

UNIVERSIDADE FEDERAL DO PAMPA

Iderli Pereira de Souza Filho

**Um Processo de Engenharia de Requisitos
para Sistemas Multiagentes**

Alegrete
2020

Iderli Pereira de Souza Filho

**Um Processo de Engenharia de Requisitos para
Sistemas Multiagentes**

Trabalho de Conclusão de Curso apresentado ao Curso de Graduação em Engenharia de Software da Universidade Federal do Pampa como requisito parcial para a obtenção do título de Bacharel em Engenharia de Software.

Orientador: Prof. Dr. Gilleanes Thorwald
Araujo Guedes

Alegrete
2020



SERVIÇO PÚBLICO FEDERAL
MINISTÉRIO DA EDUCAÇÃO
Universidade Federal do Pampa

Iderli Pereira de Souza Filho

Um Processo de Engenharia de Requisitos para Sistemas Multiagentes

Trabalho de Conclusão de Curso apresentado
ao Curso de Graduação em Engenharia de
Software da Universidade Federal do Pampa
como requisito parcial para a obtenção do tí-
tulo de Bacharel em Engenharia de Software.

Trabalho de Conclusão de Curso defendido e aprovado em: 18/09/2020

Banca examinadora:

Prof. Dr. Gilleanes Thorwald Araujo Guedes
Orientador
(Unipampa)

Prof. Dr. João Pablo Silva da Silva
(Unipampa)

Prof. Dr. Maicon Bernardino da Silveira
(Unipampa)



Documento assinado eletronicamente por **MAICON BERNARDINO DA SILVEIRA, PROFESSOR DO MAGISTERIO SUPERIOR**, em 18/09/2020, às 16:31, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do [Decreto nº 8.539, de 8 de outubro de 2015](#).



Documento assinado eletronicamente por **JOAO PABLO SILVA DA SILVA, PROFESSOR DO MAGISTERIO SUPERIOR**, em 18/09/2020, às 16:32, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do [Decreto nº 8.539, de 8 de outubro de 2015](#).



Documento assinado eletronicamente por **GILLEANES THORWALD ARAUJO GUEDES, PROFESSOR DO MAGISTERIO SUPERIOR**, em 18/09/2020, às 16:32, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do [Decreto nº 8.539, de 8 de outubro de 2015](#).



A autenticidade deste documento pode ser conferida no site https://sei.unipampa.edu.br/sei/controlador_externo.php?acao=documento_conferir&id_orgao_acesso_externo=0, informando o código verificador **0363027** e o código CRC **F34EF11B**.

Universidade Federal do Pampa, Campus Alegrete
Av. Tiarajú, 810 – Bairro: Ibirapuitã – Alegrete – RS CEP: 97.546-550

Telefone: (55) 3422-8400

Este trabalho é dedicado à todos que me apoiaram durante seu desenvolvimento.

AGRADECIMENTOS

Agradeço primeiramente à minha família. Meus irmãos, que estão sempre me apoiando e meus pais que ficam ao meu lado em todos os momentos. Sei que sempre posso contar com vocês e me orgulho muito disso.

Ao meu orientador Prof. Dr. Gilleanes Thorwald Araujo, por ter aceitado me orientar a distância, por estar sempre disponível para tirar dúvidas e por todas as correções feitas no texto. Sem sua orientação não seria possível finalizar este trabalho.

Um agradecimento especial ao Guilherme Souza Santos, a Prof Dra. Fabiane Flores Penteadó Galafassi e ao Prof. Cristiano Galafassi, pelas contribuições dadas a este trabalho.

Agradeço também ao colega do grupo de pesquisa Giovane Davila Mendonça, que contribuiu no desenvolvimento do processo que estamos propondo.

Por fim, agradeço a Universidade Federal do Pampa (UNIPAMPA) pela oportunidade de um ensino de qualidade de forma gratuita.

Obrigado a todos!

Enquanto alguns já fazem planos
De hoje até o fim do ano
Fica esperto pra não ter engano
Mande um abraço no seu prédio
Vença o tédio e diga
Fica em casa
(Fica em Casa, Dingo Bells)

RESUMO

Na engenharia de software, a subárea de Engenharia de Requisitos (ER) é uma fase crucial para todo o processo de desenvolvimento. Isso não é diferente para Sistemas Multiagentes (SMAs). Esse tipo de sistema se mostrou uma boa alternativa para lidar com soluções complexas, visto a sua capacidade de dividir a complexidade entre diversos agentes. No entanto, SMAs possuem requisitos específicos, como por exemplo as crenças, desejos e intenções dos agentes. Tais informações podem ser representadas por meio do modelo *Belief-Desire-Intention* ou crença–desejo–intenção (BDI), uma das abordagens mais conhecidas para integrar as habilidades cognitivas desejadas para agentes autônomos. Além dos requisitos necessários à aplicação do modelo BDI, podemos destacar as percepções que o agente pode possuir sobre o ambiente, os planos desses agentes para atingir um objetivo (intenção) e as ações que esses agentes podem executar no ambiente quando tentam atingir um objetivo. Tendo em mente esses requisitos, percebemos a necessidade de um processo de engenharia de requisitos para Sistemas Multiagentes. Deste modo, este trabalho tem como objetivo apresentar um processo que tenha como base o suporte para a execução das subáreas de ER, elicitacão, análise, especificacão e validacão, e permita a representacão destes requisitos específicos para Sistemas Multiagentes. Para a concepçao deste processo, analisamos os processos que suportam ER para SMAs, realizando uma revisao sistemática de literatura, visando identificar os pontos fortes e fracos desses processos. Deste modo, definimos as etapas do processo e propomos um processo de ER para SMAs. O processo proposto foi validado por meio de sua aplicacão no sistema Heráclito, em que se demonstra a execucao das etapas do processo e os resultados obtidos em cada uma delas, dando destaque à documentacão de cenários e diagramas de casos de uso produzidos durante a execucao. Com essa execucao do processo, conseguimos avaliar suas etapas, identificar trabalhos futuros e pontos de melhoria no processo.

Palavras-chave: Engenharia de Requisitos. Sistemas multiagentes. Agentes, Papeis de Agentes, Cenários Internos, Modelo BDI.

ABSTRACT

In software engineering, the requirements engineering subárea (RE) is a crucial phase for all the development process. It is not different for multiagent systems (MAS). This kind of system has demonstrated to be a good alternative to dealing with complex solutions, given its capability to divide the complexity among several agents. However, multiagent systems have specific requirements, such as beliefs, desires and intentions. This information can be represented by means of Belief-Desire-Intention (BDI) model, one of the most known approaches to integrate the cognitive capabilities desired to autonomous agents. Beyond of the requirements needed to the BDI model application, we can highlight the perceptions that an agent can have about the environment, the plans of this agent to achieve a goal (intention) and the actions that these agents can perform in the environment while trying to achieve a goal. Keeping in mind these requirements, we realized the necessity of a requirements engineering process for multiagent systems. This way, this work has as its goal to present a process supporting the RE subareas, elicitation, analysis, specification and validation and allowing the representation of these specific requirements for multiagent systems. During the conception of this process, we analysed processes that support RE for MAS, performing a systematic literature review, aiming to identify the strengths and weaknesses of these processes. Thus, we defined the process phases and we proposed a RE process for MAS. The proposed process was validated by its application in Heraclito system, in which we demonstrated the execution of the process phases and results obtained in each one of them, highlighting the scenarios documentation and use-case diagrams produced during the execution. With the process application, we managed to evaluate its phases, to identify future works and improvements points in the process.

Key-words: Requirements engineering. Multiagent systems. Agents, Agents Roles, Internal Scenarios, BDI Model.

LISTA DE FIGURAS

Figura 1 – Estrutura típica de um sistema multiagente.	28
Figura 2 – Metamodelo de Guedes	34
Figura 3 – Processo de Pesquisa	42
Figura 4 – Visão geral do estágio de condução.	43
Figura 5 – Número de trabalhos rejeitados por critério de exclusão.	44
Figura 6 – Visão geral do processo proposto.	58
Figura 7 – Subprocesso de elicitação de requisitos do processo.	59
Figura 8 – Subprocesso de identificação de cenários internos dos agentes	61
Figura 9 – Análise SWOT da Metodologia Homer.	64
Figura 10 – Diagrama de Casos de Uso Parcial - Verificar Prova do Aluno	66
Figura 11 – Diagrama de Casos de Uso Parcial - Verificar questão do aluno.	68
Figura 12 – Diagrama de Casos de Uso Parcial - Verificar ação do aluno	70
Figura 13 – Diagrama de Casos de Uso Parcial - Planejar questão.	73
Figura 14 – Diagrama de Casos de Uso Parcial - Planejar sequência de conteúdo	76
Figura 15 – Diagrama de Casos de Uso Parcial - Informar questão ao aluno.	78
Figura 16 – Diagrama de Casos de Uso Parcial - Definir estratégia de ensino.	81
Figura 17 – Diagrama de Casos de Uso Parcial - Aplicar reforço ao aluno.	85
Figura 18 – Diagrama de Casos de Uso Parcial - Fornecer avaliação de questão	88
Figura 19 – Diagrama de Casos de Uso Parcial - Fornecer resultado de prova.	91
Figura 20 – Diagrama de Casos de Uso Parcial - Informar erro em questão.	95
Figura 21 – Diagrama de Casos de Uso Parcial - Informar ociosidade do aluno.	97
Figura 22 – Diagrama de Casos de Uso Parcial - Fornecer ajuda ao aluno.	99
Figura 23 – Diagrama de Casos de Uso Parcial - Informar perfil do aluno.	101
Figura 24 – Diagrama de Casos de Uso Parcial - Informar rede bayesiana	103
Figura 25 – Diagrama de casos de uso Final	105

LISTA DE TABELAS

Tabela 1 – Strings e filtros aplicados em cada base bibliográfica	39
Tabela 2 – Critérios de inclusão e exclusão.	40
Tabela 3 – Índices de Qualidade dos Estudos.	41
Tabela 4 – Cobertura das etapas de engenharia de requisitos das metodologias. . .	46
Tabela 5 – Objetivo - Verificar prova do aluno	67
Tabela 6 – Percepção - Perceber que um aluno finalizou a prova	67
Tabela 7 – Objetivo - Verificar questão do aluno	69
Tabela 8 – Percepção - Perceber solicitação de verificação	69
Tabela 9 – Plano - Avaliar questão	70
Tabela 10 – Objetivo - Verificar ação do aluno	71
Tabela 11 – Percepção - Perceber ação do aluno	71
Tabela 12 – Plano - Verificar questão em tempo de execução	72
Tabela 13 – Objetivo - Planejar questão	74
Tabela 14 – Percepção - Perceber solicitação de nova questão pelo Mediador	74
Tabela 15 – Plano - Selecionar questão na base de conhecimento	75
Tabela 16 – Percepção - Perceber envio do perfil do aluno pelo Mediador	75
Tabela 17 – Percepção - Perceber envio de rede bayesiana do aluno pelo Analista .	76
Tabela 18 – Objetivo - Planejar sequência de conteúdo	77
Tabela 19 – Percepção - Perceber solicitação de nova sequência de conteúdo	77
Tabela 20 – Plano - Determinar conteúdo	78
Tabela 21 – Objetivo - Informar questão ao aluno	79
Tabela 22 – Percepção - Perceber solicitação de nova questão	79
Tabela 23 – Percepção - Perceber envio de nova questão do Especialista Técnico . .	80
Tabela 24 – Plano - Preparar aplicação da questão	80
Tabela 25 – Objetivo - Definir estratégia de ensino	82
Tabela 26 – Percepção - Perceber que novo aluno se cadastrou no sistema	82
Tabela 27 – Percepção - Perceber envio de sequência de conteúdo pelo Especialista Técnico	83
Tabela 28 – Percepção - Perceber envio de rede bayesiana do aluno pelo Analista Bayesiano	83
Tabela 29 – Plano - Atualizar perfil do aluno	84
Tabela 30 – Plano - Atualizar estratégia de ensino	84
Tabela 31 – Objetivo - Aplicar reforço ao aluno	86
Tabela 32 – Percepção - Perceber que um Aluno falhou em uma prova	86
Tabela 33 – Percepção - Perceber que um Aluno solicitou reforço	87
Tabela 34 – Plano - Preparar reforço	87
Tabela 35 – Objetivo - Fornecer avaliação de questão	89
Tabela 36 – Percepção - Perceber finalização de uma questão	89

Tabela 37 – Percepção - Perceer verificação de questão pelo Especialista	90
Tabela 38 – Plano - Preparar avaliação de questão	90
Tabela 39 – Objetivo - Fornecer resultado de prova	92
Tabela 40 – Percepção - Perceber verificação de prova pelo Especialista Técnico . . .	93
Tabela 41 – Percepção - Perceber finalização de uma prova	93
Tabela 42 – Percepção - Perceber etapa final da prova	94
Tabela 43 – Percepção - Perceber metade da resolução da prova	94
Tabela 44 – Plano - Preparar verificação da prova	95
Tabela 45 – Objetivo - Informar erro em questão ao aluno	96
Tabela 46 – Percepção - Perceber mensagem de anomalia na questão do aluno . . .	96
Tabela 47 – Plano - Preparar retorno de anomalia na questão	97
Tabela 48 – Objetivo - Informar ociosidade do aluno	98
Tabela 49 – Percepção - Perceber ociosidade do aluno	98
Tabela 50 – Objetivo - Fornecer ajuda ao aluno	100
Tabela 51 – Percepção - Perceber solicitação de ajuda	100
Tabela 52 – Plano - Preparar ajuda ao aluno	101
Tabela 53 – Objetivo - Informar perfil do aluno	102
Tabela 54 – Percepção - Perceber solicitação de perfil do aluno pelo Especialista Técnico	102
Tabela 55 – Objetivo - Informar rede bayesiana do aluno	104
Tabela 56 – Percepção - Perceber solicitação de envio de rede bayesiana	104
Tabela 57 – Percepção - Atualizar rede bayesiana do aluno	105

LISTA DE SIGLAS

BDI *Belief-Desire-Intention* ou crença–desejo–intenção

ER Engenharia de Requisitos

SMA Sistema Multiagentes

SWEBOK *Software Engineering Body of Knowledge* ou Guia do Conhecimento da Engenharia de Software

SUMÁRIO

1	INTRODUÇÃO	23
1.1	Motivação	24
1.2	Objetivos e Cronograma das atividades	25
1.2.1	Objetivo Geral	25
1.2.2	Objetivos Específicos	25
1.3	Contribuições	25
1.4	Metodologia Utilizada no Trabalho	26
1.5	Organização do Trabalho	26
2	FUNDAMENTAÇÃO TEÓRICA	27
2.1	Sistemas Multiagentes	27
2.1.1	Características de Agentes	28
2.1.2	Modelo BDI	28
2.1.3	Papel de Agente	29
2.1.4	Percepções do Agente	29
2.1.5	Cenários de Agente	29
2.1.6	Plano de Agente	30
2.1.7	Ação do Agente	30
2.1.8	Organização	30
2.2	Engenharia de Requisitos	30
2.2.1	Elicitação de Requisitos	31
2.2.2	Análise de Requisitos	32
2.2.3	Especificação de Requisitos	32
2.2.4	Validação de Requisitos	32
2.2.5	Processo de requisitos de software	33
2.3	Casos de Uso Internos	33
3	REVISÃO SISTEMÁTICA DE LITERATURA	37
3.1	Apresentação da Revisão Sistemática	37
3.1.1	Questões de Pesquisa	37
3.1.2	Identificação e seleção de estudos primários	38
3.1.3	Critérios de inclusão e exclusão	40
3.1.4	Avaliação de qualidade dos estudos	40
3.1.5	Etapa de Condução	42
3.1.6	Etapa de Extração dos dados	44
3.2	Resultados da Revisão	45
3.2.1	Resultados Gerais	45
3.2.2	Primeira Questão de Pesquisa (RQ1)	45

3.2.3	Segunda Questão de Pesquisa (RQ2)	47
3.2.4	Terceira Questão de Pesquisa (RQ3)	47
3.2.5	Quarta Questão de Pesquisa (RQ4)	54
3.3	Ameaças a Revisão	55
4	UM PROCESSO DE ENGENHARIA DE REQUISITOS PARA SISTEMAS MULTIAGENTES	57
4.1	Apresentação do Processo proposto	57
4.2	Execução do Processo	62
4.2.1	Elicitação de Requisitos com Homer	62
4.2.2	Análise SWOT da Metodologia Homer	63
4.2.3	Identificação dos Agentes e Papéis de Agentes	64
4.2.4	Produção dos Cenários	66
4.2.5	Validação dos Requisitos	106
4.2.6	Resultados Extraídos da Execução do Processo	107
5	CONSIDERAÇÕES FINAIS	109
	REFERÊNCIAS	111
	Índice	119

1 INTRODUÇÃO

Agentes de software são um paradigma de software que fornece abstrações de agentes para o desenvolvimento de sistemas abertos, distribuídos e heterogêneos (GAN et al., 2020). Esses agentes de software se caracterizam por serem autônomos, possuem habilidade social, reatividade e pró-atividade (HAJDUK; SUKOP; HAUN, 2019).

Um Sistema Multiagente (SMA) é um tipo específico de sistema, composto de vários agentes que interagem entre si para atingir determinados objetivos. Esses sistemas podem ser usados para resolver problemas que são difíceis ou impossíveis para um sistema monolítico ou para um único agente resolver (VIRUEL, 2011). Além disso, Wooldridge e Jennings (1995) defendem que SMAs parecem trabalhar como uma metáfora natural para entender e construir uma ampla gama de sistemas sociológicos artificiais.

Segundo Wooldridge (2009), o campo de SMAs ganhou reconhecimento na década de 1990. Esse reconhecimento, na época, pode ser atribuído, pelo menos em parte, à crença de que os agentes foram vistos como um paradigma de software apropriado para explorar as possibilidades apresentadas pelos sistemas distribuídos massivos abertos, como a Internet. No entanto, Botti et al. (2019) explica que, com o atual avanço da tecnologia, os aplicativos baseados em agentes estão se tornando um padrão em uma grande variedade de domínios, como *e-commerce*, logística, gerenciamento de cadeia de suprimentos, telecomunicações, na área de saúde e na manufatura. Outra razão para o amplo interesse em sistemas multiagentes é que esses sistemas são vistos como uma tecnologia e uma ferramenta que auxilia na análise e desenvolvimento de novos modelos e teorias em sistemas distribuídos de grande escala ou em sistemas centrados no homem.

De acordo com Dorri, Kanhere e Jurdak (2018), sistemas multiagentes tem recebido grande atenção de estudiosos em diferentes disciplinas, incluindo ciência da computação e engenharia civil, como um meio de resolver problemas complexos, subdividindo-os em tarefas menores. O uso de SMAs está presente em diversas aplicações, incluindo modelagem de sistemas complexos, redes inteligentes e redes de computadores.

Contudo, o desenvolvimento de SMAs apresenta sua própria complexidade e seus próprios desafios, o que demonstra a necessidade de aplicar práticas específicas de engenharia de software no desenvolvimento desses sistemas. Isso levou ao surgimento da AOSE (Engenharia de Software Orientada por Agente), uma área que mistura características da Engenharia de Software e da Inteligência Artificial. Segundo Guedes e Vicari (2010) a área de AOSE tem como objetivo adaptar as práticas de engenharia de software especificamente para o desenvolvimento de sistemas baseados em agentes.

Uma das várias subáreas da Engenharia de Software é a ER, que visa elicitar, analisar, especificar e validar os requisitos dos sistemas, de modo a permitir a compreensão correta do que realmente um sistema deve fazer (BOURQUE; FAIRLEY et al., 2014). Apesar de existirem processos de desenvolvimento que abordam ER para SMAs, verificamos diversas lacunas em suas abordagens e pontos possíveis de melhorias. Dentre essas

lacunas, podemos destacar que nenhuma metodologia verificada, apresenta uma etapa de Engenharia de Requisitos com foco no Modelo BDI, suportando todas as etapas de ER definidas no *Software Engineering Body of Knowledge* ou Guia do Conhecimento da Engenharia de Software (SWEBOK).

Segundo Herzig et al. (2017), os conceitos de crença e objetivo desempenham um papel central na concepção e implementação de agentes autônomos. Esse conceito, considera atitudes mentais fundamentais dos agentes, em que as crenças se adaptam às verdades do ambiente, enquanto nas intenções, os agentes tentam fazer o ambiente corresponder aos seus objetivos.

Alzetta et al. (2020) afirmam que esse modelo representa uma das abordagens mais reconhecidas para integrar as habilidades cognitivas desejadas em agentes autônomos e facilita a descrição da relação causa-efeito necessária para que um agente alcance seus objetivos. Além disso, Larsen (2018) afirma que esse modelo permite aos agentes um comportamento mais complexo do que os modelos reativos, sem a sobrecarga computacional das arquiteturas cognitivas, tornando mais fácil especificar o conhecimento.

1.1 Motivação

Segundo Koscianski e Soares (2007) as principais falhas verificadas no desenvolvimento de software são relativas a problemas na especificação de requisitos, devido principalmente às dificuldades no entendimento das necessidades do usuário. Portanto, realizar corretamente o levantamento e administração de requisitos é essencial para garantir a qualidade do software.

Nesse sentido, Pressman e Maxim (2016) nos ensina que a ER fornece os instrumentos apropriados para compreender o desejo do cliente, analisando as necessidades do mesmo, negociando soluções razoáveis e viáveis, validando as especificações e gerenciando as necessidades durante o ciclo de desenvolvimento do software.

Isso não poderia ser diferente em relação a sistemas multiagentes, que possuem requisitos particulares, além dos compartilhados com outros tipos de sistemas. Dessa maneira, visando identificar processos que atuem na necessidade de uma ER própria para SMAs, realizamos uma revisão sistemática de literatura buscando verificar como estes processos atuam com relação ao modelo BDI e as subáreas de ER definidas no SWEBOK.

As lacunas encontradas na revisão sistemática e que se tornaram a base desta proposta foram:

- necessidade de atuação na elicitação de requisitos, voltada a interação com o stakeholder, oferecendo suporte ao modelo BDI;
- esse mesmo processo deve auxiliar na identificação de requisitos específicos para SMAs, como percepções, crenças, objetivos, planos e ações.

Considerando as lacunas encontradas nos processos para desenvolvimento de SMAs com relação à cobertura das etapas da ER e suporte ao modelo BDI, percebemos a necessidade de propor um processo de ER específico para SMAs com todas as fases da ER bem definidas e que suporte inteiramente o modelo BDI.

1.2 Objetivos e Cronograma das atividades

Esta Seção tem como propósito apresentar os objetivos desse trabalho e o cronograma das atividades. Dessa forma, a Seção 1.2.1 apresenta o objetivo geral desse trabalho. A Seção 1.2.2 apresenta os objetivos específicos presentes no trabalho e por ultimo a Seção ?? apresenta o cronograma delimitado para o trabalho.

1.2.1 Objetivo Geral

O objetivo geral desse trabalho é propor um processo de engenharia de requisitos para sistemas multiagentes, visando garantir a correta identificação dos requisitos necessários para esse tipo de sistema, com ênfase na especificação correta dos requisitos referente ao modelo BDI.

1.2.2 Objetivos Específicos

- Realizar uma revisão sistemática de literatura que identifique os processos de ER para SMA, sua cobertura com relação as subáreas de ER definidas no SWEBOK e seu suporte ao modelo BDI;
- Propor um processo de engenharia de requisitos específico para sistemas multiagentes;
- Validar o processo proposto executando na segunda versão do sistema Heráclito, atualmente em desenvolvimento.

1.3 Contribuições

As principais contribuições desse trabalho são:

- apresentar uma análise dos processos que trabalham com ER para SMA;
- oferecer um processo de ER para SMA com base no modelo BDI que contemple todas as subáreas definidas no *SWEBOK* (BOURQUE; FAIRLEY et al., 2014).

1.4 Metodologia Utilizada no Trabalho

No início de nossa pesquisa foi realizada uma revisão sistemática de literatura que pode ser vista no capítulo 3, Seção 3.1. Essa revisão teve como objetivo determinar possíveis lacunas nos processos de desenvolvimento que cubram ER para SMA.

Com os resultados dessa revisão determinamos a necessidade de propor um processo de ER específico para SMAs.

Desse modo, delimitamos pontos a serem cobertos pelo processo e identificamos um ciclo de vida geral para o mesmo. Após a identificação do ciclo de vida geral identificamos a documentação utilizada no mesmo com base na extensão do diagrama de casos de uso UML voltado para SMAs (GUEDES, 2012).

Por fim, buscamos avaliar o processo, executando a ER na segunda versão do sistema Heráclito. O sistema Heráclito é um sistema multiagente voltado ao aprendizado interativo e dialético com foco no ensino de dedução natural da lógica proposicional (GALAFASSI et al., 2019). Atualmente está sendo desenvolvida a segunda versão desse sistema.

1.5 Organização do Trabalho

O restante do documento está organizado da seguinte maneira. O Capítulo 2 apresenta os conceitos necessários para o entendimento da solução proposta. Nesse Capítulo são descritos os conceitos necessários para entender um sistema multiagente e seus requisitos particulares. Em seguida, o Capítulo 3 apresenta a revisão sistemática de literatura realizada, onde foram analisados os processos de desenvolvimento para SMAs que suportam a etapa de ER. O Capítulo 4, apresenta o processo desenvolvido, juntamente com a execução do processo e os resultados obtidos. E por último o capítulo 5 relata as considerações finais sobre a execução do trabalho.

2 FUNDAMENTAÇÃO TEÓRICA

Esse capítulo apresenta os conceitos utilizados para desenvolver esse trabalho. Na Seção 2.1 são apresentados os conceitos de agente e sistema multiagente. A Seção 2.1.1 apresenta as características de agentes suportadas pelo processo apresentado nesse trabalho. Na Seção 2.2 é abordada a engenharia de requisitos. A Seção 2.2.5 apresenta uma explicação sobre processos de engenharia de requisitos. E por ultimo, a Seção 2.3 apresenta a forma que serão documentados os casos de uso internos dos agentes, método que servira para documentar os cenários internos dos agentes.

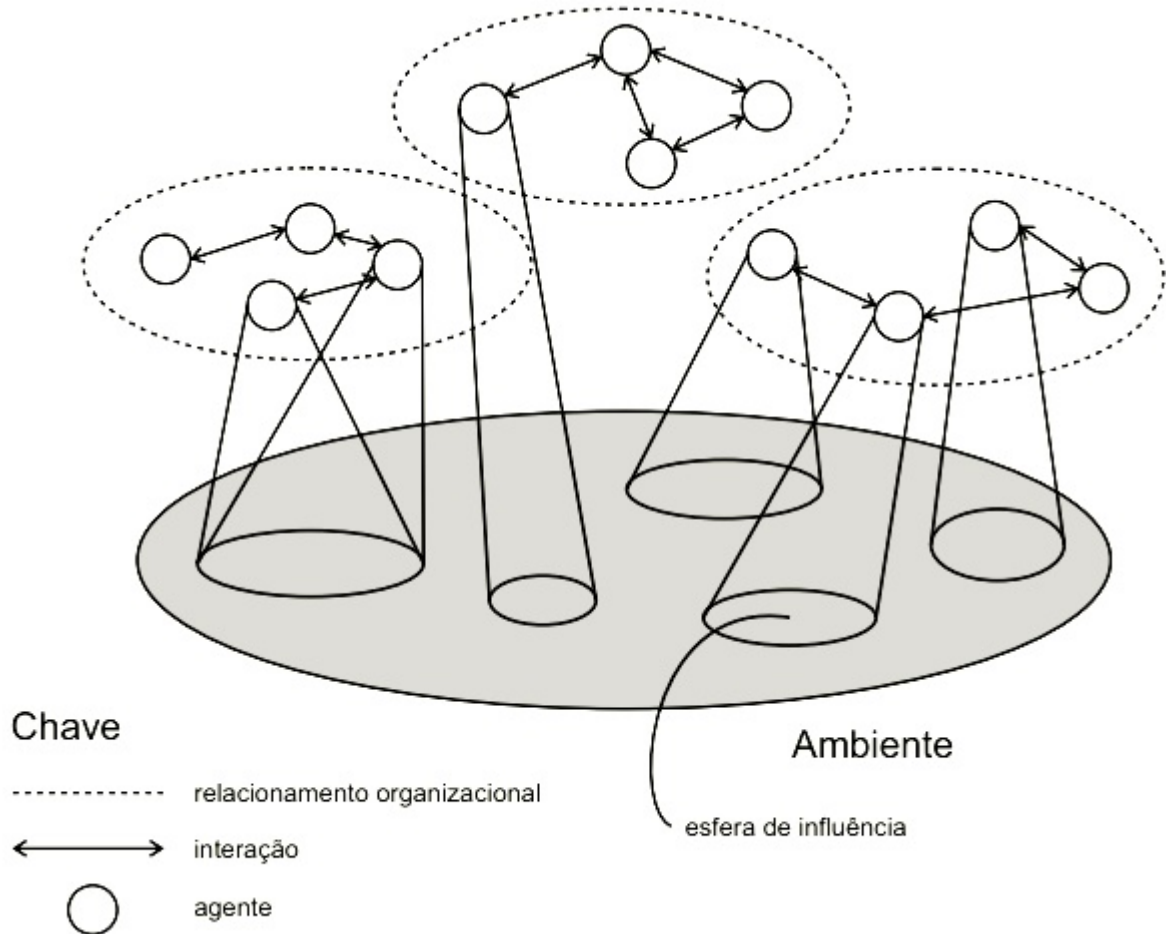
2.1 Sistemas Multiagentes

Para explicar o que é um SMA primeiro precisamos definir o que são agentes de software. Nesse trabalho vamos usar a definição de Wooldridge e Jennings (1995), que estabelece que um agente de software é um sistema de computador baseado em software que tenha as seguintes características:

- **autonomia:** os agentes operam sem a intervenção direta de humanos, e possuem algum controle sobre suas ações e estados internos;
- **habilidade social:** agentes interagem com outros agentes (e possivelmente humanos) por meio de algum tipo de linguagem ou protocolo de comunicação de agentes;
- **proatividade:** os chamados **agentes cognitivos** não agem simplesmente em resposta ao seu ambiente, eles são capazes de exibir um comportamento direcionado aos objetivos e tomar a iniciativa quando acreditam que eles possam ser atingidos;
- **reatividade:** agentes percebem seu ambiente (que pode ser o mundo físico, uma coleção de outros agentes, a Internet, ou talvez todos esses combinados), e podem responder às mudanças que ocorrem nele. Agentes que só reagem ao ambiente são chamados de **agentes reativos**.

Segundo Wooldridge (2009) um SMA consiste em um conjunto de agentes que interagem uns com os outros. A Figura 1 ilustra a estrutura típica de um SMA. O sistema contém vários agentes, que interagem entre si. Os agentes são capazes de agir em um ambiente, sendo que agentes diferentes têm diferentes "esferas de influência", podendo ter controle sobre, ou pelo menos podem influenciar, diferentes partes do ambiente. Por fim, os agentes também podem ser ligados por relações. Um exemplo disso pode ser uma relação de "poder", em que um agente pode atuar como "chefe" de outro agente.

Figura 1 – Estrutura típica de um sistema multiagente.



Fonte: (WOOLDRIDGE, 2009) pagina 106

2.1.1 Características de Agentes

Esta Seção tem como objetivo apresentar e detalhar todas as características de agentes que vão ser trabalhadas no processo.

2.1.2 Modelo BDI

Segundo Rao, Georgeff et al. (1995) o modelo *BDI*, é um modelo de software desenvolvido para programar agentes inteligentes caracterizado por incluir crenças, desejos e intenções na arquitetura do agente. A seguir detalharemos essas características:

- **crenças:** representam o estado de informações que o agente possui, em outras palavras, suas crenças sobre o mundo (incluindo ele mesmo e outros agentes);

- **desejos:** representam o estado motivacional do agente. Eles representam objetivos ou situações que o agente gostaria de atingir;
- **intenções:** representam desejos com os quais o agente tem, até certo ponto, comprometimento na medida em que ele acredita que esses possam ser atingidos.

Com relação aos objetivos, é definido que um objetivo é o intermédio entre um desejo e uma intenção. Dessa forma, enquanto a crença de um agente não for verdadeira, ou seja, enquanto o agente não acreditar que o objetivo pode ser atingido, ele possui apenas o desejo de atingi-lo, mas não age para isso. A partir do momento em que a crença (ou crenças) tornar-se verdadeira, ou seja, o agente passar a acreditar que o objetivo possa ser atingido, esse objetivo passa a ser uma intenção e o agente passa a agir para atingi-lo, muitas vezes por meio da execução de um plano associado ao objetivo.

2.1.3 Papel de Agente

Cossentino e Seidita (2014a) definem papel de agente como uma parte do comportamento social de um agente que é caracterizado por possuir um objetivo e/ou fornecer um serviço. O objetivo de cada papel é contribuir para o cumprimento de uma parte dos requisitos da organização em que ele está contido.

2.1.4 Percepções do Agente

Segundo Russell e Norvig (2016) a percepção fornece aos agentes informações sobre o mundo em que eles habitam. A percepção é causada por um ou mais sensores. Um sensor é qualquer coisa que possa alterar o estado computacional do agente em resposta a uma mudança no estado do ambiente. Um sensor pode ser tão simples quanto um bit que representa se um interruptor está ligado ou desligado, ou tão complexo quanto a retina do olho humano, a qual contém mais de cem milhões de elementos fotossensíveis.

2.1.5 Cenários de Agente

Segundo Choren e Lucena (2005) um cenário ilustra um comportamento do agente (sequência de ações) enquanto ele pretende atingir um objetivo em um contexto específico (um estado específico do sistema), ou seja, mostra a realização do objetivo. Pode haver contextos usuais, que são contextos que mostram algumas execuções usuais de um Agente, e contextos variantes, que são contextos que requerem alguma adaptação do Agente (execução de outras ações).

Cossentino et al. (2014a) descreve cenários como uma sequência de interações de função, que atende a um, ou parte de um requisito.

2.1.6 Plano de Agente

Russell e Norvig (2016) definem um plano de agente como uma forma do agente atingir seus objetivos. Um plano é composto por ações que podem causar transições nos estados do ambiente.

Segundo García-Ojeda e Deloach (2014), um plano é responsável por capturar como um agente pode atingir um tipo específico de objetivo usando um conjunto de ações (que inclui o envio e recebimento de mensagens).

Cossentino e Seidita (2014a) expressa que o comportamento de um agente é especificado em seu plano. O plano é a descrição de como combinar e ordenar tarefas e interações para cumprir total ou parcialmente um requisito.

2.1.7 Ação do Agente

Segundo Choren e Lucena (2005) uma ação é um ato computacional que resulta em uma alteração no estado do ambiente. Podem haver dois tipos de ação, sendo elas:

- **ação direta:** ação realizada enquanto o agente participa de um cenário para atingir um objetivo;
- **ação adaptativa:** ação em que, no meio de um determinado cenário, é necessário que o agente se adapte e essa adaptação requer alguma forma de raciocínio.

2.1.8 Organização

Segundo Choren e Lucena (2005) uma Organização é um grupo de (um ou mais) agentes trabalhando juntos para executar alguma função ou entregar um serviço. Em um SMA podem haver diversas organizações.

Cossentino et al. (2014a) define uma organização como uma coleção de papéis que participam de padrões sistemáticos institucionalizados de interações com outros papéis em um contexto comum.

2.2 Engenharia de Requisitos

Segundo o *Software Engineering Body of Knowledge* ou Guia do Conhecimento da Engenharia de Software (SWEBOK) Bourque, Fairley et al. (2014) os requisitos expressam as necessidades e restrições colocadas sobre o produto de software que contribuem para a solução de algum problema do mundo real.

A ER fornece os instrumentos apropriados para compreender o desejo do cliente, analisando as necessidades do mesmo, negociando soluções razoáveis e viáveis, validando as especificações e gerenciando as necessidades à medida em que o desenvolvimento do software ocorre (PRESSMAN; MAXIM, 2016).

Outro ponto importante sobre requisitos de software é que compreender os requisitos é uma das tarefas mais críticas enfrentadas na engenharia de software. Requisitos de um sistema são descrições do que o sistema deve fazer, os serviços que oferece e as restrições sobre seu funcionamento e sua implementação (KOTONYA; SOMMERVILLE, 1998).

As próximas seções abordam as etapas da engenharia de requisitos do *SWEBOK* Bourque, Fairley et al. (2014), que servem como base inicial para esse trabalho.

2.2.1 Elicitação de Requisitos

A elicitação de requisitos, área que envolve a captura, descoberta ou aquisição dos requisitos de software, é o primeiro estágio para o entendimento do problema (BOURQUE; FAIRLEY et al., 2014). A elicitação de requisitos está dividida da seguinte forma:

- **Fontes de requisitos:** Fonte de onde os requisitos serão elicitados. As principais fontes de requisitos são:
 - **stakeholders:** Todos aqueles que têm alguma relação direta sobre os requisitos do sistema;
 - **documentos:** Documentos que podem ter alguma informação importante sobre o sistema, como documentação sobre normas, leis e relatório de falhas de sistemas legados;
 - **sistemas em operação:** O *stakeholder* pode tomar como base um sistema legado ou até mesmo um sistema concorrente para dar sugestões para a elicitação de requisitos.

- **Técnicas de elicitação:** Muitas vezes os *stakeholders* podem ter dificuldades em descrever o que desejam ou omitir informações. Para isto existem métodos ou ferramentas para auxiliar na extração de requisitos, alguns deles são:
 - **entrevistas:** A entrevista é uma das técnicas tradicionais mais simples de utilizar e que produz bons resultados na fase inicial de obtenção de dados. Segundo Kotonya e Sommerville (1998), entrevistas oferecem uma solução eficaz para coletar grandes quantidades de dados rapidamente;
 - **observação:** o engenheiro de software acompanha o especialista ou os usuários do sistema no próprio local de trabalho, observando e documentando os processos e procedimentos executados;
 - **prototipagem:** Esta técnica disponibiliza uma versão gráfica do produto para o cliente, podendo assim obter maiores resultados quanto à conformidade dos requisitos;

- **brainstorming (tempestade cerebral):** Usada para coletar grande quantidade de ideias em um espaço curto de tempo. O problema é discutido entre os participantes de diferentes grupos de interesse.

2.2.2 Análise de Requisitos

Procura detectar e resolver conflitos entre requisitos, descobrir as fronteiras do software e como esse interage com o ambiente (BOURQUE; FAIRLEY et al., 2014). A análise de requisitos está dividida da seguinte forma:

- **classificação dos Requisitos:** classifica os requisitos em funcionais, não funcionais, por prioridade, estabilidade, etc;
- **modelagem conceitual:** Cria modelos para auxiliar na compreensão do problema. Isto já é uma forma de especificação de requisitos inicial;
- **negociação de requisitos** Resolve conflitos com relação aos requisitos.

2.2.3 Especificação de Requisitos

A principal atividade da especificação de requisitos é produzir a documentação do sistema, que poderá ser sistematicamente revisada, validada e aprovada, estabelecendo os componentes de software (BOURQUE; FAIRLEY et al., 2014). Nessa etapa os requisitos devem estar claros, inequívocos, completos, consistentes e ser de fácil compreensão. O ideal é que os requisitos de usuário sejam descritos em linguagem natural, com tabelas simples e diagramas intuitivos. (KOTONYA; SOMMERVILLE, 1998).

2.2.4 Validação de Requisitos

A validação de requisitos é responsável pela verificação dos documentos, buscando encontrar omissões, conflitos e ambiguidades, visando assegurar que os requisitos obtenham maior qualidade. Os requisitos devem ser escritos de uma maneira que deixe o menor espaço possível para erros de interpretação (BOURQUE; FAIRLEY et al., 2014). Existem diversas técnicas para validação de requisitos, algumas delas são:

- **revisão dos requisitos:** meio mais comum de validação, efetuado geralmente por meio do uso de listas de verificação. Eles podem incluir um conjunto padrão de elementos de qualidade que o analista de negócios (ou outros revisores) usam para validar os requisitos, ou ser desenvolvidos especificamente para capturar questões relacionadas ao projeto (BRENNAN et al., 2009).
- **prototipação:** Cria protótipos para validar a visão do desenvolvedor com o cliente;
- **validação de modelos:** Valida conformidade e padrões nos modelos;

- **testes de aceitação:** Projeta os testes de requisitos de software para determinar se os requisitos de software foram atendidos.

É importante destacar que a validação de requisitos é uma etapa paralela, ou seja, ela não necessariamente precisa esperar a conclusão das etapas anteriores, podendo ser aplicada à medida em que artefatos vão sendo produzidos.

2.2.5 Processo de requisitos de software

O *SWEBOK* (BOURQUE; FAIRLEY et al., 2014) descreve um processo de requisitos de software como um planejamento de requisitos, sendo dividido em quatro subáreas:

- **modelos de processos:** trata da forma que as atividades de análise de levantamento de requisitos, especificação e validação estão configurados para diferentes tipos de projetos e restrições. Também trabalha com atividades que contribuem para o processo de requisitos, tais como marketing e estudos de viabilidade;
- **atores dos processos:** os papéis das pessoas que atuam no processo de requisitos;
- **processo de suporte e gestão:** seu objetivo principal é fazer uma conexão entre as atividades do processo, identificado na subárea modelos de processo, e as questões de custos, recursos humanos, treinamento e ferramentas;
- **processos de qualidade e melhoria:** se preocupa com a qualidade do processo de requisitos com relação a custo e satisfação do cliente.

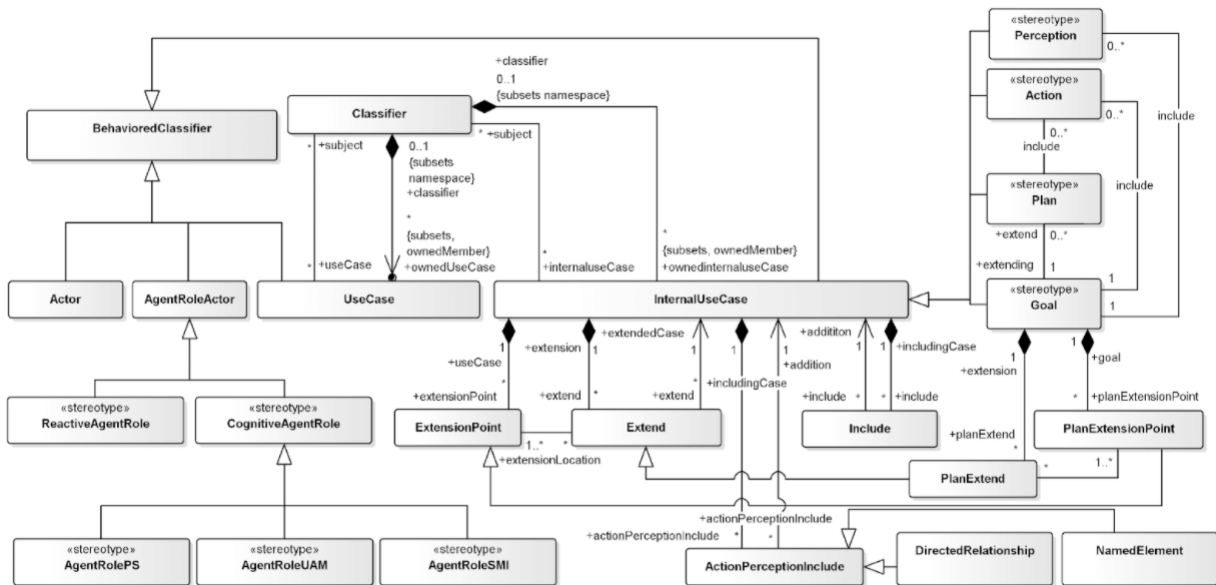
2.3 Casos de Uso Internos

Os casos de uso fornecem uma maneira estruturada de capturar os requisitos comportamentais de um sistema. Casos de uso tem como objetivo ajudar a responder a algumas questões fundamentais: O que os usuários do sistema estão tentando fazer? Qual é a experiência do usuário? Uma quantidade surpreendente do que o seu software faz é ditada pela maneira como os usuários devem interagir com ele (ROSENBERG; SCOTT, 1999).

Em relação à modelagem de requisitos, muitos autores recomendam empregar o diagrama de casos de uso UML (BERENBACH et al., 2009), (REGNELL, 1999), (SOMMERVILLE, 2011). Diagramas de casos de uso identificam assuntos, casos de uso e atores como conceitos-chave do sistema. Um assunto representa o sistema modelado. Os casos de uso representam os requisitos externos de um assunto. Cada caso de uso especifica uma funcionalidade exercida pelos usuários. Atores representam entidades externas que interagem com o assunto e acessam suas funcionalidades (UML, 2019).

Nesse trabalho utilizamos uma documentação e modelagem de casos de uso próprias para SMAs. Os modelos de casos de uso internos utilizados foram produzidos a partir do metamodelo de Guedes. A versão atualizada do Metamodelo de Guedes pode ser vista na Figura 2.

Figura 2 – Metamodelo de Guedes



Fonte: Guedes (2012)

O Metamodelo de Guedes cria a metaclassa *AgentRoleActor* derivada da mesma metaclassa da qual a metaclassa *Actor* é derivada. A partir do *AgentRoleActor*, as metaclassas *ReactiveAgentRole* e *CognitiveAgentRole* são derivadas para representar papéis interpretados por agentes reativos e cognitivos. O metamodelo também especializa a metaclassa *CognitiveAgentRole* derivando as metaclassas *AgentRolePS*, *AgentRoleUAM* e *AgentRoleSMI*. Essas metaclassas foram inspiradas em Vicari e Gluz (2007), que sugeriram o uso dos agentes especialistas de Resolução de Problemas, Modelagem de Usuários e Agentes e Interações Mediadas Sociais.

Também foi criada a metaclassa *InternalUseCase*. Segundo Guedes (2012) um caso de uso interno (Internal Use Case) se difere do conceito de caso de uso da UML uma vez que esse é voltado para a representação dos requisitos externos do sistema, enquanto os casos de uso internos foram concebidos para representar os requisitos internos de um SMA. Em um SMA são desenvolvidas funcionalidades internas no sistema, enquanto os usuários externos normalmente não têm ideia de quais são essas funcionalidades ou mesmo que eles existem.

As metaclassas *Perception* e *Action*, foram criadas a partir da metaclassa *InternalUseCase* para modelar casos de uso internos contendo os passos necessários para que

um agente perceba ou execute uma dada ação externa, como enviar uma mensagem para outro agente.

A metaclassa `Goal` foi criada a partir da metaclassa `InternalUseCase` para representar os objetivos que um agente deseja alcançar. A documentação de um caso de uso interno com o estereótipo `Goal` conterà uma descrição do desejo de um agente e as condições para que esse desejo se torne uma intenção.

Para que um objetivo seja alcançado, normalmente deve haver um ou mais planos para atingir esse objetivo. Considerando que um plano associado a um objetivo só será executado se o objetivo tornar-se uma intenção, foi derivado uma quarta metaclassa a partir da metaclassa `InternalUseCase`, denominada `Plan`, para identificar os planos associados às metas de uma função de agente.

Como um plano só é disparado depois que alguma condição é satisfeita, a metaclassa `PlanExtend` foi derivada a partir da metaclassa `Extend` e associada à metaclassa `Goal`. A associação `PlanExtend` permite que o comportamento de um plano seja inserido no comportamento de um objetivo quando uma condição for satisfeita, em geral quando uma percepção for verdadeira.

Por último, foi criada a metaclassa `ActionPerceptionInclude`, derivada a partir das mesmas metaclasses que a metaclassa `Include` foi derivada. Esta metaclassa contém as mesmas características da metaclassa `Include`, sem a restrição de que a metaclassa deve ser usada quando há trechos de comportamento exatamente iguais no comportamento de dois ou mais casos de uso. A retirada dessa restrição permite que ao modelar um objetivo, seja possível associar um ou mais `Perceptions` por meio da associação `actionPerceptionInclude`.

3 REVISÃO SISTEMÁTICA DE LITERATURA

Esse Capítulo tem como objetivo apresentar os estudos que serviram como base ou referência para a realização desse trabalho. Para determinar esses estudos foi realizado uma revisão sistemática de literatura com o objetivo de analisar os processos de desenvolvimento para SMAs que suportem a etapa de ER.

Uma Revisão Sistemática da Literatura (RSL) é uma técnica de pesquisa cujo objetivo é identificar, selecionar, avaliar, interpretar e resumir os estudos disponíveis considerados relevantes para o tema da pesquisa ou fenômeno de interesse (KITCHENHAM; CHARTERS, 2007). Essa técnica procura estudos primários relacionados ao tema e fornece uma síntese mais profunda sobre os dados obtidos com esses estudos (KITCHENHAM; BRERETON, 2013).

Uma SLR tem como base um protocolo previamente definido, que formaliza a execução da SLR, começando pela estipulação das questões de pesquisa, passando pelo estabelecimento dos critérios de inclusão e exclusão dos estudos, selecionando a base digital para a extração de trabalhos relacionados à palavras-chave aplicadas durante a pesquisa nessas bases e concluindo com a definição de como os resultados serão apresentados (BIOLCHINI et al., 2005).

Dessa forma a Seção 3.1 tem como objetivo apresentar o protocolo utilizado na Revisão Sistemática, juntamente de alguns decisões tomadas durante a revisão. A Seção 3.2 apresenta os resultados desta revisão, mostrando um breve resumo de cada processo analisado. Por ultimo a Seção 3.3, apresenta as decisões tomadas para mitigar as possíveis ameaças a confiabilidade desta revisão.

3.1 Apresentação da Revisão Sistemática

A seguinte revisão teve como objetivo estabelecer o estado da arte dos processos/metodologias para o desenvolvimento de SMA que suportam, de alguma forma, a engenharia de requisitos para esse tipo de sistema. Nosso principal interesse é sobre como esses processos representam características do modelo BDI e a cobertura dos mesmos com relação as subáreas definidas no SWEBOK.

Tendo isso em vista, definiu-se um protocolo de pesquisa para revisão sistemática contendo perguntas de pesquisa, critérios para identificar e selecionar estudos, critérios para inclusão ou exclusão de estudos, método de avaliação da qualidade, estratégia de extração de dados e também como conduzir a revisão.

3.1.1 Questões de Pesquisa

Para esta revisão sistemática, foram definidas quatro questões de pesquisa. A primeira questão de pesquisa (QP1) tem como objetivo identificar quais metodologias/processos suportam a engenharia de requisitos para SMAs.

A segunda questão de pesquisa (QP2) foi definida para identificar a cobertura da engenharia de requisitos por essas metodologias. Acreditamos que, com essa questão, possamos descobrir possíveis lacunas na área e que isso permitirá pesquisas futuras.

A terceira questão de pesquisa (QP3) tem como objetivo verificar quais metodologias suportam o modelo BDI. Como foi explicado no Capítulo 2, Seção 2.1.2, esse é um modelo consolidado no desenvolvimento de SMAs, assim acreditamos que a aplicação desse modelo agrega uma melhor confiabilidade ao usar as metodologias que o suportam.

Finalmente, a quarta questão de pesquisa (QP4) tem como objetivo mostrar uma visão mais ampla das necessidades da área e focar nos pontos que podem ser abordados em trabalhos futuros.

As quatro questões de pesquisa estão listadas abaixo:

- **QP1:** Quais metodologias para o desenvolvimento de SMAs suportam um ciclo de vida de engenharia de requisitos específicos (RE) para esse tipo de sistema?
- **QP2:** Qual é a cobertura da engenharia de requisitos por essas metodologias, tomando como base as subáreas definidas pelo SWEBOK (BOURQUE; FAIRLEY et al., 2014)?
- **QP3:** Quais dessas metodologias se concentram no modelo BDI durante a engenharia de requisitos?
- **QP4:** Quais são as lacunas existentes nas metodologias que suportam ER para SMAs?

3.1.2 Identificação e seleção de estudos primários

Para recuperar trabalhos relevantes para esse estudo, construímos uma string genérica contendo um conjunto de palavras-chave com base nas perguntas da pesquisa. Essa string foi adaptada às particularidades de cada base bibliográfica.

Para realizar esta revisão sistemática, foram utilizadas bases bibliográficas que (i) possuem um mecanismo de busca baseado na web; (ii) possuem um mecanismo capaz de usar palavras-chave; (iii) conter documentos da área de computação científica; e (iv) suas bases de dados são atualizadas regularmente.

Além disso, foi incluído o livro (COSSENTINO; SEIDITA, 2014a) sobre metodologias para sistemas multiagentes, além de outros estudos clássicos. Esses estudos foram selecionados manualmente por um especialista na área, pois consideramos que eles não seriam selecionados na string de pesquisa, pois não apresentavam em seu título, resumo e tópicos palavras-chave relacionados à engenharia de requisitos, embora seus ciclos de vida abranjam o área de engenharia de requisitos, esses processos não citavam ER nestes tópicos pois trata-se de processos que abrangem outras etapas além da Engenharia de Requisitos.

Na Tabela 1, pode ser visto as bases bibliográficas utilizadas, as strings adaptadas para cada base e os filtros selecionados.

Além da pesquisa por string, aplicamos filtros manuais nas bases bibliográficas. Os filtros podem ser observados na Tabela 1. Foi necessário aplicar esses filtros manuais porque, em algumas bases, os resultados obtidos foram altos e muitos dos estudos retornados estavam fora do escopo, isso foi resultado na nossa pesquisa ser muito abrangente, visando encontrar o máximo de processos possíveis.

Tabela 1 – Strings e filtros aplicados em cada base bibliográfica

Base	string	Filter
ACM Library	("multiagent" OR "multi-agent" OR "multi agent" OR "agent-based" OR "agent society") AND ("Methodology" OR "method" OR "process") AND ("requirements engineering" OR "requirements elicitation" OR "requirements modeling" OR "requirements analysis" OR "requirements specification")	Title/Abstract/keywords
Engineering Village	("multiagent" OR "multi-agent" OR "multi agent" OR "agent-based" OR "agent society") AND ("Methodology" OR "method" OR "process") AND ("requirements engineering" OR "requirements elicitation" OR "requirements modeling" OR "requirements analysis" OR "requirements specification")	Subject/Title/Abstract
IEEE Xplore	("multiagent" OR "multi-agent OR "multi agent" OR "agent-based" OR "agent society") AND ("Methodology" OR "method" OR "process") AND ("requirements engineering" OR "requirements elicitation" OR "requirements modeling" OR "requirements analysis" OR "requirements specification")	All metadata, filters suggested by the base software agents and multi-agent systems
Science Direct	("multiagent" OR "multi-agent" OR "agent-based") AND ("Methodology" OR "method" OR "process") AND ("requirements engineering" OR "requirements elicitation" OR "requirements modeling" OR "requirements analysis" OR "requirements specification")	Title/Abstract/keywords and commands "multiagent" OR multi-agent OR "agent-based"
Scopus	("multiagent" OR "multi-agent" OR "multi agent" OR "agent-based" OR "agent society") AND ("Methodology" OR "method" OR "process") AND ("requirements engineering" OR "requirements elicitation" OR "requirements modeling" OR "requirements analysis" OR "requirements specification")	Title/Abstract/keywords
Springer Link	((("multiagent" OR "multi-agent" OR "multi agent" OR "agent-based" OR "agent society") AND ("Methodology" OR "method" OR "process") AND ("requirements engineering" OR "requirements elicitation" OR "requirements modeling" OR "requirements analysis" OR "requirements specification"))	Filter of the area: Computer science; Filter of the subarea: Software Engineering and Artificial intelligence

Fonte: O Autor

3.1.3 Critérios de inclusão e exclusão

O critério de seleção tem como objetivo identificar os estudos primários que fornecem conteúdo para responder às perguntas da pesquisa. Assim, primeiramente, os estudos foram analisados com base no título, resumo e palavras-chave. Se ainda houvesse dúvidas sobre a classificação final de um estudo em relação aos critérios de inclusão ou exclusão, um especialista seria consultado. Esses critérios estão descritos na tabela 2.

Tabela 2 – Critérios de inclusão e exclusão.

Critério	ID	Descrição
	IC1	O estudo apresenta uma metodologia ou uma extensão de uma metodologia para sistemas multi-agente que contemple pelo menos uma das subáreas de engenharia de requisitos definidas no SWEBOK?
	EC1	Estudos não escritos em inglês.
	EC2	Estudos que abrangem uma metodologia já incluída em um trabalho mais recente.
	EC3	Estudos que não são um artigo ou um capítulo de livro.
	EC4	Estudos com menos de 6 páginas.
	EC5	Estudos introdutórios em edições especiais (workshops ou conferências).
	EC6	Estudos que se resumem a um estudo de caso ou avaliação de metodologia.
	EC7	Estudos que se resumem a uma comparação de metodologias.
	EC8	Estudos que não apresentam uma metodologia. (ou extensão de uma metodologia) para sistemas multi-agente que contemplem pelo menos uma das subáreas de ER descritas no SWEBOK.
	EC9	Estudos que se concentram em outras áreas da Engenharia de Software.
	EC10	Estudos que se resumem ao desenvolvimento de um sistema.
	EC11	Estudos que se resumem ao desenvolvimento de um sistema. de uma metodologia criada apenas para um tipo específico de aplicação.

Fonte: O Autor

3.1.4 Avaliação de qualidade dos estudos

Definimos dois critérios de qualidade para avaliar a relevância dos estudos para o escopo desta pesquisa. Esses critérios não foram utilizados para exclusão de estudos, apenas para a classificação dos estudos mais relevantes. A seguir, descrevemos os dois critérios qualitativos e a pontuação atribuída a cada critério definido.

- **QC1:** O trabalho suporta o modelo BDI?
 - Sim(Y): O trabalho suporta totalmente o modelo BDI.
 - Parcialmente(P): O trabalho suporta pelo menos um dos recursos do modelo BDI.
 - Não(N): O trabalho não suporta o modelo BDI.
- **QC2:** O trabalho aplica algum estudo empírico (experimento, estudo de caso etc.)?
 - Sim(Y): O estudo aplica algum estudo empírico.
 - Não(N): O estudo não aplica algum estudo empírico.

Para estabelecer um índice geral de qualidade dos estudos selecionados, atribuímos escores a cada critério definido, onde Sim(Y) corresponde a 1, Parcialmente(P) 0,5 e Não(N) 0. A tabela 3 mostra a pontuação de cada estudo selecionado. Percebemos que apenas três estudos (S17, S18 e S22) atingiram a classificação máxima de 2 escores.

Tabela 3 – Índices de Qualidade dos Estudos.

ID	Estudo	QC1	QC2	Total
S1	(ULFAT-BUNYADI; MOHAMMADI; HEISEL, 2018)	0.50	0.00	0.50
S2	(ABUSHARK et al., 2016)	0.50	1.00	1.50
S3	(RIBINO et al., 2013)	1.00	0.00	1.00
S4	(LIU et al., 2011)	0.00	1.00	1.00
S5	(ARGENTE; BOTTI; JULIÁN, 2009)	0.50	0.00	0.50
S6	(SEN; HEMACHANDRAN, 2010)	0.50	1.00	1.50
S7	(BLANES; INSFRAN; ABRAHÃO, 2009)	0.50	1.00	1.50
S8	(HUIYING; ZHI, 2009)	0.50	0.00	0.50
S9	(FUENTES-FERNÁNDEZ; GÓMEZ-SANZ; PAVÓN, 2009)	0.50	1.00	1.50
S10	(RODRIGUEZ et al., 2009)	0.50	1.00	1.50
S11	(BRYL et al., 2008)	0.50	0.00	0.50
S12	(LEE; LEE, 2008)	0.50	1.00	1.50
S13	(RANJAN; MISRA, 2006)	0.50	0.00	0.50
S14	(LEE; LEE, 2006)	0.50	0.00	0.50
S15	(SHEN et al., 2005)	0.50	0.00	0.50
S16	(ALONSO et al., 2004)	0.00	0.00	0.00
S17	(JO; EINHORN, 2005)	1.00	1.00	2.00
S18	(MYLOPOULOS; CASTRO; KOLP, 2013)	1.00	1.00	2.00
S19	(CYSNEIROS; YU, 2002)	1.00	0.00	1.00
S20	(LEE; LIU, 2002)	0.50	0.00	0.50
S21	(BANACH, 2010)	0.50	0.00	0.50
S22	(MORREALE et al., 2006)	1.00	1.00	2.00
S23	(MURRAY, 2003)	0.00	1.00	1.00
S24	(COSSENTINO et al., 2010)	0.50	0.00	0.50
S25	(BRESCIANI; DONZELLI, 2003)	0.50	1.00	1.50
S26	(JAIN, 2007)	0.50	1.00	1.50
S27	(Hsieh et al., 2008)	0.00	1.00	1.00
S28	(SUTCLIFFE, 2001)	0.50	1.00	0.00
S29	(Sen; Jain, 2007)	0.50	1.00	1.50
S30	(Haumer et al., 1999)	0.50	1.00	1.50
S31	(Longbing Cao et al., 2004)	0.50	1.00	1.50
S32	(Wilmann; Sterling, 2005)	0.50	0.00	0.50
S33	(Wu; Liu; Ma, 2010)	0.50	1.00	1.50
S34	(Liu; Li, 2015)	0.00	1.00	1.00
S35	(Hajer; Taieb; Raouf, 2009)	0.00	0.00	0.00
S36	(ASHAMALLA; BEYDOUN; LOW, 2017)	0.50	1.00	1.50
S37	(BOKMA et al., 1994)	0.00	0.00	0.00
S38	(HILAIRE et al., 2012)	0.50	0.00	0.50
S39	(GAUR; SONI, 2012)	0.50	1.00	1.50
S40	(PASSOS; ROSSETTI; GABRIEL, 2015)	0.50	1.00	1.50
S41	(WANG et al., 2013)	0.50	1.00	1.50
S42	(RONALD et al., 2012)	0.50	1.00	1.50
S43	(DOMANN et al., 2014)	0.00	1.00	1.00
S44	(M. et al., 2014)	0.50	0.00	0.50
S45	(COSSENTINO; SEIDITA, 2014b)	0.00	1.00	1.00
S46	(GONZÁLEZ-MORENO et al., 2014)	0.00	0.00	0.00
S47	(BONJEAN et al., 2014)	0.00	1.00	1.00
S48	(DELOACH; GARCIA-OJEDA, 2014)	0.50	0.00	0.50
S49	(PADGHAM; THANGARAJAH; WINIKOFF, 2014)	1.00	0.00	1.00
S50	(CAIRE et al., 2004)	0.50	1.00	1.50
S51	(CAO, 2015)	0.50	0.00	0.50
S52	(N., 1997)	1.00	0.00	1.00
S53	(IGLESIAS et al., 1998)	1.00	0.00	1.00
S54	(LIND, 2001)	0.00	1.00	1.00

Fonte: O Autor

Por outro lado, alguns estudos obtiveram pontuação 0 (S16, S35, S37 e S46), embora esses estudos não tenham atingido nenhum escore. Estes estudos foram mantidos porque os critérios qualitativos foram utilizados apenas para classificar os estudos, não para eliminá-los.

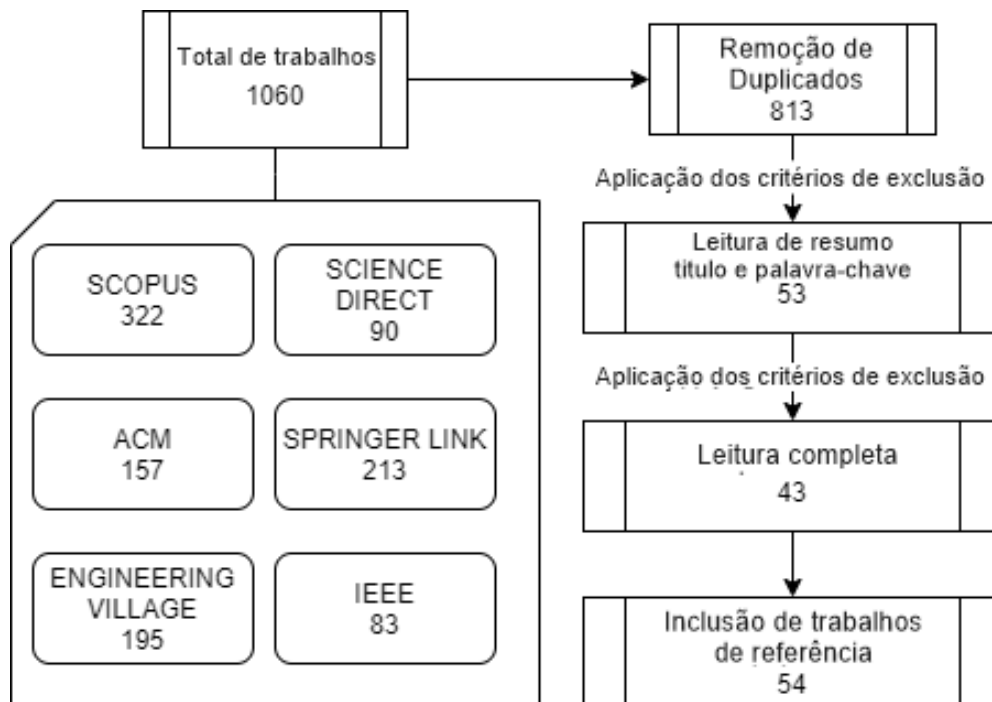
3.1.5 Etapa de Condução

A condução desta revisão sistemática foi realizada entre os meses de fevereiro e maio de 2020. Foram definidas quatro etapas para a seleção dos estudos: (i) execução da string de busca nas bases bibliográficas; (ii) remoção dos estudos duplicados; (iii) aplicação dos critérios de inclusão e exclusão aos trabalhos; e (iv) leitura e extração das informações dos estudos resultantes da etapa (iii). Os estudos foram lidos por dois revisores em consulta com um especialista da área.

No Estágio 1, a pesquisa por string foi executada nas bases bibliográficas selecionadas para esta revisão. A visão geral desse estágio pode ser observada na Figura 3.

Essa etapa teve início com 1060 trabalhos importados das bases bibliográficas selecionadas. No estágio 2, um total de 247 estudos duplicados foram removidos. No estágio 3, foram aplicados os critérios de inclusão e exclusão com base na leitura do título, resumo e palavras-chave, resultando na seleção de 53 estudos considerados promissores.

Figura 3 – Processo de Pesquisa



Fonte: O Autor

Para evitar a seleção de trabalhos que não se enquadram no escopo desta revisão, foram lidos completamente os 53 estudos selecionados na terceira etapa, o que resultou na exclusão de 10 trabalhos, totalizando 43 trabalhos selecionados. Os trabalhos rejeitados nessa etapa estavam dentro de dois critérios de exclusão:

1. Trabalhos que se concentram em outras áreas de Engenharia de Software: os trabalhos excluídos que estavam dentro desse critério eram metodologias que trabalhavam com SMA apenas em estágios posteriores à engenharia de requisitos. A engenharia de requisitos foi realizada de maneira tradicional, sem focar em nenhum recurso específico de SMA;
2. Trabalhos que abrangem uma metodologia já incluída em um trabalho: para esse critério, selecionamos o trabalho mais recente de forma que possamos entender o estado atual da metodologia.

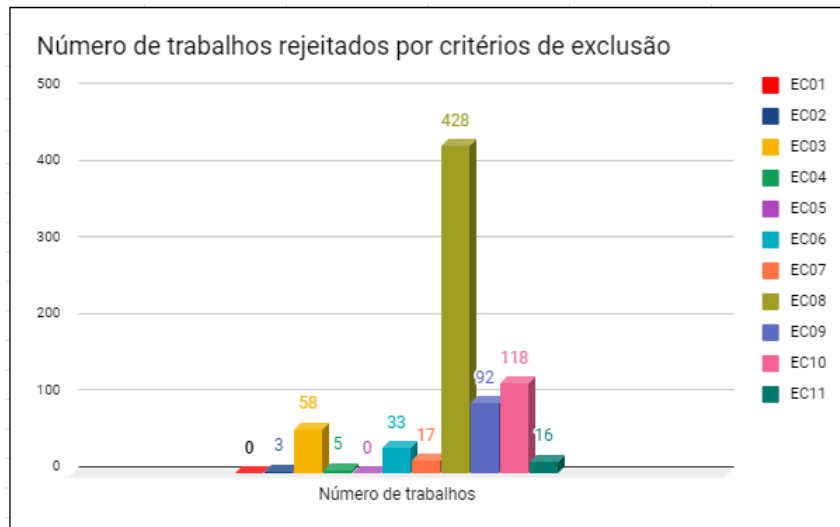
Criamos dois gráficos para mostrar os resultados no estágio de condução. A Figura 4 apresenta a visão geral da leitura dos trabalhos, com a contagem dos trabalhos aceitos, rejeitados e duplicados. Outra informação pode ser visualizada na Figura 5 que apresenta o número de artigos rejeitados por cada critério de exclusão.

Figura 4 – Visão geral do estágio de condução.



Fonte: O Autor

Figura 5 – Número de trabalhos rejeitados por critério de exclusão.



Fonte: O Autor

Ao final da etapa de condução, os estudos selecionados manualmente ((M. et al., 2014), (COSSENTINO; SEIDITA, 2014b), (GONZÁLEZ-MORENO et al., 2014), (BONJEAN et al., 2014), (DELOACH; GARCIA-OJEDA, 2014), (PADGHAM; THANGARAJAH; WINIKOFF, 2014), (CAIRE et al., 2004), (CAO, 2015), (N., 1997), (IGLESIAS et al., 1998), (LIND, 2001)) foram adicionados ao conjunto de artigos pesquisados nas bases, conforme definido no protocolo. Isso resultou em um total de 54 estudos aceitos.

3.1.6 Etapa de Extração dos dados

A extração feita nos trabalhos aceitos teve como estratégia a definição das questões a serem observadas nos 54 trabalhos. Essas questões foram definidas para ajudar o pesquisador a responder às questões da pesquisa. Esses problemas são:

- Quais subáreas da ER do SWEBOK o processo cobre?
- O processo possui um ciclo de vida de ER bem definido?
- A ER usada no processo é adequada para SMAs?
- O processo suporta o uso de modelo BDI?

A condução dessa etapa foi realizada em dupla, onde cada pesquisador leu o artigo e extraiu as informações sobre as questões citadas anteriormente. Os conflitos entre os pesquisadores foram resolvidos por um especialista da área.

3.2 Resultados da Revisão

As informações relevantes dos estudos selecionados foram obtidas por meio de uma planilha de extração de dados. Os resultados gerais da revisão e as evidências encontradas sobre cada questão de pesquisa são discutidas nas próximas subseções.

3.2.1 Resultados Gerais

A Figura 3 mostra a frequência dos artigos selecionados durante as quatro etapas da revisão. Na Etapa 1, de um total de 1060 artigos, a base da Scopus teve o maior número de artigos pesquisados, com 322 (30% do total), seguida da base da Springer Link, com 213 (20% do total), Engineering Village com 195 (18% do total), ACM Library com 157 (15% do total), Science Direct com 90 (9% do total) e IEEE Xplore com 83 estudos (8% do total).

Já no Estágio 2, de um total de 53 estudos, a base do Scopus teve 26 artigos pesquisados (49% do total), seguido do Science Direct com 11 (20 % do total), IEEE Xplore com 10 (19% do total), ACM Digital Library com 3 estudos (6% do total), Engineering Village com 2 (4% do total) e Springer Link com 1 (2% do total) .

Na Etapa 3, de um total de 43 estudos a base do Scopus teve o maior número de artigos pesquisados com 20 (47% do total), seguido pelo IEEE Xplore com 10 (23% do total), Science Direct com 8 (18% do total), ACM Digital Library com 2 estudos (5% do total) e Springer Link com 1 (2% do total). Por fim, na Etapa 4, 11 artigos, selecionados manualmente, foram agregados ao conjunto de estudos pesquisados nas bases, totalizando 54 artigos relevantes.

Ao todo, como pode ser visto na Figura 4, notamos que 770 estudos foram rejeitados, o que corresponde a 72,6% do total. Encontramos 247 estudos duplicados, o que corresponde a 23,3% do total. Após a leitura total dos artigos, foram aceitos 43 estudos, o que corresponde a 4,1% do total.

3.2.2 Primeira Questão de Pesquisa (RQ1)

Esta subseção tem como objetivo responder quais metodologias para o desenvolvimento de sistemas multiagentes (SMA) suportam um ciclo de vida de Engenharia de Requisitos (RE) específico para esse tipo de sistema. Para responder a esta pergunta, encontramos 54 metodologias que abordaram ER para SMAs. Esses estudos podem ser observados na Tabela 4.

Tabela 4 – Cobertura das etapas de engenharia de requisitos das metodologias.

Metodologia	Elicitação	Análise	Especificação	Validação
KAOS using the six-variable model		X	X	
JAAMAS		X		
Patrizia Ribino			X	
AGSIRA			X	
GORMAS		X	X	
ATABGE	X			
RE4GAIA		X		
Xu Huiying		X	X	
REG for AOSE	X	X		
Extensão da GAIA		X		
B-Tropos	X	X	X	
JONGWON LEE		X		
Prabhat Ranjan	X	X	X	
SRAMO		X		
Zhiqi Shen		X		
SONIA		X	X	
BDI ASP		X		
Tropos		X	X	
Cysneiros		X	X	X
Chiung-Hui	X	X		
KAOS		X	X	
PRACTIONIST		X	X	
Murray			X	
ASPECS		X	X	
REF		X	X	
Sen and Jain	X			
Hsieh et al.		X	X	
Sutcliffe		X	X	
Ágil Sen and Jain	X			
CREWS-EVE	X		X	X
Cao et al.		X	X	
HOMER	X			
Wu et al.	X			
Liu and Li		X		
Mahmoud et al.		X		
Ashamalla et al.	X	X		
Consensus		X	X	
Hilaire et al.		X		
Gaur and Soni		X		
Passos et al.	X	X		
PLANT		X	X	
Ronald et al.		X	X	
aMIAC		X	X	
GAIA	X	X	X	
PASSI		X	X	
INGENIAS-SCRUM	X	X	X	
ADELFE	X	X	X	X
O-MaSE		X	X	
PROMETHEUS		X		
MESSAGE		X	X	
OSOAD		X		
COMOMAS		X		
MAS-COMMONKADS		X	X	
MASSIVE	X	X	X	
Total:	17	46	31	3

Fonte: O Autor

3.2.3 Segunda Questão de Pesquisa (RQ2)

Esta subseção tem como objetivo responder qual é a abrangência da engenharia de requisitos nas metodologias analisadas, tomando como base as subáreas definidas no SWEBOK (BOURQUE; FAIRLEY et al., 2014).

Dos 54 estudos selecionados observamos que todos apresentam adequação da Engenharia de Requisitos para sistemas multiagentes. Assim, extraímos quais subáreas de ER definidas no SWEBOK (BOURQUE; FAIRLEY et al., 2014) são suportadas por esses estudos.

A Tabela 4 mostra os 54 estudos e as subáreas que os sustentam. Grande parte desses estudos, 46 no total, atende à subárea de análise de requisitos. Enquanto 31 deles suportam a subárea de especificação de requisitos.

A subárea de elicitação de requisitos, por sua vez, está amparada em 17 estudos. Por fim, a subárea de validação de requisitos apresenta o menor número de estudos, com apenas 3 do total.

Observamos também que, a partir desses estudos, apenas a metodologia ADELFE (BONJEAN et al., 2014) suporta as quatro subáreas de ER (elicitação, análise, especificação e validação).

Outro fato importante que percebemos na extração é que apenas 30 estudos apresentaram algum experimento empírico para validação do processo. Corresponde apenas a 55,55 % do total.

3.2.4 Terceira Questão de Pesquisa (RQ3)

Esta subseção tem como objetivo responder quais processos analisados oferecem suporte ao modelo BDI durante a engenharia de requisitos.

Tentamos identificar quais metodologias suportam o modelo BDI. Observamos que a maior parte dos estudos, 35 no total, apoia parcialmente o modelo BDI identificando pelo menos uma das características desse modelo. Esses recursos são: crenças do agente; objetivos/desejos do agente; e intenções do agente. No entanto, é necessário afirmar que a maioria desses trabalhos não cita explicitamente o modelo BDI, a maioria deles são metodologias orientadas a objetivos, se concentrando em apenas uma característica do modelo BDI e não necessariamente usam esse modelo.

Os objetivos dos agentes foram a característica identificada em maior quantidade nos estudos. A representação de intenções foi identificada em apenas três estudos e as crenças foram identificadas em apenas um estudo. Percebemos que as intenções e crenças não são identificadas isoladamente, sempre acompanhadas de objetivos. Por outro lado, os objetivos são identificados de forma isolada com frequência, sem serem acompanhados pelos demais recursos.

Outra questão a ser destacada é que 12 estudos não apresentam suporte para o

modelo BDI e apenas 8 apresentam suporte para esse modelo, mesmo que apresentando limitações nesse suporte.

A seguir, apresentamos uma sinopse de cada trabalho selecionado, com foco na engenharia de requisitos e no suporte aos recursos do modelo BDI de cada metodologia. Os estudos são ordenados da seguinte forma (i) trabalhos que não suportam o modelo BDI; (ii) trabalhos que suportam apenas objetivos; e (iii) estudos que suportam totalmente o modelo BDI.

A metodologia SONIA (ALONSO et al., 2004) não identifica nenhuma característica do modelo BDI. Essa metodologia permite a geração de uma arquitetura multiagente para resolver um problema de acordo com um modelo de projeto multiagente que não está centrado apenas nos agentes.

O estudo de Murray (MURRAY, 2003) também não atende o modelo BDI. Essa abordagem especifica os comportamentos dos agentes em diferentes níveis de abstração com o auxílio de gráficos de estado UML, permitindo a modelagem da cooperação entre diferentes tipos de agentes.

O estudo de Hsieh et al. (Hsieh et al., 2008) propõe uma metodologia para obtenção de requisitos, enfatizando a grande participação dos stakeholders e mínima documentação. Esse estudo não atende ao modelo BDI.

O estudo de Liu (Liu; Li, 2015) é focado no planejamento de requisitos de tempo de execução e não suporta o modelo BDI. Esse trabalho propõe uma abordagem de planejamento orientado a requisitos para autoadaptação, com foco na modelagem de requisitos contextuais e executáveis.

O trabalho de Ben Mahmoud (Hajer; Taieb; Raouf, 2009) visa modelar um sistema de gestão da qualidade baseado em sistemas multiagentes. Esse estudo não suporta o modelo BDI e é voltado para o esquema PDCA (Plan, Do, Check, Act).

O trabalho de Bokma cite bokma: 1994 também não suporta o modelo BDI. Apresenta o método de consenso para o desenvolvimento de sistemas baseados em agentes em larga escala. O processo adapta a engenharia de requisitos tendo como diferencial a rastreabilidade dos requisitos em componentes de projeto e código que facilita a verificação.

Domann in (DOMANN et al., 2014) propõe aMIAC, uma metodologia ágil para sistemas multiagentes. A metodologia utiliza o processo MIAC adaptado ao modelo ágil. O processo é dividido em 6 etapas, sendo que as fases de gerenciamento de requisitos, derivação do sistema e interface com o usuário e modelagem de funções são relacionadas à engenharia de requisitos. Nessas fases, sprints semelhantes à metodologia SCRUM são incluídos. Essa metodologia não suporta o modelo BDI.

PASSI (COSSENTINO; SEIDITA, 2014b) é uma metodologia para sociedades de agentes. As fases dessa metodologia, englobadas pela engenharia de requisitos, são classificadas como descrição de requisitos de domínio, identificação de agentes, identificação

de papéis e especificação de tarefas. Essa metodologia não suporta o modelo BDI.

INGENIAS SCRUM (GONZÁLEZ-MORENO et al., 2014) é uma metodologia que adapta o SCRUM com o metamodelo INGENIAS, focado em sistemas multiagentes. Analisamos a metodologia e verificamos que ela não suporta o modelo BDI.

ADELFE (BONJEAN et al., 2014) é uma metodologia para sistemas multiagentes adaptativos. Nesse tipo de sistema, um ambiente é composto por agentes interdependentes. Cada agente desempenha uma função parcial e a organização, durante o tempo de execução, faz com que o sistema execute uma função emergente. Essa metodologia não suporta o modelo BDI.

A metodologia MASSIVE (LIND, 2001) não oferece nenhum suporte ao modelo BDI. Essa metodologia não define um ciclo de vida de ER, no entanto, descreve tarefas, como classificação de casos de uso, visão do ambiente e identificação de funções que o analista deve seguir.

Em relação ao suporte do modelo BDI, GAIA (M. et al., 2014) é o primeiro processo completo (suportante diferentes áreas de atuação da Engenharia de Software) proposto para o desenvolvimento de sistemas multiagentes. No entanto, Gaia suporta apenas objetivos.

A metodologia JAAMAS (ABUSHARK et al., 2016) também identifica apenas os objetivos do agente. Esta é uma metodologia baseada em Prometheus (PADGHAM; THANGARAJAH; WINIKOFF, 2014) e orientada para objetivos. Os objetivos do agente são especificados por meio de diagramas de atividades.

A metodologia GORMAS (ARGENTE; BOTTI; JULIÁN, 2009) também identifica apenas os objetivos dos agentes. É uma metodologia de engenharia de software orientada a agentes para desenvolver organizações virtuais que permite descrever e projetar SMA orientados a organizações.

Assim como GORMAS, ATABGE (SEN; HEMACHANDRAN, 2010) também permite identificar apenas os objetivos dos agentes. Os autores propõem uma técnica ágil de elicitação de objetivos baseada em agentes que permite a extração de objetivos dos stakeholders.

No estudo proposto em RE4Gaia (BLANES; INSFRAN; ABRAHÃO, 2009) também são identificados apenas os objetivos dos agentes. Essa abordagem de modelagem de requisitos para o desenvolvimento de SMA estende a metodologia GAIA (ZAMBONELLI; JENNINGS; WOOLDRIDGE, 2003). Ele fornece uma abordagem sistemática que trata dos requisitos estabelecendo mecanismos de rastreabilidade que ajudam os analistas a entender as necessidades dos usuários.

No estudo de (FUENTES-FERNÁNDEZ; GÓMEZ-SANZ; PAVÓN, 2009), que apresenta a metodologia REG para AOSE, foram identificados apenas os objetivos dos agentes. O estudo apresenta diretrizes que permitem capturar informações sobre os sistemas multiagentes, ambientes humanos e outras características que auxiliam na elicitação de

requisitos. Entre essas diretrizes está um conjunto de perguntas e respostas que orientam a fase de elicitação de requisitos.

Da mesma forma, o estudo de (RODRIGUEZ et al., 2009), apresenta uma extensão do GAIA, identificando apenas os objetivos dos agentes. A proposta do estudo inclui a aplicação de casos de uso e diagramas de sequencia para a especificação de requisitos. Além disso, permite identificar relações de dependência entre os objetivos.

O estudo de (BRYL et al., 2008) B-Tropos, também identifica apenas os objetivos dos agentes. Essa metodologia atribui grande importância à análise de requisitos inicial. A análise de objetivos consiste em refinar objetivos e motivar novas relações sociais entre os agentes.

Nessa mesma linha, o estudo de Lee (LEE; LEE, 2008) também identifica apenas os objetivos dos agentes. A metodologia consiste em análise de requisitos estratégicos, modelagem de agentes, projeto e implementação de sistemas. Uma característica interessante do processo é a identificação dos agentes estratégicos que, são extraídos de três perspectivas de interação com o stakeholder.

O estudo proposto por Ranjan (RANJAN; MISRA, 2006) também apresenta apenas a identificação de objetivos. Essa metodologia de análise de requisitos divide a fase de análise em análise centrada no usuário e análise centrada no sistema. Esse refinamento é obtido pela identificação e separação de todos os requisitos em três tipos (i) requisitos funcionais; (ii) requisitos não funcionais; e (iii) requisitos de corte transversal.

O estudo de (LEE; LEE, 2006) apresenta o método SRAMO que faz parte da metodologia MAAEM. Esse método identifica apenas os objetivos do agente. Os autores apresentam uma técnica baseada em ontologia para análise de requisitos de SMA por meio de modelos de domínio de reutilização.

O estudo de Shen (SHEN et al., 2005) também identifica apenas os objetivos dos agentes. A metodologia abrange todo o ciclo de vida do desenvolvimento do MAS, desde a análise de requisitos, passando pelo projeto arquitetônico e detalhado, até a implementação.

O estudo de Chiung-Hui (LEE; LIU, 2002) também identifica apenas os objetivos dos agentes. Nesse estudo os autores apresentam um método baseado em casos de uso orientado a objetivos para análise de requisitos. A abordagem de casos de uso é usada para obter requisitos do ponto de vista do usuário.

Da mesma forma, o estudo de KAOS (BANACH, 2010) também identifica apenas os objetivos dos agentes. O estudo apresenta uma estrutura de engenharia de requisitos orientada por objetivos, baseada na decomposição e refinamento de objetivos.

O estudo de (COSSENTINO et al., 2010) sobre a metodologia ASPECS identifica apenas os objetivos dos agentes. O ciclo de vida dessa metodologia é dividido em três fases, sendo que a primeira consiste na fase de requisitos do sistema, que visa identificar a hierarquia das organizações. A segunda fase define a sociedade agente e a terceira consiste

na implantação e implantação do SMA.

No estudo de (BRESCIANI; DONZELLI, 2003) é proposto um framework, denominado REF, para engenharia de requisitos de SMA. Ele permite que partes interessadas não técnicas elicitam requisitos auxiliados por engenheiros de requisitos. Ao mesmo tempo, fornece um processo para aquisição, análise e refinamento de requisitos. No entanto, ele apenas identifica os objetivos dos agentes.

O estudo de Sen e Jain (JAIN, 2007) propõe uma metodologia que identifica apenas os objetivos dos agentes. Essa metodologia envolve a participação máxima dos stakeholders onde os objetivos são eliciados em sessões de grupo de forma interativa.

O estudo de Sutcliffe (SUTCLIFFE, 2001) trabalha com engenharia de requisitos para sistemas colaborativos complexos. Esses sistemas incluem sistemas multiagentes. Nessa metodologia os agentes podem atuar de forma colaborativa para realizar um objetivo, assim, esse estudo dá suporte para a identificação dos objetivos.

O estudo de Sen (Sen; Jain, 2007) tem como objetivo apresentar uma técnica ágil de refinamento de objetivos com base em agentes para obtenção de objetivos flexíveis. Para apresentar a técnica, o trabalho também propõe uma metodologia ágil, focada na fase de elicitação.

O estudo de Haumer (Haumer et al., 1999) enfoca o uso de cenários e objetivos de suporte. Esse estudo cobre a rastreabilidade de requisitos, investiga impactos e facilita possíveis mudanças no sistema. Adaptando apenas objetivos, com relação ao modelo BDI.

O estudo de Cao (Longbing Cao et al., 2004) suporta o uso de objetivos e apresenta uma estrutura de modelagem para análise de requisitos iniciais orientada a agentes.

Em outro estudo, Wilmann (Wilmann; Sterling, 2005) propôs a abordagem para elicitação de requisitos chamada Homer. Ele suporta o uso de objetivos. Homer visa aumentar a facilidade para projetar e desenvolver sistemas baseados em agentes. Essa abordagem pode ser facilmente combinada com outras abordagens existentes.

O trabalho de Haochi (Wu; Liu; Ma, 2010) também apoia apenas objetivos e tem como proposta otimizar o processo de obtenção de requisitos por meio do uso combinado da estrutura de modelagem i^* e do modelo quantitativo de análise de risco de Redes Bayesianas, de forma que os atores estratégicos possam estimar as oportunidades e riscos potenciais de alto nível em relação aos objetivos em uma rede multiagente.

O trabalho de Ashamalla (ASHAMALLA; BEYDOUN; LOW, 2017) defende uma abordagem orientada a modelos para garantir que as restrições de requisitos em tempo real sejam levadas em consideração antes do projeto de um sistema multiagente. Assim, eles apresentam um metamodelo que suporta requisitos de tempo real. Essa abordagem cobre apenas objetivos.

Em relação ao modelo BDI, o estudo de Hilaire em (HILAIRE et al., 2012) apoia apenas objetivos. Apresenta uma abordagem para integrar inteligência de enxame a sistemas multiagentes.

Apoiando apenas objetivos, o trabalho de Gaur (GAUR; SONI, 2012) propõe uma abordagem para explorar as dependências entre os agentes dos requisitos dos usuários. Ele propõe uma classificação mais explícita e sistemática das dependências sociais como parte integrante do processo de engenharia de requisitos que eventualmente ajudaria o desenvolvedor a avaliar o número ideal de agentes dependentes de acordo com o tipo e a natureza das dependências múltiplas.

O estudo de Sanchez em (PASSOS; ROSSETTI; GABRIEL, 2015) apoia apenas objetivos. Esse artigo propõe uma metodologia de agente com foco em processos, meio ambiente e serviços.

Wang in (WANG et al., 2013) apresenta PLANT, um processo padrão para transformar cenários em modelos de requisitos, primeiro estabelecendo a base teórica desta linguagem e, a seguir, fornecendo uma descrição detalhada, ilustração e avaliação desta linguagem. Essa abordagem suporta apenas objetivos.

O trabalho de Ronald in (RONALD et al., 2012) apoia apenas objetivos. Esse artigo apresenta a concepção e a implementação de um modelo baseado em agentes de geração e programação de atividades sociais para fins experimentais, com o objetivo de explorar os efeitos do espaço social.

MESSAGE (CAIRE et al., 2004) é uma metodologia que engloba a análise e o projeto para SMA. A notação do MESSAGE estende a UML com conceitos no nível do conhecimento do agente, fornecendo diagramas para visualizá-los. Nessa metodologia o suporte ao modelo BDI é parcial, abrangendo apenas objetivos.

A metodologia OSOAD (CAO, 2015) aborda as características de sistemas complexos, incluindo abertura de ambiente, interações, relacionamentos, regras e sociabilidade. Essa metodologia cobre apenas a representação de objetivos.

O estudo de Nelufar et al. (ULFAT-BUNYADI; MOHAMMADI; HEISEL, 2018), identifica os objetivos dos agentes. Os objetivos de diversos agentes são refinados até que possam ser atribuídos a agentes únicos no software ou no ambiente. A metodologia apresentada no estudo é orientada a objetivos.

Já o estudo proposto por Huiying e Zhi (HUIYING; ZHI, 2009) define um grupo de métodos que utilizam símbolos gráficos para representar os agentes e propõe um método formal que permite a identificação dos objetivos dos agentes.

O-MaSE (DELOACH; GARCIA-OJEDA, 2014) é um framework para customizar metodologias para sistemas multiagentes. Apresenta modelos e tarefas para a execução de ER, mas não define um ciclo de vida ou obrigatoriamente a utilização desses modelos. Essa estrutura oferece suporte a objetivos.

A abordagem AGSIRA (LIU et al., 2011) suporta apenas objetivos. Esse estudo apresenta uma abordagem de análise de requisitos integrada ao cenário de destino do agente.

O estudo de Ribino et al. (RIBINO et al., 2013) apresenta uma metodologia

que suporta totalmente o modelo BDI. O estudo propõe a utilização de um modelo ontológico para representação de requisitos e um modelo de objetivo para a formalização do domínio com foco nos aspectos normativos e organizacionais do desenvolvimento de sistemas multiagentes BDI.

Agora, o estudo BDI ASP (JO; EINHORN, 2005) atende totalmente o modelo BDI. Nesse estudo, o modelo BDI foi usado como uma ferramenta para analisar os ambientes, objetivos e comportamentos dos agentes.

A metodologia Tropos (MYLOPOULOS; CASTRO; KOLP, 2013) também atende plenamente o modelo BDI. Nessa metodologia, as noções de agentes e as noções mentais relacionadas são utilizadas em todas as fases de desenvolvimento do SMA. Permite entender mais profundamente o ambiente em que o software deve operar e o tipo de interação que deve ocorrer entre o software e os agentes.

O estudo de Cysneiros (CYSNEIROS; YU, 2002) também atende plenamente o modelo BDI. Nessa metodologia, o ator é usado como um construtor central durante a elicitação e análise de requisitos. Durante a fase de elicitação e análise de requisitos, a metodologia permite desenvolver a compreensão do problema, bem como analisar estruturas organizacionais, processos e relações sociais.

Agora, o estudo do PRACTIONIST (MORREALE et al., 2006) atende totalmente ao modelo BDI. Esse estudo adotou uma abordagem orientada a objetivos para desenvolver agentes BDI. Os agentes, nesse trabalho, podem ser programados em termos de objetivos que estão relacionados a desejos ou intenções, dependendo de algumas condições específicas a serem satisfeitas ou não.

Prometheus (PADGHAM; THANGARAJAH; WINIKOFF, 2014) é uma metodologia para o desenvolvimento de SMA que evoluiu a partir da experiência industrial e pedagógica. Essa metodologia oferece suporte para o modelo BDI, permitindo a representação de crenças, desejos e intenções.

A metodologia CoMoMAS (N., 1997) deriva modelos da metodologia CommonKADS, porém, diferentemente do CommonKADS, essa abordagem considera vários aspectos específicos do agente. Particularmente o uso de estruturas de conhecimento adicionais e sua representação flexível para garantir a autonomia do agente em tempo de execução. Essa metodologia oferece cobertura total ao modelo BDI, permitindo a utilização de crenças, desejos e intenções pelos agentes.

A metodologia MAS-CommonKADS (IGLESIAS et al., 1998) estende a metodologia de engenharia de conhecimento CommonKADS com técnicas de metodologias de ER voltadas objetos e protocolos de forma a desenvolver sistemas multiagentes. Em relação ao suporte do modelo BDI, essa metodologia fornece suporte total.

3.2.5 Quarta Questão de Pesquisa (RQ4)

Esta subseção tem como objetivo responder quais são as lacunas existentes nas metodologias que suportam a engenharia de requisitos para SMA.

Observamos que apenas três estudos abrangem a subárea de validação em seu ciclo de ER. Isso demonstra que a maioria das metodologias não se preocupa com essa fase tão importante para a qualidade dos sistemas.

Observamos também que apenas um estudo abrange as quatro subáreas da ER em seu ciclo (BONJEAN et al., 2014). Por outro lado, esse estudo não suporta o modelo BDI, o que demonstra uma lacuna e a necessidade da proposição de uma metodologia contendo todas as subáreas de engenharia de requisitos que suporte o modelo BDI. Outro ponto é que observamos que a validação na metodologia ADELFE é feita apenas nas primeiras etapas, validando interfaces gráficas, requisitos de usuário preliminares e casos de uso (não específicos para SMAs), além disso, a etapa que ADELFE denomina como fase de análise, que trata de características específicas de agentes, não apresenta nenhuma forma de validação, deixando a validação apenas para etapas de ER tradicional.

Com relação à elicitação, apenas a metodologia Homer (Wilmann; Sterling, 2005) determinou atividades totalmente voltadas a SMA em conjunto com os stakeholders, mesmo assim a metodologia não cobre o modelo BDI, além de que a metodologia Homer limita-se a identificar os agentes e suas funções dentro do sistema, não cobrindo outras funções do sistema que o agente irá atuar. As outras metodologias apresentam uma elicitação baseada apenas em documentação, não tendo interação direta com o stakeholder, ou com pouco foco em SMA.

Em relação à cobertura do modelo de BDI, entendemos que o apoio a apenas 8 estudos de um total de 54 é um número baixo, considerando que esse apoio não é total pois as metodologias tem problemas quanto a cobertura das subáreas de ER. Além disso, apenas duas metodologias têm como foco cobrir esse modelo ((JO; EINHORN, 2005), (RIBINO et al., 2013)). Outro ponto que podemos identificar é que, entre as metodologias que suportam o modelo BDI, apenas a metodologia Tropos (MYLOPOULOS; CASTRO; KOLP, 2013) cobre a elicitação de requisitos, mesmo assim sua elicitação não aborda especificamente o modelo BDI, atribuindo a quem executa a elicitação extrair as crenças, desejos e intenções, sem suporte da metodologia. E apenas a metodologia proposta por Cysneiros (CYSNEIROS; YU, 2002) suporta a validação de requisitos. Isso demonstra que grande parte das metodologias que suportam o BDI tem como foco a análise e especificação de requisitos e que, além dessas áreas, há espaço a ser explorado nas áreas de elicitação e validação.

3.3 Ameaças a Revisão

Durante o planejamento e execução dessa revisão, alguns fatores foram caracterizados como ameaças à validade da pesquisa. As ameaças potenciais são discutidas para orientar a interpretação desse trabalho:

1. **validade interna:** O protocolo dessa revisão foi rigorosamente seguido, considerando principalmente os critérios de inclusão e exclusão. Quando necessário, foi consultado pesquisador com experiência na área para chegar a um consenso sobre a aceitação dos estudos identificados;
2. **validade Externa:** Como a string de pesquisa é limitada e o tema de estudo é abrangente, alguns estudos podem não ter sido encontrados. Para minimizar essa ameaça, o livro "Handbook on Agent-Oriented Design Processes" (COSSENTINO et al., 2014c) foi usado como fonte de pesquisa e alguns artigos clássicos foram selecionados manualmente por um especialista na área. Para complementar a pesquisa foi realizada uma busca manual nas metodologias encontradas com o objetivo de garantir a utilização de estudos com a versão mais recente;
3. **validade da conclusão:** a confiabilidade da string de pesquisa definida para selecionar trabalhos relevantes pode ser uma ameaça à conclusão. Para minimizar essa ameaça, o string foi desenvolvida com a execução de diversos testes e o especialista da área foi consultado sobre os termos mais utilizados.

4 UM PROCESSO DE ENGENHARIA DE REQUISITOS PARA SISTEMAS MULTIAGENTES

Neste capítulo apresentamos o processo de Engenharia de Requisitos para Sistemas Multiagentes proposto neste trabalho. Desse modo, a Seção 4.1 apresenta uma visão geral do processo, detalhando as etapas do mesmo. E a Seção 4.2 apresenta a execução do processo na segunda versão do sistema Heráclito.

4.1 Apresentação do Processo proposto

Após estudar as lacunas encontradas na revisão sistemática apresentado no capítulo 3, Seção 3.2, percebemos a necessidade de propor um processo de ER específico para SMAs que seja abrangente em sua cobertura e de fácil aplicação. É importante destacar que se pretende que o processo em questão faça parte de um processo de desenvolvimento completo para SMAs atualmente em desenvolvimento. A etapa de ER do processo proposto, que é o foco deste trabalho, visa a cobertura de todas as etapas definidas no SWEBOK, elicitación, análise, especificação e validación.

A elicitación de requisitos visa descobrir os requisitos específicos para SMAs em um contato direto com o stakeholder, elicitando agentes de software e suas funções, por meio de entrevistas, realizando uma metáfora organizacional, como a contratação de novos funcionários para uma empresa. A análise de requisitos permite identificar requisitos específicos para SMAs, como percepções, objetivos, crenças, planos e ações, através da representação de cenários. Em seguida a especificación de requisitos é realizada por meio de uma extensão do diagrama de casos de uso UML voltado para SMAs (GUEDES, 2012) e sua documentação.

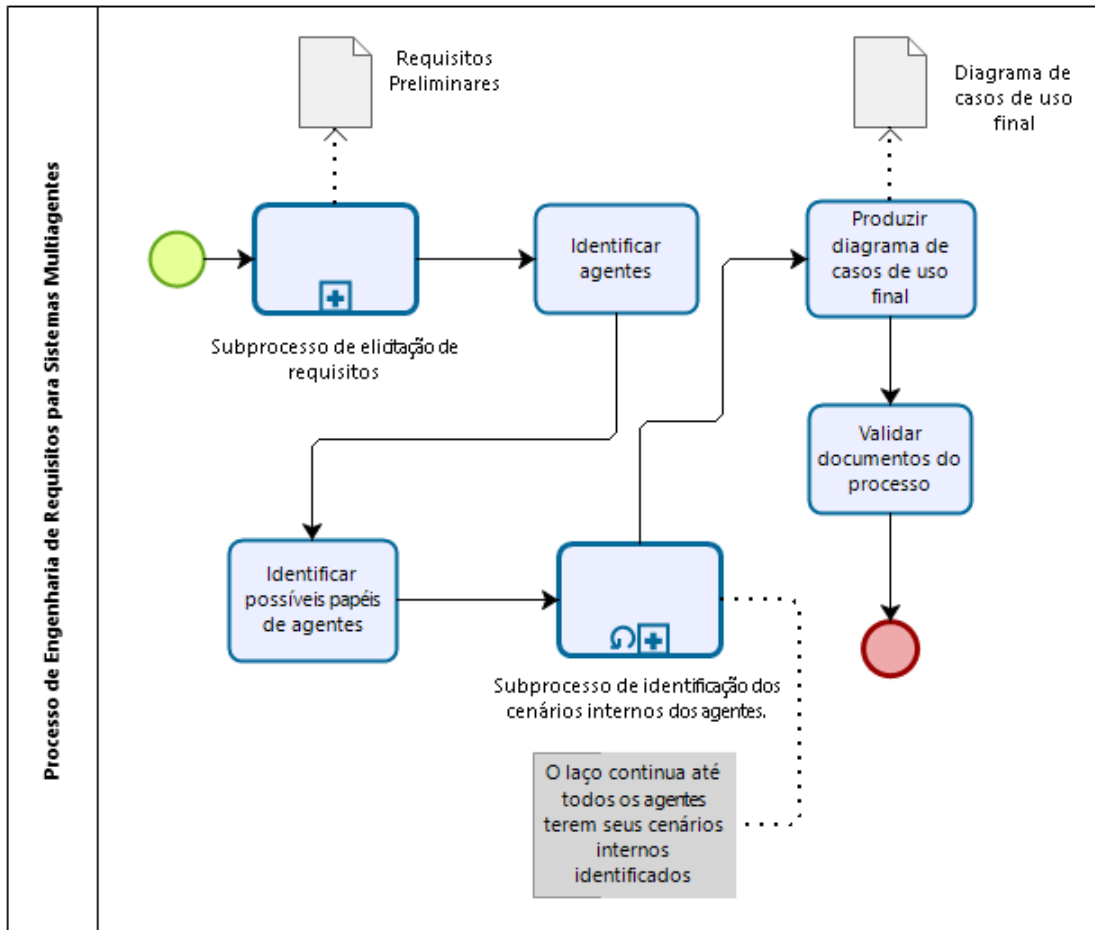
Uma validación mais profunda não faz parte do escopo deste trabalho, sendo estudada paralelamente por um membro do grupo de pesquisa, Giovane D'Ávila Mendonça, discente do Programa de Pós-Graduação em Engenharia de Software (PPGES) da Universidade Federal do Pampa. Esta divisão ocorreu pois entendemos que as lacunas encontradas na subárea de validación necessitariam de um estudo a parte e pretende-se que este estudo futuramente componha este processo.

A Figura 6 apresenta uma visão geral do processo proposto. O processo começa executando o subprocesso de Elicitación de Requisitos (Figura 7), que gera o documento de requisitos preliminares. Após elicitación os requisitos preliminares, o analista deve identificar os agentes presentes no sistema e em seguida quais papéis esses agentes podem assumir. Os papéis de agentes podem ser identificados verificando quais funcionalidades foram descritas nos cenários preliminares e agrupando essas funcionalidades por especialidade.

Para cada um dos papéis de agentes deve ser executado o subprocesso de Identificación dos Cenários Internos dos Agentes (Figura 8), nesse processo serão gerados os diagramas de casos de uso parciais de cada objetivo do agente, resultando na etapa de produção do diagrama de casos de uso final. Nesses cenários cada papel de agente, assu-

mindando funções que serão executadas no sistema. Após a produção do diagrama de casos de uso final se deve validar os documentos do processo produzidos durante a execução do mesmo.

Figura 6 – Visão geral do processo proposto.



Fonte: O Autor

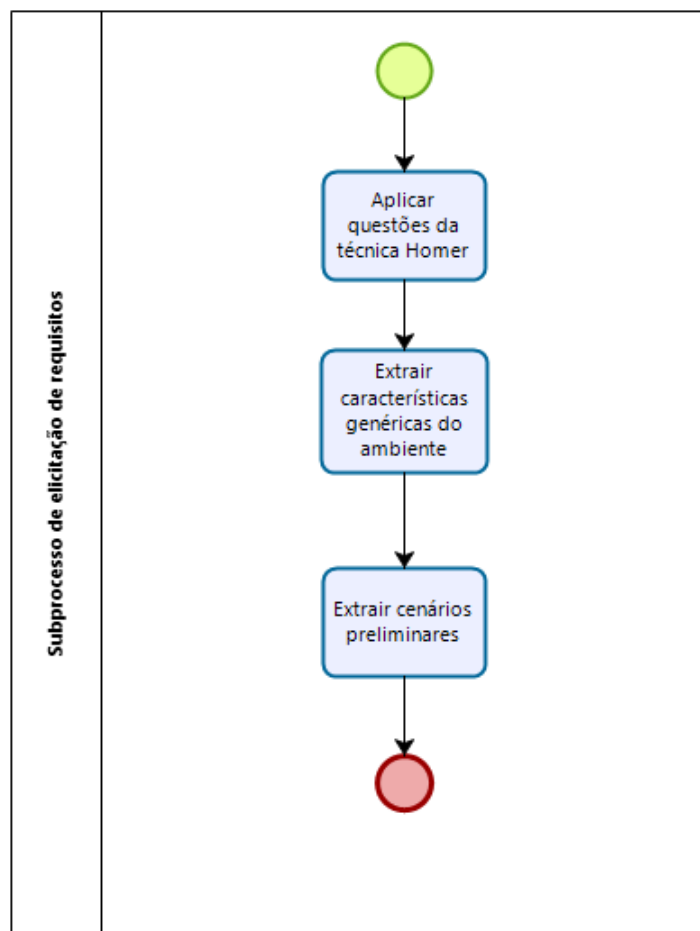
O subprocesso de Elicitação de Requisitos do processo proposto (Figura 7), se baseia na aplicação das questões definidas na técnica Homer, as perguntas estão listadas a seguir:

- Se você fosse contratar mais funcionários para a empresa, quais cargos precisaria preencher?
- Para cada cargo, faça uma descrição do trabalho:
 - Objetivo do cargo;
 - Que tarefas geralmente serão necessárias?

- Para cada tarefa, quais sub tarefas são necessárias?
 - Que restrições existem para essas tarefas?
 - Quais outras pessoas da empresa são necessárias para essa posição exercer o cargo?
 - Que conhecimento da empresa a pessoa precisa para executar essa tarefa?
 - Existem recursos necessários para cumprir a função?
- Quais condutas devem ser seguidas por todos os funcionários?
 - Quais outras regras a empresa deve cumprir?

Com a aplicação das perguntas pretende-se identificar características genéricas do ambiente e extrair os cenários preliminares, que servirão de base para a identificação de cenários internos dos agentes e da produção dos diagramas.

Figura 7 – Subprocesso de elicitação de requisitos do processo.



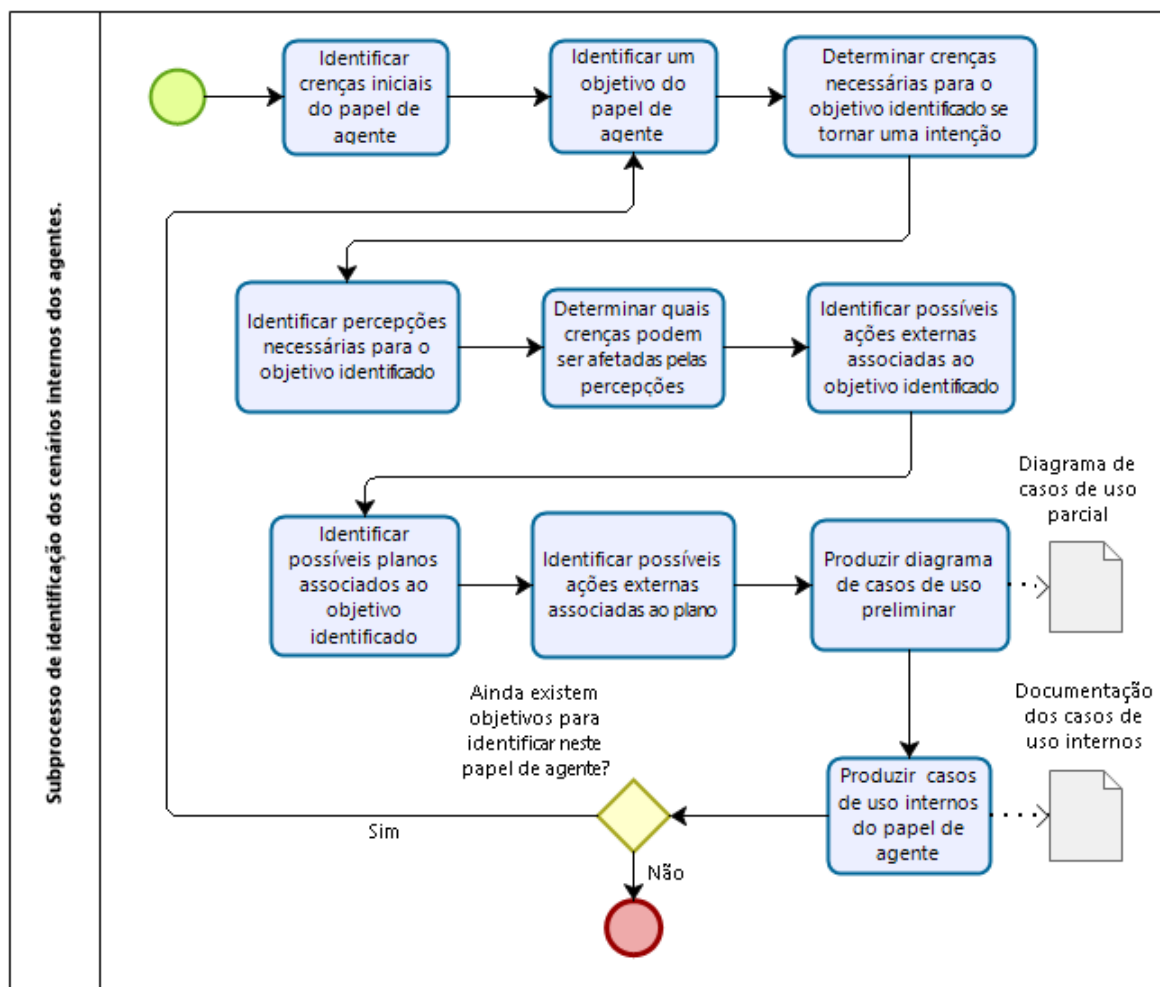
O subprocesso de Identificação de Cenários Internos dos Agentes (Figura 8), pretende extrair as características específicas dos agentes além da documentação de casos de uso internos e da produção de diagramas de casos de uso parciais.

Para cada papel de agente definido anteriormente, o analista deve identificar as crenças iniciais associadas a esse papel e identificar um objetivo desse papel de agente. Com a identificação desse objetivo o analista deve determinar as crenças necessárias para o objetivo virar uma intenção, identificar percepções necessárias para o objetivo identificado, determinar quais crenças podem ser afetadas pelas percepções identificadas e possíveis ações externas associadas ao objetivo.

O analista deve verificar se para esse objetivo é necessário um plano, é possível que um objetivo não tenha planos associados (em situações em que as reações do agente são simples e exijam pouco raciocínio), porém é possível ter mais de um plano, isso irá depender da complexidade do objetivo. Se for entendido que é necessário um plano, deve-se identificar os passos para cumprir este plano e as ações externas associadas ao mesmo.

O subprocesso é finalizado com a produção do diagrama de casos de uso parcial e a documentação dos cenários relativos aos casos de uso que o compõem.

Figura 8 – Subprocesso de identificação de cenários internos dos agentes



Fonte: O Autor

4.2 Execução do Processo

Esta Seção tem como objetivo apresentar a execução do processo proposto em uma segunda versão do Sistema Heráclito. O sistema Heráclito é um sistema multiagente voltado ao aprendizado interativo e dialético com foco no ensino de dedução natural da lógica proposicional (GALAFASSI et al., 2019). A execução do processo proposto teve como objetivo aplicar a ER para a segunda versão do sistema Heráclito que será desenvolvida.

Deste modo a subseção 4.2.1 apresenta a execução da elicitação de requisitos, a subseção 4.2.2 demonstra uma avaliação da elicitação utilizando a metodologia Homer, a subseção 4.2.4 apresenta os cenários especificados do Sistema Heráclito, a subseção 4.2.5 apresenta a validação de requisitos utilizada na execução do processo e a subseção 4.2.6 destaca os resultados obtidos através da execução do processo.

4.2.1 Elicitação de Requisitos com Homer

Durante a revisão sistemática analisamos 17 metodologias que trabalham elicitação em sistemas multiagentes. A metodologia Homer (*Human Oriented Method for Eliciting Requirements*) (Wilmann; Sterling, 2005), teve destaque entre essas metodologias, pois era a única que apresentava uma forma de levantar requisitos voltado ao contato direto com o stakeholder. Além disso, a metodologia Homer é feita especificamente para ser integrada a outros processos de desenvolvimento, motivo pelo qual, decidimos testar sua aplicação em nosso processo.

Outro ponto importante é que nenhuma metodologia apresentava uma elicitação própria para BDI, apesar da metodologia Tropos (MYLOPOULOS; CASTRO; KOLP, 2013) suportar o modelo BDI e apresentar uma etapa de elicitação, a Tropos apenas apresenta modelos que podem ser usados na elicitação, entretanto não é focada em como elicitar esses requisitos com os stakeholders.

O funcionamento da Homer pode ser descrito como o uso de metáforas organizacionais, como a contratação de novos funcionários para elicitar requisitos dos clientes. Esse estilo de elicitação visa descobrir com mais facilidade os papéis de agentes e os objetivos dos mesmos dentro do sistema. Para sua execução a Homer apresenta uma serie de perguntas que devem ser feitas ao stakeholder.

É importante destacar, que a metodologia Homer presume que é valido o uso de sistemas multiagentes e que o cliente quer uma solução orientada a agentes. Em nenhum momento é feito um teste de viabilidade do uso de sistemas multiagentes.

Para verificar seu funcionamento e sua aplicabilidade, executamos a metodologia Homer em entrevistas com os desenvolvedores da primeira versão do Sistema Heráclito. Esperamos que a engenharia de requisitos realizada nesse trabalho, contribua com o desenvolvimento da segunda versão do sistema Heráclito.

Ao total foram três entrevistas com stakeholders do sistema Heraclito. Também é importante destacar que, os conflitos encontrados durante a realização das entrevistas foram sanados por meio de novas entrevistas com os stakeholders.

As entrevistas realizadas serviram para extrair os cargos, dentro da metáfora organizacional proposta, e suas tarefas dentro da empresa. Conseguimos extrair também, as relações de dependência entre os cargos e suas limitações. O sistema Heráclito não possui uma Hierarquia entre os cargos, entretanto acreditamos que seria possível delimitar uma hierarquia com estas questões, para outros sistemas.

Com o conteúdo elicitado conseguimos criar uma descrição preliminar dos cenários presentes no Heráclito. Essa base serviria para identificarmos os Agentes do sistema e desenvolvermos os modelos de casos de uso, presentes nos cenários internos do sistema.

Entretanto gostaríamos de destacar que as funcionalidades referentes aos agentes externos (usuários), não foram possíveis de elicitare, visto que a metodologia Homer não cobre as ações do usuário. Neste caso realizamos perguntas a parte das questões definidas na metodologia Homer, com o objetivo de entender quais funcionalidades poderiam ser acessadas pelos usuários do sistema.

4.2.2 Análise SWOT da Metodologia Homer

Ao aplicarmos a Metodologia Homer podemos destacar diversos pontos sobre seu funcionamento, seus pontos fortes e suas limitações. Deste modo, utilizamos Análise SWOT para identificar as forças, fraquezas, oportunidades e ameaças da metodologia Homer.

Como pode ser observado na Figura 9, a metodologia Homer possui diversas fraquezas, sendo a maior parte delas com relação a sua cobertura ou dificuldade de extrair alguns requisitos particulares para SMAs. Entretanto consideramos a metáfora organizacional uma ótima técnica para extrair requisitos para um SMA. Deste modo, estender a metodologia Homer parece ser uma ótima oportunidade de estudo.

Figura 9 – Análise SWOT da Metodologia Homer.

<p style="text-align: center;">Fraquezas (-)</p> <p>A metáfora organizacional é útil mas pode deixar o Stakeholder confuso; Ações podem ser extraídas as tarefas, mas com dificuldade. Não existem meios simples de extrair percepções Algumas perguntas não são intuitivas de responder A profundidade da extração depende do entrevistado Não é possível extrair funções dos agentes de software, como por exemplo de agentes externos (como usuários) Não é possível extrair, com facilidade, as crenças dos agentes. Não é possível extrair, com facilidade, os planos dos agentes.</p>	<p style="text-align: center;">AMEAÇAS (-)</p> <p>Perguntas não intuitivas podem gerar falhas, conflitos ou erros de interpretações. Perguntas com necessidade de explicação prévia. Crenças e planos só são possíveis de extrair a partir da descrição da função</p>
<p style="text-align: center;">FORÇAS (+)</p> <p>A metáfora organizacional se encaixa muito bem no contexto de um Sistema Multiagente. Fácil extração dos papéis dos agentes Fácil extração dos objetivos dos agentes A descrição de funções é intuitiva.</p>	<p style="text-align: center;">OPORTUNIDADES (+)</p> <p>É fácil encaixar a metologia Homer em um processo novo Facilidade de extensão da metodologia. Diversas oportunidades de melhoria.</p>

Fonte: O Autor

4.2.3 Identificação dos Agentes e Papéis de Agentes

Ao executar as questões da metodologia Homer, adquirimos uma descrição das funções necessárias na organização. As funções descritas pelos entrevistados podem ser vistas a seguir:

- Entrevistado 1
 - Professor: Quem tem o conhecimento técnico, planeja as questões e o conteúdo do aluno;
 - Mediador: Único que tem contato com aluno para apresentar um feedback. Aplica as atividades ao aluno;
 - Analista: Analisa os dados do aluno para verificar um nível para o aluno.
- Entrevistado 2
 - Professor: Planeja o conteúdo do aluno e acompanha o desenvolvimento do aluno;
 - Tutor: Acompanha o uso do sistema;
 - Design de conteúdo: Planeja o uso do Heráclito.
- Entrevistado 3
 - Especialista Técnico: Verifica o que o aluno está fazendo e observa as ações do aluno;

- Mediador: Acompanha o desenvolvimento do aluno, dá dicas, aplica reforços e tem um plano de conteúdo;
- Gerente: Acompanha os passos do aluno, serve como um histórico.

Verificando as respostas notamos que todos os entrevistados apresentaram 3 funções base para o Heráclito. Todavia, o Entrevistado 2 mostrou respostas muito desconexas com os demais, ao analisar suas respostas percebemos que diversas partes do sistema estavam faltando. Em contrapartida os entrevistados 1 e 3, tiveram respostas semelhantes, mudando muitas vezes apenas a nomenclatura.

Com relação as respostas do entrevistado 2 tomamos a decisão de utilizar as que eram comparáveis à de outros entrevistados e, validar em outra entrevistas as que eram desconexas com os demais.

A partir da descrição destas funções delimitamos os agentes do sistema. Em seguida, analisando as respostas, não somente das questões, mas também das tarefas descritas, conseguimos identificar similaridades nos agentes descritos, definindo os seguintes papéis de agentes:

- **Especialista Técnico:** O Especialista Técnico é responsável por toda a parte que necessita conhecimento técnico. Este papel de agente prepara o conteúdo do aluno, corrige provas e questões e acompanha o aluno a cada ação, na resolução de uma questão;
- **Mediador:** Único responsável pelo intermédio do sistema com o aluno. Acompanha o desenvolvimento do aluno, apresentando feedback e dando dicas. Este papel de agente também conhece o perfil do aluno;
- **Analista Bayesiano:** Este papel de agente tem conhecimento sobre o histórico de questões e provas do aluno. Com base nisso o Analista Bayesiano prepara uma Rede Bayesiana contendo diversas estatísticas dos resultados do aluno.

Destacamos também que os agentes presentes no sistema Heráclito são agentes cognitivos. Ou seja, estes agentes apresentam um mecanismo de tomada de decisão, interações avançadas e contém um objetivo fortemente estabelecido.

4.2.4 Produção dos Cenários

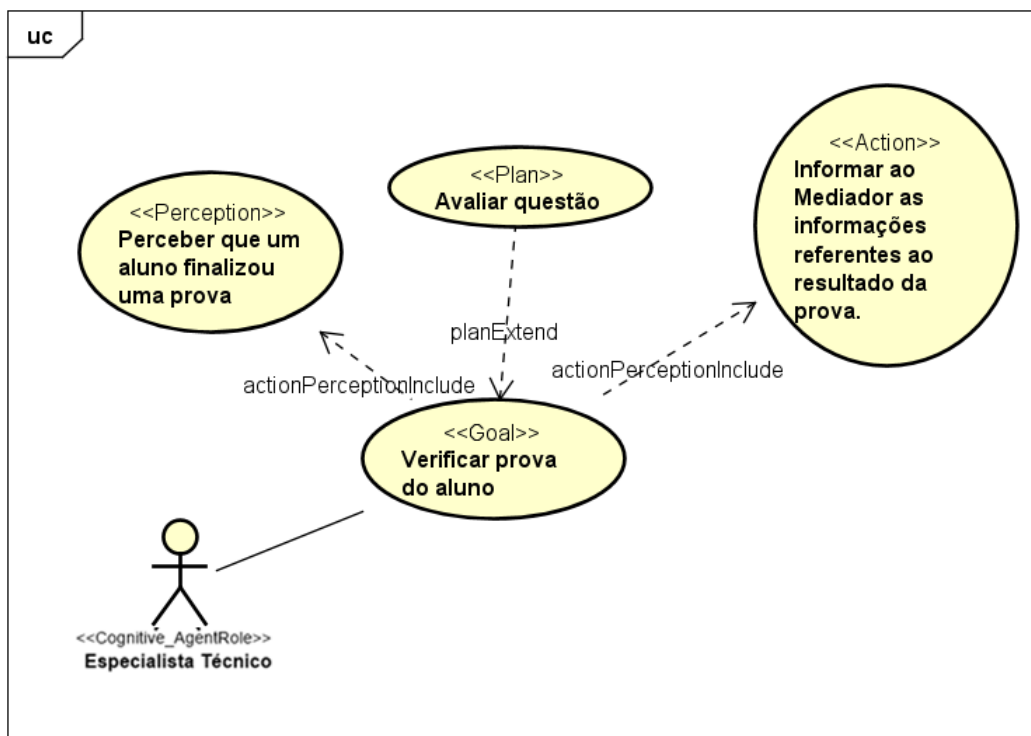
Segundo Cossentino et al. (2014b) a atividade de descrição do cenário deve descrever um conjunto de possíveis interações dentro de uma organização. O objetivo é refinar e explorar as possíveis sequências de interações entre os objetivos.

Dentro desse contexto a etapa de produção dos cenários visa identificar as características internas dos agentes, produzindo diagramas de casos de uso parciais e descrição dos cenários do sistema. Sendo estas descrições realizadas a partir dos cenários preliminares extraídos na etapa de elicitación de requisitos.

O diagrama de casos de uso parcial (Figura 10), apresenta a sequência de funcionalidades do agente Especialista Técnico utilizadas para verificar uma prova de um aluno. O agente percebe que um aluno finalizou uma prova, deste modo o objetivo "Verificar prova do aluno", se torna uma intenção, disparando o plano "Avaliar questão" para cada uma das questão da prova. Após corrigir as questões da prova o Especialista Técnico informa o Mediador as informações referentes ao resultado da prova. A descrição dos passos de cada cenário necessário para atingir este Objetivo podem ser vistas nas Tabelas 5, 6 e 9.

É importante destacar que o plano "Avaliar questão" (Tabela 9), é um plano compartilhado. Este plano é utilizado tanto no Objetivo "Verificar prova do aluno" (Tabela 5), quanto no "Verificar Questão do aluno" (Tabela 7).

Figura 10 – Diagrama de Casos de Uso Parcial - Verificar Prova do Aluno



Fonte: O Autor

Tabela 5 – Objetivo - Verificar prova do aluno

Nome do caso de uso interno	Verificar prova do aluno
Stereotype	Goal
AgentRoleActor	Especialista Técnico
Resumo	Este caso de uso interno descreve as etapas seguidas pelo agente que assume a função de Especialista Técnico a verificar uma prova realizada pelo aluno.
Crenças Iniciais	Necessidade de nova verificação de prova negativa
Percepções	Sondagem de novas finalizações de prova.
Cenário principal	
Ações do AgentRoleActor	
1. Executar o caso de uso interno “Perceber que um aluno finalizou uma prova”	
Cenário Alternativo - Prova finalizada percebida	
Ações do AgentRoleActor	
1. Alterar crença para Necessidade de nova verificação de prova positiva	
2. Tornar o objetivo uma intenção	
3. Para cada uma das questões da prova execute o caso de uso interno “ Avaliar questão”	
4. Inserir na prova a avaliação de cada questão.	
5. Realizar o cálculo de nota do aluno com base na correção realizada no plano.	
6. Informar ao Mediador o resultado da prova.	

Fonte: O Autor

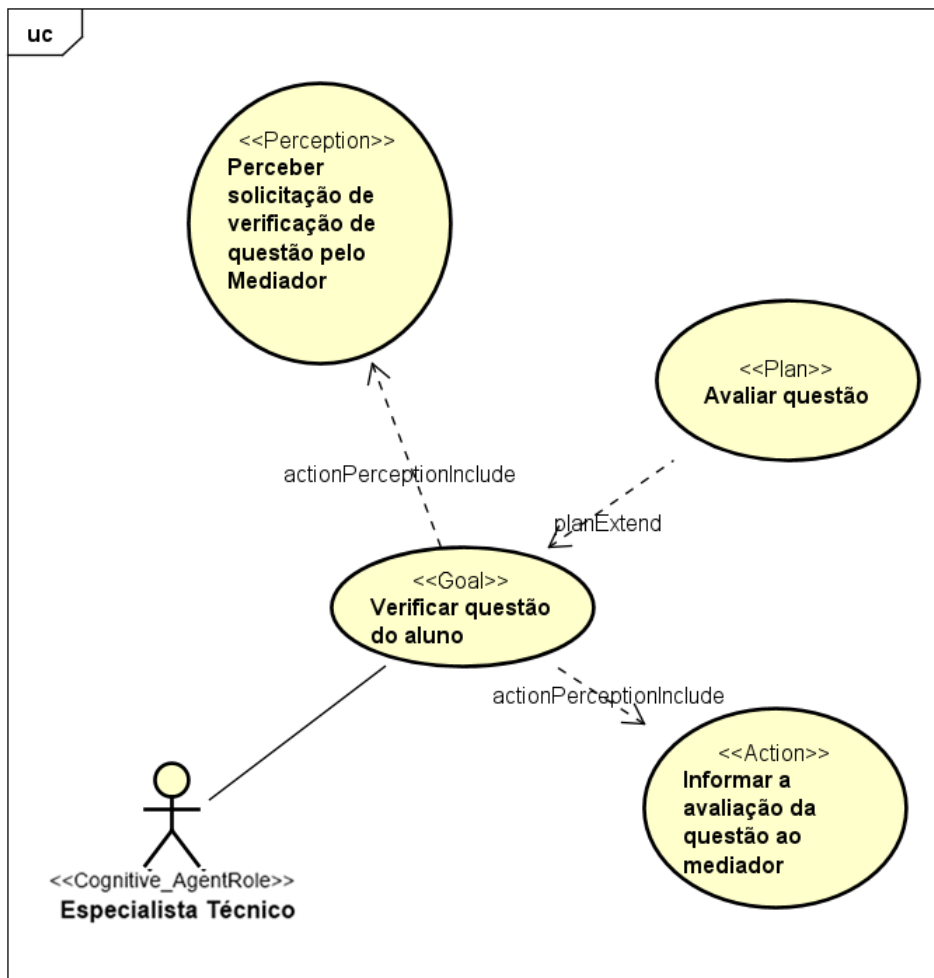
Tabela 6 – Percepção - Perceber que um aluno finalizou a prova

Nome do caso de uso interno	Perceber que um aluno finalizou uma prova
Stereotype	Perception
AgentRoleActor	Especialista Técnico
Resumo	Este caso de uso interno descreve a execução da sondagem do ambiente para verificar se um aluno finalizou uma prova.
Pré-condições	O caso de uso interno Goal “Verificar prova do Aluno” deve estar em execução.
Crenças Iniciais	Aluno está executando a prova.
Cenário principal	
Ações do AgentRoleActor	
1. Sondar o ambiente para verificar se um aluno finalizou uma prova.	
Cenário Alternativo - Percepção Verdadeira	
Ações do AgentRoleActor	
1. Passar a acreditar que o aluno finalizou a prova.	

Fonte: O Autor

O diagrama de casos de uso parcial (Figura 11), apresenta a sequência de funcionalidades do agente Especialista Técnico utilizadas para verificar uma questão de um aluno. O Especialista Técnico percebe a solicitação de verificação de questão, enviada pelo agente Mediador e o Objetivo "Verificar questão do aluno" se torna uma intenção, disparando o plano "Avaliar questão". Após corrigir a questão, o Agente Informa a avaliação da questão ao Agente Mediador. A descrição dos passos para realizar este Objetivo podem ser vistas nas Tabelas 7, 8 e 9.

Figura 11 – Diagrama de Casos de Uso Parcial - Verificar questão do aluno.



Fonte: O Autor

Tabela 7 – Objetivo - Verificar questão do aluno

Nome do caso de uso interno	Verificar questão do aluno
Stereotype	Goal
AgentRoleActor	Especialista Técnico
Resumo	Este caso de uso interno descreve as etapas seguidas pelo agente que assume a função de Especialista Técnico a verificar uma questão desenvolvida pelo aluno.
Crenças Iniciais	Solicitação de uma nova verificação de questão negativa
Percepções	Sondagem de novas solicitações de verificação de questões
Cenário principal	
Ações do AgentRoleActor	
1. Executar o caso de uso interno “Perceber solicitação de verificação de questão”	
Cenário Alternativo - Prova finalizada percebida	
Ações do AgentRoleActor	
1. Alterar crença para Solicitação de uma nova verificação de questão positiva	
2. Tornar o objetivo uma intenção	
3. Executar caso de uso interno “Avaliar questão”	
4. Informar a avaliação da questão ao Mediador	

Fonte: O Autor

Tabela 8 – Percepção - Perceber solicitação de verificação

Nome do caso de uso interno	Perceber solicitação de verificação de questão pelo Mediador
Stereotype	Perception
AgentRoleActor	Especialista Técnico
Resumo	Este caso de uso interno descreve a execução da sondagem do ambiente para verificar se uma nova verificação de questão foi solicitada.
Pré-condições	O caso de uso interno Goal “Verificar Questão do Aluno” deve estar em execução.
Crenças Iniciais	Não houve solicitação de uma nova verificação de questão
Cenário principal	
Ações do AgentRoleActor	
1. Sondar o ambiente para verificar a ocorrência de uma nova solicitação de verificação de questão	
Cenário Alternativo - Percepção Verdadeira	
Ações do AgentRoleActor	
1. Passar a acreditar que uma nova verificação de questão foi solicitada	

Fonte: O Autor

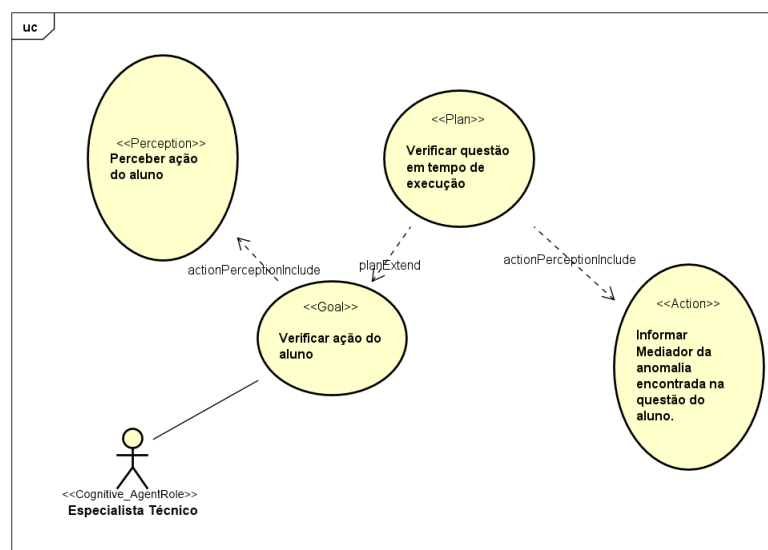
Tabela 9 – Plano - Avaliar questão

Nome do caso de uso interno	Avaliar questão
Stereotype	Plan
AgentRoleActor	Especialista Técnico
Resumo	Este caso de uso interno descreve as etapas seguidas pelo agente que assume a função de Especialista Técnico consultar base de conhecimento para verificar a questão do aluno.
Cenário principal	
Ações do AgentRoleActor	
1. Verificar a base de conhecimento	
2. Encontrar questão sob análise	
3. Comparar a questão correta com a questão do aluno	
4. Verificar questão do aluno procurando possíveis erros de dedução e erros de hipótese.	
5. Descrever uma avaliação da questão com base na comparação e nos possíveis erros encontrados.	

Fonte: O Autor

No caso do diagrama de casos de uso parcial (Figura 12), é apresentado os passos do Especialista Técnico para verificar uma ação do aluno. O Especialista Técnico percebe uma ação do aluno ao resolver uma questão. Deste modo o Objetivo "Verificar ação do aluno" se torna uma intenção, disparando o plano "Verificar questão em tempo de execução". O Especialista Técnico deve informar o Mediador caso encontrar alguma anomalia na ação do aluno. A descrição dos passos dos cenários necessários para executar este Objetivo podem ser vistas nas Tabelas 10, 11 e 12.

Figura 12 – Diagrama de Casos de Uso Parcial - Verificar ação do aluno



Fonte: O Autor

Tabela 10 – Objetivo - Verificar ação do aluno

Nome do caso de uso interno	Verificar ação do aluno
Stereotype	Goal
AgentRoleActor	Especialista Técnico
Resumo	Este caso de uso interno descreve as etapas seguidas pelo agente que assume a função de Especialista Técnico para verificar uma ação do aluno em tempo de execução.
Crenças Iniciais	Necessidade de verificar questão em tempo de execução negativa
Percepções	Sondagem de novas ações do aluno.
Cenário principal	
Ações do AgentRoleActor	
1. Executar o caso de uso interno “Perceber ação do aluno”	
Cenário Alternativo - Prova finalizada percebida	
Ações do AgentRoleActor	
1. Alterar crença para Necessidade de verificar questão em tempo de execução positiva.	
2. Tornar o objetivo uma intenção	
3. Executar caso de uso interno “Verificar questão em tempo de execução”	

Fonte: O Autor

Tabela 11 – Percepção - Perceber ação do aluno

Nome do caso de uso interno	Perceber ação do aluno
Stereotype	Perception
AgentRoleActor	Especialista Técnico
Resumo	Este caso de uso interno descreve a execução da sondagem que o especialista Técnico faz na questão, buscando ações do aluno para verificar a questão em tempo de execução.
Pré-condições	O caso de uso interno Goal “Verificar ação do Aluno” deve estar em execução.
Crenças Iniciais	O aluno não realizou uma ação.
Cenário principal	
Ações do AgentRoleActor	
1. Sondar o ambiente para verificar possíveis ações do aluno em uma questão.	
Cenário Alternativo - Percepção Verdadeira	
Ações do AgentRoleActor	
1. Passar a acreditar que o aluno realizou uma ação.	

Fonte: O Autor

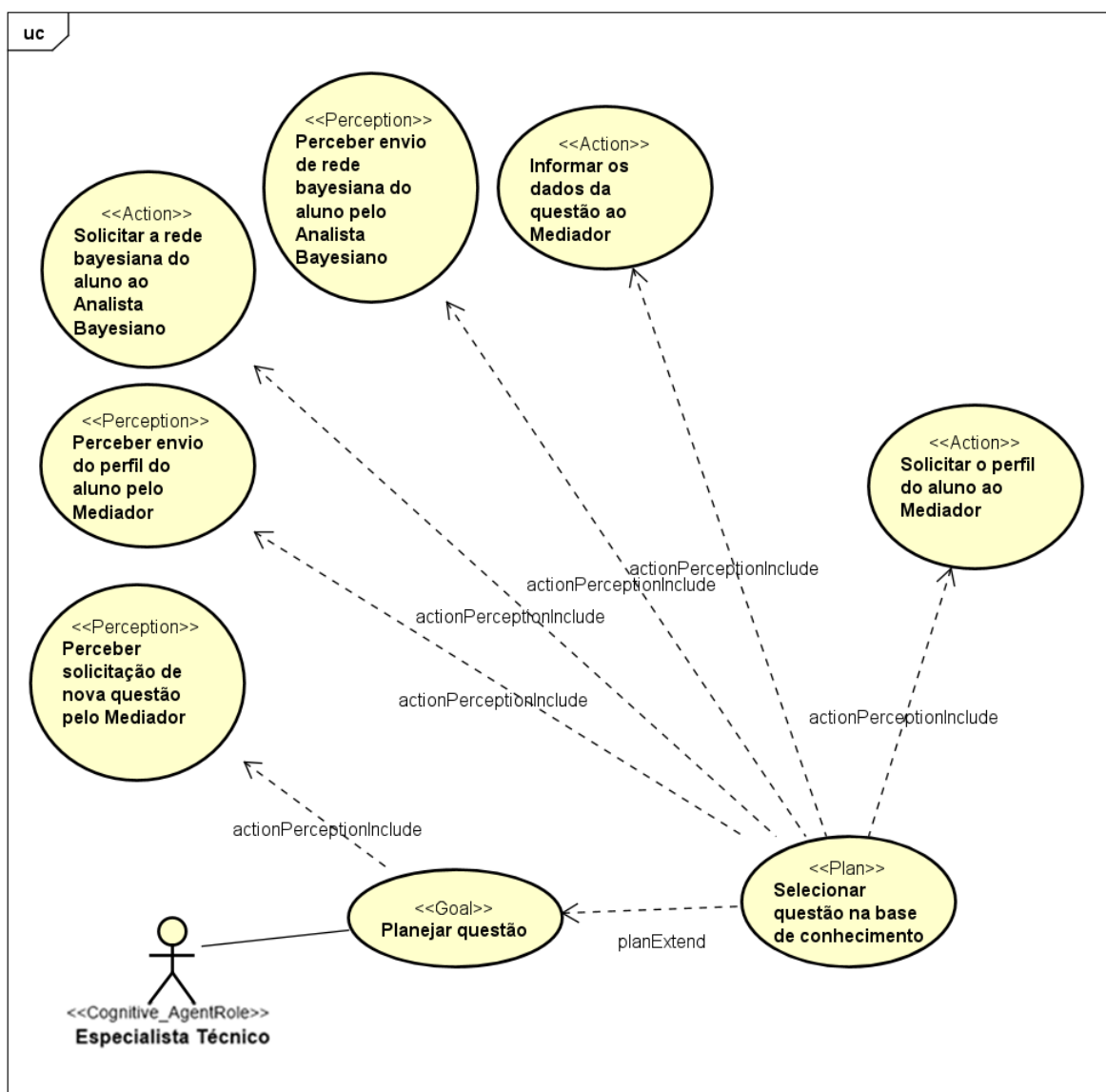
Tabela 12 – Plano - Verificar questão em tempo de execução

Nome do caso de uso interno	Verificar questão em tempo de execução
Stereotype	Plan
AgentRoleActor	Especialista Técnico
Resumo	Este caso de uso interno descreve as etapas seguidas pelo agente que assume a função de Especialista Técnico consultar base de conhecimento para a verificar a questão do aluno em tempo de execução.
Cenário principal	
Ações do AgentRoleActor	
1. Verificar a base de conhecimento	
2. Encontrar questão sob análise	
3. Comparar a questão correta com a questão do aluno	
4. Identificar que questão do aluno está correta e não possui ações perigosas.	
Cenário Alternativo - Regra de dedução incorreta.	
Ações do AgentRoleActor	
1. Identificar que regra de dedução utilizada pelo aluno está incorreta.	
2. Descrever aplicação de regra de dedução incorreta na questão do aluno.	
3. Informar Mediador da anomalia encontrada na questão do aluno.	
Cenário Alternativo - Hipótese incorreta.	
Ações do AgentRoleActor	
1. Identificar que hipótese utilizada pelo aluno está incorreta.	
2. Descrever aplicação de Hipótese incorreta na questão do aluno.	
3. Informar Mediador da anomalia encontrada na questão do aluno.	
Cenário Alternativo - Regra de dedução perigosa.	
Ações do AgentRoleActor	
1. Identificar que regra de dedução utilizada pelo aluno está correta porém para esta questão pode resultar em um erro futuro.	
2. Descrever aplicação de regra de dedução perigosa na questão do aluno.	
3. Informar Mediador da anomalia encontrada na questão do aluno.	

Fonte: O Autor

O diagrama de casos de uso parcial (Figura 13), apresenta os passos necessários para o Especialista Técnico Planejar uma questão para o aluno. O Especialista Técnico percebe a solicitação de nova questão, vinda do Mediador, e o Objetivo "Planejar questão" se torna uma intenção. Com o agente tendo a intenção de planejar uma questão o plano "Selecionar questão na base de conhecimento" é disparado. Após planejar uma questão que seja adequada ao aluno o Especialista Técnico informa os dados da questão ao Mediador. A descrição dos passos para realizar este Objetivo podem ser vistas nas Tabelas 13, 14, 15, 16 e 17.

Figura 13 – Diagrama de Casos de Uso Parcial - Planejar questão.



Fonte: O Autor

Tabela 13 – Objetivo - Planejar questão

Nome do caso de uso interno	Planejar questão
Stereotype	Goal
AgentRoleActor	Especialista Técnico
Resumo	Este caso de uso interno descreve as etapas seguidas pelo agente que assume a função de Especialista Técnico ao planejar as questões destinadas ao aluno.
Crenças Iniciais	Solicitação de uma nova questão negativa.
Percepções	Sondagem de solicitações de nova questão.
Cenário principal	
Ações do AgentRoleActor	
1. Executar o caso de uso interno “Perceber solicitação de nova questão do Mediador”	
Cenário Alternativo - Solicitação de Verificação Percebida	
Ações do AgentRoleActor	
1. Alterar crença para Solicitação de uma nova questão positiva.	
2. Tornar o objetivo uma intenção.	
3. Executar caso de uso interno “ Selecionar questão na base de conhecimento”.	

Fonte: O Autor

Tabela 14 – Percepção - Perceber solicitação de nova questão pelo Mediador

Nome do caso de uso interno	Perceber solicitação de nova questão pelo Mediador
Stereotype	Perception
AgentRoleActor	Especialista Técnico
Resumo	Este caso de uso interno descreve a execução da sondagem do ambiente para verificar se uma nova questão foi solicitada.
Pré-condições	O caso de uso interno Goal “Planejar questão” deve estar em execução.
Crenças Iniciais	Não houve solicitação de um nova questão para o aluno.
Cenário principal	
Ações do AgentRoleActor	
1. Sondar o ambiente para verificar a ocorrência de uma nova solicitação de questão para o aluno.	
Cenário Alternativo - Percepção Verdadeira	
Ações do AgentRoleActor	
1. Passar a acreditar que um nova questão foi solicitado.	
2. Receber questão.	

Fonte: O Autor

Tabela 15 – Plano - Selecionar questão na base de conhecimento

Nome do caso de uso interno	Selecionar questão na base de conhecimento
Stereotype	Plan
AgentRoleActor	Especialista Técnico
Resumo	Este caso de uso interno descreve as etapas seguidas pelo agente que assume a função de Especialista Técnico para selecionar uma questão para um aluno.
Cenário principal	
Ações do AgentRoleActor	
1. Solicitar perfil do aluno ao Mediador	
2. Solicitar rede bayesiana do aluno ao Analista Bayesiano	
3. Executar caso de uso interno “Perceber perfil do aluno”	
4. Executar caso de uso interno “Perceber envio de rede bayesiana do aluno pelo Analista Bayesiano”	
Cenário Alternativo - Envio de rede Bayesiana e de perfil do aluno Percebidos	
Ações do AgentRoleActor	
1. Verificar base de conhecimento	
2. Verificar histórico do aluno presente na rede bayesiana	
3. Filtrar as questões que não foram aplicadas anteriormente ao aluno, com base no histórico do aluno.	
4. Determinar a questão com base no nível do aluno presente em seu perfil	
5. Atribuir a complexidade da questão com base no histórico do aluno e nas informações presentes na questão.	

Fonte: O Autor

Tabela 16 – Percepção - Perceber envio do perfil do aluno pelo Mediador

Nome do caso de uso interno	Perceber envio do perfil do aluno pelo Mediador
Stereotype	Perception
AgentRoleActor	Especialista Técnico
Resumo	Este caso de uso interno descreve a execução da sondagem do ambiente para verificar se o perfil do aluno foi enviado pelo Mediador.
Pré-condições	O caso de uso interno Plan “Selecionar questão na base de conhecimento” deve estar em execução.
Crenças Iniciais	Perfil não enviado.
Cenário principal	
Ações do AgentRoleActor	
1. Sondar o ambiente para verificar a ocorrência de um novo envio de perfil do aluno por parte do Mediador.	
Cenário Alternativo - Percepção Verdadeira	
Ações do AgentRoleActor	
1. Passar a creditar que um perfil de aluno foi enviado	
2. Receber o perfil do aluno.	

Fonte: O Autor

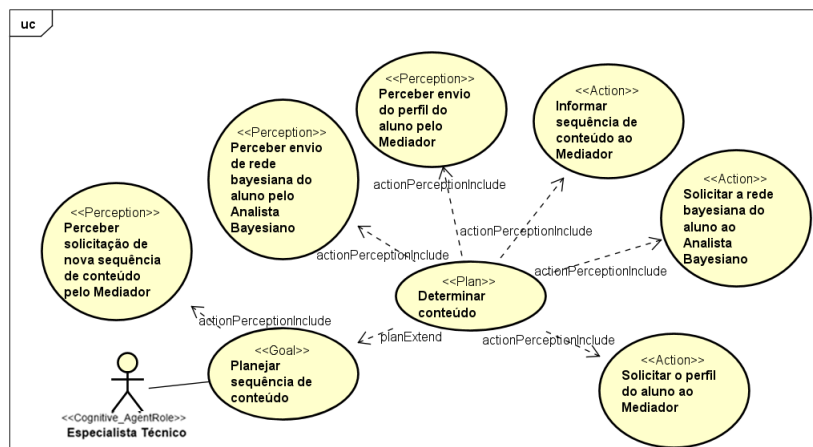
Tabela 17 – Percepção - Perceber envio de rede bayesiana do aluno pelo Analista

Nome do caso de uso interno	Perceber envio de rede bayesiana do aluno pelo Analista Bayesiano
Stereotype	Perception
AgentRoleActor	Especialista Técnico
Resumo	Este caso de uso interno descreve a execução da sondagem do ambiente para verificar se a rede bayesiana do aluno foi enviada pelo Analista Bayesiano.
Pré-condições	O caso de uso interno Plan “Selecionar questão na base de conhecimento” deve estar em execução.
Crengas Iniciais	Perfil não enviado.
Cenário principal	
Ações do AgentRoleActor	
1. Sondar o ambiente para verificar a ocorrência de um novo envio de rede bayesiana do aluno por parte do Analista Bayesiano.	
Cenário Alternativo - Percepção Verdadeira	
Ações do AgentRoleActor	
1. Passar acreditar que uma rede bayesiana foi enviada	
2. Receber rede bayesiana do aluno.	

Fonte: O Autor

O diagrama de casos de uso parcial (Figura 14), apresenta as etapas necessárias para o Agente Especialista Técnico, planejar uma sequência de conteúdo para o aluno. O agente percebe uma solicitação de nova sequência de conteúdo por parte do Mediador, tornando o Objetivo "Planejar sequência de conteúdo" uma intenção. Deste modo o plano "Determinar Conteúdo" é disparado. Após o agente determinar o conteúdo, ele informa ao Mediador a sequência de conteúdo do aluno. A descrição dos passos dos cenários necessários para realizar este Objetivo podem ser vistas nas Tabelas 18, 19, 20, 17 e 16.

Figura 14 – Diagrama de Casos de Uso Parcial - Planejar sequência de conteúdo



Fonte: O Autor

Tabela 18 – Objetivo - Planejar sequência de conteúdo

Nome do caso de uso interno	Planejar sequência de conteúdo
Stereotype	Goal
AgentRoleActor	Especialista Técnico
Resumo	Este caso de uso interno descreve as etapas seguidas pelo agente que assume a função de Especialista Técnico para planejar a sequência de conteúdo que melhor se adequa ao aluno.
Crenças Iniciais	Solicitação de uma nova sequência de conteúdo negativa.
Percepções	Sondagem de novas solicitações de sequência de conteúdo.
Cenário principal	
Ações do AgentRoleActor	
1. Executar o caso de uso interno “Perceber solicitação de nova sequência de conteúdo”	
Cenário Alternativo - Solicitação de Verificação Percebida	
Ações do AgentRoleActor	
1. Alterar crença para Solicitação de uma nova sequência de conteúdo positiva.	
2. Tornar o objetivo uma intenção.	
3. Executar caso de uso interno “Determinar conteúdo”	

Fonte: O Autor

Tabela 19 – Percepção - Perceber solicitação de nova sequência de conteúdo

Nome do caso de uso interno	Perceber solicitação de nova sequência de conteúdo pelo Mediador
Stereotype	Perception
AgentRoleActor	Especialista Técnico
Resumo	Este caso de uso interno descreve a execução da sondagem do ambiente para verificar se uma nova sequência de conteúdo foi solicitada.
Pré-condições	O caso de uso interno Goal “Planejar sequência de conteúdo” deve estar em execução
Crenças Iniciais	Não houve solicitação de uma nova sequência de conteúdo pelo Mediador
Cenário principal	
Ações do AgentRoleActor	
1. Sondar o ambiente para verificar a ocorrência de uma nova solicitação de sequência de conteúdo para um aluno.	
Cenário Alternativo - Percepção Verdadeira	
Ações do AgentRoleActor	
1. Passar a acreditar que uma nova sequência de conteúdo foi solicitada.	

Fonte: O Autor

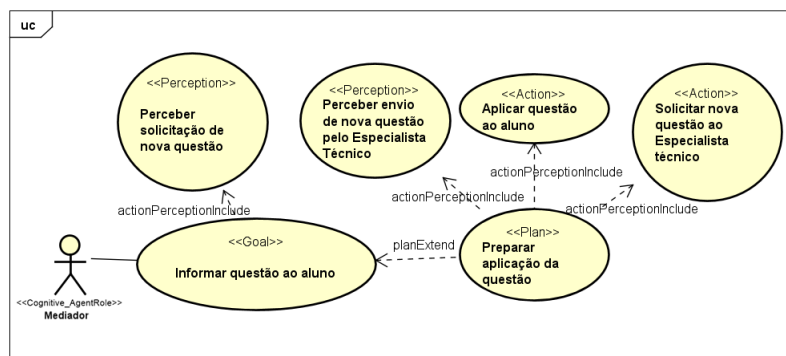
Tabela 20 – Plano - Determinar conteúdo

Nome do caso de uso interno	Determinar conteúdo
Stereotype	Plan
AgentRoleActor	Especialista Técnico
Resumo	Este caso de uso interno descreve as etapas seguidas pelo agente que assume a função de Especialista Técnico para selecionar uma questão para determinar o conteúdo de estudo do aluno.
Cenário principal	
Ações do AgentRoleActor	
1. Solicitar o perfil do aluno ao Mediador	
2. Solicitar rede bayesiana do aluno ao Analista Bayesiano	
3. Executar caso de uso interno “Perceber perfil do aluno”	
4. Executar caso de uso interno “Perceber envio de rede bayesiana do aluno pelo Analista Bayesiano”	
Cenário Alternativo - Envio de rede Bayesiana e de perfil do aluno Percebidos	
Ações do AgentRoleActor	
1. Verificar a base de conhecimento	
2. Verificar o histórico do aluno presente na rede bayesiana	
3. Separar conteúdos que melhor se adequem ao perfil e ao histórico do aluno.	
4. Determinar sequência que beneficie melhor o aprendizado desses conteúdos	
5. Enviar sequência de conteúdo ao Mediador	

Fonte: O Autor

O diagrama de casos de uso parcial (Figura 15), apresenta os passos necessários para o Agente Mediador “Informar questão ao aluno”. O Agente percebe uma solicitação de nova questão e o objetivo “Informar questão ao aluno” se torna uma intenção. Deste modo o plano “Preparar aplicação da questão” é disparado, terminando com a aplicação da questão ao aluno. A descrição dos passos dos cenários necessários para realizar este Objetivo podem ser vistas nas Tabelas 21, 22, 23 e 24.

Figura 15 – Diagrama de Casos de Uso Parcial - Informar questão ao aluno.



Fonte: O Autor

Tabela 21 – Objetivo - Informar questão ao aluno

Nome do caso de uso interno	Informar questão ao aluno
Stereotype	Goal
AgentRoleActor	Mediador
Resumo	Este caso de uso interno descreve as etapas seguidas pelo agente que assume a função de Mediador para informar nova questão de estudo ao aluno.
Crenças Iniciais	Solicitação de uma nova questão negativa
Percepções	Sondagem de novas solicitações de questões
Cenário principal	
Ações do AgentRoleActor	
1. Executar o caso de uso interno “Perceber solicitação de nova questão”	
Cenário Alternativo - Solicitação de nova questão Percebida	
Ações do AgentRoleActor	
1. Alterar crença para Solicitação de uma nova questão positiva	
2. Tornar o objetivo uma intenção.	
3. Executar caso de uso interno “Preparar aplicação da questão”	

Fonte: O Autor

Tabela 22 – Percepção - Perceber solicitação de nova questão

Nome do caso de uso interno	Perceber solicitação de nova questão
Stereotype	Perception
AgentRoleActor	Mediador
Resumo	Este caso de uso interno descreve a execução da sondagem do ambiente para verificar se um aluno solicitou nova questão de estudo.
Pré-condições	O caso de uso interno Goal “Informar questão ao aluno” conteúdo” deve estar em execução
Crenças Iniciais	Não houve solicitação de nova questão.
Cenário principal	
Ações do AgentRoleActor	
1. Sondar o ambiente para verificar a ocorrência de uma nova solicitação de questão.	
Cenário Alternativo - Percepção Verdadeira	
Ações do AgentRoleActor	
1. Passar a acreditar que um aluno solicitou nova questão.	

Fonte: O Autor

Tabela 23 – Percepção - Perceber envio de nova questão do Especialista Técnico

Nome do caso de uso interno	Perceber envio de nova questão do Especialista Técnico
Stereotype	Perception
AgentRoleActor	Mediador
Resumo	Este caso de uso interno descreve a execução da sondagem do ambiente para verificar se o Especialista Técnico enviou uma nova questão.
Pré-condições	O caso de uso interno Plan “Preparar aplicação da questão” deve estar em execução
Crenças Iniciais	O Especialista Técnico não enviou uma nova questão.
Cenário principal	
Ações do AgentRoleActor	
<ol style="list-style-type: none"> 1. Sondar o ambiente para verificar se uma nova questão do aluno foi enviada pelo 2. Especialista Técnico. 	
Cenário Alternativo - Nova questão percebida.	
Ações do AgentRoleActor	
<ol style="list-style-type: none"> 1. Passar a acreditar que o Especialista Técnico enviou uma nova questão. 2. Receber nova questão do Especialista Técnico. 	

Fonte: O Autor

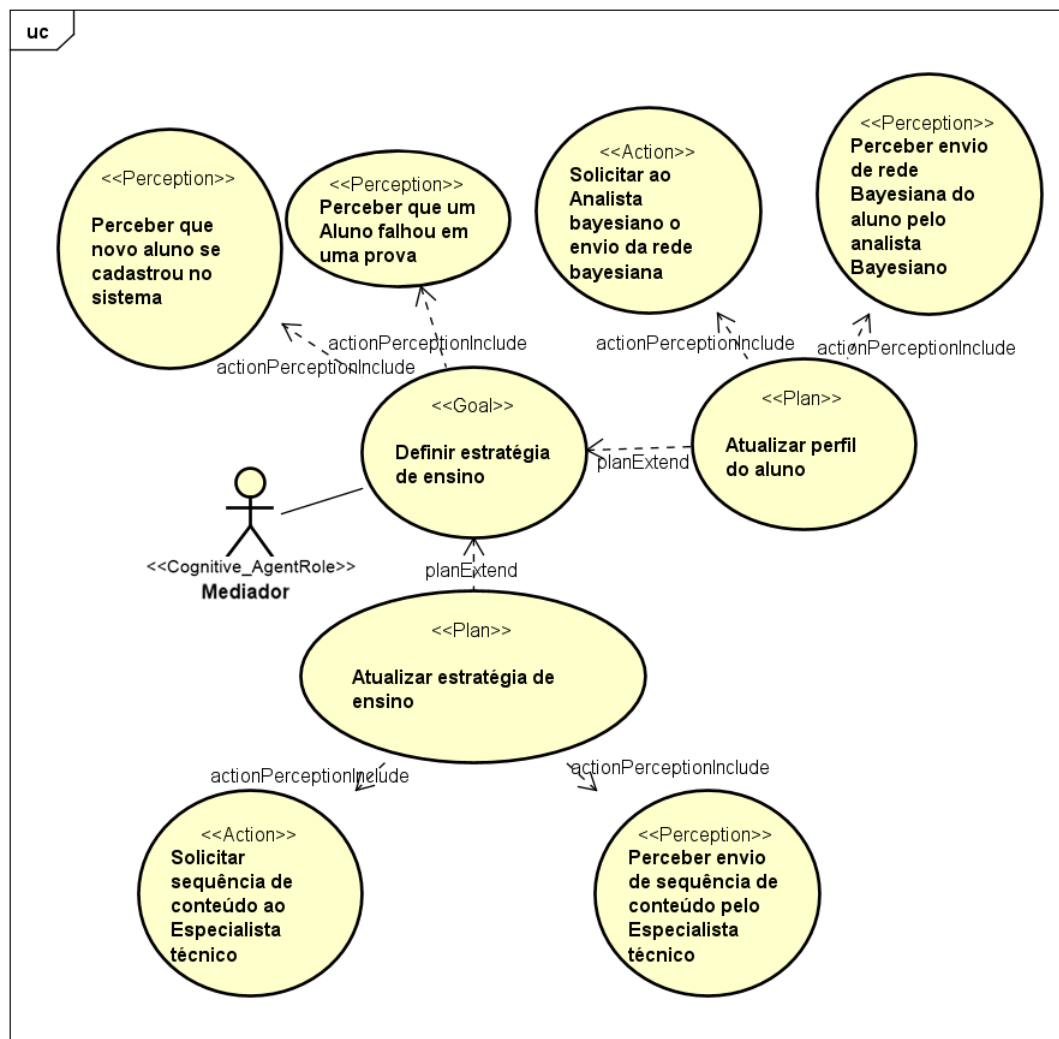
Tabela 24 – Plano - Preparar aplicação da questão

Nome do caso de uso interno	Preparar aplicação da questão
Stereotype	Plan
AgentRoleActor	Mediador
Resumo	Este caso de uso interno descreve as etapas seguidas pelo agente que assume a função de Mediador para solicitar a questão ao Especialista Técnico para possibilitar a aplicação da mesma.
Cenário principal	
Ações do AgentRoleActor	
<ol style="list-style-type: none"> 1. Solicitar nova questão ao Especialista Técnico 2. Executar caso de uso interno “Perceber envio de nova questão pelo Especialista Técnico” 	
Cenário Alternativo - Envio de nova questão do aluno percebida	
Ações do AgentRoleActor	
<ol style="list-style-type: none"> 1. Aplicar questão ao aluno. 	

Fonte: O Autor

No diagrama de casos de uso parcial (Figura 16), podemos observar os passos necessários para o Agente Mediador definir estratégia de ensino. O objetivo "Definir estratégia de ensino" passa a ser uma intenção quando o aluno se cadastra no sistema, ou quando o aluno falha em uma prova. Quando isso acontece o perfil do aluno é atualizado e o plano "Atualizar estratégia de ensino" é disparado. A descrição dos passos dos cenários necessários para executar este Objetivo podem ser vistas nas Tabelas 25, 26, 27, 28, 29, 30 e 32.

Figura 16 – Diagrama de Casos de Uso Parcial - Definir estratégia de ensino.



Fonte: O Autor

Tabela 25 – Objetivo - Definir estratégia de ensino

Nome do caso de uso interno	Definir estratégia de ensino
Stereotype	Goal
AgentRoleActor	Mediador
Resumo	Este caso de uso interno descreve as etapas seguidas pelo agente que assume a função de Mediador para definir uma estratégia de ensino.
Crenças Iniciais	Necessidade de definir estratégias de ensino negativa
Percepções	Sondagem de cadastro de novo aluno no sistema
Cenário principal	
Ações do AgentRoleActor	
1. Executar o caso de uso interno “Perceber que novo aluno se cadastrou no sistema”	
2. Executar o caso de uso interno “Perceber que um Aluno falhou em uma prova”	
Cenário Alternativo - Cadastro de novo aluno percebido	
Ações do AgentRoleActor	
1. Alterar crença para Necessidade de definir estratégias de ensino positiva	
2. Tornar o objetivo uma intenção.	
3. Atribuir uma estratégia padrão de ensino ao novo aluno.	
Cenário Alternativo - Falha de aluno em prova percebida	
Ações do AgentRoleActor	
1. Alterar crença para Necessidade de definir estratégias de ensino positiva	
2. Tornar o objetivo uma intenção.	
3. Executar caso de uso interno “Atualizar perfil do aluno”	
4. Executar caso de uso interno “Atualizar estratégia de ensino”	

Fonte: O Autor

Tabela 26 – Percepção - Perceber que novo aluno se cadastrou no sistema

Nome do caso de uso interno	Perceber que novo aluno se cadastrou no sistema
Stereotype	Perception
AgentRoleActor	Mediador
Resumo	Este caso de uso interno descreve a execução da sondagem do ambiente para verificar se um novo aluno se cadastrou no sistema.
Pré-condições	O caso de uso interno Goal “Definir estratégia de ensino” deve estar em execução
Crenças Iniciais	Não houve novo cadastro de aluno no sistema.
Cenário principal	
Ações do AgentRoleActor	
1. Sondar o ambiente para verificar a ocorrência de um novo cadastro de aluno.	
Cenário Alternativo - Percepção Verdadeira.	
Ações do AgentRoleActor	
1. Passar a acreditar que um novo aluno se cadastrou no sistema.	

Fonte: O Autor

Tabela 27 – Percepção - Perceber envio de sequência de conteúdo pelo Especialista Técnico

Nome do caso de uso interno	Perceber envio de sequência de conteúdo pelo Especialista Técnico
Stereotype	Perception
AgentRoleActor	Mediador
Resumo	Este caso de uso interno descreve a execução da sondagem do ambiente para verificar se o Especialista Técnico enviou uma nova sequência de conteúdo.
Pré-condições	O caso de uso interno Plan “Atualizar estratégia de ensino” deve estar em execução ou o caso de uso interno Plan “Preparar reforço” deve estar em execução.
Crenças Iniciais	Não houve envio de nova sequência de conteúdo pelo Especialista Técnico.
Cenário principal	
Ações do AgentRoleActor	
1. Sondar o ambiente para verificar se uma nova sequência de conteúdo do aluno foi enviada pelo Especialista Técnico.	
Cenário Alternativo - Sequência de conteúdo percebida.	
Ações do AgentRoleActor	
1. Passar a acreditar que nova sequência foi enviada pelo Especialista Técnico	
2. Receber nova sequência de conteúdo do aluno.	

Fonte: O Autor

Tabela 28 – Percepção - Perceber envio de rede bayesiana do aluno pelo Analista Bayesiano

Nome do caso de uso interno	Perceber envio de rede bayesiana do aluno pelo Analista Bayesiano
Stereotype	Perception
AgentRoleActor	Mediador
Resumo	Este caso de uso interno descreve a execução da sondagem do ambiente para verificar se a rede bayesiana foi enviada pelo Analista bayesiano.
Pré-condições	O caso de uso interno Plan “Definir perfil do aluno” deve estar em execução
Crenças Iniciais	Aluno não solicitou ajuda.
Cenário principal	
Ações do AgentRoleActor	
1. Sondar o ambiente para verificar a ocorrência de um novo envio de rede bayesiana por parte do Analista bayesiano.	
Cenário Alternativo - Percepção Verdadeira.	
Ações do AgentRoleActor	
1. Passar a acreditar que uma nova rede bayesiana foi enviada.	
2. Receber rede bayesiana.	

Fonte: O Autor

Tabela 29 – Plano - Atualizar perfil do aluno

Nome do caso de uso interno	Atualizar perfil do aluno
Stereotype	Plan
AgentRoleActor	Mediador
Resumo	Este caso de uso interno descreve as etapas seguidas pelo agente que assume a função de Mediador para Atualizar o perfil do aluno.
Cenário principal	
Ações do AgentRoleActor	
1. Solicitar ao Analista bayesiano o envio da rede bayesiana do aluno	
2. Executar caso de uso interno “Perceber envio de rede bayesiana do aluno pelo Analista Bayesiano”.	
Cenário Alternativo - Envio de rede bayesiana percebido	
Ações do AgentRoleActor	
1. Verificar as questões de perfil do aluno respondidas ao se cadastrar no sistema.	
2. Verificar disciplinas cursadas pelo aluno	
3. Verificar a área de atuação do aluno	
4. Determinar o foco do aluno com base no conteúdo necessário na área de atuação e no conhecimento esperado nas disciplinas que o aluno já cursou.	
5. Estimar o conhecimento do aluno com base na rede bayesiana	
6. Determinar um perfil com base no foco do aluno e na estimativa de conhecimento do aluno.	

Fonte: O Autor

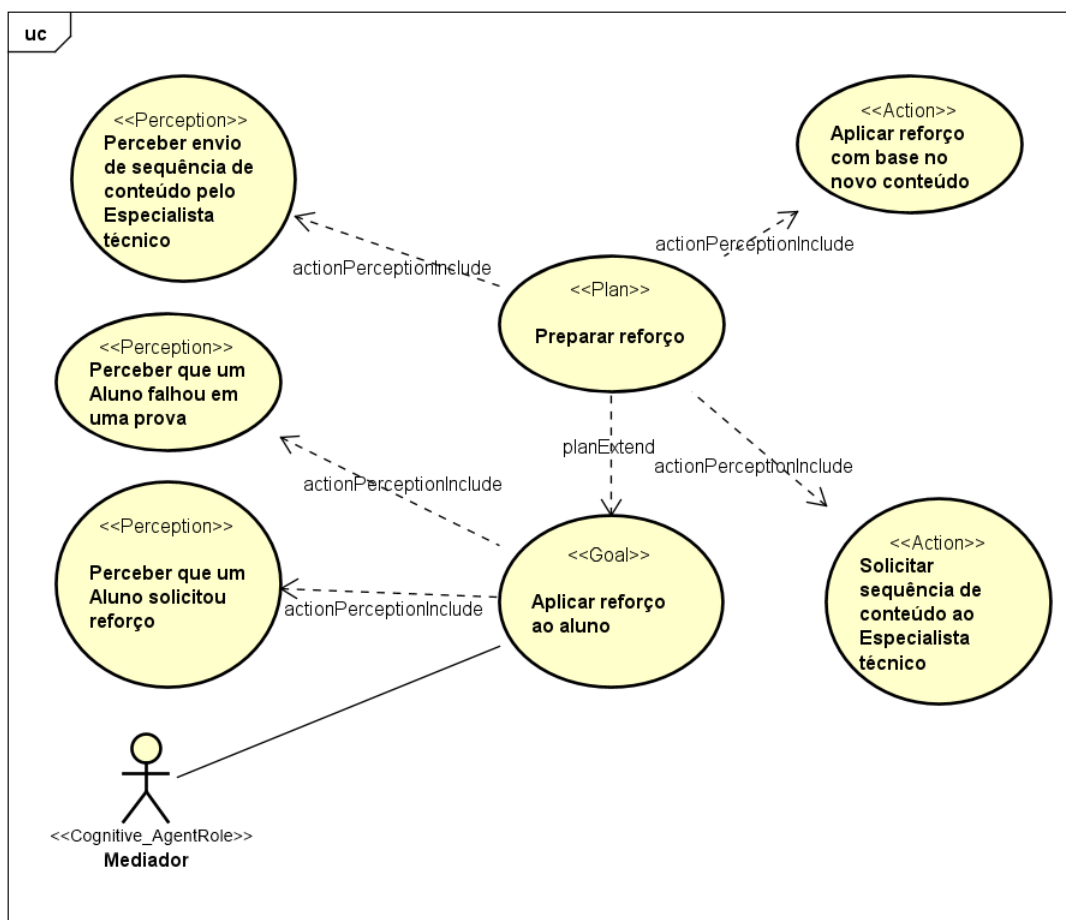
Tabela 30 – Plano - Atualizar estratégia de ensino

Nome do caso de uso interno	Atualizar estratégia de ensino
Stereotype	Plan
AgentRoleActor	Mediador
Resumo	Este caso de uso interno descreve as etapas seguidas pelo agente que assume a função de Mediador para Atualizar estratégia de ensino.
Cenário principal	
Ações do AgentRoleActor	
1. Solicitar sequência de conteúdo ao Especialista Técnico	
2. Executar caso de uso interno “Perceber envio de sequência de conteúdo pelo Especialista Técnico”.	
Cenário Alternativo - Envio de sequência de conteúdo percebido	
Ações do AgentRoleActor	
1. Verificar o perfil do aluno	
2. Verificar se o aluno está tendo rendimento ao responder as questões	
3. Verificar histórico de provas do aluno	
4. Atribuir ao aluno uma estratégia de ensino atualizada.	

Fonte: O Autor

O diagrama de casos de uso parcial (Figura 17), apresenta os passos necessários para o Agente Mediador aplicar reforço ao aluno. O objetivo "Aplicar reforço ao aluno" passa a ser uma intenção quando um aluno falha em uma prova ou quando o aluno solicita reforço. Quando o objetivo se tornar uma intenção o plano "Preparar reforço" é disparado resultando na aplicação do reforço ao aluno. A descrição dos passos para realizar este Objetivo podem ser vistas nas Tabelas 31, 27, 32, 33, e 34.

Figura 17 – Diagrama de Casos de Uso Parcial - Aplicar reforço ao aluno.



Fonte: O Autor

Tabela 31 – Objetivo - Aplicar reforço ao aluno

Nome do caso de uso interno	Aplicar reforço ao aluno
Stereotype	Goal
AgentRoleActor	Mediador
Resumo	Este caso de uso interno descreve as etapas seguidas pelo agente que assume a função de Mediador para aplicar reforço ao aluno.
Crenças Iniciais	Necessidade de aplicar reforço negativa
	Sondagem de falha em uma prova
Percepções	Sondagem de pedido de ajuda
Cenário principal	
Ações do AgentRoleActor	
	1. Executar o caso de uso interno “Perceber que um Aluno falhou em uma prova”
	2. Executar o caso de uso interno “Perceber que um Aluno solicitou reforço”
Cenário Alternativo - Falha em prova percebida	
Ações do AgentRoleActor	
	1. Alterar crença para Necessidade de aplicar reforço positiva
	2. Tornar o objetivo uma intenção.
	3. Executar caso de uso interno “Preparar reforço”.
Cenário Alternativo - Pedido de reforço percebida	
Ações do AgentRoleActor	
	1. Alterar crença para Necessidade de aplicar reforço positiva
	2. Tornar o objetivo uma intenção.
	3. Executar caso de uso interno “Preparar reforço”

Fonte: O Autor

Tabela 32 – Percepção - Perceber que um Aluno falhou em uma prova

Nome do caso de uso interno	Perceber que um Aluno falhou em uma prova
Stereotype	Perception
AgentRoleActor	Mediador
Resumo	Este caso de uso interno descreve a execução da sondagem do ambiente para verificar se algum aluno falhou em uma prova.
Pré-condições	O caso de uso interno Goal “Aplicar reforço ao aluno” deve estar em execução.
Crenças Iniciais	Nenhum aluno falhou em uma prova
Cenário principal	
Ações do AgentRoleActor	
	1. Sondar o ambiente para verificar se o resultado da prova informado pelo Especialista Técnico é de falha.
Cenário Alternativo - Percepção Verdadeira.	
Ações do AgentRoleActor	
	1. Passar a acreditar que um aluno falhou em uma prova

Fonte: O Autor

Tabela 33 – Percepção - Perceber que um Aluno solicitou reforço

Nome do caso de uso interno	Perceber que um Aluno solicitou reforço
Stereotype	Perception
AgentRoleActor	Mediador
Resumo	Este caso de uso interno descreve a execução da sondagem do ambiente para verificar se algum aluno solicitou reforço
Pré-condições	O caso de uso interno Goal “Aplicar reforço ao aluno” deve estar em execução.
Crenças Iniciais	Nenhum aluno falhou em uma prova
Cenário principal	
Ações do AgentRoleActor	
1. Sondar o ambiente para verificar se ocorreu algum pedido de reforço por parte do aluno.	
Cenário Alternativo - Percepção Verdadeira.	
Ações do AgentRoleActor	
1. Passar a acreditar que um aluno solicitou reforço.	

Fonte: O Autor

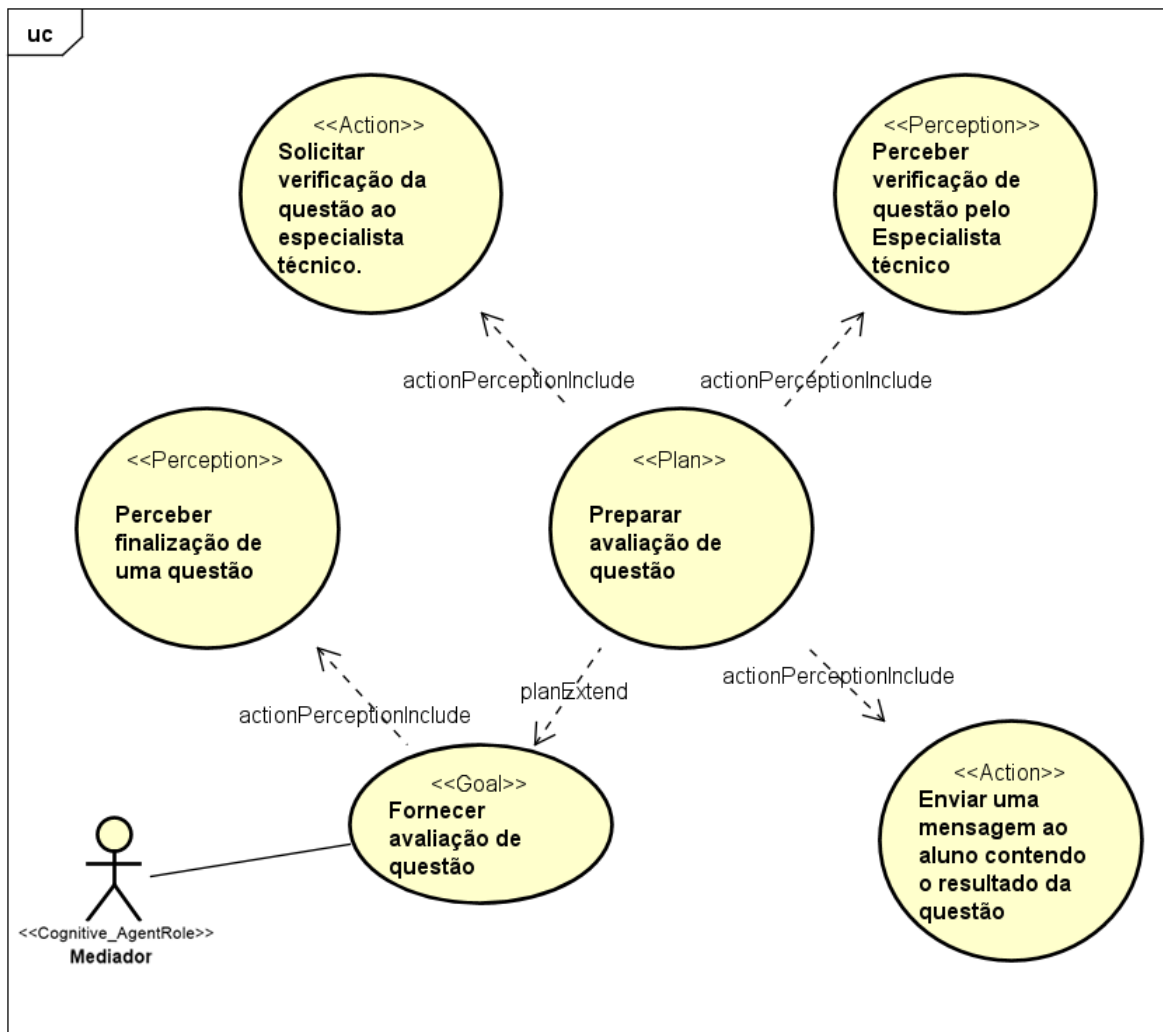
Tabela 34 – Plano - Preparar reforço

Nome do caso de uso interno	Preparar reforço
Stereotype	Plan
AgentRoleActor	Mediador
Resumo	Este caso de uso interno descreve as etapas seguidas pelo agente que assume a função de Mediador para preparar o reforço do aluno.
Cenário principal	
Ações do AgentRoleActor	
1. Solicitar nova sequência de conteúdo ao Especialista Técnico	
2. Executar caso de uso interno “Perceber envio de sequência de conteúdo pelo Especialista Técnico”.	
Cenário Alternativo - Envio de sequência de conteúdo percebido	
Ações do AgentRoleActor	
1. Aplicar reforço com base no novo conteúdo	

Fonte: O Autor

No diagrama de casos de uso parcial (Figura 18), podemos observar os passos do Agente Mediador para fornecer uma avaliação de questão ao aluno. O agente percebe a finalização de uma questão, tornando o objetivo "Fornecer avaliação de questão" uma intenção. Quando o objetivo se torna uma intenção, o plano "Preparar avaliação de questão" é disparado, resultando no envio de mensagem ao aluno contendo o resultado da questão. A descrição dos passos para realizar este Objetivo podem ser vistas nas Tabelas 35, 36, 37 e 38.

Figura 18 – Diagrama de Casos de Uso Parcial - Fornecer avaliação de questão



Fonte: O Autor

Tabela 35 – Objetivo - Fornecer avaliação de questão

Nome do caso de uso interno	Fornecer avaliação de questão
Stereotype	Goal
AgentRoleActor	Mediador
Resumo	Este caso de uso interno descreve as etapas seguidas pelo agente que assume a função de Mediador para retornar o resultado da questão ao aluno.
Crenças Iniciais	Necessidade de fornecer resultado de questão negativa
Percepções	Sondagem de término de questão
Cenário principal	
Ações do AgentRoleActor	
1. Executar o caso de uso interno “Perceber finalização de uma questão”	
Cenário Alternativo - Finalização de uma questão percebida	
Ações do AgentRoleActor	
1. Alterar crença para necessidade de fornecer resultado de questão positiva.	
2. Tornar o objetivo uma intenção.	
3. Executar o caso de uso interno “Preparar avaliação de questão”.	

Fonte: O Autor

Tabela 36 – Percepção - Perceber finalização de uma questão

Nome do caso de uso interno	Perceber finalização de uma questão
Stereotype	Perception
AgentRoleActor	Mediador
Resumo	Este caso de uso interno descreve a execução da sondagem do ambiente para verificar a finalização de uma questão
Pré-condições	O caso de uso interno Goal “Fornecer avaliação de questão” deve estar em execução.
Crenças Iniciais	Nenhuma questão foi finalizada
Cenário principal	
Ações do AgentRoleActor	
1. Sondar o ambiente para verificar a finalização de uma questão	
Cenário Alternativo - Percepção Verdadeira	
Ações do AgentRoleActor	
1. Passar a acreditar que uma questão foi finalizada	

Fonte: O Autor

Tabela 37 – Percepção - Perceer verificação de questão pelo Especialista

Nome do caso de uso interno	Perceber verificação de questão pelo Especialista Técnico
Stereotype	Perception
AgentRoleActor	Mediador
Resumo	Este caso de uso interno descreve a execução da sondagem do ambiente buscando perceber se o Especialista Técnico já realizou a verificação de uma questão do aluno.
Pré-condições	O caso de uso interno Plan “Preparar avaliação de questão” deve estar em execução ou o caso de uso interno Plan “Preparar ajuda do aluno” deve estar em execução.
Crenças Iniciais	Nenhuma verificação de questão recebida
Cenário principal	
Ações do AgentRoleActor	
1. Sondar o ambiente para verificar se o Especialista Técnico informou o resultado de uma questão.	
Cenário Alternativo - Percepção Verdadeira	
Ações do AgentRoleActor	
1. Passar a acreditar que uma nova verificação foi recebida.	
2. Receber resultado de questão.	

Fonte: O Autor

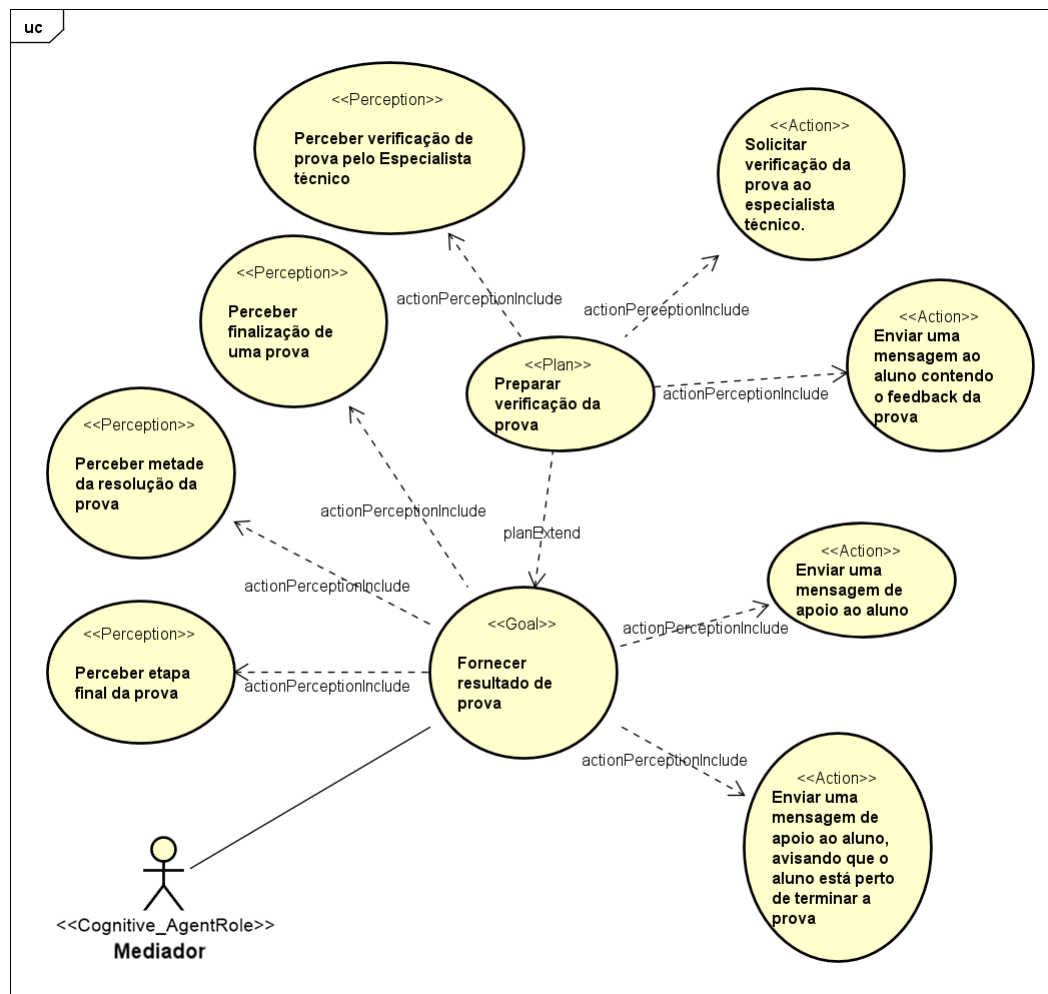
Tabela 38 – Plano - Preparar avaliação de questão

Nome do caso de uso interno	Preparar avaliação de questão
Stereotype	Plan
AgentRoleActor	Mediador
Resumo	Este caso de uso interno descreve as etapas seguidas pelo agente que assume a função de Mediador ao planejar a avaliação de questão.
Cenário principal	
Ações do AgentRoleActor	
1. Solicitar verificação da questão ao Especialista Técnico.	
2. Executar caso de uso interno “ Perceber verificação de questão pelo Especialista Técnico”	
Cenário Alternativo - Verificação de questão percebida	
Ações do AgentRoleActor	
1. Passar a acreditar que uma nova avaliação de questão foi enviada	
2. Verificar perfil atual do aluno.	
3. Determinar o modo de apresentar o resultado da questão com base no perfil do aluno e no resultado da questão.	
4. Enviar uma mensagem ao aluno contendo o resultado da atividade	

Fonte: O Autor

Os passos dos cenários necessários para fornecer o resultado de uma prova ao aluno são apresentados no diagrama de casos de uso parcial (Figura 19). Este diagrama apresenta os passos necessários para apresentar mensagens de apoio durante a execução de uma prova e os passos necessários para verificar uma prova e apresentar o resultado ao aluno. A descrição dos cenários para executar este Objetivo podem ser vistas nas Tabelas 39, 40, 41, 42, 43 e 44

Figura 19 – Diagrama de Casos de Uso Parcial - Fornecer resultado de prova.



Fonte: O Autor

Tabela 39 – Objetivo - Fornecer resultado de prova

Nome do caso de uso interno	Fornecer resultado de prova
Stereotype	Goal
AgentRoleActor	Mediador
Resumo	Este caso de uso interno descreve as etapas seguidas pelo agente que assume a função de Mediador para retornar o resultado ao aluno com relação a uma prova.
Crenças Iniciais	Necessidade de fornecer resultado de prova negativa
Percepções	Verificar se um aluno encontra-se na última questão da prova. Verificar se um aluno concluiu uma prova Verificar se um aluno encontra-se na metade da prova.
Cenário principal	
Ações do AgentRoleActor	
1. Executar o caso de uso interno “Perceber finalização de prova”	
2. Executar o caso de uso interno “Perceber metade da resolução da prova”	
3. Executar o caso de uso interno “Perceber etapa final da prova”	
Cenário Alternativo - Finalização de prova percebida	
Ações do AgentRoleActor	
1. Alterar crença para necessidade de fornecer resultado positiva.	
2. Tornar o objetivo uma intenção.	
3. Executar o caso de uso interno “Preparar verificação da prova”	
Cenário Alternativo - Metade da resolução da prova percebida	
Ações do AgentRoleActor	
1. Alterar crença para necessidade de fornecer resultado positiva.	
2. Tornar o objetivo uma intenção.	
3. Enviar uma mensagem de apoio ao aluno.	
Cenário Alternativo - Etapa final da prova percebida	
Ações do AgentRoleActor	
1. Alterar crença para necessidade de fornecer resultado positiva.	
2. Tornar o objetivo uma intenção.	
3. Enviar uma mensagem de apoio ao aluno, avisando que o aluno está perto de terminar a prova.	

Fonte: O Autor

Tabela 40 – Percepção - Perceber verificação de prova pelo Especialista Técnico

Nome do caso de uso interno	Perceber verificação de prova pelo Especialista Técnico
Stereotype	Perception
AgentRoleActor	Mediador
Resumo	Este caso de uso interno descreve a execução da sondagem do ambiente para verificar o Especialista Técnico já realizou a verificação de uma prova do aluno.
Pré-condições	O caso de uso interno Plan “Preparar verificação da prova” deve estar em execução
Crenças Iniciais	Nenhuma verificação de prova recebida
Cenário principal	
Ações do AgentRoleActor	
1. Sondar o ambiente para verificar se o Especialista Técnico informou o resultado de uma prova.	
Cenário Alternativo - Percepção Verdadeira.	
Ações do AgentRoleActor	
1. Passar a acreditar que uma verificação de prova foi recebida. 2. Receber resultado de prova.	

Fonte: O Autor

Tabela 41 – Percepção - Perceber finalização de uma prova

Nome do caso de uso interno	Perceber finalização de uma prova
Stereotype	Perception
AgentRoleActor	Mediador
Resumo	Este caso de uso interno descreve a execução da sondagem do ambiente para verificar a finalização de uma prova.
Pré-condições	O caso de uso interno Goal “Fornecer resultado de prova” deve estar em execução
Crenças Iniciais	Nenhuma prova foi finalizada
Cenário principal	
Ações do AgentRoleActor	
1. Sondar o ambiente para verificar a finalização de uma prova	
Cenário Alternativo - Percepção Verdadeira.	
Ações do AgentRoleActor	
1. Passar a acreditar que uma prova foi finalizada.	

Fonte: O Autor

Tabela 42 – Percepção - Perceber etapa final da prova

Nome do caso de uso interno	Perceber etapa final da prova
Stereotype	Perception
AgentRoleActor	Mediador
Resumo	Este caso de uso interno descreve a execução da sondagem do ambiente buscando perceber quando faltar 1 passo para finalizar a prova.
Pré-condições	O caso de uso interno Goal “Fornecer resultado de prova” deve estar em execução
Crenças Iniciais	Não falta 1 passo para finalizar a prova.
Cenário principal	
Ações do AgentRoleActor	
1. Sondar o ambiente para verificar se falta 1 passo para o aluno finalizar a prova.	
Cenário Alternativo - Percepção Verdadeira.	
Ações do AgentRoleActor	
1. Passar a acreditar que falta 1 passo para o aluno finalizar a prova.	

Fonte: O Autor

Tabela 43 – Percepção - Perceber metade da resolução da prova

Nome do caso de uso interno	Perceber metade da resolução da prova
Stereotype	Perception
AgentRoleActor	Mediador
Resumo	Este caso de uso interno descreve a execução da sondagem do ambiente buscando perceber a metade da prova para enviar uma mensagem de incentivo ao aluno.
Pré-condições	O caso de uso interno Goal “Fornecer resultado de prova” deve estar em execução
Crenças Iniciais	O aluno não está na metade da prova.
Cenário principal	
Ações do AgentRoleActor	
1. Sondar o ambiente para verificar se o aluno está na metade da prova.	
Cenário Alternativo - Percepção Verdadeira.	
Ações do AgentRoleActor	
1. Passar a acreditar que o aluno está na metade da prova.	

Fonte: O Autor

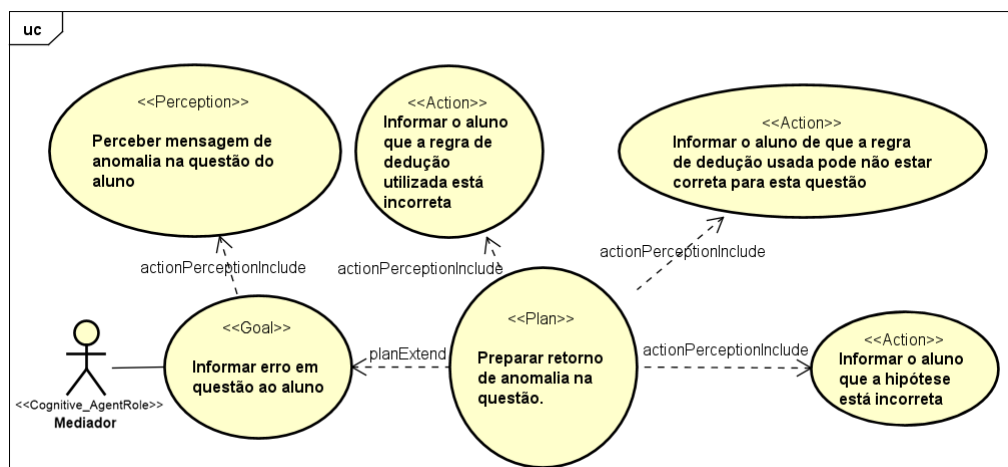
Tabela 44 – Plano - Preparar verificação da prova

Nome do caso de uso interno	Preparar verificação da prova
Stereotype	Plan
AgentRoleActor	Mediador
Resumo	Este caso de uso interno descreve as etapas seguidas pelo agente que assume a função de Mediador ao preparar a verificação de uma prova.
Cenário principal	
Ações do AgentRoleActor	
1. Solicitar verificação da prova ao especialista Técnico.	
2. Executar caso de uso interno “Perceber verificação de prova pelo Especialista” Técnico”	
Cenário Alternativo - Verificação de prova percebida	
Ações do AgentRoleActor	
1. Verificar o perfil atual do aluno	
2. Determinar o modo de apresentar o resultado da prova com relação ao perfil do aluno e o resultado da prova.	
3. Enviar uma mensagem ao aluno contendo o resultado da prova	

Fonte: O Autor

O diagrama de casos de uso parcial (Figura 20), apresenta os passos necessários para informar um erro na questão do aluno. O Agente Mediador percebe uma mensagem de anomalia na questão do aluno (enviada pelo Agente Especialista Técnico), e torna o objetivo "Informar erro em questão ao aluno", uma intenção. Após o objetivo se tornar uma intenção o plano "Preparar retorno de anomalia na questão" é disparado, finalizando com o envio de mensagem contendo o erro do aluno. A descrição dos passos para realizar este Objetivo podem ser vistas nas Tabelas 45, 46 e 47.

Figura 20 – Diagrama de Casos de Uso Parcial - Informar erro em questão.



Fonte: O Autor

Tabela 45 – Objetivo - Informar erro em questão ao aluno

Nome do caso de uso interno	Informar erro em questão ao aluno
Stereotype	Goal
AgentRoleActor	Mediador
Resumo	Este caso de uso interno descreve as etapas seguidas pelo agente que assume a função de Mediador para informar o aluno caso ocorra um erro ou ação perigosa.
Crenças Iniciais	Necessidade de fornecer retorno ao aluno negativa.
Percepções	Sondagem de mensagem de anomalia na questão do aluno.
Cenário principal	
Ações do AgentRoleActor	
1. Executar o caso de uso interno “Perceber mensagem de anomalia na questão do aluno”	
Cenário Alternativo - Aplicação de hipótese incorreta percebida	
Ações do AgentRoleActor	
1. Alterar crença para necessidade de fornecer retorno ao aluno positiva.	
2. Tornar o objetivo uma intenção.	
3. Executar caso de uso interno “Preparar retorno de anomalia na questão”	

Fonte: O Autor

Tabela 46 – Percepção - Perceber mensagem de anomalia na questão do aluno

Nome do caso de uso interno	Perceber mensagem de anomalia na questão do aluno
Stereotype	Perception
AgentRoleActor	Mediador
Resumo	Este caso de uso interno descreve a execução da sondagem do ambiente buscando perceber uma mensagem do Especialista Técnico referente a uma anomalia na questão do aluno.
Pré-condições	O caso de uso interno Goal “Informar erro em questão ao aluno” deve estar em execução
Crenças Iniciais	Nenhuma mensagem de anomalia foi encontrada.
Cenário principal	
Ações do AgentRoleActor	
1. Sondar o ambiente para verificar mensagem do Especialista Técnico sobre anomalia encontrada na questão do aluno.	
Cenário Alternativo - Percepção Verdadeira.	
Ações do AgentRoleActor	
1. Receber mensagem informando anomalia na questão do aluno.	

Fonte: O Autor

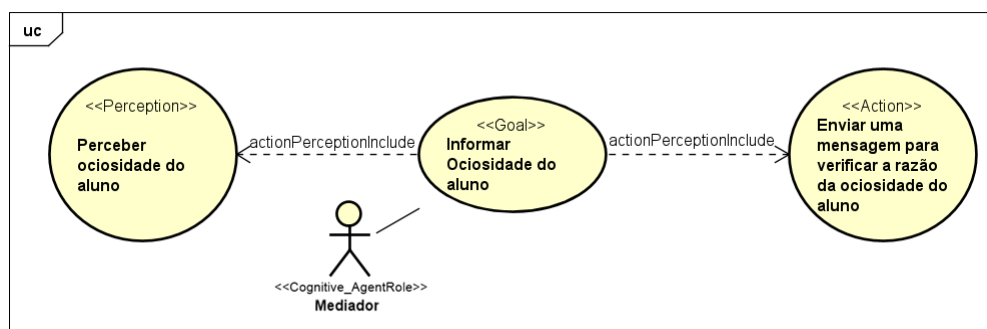
Tabela 47 – Plano - Preparar retorno de anomalia na questão

Nome do caso de uso interno	Preparar retorno de anomalia na questão.
Stereotype	Plan
AgentRoleActor	Mediador
Resumo	Este caso de uso interno descreve as etapas seguidas pelo agente que assume a função de Mediador ao preparar um retorno para anomalia em questão.
Cenário principal	
Ações do AgentRoleActor	
1. Verificar o perfil atual do aluno	
2. Verificar tipo de anomalia encontrada na questão do aluno	
Cenário Alternativo - Aplicação de hipótese incorreta	
Ações do AgentRoleActor	
1. Determinar o modo de apresentar o retorno com base no perfil do aluno.	
2. Informar o aluno que a hipótese está incorreta.	
Cenário Alternativo - Regra de dedução incorreta	
Ações do AgentRoleActor	
1. Determinar o modo de apresentar o retorno com base no perfil do aluno.	
2. Informar o aluno que a regra de dedução utilizada está incorreta.	
Cenário Alternativo - Regra de dedução perigosa	
Ações do AgentRoleActor	
1. Determinar o modo de apresentar o retorno com base no perfil do aluno.	
2. Informar o aluno que a regra de dedução utilizada pode não estar correta para esta questão	

Fonte: O Autor

No diagrama de casos de uso parcial (Figura 21), podemos visualizar os passos necessários para informar que o aluno está ocioso na resolução de uma prova/questão. O objetivo "Informar ociosidade do aluno" passa a ser uma intenção quando o Agente Mediador percebe a ociosidade do aluno. Após se tornar uma intenção, o agente envia uma mensagem para verificar a razão da ociosidade do aluno. A descrição dos passos dos cenários para realizar este Objetivo podem ser vistas nas Tabelas 48 e 49.

Figura 21 – Diagrama de Casos de Uso Parcial - Informar ociosidade do aluno.



Fonte: O Autor

Tabela 48 – Objetivo - Informar ociosidade do aluno

Nome do caso de uso interno	Informar ociosidade do aluno
Stereotype	Goal
AgentRoleActor	Mediador
Resumo	Este caso de uso interno descreve as etapas seguidas pelo agente que assume a função de Mediador para enviar uma mensagem ao aluno em caso de ociosidade.
Crenças Iniciais	Necessidade de informar ociosidade.
Percepções	Sondagem de ociosidade do aluno.
Cenário principal	
Ações do AgentRoleActor	
1. Executar o caso de uso interno “Perceber ociosidade do aluno”	
Cenário Alternativo - Ociosidade do aluno percebida	
Ações do AgentRoleActor	
1. Alterar crença para necessidade de informar ociosidade positiva.	
2. Tornar o objetivo uma intenção.	
3. Enviar uma mensagem para verificar a razão da ociosidade do aluno.	

Fonte: O Autor

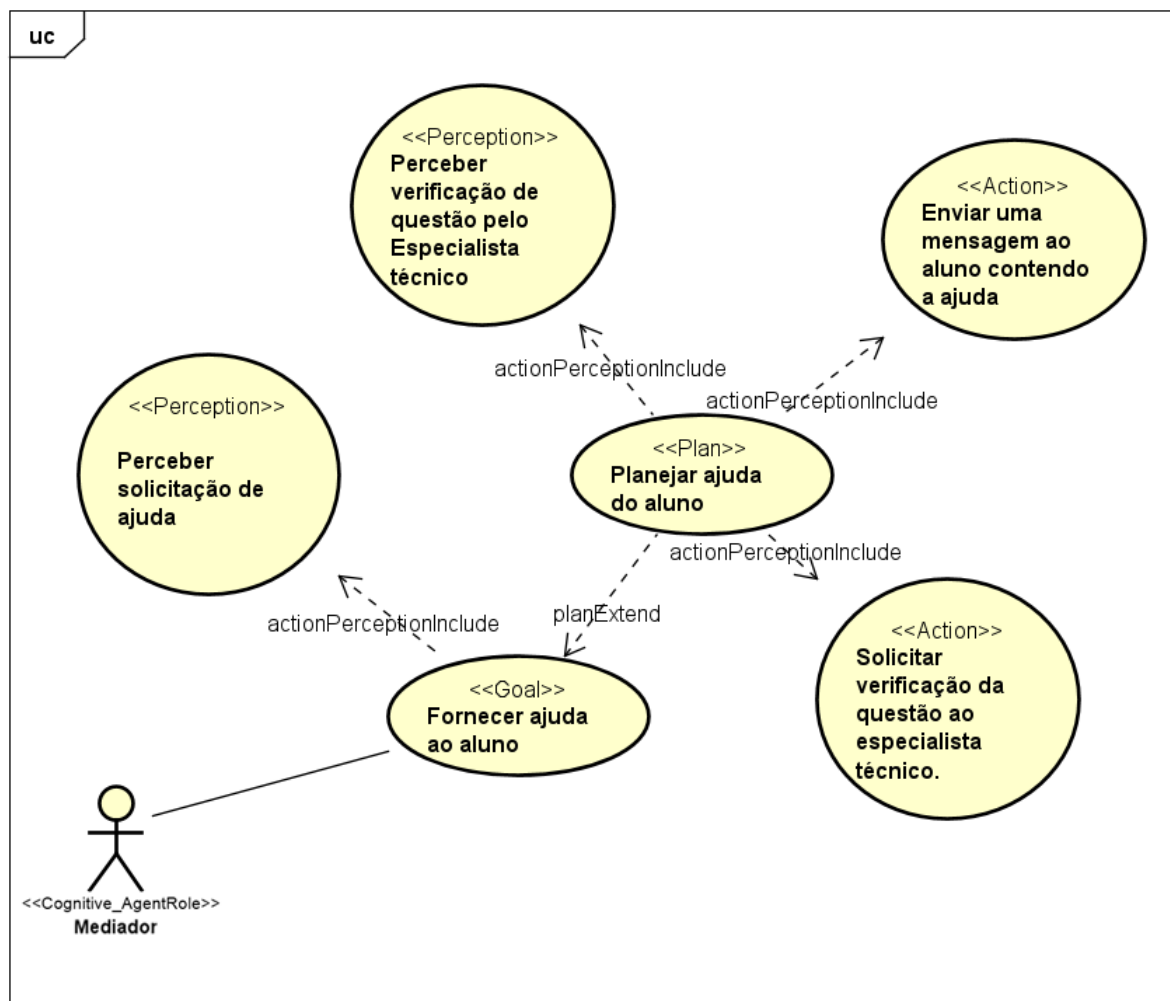
Tabela 49 – Percepção - Perceber ociosidade do aluno

Nome do caso de uso interno	Perceber ociosidade do aluno
Stereotype	Perception
AgentRoleActor	Mediador
Resumo	Este caso de uso interno descreve a execução da sondagem do ambiente buscando perceber quando o aluno está ocioso na resolução de uma questão.
Pré-condições	O caso de uso interno Goal “Informar ociosidade do aluno” deve estar em execução e o aluno deve estar respondendo uma questão.
Crenças Iniciais	Aluno ocioso negativo
Cenário principal	
Ações do AgentRoleActor	
1. Sondar tempo de ociosidade do aluno comparando com o dado de ociosidade presente no perfil do aluno e a complexidade da questão.	
Cenário Alternativo - Aluno ultrapassou o tempo máximo de ociosidade.	
Ações do AgentRoleActor	
1. Alterar crença para Aluno ocioso positiva.	
Cenário Alternativo - Aluno Agiu.	
Ações do AgentRoleActor	
1. Zerar o tempo de ociosidade do aluno.	

Fonte: O Autor

O diagrama de casos de uso parcial (Figura 22), apresenta os passos necessários para o Agente Mediador fornecer ajuda a um aluno. Quando o aluno solicita ajuda, o agente percebe essa solicitação, tornando o objetivo "Fornecer ajuda ao aluno", uma intenção. Após o objetivo se tornar uma intenção o plano "Planejar ajuda do aluno" é disparando resultando no envio de uma mensagem contendo a ajuda do aluno. A descrição dos passos dos cenários para realizar este Objetivo podem ser vistas nas Tabelas 50, 51, 37 e 52.

Figura 22 – Diagrama de Casos de Uso Parcial - Fornecer ajuda ao aluno.



Fonte: O Autor

Tabela 50 – Objetivo - Fornecer ajuda ao aluno

Nome do caso de uso interno	Fornecer ajuda ao aluno
Stereotype	Goal
AgentRoleActor	Mediador
Resumo	Este caso de uso interno descreve as etapas seguidas pelo agente que assume a função de Mediador para retornar uma ajuda ao aluno.
Crenças Iniciais	Necessidade de fornecer ajuda negativa.
Percepções	Sondagem de pedido de ajuda.
Cenário principal	
Ações do AgentRoleActor	
1. Executar o caso de uso interno “Perceber solicitação de ajuda”	
Cenário Alternativo - Solicitação de ajuda percebida	
Ações do AgentRoleActor	
1. Alterar crença para necessidade de fornecer ajuda positiva.	
2. Tornar o objetivo uma intenção.	
3. Executar o caso de uso interno “Preparar ajuda do aluno”.	

Fonte: O Autor

Tabela 51 – Percepção - Perceber solicitação de ajuda

Nome do caso de uso interno	Perceber solicitação de ajuda
Stereotype	Perception
AgentRoleActor	Mediador
Resumo	Este caso de uso interno descreve a execução da sondagem do ambiente buscando perceber uma solicitação de ajuda por parte do aluno.
Pré-condições	O caso de uso interno Goal “Fornecer ajuda ao aluno” deve estar em execução
Crenças Iniciais	Aluno não solicitou ajuda.
Cenário principal	
Ações do AgentRoleActor	
1. Sondar o ambiente para verificar se ocorreu solicitação de ajuda por parte do aluno. sobre anomalia encontrada na questão do aluno.	
Cenário Alternativo - Percepção Verdadeira.	
Ações do AgentRoleActor	
1. Passar a acreditar que o aluno solicitou ajuda.	

Fonte: O Autor

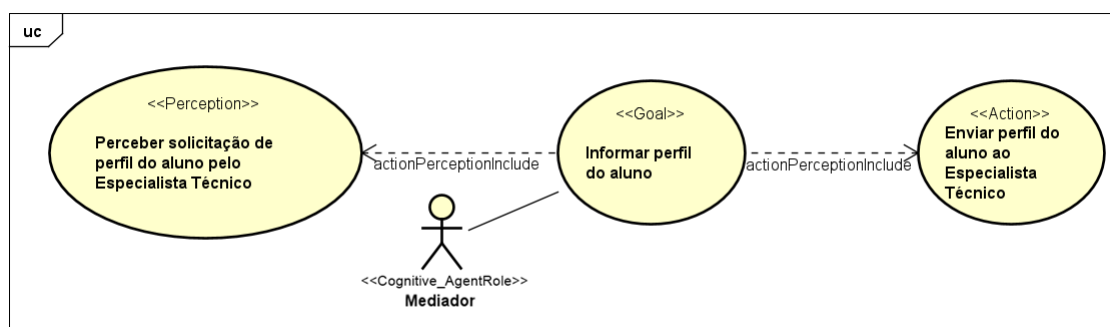
Tabela 52 – Plano - Preparar ajuda ao aluno

Nome do caso de uso interno	Preparar ajuda do aluno
Stereotype	Plan
AgentRoleActor	Mediador
Resumo	Este caso de uso interno descreve as etapas seguidas pelo agente que assume a função de Mediador a ajuda que será fornecida ao aluno.
Cenário principal	
Ações do AgentRoleActor	
1. Solicitar verificação da questão ao Especialista Técnico.	
2. Executar caso de uso interno “ Perceber verificação de questão do Especialista” Técnico.”	
Cenário Alternativo - Verificação de questão recebida	
Ações do AgentRoleActor	
1. Verificar o perfil atual do aluno	
2. Selecionar um retorno de acordo com o perfil do aluno, com o tipo de ajuda solicitada pelo aluno e com a verificação de questão informada pelo Especialista Técnico.	
3. Enviar uma mensagem ao aluno contendo a ajuda.	

Fonte: O Autor

Os passos necessários para informar o perfil do aluno estão presentes no diagrama de casos de uso parcial (Figura 23). O Agente Mediador percebe a solicitação de envio do perfil do aluno pelo Especialista Técnico, torando o objetivo "Informar perfil do aluno" uma intenção. Quando o objetivo se torna uma intenção o Mediador envia o perfil do aluno ao Especialista Técnico. A descrição dos passos dos cenários necessários para executar este Objetivo podem ser vistas nas Tabelas 53 e 54.

Figura 23 – Diagrama de Casos de Uso Parcial - Informar perfil do aluno.



Fonte: O Autor

Tabela 53 – Objetivo - Informar perfil do aluno

Nome do caso de uso interno	Informar perfil do aluno
Stereotype	Goal
AgentRoleActor	Mediador
Resumo	Este caso de uso interno descreve as etapas seguidas pelo agente que assume a função de Mediador a informar perfil do aluno.
Crenças Iniciais	Solicitação de perfil do aluno negativa
Percepções	Sondagem de solicitação de perfil do aluno.
Cenário principal	
Ações do AgentRoleActor	
1. Executar caso de uso interno “Perceber solicitação de perfil do aluno”	
Cenário Alternativo - Solicitação de perfil do aluno percebida	
Ações do AgentRoleActor	
1. Alterar crença para Solicitação de perfil do aluno positiva	
2. Tornar o objetivo uma intenção.	
3. Verificar o perfil atual do aluno	
4. Enviar o perfil do aluno ao agente Especialista Técnico.	

Fonte: O Autor

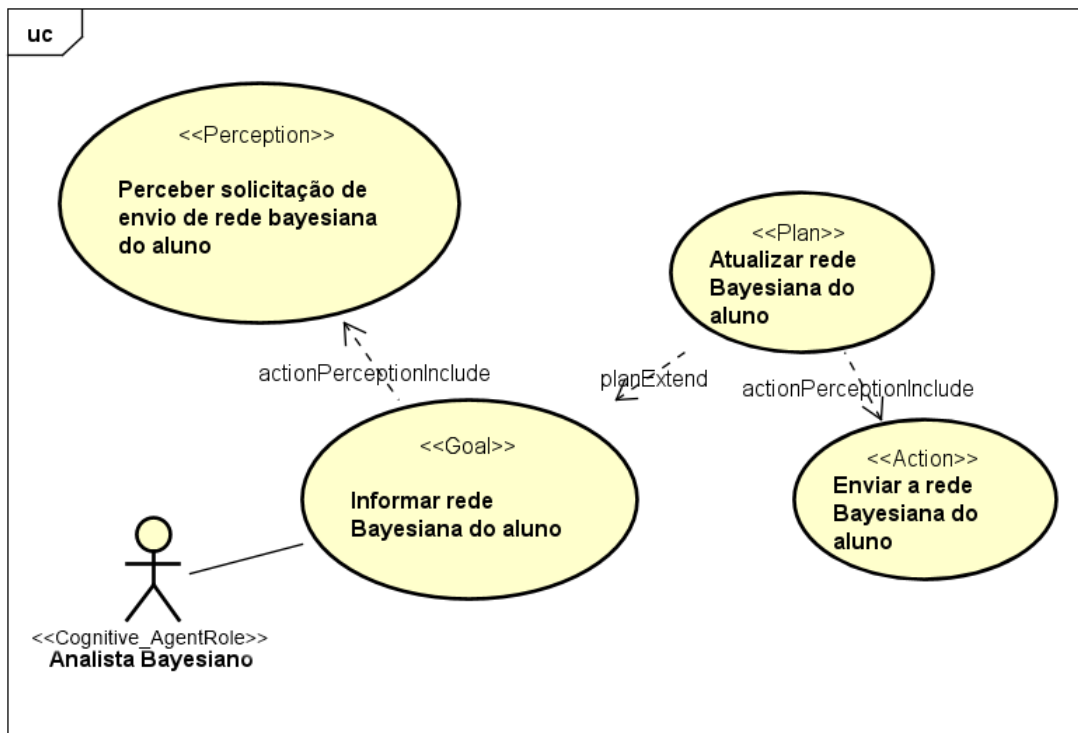
Tabela 54 – Percepção - Perceber solicitação de perfil do aluno pelo Especialista Técnico

Nome do caso de uso interno	Perceber solicitação de perfil do aluno pelo Especialista Técnico
Stereotype	Perception
AgentRoleActor	Mediador
Resumo	Este caso de uso interno descreve a execução da sondagem do ambiente para verificar se o perfil do aluno foi solicitado pelo Especialista Técnico.
Pré-condições	O caso de uso interno Goal “Informar perfil do aluno” deve estar em execução
Crenças Iniciais	Não houve solicitação de perfil do aluno.
Cenário principal	
Ações do AgentRoleActor	
1. Sondar o ambiente para verificar a ocorrência de uma nova solicitação de perfil do aluno.	
Cenário Alternativo - Percepção Verdadeira.	
Ações do AgentRoleActor	
1. Passar a acreditar que o perfil do aluno foi solicitado.	

Fonte: O Autor

O diagrama de casos de uso parcial (Figura 24), apresenta os passos necessários para o Agente Analista Bayesiano, informar a rede Bayesiana do aluno. O Analista Bayesiano percebe a solicitação de envio de rede Bayesiana, tornando o objetivo "Informar rede Bayesiana" uma intenção. Quando o objetivo se torna uma intenção, o plano "Atualizar rede Bayesiana" é disparado resultando no envio da rede Bayesiana. A descrição dos passos dos cenários necessários para realizar este Objetivo podem ser vistos nas Tabelas 55, 56 e 57.

Figura 24 – Diagrama de Casos de Uso Parcial - Informar rede bayesiana



Fonte: O Autor

Tabela 55 – Objetivo - Informar rede bayesiana do aluno

Nome do caso de uso interno	Informar rede bayesiana do aluno
Stereotype	Goal
AgentRoleActor	Analista bayesiano
Resumo	Este caso de uso interno descreve as etapas seguidas pelo agente que assume a função de “Analista Bayesiano” a enviar a rede Bayesiana do aluno.
Crenças Iniciais	Solicitação de rede Bayesiana do aluno negativa
Percepções	Sondagem de novas solicitações de envio de rede bayesiana do aluno.
Cenário principal	
Ações do AgentRoleActor	
1. Executar o caso de uso interno “Perceber solicitação de envio de rede bayesiana do aluno”	
Cenário Alternativo - Solicitação de Verificação Percebida	
Ações do AgentRoleActor	
1. Alterar crença para Solicitação de rede Bayesiana do aluno positiva	
2. Tornar o objetivo uma intenção.	
3. Executar caso de uso interno “Atualizar rede Bayesiana do aluno”	

Fonte: O Autor

Tabela 56 – Percepção - Perceber solicitação de envio de rede bayesiana

Nome do caso de uso interno	Perceber solicitação de envio de rede bayesiana do aluno
Stereotype	Perception
AgentRoleActor	Analista Bayesiano
Resumo	Este caso de uso interno descreve a execução da sondagem do ambiente para verificar se o envio de rede Bayesiana do aluno foi solicitado pelo Mediador.
Pré-condições	O caso de uso interno Goal “Informar rede Bayesiana do aluno” deve estar em execução
Crenças Iniciais	Não houve solicitação de envio de rede Bayesiana do aluno.
Cenário principal	
Ações do AgentRoleActor	
1. Sondar o ambiente para verificar a ocorrência de uma nova solicitação de envio de Rede Bayesiana do aluno.	
Cenário Alternativo - Percepção Verdadeira.	
Ações do AgentRoleActor	
1. Passar a acreditar que o envio de Rede Bayesiana do aluno foi solicitado.	

Fonte: O Autor

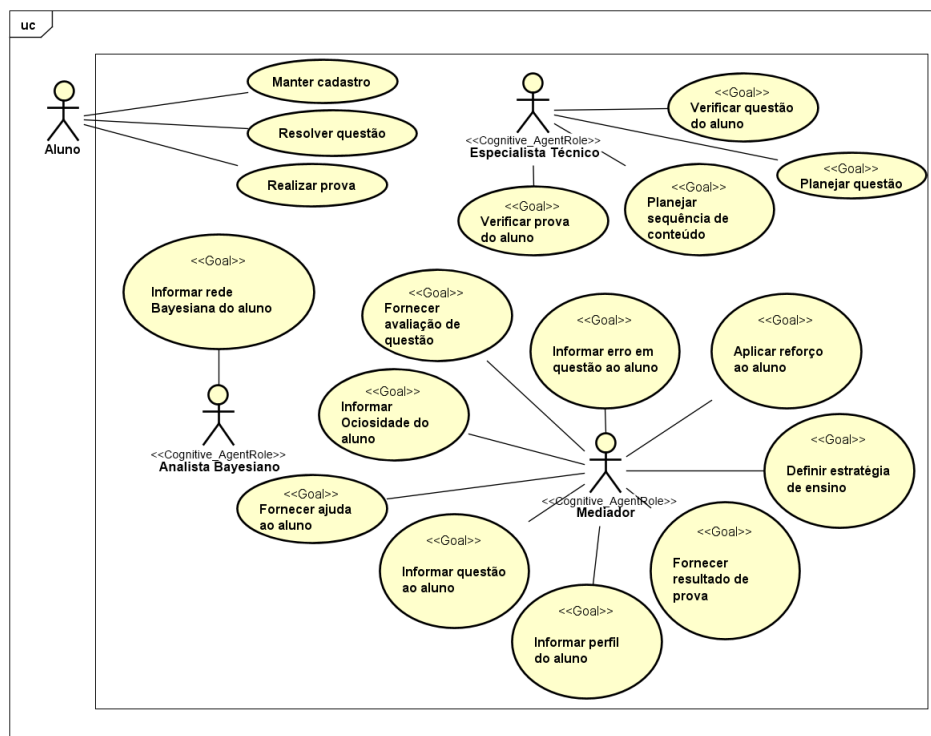
Tabela 57 – Percepção - Atualizar rede bayesiana do aluno

Nome do caso de uso interno	Atualizar rede bayesiana do aluno
Stereotype	Plan
AgentRoleActor	Analista Bayesiano
Resumo	Este caso de uso interno descreve as etapas seguidas pelo agente que assume a função de Analista Bayesiano para atualizar a rede bayesiana do aluno de um determinado aluno.
Cenário principal	
Ações do AgentRoleActor	
1. Verificar o histórico do aluno em sua base de conhecimento.	
2. Realizar os cálculos necessários para determinar as estatísticas presentes na rede bayesiana do aluno.	
3. Enviar a rede Bayesiana do aluno ao Mediador.	

Fonte: O Autor

Como resultado dos diagramas parciais, foi desenvolvido um diagrama de casos de uso geral, apresentado na Figura 25, contendo não só os objetivos de todos os agentes, como também as funcionalidades que podem ser solicitadas pelos usuários externos.

Figura 25 – Diagrama de casos de uso Final



Fonte: O Autor

4.2.5 Validação dos Requisitos

Como explicado anteriormente, a validação de requisitos deste processo está sendo desenvolvida em paralelo com este trabalho. Entretanto verificamos a necessidade de validar os requisitos especificados na execução do processo no Sistema Heráclito. Para isso buscamos validar estes requisitos mesmo não seguindo uma etapa documentada.

Na validação de requisitos verificamos a aceitação dos cenários propostos. Esta validação de revisão foi aplicada após a primeira versão da especificação dos cenários e serviu como método para identificar erros na descrição dos cenários, juntamente com uma validação dos objetivos dos agentes.

O funcionamento desta validação foi feita por meio da apresentação dos cenários para o stakeholder, perguntando se os mesmos estavam corretos e solicitando sugestões. Caso o stakeholder encontrasse um erro ou se estivesse faltando algo na descrição do cenário, deveria ser feito um comentário sugerindo alterações no cenário.

Com esta validação conseguimos identificar diversos erros presentes na primeira versão da especificação. O comentário do stakeholder para cada um dos objetivos validados esta descrito a seguir:

- Objetivos do Especialista técnico.
 - **Verificar prova do aluno:** Foi validado que as mensagens informando uso de regra de dedução incorreta, regra de dedução perigosa e hipótese incorreta só são aplicadas em questões. A descrição do erro deve estar presente juntamente com a avaliação da prova, mas não deve conter uma mensagem separada para o aluno;
 - **Verificar questão do aluno:** Foi solicitado pelo stakeholder a verificação da questão em tempo de execução, sendo necessário o monitoramento do agente durante a execução de uma questão;
 - **Planejar questões:** O stakeholder apontou que deveria constar no plano do agente a complexidade da questão e que na mensagem informando a questão deveriam ser informados todos os dados da questão (complexidade, tempo necessário para desenvolver, conteúdo, etc);
 - **Planejar sequência de conteúdo:** O stakeholder não apontou erros neste objetivo.

- Objetivos do Mediador
 - **Informar questão ao aluno:** O stakeholder não apontou erros neste objetivo;
 - **Definir estratégia de ensino:** Este foi o objetivo que mais foram encontrados problemas. O stakeholder apontou a necessidade do agente Mediador

redefinir estratégia de ensino quando perceber falha do aluno em uma prova. Também foi observado que nos passos da estratégia de ensino deveria constar a possibilidade de mudar estratégia de avaliação, além do conteúdo de estudo do aluno. O último ponto é que o perfil do aluno poderia ser melhor descrito, separando o tipo de necessidade do aluno, área de atuação e disciplinas cursadas anteriormente;

- **Aplicar reforço ao aluno:** O stakeholder observou que o aluno pode solicitar reforço se entender que é necessário;
 - **Fornecer feedback de questão:** O stakeholder não encontrou erros neste objetivo. Entretanto observamos que o nome do objetivo "Feedback de questão" confundiu o stakeholder pois, existe outro objetivo que apenas informa os erros do aluno em uma questão. Desta maneira, mudamos o nome do objetivo para "Fornecer avaliação de questão";
 - **Fornecer feedback de prova:** O stakeholder não encontrou erros neste objetivo;
 - **Informar erro em questão ao aluno:** O stakeholder não encontrou erros neste objetivo;
 - **Informar ociosidade do aluno:** Para este objetivo foram encontrados erros na logística de seu funcionamento. O tempo de ociosidade permitido antes do informe, depende da complexidade da questão e de um cálculo de ociosidade realizado a partir do tempo médio que o aluno demora para realizar uma questão. Este dado é obtido no histórico do aluno;
 - **Fornecer ajuda ao aluno:** O stakeholder não encontrou erros neste objetivo;
 - **Informar perfil do aluno:** O stakeholder não encontrou erros neste objetivo.
- **Objetivos do Analista Bayesiano.**
 - **Informar rede Bayesiana:** O stakeholder não encontrou erros neste objetivo;
 - **Informar histórico do aluno:** O stakeholder apontou que este objetivo não era necessário pois o histórico do aluno pode ser obtido através da Rede Bayesiana.

4.2.6 Resultados Extraídos da Execução do Processo

Ao executar o processo em um ambiente real, conseguimos extrair algumas informações que podem contribuir com trabalhos futuros.

A etapa de eliciação de requisitos foi realizada com base na metodologia Homer. Entretanto a metodologia Homer contém diversas fraquezas que foram apontadas no Capítulo 4, Seção 4.2.2. Deste modo entendemos que estender a metodologia Homer é uma

ótima oportunidade de trabalho futuro além de ser essencial para um melhor funcionamento deste processo.

Outro ponto que podemos citar é o fato de que as crenças dos agentes não estão sendo expressas nos modelos. O modelo de caso de uso utilizado para representar os cenários dos agentes não fornece um modo de expressar as crenças dos agentes. Sendo necessário realizar um estudo para verificar se existem formas de modelagem de crenças ou se é preciso propor um método e/ou notação para isso. Este estudo está sendo realizado em paralelo com este trabalho e se está verificando a possibilidade de modelar crenças através de modelos conceituais adaptados para representar crenças de agentes, além de outras informações estruturais como a representação de objetivos, percepções e planos.

Por fim, de uma perspectiva mais ampla, consideramos que o trabalho contribuiu com o projeto Heráclito visto que seus requisitos não tinham sido documentados na primeira versão e agora com a aplicação de nosso processo todos os cenários da nova versão relativos aos objetivos, percepções e planos de cada agente estão documentados.

5 CONSIDERAÇÕES FINAIS

Neste trabalho propomos um processo de Engenharia de requisitos para Sistemas Multiagentes. O processo proposto englobou as 4 etapas de engenharia de requisitos inspiradas no *SWEBOK*: elicitação, análise, especificação e validação de requisitos. Procuramos deste modo dar um maior suporte para o engenheiro de software na execução de um processo de engenharia de requisitos para este tipo de sistema.

A etapa de elicitação deste processo visa descobrir os requisitos específicos para SMAs em um contato direto com o Stakeholder, elicitando agentes de software e suas funções, por meio de entrevistas, realizando uma metáfora organizacional, como a contratação de novos funcionários para uma empresa. Esta elicitação foi realizada através da metodologia Homer (Wilmann; Sterling, 2005). Também foi feita uma avaliação da execução desta metodologia, onde foram encontrados diversos pontos fracos que podem servir como base para trabalhos futuros.

A análise foi proposta por meio da identificação de cenários, especificados através de uma extensão do diagrama de casos de uso (GUEDES, 2012). Esta análise visa a identificação de requisitos específicos para SMAs com foco no suporte do modelo BDI. Por último, a etapa de validação deste processo visa validar os artefatos especificados nas etapas de elicitação e análise.

Com objetivo de avaliar o processo proposto, nós o aplicamos na segunda versão do sistema Heráclito (GALAFASSI et al., 2019), sistema multiagente que visa auxiliar no ensino de lógica. Esperamos que a ER apresentada neste processo possa auxiliar no desenvolvimento deste projeto.

Com relação a execução do processo, realizamos entrevistas com stakeholders do sistema Heráclito, com o objetivo de elicitar os requisitos, identificando cenários preliminares do sistema. Estes cenários preliminares serviram de base para a identificação de diagramas de casos de uso preliminares, no qual identificamos os objetivos, percepções, planos e ações dos agentes do sistema Heráclito. Em seguida, descrevemos os cenários completos do sistema através da descrição dos casos uso internos, desta vez os objetivos, percepções, planos e ações, foram detalhados além da identificação das crenças dos agentes. Por fim, validamos estes cenários através de uma leitura completa junto a um dos stakeholders.

Concluindo, identificamos também oportunidades de trabalhos futuros, tanto em relação a uma possível extensão da metodologia Homer, fornecendo um suporte maior a requisitos particulares de SMAs como ações, percepções e planos. Como também a necessidade de uma notação que expresse as crenças dos agentes.

REFERÊNCIAS

- ABUSHARK, Y. et al. Requirements specification via activity diagrams for agent-based systems. **Autonomous Agents and Multi-Agent Systems**, 02 2016. Citado 2 vezes nas páginas 41 e 49.
- ALONSO, F. et al. Sonia: A methodology for natural agent development. In: . [S.l.: s.n.], 2004. p. 245–260. Citado 2 vezes nas páginas 41 e 48.
- ALZETTA, F. et al. In-time explainability in multi-agent systems: Challenges, opportunities, and roadmap. In: SPRINGER. **International Workshop on Explainable, Transparent Autonomous Agents and Multi-Agent Systems**. [S.l.], 2020. p. 39–53. Citado na página 24.
- ARGENTE, E.; BOTTI, V.; JULIÁN, V. Gormas: An organizational-oriented methodological guideline for open mas. In: . [S.l.: s.n.], 2009. p. 32–47. Citado 2 vezes nas páginas 41 e 49.
- ASHAMALLA, A.; BEYDOUN, G.; LOW, G. Model driven approach for real-time requirement analysis of multi-agent systems. **Computer Languages, Systems Structures**, v. 50, 06 2017. Citado 2 vezes nas páginas 41 e 51.
- BANACH, R. A deidealisation semantics for kaos. In: . [S.l.: s.n.], 2010. p. 267–274. Citado 2 vezes nas páginas 41 e 50.
- BERENBACH, B. et al. **Software & systems requirements engineering: in practice**. [S.l.]: McGraw-Hill, Inc., 2009. Citado na página 33.
- BIOLCHINI, J. et al. **Systematic review in software engineering**. [S.l.]: System Engineering and Computer Science Department COPPE/UFRJ, 2005. Citado na página 37.
- BLANES, D.; INSFRAN, E.; ABRAHÃO, S. Re4gaia: A requirements modeling approach for the development of multi-agent systems. In: . [S.l.: s.n.], 2009. v. 59, p. 245–252. Citado 2 vezes nas páginas 41 e 49.
- BOKMA, A. et al. Engineering large-scale agent-based systems with consensus. **Robotics and Computer-Integrated Manufacturing**, v. 11, n. 2, p. 81 – 90, 1994. ISSN 0736-5845. Disponível em: <<http://www.sciencedirect.com/science/article/pii/0736584594900124>>. Citado na página 41.
- BONJEAN, N. et al. Adelfe 2.0. In: _____. **Handbook on Agent-Oriented Design Processes**. Berlin, Heidelberg: Springer Berlin Heidelberg, 2014. p. 19–63. ISBN 978-3-642-39975-6. Disponível em: <https://doi.org/10.1007/978-3-642-39975-6_3>. Citado 5 vezes nas páginas 41, 44, 47, 49 e 54.
- BOTTI, V. et al. Multi-agent systems. MDPI-Multidisciplinary Digital Publishing Institute, 2019. Citado na página 23.
- BOURQUE, P.; FAIRLEY, R. E. et al. **Guide to the software engineering body of knowledge (SWEBOK (R)): Version 3.0**. [S.l.]: IEEE Computer Society Press, 2014. Citado 8 vezes nas páginas 23, 26, 30, 31, 32, 33, 38 e 47.
- BRENNAN, K. et al. **A Guide to the Business Analysis Body of Knowledge**. [S.l.]: Iiba, 2009. Citado na página 32.

- BRESCIANI, P.; DONZELLI, P. Ref: A practical agent-based requirement engineering framework. In: . [S.l.: s.n.], 2003. v. 2814, p. 217–228. Citado 2 vezes nas páginas 41 e 51.
- BRYL, V. et al. B-tropos: Agent-oriented requirements engineering meets computational logic for declarative business process modeling and verification. In: . [S.l.: s.n.], 2008. p. 157–176. ISBN 9783540888321. Citado 2 vezes nas páginas 41 e 50.
- CAIRE, G. et al. The message methodology. In: _____. [S.l.: s.n.], 2004. p. 177–194. Citado 3 vezes nas páginas 41, 44 e 52.
- CAO, L. Osoad methodology. In: _____. [S.l.: s.n.], 2015. p. 111–129. ISBN 978-1-4471-6550-7. Citado 3 vezes nas páginas 41, 44 e 52.
- CHOREN, R.; LUCENA, C. The anote modeling language for agent-oriented specification. In: CHOREN, R. et al. (Ed.). **Software Engineering for Multi-Agent Systems III**. Berlin, Heidelberg: Springer Berlin Heidelberg, 2005. p. 198–212. ISBN 978-3-540-31846-0. Citado 2 vezes nas páginas 29 e 30.
- COSENTINO, M. et al. A glimpse of the aspecs process documented with the fipa dpdf template. In: . [S.l.: s.n.], 2010. v. 627. Citado 2 vezes nas páginas 41 e 50.
- COSENTINO, M. et al. The aspecs process. In: _____. **Handbook on Agent-Oriented Design Processes**. Berlin, Heidelberg: Springer Berlin Heidelberg, 2014. p. 65–114. ISBN 978-3-642-39975-6. Disponível em: <https://doi.org/10.1007/978-3-642-39975-6_4>. Citado 2 vezes nas páginas 29 e 30.
- COSENTINO, M. et al. The aspecs process. In: _____. **Handbook on Agent-Oriented Design Processes**. Berlin, Heidelberg: Springer Berlin Heidelberg, 2014. p. 65–114. ISBN 978-3-642-39975-6. Disponível em: <https://doi.org/10.1007/978-3-642-39975-6_4>. Citado na página 66.
- COSENTINO, M. et al. **Handbook on Agent-Oriented Design Processes**. [S.l.]: Springer-Verlag Berlin Heidelberg, 2014. Citado na página 55.
- COSENTINO, M.; SEIDITA, V. Passi: Process for agent societies specification and implementation. In: _____. **Handbook on Agent-Oriented Design Processes**. Berlin, Heidelberg: Springer Berlin Heidelberg, 2014. p. 287–329. ISBN 978-3-642-39975-6. Disponível em: <https://doi.org/10.1007/978-3-642-39975-6_10>. Citado 3 vezes nas páginas 29, 30 e 38.
- COSENTINO, M.; SEIDITA, V. Passi: Process for agent societies specification and implementation. In: _____. [S.l.: s.n.], 2014. p. 287–329. Citado 3 vezes nas páginas 41, 44 e 48.
- CYSNEIROS, L.; YU, E. Requirements engineering for large-scale multi-agent systems. In: . [S.l.: s.n.], 2002. v. 2603, p. 39–56. Citado 3 vezes nas páginas 41, 53 e 54.
- DELOACH, S.; GARCIA-OJEDA, J. The o-mase methodology. In: _____. [S.l.: s.n.], 2014. p. 253–285. Citado 3 vezes nas páginas 41, 44 e 52.

- DOMANN, J. et al. An agile method for multiagent software engineering. **Procedia Computer Science**, v. 32, p. 928 – 934, 2014. ISSN 1877-0509. The 5th International Conference on Ambient Systems, Networks and Technologies (ANT-2014), the 4th International Conference on Sustainable Energy Information Technology (SEIT-2014). Disponível em: <<http://www.sciencedirect.com/science/article/pii/S1877050914007133>>. Citado 2 vezes nas páginas 41 e 48.
- Dorri, A.; Kanhere, S. S.; Jurdak, R. Multi-agent systems: A survey. **IEEE Access**, v. 6, p. 28573–28593, 2018. Citado na página 23.
- FUENTES-FERNÁNDEZ, R.; GÓMEZ-SANZ, J.; PAVÓN, J. Requirements elicitation and analysis of multiagent systems using activity theory. **Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on**, v. 39, p. 282 – 298, 04 2009. Citado 2 vezes nas páginas 41 e 49.
- GALAFASSI, F. F. P. et al. Heráclito: Intelligent tutoring system for logic. In: SPRINGER. **International Conference on Practical Applications of Agents and Multi-Agent Systems**. [S.l.], 2019. p. 251–254. Citado 3 vezes nas páginas 26, 62 e 109.
- GAN, K. S. et al. Enforcing social semantic in fipa-acl using spin. In: **Agents and Multi-agent Systems: Technologies and Applications 2019**. [S.l.]: Springer, 2020. p. 3–13. Citado na página 23.
- GARCÍA-OJEDA, J.; DELOACH, S. The o-mase methodology. In: _____. [S.l.: s.n.], 2014. p. 253–285. ISBN 978-3-642-39974-9. Citado na página 30.
- GAUR, V.; SONI, A. A novel approach to explore inter agent dependencies from user requirements. **Procedia Technology**, v. 1, p. 412–419, 12 2012. Citado 2 vezes nas páginas 41 e 52.
- GONZÁLEZ-MORENO, J. et al. Ingenias-scrum. In: _____. [S.l.: s.n.], 2014. p. 219–251. Citado 3 vezes nas páginas 41, 44 e 49.
- GUEDES, G. T. A. Um metamodelo uml para a modelagem de requisitos em projetos de sistemas multiagentes. 2012. Citado 4 vezes nas páginas 26, 34, 57 e 109.
- GUEDES, G. T. A.; VICARI, R. M. A uml profile oriented to the requirements modeling in intelligent tutoring systems projects. In: BRAMER, M. (Ed.). **Artificial Intelligence in Theory and Practice III**. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010. p. 133–142. ISBN 978-3-642-15286-3. Citado na página 23.
- HAJDUK, M.; SUKOP, M.; HAUN, M. Cognitive multi-agent systems. Springer, 2019. Citado na página 23.
- Hajer, B. M.; Taieb, B. R.; Raouf, K. A new mas based approach modeling the qms continual improvement. In: **2009 IEEE International Conference on Systems, Man and Cybernetics**. [S.l.: s.n.], 2009. p. 4734–4739. Citado 2 vezes nas páginas 41 e 48.
- Haumer, P. et al. Bridging the gap between past and future in re: a scenario-based approach. In: **Proceedings IEEE International Symposium on Requirements Engineering (Cat. No.PR00188)**. [S.l.: s.n.], 1999. p. 66–73. Citado 2 vezes nas páginas 41 e 51.

- HERZIG, A. et al. Bdi logics for bdi architectures: old problems, new perspectives. **KI-Künstliche Intelligenz**, Springer, v. 31, n. 1, p. 73–83, 2017. Citado na página 24.
- HILAIRE, V. et al. An approach for the integration of swarm intelligence in mas: An engineering perspective. **Expert Systems with Applications**, ?, p. 1–24, 01 2012. Citado 2 vezes nas páginas 41 e 51.
- Hsieh, M. et al. Spoken dialogue agent interface requirements modeling based on passi methodology. In: **2008 Eighth International Conference on Intelligent Systems Design and Applications**. [S.l.: s.n.], 2008. v. 1, p. 339–342. Citado 2 vezes nas páginas 41 e 48.
- HUIYING, X.; ZHI, J. An agent-oriented requirement graphic symbol representation and formalization modeling method. In: . [S.l.: s.n.], 2009. v. 4, p. 569 – 574. Citado 2 vezes nas páginas 41 e 52.
- IGLESIAS, C. et al. Analysis and design of multiagent systems using mas-commonkads. **Lecture Notes in Computer Science**, 11 1998. Citado 3 vezes nas páginas 41, 44 e 53.
- JAIN, S. A visualization technique for agent based goal refinement to elicit soft goals in goal oriented requirements engineering. In: . [S.l.: s.n.], 2007. p. 2–2. ISBN 978-0-7695-3248-6. Citado 2 vezes nas páginas 41 e 51.
- JO, C.-H.; EINHORN, J. A bdi agent-based software process. **Journal of Object Technology**, v. 4, p. 101–121, 11 2005. Citado 3 vezes nas páginas 41, 53 e 54.
- KITCHENHAM, B.; BRERETON, P. A systematic review of systematic review process research in software engineering. **Inf. Softw. Technol.**, Butterworth-Heinemann, USA, v. 55, n. 12, p. 2049–2075, dez. 2013. ISSN 0950-5849. Disponível em: <<https://doi.org/10.1016/j.infsof.2013.07.010>>. Citado na página 37.
- KITCHENHAM, B.; CHARTERS, S. Guidelines for performing systematic literature reviews in software engineering. v. 2, 01 2007. Citado na página 37.
- KOSCIANSKI, A.; SOARES, M. dos S. **Qualidade de Software-2ª Edição: Aprenda as metodologias e técnicas mais modernas para o desenvolvimento de software**. [S.l.]: Novatec Editora, 2007. Citado na página 24.
- KOTONYA, G.; SOMMERVILLE, I. **Requirements engineering: processes and techniques**. [S.l.]: Wiley Publishing, 1998. Citado 2 vezes nas páginas 31 e 32.
- LARSEN, J. B. Agent programming languages and logics in agent-based simulation. In: **Modern approaches for intelligent information and database systems**. [S.l.]: Springer, 2018. p. 517–526. Citado na página 24.
- LEE, C.-H.; LIU, A. A method for agent-based system requirements analysis. In: . [S.l.: s.n.], 2002. p. 214 – 221. ISBN 0-7695-1857-5. Citado 2 vezes nas páginas 41 e 50.
- LEE, J.; LEE, H. The sramo technique for analysis and reuse of requirements in multi-agent application engineering. **9th Workshop on Requirements Engineering**, p. 41–50, 2006. Citado 2 vezes nas páginas 41 e 50.

- LEE, J.; LEE, H. Strategic agent based web system development methodology. **International Journal of Information Technology Decision Making**, v. 7, p. 309–337, 2008. Citado 2 vezes nas páginas 41 e 50.
- LIND, J. **Iterative Software Engineering for Multiagent Systems: The MASSIVE Method**. [S.l.: s.n.], 2001. ISBN 3-540-42166-1. Citado 3 vezes nas páginas 41, 44 e 49.
- LIU, L. et al. Agent-oriented requirements analysis from scenarios. In: O'SHEA, J. et al. (Ed.). **Agent and Multi-Agent Systems: Technologies and Applications**. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011. p. 394–405. ISBN 978-3-642-22000-5. Citado 2 vezes nas páginas 41 e 52.
- Liu, W.; Li, M. Requirements planning with event calculus for runtime self-adaptive system. In: **2015 IEEE 39th Annual Computer Software and Applications Conference**. [S.l.: s.n.], 2015. v. 2, p. 77–82. Citado 2 vezes nas páginas 41 e 48.
- Longbing Cao et al. Integrative early requirements analysis for agent-based systems. In: **Fourth International Conference on Hybrid Intelligent Systems (HIS'04)**. [S.l.: s.n.], 2004. p. 118–123. Citado 2 vezes nas páginas 41 e 51.
- M., C. et al. The gaia methodology process. In: _____. **Handbook on Agent-Oriented Design Processes**. Berlin, Heidelberg: Springer Berlin Heidelberg, 2014. p. 141–172. ISBN 978-3-642-39975-6. Disponível em: <https://doi.org/10.1007/978-3-642-39975-6_6>. Citado 3 vezes nas páginas 41, 44 e 49.
- MORREALE, V. et al. Goal-oriented development of bdi agents: the practionist approach. In: . [S.l.: s.n.], 2006. p. 66–72. Citado 2 vezes nas páginas 41 e 53.
- MURRAY, J. Specifying agent behaviors with uml statecharts and statedit. In: . [S.l.: s.n.], 2003. p. 145–156. Citado 2 vezes nas páginas 41 e 48.
- MYLOPOULOS, J.; CASTRO, J.; KOLP, M. The evolution of tropos. In: _____. [S.l.: s.n.], 2013. p. 281–287. ISBN 978-3-642-36925-4. Citado 4 vezes nas páginas 41, 53, 54 e 62.
- N., G. The comomas methodology and environment for multi-agent system development. **Multi-Agent Systems Methodologies and Applications**, v. 1286, p. 1–16, 1997. Australian Workshop on Distributed Artificial Intelligence. Disponível em: <<https://link.springer.com/chapter/10.1007/BFb0030078>>. Citado 3 vezes nas páginas 41, 44 e 53.
- PADGHAM, L.; THANGARAJAH, J.; WINIKOFF, M. Prometheus research directions. In: _____. [S.l.: s.n.], 2014. v. 9783642544323, p. 155–171. Citado 4 vezes nas páginas 41, 44, 49 e 53.
- PASSOS, L. S.; ROSSETTI, R. J.; GABRIEL, J. Chapter 3 - an agent methodology for processes, the environment, and services. In: ROSSETTI, R. J.; LIU, R. (Ed.). **Advances in Artificial Transportation Systems and Simulation**. Boston: Academic Press, 2015. p. 37 – 53. ISBN 978-0-12-397041-1. Disponível em: <<http://www.sciencedirect.com/science/article/pii/B9780123970411000030>>. Citado 2 vezes nas páginas 41 e 52.

- PRESSMAN, R.; MAXIM, B. **Engenharia de Software-8ª Edição**. [S.l.]: McGraw Hill Brasil, 2016. Citado 2 vezes nas páginas 24 e 30.
- RANJAN, P.; MISRA, A. A novel approach of requirements analysis for agent based system. In: . [S.l.: s.n.], 2006. p. 299–304. ISBN 0-7695-2611-X. Citado 2 vezes nas páginas 41 e 50.
- RAO, A. S.; GEORGEFF, M. P. et al. Bdi agents: from theory to practice. In: **ICMAS**. [S.l.: s.n.], 1995. v. 95, p. 312–319. Citado na página 28.
- REGNELL, B. Requirements engineering with use cases—a basis for software development. **Department of Communication Systems, Lund Institute of Technology. Lund University, Sweden**, p. 33, 1999. Citado na página 33.
- RIBINO, P. et al. Ontology and goal model in designing bdi multi-agent systems. In: . [S.l.: s.n.], 2013. v. 1099. Citado 3 vezes nas páginas 41, 52 e 54.
- RODRIGUEZ, L. et al. Improving the quality of agent-based systems: Integration of requirements modeling into gaia. In: . [S.l.: s.n.], 2009. p. 278–283. Citado 2 vezes nas páginas 41 e 50.
- RONALD, N. et al. On the engineering of agent-based simulations of social activities with social networks. **Information and Software Technology**, v. 54, n. 6, p. 625 – 638, 2012. ISSN 0950-5849. Special Section: Engineering Complex Software Systems through Multi-Agent Systems and Simulation. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S0950584911002485>>. Citado 2 vezes nas páginas 41 e 52.
- ROSENBERG, D.; SCOTT, K. **Use case driven object modeling with UML**. [S.l.]: Springer, 1999. Citado na página 33.
- RUSSELL, S. J.; NORVIG, P. **Artificial intelligence: a modern approach**. [S.l.]: Malaysia; Pearson Education Limited,, 2016. Citado 2 vezes nas páginas 29 e 30.
- SEN, A.; HEMACHANDRAN, K. Elicitation of goals in requirements engineering using agile methods. In: . [S.l.: s.n.], 2010. p. 263–268. Citado 2 vezes nas páginas 41 e 49.
- Sen, A. M.; Jain, S. K. An agile technique for agent based goal refinement to elicit soft goals in goal oriented requirements engineering. In: **15th International Conference on Advanced Computing and Communications (ADCOM 2007)**. [S.l.: s.n.], 2007. p. 41–47. Citado 2 vezes nas páginas 41 e 51.
- SHEN, Z. et al. Goal-oriented methodology for agent system development. In: . [S.l.: s.n.], 2005. E89-D, p. 95–101. Citado 2 vezes nas páginas 41 e 50.
- SOMMERVILLE, I. Software engineering 9th edition. **ISBN-10**, v. 137035152, p. 18, 2011. Citado na página 33.
- SUTCLIFFE, A. Requirements engineering for complex collaborative systems. In: . [S.l.: s.n.], 2001. p. 110–117. ISBN 0-7695-1125-2. Citado 2 vezes nas páginas 41 e 51.
- ULFAT-BUNYADI, N.; MOHAMMADI, N.; HEISEL, M. Supporting the systematic goal refinement in kaos using the six-variable model. In: . [S.l.: s.n.], 2018. p. 102–111. Citado 2 vezes nas páginas 41 e 52.

UML, O. **Unified modeling language version 2.5. 1. Unified modelling**. 2019. Citado na página 33.

VICARI, R. M.; GLUZ, J. C. An intelligent tutoring system (its) view on aose. **International Journal of Agent-Oriented Software Engineering**, Inderscience Publishers, v. 1, n. 3-4, p. 295–333, 2007. Citado na página 34.

VIRUEL, M. L. R. Requirements modeling for multi-agent systems. 2011. Citado na página 23.

WANG, Y. et al. Plant: A pattern language for transforming scenarios into requirements models. **International Journal of Human-Computer Studies**, v. 71, p. 1026–1043, 11 2013. Citado 2 vezes nas páginas 41 e 52.

Wilmann, D.; Sterling, L. Guiding agent-oriented requirements elicitation: Homer. In: **Fifth International Conference on Quality Software (QSIC'05)**. [S.l.: s.n.], 2005. p. 419–424. Citado 5 vezes nas páginas 41, 51, 54, 62 e 109.

WOOLDRIDGE, M. **An introduction to multiagent systems**. [S.l.]: John Wiley & Sons, 2009. Citado 3 vezes nas páginas 23, 27 e 28.

WOOLDRIDGE, M.; JENNINGS, N. R. Agent theories, architectures, and languages: A survey. In: WOOLDRIDGE, M. J.; JENNINGS, N. R. (Ed.). **Intelligent Agents**. Berlin, Heidelberg: Springer Berlin Heidelberg, 1995. p. 1–39. ISBN 978-3-540-49129-3. Citado 2 vezes nas páginas 23 e 27.

Wu, H.; Liu, L.; Ma, W. Optimizing requirements elicitation with an i* and bayesian network integrated modelling approach. In: **2010 IEEE 34th Annual Computer Software and Applications Conference Workshops**. [S.l.: s.n.], 2010. p. 182–188. Citado 2 vezes nas páginas 41 e 51.

ZAMBONELLI, F.; JENNINGS, N.; WOOLDRIDGE, M. Developing multiagent systems: The gaia methodology. v. 12, 11 2003. Citado na página 49.

ÍNDICE

BDI, 11, 24–26, 28, 37, 38, 40, 44, 47–49,
51–54, 62, 109

ER, 11, 23–26, 30, 37, 38, 40, 44, 45, 47,
49, 52–54, 57, 62, 109

SMA, 11, 23–27, 30, 34, 37, 38, 43–45,
49–54, 57, 63, 109

SWEBOK, 24–26, 30, 31, 33, 37, 38, 40,
44, 47, 57, 109