 – A questão é respondida pelo aluno; 4 – O sistema apresenta uma mensagem para o usuário; 5 – O sistema grava a informação de acerto ou erro; 6 – O sistema verifica se existem questões do duelo a serem respondidas; 7 – O sistema apresenta mensagem ao aluno e o caso de uso termina.
Fluxo Alternativo		Fluxo Alternativo (6a): Se existirem mais questões a serem respondidas dentro do duelo, o fluxo é retornado para o fluxo principal (1) Fluxo Alternativo (6b): Se não existirem mais questões a serem respondidas dentro do duelo, o fluxo é retornado para o fluxo principal (7).

5. ESTUDO DE CASO

Com o propósito de efetivar o teste de *realease* ou teste funcional do *software*, desenvolveu-se um estudo de caso.

Para a elucidação do estudo de caso utilizou-se o *software* MedicalGame, em testes que foram propostos com o objetivo da avaliação dos agentes nele desenvolvidos. Cita-se que a utilização do *software* MedicalGame, em nenhum momento, teve ou tem o intuito de avaliar o aprendizado de qualquer aluno que tenha participado do teste de *realease*/teste funcional de *software*. Por este motivo, os participantes envolvidos não possuíam compromisso de chegarem ao final de qualquer experimento, ou etapa do experimento/teste funcional de *software*.

Por conseguinte, as próximas seções apresentam as considerações éticas, o grupo avaliado, o experimento, a análise dos resultados alcançados, bem como discussões sobre os mesmos.

5.1. CONSIDERAÇÕES ÉTICAS

Todos os participantes que foram convidados para participar do teste funcional do *software*, denominado MedicalGame, assinaram o Termo de Consentimento, que está disposto no Apêndice A -Termo de Consentimento.

5.2. CARACTERIZAÇÃO DOS PARTICIPANTES

Participaram do teste funcional de *software* quatorze alunos, seis desses do sexo masculino e oito do sexo feminino. Estão, respectivamente, cursando os seguintes níveis da graduação: um aluno no segundo nível, cinco alunos no terceiro nível, três alunos no quarto nível, três alunos no sexto nível, dois alunos no sétimo nível de graduação, todos vinculados à Faculdade de Medicina da Universidade de Passo Fundo – UPF, estes alunos foram previamente convidados pelo coorientador do trabalho.

5.3. EXPERIMENTO

O teste de *realease* ou também chamado de teste funcional de *software*, foi realizado no dia 16 de Janeiro de 2017, no Laboratório de Informática cedido pela Faculdade de Medicina da Universidade de Passo Fundo. A sala conta com o ambiente de 20 computadores, como pode ser visualizado na Figura 38.



Figura 38: Laboratório de Informática da Faculdade de Medicina UPF.

O convite aos participantes do teste de avaliação foi efetivado pelo Dr. Cassiano Mateus Forcelini, coorientador deste trabalho e coordenador de Pesquisa e Pós-Graduação da Faculdade de Medicina da Universidade de Passo Fundo.

Os participantes que aceitaram o convite foram recepcionados, pelo pesquisador, na sala em que o teste funcional do *software* foi executado. Quando os 14 alunos convidados estavam presentes na sala, foram instruídos pelo pesquisador sobre o procedimento de preenchimento do termo de consentimento (Apêndice A -).

Na sequência, o Dr. Roberto dos Santos Rabello, orientador do trabalho, efetivou uma breve fala sobre a pesquisa, posteriormente o Dr. Cassiano Mateus Forcelini também efetivou uma fala, sobre o trabalho e sua importância para o estudo e preparação para a prova de residência médica, assim como para estudos para provas de disciplinas durante a graduação.

No decorrer do encontro demonstrou-se a forma em que o teste de *realease* iria ocorrer, baseado na história apresentada no Apêndice D -.

A partir deste momento, os participantes foram convidados a iniciar o teste de *realease* do *software* MedicalGame.

5.4. TESTE DE REALEASE E DISCUÇÃO DOS RESULTADOS

Esta seção tem por objetivo discutir os resultados obtidos durante o teste *realease* dos agentes de *software*, presentes no sistema MedicalGame, sendo eles: O Agente de Nivelamento; O Agente de Alocação; O Agente de Tutoria e o Agente de Duelo respectivamente. Para o teste de *realease*, o banco de dados da aplicação contou com 250 questões das cinco áreas da medicina, estas também subdivididas em cinco níveis de dificuldade. Estas questões foram desenvolvidas pela equipe de medicina que trabalhou durante a execução do projeto.

5.4.1. Agente de Nivelamento

O Agente de Nivelamento, primeiro agente a ter suas interações discutidas, teve sua configuração para a execução dos testes com os seguintes parâmetros. Foram respondidas duas questões sorteadas de forma randômica, para cada nível de dificuldade configurado no módulo administrativo do sistema, desta forma se teve um total de 10 questões a serem respondidas por cada aluno.

A alteração de nível se daria com as seguintes condições pré-definidas no sistema: se o número de acertos total, fosse menor ou igual a três, o aluno iniciaria a resolução de questões no nível 1 de dificuldade. Se o número de acertos fosse maior ou igual a 4 ou menor ou igual a 7 o aluno seria elencado a ter acesso ao nível 2 da aplicação. Se o número de acertos efetivados pelo aluno fosse maior ou igual a 8 e menor ou igual a 10, o aluno seria alocado no nível de dificuldade 3.

Seguiu-se este padrão de configuração devido à solicitação efetivada pelos professores de Medicina que fizeram parte do projeto, ou seja, nenhum aluno deveria ser alocado em um nível maior do que o terceiro, mesmo acertando a totalidade das questões. Outro item importante a ser levantado diz respeito à alocação destes alunos aos níveis alcançados, para que se seguisse uma padronização da resolução das questões de todos os níveis. O Agente de Nivelamento, ao final da resolução das questões, mostrava uma mensagem dizendo em qual nível o usuário seria alocado, mas iniciaria a resolução no nível inicial, ou seja, o nível de dificuldade 1.

Dados do Agente de Nivelamento foram organizados e apresentados na Figura 39, em que se verifica que 14 alunos acessaram o sistema MedicalGame e interagiram com o Agente de Nivelamento, com as configurações apresentadas anteriormente.

Além da informação da quantidade de alunos que acessaram o sistema MedicalGame, apresentam-se as seguintes informações: Na cor verde, o nível em que o aluno

seria alocado após a resolução do nivelamento; os dados na cor vermelha referem-se à quantidade de questões respondidas de forma errada; e os dados na cor azul representam a quantidade de questões respondidas de forma correta.

Verificou-se que 50% dos alunos que participaram do teste do Agente de Nivelamento teriam alterado seu nível dentro do sistema MedicalGame; 5 deles, o que equivale a 35,7%, seriam alocados no nível de dificuldade 2, e 2 deles, equivalendo a 4,3%, seriam alocados no nível 3 de dificuldade, os 50% restante, equivalente a 7 alunos, seriam alocados no nível mais básico ou seja no nível 1.

Com a concatenação destes resultados e baseando-se na configuração efetivada previamente ao agente de nivelamento, verificou-se que o mesmo cumpriu com os parâmetros de sua configuração, pois se verificou que os alunos com identificadores 52, 54, 55, 58, 59, 60 efetivaram a quantidade de acertos menor ou igual a 3, o que quer dizer que seriam alocados no nível de dificuldade 1. Os alunos com os identificadores 53, 56, 57, 63, 65 seriam alocados ao nível 2, pois tiveram a quantidade de respostas corretas entre 4 e 7, e os alunos, com identificador 62 e 66 seriam alocados no nível de dificuldade 3, pois tiveram a quantidade de acertos maior ou igual a 8 e menor ou igual a 10.

Verificou-se, desta forma, que o Agente de Nivelamento se comportou de forma assertiva em 100% dos casos apresentados, deixando clara sua importância no processo de Nivelamento de alunos na Aplicação MedicalGame.

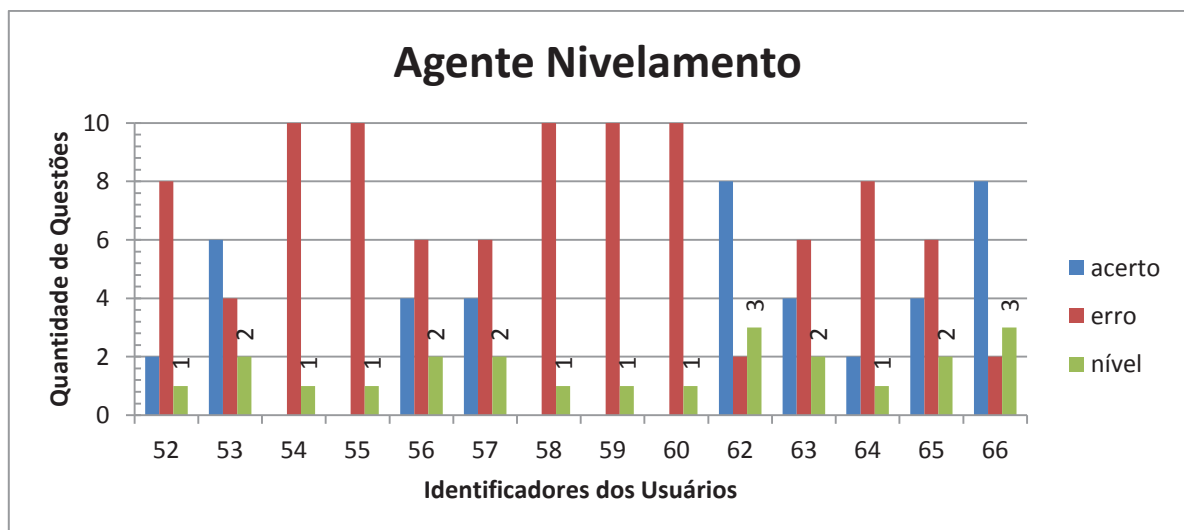


Figura 39: Dados Agentes Nivelamento. Fonte: elaborado pelo Autor.

5.4.2. Agente de Alocação

Na Figura 40, estão apresentados dados referentes ao Agente de Alocação, que teve sua configuração efetivada para o teste com os seguintes parâmetros: a) O número de questões a serem respondidas por nível foi de 10; b) O número mínimo de acertos dentro do nível é de 7 questões, sendo que estes deveriam ser efetivados na primeira tentativa de resolução da questão.

A Figura 40 **Erro! Fonte de referência não encontrada.** tem como intuito a demonstração dos dados referentes à alocação de alunos dentro do sistema MedicalGame, ou seja, verificar se o Agente de Alocação efetivou suas tarefas de forma correta, efetivando a troca de níveis baseado em sua configuração proposta.

Na Figura 40, optou-se pela utilização de dados aninhados, para que a interpretação fosse simplificada, ou seja, no eixo horizontal estão a quantidade de questões dos níveis e a descrição da troca de cada nível. No eixo vertical apresenta-se a quantidade de respostas efetivadas de forma correta para cada aluno; e dentro de cada nível de dificuldade, da esquerda para a direita: nível1, nível2, nível3, nível4, nível5, com seus devidos identificadores.

Pôde-se verificar na Figura 40 que, com base na configuração efetivada previamente e já apresentada, o Agente de Alocação executou sua tarefa de forma satisfatória em 100% das alocações de alunos. Como exemplo, no aluno com identificador 52, a troca do nível 1 para o nível 2 foi efetivada, com 100% de acertos nas questões em sua primeira tentativa, do nível 2 para o 3 e do 3 para o 4, as trocas de nível ou alocações foram efetivadas com oito acertos, e no nível 5 o aluno teve 3 acertos computados, não conseguindo chegar ao final deste nível. Outro exemplo são os alunos com identificadores 60 e 63, que apresentaram apenas uma troca de nível, do nível 1 para o nível 2 com um total de 8 acertos não conseguindo efetivar nenhuma resposta correta no nível 2 dentro do teste estipulado.

Notou-se, com a apresentação e discussão dos dados apresentados na Figura 40, que o Agente de Alocação teve participação importante na alteração ou troca de nível dos Alunos e se portou de forma concisa e assertiva em 100% de suas interações.

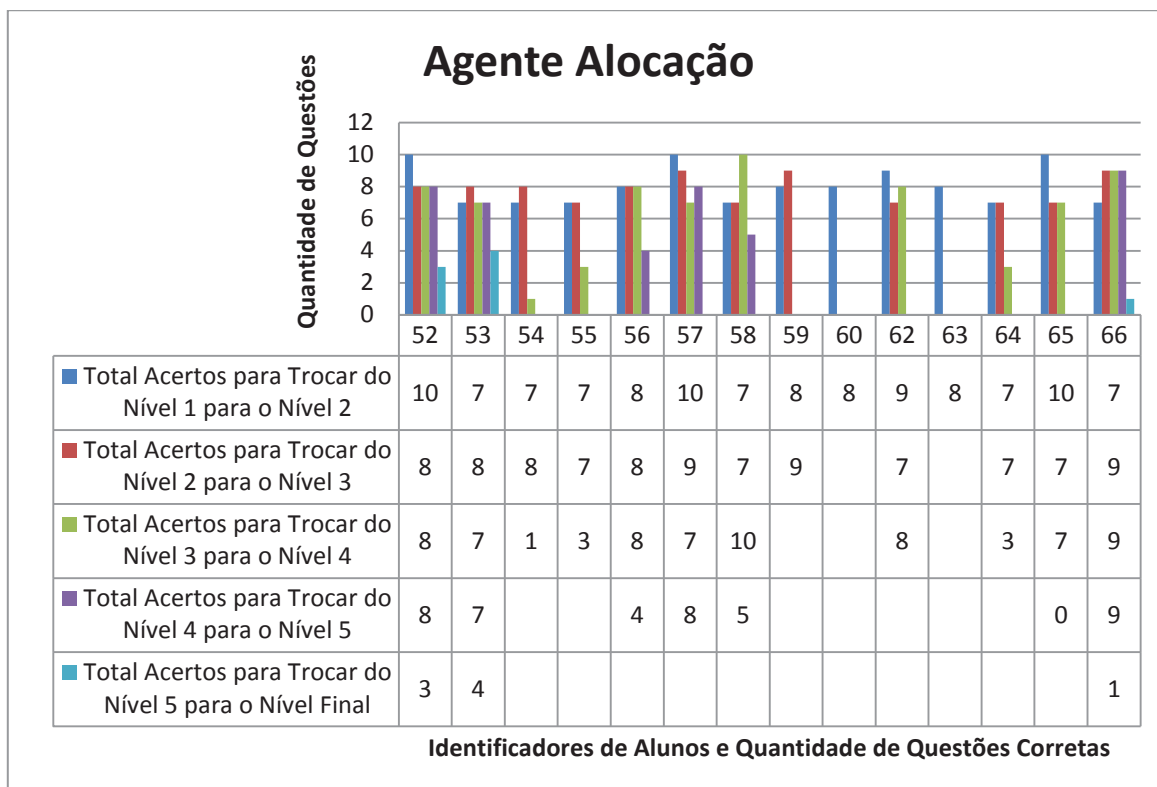


Figura 40: Dados do Agente de Alocação. Fonte: Elaborado pelo autor.

5.4.3. Agente de Tutoria

Inicia-se nesta seção a descrição da configuração e a apresentação dos resultados do teste proposto para o Agente de Tutoria. O agente de tutoria objetivou a apresentação de diversas formas de tutoria para o aluno. A apresentação destas tutorias é realizada logo após a verificação de erro da questão, e dentro de qualquer das três tentativas de resolução. Sendo assim, no teste do Agente de tutoria foram gerados dados para todos os níveis de dificuldade do sistema MedicalGame, e estes serão apresentados em forma de cinco gráficos.

Inicia-se a apresentação dos dados do Agente de Tutoria do nível 1 de dificuldade que é demonstrado na Figura 41, em que este agente não foi utilizado pelos alunos com identificador 52, 57, 65, isto porque como demonstrado na Figura 40, estes alunos responderam de forma correta todas as questões do nível 1, conseqüentemente, nenhuma tutoria foi apresentada.

Outro ponto importante a ser considerado é que em cerca de 50% dos alunos, a apresentação de tutoria contribuiu para a resolução da questão de forma correta, o que foi visualizado nos seguintes casos: o aluno com identificador 53, que respondeu três questões de

forma incorreta na primeira tentativa de resolução, após a tutoria ser apresentada, em nova tentativa de resolução destas questões, teve o número de erros decrescido para 1.

Já nos alunos com Identificador 54, 58, 62, 63 a apresentação de tutorias pôde ser considerada como mais efetiva, pois em todos estes casos em que as questões foram respondidas de forma errada na primeira tentativa de resolução, foram respondidas de forma correta após a apresentação da tutoria.

Neste nível de dificuldade, foram apresentadas 48 tutorias, sendo destas 27 apresentadas na primeira tentativa de resolução das questões, 13 na segunda tentativa de resolução das questões e 8 tutorias para a terceira tentativa de resolução das questões.

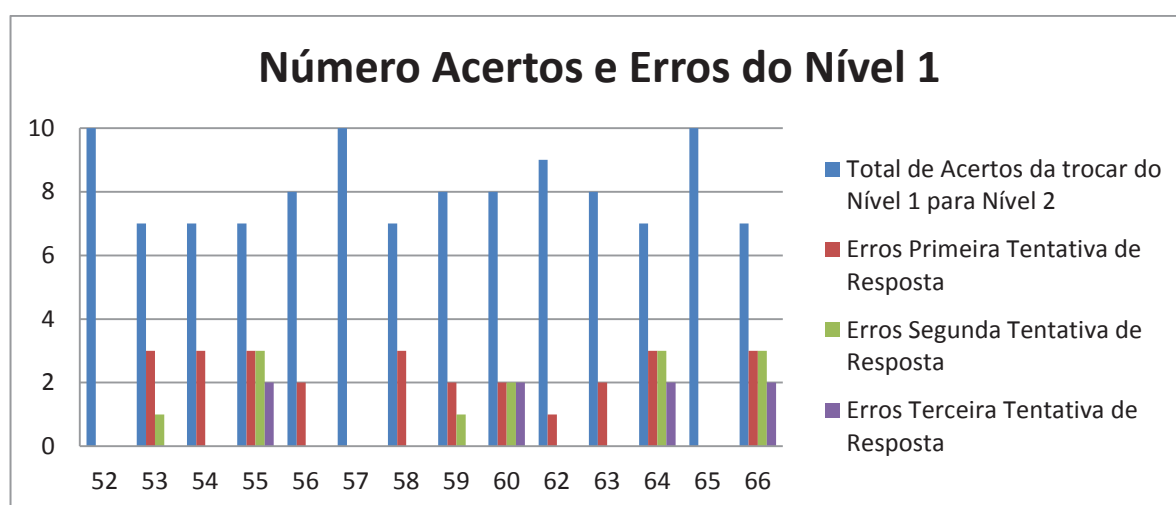


Figura 41: Número Acertos e Erros do Nível 1. Fonte: Elaborado pelo autor.

Como segundo grupo de dados apresentado, tem-se o número de tutorias do nível 2 que é visualizado na Figura 42. Neste conjunto de dados, averiguou-se que 50% dos alunos, os com identificador 52, 53, 54, 56, 57, 65, 67, responderam de forma incorreta questões na primeira tentativa de resolução, mas com a tutoria apresentada, na segunda tentativa de resolução conseguiram responder de forma correta a totalidade das questões que haviam sido respondidas de forma incorreta. Já para os alunos com identificador 55, 58, 62 e 64 a tutoria apresentada melhorou em aproximadamente 66% a apresentação de respostas corretas. O número total de tutorias apresentados neste nível foi de 33, onde 27 destas para a primeira tentativa de resolução, 5 para a segunda tentativa de resolução e 1 tutoria para a terceira tentativa de resolução. Uma observação importante a ser considerada a partir da Figura 42, é de que cadeias de dados de alguns identificadores de alunos não estão sendo apresentadas, isto porque os mesmos não efetivaram transição de nível dentro da aplicação, como exemplo o aluno com identificador 63.

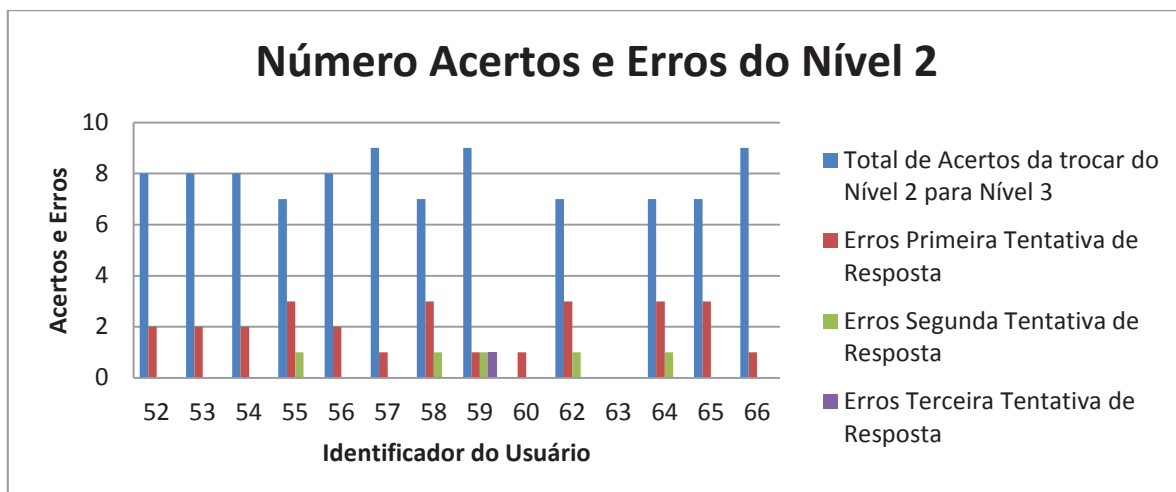


Figura 42: Número Acertos e Erros do Nível 2. Fonte: Elaborado pelo Autor.

No nível três de dificuldade da aplicação, apresentado na Figura 43, foram utilizados 28 tutorias, destes, nenhum para a terceira tentativa de resolução das questões, 5 para a segunda tentativa de resolução das questões e, 23 apresentadas para a primeira tentativa de resolução das questões. Neste gráfico, pôde-se destacar o aluno com identificador 64 que teve uma melhoria significativa no acerto das questões após a apresentação das tutorias. Os alunos com identificador 52, 53, 55, 59, 62 e 65 também tiveram acertos após a apresentação da primeira ou da segunda da tutoria, o único aluno que apresentou o mesmo desempenho após a apresentação da primeira e segunda tutoria foi o de identificador 66, ou seja, não conseguiu acertar a questão mesmo utilizando-se das tutorias a ele apresentadas.

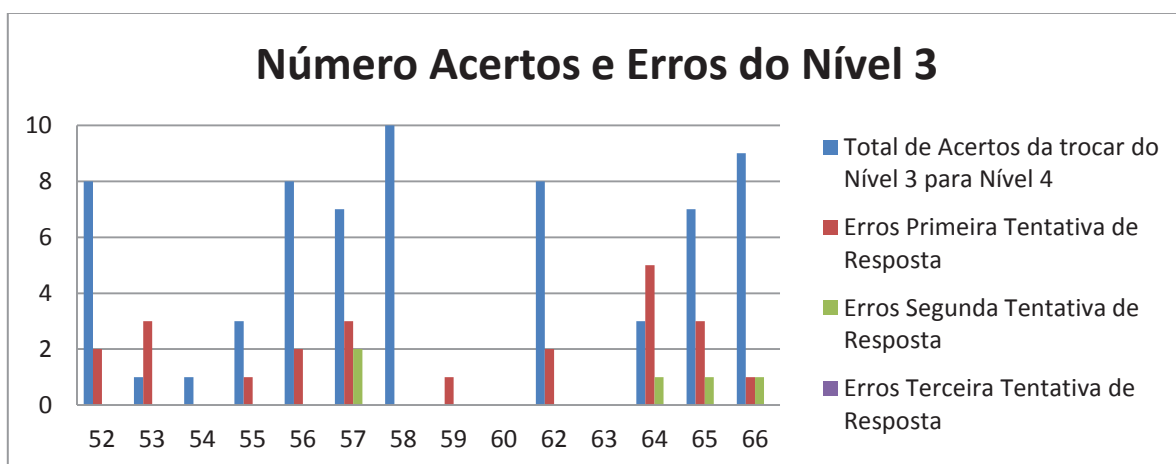


Figura 43: Número Acertos e Erros do Nível 3. Fonte: Elaborado pelo Autor.

Pela apresentação da Figura 44, constatou-se que a quantidade de identificadores com dados atrelados diminuiu em 50%, isso se dá porque o restante dos alunos não conseguiu chegar a este nível de dificuldade, dentro do tempo estipulado no teste proposto.

No nível 4 de dificuldade, de um total de 41 questões, em 11 delas foram apresentadas tutorias, destas 10 tutorias para erros efetivados na primeira tentativa de resposta e somente 1 aluno, o com identificador 52, teve de utilizar a tutoria para a segunda e terceira tentativa de resolução ou de resposta da questão.

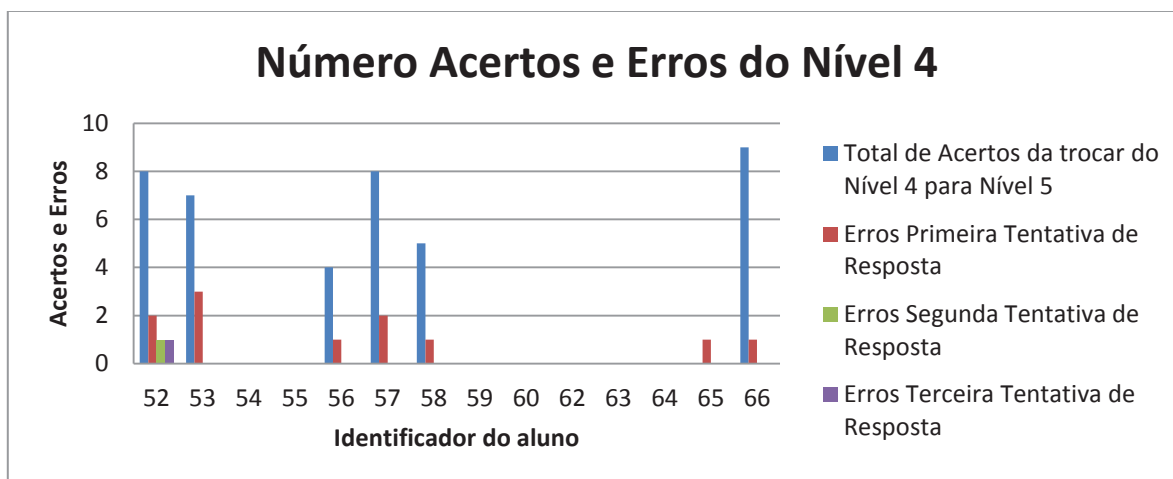


Figura 44: Número Acertos e Erros do Nível 4. Fonte: Elaborado pelo Autor.

Os dados referentes à apresentação de tutorias do nível cinco podem ser visualizados na Figura 45. Os números basearam-se em três identificadores de alunos, sendo eles 52, 53 e 66, neste nível da aplicação foram apresentadas 23 tutorias, destas 19 para o aluno com identificador 52, ocorrendo somente um acerto após a apresentação das tutorias no primeiro erro da questão, na segunda, as questões foram respondidas de forma incorreta, mesmo assim. Para o aluno com identificador 53, as tutorias se mostraram eficientes, pois o mesmo teve 2 erros na primeira tentativa e após a apresentação das tutorias para as questões o mesmo respondeu uma de forma correta. O Aluno com identificador 66 resolveu 1 questão de forma correta, sem utilizar tutoria, e efetivou a tentativa de resolução de outras duas questões sendo que em uma delas utilizou tutoria na primeira tentativa e a respondeu de forma correta, e em outra houve tutoria para a segunda tentativa de resposta e a mesma foi respondida de forma incorreta.

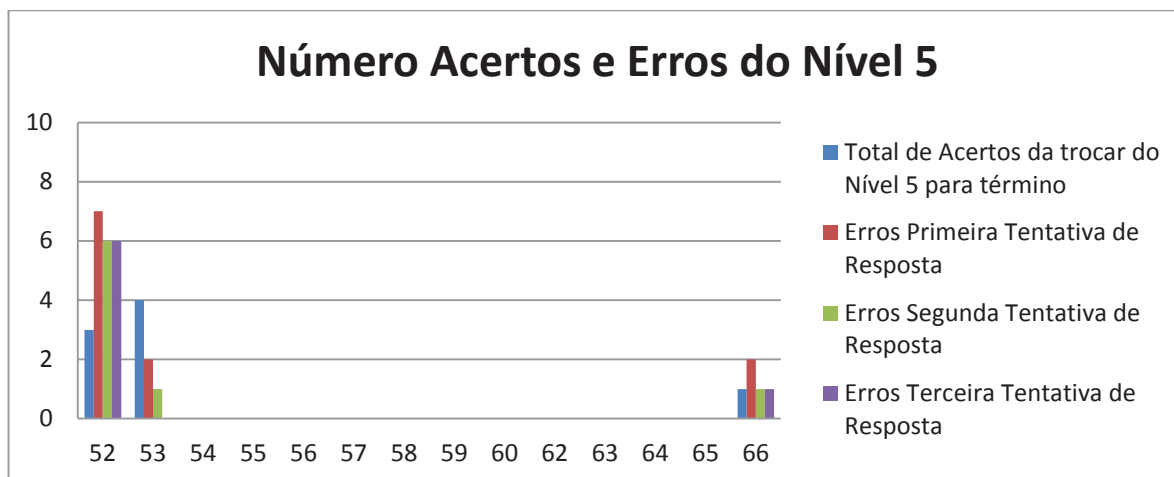


Figura 45: Número Acertos e Erros do Nível 5. Fonte: Elaborado pelo Autor.

5.4.4. Agente Duelo

O agente implementado, nomeado como Agente de Duelo, teve sua funcionalidade testada somente durante sua implementação. Ou seja, durante a criação de sua estrutura foram simuladas as situações referentes à sua utilização, da seguinte forma: o Agente de Duelo foi configurado para determinado nível de dificuldade de questões, ou seja, um duelo, entre os dois primeiros colocados deste nível foi criado. Logo depois, dois usuários de teste descritos como usuário1 e usuário2 foram criados e alocados neste nível como primeiro e segundo colocado.

Dando continuidade ao processo de teste funcional deste agente, utilizou-se a funcionalidade do duelo, pelos dois usuários criados e citados anteriormente. Primeiro o sistema foi acessado pelo usuário1, que respondeu as questões de duelo, sendo que as respostas foram inseridas na base de dados do STI. Na sequência, o usuário2 acessou o sistema, as questões do duelo foram apresentadas a ele e posteriormente respondidas.

Para a verificação do relatório desta funcionalidade, os dois usuários, usuário1 e usuário2, acessaram também o relatório de duelos respondidos, que demonstrou o primeiro e o segundo colocado do duelo, com a colocação correta de cada um, ou seja, quem ficou em primeiro e quem ficou na segunda colocação.

Sabe-se que para cada nível, o aluno pode ter duelos cadastrados, desta forma são apresentadas áreas para estes possíveis níveis na tela de relatório de duelos. Durante o teste funcional organizado e efetivado durante a implementação deste agente, erros não foram relatados, e o agente de duelo se comportou conforme foi programado.

6. CONCLUSÃO

Os desafios enfrentados pelos alunos ao realizarem a prova de residência médica são grandes, principalmente pela exigência, diversidade e complexidade dos conteúdos abordados, permitindo que ferramentas de apoio na sua preparação possam colaborar e vir a ser um diferencial na sua aprovação.

Pensando justamente nesta dificuldade, o presente trabalho visou o desenvolvimento de um protótipo de STI dotado de agentes de *software*, com o objetivo de apoiar a preparação para a prova de residência médica. Após o desenvolvimento do protótipo, um teste funcional foi realizado, com a validação dos quatro agentes definidos, sendo eles: Agente de nivelamento; Agente de tutoria; Agente alocação e Agente duelo. Uma reflexão a cerca dos resultados oriundos do teste funcional demonstrou que a utilização dos agentes de *software* contribuiu quanto à resolução dos exercícios propostos para os alunos. Além disso, também permitiu a utilização de diferentes recursos didáticos na apresentação de conteúdos e tutorias.

Verificou-se, a partir da utilização dos dois módulos, tanto o administrativo quanto o *frontend*, que a utilização de agentes de *software* em STIs é importante e pode auxiliar os alunos a melhorarem seu desempenho em testes e avaliação, bem como quanto à apresentação de conteúdos específicos para cada aluno, conforme o nível de dificuldade das questões selecionadas para resolução.

A arquitetura proposta para a criação do protótipo de STI foi baseada na arquitetura denominada tripartida que, em uma visão geral, conta com um banco de dados, onde os dados, tanto do sistema administrativo do STI, quanto os do sistema aqui denominado MedicalGame, podem ser inseridos, consultados e tratados. Por meio destas aplicações, os modelos básicos que compõem um STI foram contemplados, sendo eles: O Modelo de Domínio; O Modelo Aluno; O Modelo Pedagógico; e o Modelo de Interface. Com esta modelagem especificada, implementou-se os seguintes agentes de *software*: O Agente de Nivelamento; O Agente de Alocação; O Agente de Tutoria; E o Agente de Interface.

Para o desenvolvimento dos agentes, utilizou-se o conceito de agentes reativos simples, que têm como sua premissa a relação de atuação, baseada em estímulos e repostas, examinando regras específicas para essas ações e desencadeando uma reação. Os agentes foram desenvolvidos em linguagem de programação, sem a utilização de *frameworks* específicos.

Outro ponto importante no desenvolvimento deste trabalho foi a utilização das tecnologias e ferramentas como: CSS, HTML5, *BootStrap*, PHP e MySQL, por serem tecnologias e *softwares* livres, assim como disponibilizarem atualização constante. Estas características possibilitam a expansão e o uso de novas ferramentas dentro do contexto do *software* que foi desenvolvido.

O desenvolvimento do *software* obedeceu a arquiteturas de construção de *softwares* baseado em métricas de Engenharia de *Software*. Além disso, seguiu o desenvolvimento de documentação de requisitos de *software*. Esta implementação foi dividida em três partes efetivas, sendo elas: A página institucional, que teve como intuito a divulgação e a descrição sucinta do trabalho aqui desenvolvido; a página administrativa, esta com as funcionalidades de configuração e cadastro de perguntas e respostas para o STI e a configuração dos parâmetros de funcionamento dos agentes de *software*; a página do sistema MedicalGame, com: O modelo aluno; O modelo tutor; O modelo de interface; e parte do modelo de Domínio foi implementado. Também fez-se a utilização dos agentes de *softwares* desenvolvidos.

Quatro agentes de *softwares* foram desenvolvidos durante a implementação deste trabalho, quais sejam: O agente de nivelamento, que tem como objetivo a verificação e validação do nível de dificuldade a que o aluno será atrelado dentro do sistema MedicalGame; O agente de alocação, que visa alocar o aluno em um nível específico dentro do sistema MedicalGame; Como terceiro agente, o Agente Tutoria, que interage no momento em que o aluno efetiva a resolução de questões. E o agente duelo, que tem como característica o envolvimento do aluno na tarefa que está desenvolvendo. Após o desenvolvimento e configuração dos três módulos descritos, optou-se por um estudo de caso, com o propósito de efetivação de um teste funcional, dos agentes de *softwares* implementados.

Perante os resultados dos testes de *realease*, que foram propostos e executados para os agentes, chegou-se aos seguintes resultados: Agente de nivelamento teve resultados assertivos em 100%, deixando clara sua importância no processo de Nivelamento de alunos na Aplicação MedicalGame; O Agente de Alocação, após o teste de *realease*, também apresentou 100% de assertividade na alocação dos alunos; Nos resultados dos testes do Agente de Tutoria, verificou-se que o mesmo se mostrou eficiente quanto à apresentação de tutorias das questões para os alunos, bem como a possibilidade de aumento de acertos de questões após essa apresentação. O agente nomeado como Agente de Duelo teve sua funcionalidade testada, de forma simulada, ou seja, pelos próprios pesquisadores.

Considera-se que os benefícios da arquitetura e sistemas aqui propostos e desenvolvidos tendem a se consolidar no decorrer do tempo, por meio de um processo natural de evolução, baseado em novas ideias e *feedbacks*, tanto de professores, quanto de alunos que venham a utilizar os mesmos.

6.1. DISSEMINAÇÃO DO CONHECIMENTO

Compartilhar conhecimento é parte importante em qualquer produção científica. A ciência se constrói por muitas expressões que interagem entre si, de modo complementar ou opositivo. Neste sentido, durante o decorrer do desenvolvimento deste trabalho, desenvolveram-se atividades voltadas à disseminação do conhecimento gerado, estas elencadas na sequência:

- Apresentação de resumo do presente trabalho na Semana do Conhecimento⁷.

Este evento foi organizado pela Universidade de Passo Fundo, e é integrado com a Mostra de Iniciação Científica e a Mostra de Extensão. A apresentação do trabalho ocorreu no dia 4 de Novembro de 2015.

- Redação e submissão do artigo intitulado, Novas TICs aplicadas à educação, ao *e-book*, A pesquisa em educação e tecnologias: entre perguntas e respostas⁸, organizado pela URI – Universidade Regional Integrada do Alto Uruguai e das Missões.

6.2. TRABALHOS FUTUROS

Considerando-se as características da aplicação MedicalGame, em potencial, sua característica, enquanto ferramenta de apoio a estudos, torna possível traçar diversos trabalhos futuros que podem ser associados a mesma. O primeiro deles refere-se à construção e geração de relatórios gerenciais que, dispostos no módulo administrativo do *software*, podem gerar informações consistentes, embasados nos dados originados pelos alunos enquanto utilizam a aplicação.

Também é interessante tornar o sistema MedicalGame, além de acessível por meio de navegadores em celulares, também a partir de aplicativos para dispositivos móveis, especificamente desenvolvidos com o uso de tecnologias nativas de cada plataforma operacional. A relevância deste trabalho está fundamentada no avanço da tecnologia móvel

⁷ <http://www.upf.br/semanadoconhecimento>

⁸ <http://www.fw.uri.br/site/publicacoes/?area=aluno>

perante a sociedade em geral. Ao torná-la mais próxima dos alunos, o sistema MedicalGame, pode ganhar mais visibilidade, aprimorando-se assim a interação.

Outra proposta de interesse seria a de configuração e aplicação de provas de residência médica simuladas em que o professor, ou o administrador do grupo de alunos, poderá, ao final de determinado conteúdo, selecionar questões referentes ao que foi tratado e disponibilizar esta prova para a turma. Posterior à resposta da turma, o sistema geraria o resultado final da avaliação e gráficos sobre a avaliação dos alunos. Esta nova característica que pode ser implementada tendo como objetivo um maior envolvimento entre professor e alunos dentro do âmbito do STI.

Proposta inerente, também foi constatada durante o processo de desenvolvimento e documentação do sistema MedicalGame, como se sabe, a apresentação das questões dentro de cada nível é efetivada de forma aleatória, ou seja, questões das cinco áreas da medicina podem ser apresentadas, dentro de seus respectivos níveis de dificuldade. Neste contexto, alunos podem ter a necessidade de estudarem questões de determinada área da medicina de forma individual. Sendo assim, a criação de uma configuração que possa ser efetivada pelo aluno, em seu perfil, deve ser implementada, para que o mesmo tenha a liberdade de escolha perante as questões que a ele são sorteadas e apresentas.

REFERÊNCIAS

- [1] EDUCAÇÃO, M. D. Residência Médica. *Residência Médica - Ministério da Educação*, 2015. Disponível em: <http://portal.mec.gov.br/index.php?option=com_content&view=article&id=12263&Itemid=507>. Acesso em: 23 out. 2015.
- [2] HUYLMER LUCENA CHAVES. Vagas na Residência Médica no Brasil: Onde estão e o que é avaliado. *Revista Brasileira de Educação Médica*, n. 37(4), p. 557-567.
- [3] CREMERS. CREMERS - Conselho Regional de Medicina do Estado do Rio Grande do Sul. *CREMERS - Conselho Regional de Medicina do Estado do Rio Grande do Sul*, 2017. Disponível em: <<http://www.cremers.org.br/>>. Acesso em: 03 abr. 2017.
- [4] EBC, A. Agência Brasil. *Em São Paulo exame reprova mais da metade dos recém-formados em medicina*, 2015. Disponível em: <<http://agenciabrasil.ebc.com.br/educacao/noticia/2015-01/exame-do-conselho-reprova-mais-da-metade-dos-recem-formados-em-medicina>>. Acesso em: 23 out. 2015.
- [5] SALEM, A. C. Cremesp estuda "vigiar" e capacitar os egressos que tiveram mau desempenho na prova. *Academia Médica*, 2015. Disponível em: <<http://academiamedica.com.br/exame-cremesp/>>. Acesso em: 23 out. 2015.
- [6] KARIM QAYUMI A, Q. T. Computer-assisted learning: cyberPatient - a step in the future of surgical education. *J Invest Surg*, v. 12(6), p. 307-317, 1999.
- [7] FARRIMOND H, D. T. C. A. R. L. Development and evaluation of an e-learning package for teaching skin examination. *Action research*, v. 155(3), p. 592-599, 2006.
- [8] MNGUNI, M. The theoretical cognitive process of visualization for science education. *Springerplus*, p. 184-187, 2014.
- [9] MEDECEL. MEDECEL. *MEDECEL*, 2017. Disponível em: <<http://www.medcel.com.br/>>. Acesso em: 13 abr. 2017.
- [10] SJT. SJT. *SJT*, 2017. Disponível em: <<http://www.sjteducacaomedica.com.br/>>. Acesso em: 13 abr. 2017.
- [11] SANTACASABH. SANTACASABH. *SANTACASABH*, 2017. Disponível em: <<http://www.santacasabh.org.br/>>. Acesso em: 13 abr. 2017.
- [12] BERCHT, M. *Em direção a agentes pedagógicos com dimensões afetivas*. [S.l.]: [s.n.], 2001.
- [13] VICCARI, R. M.; GIRAFFA, L. M. F. D. S. T. I. I. D. B. Fundamentos de Sistemas] Tutores Inteligentes. *Sociedades Artificiais*, São Paulo, 2003.
- [14] RUSSELL, S. J.; NORVIG, P. *Inteligência artificial*. Rio de Janeiro: Elsevier, 2004.
- [15] RUSSELL, S. J.; NORVIG, P. *Artificial Intelligence: a modern approach*. New Jersey:] PRENTICE HALL, 1995.
- [16] COPPIN, B. *Inteligência artificial*. Rio de Janeiro: LTC, 2010.
- [17] WOOLDRIDGE, M. *An Introduction to MultiAgent Systems*. 2. ed. [S.l.]: John Wiley &] Sons, 2009.
- [18] BRENNER, W.; WITTIG, H.; ZARNEKOW, R. *Intelligent Software Agents*:

-] Foundations and Applications. *Springer-Verlag*, New York, 1998.
- [19 PATTIE, M.; ROBYN, K. Learning: Interface Agents. *MIT Media Laboratory*,
] Cambridge, p. 459 - 465, 1993.
- [20 RUSSELL, S. J. *Inteligência Artificial*. São Paulo: CAMPUS, 2013.
]
- [21 UNESCO. REPRESENTAÇÃO DA UNESCO NO BRASIL. *REPRESENTAÇÃO DA*
] *UNESCO NO BRASIL*, 2015. Disponível em:<. Acesso em: 23 out. 2015.
- [22 WENGER, E. Artificial Intelligence and Tutoring Systems: Computational and Cognitive
] Approaches to the Communications of Knowledge. *Los Altos, CA*.
- [23 FREEMAN, R. What is an Intelligent Tutoring System? *Published in Intelligenge*, v.
] 11(3), p. 15-16, 2000.
- [24 GREENO, J. G., COLLINS, A., BERANEK, B., & RESNICK, L. B.. Cognition and
] Learning. *Handbook of educational psychology*, p. 1-51, 1994.
- [25 MITCHELL, P. D. . G. P. D. Modelling Techniques for Tutoring Systems. *Computers and*
] *Education*, v. 20, n. 1, p. 56-61, 1993.
- [26 VICCARI, R. U. D. C. 1. (. D. D. *Um Tutor Inteligente para a programação em Lógica*.
] 1990. (Tese de Doutorado).
- [27 OREY, M. A. . N. W. A. Development principles for intelligent tutoring systems:
] integrating cognitive theory into the development of computer-based instruction.
Educational technology Research and Development, v. 1, n. 41, p. 59-72, 1993.
- [28 PÁDUA, D. W. *Engenharia de Software*. 3. ed. Rio de Janeiro: LTC, 2013.
]
- [29 LYRA, F.; SANTOS, N. *Agentes de Software no Monitoramento de Alunos em Educação*
] *a Distância*. 2014.
- [30 KAUTZMANN, T. R.; JAQUES, P. *Um modelo de treinamento adaptativo da habilidade*
] *metacognitiva de monitoramento do conhecimento*. 2015. 10p.
- [31 MEDEIROS, J. B. *A prática de fichamento, resumos, resenhas*. 11. ed. São Paulo: Atlas,
] 2013.
- [32 WAZLAWICK, R. S. *Metodologia de pesquisa para ciência da computação*. Rio de
] Janeiro: Elsevier, 2009.
- [33 SOMERVILE, I. *Engenharia de Software*. 9. ed. São Paulo: Pearson, 2011.
]
- [34 SILVA, M. S. *Web Design Responsivo*. [S.l.]: novatec, 2014.
]
- [35 CHEE, B. J. S. *Computação em Nuvem*. [S.l.]: M. Books, 2013.
]

APÊNDICE A - TERMO DE CONSENTIMENTO

TERMO DE CONSENTIMENTO

Eu _____ declaro ter sido informado e concordo em participar de um teste de avaliação do software intitulado MedicalGame, em participação voluntária, bem como a liberação das informações geradas a partir deste software, para fins acadêmicos e de pesquisa. Este trabalho está sendo desenvolvido pelo aluno do Mestrado em Computação Aplicada da Universidade de Passo Fundo, André Luís Stefanello, orientado pelo Prof. Dr Roberto dos Santos Rabello e coorientado pelo professor Dr. Cassiano Mateus Forcelini.

Os dados e resultados individuais desta pesquisa estarão sempre sob sigilo ético, não sendo mencionados os nomes dos participantes em nenhuma apresentação oral ou trabalho escrito, que venha a ser publicado.

A participação nesta pesquisa não oferece risco ou prejuízo à pessoa participante. Se no decorrer da pesquisa o(a) participante resolver não mais continuar terá toda a liberdade de o fazer, sem que isso lhe acarrete qualquer prejuízo.

Assinatura



APÊNDICE B - PÁGINA INSTITUCIONAL

A página institucional do projeto foi desenvolvida e colocada no ar em meados do mês de março de 2016, com o intuito de divulgar e descrever o trabalho que vinha sendo desenvolvido, esta página foi criada utilizando-se os softwares citados na seção 4.3.

Esta foi a primeira implementação desenvolvida, baseada nos softwares discutidos e selecionados durante os estudos anteriores. A mesma pode ser visualizada na Figura 13, tendo como premissa um layout responsivo⁹, baseado no framework Bootstrap, implementando, desta forma, o carregamento do conteúdo da aplicação em qualquer tipo de dispositivo. Buscou-se, assim, alcançar a característica de responsividade, pela grande utilização de dispositivos com tamanho reduzido, possibilitando o acesso a esta página ou aplicação por qualquer um deles.

Na página institucional, que está acessível no endereço: <http://MedicalGame.com.br>, estão dispostos alguns itens, que fazem menção e descrevem o trabalho e sua equipe, estes itens de *menu* podem ser elencados na seguinte ordem: Bem Vindos; Quem Somos; O que fazemos; Entre em contato, podem ser visualizados na Figura 46.



Figura 46: Página de Apresentação do Sistema MedicalGame. Fonte: Elaborado pelo autor.

⁹ Design Responsivo é uma técnica de estruturação HTML e CSS, em que o site se adapta ao browser do usuário sem precisar definir diversas folhas de estilos para cada resolução. SILVA [34].

**APÊNDICE C - DOCUMENTO DE REQUISITO DE SOFTWARE
– PÁGINA ADMINISTRATIVA MEDICAL GAME E SISTEMA
MEDICALGAME.**

<MedicalGame>

**<Medical Game>
Documento de Levantamento
de Requisitos**

Versão:<Versão 1.0>

Data:<14/04/2017>

Histórico de revisões do modelo

Versão (XX.YY)	Data (DD/MMM/YY YY)	Autor	Descrição	Localização
00.01	15/04/2017	André Luís Stefanello	Versão inicial	
00.02	25/04/2017	André Luís Stefanello	Formatação do doc. e revisão para fechar uma versão.	
00.03	20/04/2017	André Luís Stefanello	Mudanças menores p/finalização do documento	
01.00	05/05/2017	André Luís Stefanello	Formato final	
01.01	05/06/2017	André Luís Stefanello	Versão revisada	

Aprovadores

Nome	Função
André Luís Stefanello	Gerente de Projeto

Visão Geral do Documento

INTRODUÇÃO.....	102
PROPÓSITO	102
PÚBLICO ALVO.....	102
ESCOPO	102
VISÃO GERAL DO PRODUTO.....	102
DESCRIÇÃO DOS USUÁRIOS	102
PREMISSAS E RESTRIÇÕES	103
REQUISITOS FUNCIONAIS (CASOS DE USO)	103
<ATOR ADMINISTRADOR>	103
1.1.1. <RF_ADM001><MANTER ACESSAR SISTEMA>.....	103
1.1.2. <RF_ADM002><MANTER CATEGORIA DE USUÁRIO>.....	104
1.1.3. <RF_ADM003><MANTER INSTITUIÇÃO>	107
1.1.4. <RF_ADM004><MANTER ÁREAS DAS QUESTÕES>	109
1.1.5. <RF_ADM005><MANTER SUBÁREAS DAS QUESTÕES>	111
1.1.6. <RF_ADM006>< MANTER NÍVEL DIFICULDADE QUESTÃO>	114
1.1.7. <RF_ADM007>< MANTER QUESTÃO>	116
1.1.8. <RF_ADM008><MANTER DICAS DE NÍVEL DE DIFICULDADE DE QUESTÃO>.....	119
1.1.9. <RF_ADM009><MANTER QUANTIDADE DE QUESTÕES RESPONDIDAS NO NÍVEL>	121
1.1.10. <RF_ADM010><MANTER QUANTIDADE DE QUESTÕES DO MÍNIMO DE ACERTOS NO NÍVEL>	124
1.1.11. <RF_ADM010><MANTER QUANTIDADE DE QUESTÕES RESPONDIDAS NO NIVELAMENTO>	126
1.1.12. <RF_ADM0011><MANTER A QUANTIDADE DE QUESTÕES DO DUELO>	128
<ATOR ALUNO>.....	131
1.1.13. <RF_USU001>< MANTER USUÁRIO >	131
1.1.14. <RF_USU002>< RESPONDER QUESTÃO >.....	133
REQUISITOS NÃO FUNCIONAIS	135
<RNF001><REQUISITO NÃO FUNCIONAL DESEMPENHO>.....	135
<RNF002><REQUISITO NÃO FUNCIONAL SEGURANÇA>	135
<RNF003><REQUISITO NÃO FUNCIONAL USABILIDADE>.....	135
<RNF004><REQUISITO NÃO FUNCIONAL COMPATIBILIDADE>.....	136
<RNF005><REQUISITO NÃO FUNCIONAL PADRÃO>	136

INTRODUÇÃO

Propósito

Este documento especifica os requisitos dos sistemas a serem desenvolvidos por André Luís Stefanello, fornecendo aos desenvolvedores que posteriormente venham a trabalhar neste projeto, as informações necessárias para possíveis manutenções do mesmo, assim como para a realização dos testes e homologação do sistema.

Público Alvo

Este documento se destina aos arquitetos de *software*, engenheiros de *software* e testadores, bem como desenvolvedores.

Escopo

Este documento realiza a elicitação de requisitos de um sistema para a preparação de alunos de medicina para a prova de Residência Médica, com a utilização de agentes de *software*.

VISÃO GERAL DO PRODUTO

O sistema MedicalGame, tem por objetivo inicial, proporcionar a estudantes dos cursos de medicina, uma ferramenta que os ajude quanto à preparação para a prova de Residência Médica, como citado anteriormente, o público alvo inicialmente será, alunos dos cursos de medicina, a necessidade de implantação deste sistema surgiu, pela demanda corrente de novas formas e métodos de aprendizagem que vem crescendo dia a dia, os impactos do sistema serão a priori positivos, pois os alunos ou candidatos à prova de Residência Médica poderão acessar e utilizar o sistema em qualquer lugar onde haja acesso a *internet*. A grande vantagem além do acesso, será a forma de acesso, ou seja, a aplicação poderá ser acessada de qualquer dispositivo, que seja dotado de um navegador.

Descrição dos usuários

O sistema em seu entorno terá ou envolverá dois tipos de usuário, os administradores do sistema, que farão toda a parte de configuração e parametrização, bem como os usuários finais, neste caso os alunos interessados em estudar ou treinar para as provas de Residência Médica.

PREMISSAS E RESTRIÇÕES

Os casos de uso foram divididos em blocos, para auxiliar no entendimento dos mesmos, onde foram utilizadas as seguintes padronizações:

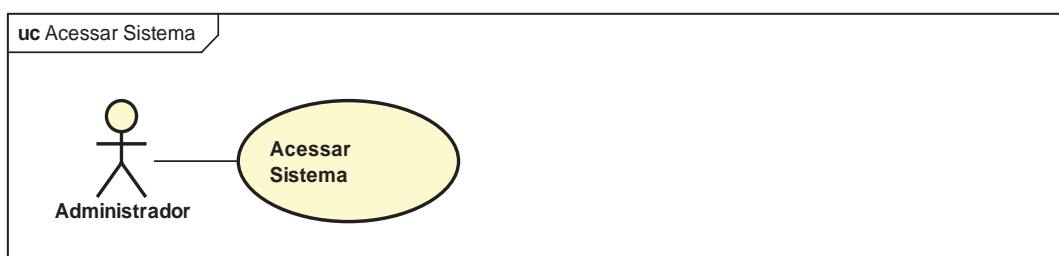
- RF_ADM000 – Requisito funcional do ator administrador, incrementado com um número sequencial de três dígitos.
- RF_USU000 – Requisito funcional do ator usuário, incrementado com um número sequencial de três dígitos.
- RNF000 – Requisitos não funcionais dos sistemas, incrementados com um número sequencial de três dígitos.

REQUISITOS FUNCIONAIS (CASOS DE USO)

Os requisitos aqui descritos serão divididos em duas categorias diferentes, ou seja, uma categoria para o sistema de gerenciamento ou administração da aplicação e uma categoria destinada a parte do aluno, onde o mesmo irá trabalhar efetivamente, as nomenclaturas seguirão a padronização RF_ADMXXX e RF_APLIXXX.

<ATOR ADMINISTRADOR>

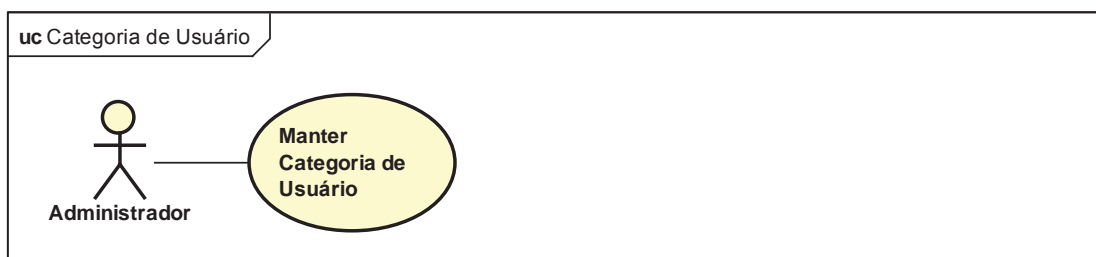
1.1.1. <RF_ADM001><Manter Acessar Sistema>



Manter Acessar Sistema	
Acessar Sistema	
Item	Descrição
Caso de Uso	ACESSAR SISTEMA
Resumo	O usuário deverá entrar com seus dados: <i>login</i> e senha. O Sistema deverá permitir acesso ao conteúdo da aplicação se somente se os dados estiverem corretos.
Ator	Administrador.
Pré-condições	O administrador deve estar previamente cadastrado.
Pós-Condições	O administrador tem acesso à aplicação
Fluxo Principal	<ol style="list-style-type: none"> 1. O administrador solicita autenticação na aplicação. 2. O administrador requisita a autenticação de seu

		usuário. 3. O administrador informa os dados solicitados para a autenticação. 4. O administrador confere os dados e confirma a autenticação. 5. O sistema cadastra a autenticação no banco de dados e o caso de uso termina.
	Fluxo Alternativo	Fluxo Alternativo (3): Erro no Lançamento a. O administrador detecta que lançou uma informação errada. b. O administrador corrige a informação que foi lançada erroneamente. c. O sistema aceita a correção e o caso de uso continua a partir do passo 3.
	Fluxo de Exceção	Fluxo de Exceção (5): Dados obrigatórios para a autenticação com formato incorretos. a) Se o administrador não informar algum dado obrigatório, ou fornecer dados inválidos: o sistema reporta o fato e o caso de uso retorna ao passo 3.
Alterar Acessar Sistema		
	Item	Descrição
	Caso de Uso	ALTERAR Acessar Sistema (NÃO SE APLICA)
Excluir Acessar Sistema		
	Item	Descrição
	Caso de Uso	EXCULIR Acessar Sistema (NÃO SE APLICA)

1.1.2. <RF_ADM002><Manter Categoria de Usuário>



Manter Categoria de Usuário	
Cadastrar Categoria de Usuário	
Item	Descrição
Caso de Uso	CADASTRAR CATEGORIA DE USUÁRIO
Resumo	O administrador do sistema efetiva o cadastro de novas categorias de usuários.
Ator	Administrador.
Pré-condições	O administrador deve estar autenticado no sistema.
Pós-Condições	Uma nova categoria de usuário deve ser cadastrada no sistema.
Fluxo Principal	1. O administrador solicita o cadastro de uma nova categoria de usuário.

		<ol style="list-style-type: none"> 2. O sistema requisita o cadastro de uma nova categoria de usuário. 3. O administrador fornece as informações básicas para o cadastro de um novo usuário. 4. O administrador confere os dados e confirma o cadastro. 5. O sistema cadastra a nova categoria de usuário e o caso de uso termina.
	Fluxo Alternativo	<p>Fluxo Alternativo (4): Erro no Lançamento</p> <ol style="list-style-type: none"> a. O administrador detecta que lançou uma informação errada da categoria de usuário. b. O administrador corrige a informação que foi lançada erroneamente da categoria de usuário c. O sistema aceita a correção e o caso de uso continua a partir do passo 5.
	Fluxo de Exceção	<p>Fluxo de Exceção (3): Dados obrigatórios ao cadastro de categoria do usuário em branco ou com formatos errados.</p> <ol style="list-style-type: none"> a) Se o administrador não informar algum dado obrigatório, ou fornecer dados inválidos: o sistema reporta o fato e o caso de uso retorna ao passo 2.

Alterar Categoria de Usuário

Item	Descrição
Caso de Uso	ALTERAR CATEGORIA DE USUÁRIO
Resumo	O administrador do sistema altera o cadastro da categoria de usuários
Ator	Administrador
Pré-condições	<ol style="list-style-type: none"> 1. O administrador deve estar autenticado no sistema. 2. A categoria de usuário deve existir no banco de dados.
Pós-Condições	A categoria de usuário deve ser alterada
Fluxo Principal	<ol style="list-style-type: none"> 1. O administrador solicita a alteração da categoria de usuário. 2. O administrador fornece as informações de alteração da categoria do usuário. 3. O administrador confere os dados e confirma a alteração do cadastro do usuário. 4. O sistema altera o cadastro do usuário e o caso de uso termina.
Fluxo Alternativo	<p>Fluxo Alternativo (3): Erro no Lançamento</p> <ol style="list-style-type: none"> a. O administrador detecta que lançou uma informação errada da categoria de usuário. b. O administrador corrige a informação da categoria de usuário que foi lançada erroneamente. c. O sistema aceita a correção e o caso de uso continua a partir do passo 3.
Fluxo de Exceção	<p>Fluxo de Exceção (4): Dados obrigatórios ao cadastro de categoria do usuário em branco ou com formatos errados</p> <ol style="list-style-type: none"> a) Se o administrador não informar algum dado

		obrigatório, ou fornecer dados inválidos: o sistema reporta o fato e o caso de uso retorna ao passo 2.
Excluir Categoria de Usuário		
Item	Descrição	
Caso de Uso	EXCLIR CATEGORIA DE USUÁRIO	
Resumo	O administrador do sistema efetiva a exclusão de uma categoria de usuário.	
Ator	Administrador	
Pré-condições	<ol style="list-style-type: none"> 1. O administrador deve estar autenticado no sistema. 2. A categoria de usuário deve existir no banco de dados. 	
Pós-Condições	A categoria de usuários deve ser excluída.	
Fluxo Principal	<ol style="list-style-type: none"> 1. O administrador solicita a exclusão de uma categoria de usuário. 2. O sistema exibe os dados da categoria de usuário e requisita a exclusão do mesmo. 3. O administrador procede com exclusão da categoria de usuário. 4. O administrador confere os dados e confirma a exclusão do cadastro da categoria de usuários. 5. O sistema efetiva a exclusão e o caso de uso termina. 	
Fluxo Alternativo	Fluxo Alternativo (5): Exclusão Abortada <ol style="list-style-type: none"> a. Se o sistema detecta que a categoria de usuário tem vínculos efetivos: o sistema reporta o fato e o caso de uso retorna ao passo 1. 	
Fluxo de Exceção	Não se aplica.	
Consultar Categoria de Usuário		
Item	Descrição	
Caso de Uso	CONSULTAR CATEGORIA DE USUÁRIO	
Resumo	O administrador do sistema efetiva a consulta de determinada categoria de usuário.	
Ator	Administrador	
Pré-condições	<ol style="list-style-type: none"> 1. O administrador deve estar autenticado no sistema. 2. A categoria de usuário deve estar cadastrada no banco de dados. 	
Pós-Condições	A categoria de usuários deve ter seus dados consultados no sistema.	
Fluxo Principal	<ol style="list-style-type: none"> 1. O administrador solicita a consulta da categoria de usuário. 2. O sistema exibe os dados da categoria de usuário. 3. O administrador visualiza os dados. 4. O caso de uso termina. 	
Fluxo Alternativo	Fluxo Alternativo (2): Categoria de Usuário inexistente	

		a. Se o sistema não encontrar a categoria de usuário, o fato é reportado e o caso de uso retorna ao passo 1.
	Fluxo de Exceção	Não há.

1.1.3. <RF_ADM003><Manter Instituição>



Manter Instituição	
Cadastrar Instituição	
Item	Descrição
Caso de Uso	CADASTRAR INSTITUIÇÃO
Resumo	O administrador do sistema efetiva o cadastro de nova instituição.
Ator	administrador.
Pré-condições	O administrador deve estar autenticado no sistema.
Pós-Condições	Uma nova instituição é cadastrada no sistema.
Fluxo Principal	<ol style="list-style-type: none"> 1. O administrador solicita o cadastro de uma nova instituição. 2. O administrador fornece as informações básicas para o cadastro de uma nova instituição. 3. O administrador confere os dados e confirma o cadastro. 4. O sistema cadastra a instituição e o caso de uso termina.
Fluxo Alternativo	Fluxo Alternativo (3): Erro no Lançamento: <ol style="list-style-type: none"> a. O administrador detecta que lançou uma informação errada para a instituição. b. O administrador corrige a informação da instituição que foi lançada erroneamente. c. O sistema aceita a correção e o caso de uso continua a partir do passo 4.
Fluxo de Exceção	Fluxo de Exceção (2): Dados obrigatórios ao cadastro da instituição em branco ou com formatos incorretos: <ol style="list-style-type: none"> a) Se o administrador não informar algum dado obrigatório, CNPJ, e-mail institucional ou nome da instituição, ou fornecer tipos de dados inválidos: o sistema reporta o fato e o caso de uso retorna ao passo 2.
Alterar Instituição	
Item	Descrição
Caso de Uso	ALTERAR INSTITUIÇÃO

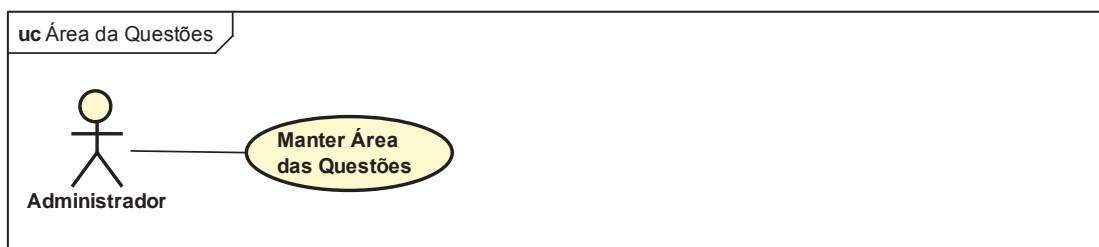
Resumo	O administrador do sistema altera o cadastro da Instituição.
Ator	Administrador.
Pré-condições	<ol style="list-style-type: none"> 1. O administrador deve estar autenticado no sistema. 2. A instituição deve estar cadastrada.
Pós-Condições	A questão deve ser alterada.
Fluxo Principal	<ol style="list-style-type: none"> 1. O administrador solicita a alteração da instituição. 2. O administrador fornece as informações de alteração da instituição. 3. O administrador confere os dados e confirma a alteração do cadastro da instituição. 4. O sistema altera o cadastro da instituição e o caso de uso termina.
Fluxo Alternativo	<p>Fluxo Alternativo (2): Erro no Lançamento</p> <ol style="list-style-type: none"> a. O Administrador detecta que lançou uma informação errada para a instituição. b. O Administrador corrige a informação que foi lançada erroneamente da instituição. c. O sistema aceita a correção e o caso de uso continua a partir do passo 3.
Fluxo de Exceção	<p>Fluxo de Exceção (4): Dados obrigatórios ao cadastro de instituição em branco ou com formatos incorretos.</p> <ol style="list-style-type: none"> a) Se o administrador não informar algum dado obrigatório, ou fornecer dados inválidos: o sistema reporta o fato e o caso de uso retorna ao passo 2.

Excluir Instituição

Item	Descrição
Caso de Uso	EXCLIR INSTITUIÇÃO
Resumo	O administrador do sistema efetiva a exclusão de uma instituição.
Ator	Administrador.
Pré-condições	<ol style="list-style-type: none"> 1. O administrador deve estar autenticado no sistema. 2. A instituição deve estar cadastrada.
Pós-Condições	A instituição deve ser excluída.
Fluxo Principal	<ol style="list-style-type: none"> 1. O administrador solicita a exclusão de uma instituição. 2. O sistema exibe os dados da questão e requisita a exclusão da mesma. 3. O administrador procede com exclusão da instituição. 4. O administrador confere os dados e confirma a exclusão da instituição. 5. O sistema efetiva a exclusão e o caso de uso termina.
Fluxo Alternativo	<p>Fluxo Alternativo (5): Exclusão Abortada</p> <ol style="list-style-type: none"> a. Se o sistema detecta que a instituição tem vínculos

		efetivos: o sistema reporta o fato e o caso de uso retorna ao passo 1.
	Fluxo de Exceção	Não se aplica.
Consultar Instituição		
	Item	Descrição
	Caso de Uso	CONSULTAR INSTITUIÇÃO
	Resumo	O administrador do sistema efetiva a consulta de determinada instituição.
	Ator	Administrador
	Pré-condições	1. O administrador deve estar autenticado no sistema. 2. A instituição deve estar cadastrada.
	Pós-Condições	A instituição deve ter seus dados consultados no sistema.
	Fluxo Principal	1. O administrador solicita a consulta da instituição. 2. O sistema exibe os dados da instituição. 3. O administrador visualiza os dados. 4. O caso de uso termina.
	Fluxo Alternativo	Fluxo Alternativo (2): A instituição não está cadastrada. a. Se o sistema não encontrar a instituição, o fato é reportado e o caso de uso retorna ao passo 1.
	Fluxo de Exceção	Não se aplica.

1.1.4. <RF_ADM004><Manter Áreas das Questões>



Manter Áreas das Questões		
Cadastrar Áreas das Questões		
	Item	Descrição
	Caso de Uso	CADASTRAR ÁREAS DAS QUESTÕES
	Resumo	O administrador do sistema efetiva o cadastro de novas áreas para a vinculação das questões.
	Ator	Administrador
	Pré-condições	O administrador deve estar autenticado no sistema.
	Pós-Condições	Uma nova área de questões é cadastrada no sistema
	Fluxo Principal	1. O administrador solicita o cadastro de uma nova área. 2. O administrador fornece as informações básicas para o cadastro de uma nova área. 3. O administrador confere os dados e confirma o cadastro.

		4. O sistema cadastra a nova área e o caso de uso termina.
	Fluxo Alternativo	Fluxo Alternativo (2): Erro no Lançamento. a. O administrador detecta que lançou uma informação ou área, de forma errada. b. O administrador corrige a informação que foi lançada erroneamente na área. c. O sistema aceita a correção e o caso de uso continua a partir do passo 4.
	Fluxo de Exceção	Fluxo de Exceção (2): Dados obrigatórios ao cadastro da área, em branco ou com formatos errados. a) Se o administrador não informar algum dado obrigatório, o sistema reporta o fato e o caso de uso retorna ao passo 2.
Alterar Áreas das Questões		
	Item	Descrição
	Caso de Uso	ALTERAR ÁREAS DAS QUESTÕES
	Resumo	O administrador do sistema altera o cadastro das áreas que as questões podem estar vinculadas.
	Ator	Administrador.
	Pré-condições	O administrador deve estar autenticado no sistema. A área deve estar cadastrada no banco de dados do sistema.
	Pós-Condições	A área deve ser alterada.
	Fluxo Principal	1. O administrador solicita a alteração da área. 2. O administrador fornece as informações de alteração da área. 3. O administrador confere os dados e confirma a alteração do cadastro da área. 4. O sistema altera o cadastro da área e o caso de uso termina.
	Fluxo Alternativo	Fluxo Alternativo (2): Erro no lançamento. a. O administrador detecta que lançou uma informação errada, para a área. b. O administrador corrige a informação que foi lançada erroneamente para a área. c. O sistema aceita a correção e o caso de uso continua a partir do passo 2.
	Fluxo de Exceção	Fluxo de Exceção (4): Dados obrigatórios ao cadastro de área em branco ou com formatos errados. a) Se o administrador não informar algum dado obrigatório, ou fornecer, dados inválidos: o sistema reporta o fato e o caso de uso retorna ao passo 2.
Excluir Áreas das Questões		
	Item	Descrição
	Caso de Uso	EXCLIR ÁREAS DAS QUESTÕES
	Resumo	O administrador do sistema efetiva a exclusão de uma área.
	Ator	Administrador.
	Pré-condições	O administrador deve estar autenticado no sistema. A área deve estar cadastrada no banco de dados.

Pós-Condições	A área deve ser excluída.
Fluxo Principal	<ol style="list-style-type: none"> 1. O administrador solicita a exclusão de uma. 2. O sistema exibe os dados da área, e requisita a exclusão da mesma. 3. O administrador procede com exclusão da área. 4. O administrador confere os dados e confirma a exclusão da área. 5. O sistema efetiva a exclusão e o caso de uso termina.
Fluxo Alternativo	<p>Fluxo Alternativo (5): Exclusão abortada</p> <ol style="list-style-type: none"> a. Se o sistema detecta que a área tem vínculos efetivos: o sistema reporta o fato e o caso de uso retorna ao passo 1.
Fluxo de Exceção	Não se aplica.

Consultar Áreas das Questões

Item	Descrição
Caso de Uso	CONSULTAR ÁREAS DAS QUESTÕES
Resumo	O administrador do sistema efetiva a consulta de determinada área.
Ator	Administrador.
Pré-condições	O administrador deve estar autenticado no sistema. A área deve estar cadastrada no banco de dados.
Pós-Condições	A área deve ter seus dados consultados no sistema
Fluxo Principal	<ol style="list-style-type: none"> 1. O administrador solicita a consulta da área. 2. O sistema exibe os dados da área. 3. O administrador visualiza os dados. 4. O caso de uso termina.
Fluxo Alternativo	<p>Fluxo Alternativo (2): A área inexistente.</p> <ol style="list-style-type: none"> a. Se o sistema não encontrar a área, o fato é reportado e o caso de uso retorna ao passo 1.
Fluxo de Exceção	Não se aplica

1.1.5. <RF_ADM005><Manter SubÁreas das Questões>



Manter Sub Áreas das Questões

Cadastrar Sub Áreas das Questões

Item	Descrição
Caso de Uso	CADASTRAR SUBÁREAS DAS QUESTÕES
Resumo	O administrador do sistema efetiva o cadastro de novas subáreas.
Ator	Administrador

Pré-condições	O administrador deve estar autenticado no sistema. Uma área deve estar cadastrada no sistema.
Pós-Condições	Uma nova subárea é cadastrada no sistema.
Fluxo Principal	<ol style="list-style-type: none"> 1. O administrador solicita o cadastro de uma nova subárea. 2. O administrador fornece as informações básicas para o cadastro de uma nova subárea. 3. O administrador confere os dados e confirma o cadastro. 4. O sistema cadastra a nova subárea e o caso de uso termina.
Fluxo Alternativo	<p>Fluxo Alternativo (2): Erro no Lançamento</p> <ol style="list-style-type: none"> a. O Administrador detecta que lançou de forma equivocada uma informação ou subárea. b. O Administrador corrige a informação que foi lançada erroneamente na subárea. c. O sistema aceita a correção e o caso de uso continua a partir do passo 3.
Fluxo de Exceção	<p>Fluxo de Exceção (2): Dados obrigatórios ao cadastro da subárea da questão, em branco ou com formatos errados.</p> <ol style="list-style-type: none"> a) Se o administrador não informar algum dado obrigatório, o sistema reporta o fato e o caso de uso retorna ao passo 2.

Alterar Sub Áreas das Questões

Item	Descrição
Caso de Uso	ALTERAR SUBÁREAS DAS QUESTÕES
Resumo	O administrador do sistema altera o cadastro da subárea.
Ator	Administrador
Pré-condições	O administrador deve estar autenticado no sistema. A subárea deve estar cadastrada no banco de dados do sistema.
Pós-Condições	A subárea deve ser alterada
Fluxo Principal	<ol style="list-style-type: none"> 1. O administrador solicita a alteração da subárea. 2. O administrador fornece as informações de alteração da subárea. 3. O administrador confere os dados e confirma a alteração do cadastro da subárea. 4. O sistema altera o cadastro da subárea e o caso de uso termina.
Fluxo Alternativo	<p>Fluxo Alternativo (2): Erro no Lançamento.</p> <ol style="list-style-type: none"> a. O administrador detecta o lançamento de uma informação errada para a subárea. b. O administrador corrige a informação da subárea que foi lançada erroneamente. c. O sistema aceita a correção e o caso de uso continua a partir do passo 2.
Fluxo de Exceção	<p>Fluxo de Exceção (4): Dados obrigatórios ao cadastro de área em branco ou com formatos errados.</p> <ol style="list-style-type: none"> a) Se o administrador não informar algum dado obrigatório, ou fornecer dados inválidos: o sistema

		reporta o fato e o caso de uso retorna ao passo 2.
Excluir Sub Áreas das Questões		
Item	Descrição	
Caso de Uso	EXCLIR SUBÁREAS DAS QUESTÕES	
Resumo	O administrador do sistema efetiva a exclusão de uma subárea.	
Ator	Administrador	
Pré-condições	O administrador deve estar autenticado no sistema. A subárea deve existir no banco de dados do sistema.	
Pós-Condições	A subárea da questão deve ser excluída.	
Fluxo Principal	<ol style="list-style-type: none"> 1. O administrador solicita a exclusão de uma subárea. 2. O sistema exibe os dados da subárea e requisita a exclusão da mesma. 3. O administrador procede com exclusão da subárea. 4. O administrador confere os dados e confirma a exclusão da subárea. 5. O sistema efetiva a exclusão e o caso de uso termina. 	
Fluxo Alternativo	Fluxo Alternativo (5): Exclusão Abortada. <ol style="list-style-type: none"> a. Se o sistema detecta que a subárea tem vínculos efetivos: o sistema reporta o fato e o caso de uso retorna ao passo 1. 	
Fluxo de Exceção	Não se aplica.	
Consultar Sub Áreas das Questões		
Item	Descrição	
Caso de Uso	CONSULTAR SUBÁREAS DAS QUESTÕES	
Resumo	O administrador do sistema efetiva a consulta de determinada subárea.	
Ator	Administrador.	
Pré-condições	O administrador deve estar autenticado no sistema. A subárea deve estar cadastrada no banco de dados do sistema.	
Pós-Condições	A subárea deve ter seus dados consultados no sistema.	
Fluxo Principal	<ol style="list-style-type: none"> 1. O administrador solicita a consulta da subárea. 2. O sistema exibe os dados da subárea. 3. O administrador visualiza os dados. 4. O caso de uso termina. 	
Fluxo Alternativo	Fluxo Alternativo (2): A subárea inexistente. <ol style="list-style-type: none"> a. Se o sistema não encontrar a subárea, o fato é reportado e o caso de uso retorna ao passo 1. 	
Fluxo de Exceção	Não se aplica.	

1.1.6. <RF_ADM006>< Manter Nível Dificuldade Questão>

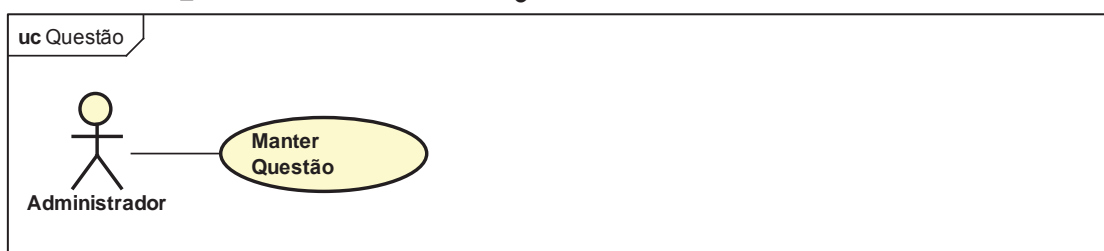


Manter Nível de Dificuldade de Questão	
Cadastrar Nível de Dificuldade de Questão	
Item	Descrição
Caso de Uso	CADASTRAR NÍVEL DE DIFICULDADE DE QUESTÃO
Resumo	O administrador do sistema efetiva o cadastro de novo nível de dificuldade de questões.
Ator	Administrador.
Pré-condições	O administrador deve estar autenticado no sistema.
Pós-Condições	Um novo nível de dificuldade é cadastrado no sistema.
Fluxo Principal	<ol style="list-style-type: none"> 1. O administrador solicita o cadastro de um novo nível de dificuldade. 2. O sistema requisita o cadastro de um novo nível de dificuldade. 3. O administrador fornece as informações básicas para o cadastro de um novo nível de dificuldade. 4. O administrador confere os dados e confirma o cadastro. 5. O sistema cadastra o nível de dificuldade e o caso de uso termina.
Fluxo Alternativo	Fluxo Alternativo (4): Erro no Lançamento <ol style="list-style-type: none"> a. O administrador detecta o lançamento de dados errados no nível de dificuldade. b. O administrador corrige a informação que foi lançada erroneamente. c. O sistema aceita a correção e o caso de uso continua a partir do passo 4.
Fluxo de Exceção	Fluxo de Exceção (2): Dados obrigatórios ao cadastro do nível de dificuldade em branco ou com formatos errados. <ol style="list-style-type: none"> a) Se o administrador não informar algum dado obrigatório, ou fornecer dados inválidos: o sistema reporta o fato e o caso de uso retorna ao passo 2.
Alterar Nível de Dificuldade Questão	
Item	Descrição
Caso de Uso	ALTERAR NÍVEL DE DIFICULDADE QUESTÃO
Resumo	O administrador do sistema altera o cadastro nível de dificuldade.
Ator	Administrador.
Pré-condições	O administrador deve estar autenticado no sistema. O nível de dificuldade deve estar cadastrado no banco de dados.

	Pós-Condições	O nível de dificuldade deve ser alterado.
	Fluxo Principal	<ol style="list-style-type: none"> 1. O administrador solicita a alteração do nível de dificuldade. 2. O administrador fornece as informações de alteração de nível de dificuldade. 3. O administrador confere os dados e confirma a alteração de nível de dificuldade. 4. O sistema altera o cadastro de nível de dificuldade e o caso de uso termina.
	Fluxo Alternativo	<p>Fluxo Alternativo (2): Erro no Lançamento</p> <ol style="list-style-type: none"> a. O administrador detecta que lançou uma informação errada. b. O administrador corrige a informação que foi lançada erroneamente. c. O sistema aceita a correção e o caso de uso continua a partir do passo 2. <p>Fluxo Alternativo (4): Alteração</p> <ol style="list-style-type: none"> a) Efetivada - Se o sistema detecta que o módulo de questão tem vínculos efetivos: o sistema reporta o fato, solicitando ao administrador uma confirmação, se essa for positiva, o caso de uso é finalizado. b) Abortada - Se o sistema detecta que o módulo de questão tem vínculos efetivos: o sistema reporta o fato, solicitando ao administrador uma confirmação, se essa for negativa, o caso de uso continua a partir do passo 1.
	Fluxo de Exceção	<p>Fluxo de Exceção (3): Dados obrigatórios ao cadastro da dificuldade da questão, em branco ou com formatos errados.</p> <ol style="list-style-type: none"> a) Se o administrador não informar algum dado obrigatório, ou fornecer dados inválidos: o sistema reporta o fato e o caso de uso retorna ao passo 2.
Excluir Nível de Dificuldade Questão		
	Item	Descrição
	Caso de Uso	EXCLUIR NÍVEL DE DIFICULDADE DE QUESTÃO
	Resumo	O administrador do sistema efetiva a exclusão de um nível de dificuldade.
	Ator	Administrador.
	Pré-condições	O administrador deve estar autenticado no sistema. Um nível de dificuldade deve estar cadastrado no banco de dados do sistema.
	Pós-Condições	O nível de dificuldade deve ser excluído.
	Fluxo Principal	<ol style="list-style-type: none"> 1. O administrador solicita a exclusão do nível de dificuldade. 2. O sistema exibe os dados de nível de dificuldade e requisita a exclusão do mesmo. 3. O administrador procede com exclusão do nível de dificuldade. 4. O administrador confere os dados e confirma a exclusão do nível de dificuldade. 5. O sistema efetiva a exclusão e o caso de uso termina.

	Fluxo Alternativo	Fluxo Alternativo (5): Exclusão Abortada a. Se o sistema detecta que a o nível de dificuldade tem vínculos efetivos: o sistema reporta o fato e o caso de uso retorna ao passo 1.
	Fluxo de Exceção	Não se aplica.
Consultar Nível de Dificuldade Questão		
	Item	Descrição
	Caso de Uso	CONSULTAR NÍVEL DE DIFICULDADE DE QUESTÃO
	Resumo	O administrador do sistema efetiva a consulta de determinado nível de dificuldade.
	Ator	Administrador.
	Pré-condições	O administrador deve estar autenticado no sistema. Um nível de dificuldade deve estar cadastrado no sistema.
	Pós-Condições	O nível de dificuldade deve ter seus dados consultados no sistema.
	Fluxo Principal	1. O administrador solicita a consulta do nível de dificuldade. 2. O sistema exibe os dados de nível de dificuldade. 3. O administrador visualiza os dados. 4. O caso de uso termina.
	Fluxo Alternativo	Fluxo Alternativo (2): O nível de dificuldade de questão não existe. a. Se o sistema não encontrar o nível de dificuldade, o fato é reportado e o caso de uso retorna ao passo 1.
	Fluxo de Exceção	Não se aplica

1.1.7. <RF_ADM007>< Manter Questão>



Manter Questão		
Cadastrar Questão		
	Item	Descrição
	Caso de Uso	CADASTRAR QUESTÃO
	Resumo	O administrador do sistema efetiva o cadastro de novas questões, vinculadas a seus respectivos módulos.
	Ator	Administrador.
	Pré-condições	O administrador deve estar autenticado no sistema
	Pós-Condições	Uma nova questão é cadastrada no sistema
	Fluxo Principal	1. O administrador solicita o cadastro de uma nova questão.

		<ol style="list-style-type: none"> 2. O sistema exibe as categorias de questões, categorias de dificuldade, os módulos cadastrados, requisita, requisitando desta forma o cadastro de uma nova questão. 3. O administrador seleciona a categoria da questão, o nível de dificuldade e a que módulo ela está vinculada. 4. O administrador fornece as informações básicas para o cadastro de uma nova questão. 5. O administrador confere os dados e confirma o cadastro. 6. O sistema cadastra a questão e o caso de uso termina.
	Fluxo Alternativo	<p>Fluxo Alternativo (5): Erro no Lançamento</p> <ol style="list-style-type: none"> a. O administrador detecta que lançou uma informação ou categoria errada para a questão. b. O administrador corrige a informação que foi lançada erroneamente para com a questão. c. O sistema aceita a correção e o caso de uso continua a partir do passo 5.
	Fluxo de Exceção	<p>Fluxo de Exceção (2): Dados obrigatórios ao cadastro da questão em branco ou com formatos errados</p> <ol style="list-style-type: none"> a) Se o administrador não informar algum dado obrigatório, categorias de questões, categorias de dificuldade, a que módulo, ou fornecer dados inválidos: o sistema reporta o fato e o caso de uso retorna ao passo 2.

Alterar Questão

Item	Descrição
Caso de Uso	ALTERAR QUESTÃO
Resumo	O administrador do sistema altera o cadastro da questão
Ator	Administrador
Pré-condições	O administrador deve estar autenticado no sistema
Pós-Condições	A questão deve ser alterada
Fluxo Principal	<ol style="list-style-type: none"> 1. O administrador solicita a alteração da questão. 2. O administrador fornece as informações de alteração da questão. 3. O administrador confere os dados e confirma a alteração do cadastro da questão. 4. O sistema altera o cadastro do usuário e o caso de uso termina.
Fluxo Alternativo	<p>Fluxo Alternativo (2): Erro no Lançamento</p> <ol style="list-style-type: none"> a. O Administrador detecta que lançou uma informação errada para a questão. b. O Administrador corrige a informação que foi lançada erroneamente da questão. c. O sistema aceita a correção e o caso de uso continua a partir do passo 2.
Fluxo de Exceção	Fluxo de Exceção (4): Dados obrigatórios ao cadastro de questão em branco ou com formatos errados

		a) Se o administrador não informar algum dado obrigatório, ou fornecer dados inválidos: o sistema reporta o fato e o caso de uso retorna ao passo 2.
Excluir Questão		
Item	Descrição	
Caso de Uso	EXCLIR QUESTÃO	
Resumo	O administrador do sistema efetiva a exclusão de uma questão.	
Ator	Administrador.	
Pré-condições	O administrador deve estar autenticado no sistema. A questão deve existir no banco de dados da aplicação.	
Pós-Condições	A questão deve ser excluída.	
Fluxo Principal	<ol style="list-style-type: none"> 1. O administrador solicita a exclusão de uma questão. 2. O sistema exibe os dados da questão e requisita a exclusão da mesma. 3. O administrador procede com exclusão da questão. 4. O administrador confere os dados e confirma a exclusão da questão. 5. O sistema efetiva a exclusão e o caso de uso termina. 	
Fluxo Alternativo	Fluxo Alternativo (5): Exclusão Abortada <ol style="list-style-type: none"> a. Se o sistema detecta que a questão tem vínculos efetivos: o sistema reporta o fato e o caso de uso retorna ao passo 1. 	
Fluxo de Exceção	Não se aplica.	
Consultar Questão		
Item	Descrição	
Caso de Uso	CONSULTAR QUESTÃO	
Resumo	O administrador do sistema efetiva a consulta de determinada questão.	
Ator	Administrador.	
Pré-condições	O administrador deve estar autenticado no sistema.	
Pós-Condições	A questão deve ter seus dados consultados no sistema.	
Fluxo Principal	<ol style="list-style-type: none"> 1. O administrador solicita a consulta da questão. 2. O sistema exibe os dados da questão. 3. O administrador visualiza os dados. 4. O caso de uso termina. 	
Fluxo Alternativo	Fluxo Alternativo (2): A questão inexistente <ol style="list-style-type: none"> a. Se o sistema não encontrar a questão, o fato é reportado e o caso de uso retorna ao passo 1. 	
Fluxo de Exceção	Não se aplica.	

1.1.8. <RF_ADM008><Manter Dicas de Nível de Dificuldade de Questão>



Manter Dicas de Nível de Dificuldade Questão

Cadastrar Dicas de Nível de Dificuldade Questão

Item	Descrição
Caso de Uso	CADASTRAR DICAS DE NÍVEL DE DIFICULDADE
Resumo	O administrador do sistema efetiva o cadastro de nova dica de nível.
Ator	Administrador
Pré-condições	O administrador deve estar autenticado no sistema.
Pós-Condições	Uma nova dica de nível é cadastrada no sistema
Fluxo Principal	<ol style="list-style-type: none"> 1. O administrador solicita o cadastro de uma nova dica de nível. 2. O sistema requisita o cadastro de uma nova dica de nível. 3. O administrador fornece as informações básicas para o cadastro de uma nova dica de nível. 4. O administrador confere os dados e confirma o cadastro. 5. O sistema cadastra a dica de nível e o caso de uso termina.
Fluxo Alternativo	<p>Fluxo Alternativo (4): Erro no Lançamento</p> <ol style="list-style-type: none"> a. O administrador detecta que lançou uma informação errada para a dica de nível. b. O administrador corrige a informação que foi lançada erroneamente da dica de nível. c. O sistema aceita a correção e o caso de uso continua a partir do passo 4.
Fluxo de Exceção	<p>Fluxo de Exceção (2): Dados obrigatórios ao cadastro da dica de nível em branco, ou com formatos errados.</p> <ol style="list-style-type: none"> a) Se o administrador não informar algum dado obrigatório, ou fornecer dados inválidos: o sistema reporta o fato e o caso de uso retorna ao passo 2.

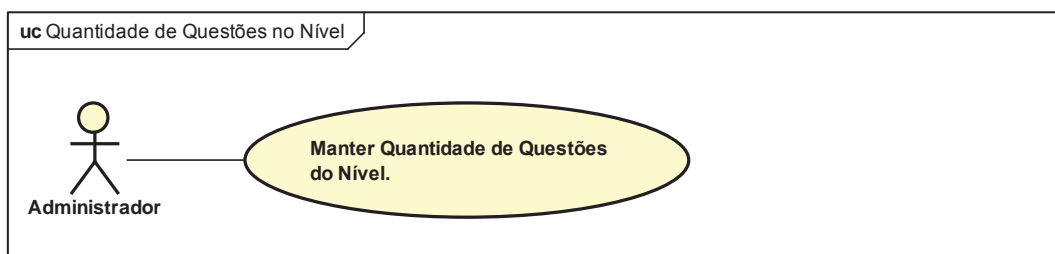
Alterar Dicas de Nível de Dificuldade de Questão

Item	Descrição
Caso de Uso	ALTERAR DICAS DE NÍVEL DE DIFICULDADE DE QUESTÃO
Resumo	O administrador do sistema altera o cadastro de dica de nível.
Ator	Administrador.
Pré-condições	O administrador deve estar autenticado no sistema. A dica de nível deve existir no banco de dados do sistema.

	Pós-Condições	A dica de nível de dificuldade deve ser alterada.
	Fluxo Principal	<ol style="list-style-type: none"> 1. O administrador solicita a alteração da dica de nível. 2. O administrador fornece as informações de alteração da dica de nível. 3. O administrador confere os dados e confirma a alteração da dica de nível. 4. O sistema altera o cadastro da dica de nível e o caso de uso termina.
	Fluxo Alternativo	<p>Fluxo Alternativo (2): Erro no Lançamento.</p> <ol style="list-style-type: none"> a. O Administrador detecta que lançou uma informação errada para a dica de nível. b. O Administrador corrige a informação que foi lançada erroneamente da dica de nível. c. O sistema aceita a correção e o caso de uso continua a partir do passo 2. <p>Fluxo Alternativo (4): Alteração</p> <ol style="list-style-type: none"> a) Efetivada - Se o sistema detecta que o módulo de dicas de nível tem vínculos efetivos: o sistema reporta o fato, solicitando ao administrador uma confirmação, se essa for positiva, o caso de uso é finalizado. b) Abortada - Se o sistema detecta que o módulo de dica de níveis tem vínculos efetivos: o sistema reporta o fato, solicitando ao administrador uma confirmação, se essa for negativa, o caso de uso continua a partir do passo 1.
	Fluxo de Exceção	<p>Fluxo de Exceção (3): Dados obrigatórios ao cadastro da dica de nível de dificuldade, em branco ou com formatos errados.</p> <ol style="list-style-type: none"> a) Se o administrador não informar algum dado obrigatório, ou fornecer dados inválidos: o sistema reporta o fato e o caso de uso retorna ao passo 2.
Excluir Dicas de Nível de Dificuldade Questão		
	Item	Descrição
	Caso de Uso	EXCLUIR DICAS DE NÍVEL DE DIFICULDADE DE QUESTÃO
	Resumo	O administrador do sistema efetiva a exclusão da dica de um nível de dificuldade.
	Ator	Administrador
	Pré-condições	O administrador deve estar autenticado no sistema
	Pós-Condições	A dica de nível de dificuldade deve ser excluída.
	Fluxo Principal	<ol style="list-style-type: none"> 1. O administrador solicita a exclusão da dica de nível de dificuldade. 2. O sistema exibe os dados da dica do nível de dificuldade e requisita a exclusão da mesma. 3. O administrador procede com exclusão da dica de nível de dificuldade. 4. O administrador confere os dados e confirma a exclusão da dica de nível de dificuldade. 5. O sistema efetiva a exclusão e o caso de uso termina.

	Fluxo Alternativo	Fluxo Alternativo (5): Exclusão Abortada a. Se o sistema detecta que a dica de nível de dificuldade tem vínculos efetivos: o sistema reporta o fato e o caso de uso retorna ao passo 1.
	Fluxo de Exceção	Não se aplica
Consultar Dicas de Nível de Dificuldade de Questão		
	Item	Descrição
	Caso de Uso	CONSULTAR DICAS DE NÍVEL DE DIFICULDADE DE QUESTÃO
	Resumo	O administrador do sistema efetiva a consulta de determinada dica de nível de dificuldade.
	Ator	Administrador
	Pré-condições	O administrador deve estar autenticado no sistema.
	Pós-Condições	A dica de nível de dificuldade deve ter seus dados consultados no sistema.
	Fluxo Principal	<ol style="list-style-type: none"> 1. O administrador solicita a consulta da dica de nível de dificuldade. 2. O sistema exibe os dados de dica de nível de dificuldade. 3. O administrador visualiza os dados. 4. O caso de uso termina.
	Fluxo Alternativo	Fluxo Alternativo (2): A dica de nível de dificuldade não existe a. Se o sistema não encontrar a dica de nível de dificuldade, o fato é reportado e o caso de uso retorna ao passo 1.
	Fluxo de Exceção	Não se aplica.

1.1.9. <RF_ADM009><Manter Quantidade de Questões Respondidas no Nível>

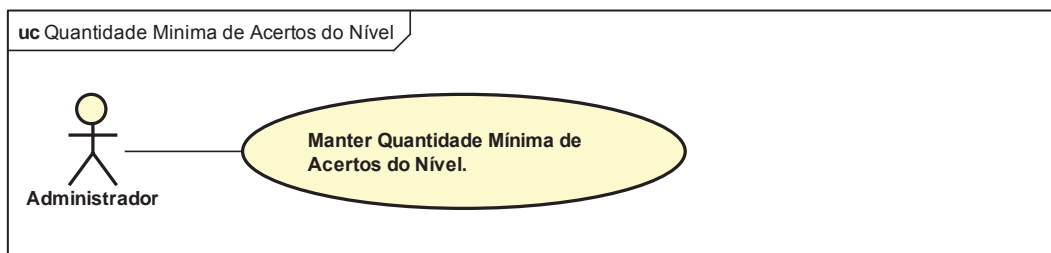


Manter Quantidade de Questões Respondidas no Nível	
Cadastrar a Quantidade de Questões Respondidas no Nível	
Item	Descrição
Caso de Uso	CADASTRAR A QUANTIDADE DE QUESTÕES RESPONDIDAS NO NÍVEL
Resumo	O administrador do sistema efetiva o cadastro da

		quantidade de questões a serem respondidas em um nível de dificuldade.
Ator		Administrador
Pré-condições		O administrador deve estar autenticado no sistema.
Pós-Condições		Uma nova quantidade de questões deve ser respondida no nível de dificuldade.
Fluxo Principal		<ol style="list-style-type: none"> 1. O administrador solicita o cadastro de uma nova quantidade de questões. 2. O sistema requisita o cadastro de uma nova quantidade de questões a serem respondidas naquele nível de dificuldade. 3. O administrador fornece as informações da quantidade de questões a serem respondidas naquele nível de dificuldade. 4. O administrador confere os dados e confirma o cadastro. 5. O sistema cadastra a quantidade de questões e o caso de uso termina.
Fluxo Alternativo		<p>Fluxo Alternativo (4): Erro no Lançamento</p> <ol style="list-style-type: none"> a. O administrador detecta o lançamento de uma informação errada para a quantidade de questões. b. O administrador corrige a informação que foi lançada erroneamente. c. O sistema aceita a correção e o caso de uso continua a partir do passo 4.
Fluxo de Exceção		<p>Fluxo de Exceção (2): Dados obrigatórios ao cadastro da quantidade de questões em branco ou com formatos errados.</p> <ol style="list-style-type: none"> a) Se o administrador não informar algum dado obrigatório, ou fornecer dados inválidos: o sistema reporta o fato e o caso de uso retorna ao passo 2.
Alterar a Quantidade de Questões Respondidas no Nível		
Item		Descrição
Caso de Uso		ALTERAR A QUANTIDADE DE QUESTÕES RESPONDIDAS NO NÍVEL.
Resumo		O administrador do sistema altera o cadastro da quantidade de questões a serem respondidas.
Ator		Administrador
Pré-condições		O administrador deve estar autenticado no sistema.
Pós-Condições		A quantidade de questões a serem respondidas deve ser alterada.
Fluxo Principal		<ol style="list-style-type: none"> 1. O administrador solicita a alteração da quantidade de questões a serem respondidas. 2. O administrador fornece as informações da quantidade de questões a serem respondidas. 3. O administrador confere os dados e confirma a alteração da quantidade de questões a serem respondidas. 4. O sistema altera o cadastro da quantidade de

		questões a serem respondidas e o caso de uso termina.
	Fluxo Alternativo	Fluxo Alternativo (2): Erro no Lançamento. a. O Administrador detecta que lançou uma informação errada para a quantidade de questões a serem respondidas. b. O Administrador corrige a informação que foi lançada erroneamente. c. O sistema aceita a correção e o caso de uso continua a partir do passo 2.
	Fluxo de Exceção	Fluxo de Exceção (3): Dados obrigatórios ao cadastro da quantidade de questões a serem respondidas estão faltando ou em formato incorreto. a) Se o administrador não informar algum dado obrigatório, ou fornecer dados inválidos: o sistema reporta o fato e o caso de uso retorna ao passo 2.
Excluir Quantidade de Questões Respondidas no Nível (Não se aplica)		
Consultar Quantidade de Questões Respondidas no Nível		
	Item	Descrição
	Caso de Uso	CONSULTAR A QUANTIDADE DE QUESTÕES RESPONDIDAS NO NÍVEL
	Resumo	O administrador do sistema efetiva a consulta da quantidade de questões a serem respondidas em determinado nível.
	Ator	Administrador
	Pré-condições	O administrador deve estar autenticado no sistema.
	Pós-Condições	A quantidade de questão deve ter sido consultada no banco de dados do sistema.
	Fluxo Principal	<ol style="list-style-type: none"> 1. O administrador solicita a consulta da quantidade de questões a serem respondidas. 2. O sistema exibe os dados da quantidade de questões a serem respondidas. 3. O administrador visualiza os dados. 4. O caso de uso termina.
	Fluxo Alternativo	Fluxo Alternativo (2): A quantidade de questões a serem respondidas não existe. a. Se o sistema não encontrar o número de questões a serem respondidas no nível, o fato é reportado e o caso de uso retorna ao passo 1.
	Fluxo de Exceção	Não se aplica.

1.1.10. <RF_ADM010><Manter Quantidade de Questões do Mínimo de Acertos no Nível>

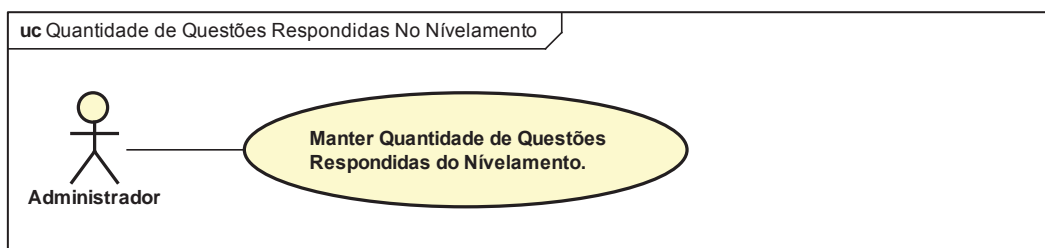


Manter Quantidade Mínima de Acertos do Nível	
Cadastrar a Quantidade Mínima de Acertos do Nível	
Item	Descrição
Caso de Uso	CADASTRAR A QUANTIDADE MÍNIMA DE ACERTOS DO NÍVEL
Resumo	O administrador do sistema efetiva o cadastro da quantidade mínima de acertos a serem efetivados no nível.
Ator	Administrador.
Pré-condições	O administrador deve estar autenticado no sistema.
Pós-Condições	A quantidade mínima de questões que deve ser respondida de forma correta deve ser cadastrada.
Fluxo Principal	<ol style="list-style-type: none"> 1. O administrador solicita o cadastro de uma nova quantidade mínima de questões. 2. O sistema requisita o cadastro de uma nova quantidade mínima de questões a serem respondidas de forma correta. 3. O administrador fornece as informações da quantidade mínima de questões a serem respondidas no nível. 4. O administrador confere os dados e confirma o cadastro. 5. O sistema cadastra a quantidade mínima de questões a serem respondidas de forma correta e o caso de uso termina.
Fluxo Alternativo	<p>Fluxo Alternativo (4): Erro no Lançamento.</p> <ol style="list-style-type: none"> a. O Administrador detecta o lançamento de uma informação errada, para a quantidade mínima de acerto. b. O Administrador corrige a informação que foi lançada erroneamente da quantidade mínima de questões, de forma correta. c. O sistema aceita a correção e o caso de uso continua a partir do passo 4.
Fluxo de Exceção	<p>Fluxo de Exceção (2): Dados obrigatórios ao cadastro da quantidade mínima de acertos em branco ou com formatos errados.</p> <ol style="list-style-type: none"> a) Se o administrador não informar algum dado obrigatório, ou fornecer dados inválidos: o

		sistema reporta o fato e o caso de uso retorna ao passo 2.
Alterar a Quantidade Mínima de Acertos do Nível		
Item	Descrição	
Caso de Uso	ALTERAR A QUANTIDADE MÍNIMA DE ACERTOS DO NÍVEL.	
Resumo	O administrador do sistema altera o cadastro da quantidade mínima de questões que devem ser respondidas de forma correta no nível.	
Ator	Administrador	
Pré-condições	O administrador deve estar autenticado no sistema.	
Pós-Condições	A quantidade mínima de questões a serem respondidas de forma correta no nível deve ser alterada.	
Fluxo Principal	<ol style="list-style-type: none"> 1. O administrador solicita a alteração da quantidade de questões a serem respondidas de forma correta dentro de determinado nível. 2. O administrador fornece as informações da quantidade mínima de questões a serem respondidas de forma correta em determinado nível. 3. O administrador confere os dados e confirma a alteração da quantidade mínima de questões a serem respondidas de forma correta. 4. O sistema altera o cadastro da quantidade mínima de questões a serem respondidas de forma correta, no nível, e o caso de uso termina. 	
Fluxo Alternativo	<p>Fluxo Alternativo (2): Erro no Lançamento.</p> <ol style="list-style-type: none"> a. O administrador detecta que lançou uma informação errada para a quantidade mínima de questões a serem respondidas de forma correta. b. O administrador corrige a informação que foi lançada erroneamente da quantidade de questões a serem respondidas de forma correta no nível. c. O sistema aceita a correção e o caso de uso continua a partir do passo 2. 	
Fluxo de Exceção	<p>Fluxo de Exceção (3): Dados obrigatórios ao cadastro da quantidade mínima de questões a serem respondidas de forma correta estão faltando ou em formato incorreto.</p> <ol style="list-style-type: none"> a) Se o administrador não informar algum dado obrigatório, ou fornecer dados inválidos: o sistema reporta o fato e o caso de uso retorna ao passo 2. 	
Excluir a Quantidade Mínima de Acertos do Nível (Não se aplica)		
Consultar a Quantidade Mínima de Acertos do Nível		
Item	Descrição	
Caso de Uso	CONSULTAR A QUANTIDADE MÍNIMA DE ACERTOS DO NÍVEL	
Resumo	O administrador do sistema efetiva a consulta da quantidade mínima de questões a serem respondidas de forma correta em determinado nível.	

Ator	Administrador.
Pré-condições	O administrador deve estar autenticado no sistema.
Pós-Condições	A quantidade mínima de questão deve ter sido consultada no banco de dados do sistema
Fluxo Principal	<ol style="list-style-type: none"> 1. O administrador solicita a consulta da quantidade mínima de questões a serem respondidas de forma correta. 2. O sistema exibe os dados da quantidade de questões a serem respondidas de forma correta. 3. O administrador visualiza os dados. 4. O caso de uso termina.
Fluxo Alternativo	<p>Fluxo Alternativo (2): A quantidade mínima de questões a serem respondidas de forma correta não existe.</p> <ol style="list-style-type: none"> a. Se o sistema não encontrar o número de questões a serem respondidas de forma correta no nível de dificuldade, o fato é reportado e o caso de uso retorna ao passo 1.
Fluxo de Exceção	Não se aplica.

1.1.11. <RF_ADM010><Manter Quantidade de Questões Respondidas no Nivelamento>

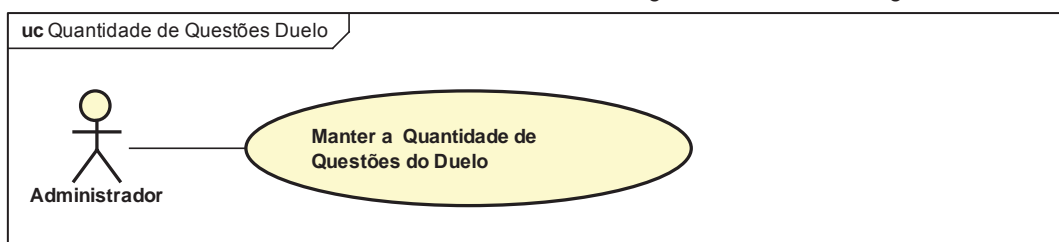


Manter Quantidade de Questões Respondidas no Nivelamento	
Cadastrar Quantidade de Questões Respondidas no Nivelamento	
Item	Descrição
Caso de Uso	CADASTRAR QUANTIDADE DE QUESTÕES RESPONDIDAS NO NIVELAMENTO
Resumo	O administrador do sistema efetiva o cadastro de nova quantidade de questões a serem respondidas na etapa de nivelamento.
Ator	Administrador
Pré-condições	O administrador deve estar autenticado no sistema.
Pós-Condições	Uma nova quantidade de questões a ser respondida no processo de nivelamento é cadastrada no sistema
Fluxo Principal	<ol style="list-style-type: none"> 1. O administrador solicita o cadastro de uma nova quantidade de questões a serem respondidas no processo de nivelamento. 2. O sistema requisita o cadastro de uma nova quantidade de questões. 3. O administrador fornece o número de questões para o cadastro de uma nova quantidade de questões a serem respondidas no processo de

		<p>nivelamento.</p> <p>4. O administrador confere os dados e confirma o cadastro.</p> <p>5. O sistema cadastra a quantidade de questões e o caso de uso termina.</p>
Fluxo Alternativo		<p>Fluxo Alternativo (4): Erro no Lançamento.</p> <p>a. O Administrador detecta que lançou uma informação errada para a quantidade de questões.</p> <p>b. O Administrador corrige a informação que foi lançada erroneamente.</p> <p>c. O sistema aceita a correção e o caso de uso continua a partir do passo 4.</p>
Fluxo de Exceção		<p>Fluxo de Exceção (2): Dados obrigatórios ao cadastro da quantidade de questões do processo de nivelamento estão em branco ou em formato incorreto.</p> <p>a) Se o administrador não informar algum dado obrigatório, ou fornecer dados inválidos: o sistema reporta o fato e o caso de uso retorna ao passo 2.</p>
Alterar Quantidade de Questões Respondidas no Nivelamento		
Item	Descrição	
Caso de Uso	ALTERAR A QUANTIDADE DE QUESTÕES RESPONDIDAS NO NIVELAMENTO.	
Resumo	O administrador do sistema altera o cadastro da quantidade de questões a serem respondidas no processo de nivelamento.	
Ator	Administrador	
Pré-condições	O administrador deve estar autenticado no sistema	
Pós-Condições	A quantidade de questões a serem respondidas no processo de nivelamento deve ser alterada.	
Fluxo Principal	<p>1. O administrador solicita a alteração da quantidade de questões a serem respondidas no processo de nivelamento.</p> <p>2. O administrador fornece as informações de alteração da quantidade de questões a serem respondidas no processo de nivelamento.</p> <p>3. O administrador confere os dados e confirma a alteração da quantidade de questões a serem respondidas no processo de nivelamento.</p> <p>4. O sistema altera o cadastro da quantidade de questões a serem respondidas no processo de nivelamento e o caso de uso termina.</p>	
Fluxo Alternativo	<p>Fluxo Alternativo (2): Erro no Lançamento</p> <p>a. O Administrador detecta o lançamento de uma errada.</p> <p>b. O Administrador corrige a informação que foi lançada erroneamente.</p> <p>c. O sistema aceita a correção e o caso de uso continua a partir do passo 3.</p>	
Fluxo de Exceção	Fluxo de Exceção (3): Dados obrigatórios ao cadastro da	

		quantidade de questões de nivelamento, em branco ou com formatos errados. a) Se o administrador não informar algum dado obrigatório, ou fornecer dados inválidos: o sistema reporta o fato e o caso de uso retorna ao passo 2.
Excluir Quantidade de Questões Respondidas no Nivelamento – Não se Aplica		
Consultar Quantidade de Questões Respondidas no Nivelamento		
Item	Descrição	
Caso de Uso	CONSULTAR A QUANTIDADE DE QUESTÕES A SEREM RESPONDIDAS NO NIVELAMENTO.	
Resumo	O administrador do sistema efetiva a consulta da quantidade de questões a serem respondidas no nivelamento.	
Ator	Administrador.	
Pré-condições	O administrador deve estar autenticado no sistema.	
Pós-Condições	A quantidade de questões a serem respondidas no nivelamento deve ser consultada no sistema.	
Fluxo Principal	<ol style="list-style-type: none"> 1. O administrador solicita a consulta da quantidade de questão a serem respondidas no nivelamento. 2. O sistema exibe a quantidade de questões a serem respondidas no nivelamento. 3. O administrador visualiza os dados. 4. O caso de uso termina. 	
Fluxo Alternativo	Fluxo Alternativo (2): A quantidade de questões de nivelamento não existe. a. Se o sistema não encontrar o número de questões a serem respondidas no nivelamento, o fato é reportado e o caso de uso retorna ao passo 1.	
Fluxo de Exceção	Não se aplica.	

1.1.12. <RF_ADM0011><Manter a Quantidade de Questões do Duelo>



Manter a Quantidade de Questões do Duelo		
Cadastrar a Quantidade de Questões do Duelo		
Item	Descrição	
Caso de Uso	CADASTRAR A QUANTIDADE DE QUESTÕES DO DUELO	
Resumo	O administrador do sistema efetiva o cadastro de nova quantidade de questões do duelo.	
Ator	Administrador.	
Pré-condições	O administrador deve estar autenticado no sistema.	

Pós-Condições	Uma nova quantidade de questões vai ser cadastrada para o duelo.
Fluxo Principal	<ol style="list-style-type: none"> 1. O administrador solicita o cadastro de uma nova quantidade de questões para o processo de duelo. 2. O sistema requisita o cadastro de uma nova quantidade de questões para o duelo. 3. O administrador fornece as informações básicas para o cadastro da nova quantidade de questões do duelo. 4. O administrador confere os dados e confirma o cadastro. 5. O sistema cadastra a quantidade de duelos e o caso de uso termina.
Fluxo Alternativo	<p>Fluxo Alternativo (4): Erro no Lançamento</p> <ol style="list-style-type: none"> a. O Administrador detecta que lançou uma informação errada para a quantidade de questões do duelo. b. O Administrador corrige a informação que foi lançada erroneamente da quantidade de questões do duelo. c. O sistema aceita a correção e o caso de uso continua a partir do passo 4.
Fluxo de Exceção	<p>Fluxo de Exceção (2): Dados obrigatórios ao cadastro da quantidade de questões em branco ou com formatos errados.</p> <ol style="list-style-type: none"> a) Se o administrador não informar algum dado obrigatório, ou fornecer dados inválidos: o sistema reporta o fato e o caso de uso retorna ao passo 2.

Alterar a Quantidade de Questões do Duelo

Item	Descrição
Caso de Uso	ALTERAR A QUANTIDADE DE QUESTÕES DO DUELO
Resumo	O administrador do sistema altera o cadastro da quantidade de questões de duelo.
Ator	Administrador.
Pré-condições	O administrador deve estar autenticado no sistema.
Pós-Condições	A quantidade de questões de duelo deve ser alterada.
Fluxo Principal	<ol style="list-style-type: none"> 1. O administrador solicita a alteração da quantidade de questões. 2. O administrador fornece as informações de alteração de quantidade de questões. 3. O administrador confere os dados e confirma a alteração. 4. O sistema altera o cadastro da quantidade de questões e o caso de uso termina.
Fluxo Alternativo	<p>Fluxo Alternativo (2): Erro no Lançamento</p> <ol style="list-style-type: none"> a. O administrador detecta o lançamento de informações erradas. b. O administrador corrige a informação que foi lançada erroneamente da dica de nível de dificuldade de questão. c. O sistema aceita a correção e o caso de uso continua a partir do passo 3.

	Fluxo de Exceção	Fluxo de Exceção (3): Dados obrigatórios ao cadastro de duelo, em branco ou com formatos errados. a) Se o administrador não informar algum dado obrigatório, ou fornecer dados inválidos: o sistema reporta o fato e o caso de uso retorna ao passo 2.
Excluir a Quantidade de Questões do Duelo		
	Item	Descrição
	Caso de Uso	EXCLUIR QUANTIDADE DE QUESTÕES DO DUELO.
	Resumo	O administrador do sistema efetiva a exclusão da quantidade de questões a serem respondidas no duelo.
	Ator	Administrador.
	Pré-condições	O administrador deve estar autenticado no sistema.
	Pós-Condições	A quantidade de questões de um duelo deve ser excluída.
	Fluxo Principal	<ol style="list-style-type: none"> 1. O administrador solicita a exclusão da quantidade de questões do duelo. 2. O sistema exibe os dados da quantidade de questões do duelo e requisita a exclusão da mesma. 3. O administrador procede com exclusão da quantidade de questões d duelo. 4. O administrador confere os dados e confirma a exclusão da quantidade de questões do duelo. 5. O sistema efetiva a exclusão e o caso de uso termina.
	Fluxo Alternativo	Não se aplica.
	Fluxo de Exceção	Não se aplica.
Consultar a Quantidade de Questões do Duelo		
	Item	Descrição
	Caso de Uso	CONSULTAR A QUANTIDADE DE QUESTÕES DO DUELO
	Resumo	O administrador do sistema efetiva a consulta da quantidade de questões de um duelo.
	Ator	Administrador.
	Pré-condições	O administrador deve estar autenticado no sistema.
	Pós-Condições	A quantidade de questões de determinado duelo deve ter seus dados consultados no sistema.
	Fluxo Principal	<ol style="list-style-type: none"> 1. O administrador solicita a consulta da quantidade de questões de um duelo. 2. O sistema exibe os dados da quantidade de questões de um duelo. 3. O administrador visualiza os dados. 4. O caso de uso termina.
	Fluxo Alternativo	Não se Aplica.
	Fluxo de Exceção	Não se Aplica.

<ATOR ALUNO>

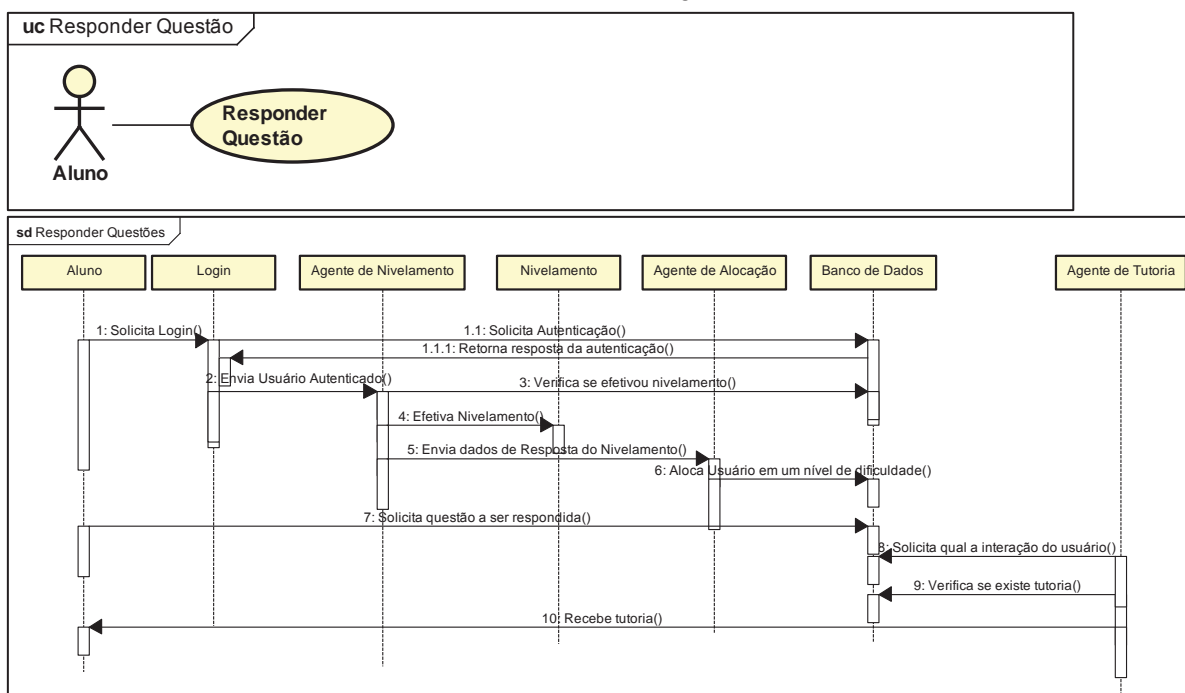


1.1.13. <RF_USU001>< MANTER USUÁRIO >

Manter Usuário	
Cadastrar Usuário	
Item	Descrição
Caso de Uso	CADASTRAR USUÁRIO
Resumo	O usuário do sistema efetiva seu próprio cadastro, com suas devidas permissões para posteriores acessos. São gerados logs desse caso de uso que são inseridos no banco de dados.
Ator	Aluno
Pré-condições	O aluno deve ter acesso à aplicação.
Pós-Condições	Um novo usuário deve ser cadastrado no sistema
Fluxo Principal	<ol style="list-style-type: none"> 1. O aluno solicita o cadastro de um novo usuário. 2. O sistema requisita o cadastro de um novo usuário. 3. O aluno fornece as informações básicas para o cadastro de um novo usuário. 4. O aluno confere os dados e confirma o cadastro. 5. O sistema cadastra o usuário e o caso de uso termina.
Fluxo Alternativo	Fluxo Alternativo (4): Erro no Lançamento <ol style="list-style-type: none"> a. O aluno detecta que lançou uma informação errada. b. O aluno corrige a informação que foi lançada erroneamente. c. O sistema aceita a correção e o caso de uso continua a partir do passo 4 .
Fluxo de Exceção	Fluxo de Exceção (5): Dados obrigatórios ao cadastro do usuário em branco ou com formatos errados. <ol style="list-style-type: none"> a) Se o aluno não informar algum dado obrigatório, ou fornecer dados inválidos: o sistema reporta o fato e o caso de uso retorna ao passo 2.
Alterar Usuário	
Item	Descrição
Caso de Uso	ALTERAR USUÁRIO
Resumo	O aluno efetiva a alteração dados referentes a seu usuário.
Ator	Aluno.
Pré-condições	O aluno deve estar autenticado no sistema.
Pós-Condições	O aluno deve ter seus dados alterados no sistema.
Fluxo Principal	<ol style="list-style-type: none"> 1. O aluno solicita a alteração de dados de seu usuário. 2. O sistema exibe os dados do usuário e requisita a alteração do mesmo. 3. O aluno procede com alteração do usuário.

		<ol style="list-style-type: none"> 4. O aluno confere os dados e confirma a alteração do cadastro. 5. O sistema efetiva a alteração e o caso de uso termina.
	Fluxo Alternativo	<p>Fluxo Alternativo (5): Erro no Lançamento</p> <ol style="list-style-type: none"> a. O aluno detecta o lançamento de uma informação errada. b. O aluno corrige a informação o que foi lançada erroneamente no cadastro de usuário. c. O sistema aceita a correção e o caso de uso continua a partir do passo 4.
	Fluxo de Exceção	Não de aplica.
Excluir Usuário		
	Item	Descrição
	Caso de Uso	EXCLIR USUÁRIO
	Resumo	O aluno solicita a exclusão de seu usuário.
	Ator	Aluno.
	Pré-condições	O aluno deve estar autenticado no sistema.
	Pós-Condições	O aluno deve ter seu status alterado para inativo.
	Fluxo Principal	<ol style="list-style-type: none"> 1. O aluno solicita a exclusão de seu usuário. 2. O sistema exibe os dados do Aluno e requisita a exclusão do mesmo. 3. O aluno procede com exclusão do usuário. 4. O aluno confere os dados e confirma a exclusão do cadastro. 5. O sistema efetiva a alteração e o caso de uso termina.
	Fluxo Alternativo	Não se aplica.
	Fluxo de Exceção	Não se aplica.
Consultar Usuário		
	Item	Descrição
	Caso de Uso	CONSULTAR USUÁRIO
	Resumo	O aluno efetiva a consulta dos dados de seu usuário.
	Ator	Aluno.
	Pré-condições	O aluno deve estar autenticado no sistema.
	Pós-Condições	O aluno deve ter seus dados consultados no sistema
	Fluxo Principal	<ol style="list-style-type: none"> 1. O Aluno solicita a consulta dos dados de seu usuário. 2. O sistema exibe os dados do usuário. 3. O aluno visualiza os dados. 4. O caso de uso termina.
	Fluxo Alternativo	Não se aplica.
	Fluxo de Exceção	Não se aplica.

1.1.14. <RF_USU002>< RESPONDER QUESTÃO >



Manter Responder Questão

Cadastrar Resposta Questão

Item	Descrição
Caso de Uso	CADASTRAR RESPOSTA QUESTÃO
Resumo	O aluno requisita questões, para serem respondidas. O sistema as apresenta. Desta forma o aluno procede com a leitura, interpretação e devida resposta. São gerados logs desse caso de uso, que serão inseridos em um banco de dados.
Ator	Aluno.
Pré-condições	O aluno deve estar autenticado na aplicação.
Pós-Condições	Um novo registro de resposta da questão deve ser cadastrado no sistema.
Fluxo Principal	<ol style="list-style-type: none"> 1. O Aluno responde a uma questão. 2. O sistema requisita a correção de uma questão. 3. O sistema retorna uma mensagem se a questão foi respondida de forma certa ou errada. 4. O sistema cadastra a resposta da questão e o caso de uso termina.
Fluxo Alternativo	Fluxo Alternativo (3): Verifica se é a primeira, segunda ou terceira tentativa de resposta do aluno, para que a tutoria da resposta seja apresentada. O sistema apresenta a tutoria e o caso de uso é finalizado.
Fluxo de Exceção	Não se aplica .

Alterar Resposta Questão

Item	Descrição
Caso de Uso	ALTERAR RESPOSTA QUESTÃO – Não se aplica.

Excluir Resposta Questão

Item	Descrição
Caso de Uso	EXCULIR RESPOSTA QUESTÃO – Não se aplica.

Consultar Resposta Questão

Item	Descrição
Caso de Uso	CONSULTAR RESPOSTA QUESTÃO – Não se aplica.

REQUISITOS NÃO FUNCIONAIS

Nesta seção são descritos os requisitos não funcionais do sistema, conforme segue:

<RNF001><Requisito não funcional Desempenho>

Identificador	RNF001	Categoria	Desempenho
Nome	Tempo limite para processamento de resposta a requisições.		
Data de criação	15/04/2017	Autor	André Luís Stefanello
Data da última alteração	N/A	Autor	N/A
Versão	1	Prioridade	Essencial
Descrição	<p>O módulo do Aluno do sistema MedicaGame, requer agilidade nas respostas a requisições. Em função desta realidade o sistema provê de ambiente com:</p> <p>Mais de um CPU, para o processamento de requisições;</p> <p>Trafego ilimitado;</p> <p>Utilização de memória auto escalável;</p> <p>Espaço em disco de 10,44 GB, este também escalável;</p>		

<RNF002><Requisito não funcional Segurança>

Identificador	RNF002	Categoria	segurança
Nome	Autenticação dos usuários para acesso aos sistemas.		
Data de criação	15/04/2017	Autor	André Luís Stefanello
Data da última alteração	N/A	Autor	N/A
Versão	1	Prioridade	Essencial
Descrição	<p>Tanto a aplicação administrativa do sistema MedicalGame, quanto a aplicação que o aluno acessa, são aprimoradas por autenticação de usuário e senha, o que é requisito essencial para um sistema que esteja rodando na <i>internet</i>.</p> <p>As aplicações incorporam políticas de acesso particulares para cada módulo da aplicação.</p>		

<RNF003><Requisito não funcional Usabilidade>

Identificador	RNF003	Categoria	Usabilidade
Nome	Usabilidade dos sistemas.		
Data de criação	15/04/2017	Autor	André Luís Stefanello
Data da última alteração	N/A	Autor	N/A
Versão	1	Prioridade	Importante
Descrição	<p>Tanto a aplicação administrativa do sistema MedicalGame, quanto à aplicação que o aluno acessa, é desenvolvida para rodar na Web, foram desenvolvidas utilizando padrões de responsividade;</p> <p>A interface das duas aplicações deverá se comportar de forma adequada indiferente do <i>front-end</i> que será utilizado para acesso – Smartphone, tablete, browser.</p>		

<RNF004><Requisito não funcional Compatibilidade>

Identificador	RNF004	Categoria	Compatibilidade
Nome	Compatibilidade dos sistemas.		
Data de criação	15/04/2017	Autor	André Luís Stefanello
Data da última alteração	N/A	Autor	N/A
Versão	1	Prioridade	Importante
Descrição	Como a aplicação, tanto administrativa quanto a aplicação do aluno, são disponibilizadas na Web, as mesmas devem rodar em qualquer sistema operacional, que hoje está no mercado, em todos os sistemas operacionais o comportamento deve ser o mesmo, quanto às funcionalidades das aplicações.		

<RNF005><Requisito não funcional Padrão>

Identificador	RNF005	Categoria	Padrão
Nome	Padrão dos sistemas.		
Data de criação	15/04/2017	Autor	André Luís Stefanello
Data da última alteração	N/A	Autor	N/A
Versão	1	Prioridade	Importante
Descrição	<p>Interface: abrigar lógicas de tela, validação de campos, acionamento de comandos, códigos para design de interface.</p> <p>Dados: abrigar lógicas de acesso a dados, comandos SQL.</p> <p>Segurança: abrigar lógicas de autenticação, auditoria, manutenção de usuários.</p> <p>Infraestrutura: abrigar lógicas não relacionadas a regras de negócio, interfaces gráficas, dados ou segurança, mas que poderão ser utilizadas em todas estas camadas. Conterá recursos para gravação de logs e envio de e-mails de validação.</p>		

APÊNDICE D - ESTÓRIA CENÁRIO DE TESTES MEDICALGAME

ESTÓRIA - CENÁRIO DE TESTES, MEDICALGAME.

Aluno é um estudante de medicina. Uma de suas atividades é acessar e interagir com sistema MedicalGame.

Em um dia de estudos, Aluno efetiva o acesso ao sistema MedicalGame, logo após seu acesso, aluno é convidado a realizar a resolução de algumas questões para seu possível nivelamento, dentro da aplicação.

Após a resolução das questões de nivelamento, se houver desafio para o mesmo, ele será convidado a respondê-lo, no desafio, o Aluno responde questões referentes ao nível de dificuldade a que está vinculado, quando o processo de desafio for terminado o sistema passa para o próximo passo.

O aluno inicia o processo de resolução das questões, que estão separadas por nível de dificuldade, a quantidade de questões e níveis de dificuldade foram configurados pelo administrador do sistema, após cada resposta do aluno, o sistema retorna uma mensagem de acerto ou erro para a questão, sendo que o mesmo tem três tentativas de resolução para a mesma questão, todas as questões tem tutorias cadastradas, que são apresentadas em cada interação do aluno com o sistema.

Quando o Aluno efetivar a resolução do número de questões necessárias para a alteração de nível, o sistema verifica os acertos e modifica ou não o nível do Aluno.

Depois de finalizar a resolução de todas as questões de todos os níveis, o sistema retorna para o Aluno que o mesmo efetivou todos os exercícios de todos os níveis de forma satisfatória.

O cenário testa uma série de características do MedicalGame.

1. Cadastro de novo Aluno no sistema.
2. Autenticação por logon no sistema.
3. Funcionamento do Agente de Nivelamento.
4. Funcionamento do Agente Desafio.
5. Funcionamento do Agentes de Alocação.
6. Funcionamento do Agentes de Tutoria.
7. Funcionamento do Sistema MedicalGame.
8. Recuperação e modificação de registros.
9. Verificação da conexão com o banco de dados das questões e tutorias.