



UNIVERSIDADE FEDERAL DE SANTA CATARINA
CENTRO TECNOLÓGICO
CURSO DE GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

Mariana Mazzo Heitor

Sistema Multiagente para Verificação de Indício de Fraude em Licitações

Florianópolis
2025

Mariana Mazzo Heitor

Sistema Multiagente para Verificação de Indício de Fraude em Licitações

Trabalho de Conclusão de Curso do Curso de Graduação em Ciência da Computação do Centro Tecnológico da Universidade Federal de Santa Catarina para a obtenção do título de bacharel em Ciência da Computação.
Orientadora: Jerusa Marchi

Ficha de identificação da obra

A ficha de identificação é elaborada pelo próprio autor.

Orientações em:

<http://portalbu.ufsc.br/ficha>

Mariana Mazzo Heitor

Sistema Multiagente para Verificação de Indício de Fraude em Licitações

Este Trabalho de Conclusão de Curso foi julgado adequado para obtenção do Título de “bacharel em Ciência da Computação” e aprovado em sua forma final pelo Curso de Graduação em Ciência da Computação.

Florianópolis, 02 de Dezembro de 2025.

Lúcia Helena Pacheco
Coordenadora do Curso

Banca Examinadora:

Jerusa Marchi
Orientadora

Prof. Maicon Rafael Zatelli
Avaliador
Universidade Federal de Santa Catarina

Prof. Ricardo Azambuja Silveira
Avaliador
Universidade Federal de Santa Catarina

Este trabalho é dedicado aos meus amados pais.

AGRADECIMENTOS

Agradeço à minha família, especialmente meus pais e meu irmão, por todo amor, apoio e incentivo durante toda minha vida e por serem meu porto seguro. Não chegaria onde estou sem vocês e não sou capaz de expressar o tamanho da minha gratidão e do meu amor.

Às minhas amigas, por estarem sempre ao meu lado, me apoiando e me ajudando no que eu precisasse, e por todos os momentos que compartilhamos. Não imagino mais minha vida sem vocês nela.

À minha orientadora, professora Jerusa Marchi, pela paciência e orientação durante o desenvolvimento do trabalho e por todo apoio e carinho durante essa jornada. Você foi uma parte essencial para tornar a experiência mais leve e prazerosa.

Por fim, agradeço à UFSC, por todo ensinamento e contribuição para minha formação, e ao CÉOS, pela oportunidade de trabalhar em um projeto interessante e de alta relevância social.

RESUMO

Ao haver demanda de contratação de obras, serviços ou compras por parte do órgão público, a Administração Pública deve necessariamente abrir um processo licitatório. Apesar de se tratar de um procedimento que visa garantir uma competição justa entre os licitantes, ele está sujeito à corrupção, podendo ocorrer fraude durante qualquer uma de suas etapas. Como consequência disso, é atribuída aos promotores a função de exercer a apuração de possíveis fraudes, havendo diversas metodologias que buscam garantir seu combate e sua prevenção. Entretanto, essa verificação pode se tornar complexa com a extensa quantidade de dados a serem analisados. Para isso, a tecnologia de Sistema Multiagente busca, por meio de um conjunto de agentes inteligentes, automatizar esse processo ao simular o comportamento dos promotores na investigação da licitação, percebendo informações do documento e trocando informações entre si. Nesse contexto, este trabalho pretende apresentar um sistema multiagente capaz de identificar indícios de irregularidades de um grupo de hipertipologias - conceito que representa um conjunto de verificações de indício de fraude dentro de uma categoria de irregularidade - baseado nas principais categorias de fraude licitatória. O sistema multiagente desenvolvido contém diversos agentes inteligentes, agrupados em uma organização, divididos em um grupo por hipertipologia. Dessa forma, cada grupo é responsável por buscar os indicativos de fraude presentes na hipertipologia pela qual é responsável, retornando um alerta caso seja detectado. O sistema desenvolvido foi validado a partir de um estudo de caso elaborado por meio de dados fictícios, demonstrando, apesar de algumas falhas e limitações, um resultado satisfatório e, com isso, revelando-se uma alternativa viável e promissora no processo de automatização da investigação de fraudes em procedimentos de licitação.

Palavras-chave: Agente Inteligente. Sistema Multiagente. Licitação. Fraude em Licitação.

ABSTRACT

When there is a demand for contracting works, services or purchases by a public agency, the Brazilian Public Administration must necessarily open a bidding process. Although this is a procedure that aims to ensure fair competition among bidders, it is subject to corruption, and fraud can occur during any of its stages. As a result, prosecutors are tasked with investigating possible fraud, and there are several methodologies that seek to combat and prevent it. However, this verification can become complex due to the extensive amount of data to be analyzed. To this end, Multi-Agent System technology seeks, through a set of intelligent agents, to automate this process by simulating the behavior of prosecutors in the investigation of the bidding process, perceiving information from the bidding document and exchanging information among themselves. In this context, this work aims to present a multi-agent system capable of identifying evidence of irregularities in a group of hyper-typologies - concept representing a set of checks for evidence of fraud within a category of irregularity - based on the main categories of bidding fraud. The multi-agent system developed contains several intelligent agents, grouped into an organization, divided into a group by hyper-typology. Thus, each group is responsible for searching for signs of fraud present in the hyper-typology for which it is responsible, returning an alert if detected. The developed system was validated based on a case study prepared using fictitious data, demonstrating, despite some flaws and limitations, a satisfactory result and, thus, proving to be a viable and promising alternative in the process of automating fraud investigation in bidding procedures.

Keywords: Intelligent Agent. Multi-Agent System. Bidding. Bidding Fraud.

LISTA DE FIGURAS

Figura 1 – Tabela de hipertipologias analisadas.	13
Figura 2 – Fases do processo licitatório.	16
Figura 3 – Diagrama das dimensões de um SMA.	21
Figura 4 – Diagrama dos principais conceitos de um agente.	22
Figura 5 – Diagrama de uma arquitetura BDI.	24
Figura 6 – Diagrama da estrutura de um Sistema Multiagente.	25
Figura 7 – Diagrama dos principais conceitos de um ambiente.	26
Figura 8 – Diagrama dos principais conceitos de uma organização.	28
Figura 9 – Modelagem da arquitetura dos grupos da organização.	34
Figura 10 – Modelagem da funcionalidade do grupo <i>groupHipertipologia</i>	35
Figura 11 – Modelagem da funcionalidade do grupo <i>groupHipertipologiaB</i>	36
Figura 12 – Fluxo de interação dos agentes em <i>resultEdital</i> do agente <i>gerente</i>	39
Figura 13 – Fluxo de interação dos agentes em <i>sendEdital</i> do agente <i>hipertipologiaB</i>	41
Figura 14 – Fluxo de geração de alertas do artefato <i>ArtefatoHipertipologia</i>	49
Figura 15 – Fluxo de interação dos agentes no retorno de resultado do agente <i>tipo-</i> <i>logia8</i>	55
Figura 16 – Alertas gerados no banco de dados após a execução do sistema.	63

LISTA DE CÓDIGOS

3.1	<i>Plan start</i> do agente <i>gerente</i>	36
3.2	Funções <i>buscaEdital</i> e <i>consultaSQL</i> do artefato <i>ArtefatoHipertipologia</i>	38
3.3	Reação à mudança da propriedade <i>resultEdital</i> do agente <i>gerente</i>	39
3.4	<i>Plan</i> de junção à organização e adoção de papel do agente <i>hipertipologiaB</i>	40
3.5	<i>Plan sendEdital</i> do agente <i>hipertipologiaB</i> para envio de editais.	41
3.6	<i>Plan saveResultTipologia</i> do agente <i>hipertipologiaB</i> para salvar resultado obtido da tipologia.	42
3.7	Definição de regras <i>hypertipologiaSemFraude</i> e <i>analiseFinalizada</i> do agente <i>hipertipologiaB</i>	42
3.8	Ações do agente <i>hipertipologiaB</i> ao identificar hipertipologia sem fraude.	43
3.9	Definição de regras <i>hypertipologiaIncompleta</i> , <i>getGravidade</i> e <i>maiorGravidade</i> do agente <i>hipertipologiaB</i>	44
3.10	Ações do agente <i>hipertipologiaB</i> ao identificar hipertipologia incompleta.	44
3.11	Funções <i>artefatoAlerta</i> e <i>gerarAlerta</i> do artefato <i>artefatoHipertipologia</i>	45
3.12	Template pré-definido para tipologias da hipertipologia B.	47
3.13	Ações do agente <i>hipertipologiaB</i> ao identificar hipertipologia completa.	49
3.14	<i>Plan adopt_role</i> dos agentes <i>tipologia</i> da hipertipologia B.	50
3.15	Funcionalidade de análise de uma tipologia pelo agente <i>tipologia</i>	52
3.16	Funções <i>consultaTipologia8</i> e <i>consultaSQL</i> do artefato <i>ArtefatoHipertipologiaB</i>	52
3.17	Função <i>salvaJSON</i> do artefato <i>ArtefatoHipertipologiaB</i>	55

SUMÁRIO

	Sumário	10
1	INTRODUÇÃO	11
1.1	DESCRIÇÃO DO PROBLEMA	11
1.2	MOTIVAÇÕES E JUSTIFICATIVAS	12
1.3	OBJETIVOS	13
1.3.1	Objetivo Geral	13
1.3.2	Objetivos Específicos	13
1.4	ESTRUTURA DO TRABALHO	14
2	FUNDAMENTAÇÃO TEÓRICA	15
2.1	LICITAÇÕES E PROCESSOS LICITATÓRIOS	15
2.1.1	Modalidades de Licitações	16
2.1.2	Fraudes em Processos Licitatórios	18
2.1.3	Categorias de Fraudes	18
2.2	TECNOLOGIA DE AGENTES E SISTEMAS MULTIAGENTES	21
2.2.1	Agentes (Jason)	21
2.2.2	Sistemas Multiagente	24
2.2.3	Artefato (CArtAgO)	25
2.2.4	Organização (Moise)	27
2.3	PROJETO CÉOS	29
2.4	ESTADO DA ARTE	30
2.4.1	A multi-agent data mining system for cartel detection in Brazilian government procurement	30
2.4.2	A Multi-Agent System Based Approach to Fight Financial Fraud: An Application to Money Laundering	31
3	CONSTRUÇÃO DA PROPOSTA	32
3.1	MODELAGEM DO SISTEMA MULTIAGENTE	32
3.2	DETALHAMENTO	36
3.3	ESTUDO DE CASO	56
4	RESULTADOS	61
5	CONCLUSÃO E TRABALHOS FUTUROS	65
	REFERÊNCIAS	67
	APÊNDICE A – ARTIGO	70
	APÊNDICE B – CÓDIGO FONTE	88

1 INTRODUÇÃO

Licitação se trata do procedimento administrativo mediante o qual o poder público seleciona a proposta mais vantajosa para o contrato ou aquisição de um item ou serviço de seu interesse (ENAP, 2014), estimulando a competitividade entre os concorrentes que participam desse procedimento licitatório e oferecendo iguais condições entre eles, garantindo assim a isonomia desde que os que queiram participar do certame preencham os requisitos previamente estabelecidos no edital (MASCENA, 2017). Esse processo pode ser dividido em duas principais fases: interna e externa. A fase interna se trata de planejamento da licitação e elaboração do edital, enquanto a fase externa se trata de publicação do edital e análise, julgamento e seleção da proposta licitatória.

1.1 DESCRIÇÃO DO PROBLEMA

Corrupção é um fenômeno social, político e econômico que consiste em práticas de suborno e propina, fraude, apropriação indébita ou qualquer outro desvio de recursos por parte de um agente público, sendo considerado um crime constitucional. Ela é derivada do infringimento de lei por parte do corrupto e do corruptor, trazendo riscos ao desenvolvimento do Estado e da própria sociedade (RODRIGUES; FILHO, 2017). Vale ressaltar que, de acordo com a OECD (OECD, 2016), as compras públicas são uma das atividades governamentais mais vulneráveis à corrupção, sendo apontado pelo OECD Foreign Bribery Report que mais da metade dos casos ocorre em licitações e quase dois terços dos casos ocorre em setores profundamente associados a elas.

Nesse contexto, tem-se a fraude licitatória como uma vertente da corrupção que compromete a transparência, eficiência, qualidade e economia na utilização dos recursos públicos. Para tentar lidar com a situação, foi regulamentada na legislação a Lei nº 14.133/2021 - implementada em substituição a sua antecedente, a Lei nº 8.666/93 -, que institui no Capítulo II-B os crimes em licitações e contratos públicos, sendo definido entre o Art. 337-E e 337-P as principais ilegalidades relacionadas ao processo. Entretanto, ainda há vulnerabilidade a condutas fraudulentas, que podem ocorrer tanto na fase interna/de preparação quanto na fase externa/de seleção do processo licitatório. Isso advém do interesse em obter vantagem econômica ou política sobre a licitação, podendo ser desencadeado tanto por parte do agente público, ao combinar previamente o resultado com determinado concorrente, quanto das próprias empresas, ao aderir a cartéis e executar estratégias de conluio, eliminando o caráter competitivo do processo. Dessa forma, pode-se deduzir um desacato ao direito constitucional e administrativo que pode prejudicar setores que fornecem bens e serviços ao governo de maneira lícita, reduzir investimentos, comprometer a produtividade e a eficiência econômica e acarretar déficit financeiro (RODRIGUES; FILHO, 2017).

1.2 MOTIVAÇÕES E JUSTIFICATIVAS

Como mencionado previamente, a fraude em licitação é um crime, sendo definido na Constituição Federal suas normas e regras e a penalidade para quem cometê-lo. Mesmo assim, de acordo com a Controladoria Geral da União, ela lidera as irregularidades encontradas em municípios (RODRIGUES; FILHO, 2017), sendo papel dos ministérios públicos estaduais investigar denúncias de possíveis casos. Por isso, com o intuito de auxiliar as apurações, foram desenvolvidas diversas metodologias que buscam garantir, de forma eficiente, sua prevenção e detecção.

Entretanto, mesmo com conhecimento acerca dos principais tipos de fraude e de suas características, a análise licitatória para identificá-la não é tão simples. Devido à grande quantidade de dados que podem ser extraídos de editais e documentos, o tratamento dessa base de conhecimento em busca de informação útil para análise pode ser um trabalho complexo para ser realizado pelos promotores, podendo entrar em questões técnicas que necessitem de especialistas. Não só isso, como a busca manual por demandas específicas torna esse processo de análise demorado, tornando essa atividade mais propensa a não ser examinada de forma eficiente/exaustiva. Devido a isso, pode-se deixar passar detalhes específicos do processo licitatório que podem indicar fraude, principalmente aqueles que são difíceis de serem identificados individualmente. Além das dificuldades técnicas mencionadas, essa análise demanda uma boa quantidade de tempo do promotor na busca completa de dados para avaliar a validade da denúncia, podendo ademais haver desperdício desse tempo ao se tratar de investigações improdutivas ou de repetições de verificações que já foram feitas (WÜST *et al.*, 2024).

Portanto, torna-se interessante a busca por uma tecnologia capaz de auxiliar os promotores na identificação de indícios de fraude em um processo licitatório. Nesse contexto, aplica-se um Sistema Multiagente, uma subárea da Inteligência Artificial que consiste em modelar e simular sistemas complexos como uma organização de agentes autônomos interagindo com um ambiente e entre si (HÜBNER; BOISSIER *et al.*, 2020), devido a possibilidade de adicionar conhecimento de especialista ao agente, permitindo a racionalização a respeito dos dados obtidos e, com isso, promovendo tomadas de decisões mais precisas. Dessa forma, é feita uma analogia com a problemática analisada, em que os agentes simulam o comportamento dos promotores agindo sobre as licitações - representadas pelo ambiente - em busca das informações necessárias para a verificação de fraude, podendo trocar informações com outros promotores e/ou agentes e perceber elementos do documento licitatório analisado.

Esse trabalho focará nas fraudes encontradas dentro de uma hipertipologia - conjunto de verificações de indício de fraude dentro de uma categoria de irregularidade em um procedimento licitatório -, composta de diversas verificações - representadas como tipologias - que, analisadas em conjuntos, irão verificar possibilidade de haver fraude licitatória, baseada na investigação de algumas das principais categorias de fraude licitatória

I. Análise da relação entre empresas licitantes	Hipertipologia A	Empresas participantes de uma mesma licitação compartilham características	01	Empresa licitante com sócio em comum com outra empresa licitante
			02	Empresa licitantes com ex-sócio em comum com outra empresa licitante
			03	Familiares consanguíneos de primeiro grau em comum entre sócios de empresas licitantes
			04	Sócio de uma empresa licitante é empregado de outra empresa licitante
			05	Empresas licitantes com contador em comum
			06	Empresas licitantes estão sediados no mesmo endereço
			07	Empresas licitantes possuem mesmo telefone, e-mail ou grupo econômico
	Hipertipologia B	Empresas participantes de uma mesma licitação combinam de uma sempre perder (perdedora contumaz) para outra ganhar, ou alternam vitórias	08	Empresa licitante é perdedora contumaz
			09	Empresa licitante é vencedora contra empresa perdedora contumaz
			10	Alternância de vitórias entre empresas licitantes
	Hipertipologia C	Empresas participantes de uma mesma licitação apresentam propostas com proporcionalidade de valor/preço	11	Empresa apresenta propostas simétricas acima de patamar definido
	Hipertipologia D	Propostas idênticas entre Licitantes em Unidades Gestoras Diferentes	12	Empresa apresenta propostas idênticas acima de patamar definido
	Hipertipologia E	Conluio já identificado	13	Empresas que já tiveram conluio comprovado

Figura 1 – Tabela de hipertipologias analisadas.

Fonte: Projeto CÉOS

- como apresentado na figura 1 - e, assim, definindo uma gravidade baseada na quantidade de indícios descobertos. Para isso, será implementado um sistema multiagente capaz de verificar as informações recebidas e apontar indícios de irregularidade.

1.3 OBJETIVOS

1.3.1 Objetivo Geral

Esse trabalho tem como objetivo implementar diversos agentes inteligentes capaz de verificar dados a respeito de uma licitação com o intuito de identificar indícios de fraude. Os agentes desenvolvidos serão organizados em uma arquitetura de sistema multiagente que visa verificar a existência de diferentes categorias de irregularidade no procedimento licitatório. Com isso, o sistema busca cobrir todas as hipertipologia de um grupo e, por meio de alertas em painéis inteligentes, auxiliar os promotores do Ministério Público de Santa Catarina na detecção de possíveis fraudes licitatórias.

1.3.2 Objetivos Específicos

São objetivos específicos deste trabalho:

- Compreender o funcionamento de sistema multiagente e da plataforma JaCaMo;
- Assimilar conhecimento sobre os principais conceitos de licitação e detecção de fraude em suas principais categorias;
- Identificar oportunidade de aplicação de agentes no contexto de um processo licitatório;

- Modelar a representação do conhecimento e a estrutura dos agentes, integrados em um sistema multiagente, com o intuito de auxiliar o desenvolvimento da solução;
- Definir regras e crenças, estabelecer metas e implementar planos na base de conhecimento dos agentes que possibilitem a análise de uma hipertipologia;
- Desenvolver artefatos que representem os editais licitatórios e demais documentos de análise, permitindo a interação dos agentes;
- Integrar os agentes à uma organização, definindo seus papéis e suas missões dentro do contexto de verificação de fraude no processo licitatório;
- Elaborar um estudo de caso para apresentar e validar a solução implementada.

1.4 ESTRUTURA DO TRABALHO

O trabalho está organizado em cinco capítulos, que buscam aprofundar o tema proposto. O capítulo 2 trata da fundamentação teórica do trabalho, no qual será abordado o conceito de licitação, explicando mais detalhadamente o processo licitatório e suas modalidades e discorrendo sobre a ocorrência de fraude em suas fases, destacando as mais recorrentes. Além disso, serão introduzidos os principais conceitos utilizados acerca da tecnologia de sistema multiagente, detalhando sobre o conceito em si e suas “dimensões”, que são o agente - com uso de Jason -, o artefato - com uso de CArTAgO -, e a organização - com uso de Moise -, e da modelagem, destacando como ela será estruturada. Será, então, fornecida, no mesmo capítulo, uma visão geral a respeito do projeto CÉOS, uma parceria entre o INE e o Ministério Público de Santa Catarina com o objetivo de prevenir e combater a corrupção. Para encerrar, será apresentado o estado da arte do tema, discorrendo sobre alguns trabalhos correlatos. O capítulo 3 trata da construção da proposta, em que será apresentada a modelagem do agente/sistema multiagente elaborado, em seguida detalhando a solução proposta. Por fim, será apresentado um estudo de caso em que será aplicado o agente/sistema multiagente desenvolvido, validando sua funcionalidade. O capítulo 4 trata de apresentar os resultados atingidos com a solução proposta, discorrendo sobre eles e suas implicações. O capítulo 5 apresenta as conclusões obtidas no trabalho e propõe possíveis projetos futuros que poderão ser desenvolvidos a partir desse.

2 FUNDAMENTAÇÃO TEÓRICA

Nesta seção são apresentados os principais conceitos, princípios e abordagens utilizados no desenvolvimento deste trabalho, permitindo a contextualização do problema analisado e da metodologia empregada na solução desenvolvida.

A seção inicial apresenta brevemente o conceito de licitação, indicando as possíveis modalidades, e de fraude licitatória, ressaltando as principais categorias. A seção seguinte apresenta uma visão geral da tecnologia utilizada neste trabalho, com o objetivo de familiarizar o leitor com a nomenclatura que será utilizada no capítulo 4.

2.1 LICITAÇÕES E PROCESSOS LICITATÓRIOS

De acordo com as definições apresentadas por (TCU, 2024) e (SILVA; OLIVEIRA, 2024), licitação é o processo administrativo realizado pela Administração Pública com o objetivo de, baseado em condições estabelecidas previamente em um edital, selecionar a proposta mais vantajosa para a contratação e aquisição de bens e serviços, enquanto garante o caráter competitivo, preserva igualdade de condições entre os licitantes e promove o interesse público de forma eficiente. Para isso, além do custo-benefício inicial de aquisição, deve ser considerada a vida útil do objeto licitado, com o objetivo de prevenir contratações que gerem custos maiores a longo prazo. Portanto, exceto em casos de dispensa e inexigibilidade, observa-se a obrigatoriedade da licitação, promovendo uma gestão responsável e transparente dos recursos públicos e garantindo competitividade e imparcialidade nas contratações públicas. O processo licitatório é amparado pela Lei nº 14.133/2021, que busca estabelecer normas, procedimentos, sanções e mecanismos preventivos que regem o funcionamento de uma licitação.

Para estar em conformidade com a lei, uma licitação deve ser condicionada a diversos princípios norteadores, apresentados no Art. 5 da Lei nº 14.133/2021 (BRASIL, 2021). Alguns dos principais que costumam ser alvos de violação em fraudes, de acordo com (TCU, 2024), são a competitividade, a legalidade, a isonomia, a impessoalidade, a moralidade e probidade administrativa, a publicidade, a vinculação ao instrumento convocatório e o julgamento objetivo.

Um processo licitatório pode ser representado por meio de sete fases, divididas em interna e externa, que são apresentadas no Art. 17 da Lei nº 14.133/2021 (BRASIL, 2021). Durante a fase interna, há a fase preparatória, que consiste no planejamento interno da licitação. Por outro lado, a fase externa é composta pelas fases: de divulgação do edital, que se trata da abertura pública da licitação; de apresentação de propostas e lances, que representa o momento em que os licitantes realizam o envio das ofertas; de julgamento, que é quando é realizada a avaliação das propostas; habilitação, que corresponde à verificação da capacidade do licitante vencedor; recursal, que oferece aos participantes a oportunidade de contestar o resultado do julgamento; e de homologação, que oficializa o vencedor do

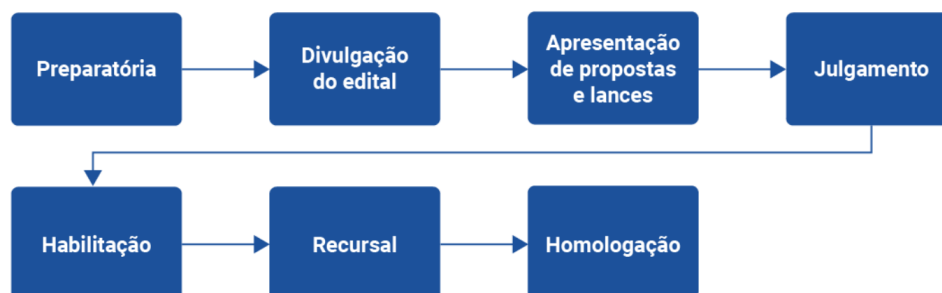


Figura 2 – Fases do processo licitatório.

Fonte: (TCU, 2024)

processo, seguindo com a assinatura do contrato. A figura 2 apresenta essas fases, indicando o fluxo em que elas ocorrem.

2.1.1 Modalidades de Licitações

Ao iniciar um processo licitatório, é necessário definir um formato específico para conduzi-lo, considerando alguns critérios pré-definidos (TCU, 2024). Dessa forma, a modalidade se trata das regras específicas que devem ser seguidas de acordo com a natureza do objeto da licitação (PORTAL DA TRANSPARÊNCIA, s.d.).

A Lei nº 14.133/2021, no Art. 28, estabelece pregão, concorrência, concurso, leilão e diálogo competitivo como as cinco modalidades de licitação disponíveis (BRASIL, 2021). A principal diferença em relação à sua antecessora, a Lei nº 8.666/1993, trata-se da remoção das modalidades de tomada de preços e de convite e da adição da modalidade de diálogo competitivo, além de a escolha de qual modalidade utilizar passar a depender da natureza do objeto da licitação e do critério de julgamento escolhido, não tendo mais o valor da contratação como requisito (TCU, 2024). Vale ressaltar também que, de acordo com o mesmo artigo - Art. 28 da Lei nº 14.133/2021 -, é possível a utilização de procedimentos auxiliares, porém sendo vedada a combinação das existentes ou criação de novas modalidades licitatórias (BRASIL, 2021). De forma complementar à modalidade da licitação, há os critérios de julgamento, que serão utilizados para selecionar o vencedor do processo licitatório. Eles são organizados, de acordo com o Art. 33 da Lei nº 14.133/2021, em menor preço, maior desconto, melhor técnica ou conteúdo artístico, técnica e preço, maior lance - sendo exclusivo de leilão - e maior retorno econômico (BRASIL, 2021).

Pregão se trata da modalidade exercida em aquisição de bens e serviços comuns - os que tem padrões de desempenho e qualidade objetivamente especificáveis pelo edital - e de serviços comuns de engenharia - os que, por objeto, ações objetivamente padronizáveis em termos de desempenho e qualidade, manutenção, adequação e adaptação de bens, preservando suas características originais. Ele deve ser conduzido, preferencialmente, de maneira eletrônica, sendo aceita a presencial apenas como exceção e com a garantia de registro em

ata e gravação em áudio e vídeo. Para essa modalidade, os critérios de julgamento adotados são de menor preço ou de maior desconto, sendo necessário garantir despesa reduzida ao mesmo tempo que alcança os parâmetro mínimos de qualidade definidos no edital. Isso é feito a fim de assegurar a contratação mais vantajosa à Administração Pública (TCU, 2024).

Concorrência se trata da modalidade exercida na contratação de bens e serviços especiais e de obras e serviços comuns e especiais de engenharia. Vale ressaltar que, entre os serviços licitados, pode-se incluir os técnicos de natureza prevalentemente intelectual. Para essa modalidade, os critérios de julgamento são de menor preço, de melhor técnica ou conteúdo artístico, de técnica e preço, de maior retorno econômico ou de maior desconto. Ou seja, o único critério não admitido nessa modalidade é o de maior lance. Com isso, nota-se que, diferente de sua antecessora, a Lei nº 14.133/2021 deixou de utilizar o valor da contratação como critério determinante na escolha dessa modalidade (TCU, 2024).

Concurso se trata da modalidade exercida em escolha de trabalho técnico, científico ou artístico. Não se deve, então, aplicá-la em contratação de cargo/emprego público ou serviços de mão de obra terceirizada. Para essa modalidade, o critério de julgamento adotado é de melhor técnica ou conteúdo artístico, havendo concessão de prêmio ou remuneração ao vencedor. Vale comentar que a utilização dessa modalidade é recomendada em contratação de projetos, tanto com o objetivo de lidar com possíveis deficiências quanto devido aos resultados insatisfatórios observados em modelos tradicionais de licitação e contratação (TCU, 2024).

Leilão se trata da modalidade exercida em cessão de bens imóveis ou de bens móveis inservíveis ou legalmente apreendidos a quem oferecer o maior lance. Essa é uma diferença introduzida em relação à Lei nº 8.666/1993, que anteriormente previa a realização da alienação de bens imóveis por meio da modalidade de concorrência e, apenas em casos específicos, de leilão. Para essa modalidade, o critério de julgamento adotado é o de maior lance. Antes de ser leiloado, entretanto, deverá haver uma avaliação do bem pela Administração com o intuito de definir um preço mínimo e as condições de pagamento. Dessa forma, caso a proposta do primeiro colocado esteja abaixo desse valor, poderá ser feita uma negociação em busca de condições mais vantajosas com ele e, caso ainda não alcance essa quantia, com os demais licitantes, respeitando a ordem de classificação no processo licitatório (TCU, 2024).

Diálogo Competitivo se trata da modalidade exercida na contratação de obras, serviços e compras na qual a Administração Pública seleciona, por meio de diálogos com licitantes previamente escolhidos, soluções que atendam às suas necessidades. As propostas devem, então, ser apresentadas após o encerramento dessa etapa. De acordo com a Lei nº 14.133/2021, Art. 32, para que possa ser adotada essa modalidade, é necessário que algumas das seguintes situações seja satisfeita: inexistência de solução que atenda à demanda, necessitando de inovação; necessidade de adaptação de solução existente

para atender à exigência; dificuldade em especificar precisamente o objeto da licitação; necessidade de coleta de informações adicionais a respeito da solução mais adequada, dos requisitos da solução estabelecida ou dos aspectos jurídicos e financeiros da contratação (TCU, 2024).

2.1.2 Fraudes em Processos Licitatórios

Fraude consiste na intenção de realizar uma ação ilícita visando a obtenção de benefícios indevidos, a partir de métodos desonestos e de má-fe, para si próprio ou para terceiros. Nesse contexto, tem-se a fraude licitatória como o acordo prévio entre licitantes ou a manipulação de editais visando beneficiar, financeiramente e administrativamente, empresas ou indivíduos, enquanto se manifesta por meio de direcionamento de editais, conluio no ajuste de preços ou superfaturamento de contratos (SILVA; OLIVEIRA, 2024). Vale ressaltar que, de acordo com (TRANSPARÊNCIA BRASIL, 2024), a fraude não precisa necessariamente ocorrer com sucesso para ser levada em consideração, tendo mais peso a intenção de cometê-la do que o resultado obtido.

Observa-se, assim, dois principais objetivos da fraude em licitação: o comprometimento da competitividade, que é essencial em um processo licitatório, a fim de garantir controle do resultado do certame; e a apropriação de recursos públicos, por meio da contratação com superfaturamento, visando um benefício financeiro. Vale ressaltar que, apesar de comumente as fraudes estarem associadas à apropriação de recursos, pode ocorrer o interesse em ingressar no mercado de obras públicas, não visando o lucro por meio do superfaturamento. Nessa situação, devido à exigência de documentação comprovando qualificação e experiência prévia, a área de atuação da fraude costuma ser a simulação ou falsificação desses documentos (TRANSPARÊNCIA BRASIL, 2024).

Dessa forma, nota-se que se trata de uma prática que impacta negativamente na eficiência de aplicação e integridade dos recursos públicos, subvertendo a finalidade dos processos licitatórios e, com isso, gerando prejuízos econômicos e sociais à população. Além disso, compromete a transparência e equidade nas contratações públicas, intensificando a desconfiança institucional e governamental, e a qualidade dos serviços públicos. Para isso, a Lei nº 14.133/2021 estabelece sanções, prevendo penalidades severas de acordo com a gravidade do ato e o prejuízo causado, e medidas preventivas, como a exigência de transparência em todas as etapas do processo licitatório (SILVA; OLIVEIRA, 2024).

2.1.3 Categorias de Fraudes

Com o objetivo de manipular e direcionar a contratação, restringir a competição, limitando a concorrência, e simular a idoneidade do processo, infratores costumam utilizar diferentes mecanismos fraudulentos, resultando na violação de princípios norteadores de uma licitação. Observa-se, então, que as principais categorias de fraudes são: projeto mágico; edital restritivo; publicidade precária; julgamento negligente, conivente ou defici-

ente; contratação direta indevida; e conluio ou cartelização (TRANSPARÊNCIA BRASIL, 2024).

Projeto Mágico consiste em especificações incompreensíveis, incompletas, defeituosas, direcionadas ou restritivas. As especificações a respeito do objeto a ser licitado devem ser objetivas, claras e suficientemente detalhadas, de modo a assegurar a isonomia entre os participantes. Entretanto, deve haver um equilíbrio no grau de especificação, não admitindo descrições genéricas, imprecisas ou incompletas nem detalhamento excessivo - para não direcionar a licitação ou restringir seu caráter competitivo. Outras duas situações que se encaixam dentro dessa tipologia são: vínculo entre projetista e licitante, sendo proibida a participação, de forma direta ou indireta - definida como a existência de qualquer vínculo técnico, comercial, econômico, financeiro ou trabalhista com o participante -, na licitação por parte do autor do projeto; e fraude na pesquisa de preços - responsável pela verificação de preços no mercado e definição de um valor de referência -, por meio de simulação ou negligência deliberada (TRANSPARÊNCIA BRASIL, 2024).

Edital Restritivo se trata da inclusão de cláusulas restritivas nos editais de licitação - responsáveis por definir condições necessárias para a participação de um processo licitatório -, comprometendo o caráter competitivo por meio do direcionamento indevido a fim de favorecer um licitante por meio da exclusão indevida da participação dos concorrentes ao incluir requisitos desnecessários e específicos. Essa prática se trata de um dos mecanismos mais comuns de fraude, definindo no edital exigências que restringem a competitividade ao impor condições que inviabilizam a participação de licitantes. As principais cláusulas e condições restritivas observadas são relativas à habilitação, incluindo qualificação econômico-financeira e técnica e regularidade fiscal e trabalhista, e a condições de participação (TRANSPARÊNCIA BRASIL, 2024).

Publicidade Precária indica a restrição de publicidade da licitação, limitando a quantidade de participantes que terão conhecimento a respeito da existência do processo licitatório. Apesar de se tratar de uma prática obrigatória, é comum não haver divulgação adequada da licitação. Observa-se então, como principais formas de restrição, o prazo útil restrito, publicando o aviso da licitação em uma sexta-feira ou pré-feriado e, com isso, reduzindo os dias úteis para conhecimento e participação do processo; a insuficiência de informação, que é quando as informações contidas no aviso são apresentadas de forma pouco compreensível ou atrativa, como a descrição inadequada do objeto licitado; e a imposição de dificuldade de acesso ao edital, estabelecendo exigências desproporcionais e restritivas para a efetiva participação, dificultando-a mesmo com a publicação do aviso em todos os meios previstos na lei (TRANSPARÊNCIA BRASIL, 2024).

Julgamento Negligente, Conivente ou Deficiente refere-se a fraudes relacionadas com a autoridade julgadora. Conivente é quando há a participação direta dos responsáveis pelo órgão licitante, buscando direcionar resultados e comprometer o caráter competitivo do certame. Negligente é quando há negligência, passando despercebido erros evidentes e

falhas notórias, colaborando, assim, para a fraude ao não exercer corretamente seu papel de controle interno. Deficiente é quando possui origem na conduta de agentes externos ao órgão contratante, visando a elaboração de elementos com aparência de credibilidade e legitimidade, e, assim, dificultando sua detecção. Ocorrências comumente relacionadas a essas situações são: propostas fictícias ou de cobertura, evitando investigações ao simular uma situação de aparente competitividade ao processo licitatório; empresas-fantasma ou de fachada, representando a criação de empresas sem efetiva atuação no mercado, visando criar uma falsa impressão de competitividade ao certame; coincidência de representantes e responsáveis técnicos, sendo vedada a possibilidade de uma mesma pessoa atuar por mais de um concorrente devido ao possível comprometimento da isonomia ao violar o sigilo das propostas; vínculos entre licitantes e servidores públicos; e documentos falsos, forjando documentos que comprovem a qualificação técnica da empresa (TRANSPARÊNCIA BRASIL, 2024).

Contratação Direta Indevida corresponde ao uso da contratação direta, prevista em lei, de forma ilícita. Apesar da realização da licitação ser regra geral em contratação de bens, obras e serviços, há situações excepcionais em que ela poderá não acontecer, na hipótese de ser legalmente inexigível, dispensada ou dispensável. Entretanto, essa modalidade pode ser utilizada de forma indevida, dando margem à ocorrência de fraudes. Os principais tipos observados são: fracionamento de despesa para dispensar, que é quando há a divisão de um montante global que ultrapassa os limites previstos para a dispensa em valores menores, caracterizando-se por fracionar a despesa com o intuito de realizar uma contratação direta; emergência indevida, que é quando a emergência é artificialmente criada, tanto pela sua inexistência quanto por ter ocorrido em função da negligência administrativa intencional; falsa exclusividade, que é quando, apesar da existência de outros possíveis fornecedores, o licitante é considerado o único capaz de fornecê-lo; e falsa singularidade, que é quando, apesar da existência de outros bens ou serviços equivalentes no mercado, é alegada unicidade para o objeto licitado (TRANSPARÊNCIA BRASIL, 2024).

Cartelização, também conhecida como conluio ou cartel, é caracterizado como um acordo, explícito ou implícito, entre licitantes para realizar combinação de propostas e ajuste de vantagens, buscando reduzir ou eliminar a concorrência com o objetivo de favorecer a fixação de preços superiores aos de mercado, aumentando o lucro dos envolvidos. Cartel é uma prática comum em processos licitatórios, favorecendo irregularmente empresas que, a partir de acordo entre si, restringem o caráter competitivo do processo licitatório. Observa-se, então, como situações mais comuns nesse esquema: propostas fictícias ou de cobertura; supressão de propostas, que é quando, por meio de acordos entre os concorrentes, a empresa se abstém de concorrer ou retira uma proposta a fim de garantir a vitória de outra; propostas rotativas ou rodízio, que é quando há alternância de empresa/proposta vencedora; e divisão de mercado, que é quando os licitantes definem um limite de

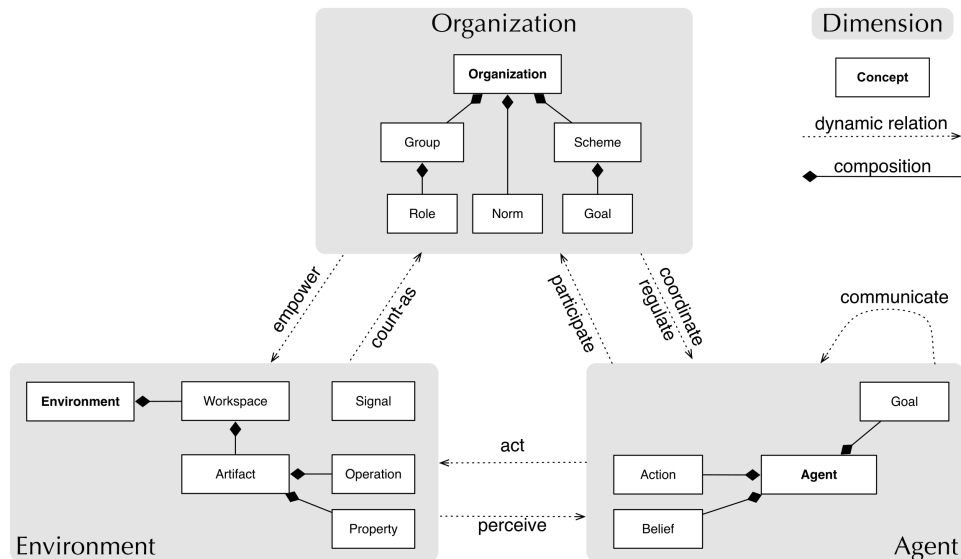


Figura 3 – Diagrama das dimensões de um SMA.

Fonte: (HÜBNER; BOISSIER *et al.*, 2020)

atuação no mercado, acordando em não concorrer entre si em determinados segmentos, localidades ou frentes de contratação, podendo incluir a divisão de lotes licitatórios entre os participantes do conluio (TRANSPARÊNCIA BRASIL, 2024).

2.2 TECNOLOGIA DE AGENTES E SISTEMAS MULTIAGENTES

Um Sistema Multiagente (*SMA*) pode ser dividido em três diferentes abstrações de dimensões: agente, ambiente e organização. Para implementá-las, este trabalho utilizará a tecnologia JaCaMo, que corresponde à junção da linguagem Jason com os frameworks CArtAgO e Moise. A figura 3 apresenta um diagrama das relações entre as diferentes dimensões utilizadas na elaboração de um *SMA*. Essas dimensões e suas respectivas tecnologias serão detalhadas nas subseções seguintes.

2.2.1 Agentes (Jason)

De acordo com (HÜBNER; BORDINI; WOOLDRIDGE, 2007) e (HÜBNER; BOISSIER *et al.*, 2020), o conceito de agente se trata de uma abstração para programar entidades reativas que exibem um certo grau de autonomia. Dessa forma, ao atribuímos uma tarefa ou missão, ele é capaz de determinar por si próprio como completá-la, tomando decisões com flexibilidade e dinamismo ao reagir a eventos enquanto proativamente direciona seu comportamento para satisfazer suas metas. Uma de suas principais características é o fato de estar situado em um ambiente, podendo percebê-lo e comunicar-se com ele, assim como realizar ações capazes de alterar seu estado. Dessa forma, por meio de interações com o ambiente e com outros agentes, ele é capaz de atingir seus objetivos de forma autônoma,

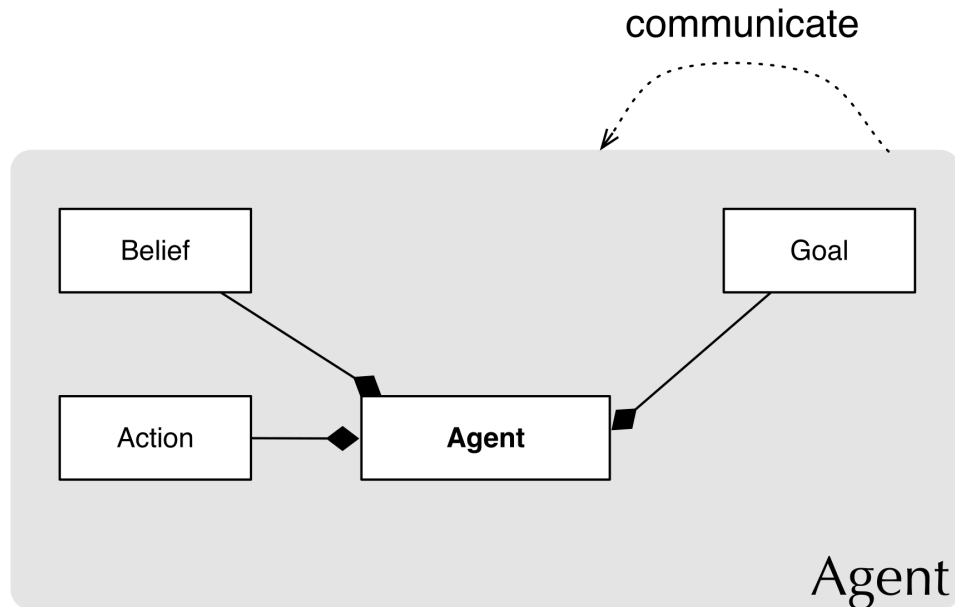


Figura 4 – Diagrama dos principais conceitos de um agente.

Fonte: (HÜBNER; BOISSIER *et al.*, 2020)

racionalizando, em tempo de execução, sobre quais metas irá perseguir e qual é a melhor ação a ser realizada para alcançá-las.

Um agente possui três conceitos fundamentais que regem seu funcionamento: *goal*, *belief* e *action*. Um *goal* é responsável por representar uma situação/configuração que o agente deseja alcançar, tratando-se de uma representação explícita de suas metas e seus objetivos durante a execução. Seu propósito é orientá-lo a realizar determinada ação que ajude-o a atingir o que almeja realizar. Uma *belief* é a representação do conhecimento que o agente possui, fornecendo a informação disponível para auxiliar seu raciocínio e guiar sua decisão a respeito de qual é a melhor estratégia a seguir. É importante ressaltar que, como o próprio nome informa, se trata de uma crença - algo em que acredita-se ser verdade - que o agente possui, podendo estar incorreta ou incompleta. Uma nova *belief* pode ser adquirida por dois métodos distintos: a comunicação com outros agentes e a percepção do estado do ambiente no qual ele se encontra. E, por fim, uma *action*, que trata-se de qual medida o agente irá decidir implementar para atingir suas metas. Ela pode ser classificada em três tipos distintos: externa, que é responsável pela alteração do ambiente no qual o agente está situado, sendo implementada fora de seu raciocínio; interna, que é responsável pela mudança do estado mental interno do agente, sendo implementada em sua própria arquitetura; e comunicativa, que é responsável pela comunicação entre os agentes, realizando a troca de mensagens entre eles. Uma sequência de ações é chamada de *plan* (HÜBNER; BOISSIER *et al.*, 2020). A figura 4 apresenta um diagrama do agente com os conceitos introduzidos.

Para seu funcionamento adequado, é esperado que um agente inteligente possua

autonomia, proatividade, reatividade e habilidade social. Autonomia é a capacidade de agir de forma independente na tomada de decisões, buscando as melhores ações e planos para atingir suas metas designadas de modo autônomo e garantindo que elas não necessitem de influência ou direcionamento externo. Entretanto, apesar da independência, o agente não é livre para fazer absolutamente qualquer coisa que desejar, sendo limitado por metas e planos que atribui-se a ele e por "regras" da organização ao qual ele está inserido. Proatividade é a manifestação de um comportamento orientado a cumprir objetivo, ou seja, ao receber uma meta, ele tentará atingi-la. Dessa forma, impede que o agente seja passivo e fique ocioso, sem tentar executar nenhuma ação. Reatividade é a capacidade de ser responsivo a mudanças no ambiente no qual o agente está situado, possibilitando a escolha de diferentes estratégias/abordagens ao tornar-se consciente da falha em executar um determinado plano. Então, para que seja efetivo, é necessário que o agente tenha um equilíbrio entre um comportamento de resposta direcionado/orientado a meta e reativo. Habilidade social é a capacidade de interação com outros agentes, proporcionando coordenação e cooperação. Essa comunicação permite não somente transmitir informações, como também compartilhar crenças, metas e planos entre si (HÜBNER; BORDINI; WOOLDRIDGE, 2007).

Para a lógica da atuação do agente é utilizado o modelo *BDI* (Belief-Desire-Intention). A figura 5 representa seus componentes. Essa arquitetura baseia-se na noção de estados mentais para descrever o comportamento do agente, concentrando-se no papel das intenções no raciocínio prático. Tendo sua base inspirada na teoria do raciocínio/comportamento humano, ela é capaz de prever e explicar o comportamento de um sistema complexo sem a necessidade de entender como exatamente ele funciona. Dessa forma, pode-se explicar esse modelo a partir de três conceitos: *belief*, *desire* e *intention*. *Belief*, semelhante à propriedade do agente, se trata do conhecimento pertencente a ele, retratando as informações que ele possui sobre o mundo e sendo baseadas em percepções sobre um ambiente ou em comunicações com outros agentes. *Desire* se trata de objetivos ou estados que o agente deseja alcançar, influenciando suas ações a serem realizadas. Entretanto, observa-se a possibilidade de esses desejos serem incompatíveis entre si, o que implica que o agente não é obrigado a seguir a risca todos eles, podendo não agir de acordo com alguns. Por fim, *intention* se trata dos desejos com os quais o agente se compromete a perseguir, podendo ser definidos por meio de metas delegadas a ele ou por meio de decisões deliberadas por ele. A intenção, então, leva o agente a executar planos com ações que tenham a capacidade de satisfazê-la (HÜBNER; BORDINI; WOOLDRIDGE, 2007).

Neste trabalho, será utilizada a linguagem de programação orientada a agentes Jason para implementar os agentes do Sistema Multiagente. Essa linguagem, desenvolvida por Jomi F. Hübner e Rafael H. Bordini, é um interpretador baseado em Java para uma versão estendida do AgentSpeak, implementando a semântica operacional da linguagem e uma plataforma para o desenvolvimento do Sistema Multiagente (HÜBNER; BORDINI,

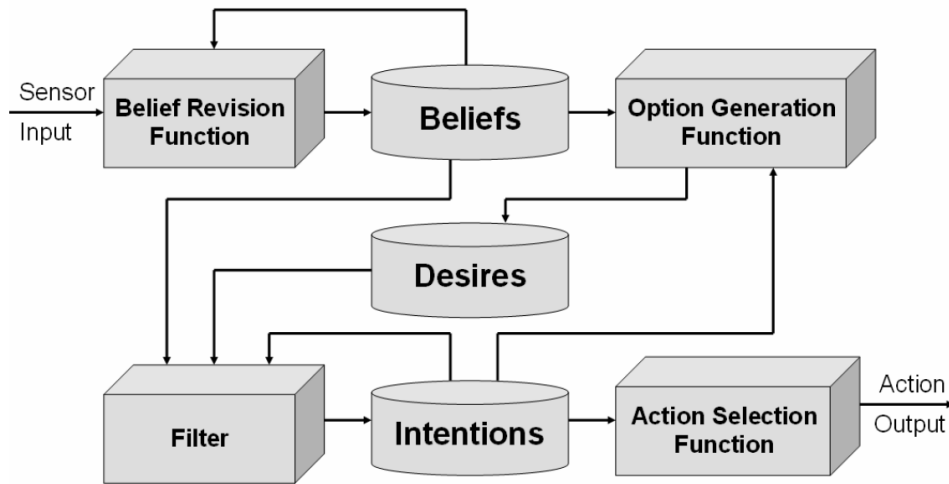


Figura 5 – Diagrama de uma arquitetura BDI.

Fonte: (NUNES, 2007)

s.d.). Os agentes implementados a partir dessa linguagem são baseados no modelo *BDI* - que foi explicado detalhadamente no parágrafo anterior -, representando seus estados de informação, motivação e deliberação (KRAUSBURG *et al.*, 2018).

2.2.2 Sistemas Multiagente

Um Sistema Multiagente (*SMA*), conforme apresentado na figura 6, é um sistema complexo modelado como um conjunto de agentes autônomos situados em um ambiente lógico compartilhado se comunicando e interagindo entre si e com o próprio ambiente com o objetivo de atingir metas em comum de forma cooperativa. Esse sistema é estruturado a partir de uma organização, que encapsula seus principais aspectos, como tarefas, funcionalidades, comportamentos, papéis, deveres e propriedades. Dessa forma, é feita a regulamentação e a coordenação das atividades para cada agente presente no sistema, sendo eles dispostos em várias relações organizacionais uns com os outros. Vale ressaltar que um ambiente é particionado entre os agentes, que possuem um escopo de atuação que permite deter controle total ou parcial sobre ele. Devido a isso, é necessário que, para alcançar o objetivo pretendido, um agente considere o comportamento e a influência dos demais em relação ao ambiente no qual está operando (HÜBNER; BORDINI; WOOLDRIDGE, 2007) (HÜBNER; BOISSIER *et al.*, 2020).

Comparado a um sistema monolítico - que possui um único agente -, um sistema multiagente pode oferecer diversos benefícios, aprimorando a precisão, adaptabilidade e escalabilidade da aplicação e aumentando a eficiência ao descentralizar o processamento e otimizar a complexidade temporal dos processos computacionais. (JUCHEM; BASTOS, 2001) e (GUTOWSKA, 2024) apontam como parte das principais características que possibilitam uma vantagem significativa em seu uso: aproveitamento do paralelismo, me-

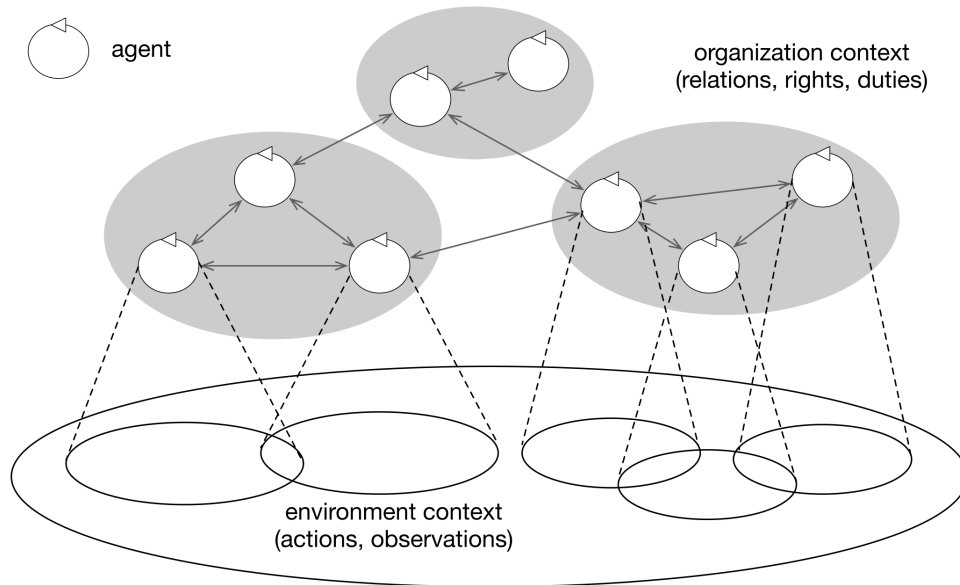


Figura 6 – Diagrama da estrutura de um Sistema Multiagente.

Fonte: (HÜBNER; BOISSIER *et al.*, 2020)

lhorando a rapidez e eficiência na resolução de problemas; agrupamento de agentes com diferentes competências e com capacidade de se ajustar e adaptar a diversos ambientes, fornecendo maior flexibilidade; cooperatividade entre agentes como meio de compartilhar uma maior quantidade de informações, possibilitando a solução de problemas mais complexos e provendo aumento de escalabilidade; especialização de domínio, permitindo que cada agente do sistema tenha aprofundamento e responsabilidade em apenas uma parte do problema; e possibilidade de agentes assumirem responsabilidades uns dos outros em caso de falhas, fortalecendo a segurança e a confiabilidade.

2.2.3 Artefato (CArtAgO)

Outra dimensão importante no contexto de um Sistem Multiagente é o *environment*. Sua principal característica é a dinamicidade, permitindo a evolução e adaptação do ambiente em função das necessidades do agente. Essa dimensão é responsável por situar os agentes dentro de um determinado cenário, fornecendo-lhes os meios para afetar o ambiente no qual se encontram, alterando determinadas propriedades, e captar estados e eventos observáveis nele, permitindo a percepção do ambiente e a atualização em suas crenças com base nas mudanças. Dessa forma, seu principal objetivo é encapsular no sistema recursos e ferramentas em que os agentes operam a fim de receber auxílio para atingir suas metas e completar seus objetivos (HÜBNER; BOISSIER *et al.*, 2020).

Os principais conceitos atribuídos a essa dimensão são *workspace* e *artifact*. A figura 7 apresenta um diagrama desses conceitos no contexto de um *environment*.

Área de trabalho (*workspace*) se trata de uma região lógica do ambientes respon-

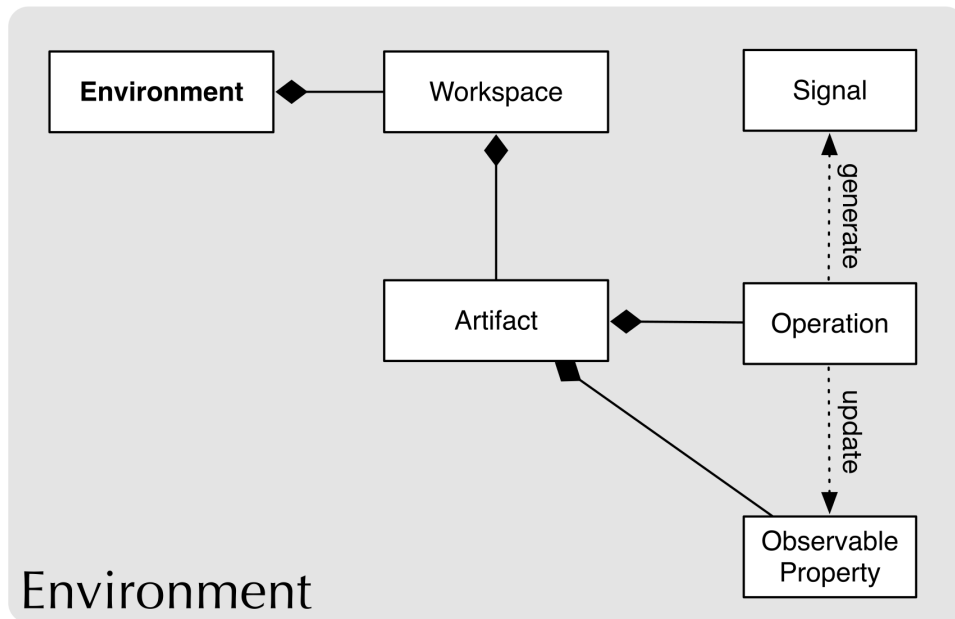


Figura 7 – Diagrama dos principais conceitos de um ambiente.

Fonte: (HÜBNER; BOISSIER *et al.*, 2020)

sável por agrupar um conjunto de artefatos e agentes, permitindo sua estruturação e organização em um ponto de vista topológico. Ela provê localidade e situabilidade aos agentes, delimitando um escopo sob o qual o agente irá atuar tanto em relação a suas interações e observações quanto em relação ao uso de artefatos (HÜBNER; BOISSIER *et al.*, 2020).

Artefato (*artifact*) se trata de um recurso do ambiente com o qual o agente interage a fim de executar ações, permitindo a alteração do estado do ambiente, ou observar determinadas propriedades. Com isso, ele permite a abstração e o encapsulamento de funcionalidades - definidas em termos de operações e propriedades - e entidades computacionais que serão utilizadas pelo agentes, possibilitando a construção de ambientes modulares e dinâmicos. Devido a essa dinamicidade, artefatos podem ser criados e destruídos em tempo de execução pelo agente, de acordo com suas metas e necessidades. Dessa forma, observa-se que sua principal importância está na possibilidade de prover ao agente a capacidade de manipulação e percepção do ambiente com o intuito de auxiliar sua racionalidade (HÜBNER; BOISSIER *et al.*, 2020).

Em síntese, um *environment* é uma representação do "mundo" sob o qual os agentes operam e estão situados, encapsulando no Sistema Multiagente serviços, recursos e ferramentas. Para isso, ele é modelado como um *workspace* no qual os agentes entram a fim de compartilhar e manusear *artifacts* com o intuito de realizar o trabalho que lhes foi designado e cumprir suas metas e objetivos. Sua principal diferença em relação à dimensão dos agentes é a falta de autonomia e de proatividade nas entidades que o compõem, considerando que elas apenas reagem a ações realizadas pelos agentes, não tendo a capacidade

de racionalizar e tomar decisões por si mesmas (HÜBNER; BOISSIER *et al.*, 2020).

Neste trabalho, será utilizado o framework CArtAgO para implementar os ambientes nos quais os agentes serão situados. Essa infraestrutura é baseada no modelo A&A (*Agents and Artifacts*) e programada em Java, sendo um ambiente de software organizado como um conjunto de artefatos, ou seja, entidades computacionais que oferecem serviços e funcionalidades ao agente (BOISSIER *et al.*, 2013). Dessa forma, um agente que foca em um determinado artefato possui percepção de suas *propriedades observáveis* - que são mantidas atualizadas com os valores mais recentes de cada passo de simulação - e é capaz de atuar sobre ele a partir de *operações* que são disponibilizadas pelo artefato (KRAUSBURG *et al.*, 2018).

2.2.4 Organização (Moise)

Finalmente, a última dimensão a ser destacada na construção de um Sistema Multiagente se trata da *organization*. Sua função é estruturar, coordenar e regulamentar agentes trabalhando em conjunto dentro de um ambiente compartilhado, auxiliando em suas decisões e interações enquanto garante um estado global coerente ao sistema. Dessa forma, garante suporte para que múltiplos agentes, dependentes entre si, trabalhem de forma coordenada e para o controle de sua autonomia dentro do sistema, a partir de regras sociais que devem ser obedecidas. Vale ressaltar que, mesmo fazendo parte de uma organização, agentes podem se coordenar entre si sem depender dela. Entretanto, caso o comportamento auto coordenado do agente não esteja de acordo com o que foi especificado na organização, é considerado uma violação. Dessa forma, como a função da organização é impor diretrizes e estabelecer normas, ela poderá aplicar uma punição ao agente transgressor, dependendo da gravidade da infração. Por fim, é importante destacar que os próprios agentes são capazes de reorganizar a organização, tanto em nível de estrutura quanto em nível de coordenação, realizando mudanças, a partir de decisões coletivas ou individuais, tanto ao alterar a atribuição de papéis, de missões ou as próprias normas quanto ao criar novos papéis em um grupo. Isso permite que eles definam e adaptem a organização enquanto agem, percebendo e cooperando sob ela (HÜBNER; BOISSIER *et al.*, 2020).

A definição das especificações e entidades dessa dimensão pode ser dividida em três abstrações: *structural*, composta por *role*, *link* e *group*; a *functional*, composta por *organizational goal*, *mission*, *social plan* e *social scheme*; e a *normative*, composta por *norm*. A figura 8 apresenta um diagrama dos principais conceitos atribuídos às abstrações no contexto de uma *organization*.

A abstração estrutural (*structural*) é responsável pela definição da estrutura geral de uma organização, sendo organizada em papel (*role*), vínculo (*link*) e grupo (*group*). Um *role* é a representação da função/responsabilidade do agente na organização. Um *link* é a determinação do tipo de interação que agentes poderão ter entre si dentro de um grupo.

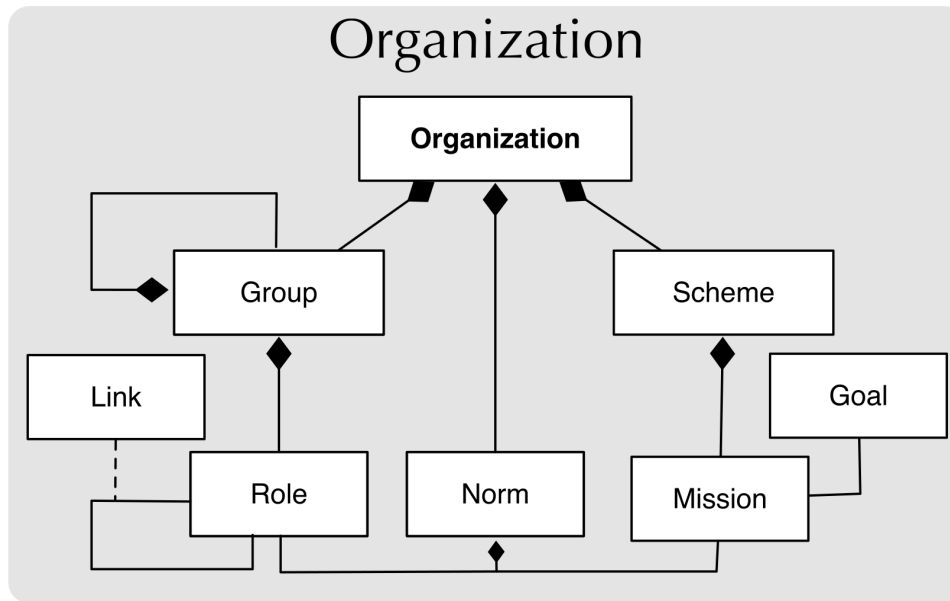


Figura 8 – Diagrama dos principais conceitos de uma organização.

Fonte: (HÜBNER; BOISSIER *et al.*, 2020)

Por fim, um *group* é um conceito que busca fornecer uma estrutura social ao sistema, possibilitando a existência de uma comunidade de agentes na organização. Observa-se que a entrada de um agente na organização se inicia ao ele adotar um papel, tornando ele parte do grupo ao qual o papel pertence e se conectando aos agentes que possuem papéis vinculados ao seu (HÜBNER; BOISSIER *et al.*, 2020).

A abstração funcional (*functional*) é responsável pela coordenação do comportamento dos agentes pertencentes a uma organização, sendo definida em termos de meta organizacional (*organizational goal*), missão (*mission*), plano social (*social plan*) e esquema social (*social scheme*). Um *organizational goal* é o estado que deve ser satisfeito por um ou vários agentes. Uma *mission* é um conjunto de metas reunidas que serão responsabilidade de um papél da organização. Nesse contexto, os agentes, ao se comprometerem a uma missão - de acordo com as normas definidas na especificação normativa, não se baseando em sua decisão individual -, devem alcançar as metas correspondentes a ela. Um *social plan* é uma estrutura de metas organizacionais que são relacionadas entre si, na qual deve haver coordenação para administrar as possíveis dependências entre elas. Por fim, um *social scheme* é a junção de todos os outros conceitos, integrando os planos sociais com suas metas organizacionais e missões correspondentes. Dessa forma, observa-se que nenhuma missão ou meta pode ser definida fora de um esquema, sendo a principal estrutura da especificação funcional. Ele é responsável, portanto, por denotar o comportamento coletivo e coordenado de um grupo de agentes. Conclui-se, então, que os conceitos apresentados são fundamentais para definir o comportamento esperado de um agente pertencente a uma organização (HÜBNER; BOISSIER *et al.*, 2020).

A abstração normativa (*normative*) cuida da regulamentação do comportamento dos agentes pertencentes a uma organização, sendo responsável por restringir a autonomia dos agentes baseando-se nas duas abstrações anteriores. Para isso, é usando o conceito de norma (*norm*), que define os direitos e deveres de um agente dentro de uma organização, definindo, para cada agente individualmente, quais são suas obrigações, permissões e restrições. Isso é feito por meio da conexão das abstrações estruturais e funcionais com *deontic modalities*, que assume o valor *obligation* ou *permission*. Dessa forma, observa-se que as normas são definidas dentro do contexto de uma organização - dependendo de sua estrutura e coordenação -, o que é a sua principal diferença em relação a outras abordagens (HÜBNER; BOISSIER *et al.*, 2020).

Vale ressaltar que, apesar de não ser obrigatório, o uso de especificações estruturais e funcionais é vantajoso em termos computacionais. Isso porque, enquanto a falta da especificação estrutural torna necessário que os agentes negociem entre si a liderança e as relações e cooperações possíveis, devido à falta de conhecimento de quem está envolvido em um determinado grupo; a falta da especificação funcional faz com que os agentes precisem raciocinar e definir planos toda vez que forem agir em conjunto, devido à possibilidade de perda dos envolvidos anteriormente pela falta de memória organizacional, e se coordenar entre si (HÜBNER; BOISSIER *et al.*, 2020).

Neste trabalho, será utilizado o framework Moise para implementar as organizações compostas por agentes. A abordagem baseada em seu uso inclui linguagem de modelagem de organização, infraestrutura de gerenciamento de organização e suporte para mecanismos de raciocínio baseado em organização a nível do próprio agente (BOISSIER *et al.*, 2013). Com isso, o Moise permite que agentes façam parte de uma organização ao adotar determinado papel e, assim, tornando-se responsáveis por metas associadas a ele. A configuração da organização é feita a partir de um arquivo XML que é constituído pelas especificações estrutural, funcional e normativa (KRAUSBURG *et al.*, 2018).

2.3 PROJETO CÉOS

O projeto CÉOS é um projeto de pesquisa em parceria entre o Departamento de Informática e Estatística (INE) da Universidade Federal de Santa Catarina (UFSC) e o Ministério Público de Santa Catarina (MPSC) com o objetivo de automatizar a análise e extração do conhecimento de grandes volumes de dados, por meio de sua coleta, integração, organização e processamento, para auxiliar e aprimorar a tomada de decisões em domínios complexos do setor público e, assim, contribuir com a prevenção e o combate à corrupção (UFSC, 2025).

Com o aumento da quantidade de dados disponível para processamento, a extração de informações de editais e documentos se torna uma tarefa complexa e demorada, tornando essencial o uso de tecnologias para administrá-la. O surgimento do projeto decorreu, portanto, da demanda de técnicas de gerenciamento e análise desses dados, possibilitando

seu tratamento e manipulação com o intuito de transformá-los em informação útil e relevante aos promotores e, assim, auxiliá-los na otimização da inspeção e análise de licitações e indicadores de possíveis fraudes licitatórias (WÜST *et al.*, 2024).

Este trabalho busca, no âmbito do projeto CÉOS, desenvolver um Sistema Multiagente - detalhado no capítulo 3 - que será integrado em um painel inteligente de licitações - uma interface de usuário contendo informações a respeito de processos licitatórios e alertas identificados associados à eles que poderão ser acessadas, sem a necessidade de consultar diferentes bases de informação, para proporcionar uma análise integrada - com o objetivo de verificar, de forma inteligente e autônoma, se há indícios de fraude nos processos licitatórios, agilizando o processo de investigação (WÜST *et al.*, 2024).

2.4 ESTADO DA ARTE

Para o fortalecimento do embasamento teórico e prático a respeito da problemática deste trabalho, foi realizada uma revisão do estado da arte, focada nas áreas de Sistema Multiagente e de Fraude Licitatória/Monetária, em repositórios como o Google Scholar e o DBLP.

Essa seção, portanto, consiste na apresentação de dois artigos encontrados que possuem relação com o tema deste trabalho, discorrendo a respeito de seu conteúdo e do diferencial da proposta apresentada pelos autores.

2.4.1 A multi-agent data mining system for cartel detection in Brazilian government procurement

Em (RALHA; SILVA, 2012), a detecção de cartéis no processo de detecção de corrupção é reconhecida como um problema complexo, devido ao grande volume de dados disponível e necessário para correlacionar as informações e as diversas estratégias distintas utilizadas para esconder as operações fraudulentas.

Devido a isso, os autores propuseram o desenvolvimento do AGMI, um *agent-mining tool* que integra as estratégias de mineração de dados distribuída (*DDM*), descoberta de conhecimento em bancos de dados (*KDD*) e sistema multiagente (*MAS*), se tratando de um sistema de mineração de dados multiagente projetado para auxiliar na detecção, prevenção e monitoramento de cartéis que atuam em processos de compras públicas.

O principal diferencial da abordagem proposta, dessa forma, se trata ao implementar um agente inteligente em conjunto com a análise tradicional de mineração de dados, possibilitando a inclusão de heurísticas baseadas no conhecimento de especialistas.

O trabalho (RALHA; SILVA, 2012), portanto, se destaca como uma solução de abordagem híbrida, demonstrando a importância da adição de conhecimento especialista na estratégia para superar limitações da mineração clássica na detecção de irregularidades, como a detecção de cartelização em compras públicas, e aumentar eficácia e desempenho.

2.4.2 A Multi-Agent System Based Approach to Fight Financial Fraud: An Application to Money Laundering

Em (ALEXANDRE; BALSÀ, 2018), é reconhecida a ineficiência do processo de investigação semi-automatizado, envolvendo regras fixas, de identificação de crimes de lavagem de dinheiro e de auxílio à tomada de decisões, devido ao crescente volume de transações e a constante publicação de novas regulamentações.

Devido a isso, os autores propuseram uma abordagem baseada em sistemas multiagente para analisar e sinalizar transações suspeitas, auxiliando o analista humano na investigação de comportamentos suspeitos. Para isso, é realizada a criação de um perfil de comportamento transacional dos clientes, utilizando tecnologias de mineração de dados para buscar um histórico de operações.

O principal diferencial da abordagem proposta, dessa forma, se trata da utilização desses dados extraídos para compor uma base de conhecimento de um sistema multiagente, implementado na plataforma JaCaMo. A arquitetura, então, é composta por agentes especializados em detectar suspeitas e por agentes que, além de sinalizar os resultados obtidos, atua como um assistente do analista humano, podendo aprender com as decisões tomadas por ele.

O trabalho (ALEXANDRE; BALSÀ, 2018), portanto, se destaca ao demonstrar a capacidade da abordagem em identificar comportamentos irregulares complexos de forma mais eficaz e eficiente, identificando, inclusive, casos de lavagem de dinheiro que não haviam sido previamente detectados por sistemas tradicionais baseados em regras estáticas.

3 CONSTRUÇÃO DA PROPOSTA

Essa seção apresenta uma visão geral da modelagem do sistema multiagente, destacando as principais escolhas arquiteturais da organização e os artefatos criados, e, em seguida, explicando detalhadamente o seu funcionamento, com destaque a trechos importantes do código desenvolvido.

3.1 MODELAGEM DO SISTEMA MULTIAGENTE

A modelagem do sistema proposto é realizada por meio de trilhas de investigação, compostas por diversas tipologias que, em conjunto, formam uma hipertipologia responsável por representar uma categoria de fraude. O sistema, então, utilizará diversos agentes, agrupados em uma organização e trabalhando de forma coordenada e cooperativa, a fim de verificar cada uma dessas tipologias e, a partir da apuração obtida, estabelecer a gravidade da fraude - caso seja encontrado algum indício -, que será exibida ao usuário por meio de um alerta. O alerta, representado neste projeto por uma tabela em uma base de dados, será responsável por exibir informações a respeito do nome da hipertipologia, do nível de gravidade, da data em que foi executada a investigação e da categoria do alerta - representada pelo nome da hipertipologia em conjunto com uma breve descrição dela -, além de uma descrição curta - representada pela listagem das tipologias que cada hipertipologia contém - e de uma descrição longa - representada por um breve texto informativo a respeito das informações a respeito da empresa/licitação analisada e do motivo de o agente ter concluído se tratar de um indicativo de fraude.

O sistema proposto neste projeto, dessa forma, é responsável por averiguar uma análise da relação entre empresas licitantes, analisando as seguintes hipertipologias: A (“empresas participantes de uma mesma licitação compartilham características”), B (“empresas participantes de uma mesma licitação combinam de uma sempre perder para outra ganhar ou alternam vitórias”), C (“empresas participantes de uma mesma licitação apresentam propostas com proporcionalidade de valor/preço”), D (“propostas idênticas entre licitantes em Unidades Gestoras diferentes”) e E (“conluio já identificado”).

Como mencionado anteriormente, cada hipertipologia é formada por um conjunto de tipologias. Nesse contexto, as tipologias que compõem cada uma das hipertipologias apresentadas no parágrafo anterior são:

- A: 1 (“empresa licitante com sócio em comum com outra empresa licitante”), 2 (“empresa licitantes com ex-sócio em comum com outra empresa licitante”), 3 (“familiares consanguíneos de primeiro grau em comum entre sócios de empresas licitantes”), 4 (“sócio de uma empresa licitante é empregado de outra empresa licitante”), 5 (“empresas licitantes com contador em comum ”), 6 (“empresas licitantes estão sediados no mesmo endereço”) e 7 (“empresas licitantes possuem mesmo telefone, e-mail ou

grupo econômico”);

- B: 8 (“empresa licitante é perdedora contumaz”), 9 (“empresa licitante é vencedora contra empresa perdedora contumaz”) e 10 (“alternância de vitórias entre empresas licitantes”);
- C: 11 (“empresa apresenta propostas simétricas acima de patamar definido”);
- D: 12 (“empresa apresenta propostas idênticas acima de patamar definido”);
- E: 13 (“empresas que já tiveram conluio comprovado”).

Vale ressaltar que, apesar do sistema estar modelado para suportar todas as hipertipologias e tipologias mencionadas nos parágrafos anteriores, não foi possível obter todos os dados necessários para a análise. Dessa forma, algumas tipologias, como a 2, a 3, a 4, a 5 e a 13, não terão a análise realizada, retornando o resultado sempre como se não tivesse sido encontrada nenhuma fraude e adicionando um log “Dados insuficientes”.

A organização do sistema multiagente será composta por:

- um agente gerente: definido no arquivo gerente.asl; responsável por gerenciar os outros agentes, criando um grupo para cada hipertipologia e definindo papéis aos agentes responsáveis por coordenar um respectivo grupo, e por buscar na base de dados os editais de licitação que serão analisados, selecionando-os e repassando-os aos agentes de hipertipologia;
- cinco agentes hipertipologia: definidos nos arquivos hipertipologiaA.asl, hipertipologiaB.asl, hipertipologiaC.asl, hipertipologiaD.asl e hipertipologiaE.asl; responsável por criar um grupo para a respectiva hipertipologia, definindo papéis aos agentes responsáveis por realizar a análise das tipologias, por repassar os editais selecionados aos agentes de tipologia, os quais são enviados de forma sequencial, um por vez, até a finalização da análise do que está com investigação em curso, e por criar o alerta da respectiva hipertipologia - ao receber o resultado de todos os agentes de tipologia pelos quais está responsável -, definindo o seu nível de gravidade - o maior valor de gravidade entre as tipologias detectadas ou o nível de gravidade da hipertipologia, caso todas tenham sido identificadas - e gerando uma breve descrição a respeito das tipologias identificadas - a partir dos dados a respeito dessas tipologias e salvos no documento JSON - para, dessa forma, notificar as principais informações a respeito da fraude identificada ao usuário;
- treze agentes tipologia: definidos nos arquivos tipologiaA.asl, tipologiaB.asl, tipologiaC.asl, tipologiaD.asl e tipologiaE.asl; responsável por realizar consultas na base de dados a respeito da tipologia da qual é encarregado, verificando se há algum indício de fraude no edital licitatório que está sendo investigado e, em caso positivo,

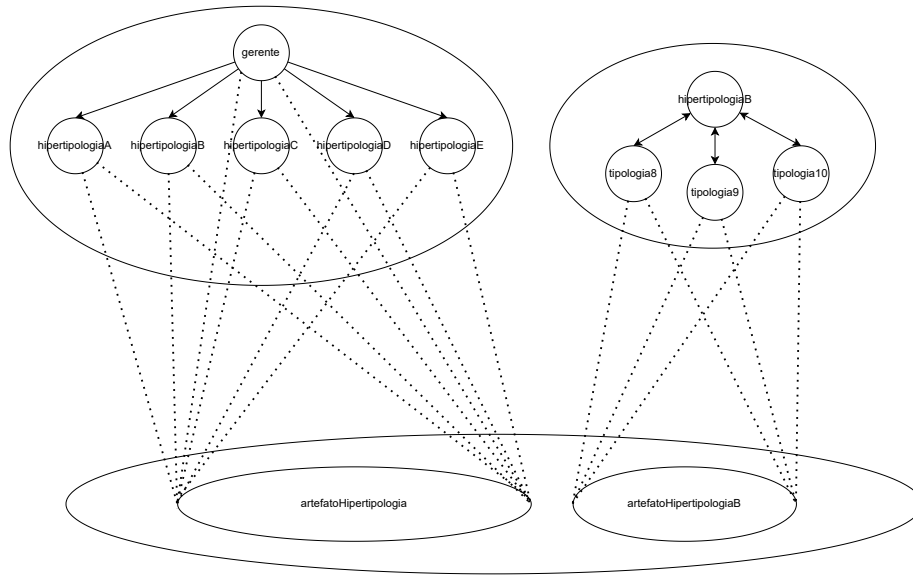


Figura 9 – Modelagem da arquitetura dos grupos da organização.

Fonte: Autor

armazenando informações relevantes a respeito da irregularidade e da empresa/licitação em um documento JSON, e retornar ao seu respectivo agente de hipertipologia o resultado da análise, sendo representado como uma lista com *false* em caso de regularidade e com informações a respeito da tipologia - número da tipologia, nome da tipologia e gravidade da tipologia - em caso de irregularidade.

Esses agentes serão divididos em dois tipos de grupos distintos. O primeiro, chamado *groupHipertipologia*, contém o agente gerente e todos os agentes hipertipologia. O segundo, chamado *groupHipertipologiaN* - onde N representa os valores A, B, C, D e E, dependendo da hipertipologia abordada no grupo -, contém o agente hipertipologia correspondente ao grupo e seus respectivos agentes tipologia. A figura 9 apresenta uma representação da arquitetura descrita em relação aos grupos *groupHipertipologia* e *groupHipertipologiaB* - considerando que os outros *groupHipertipologiaN* possuem estrutura semelhante - e aos artefatos que estão observando.

Para cada um desses grupos é definido um *scheme*, no qual são realizados *goals* de forma paralela, buscando aumentar o desempenho do sistema, considerando que não há dependência entre eles. O grupo *groupHipertipologia* possui o *schemeHipertipologia*, que é separado em duas *missions* diferentes. A primeira, denominada *missionHipertipologia*, contém o *goal createHipertipologia*, que consiste na criação e gerenciamento dos grupos de cada hipertipologia. Dessa forma, como definido na especificação normativa da organização, trata-se de uma obrigação a ser realizada pelos agentes *hipertipologia*. A outra, denominada *missionBuscaEdital*, contém o *goal goalBuscaEdital*, que consiste na busca pelos editais licitatórios na base de dados e seu posterior repasse aos agentes *hipertipologia*.

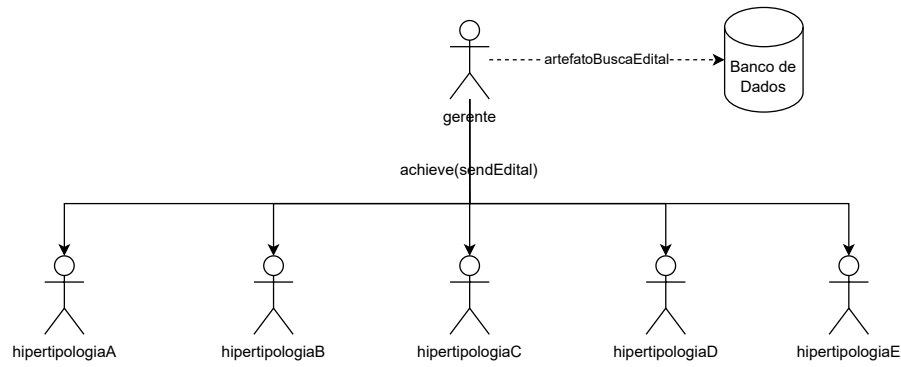


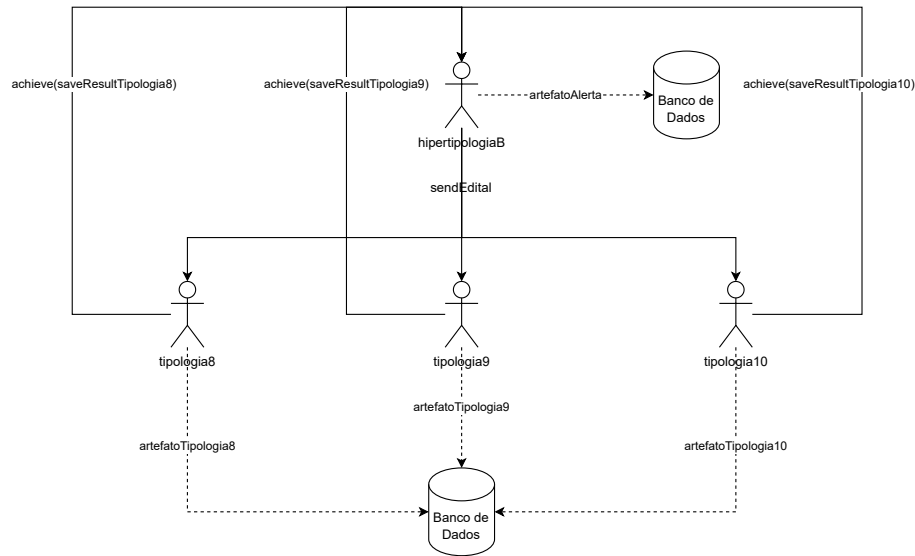
Figura 10 – Modelagem da funcionalidade do grupo *groupHipertipologia*.

Fonte: Autor

A especificação normativa, portanto, define essa *mission* como sendo uma obrigação do agente *gerente*. O funcionamento geral do grupo pode ser observado na figura 10.

Em relação ao grupo *groupHipertipologiaN*, haverá um *scheme* por hipertipologia, sendo denominados como *schemeHipertipologiaN*. Eles funcionam de forma semelhante, sendo separados em uma *mission* por tipologia presente na hipertipologia e, da mesma forma que nos *schemes* anteriores, executadas em paralelo. Cada *mission*, então, conterá um *goal* denominado de *goalTipologiaX* - onde X representa o valor numérico da tipologia que será analisada a partir desse *goal* -, que consiste em realizar a busca na base de dados a respeito da tipologia investigada, retornando o resultado da análise do indicativo de fraude. Essas *missions*, como especificado na especificação normativa, serão atribuídas, de forma obrigatória, aos agentes *tipologia*, sendo determinadas a partir do *role* da tipologia correspondente. A figura 11 apresenta o funcionamento geral do grupo *groupHipertipologiaB*, se tratando de um exemplo que representa uma execução de ações semelhante a observada no resto dos grupos *groupHipertipologiaN*.

Por fim, vale mencionar que o sistema possui seis artefatos distintos. O primeiro, chamado *ArtefatoHipertipologia*, será observado pelos agentes *gerente* e *hipertipologia*, auxiliando a busca de editais no banco de dados e a criação do alerta. Outro artefato, chamado *ArtefatoTemplate*, não será observado por nenhum agente, tendo a função apenas de auxiliar, a partir de um template pré-definido, a gerar uma descrição longa a um alerta, sendo utilizado dentro do *ArtefatoHipertipologia*. Por fim, existem os artefatos *ArtefatoHipertipologiaN* - onde, novamente, N representa os valores A, B, C, D e E -, sendo observado pelos agentes *tipologia* que pertencem ao grupo de hipertipologia correspondente e, com isso, possuindo o objetivo de auxiliar a busca e análise de indicativos de fraude na base de dados e o salvamento das informações relevantes em um documento JSON.

Figura 11 – Modelagem da funcionalidade do grupo *groupHipertipologiaB*.

Fonte: Autor

3.2 DETALHAMENTO

O sistema multiagente inicia com o agente *gerente* executando o *plan start*, o qual é responsável por criar a organização, utilizando o comando em *createWorkspace* - linha 2 do Código 3.1 -, e o artefato correspondente a ela, a partir do comando *makeArtifact* - linha 5 do Código 3.1 -, se juntando a ela em sequência com o comando *joinWorkspace* - linha 3 do Código 3.1. Em seguida, ele será responsável por criar o grupo *groupHipertipologia* e criar o artefato que será observado e utilizado pelos membros desse grupo, o *ArtefatoHipertipologia*, com a utilização dos comandos *createGroup* - linha 11 do Código 3.1 - e *makeArtifact* - linha 8 do Código 3.1 - respectivamente. Com o grupo criado, ele adotará o *role roleGerenteAgentes* - linha 14 do Código 3.1 - e, para cada agente presente em *membrosHipertipologia* - uma lista em sua base de crenças contendo os agentes que farão parte do *groupHipertipologia* -, enviará um *achieve* com o comando *send* para que eles se unam à organização - linhas 16 a 20 do Código 3.1 - e, em sequência, adotem um *role* nela - linhas 22 a 25 do Código 3.1. Por fim, ele aguarda a confirmação da formação completa do grupo - de acordo com a definição de participantes da organização - com o comando do Jason *.wait(+formationStatus(ok))* - linha 26 do Código 3.1 -, criando o *scheme* do grupo *schemeHipertipologia* em seguida - linha 28 do Código 3.1. A definição desse *plan* pode ser observada em Código 3.1.

Código 3.1 – *Plan start* do agente *gerente*.

```

1 +!start <-
2   createWorkspace("ora4mas") ;

```

```

3   joinWorkspace("ora4mas", WOrg) ;
4
5   makeArtifact(ora4mas, "ora4mas.nopl.OrgBoard", ["src/org/
      mp.xml"], OrgArtId)[wid(WOrg)] ;
6   focus(OrgArtId) ;
7
8   makeArtifact(artifact, "example.ArtefatoHipertipologia", [],
      ArtId)[wid(WOrg)] ;
9   focus(ArtId) ;
10
11  createGroup(instanceGroupHipertipologia, groupHipertipologia,
      GrArtId) ;
12  focus(GrArtId) ;
13
14  adoptRole("roleGerenteAgentes")[artifact_id(GrArtId)] ;
15  ?membrosHipertipologia(ListaMembros) ;
16  for (.member(Hipertipologia,ListaMembros))
17  {
18      .send(Hipertipologia, achieve, join_org) ;
19  }
20  .wait(1000) ;
21
22  for (.member(Hipertipologia,ListaMembros))
23  {
24      .send(Hipertipologia, achieve, adopt_role) ;
25  }
26  ?formationStatus(ok)[artifact_id(GrArtIdA)] ;
27
28  createScheme(instanceSchemeHipertipologia,
      schemeHipertipologia, SchArtId) ;
29  focus(SchArtId) ;
30  addScheme("instanceSchemeHipertipologia")[artifact_id("
      instanceGroupHipertipologia")] .
31
32  +?formationStatus(ok)[artifact_id(GrArtId)] <-
33  .wait({+formationStatus(ok)[artifact_id(GrArtId)]}) .

```

Ao finalizar esse conjunto de ações definido no *plan start*, ele irá executar a *mission* *missionBuscaEdital*, definida na especificação normativa da organização como obrigatória. Essa *mission* é composta pelo *goal* *goalBuscaEdital*, que consiste em buscar na base de

dados os editais de licitação que serão analisados. Isso é feito a partir do uso da função *buscaEdital* do artefato *ArtefatoHipertipologia* - linhas 2 a 5 do Código 3.2 -, que executa a operação interna *consultaSQL* para realizar uma consulta SQL ao banco de dados - linhas 12 e 13 do Código 3.2 - e armazenar o resultado como um array de string em uma propriedade observável do artefato, denominada *resultEdital* - linhas 17 a 23 do Código 3.2. Essa propriedade, inicializada no artefato com o valor *false*, assume um valor no formato `[id_procedimento_licitatório1, id_procedimento_licitatório2, ..., id_procedimento_licitatórioN]`, sendo N a quantidade de editais encontradas na base de dados. O funcionamento do artefato pode ser observado no trecho em Código 3.2.

Código 3.2 – Funções *buscaEdital* e *consultaSQL* do artefato *ArtefatoHipertipologia*.

```

1  @OPERATION
2  void buscaEdital()
3  {
4      execInternalOp("consultaSQL", "SELECT
          id_procedimento_licitatorio FROM public.processo_licitatorio
          ;", "resultEdital");
5  }
6
7  @INTERNAL_OPERATION
8  void consultaSQL(String sql, String propriedade)
9  {
10     try (Connection conexao = dataSource.getConnection())
11     {
12         PreparedStatement statementConsulta = conexao.
            prepareStatement(sql);
13         ResultSet resultSetConsulta = statementConsulta.executeQuery
            ();
14
15         ArrayList<String> infoTipologia = new ArrayList<>();
16
17         while (resultSetConsulta.next())
18         {
19             infoTipologia.add(resultSetConsulta.getString(1));
20         }
21
22         String[] resultInfoTipologia = infoTipologia.toArray(new
            String[0]);
23         getObsProperty(propriedade).updateValue(resultInfoTipologia)

```

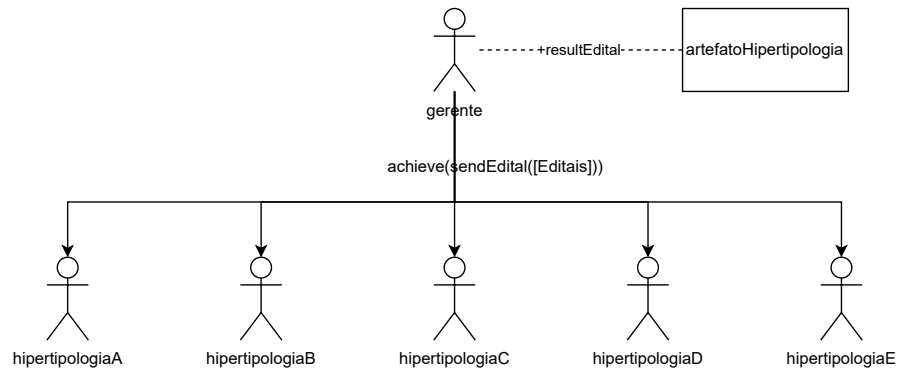


Figura 12 – Fluxo de interação dos agentes em *resultEdital* do agente *gerente*.

Fonte: Autor

```

24     }
25     catch (SQLException e)
26     {
27         e.printStackTrace();
28     }
29 }

```

O agente *gerente* então, ao observar uma mudança na propriedade do artefato que está observando (*resultEdital*), irá buscar a lista de membros do *groupHipertipologia* - armazenada em sua base de crenças como *membrosHipertipologia* (*[hipertipologiaA, hipertipologiaB, hipertipologiaC, hipertipologiaD, hipertipologiaE]*) - e enviar a todos o array de editais licitatórios selecionados, a partir de um *achieve* para que eles executem o *plan sendEdital* - linhas 3 a 6 do Código 3.3. Seu funcionamento pode ser observado a partir da definição do trecho em Código 3.3 e do fluxo de interação apresentado na figura 12.

Código 3.3 – Reação à mudança da propriedade *resultEdital* do agente *gerente*.

```

1 +resultEdital(ResultadoConsulta)
2   : ResultadoConsulta \== false
3   <- ?membrosHipertipologia(ListaMembros) ;
4     for (.member(Hipertipologia,ListaMembros)) {
5         .send(Hipertipologia,achieve,sendEdital(ResultadoConsulta
6             )) ;

```

Os agentes *hipertipologia*, por outro lado, inicializam e aguardam o agente *gerente* informar que é para eles executarem os *plan join_org* e *adopt_role*. Eles então, ao executá-

los, se unem à organização, a partir do comando *joinWorkspace* - linha 2 do Código 3.4 - e começam a focar, utilizando o comando *focus*, nos artefatos da organização e do grupo ao qual pertencem - linhas 3 a 7 do Código 3.4 -, adotando o papel correspondente à letra da hipertipologia ao qual pertencem em sequência - linha 10 do Código 3.4. Essa funcionalidade pode ser observada no trecho em Código 3.4, usando o agente *hipertipologiaB* como exemplo.

Código 3.4 – Plan de junção à organização e adoção de papel do agente *hipertipologiaB*.

```

1  +!join_org <-
2    joinWorkspace(ora4mas) ;
3    lookupArtifact(instanceGroupHipertipologia, GrArtId) ;
4    focus(GrArtId) ;
5
6    lookupArtifact(artifact, ArtId) ;
7    focus(ArtId) .
8
9  +!adopt_role
10  <- adoptRole(roleHipertipologiaB)[artifact_id(GrArtId)] .

```

Esses agentes, ao entrarem no grupo, irão executar a *mission missionHipertipologia*, definida na especificação normativa da organização como obrigatória. Essa *mission* é composta pelo *goal createHipertipologia*, o qual funciona de forma semelhante ao *goal start* do agente *gerente*, entretanto criando um *workspace* para o grupo específico da respectiva tipologia ao invés da organização e os grupos, artefatos e *schemes* correspondentes a esse grupo. Além disso, vale ressaltar que, ao invés de usar uma lista de membros na base de crenças para definir os agentes que serão membros do grupo, é utilizado uma concatenação de *tipologiaX* - sendo X a letra correspondente a do agente *hipertipologia* - com um valor N - tendo N variando de 1 até a quantidade de tipologias presente na hipertipologia em questão -, como pode ser observado nas linhas 5 a 7 do Código 3.5.

Em seguida, ao receber um *achieve* do agente *gerente*, eles irão executar o *plan sendEdital*, que consiste em selecionar o primeiro edital do array - por meio da variável *Edital* na linha 4 do Código 3.5 - e enviar para todas as tipologias pelas quais são responsáveis - linhas 5 a 9 do Código 3.5. A obtenção das tipologias presentes nesse grupo, como explicado no parágrafo anterior, é feita a partir de uma concatenação, formando o nome de todos os agentes *tipologia* pertencentes à hipertipologia. Após realizar o envio, com o comando *send*, os agentes *hipertipologia* atualizam sua crença *editalAtual* para o número do edital licitatório que foi enviado - linha 10 do Código 3.5 - e sua crença *editalRestante* para o resto do array de editais - linha 11 do Código 3.5. A definição do *plan sendEdital* pode ser observada no trecho em Código 3.5 e a partir do fluxo de interação da figura 13.

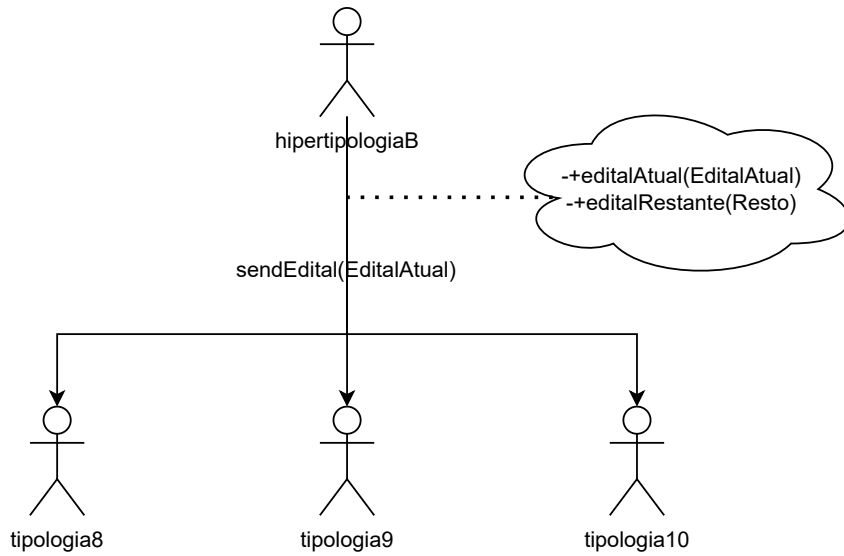


Figura 13 – Fluxo de interação dos agentes em *sendEdital* do agente *hipertipologiaB*.

Fonte: Autor

Código 3.5 – *Plan sendEdital* do agente *hipertipologiaB* para envio de editais.

```

1  +!sendEdital ([]) .
2
3  +!sendEdital ([ Edital | Resto ])
4    <- ?numTipologias(T) ;
5      for (.range(I,1,T)) {
6        .concat(tipologiaB,I,Agente) ;
7        .send(Agente, tell, num_edital(Edital)) ;
8      }
9      -+editaAtual(Edital) ;
10     -+editaRestante(Resto) .

```

Os agentes *hipertipologia*, então, passam a aguardar o resultado da análise dos agentes *tipologia* pelos quais são responsáveis. Isso é feito a partir do valor de *hipertipologia* em sua base crenças, definida inicialmente como *hipertipologia([null,null,...,null])*, sendo um array que tem N posições - N representando a quantidade de tipologias que cada hipertipologia possui - definidas como null. Seu funcionamento, então, baseia-se em aguardar os agentes *tipologia* enviar o resultado da análise, que é feito a partir do envio de um *achieve* de um *plan* denominado *saveResultTipologiaN*, no qual N é o número da tipologia analisada. Esse *plan* consiste em alterar o valor da respectiva posição no array da crença *hipertipologia*. O trecho em Código 3.6 apresenta um exemplo do funcionamento para o agente *hipertipologiaB*.

Código 3.6 – *Plan saveResultTipologia* do agente *hipertipologiaB* para salvar resultado obtido da tipologia.

```

1 +!saveResultTipologia8 (ResultadoConsulta)
2   : hipertipologia ([_,B,C])
3   <- +!hipertipologia ([ResultadoConsulta,B,C]) .
4
5 +!saveResultTipologia9 (ResultadoConsulta)
6   : hipertipologia ([A,_,C])
7   <- +!hipertipologia ([A,ResultadoConsulta,C]) .
8
9 +!saveResultTipologia10 (ResultadoConsulta)
10  : hipertipologia ([A,B,_])
11  <- +!hipertipologia ([A,B,ResultadoConsulta]) .

```

Vale ressaltar que sempre que houver uma alteração ao valor da crença *hipertipologia* os agentes *hipertipologia* irão reagir a isso, realizando uma sequência de ações apenas ao receber todos os resultados de análise das tipologias - linha 2 do Código 3.8 -, o que é verificado a partir da regra *analiseFinalizada* ao checar se todos os valores do array é diferente de null - linha 5 do Código 3.7. Com isso, seu próximo passo é realizar algumas verificações para definir que “caminho” seguir. A primeira é se houve algum indício de fraude ou não - linha 2 do Código 3.8 -, realizado por meio da regra *hipertipologiaSemFraude*, que analisa as informações recebidas dos agentes *tipologia*. Como mencionado na seção anterior, caso haja indicativo de fraude, será recebida uma lista com as informações a respeito da tipologia e, caso não, será recebida uma lista com o valor *false*. Dessa forma, a regra consiste em verificar se o primeiro valor da lista de todas as informações recebidas são *false*, indicando que não foi detectada nenhuma fraude em caso positivo - linha 2 do Código 3.7. Nessa situação, os agentes *hipertipologia* não irão criar nenhum alerta, apenas voltar a crença *hipertipologia* ao valor inicial, buscar quais são os editais restantes e, com eles, realizar novamente o *plan sendEdital* - linhas 3 a 5 do Código 3.8. Os trechos em Código 3.7 e 3.8 apresentam, respectivamente, a definição das regras *analiseFinalizada* e *hipertipologiaSemFraude* e o funcionamento da situação de não haver indicativo de fraude, ambos representando um exemplo do agente *hipertipologiaB*.

Código 3.7 – Definição de regras *hipertipologiaSemFraude* e *analiseFinalizada* do agente *hipertipologiaB*.

```

1 hipertipologiaSemFraude ([H1,H2,H3]) :-
2   .nth(0,H1,false) & .nth(0,H2,false) & .nth(0,H3,false) .
3
4 analiseFinalizada ([H1,H2,H3]) :-
5   H1 \== null & H2 \== null & H3 \== null .

```

Código 3.8 – Ações do agente *hipertipologiaB* ao identificar hipertipologia sem fraude.

```

1 +hypertipologia (Tipologias)
2   : analiseFinalizada (Tipologias) & hipertipologiaSemFraude (
      Tipologias)
3   <- —+hypertipologia ([null,null,null]) ;
4     ?editaRestante (Resto) ;
5     !sendEdita (Resto) .

```

A segunda situação se trata de quando há indicativo de fraude, porém somente de uma parte da hipertipologia, ou seja, apenas algumas tipologias detectaram indícios de fraude, enquanto outras estão regulares. Para checar isso, é utilizada a regra *hipertipologiaIncompleta* que, de forma semelhante a regra *hipertipologiaSemFraude*, verifica o primeiro valor da lista de todas as informações recebida, analisando se algum deles é igual a *false*, indicando que pelo menos um agente *tipologia* não detectou nenhuma fraude em caso positivo - linha 2 do Código 3.9. Nessa situação, será criado um alerta. Para isso, entretanto, será necessário primeiro definir qual será o nível de gravidade dele. Os agentes *hipertipologia*, então, utilizam a regra *maiorGravidade* para definir qual tipologia possui o maior valor, o qual, como definido na seção anterior, será o utilizado para definir a gravidade do alerta como um todo. Essa regra funciona de forma a, utilizando outra regra *getGravidade* - que consiste em definir a gravidade a partir de uma lista de informações de tipologia recebida, resultando em 0 caso a lista seja *[false]* e no valor da terceira posição desta lista caso contrário, levando em conta que seu formato será sempre *[NumeroTipologia, NomeTipologia, GravidadeTipologia]*, como apresentado nas linhas 4 e 5 do Código 3.9 -, buscar o nível de gravidade de cada tipologia e uní-los em um array, utilizando o comando *.max* para encontrar o maior entre eles - linhas 8 e 9 do Código 3.9.

Ao obter esse valor, eles buscam qual é o edital que está sendo analisado no momento, a partir da crença *editaAtual*, e as informações a respeito da hipertipologia, por meio da crença *info_hiptipologia*, a qual é definida na base de crenças no formato *info_hiptipologia(LetraHiptipologia, NomeHiptipologia, GravidadeHiptipologia)* - linhas 3 e 5 do Código 3.10. Por fim, usam a função *artefatoAlerta* do artefato *textitArtefatoHiptipologia* para construir o alerta. Essa função consiste em, inicialmente, verificar os resultados de fraude que foram passados aos agentes *hipertipologia* - linhas 8 a 12 do Código 3.11 -, buscando o documento JSON com as informações da fraude a partir do caminho definido com o número do edital investigado e do número da tipologia analisada - linha 15 do Código 3.11 -, caso o resultado não seja *false*. Ao identificar o documento, o conteúdo é lido e convertido em um *JSONArray* - linhas 19 e 20 do Código 3.11 -, que será usado para a construção da descrição longa do alerta. Essa descrição é gerada a partir de um template, que possui uma formatação pré-definida de acordo com a hipertipologia e com a tipologia, utilizando as informações obtidas do JSON para completar as frases gera-

das - linhas 22 e 23 do Código 3.11. É, então, gerada uma dessas sentenças pré-definidas por resultado encontrado, gerando um único texto concatenando todas no final - linhas 7 a 11, 14 a 17 ou 21 a 24 do Código 3.12, a depender da tipologia. Com isso, é utilizada a operação interna *gerarAlerta*, que consiste em definir um SQL para adicionar o alerta criado na base de dados - linhas 56 e 57 do Código 3.11 -, a partir das informações obtidas, e, caso a categoria do alerta não tenha sido registrada previamente na base de dados, um SQL para adicioná-la - linhas 42 a 44 do Código 3.11.

Por fim, são realizadas as mesmas ações descritas no parágrafo anterior, voltando a crença *hipertipologia* ao valor inicial, buscando quais são os editais restantes e, com eles, realizando novamente o *plan textitsendEdital* - linhas 7 a 9 do Código 3.10. São apresentados, primeiramente, a definição das regras *hipertipologiaIncompleta*, *getGravidade* e *maiorGravidade* em Código 3.9, seguida do funcionamento da situação de haver indicativo de fraude com a hipertipologia incompleta em Código 3.10 e, por fim, das funções *artefatoAlerta* e *gerarTemplate* em Código 3.11 e 3.12 respectivamente, todos representando um exemplo do agente *hipertipologiaB*. Para complementar, a figura 14 demonstra o fluxo de funcionamento da geração de alertas.

Código 3.9 – Definição de regras *hipertipologiaIncompleta*, *getGravidade* e *maiorGravidade* do agente *hipertipologiaB*.

```

1 hypertipologiaIncompleta ([ H1,H2,H3 ]) :-
2   .nth(0,H1,false) | .nth(0,H2,false) | .nth(0,H3,false) .
3
4 getGravidade ([ false ], 0) .
5 getGravidade ([ _,_,Gravidade ], Gravidade) .
6
7 maiorGravidade ([ H1,H2,H3 ], ResultadoGravidade) :-
8   getGravidade(H1, G1) & getGravidade(H2, G2) & getGravidade(H3,
9     G3) &
   .max([ G1,G2,G3 ], ResultadoGravidade) .

```

Código 3.10 – Ações do agente *hipertipologiaB* ao identificar hipertipologia incompleta.

```

1 +hypertipologia (Tipologias)
2   : analiseFinalizada(Tipologias) & hypertipologiaIncompleta(
3     Tipologias)
4   <- ?info_hypertipologia(LetraHipertipologia, Hipertipologia, _
5     ) ;
6     ?maiorGravidade(Tipologias, Gravidade) ;
7     ?editalAtual(Edital) ;
8     artefatoAlerta(Edital,Tipologias,[ LetraHipertipologia,
9       Hipertipologia, Gravidade]) ;

```

```

7      —+hypertipologia ([ null,null,null]) ;
8      ?editalRestante(Resto) ;
9      !sendEdital(Resto) .

```

Código 3.11 – Funções *artefatoAlerta* e *gerarAlerta* do artefato *artefatoHiptipologia*.

```

1  @OPERATION
2  void artefatoAlerta(String numEdital, Object[] hipertipologia,
3      Object[] grupoHiptipologia)
4  {
5      String path_pasta = "saida_json/edital_"+numEdital+"_Tipologia
6          ";
7      String descricao_curta = "As tipologias encontradas nessa
8          analise sao: ";
9      String descricao_longa = "Durante a analise do edital "+
10          numEdital+", foram encontrados os seguintes indicativos de
11          fraude:\n";
12
13      for (Object resultado : hipertipologia)
14      {
15          Object[] info_tipologia = (Object[]) resultado;
16          String fraude = info_tipologia[0].toString();
17          if (fraude != "false")
18          {
19              descricao_curta += info_tipologia[1].toString() + ", ";
20              String path_json = path_pasta+fraude+".json";
21
22              try
23              {
24                  String contentJSON = new String(Files.readAllBytes(Paths
25                      .get(path_json)));
26                  JSONArray infoFraude = new JSONArray(contentJSON);
27
28                  descricao_longa += "Tipologia "+fraude+" – "+infoFraude.
29                      getJSONObject(0).getString("tipologia")+"\n";
30                  descricao_longa += templates.get(grupoHiptipologia[0].
31                      toString()).gerarTemplate(fraude, infoFraude.
32                      getJSONArray(2))+"\n";
33              }
34              catch (IOException e)

```

```

26     {
27         e.printStackTrace();
28     }
29 }
30 }
31
32 descricao_curta = descricao_curta.substring(0,descricao_curta.
    length()-2);
33 descricao_longa = descricao_longa.substring(0,descricao_longa.
    length()-1);
34 execInternalOp("gerarAlerta", grupoHipertipologia[1].toString
    (), grupoHipertipologia[2].toString(),descricao_curta,
    descricao_longa);
35 }
36
37 @INTERNAL_OPERATION
38 void gerarAlerta(String nomeHipertipologia, String
    gravidadeAlerta, String descricaoCurta, String descricaoLonga
    )
39 {
40     try (Connection conexao = dataSource.getConnection())
41     {
42         String categoria_alerta = "INSERT INTO alertas.
            categoria_alertas (nome) " +
43             "VALUES ('"+nomeHipertipologia+"') " +
44             "ON CONFLICT (nome) DO NOTHING;";
45         PreparedStatement statementConsulta = conexao.
            prepareStatement(categoria_alerta);
46         int result = statementConsulta.executeUpdate();
47
48         String idCategoria = "";
49         statementConsulta = conexao.prepareStatement("SELECT
            id_categoria FROM alertas.categoria_alertas WHERE nome =
            '"+nomeHipertipologia+"'");
50         ResultSet resultConsulta = statementConsulta.executeQuery();
51         if (resultConsulta.next())
52         {
53             idCategoria = resultConsulta.getString(1);
54         }

```

```

55
56     String alerta = "INSERT INTO alertas.alertas (nome, nivel ,
        data_ultima_execucao , id_categoria , descricao_curta ,
        descricao_longa) " +
57         "VALUES ( 'Alerta "+nomeHipertipologia+" ', "+
            gravidadeAlerta+" , CURRENT_TIMESTAMP, "+
            idCategoria+" , ' "descricaoCurta+ " ', ' "+
            descricaoLonga+" ' ) ";
58     statementConsulta = conexao.prepareStatement(alerta);
59     result = statementConsulta.executeUpdate();
60 }
61 catch (SQLException e)
62 {
63     e.printStackTrace();
64 }
65
66 }

```

Código 3.12 – Template pré-defino para tipologias da hipertipologia B.

```

1  public String gerarTemplate(String tipologia , JSONArray
    infoFraude)
2  {
3      String retornoInfo = "";
4      switch (tipologia)
5      {
6          case "8":
7              for (int i=0; i<infoFraude.length(); i++)
8              {
9                  retornoInfo += String.format("A empresa %s (CNPJ: %s)
                    possui uma taxa de %s%% de derrotas em relacao as
                    licitacoes em que participou.\n", infoFraude.
                    getJSONObject(i).getString("nome_participante"),
                    infoFraude.getJSONObject(i).getString("
                    cpf_cnpj_participante_cotacao"),infoFraude.
                    getJSONObject(i).getString("percentual_perdas"));
10             }
11             break;
12
13     case "9":

```

```

14     for (int i=0; i<infoFraude.length(); i++)
15     {
16         retornoInfo += String.format("A empresa %s (CNPJ: %s)
            possui um historico de vencer as licitacoes em que
            participou contra a empresa perdedora contumaz %s (
            CNPJ: %s).\n", infoFraude.getJSONObject(i).getString(
            "nome_vencedora"),infoFraude.getJSONObject(i).
            getString("cnpj_vencedora"),infoFraude.getJSONObject(
            i).getString("nome_perdedora"),infoFraude.
            getJSONObject(i).getString("cnpj_perdedora"));
17     }
18     break;
19
20     case "10":
21         for (int i=0; i<infoFraude.length(); i++)
22         {
23             retornoInfo += String.format("Ha constatacao de uma
                possivel alternancia de vitorias entre a empresa %s (
                CNPJ: %s) e a empresa %s (CNPJ: %s).\n", infoFraude.
                getJSONObject(i).getString("nome_empresa_1"),
                infoFraude.getJSONObject(i).getString("cnpj_empresa_1
                "),infoFraude.getJSONObject(i).getString("
                nome_empresa_2"),infoFraude.getJSONObject(i).
                getString("cnpj_empresa_2"));
24         }
25         break;
26
27     default:
28         break;
29 }
30
31 return retornoInfo;
32 }

```

A última situação se trata de quando foi detectado indício de fraude para todas as tipologias presentes na hipertipologia, sendo caracterizada como uma "fraude completa". Essa situação irá funcionar de forma semelhante a anterior, buscando as informações da hipertipologia - linha 3 do Código 3.13 - e, a partir dela, criando um alerta - linha 5 do Código 3.13. Entretanto, ao invés de calcular o valor da gravidade, será utilizado o nível padrão da hipertipologia, definido em *info_hipertipologia*. O Código 3.13 apresenta

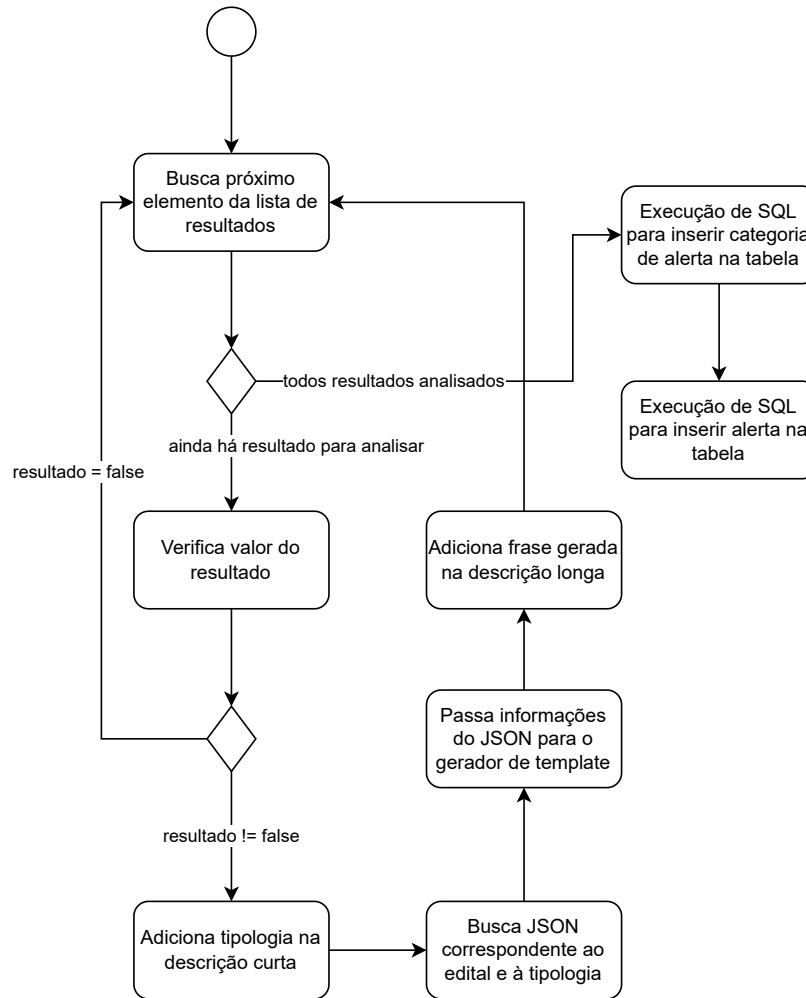


Figura 14 – Fluxo de geração de alertas do artefato *ArtefatoHipertipologia*.

Fonte: Autor

as ações do agente para lidar com a situação de "fraude completa", representando um exemplo baseado no agente *hipertipologiaB*.

Código 3.13 – Ações do agente *hipertipologiaB* ao identificar hipertipologia completa.

```

1 +hypertipologia ( Tipologias )
2   : analiseFinalizada ( Tipologias )
3   <- ?info_hypertipologia ( LetraHipertipologia, Hipertipologia,
4     Gravidade ) ;
5     ?editalAtual ( Edital ) ;
6     artefatoAlerta ( Edital,Tipologias,[ LetraHipertipologia,
      Hipertipologia, Gravidade] ) ;
  +hypertipologia ([ null,null,null] ) ;

```

```

7      ?edita1Restante(Resto) ;
8      !sendEdita1(Resto) .

```

Os agentes *tipologia*, por fim, inicializam e aguardam o agente *hipertipologia* correspondente informar que é para eles executarem os *plan join_org* e *adopt_role*. O *join_org* ocorre de forma semelhante ao dos agentes *hipertipologia*, se unindo à organização e focando nos respectivos artefatos da organização e do grupo. O *adopt_role*, por outro lado, consiste em, inicialmente, verificar se determinado *role* do grupo ao qual pertence já foi adotado por outro agente. Isso é realizado a partir da regra *adopted*, que consiste em contar quantos agentes assumiram um determinado *role*, a partir dos comandos *.count* e *play*, e somente adotá-lo caso nenhum outro agente tenha assumido ele e caso o próprio agente não tenha nenhum outro papel, checado a partir da crença *role*, que é inicializada em 0 e alterada para 1 ao ser definido um papel - linhas 1, 9 e 17 do Código 3.14. Ao definir um *role*, além de adotá-lo, o agente armazena o nome do respectivo *role*, como *agent_role*, e as informações da tipologia correspondente, no formato *info_tipologia(NumeroTipologia, NomeTipologia, GravidadeTipologia)*, na base de crença - linhas 6 e 7, 14 e 15 e 22 e 23 do Código 3.14, dependendo da tipologia. O funcionamento do *adopt_role* pode ser observado no trecho em Código 3.14, utilizando o agente *tipologiaB* de exemplo.

Código 3.14 – *Plan adopt_role* dos agentes *tipologia* da hipertipologia B.

```

1  +!adopt_role : adopted(roleTipologia8,V) & V == 0 & role(X) & X
    == 0
2  <-
3  -role(X) ;
4  +role(1) ;
5  adoptRole(roleTipologia8)[artifact_id(GrArtId)] ;
6  +agent_role(roleTipologia8) ;
7  +info_tipologia(8, "empresa licitante e perdedora contumaz",
    2) .
8
9  +!adopt_role : adopted(roleTipologia9,V) & V == 0 & role(X) & X
    == 0
10 <-
11 -role(X) ;
12 +role(1) ;
13 adoptRole(roleTipologia9)[artifact_id(GrArtId)] ;
14 +agent_role(roleTipologia9) ;
15 +info_tipologia(9, "empresa licitante e vencedora contra
    empresa perdedora contumaz", 2) .

```

```

16
17 +!adopt_role : adopted(roleTipologia10,V) & V == 0 & role(X) & X
    == 0
18 <-
19 -role(X) ;
20 +role(1) ;
21 adoptRole(roleTipologia10)[artifact_id(GrArtId)] ;
22 +agent_role(roleTipologia10) ;
23 +info_tipologia(10, "alternancia de vitorias entre empresas
    licitantes", 3) .

```

Esses agentes, ao entrarem no grupo, irão executar a *mission missionTipologiaN* - sendo N o valor numérico que representa o número da tipologia pela qual o agente é responsável -, definida na especificação normativa da organização como obrigatória. Essa *mission* é composta pelo *goal goalTipologiaN* - referindo-se ao mesmo N definido em *missionTipologiaN* -, que é responsável por, a partir do artefato *ArtefatoHipertipologiaX* - sendo X a letra da hipertipologia que contém a tipologia que está sendo analisada pelo agente *tipologia* -, realizar a consulta da tipologia sobre o edital licitatório que está sendo investigado. Para isso, os agentes, inicialmente, buscam as informações a respeito da tipologia - obtidas a partir da crença *info_tipologia*, como definido na linha 2 do Código 3.15 - e esperam o agente *hipertipologia* responsável repassar o edital que será verificado, com o uso do comando *.wait* - linha 3 do Código 3.15.

Ao receber o edital, a função *consultaTipologiaN* - novamente com N referindo-se ao mesmo valor definido anteriormente - do artefato, que realizará uma consulta SQL no banco de dados a partir da operação interna *consultaSQL* - linhas 27 e 28 do Código 3.16 -, armazenando o resultado em uma propriedade observável *resultTipologiaN* como uma matriz, ou seja, array de arrays, de tuplas, em que cada array dentro do “array principal” equivale a um resultado - representado por uma linha na tabela - da consulta, e os elementos desses arrays exibem as informações - representadas pelas colunas da tabela - obtidas a partir da consulta, sendo armazenados no formato *nomeAtributo(valorAtributo)* - linhas 33 a 45 do Código 3.16. Além disso, a função é responsável por, caso tenha pelo menos um valor de retorno, salvar os dados em um arquivo JSON a partir da operação interna *salvaJSON* - linhas 47 a 51 do Código 3.16 -, que consiste em salvar as informações obtidas da análise em um documento JSON com o nome salvo no formato *edital_NumeroEdital_TipologiaN.json* - sendo NumeroEdital o id do edital que está sendo investigado e N o número da tipologia analisada -, de acordo com as linhas 2 a 47 do Código 3.17. Por fim, ele irá indicar que o edital terminou de ser analisado, removendo *num_edital* de sua base de crenças e recomeçará o *goal*, esperando pelo próximo edital - linhas 5 e 6 do Código 3.15.

Ao mesmo tempo, os agentes irão reagir quando observarem uma mudança de valor

na propriedade observável *resultTipologiaN*, verificando se o retorno da análise é uma lista vazia - linha 9 do Código 3.15 -, o que indica que não foi identificado nenhum indicativo de fraude. Em caso afirmativo, será enviado ao agente *hipertipologia* responsável um *achieve saveResultTipologiaN - plan* responsável por guardar o resultado da investigação - com o valor `[false]`, indicando que nenhum indício foi detectado - linha 10 do Código 3.15. Em caso negativo, por outro lado, serão novamente buscadas as informações a respeito da tipologia, a partir da crença *info_tipologia*, sendo enviado, em seguida, ao agente *hipertipologia* responsável um *achieve saveResultTipologiaN* com os dados da tipologia no formato `[NumeroTipologia, NomeTipologia, GravidadeTipologia]`, indicando que foi detectado indicativo de fraude - linhas 14 e 15 do Código 3.15. Os Códigos 3.15 e 3.16 representam o funcionamento da análise de fraude de uma tipologia e do uso da função *consultaTipologia* respectivamente, utilizando a tipologia 8 como exemplo. Além disso, novamente tendo a tipologia 8 como referência, é apresentado o fluxo de retorno do resultado da consulta por meio da figura 15. Por fim, o Código 3.17 apresenta a operação interna *salvaJSON*.

Código 3.15 – Funcionalidade de análise de uma tipologia pelo agente *tipologia*.

```

1  +!goalTipologia8
2    <- ?info_tipologia(NumTipologia, Tipologia, _) ;
3    .wait(num_edital(NumEdital)) ;
4    consultaTipologia8(NumEdital, [NumTipologia, Tipologia]) ;
5    -num_edital(NumEdital)[source(hipertipologiaB)] ;
6    !goalTipologia8 .
7
8  +resultTipologia8(ResultadoConsulta)
9    : agent_role(roleTipologia8) & ResultadoConsulta = []
10   <- .send(hipertipologiaB, achieve, saveResultTipologia8([false
11     ])) .
12
13  +resultTipologia8(ResultadoConsulta)
14    : agent_role(roleTipologia8)
15    <- .wait(json8,1000,T);
16    -json8;
17    ?info_tipologia(NumTipologia, Tipologia, Gravidade) ;
18    .send(hipertipologiaB, achieve, saveResultTipologia8([
19      NumTipologia,Tipologia,Gravidade])) .

```

Código 3.16 – Funções *consultaTipologia8* e *consultaSQL* do artefato *ArtefatoHipertipologiaB*.

```

1  @OPERATION

```

```

2  void consultaTipologia8(String numEdital, Object[] tipologia)
3  {
4      execInternalOp("consultaSQL",
5          "SELECT nome_participante,
6              cpf_cnpj_participante_cotacao, percentual_perdas
7              "FROM (" +
8                  "SELECT id_participantelicitacaocotacao AS
9                      empresa,ROUND((SUM(CASE WHEN \"vencedor?\" =
10                          0 THEN 1 ELSE 0 END) * 100.0) / COUNT(*), 2)
11                      AS percentual_perdas " +
12                      "FROM cotacao " +
13                      "WHERE \"vencedor?\" IN (0, 1) " +
14                      "GROUP BY id_participantelicitacaocotacao " +
15                      "HAVING (SUM(CASE WHEN \"vencedor?\" = 0 THEN 1
16                          ELSE 0 END) * 100.0) / COUNT(*) > 75 AND
17                          COUNT(*) >= 7 " +
18                          "ORDER BY percentual_perdas DESC) AS
19                          empresa_perdedora " +
20                      "INNER JOIN (" +
21                          "SELECT id_participante_licitacao,
22                              nome_participante,
23                              cpf_cnpj_participante_cotacao,
24                              id_procedimento_licitatorio " +
25                          "FROM participante_licitacao " +
26                          "WHERE id_procedimento_licitatorio = " + numEdital
27                          + ") AS empresa_cotacao " +
28                          "ON empresa = id_participante_licitacao; ",
29                          numEdital,
30                          tipologia);
31  }
32
33  @INTERNAL_OPERATION
34  void consultaSQL(String sql, String numEdital, Object[]
35      tipologia)
36  {
37      try (Connection conexao = dataSource.getConnection())
38      {
39          PreparedStatement statementConsulta = conexao.

```

```
        prepareStatement(sql);
28     ResultSet resultSetConsulta = statementConsulta.executeQuery
        ();
29     ResultSetMetaData metaData = resultSetConsulta.getMetaData()
        ;
30
31     ArrayList<Tuple[]> infoTipologia = new ArrayList<>();
32
33     while (resultSetConsulta.next())
34     {
35         Tuple[] currentInfo = new Tuple[metaData.getColumnCount()
            ];
36         for (int i=0; i<metaData.getColumnCount(); i++)
37         {
38             currentInfo[i] = new Tuple(metaData.getColumnName(i+1),
                resultSetConsulta.getString(i+1));
39         }
40
41         infoTipologia.add(currentInfo);
42     }
43
44     Tuple[][] resultInfoTipologia = infoTipologia.toArray(new
        Tuple[0][]);
45     getObsProperty("resultTipologia"+tipologia[0]).updateValue(
        resultInfoTipologia);
46
47     if (resultInfoTipologia.length != 0)
48     {
49         String[] strTipologia = {tipologia[0].toString(),tipologia
            [1].toString()};
50         execInternalOp("salvaJSON",numEdital,strTipologia);
51     }
52 }
53 catch (SQLException e)
54 {
55     e.printStackTrace();
56 }
57 }
```

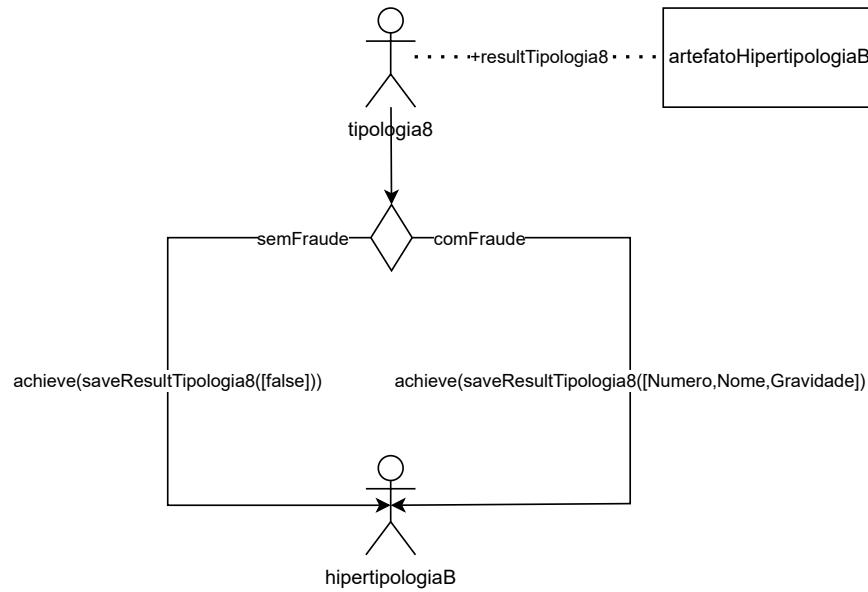


Figura 15 – Fluxo de interação dos agentes no retorno de resultado do agente *tipologia8*.

Fonte: Autor

Código 3.17 – Função *salvaJSON* do artefato *ArtefatoHipertipologiaB*.

```

1  @INTERNAL_OPERATION
2  void salvaJSON(String numEdital, String [] tipologia)
3  {
4      JSONArray valuesConsulta = new JSONArray();
5      JSONArray infosConsulta = new JSONArray();
6
7      JSONObject nomeTipologia = new JSONObject();
8      JSONObject edital = new JSONObject();
9
10     nomeTipologia.put("tipologia",tipologia[1]);
11     edital.put("edital",numEdital);
12     valuesConsulta.put(nomeTipologia);
13     valuesConsulta.put(edital);
14
15     Object resultTipologia = getObsProperty("resultTipologia"+
        tipologia[0]).getValue();
16     if (resultTipologia instanceof Tuple [][] )
17     {
18         Tuple [][] infoTipologia = (Tuple [][])resultTipologia;
19         for (Tuple [] resultConsulta : infoTipologia)

```

```
20     {
21         JSONObject consulta = new JSONObject();
22         for (Tuple infoConsulta : resultConsulta)
23         {
24             consulta.put(infoConsulta.getLabel(), infoConsulta.
                stringContent(0));
25         }
26         infosConsulta.put(consulta);
27     }
28     valuesConsulta.put(infosConsulta);
29 }
30 String path_pasta = "saida_json/";
31 String path_json = "edita_ "+numEdita+"_Tipologia "+tipologia
    [0]+ ".json ";
32
33 File pasta = new File(path_pasta);
34 if (!pasta.exists())
35 {
36     pasta.mkdirs();
37 }
38
39 try (FileWriter file = new FileWriter(path_pasta+path_json))
40 {
41     file.write(valuesConsulta.toString(4));
42 }
43 catch (IOException e)
44 {
45     e.printStackTrace();
46 }
47
48 signal("json "+tipologia[0]);
49 }
```

3.3 ESTUDO DE CASO

Devido aos dados utilizados na análise serem sigilosos, estando protegidos por um contrato de confidencialidade, optou-se pela criação de um estudo de caso com dados fictícios para demonstrar o funcionamento do trabalho. Para isso, foram projetadas tabelas em PostgreSQL com uma arquitetura semelhante a utilizada no projeto. Colunas que tra-

tam de informações não relevantes para este trabalho, entretanto, não foram preenchidas, possuindo o valor padrão *null*. Dessa forma, as tabelas e atributos utilizados no escopo de análise dessa implementação são descritos como:

- processo_licitatorio: tabela contendo os principais dados a respeito dos procedimentos licitatórios.
 - id_procedimento_licitatorio: índice do procedimento licitatório;
 - numero_processo_licitatorio: número do procedimento licitatório.
- item_licitacao: tabela contendo os itens que fazem parte de um procedimento licitatório.
 - id_item_licitacao: índice do item;
 - numero_sequencial_item: número do item dentro do seu respectivo procedimento licitatório;
 - id_procedimento_licitatorio: índice do procedimento licitatório do qual o item a ser cotado faz parte.
- participante_licitacao: tabela contendo os participantes dos procedimentos licitatórios.
 - id_participante_licitacao: índice do participante;
 - nome_participante: nome do participante;
 - id_procedimento_licitatorio: índice do procedimento licitatório do qual a pessoa física/jurídica participou;
 - id_tipo_pessoa: indicativo de participante ser uma pessoa física (1) ou jurídica (2);
 - cpf_cnpj_participante_cotacao: cpf (em caso de pessoa física) ou cnpj (em caso de pessoa jurídica) do participante.
- cotacao: tabela contendo as cotações dos participante nos itens de uma licitação.
 - id_cotacao: índice de uma cotação;
 - qtd_item_cotado: quantidade de itens cotados pelo participante;
 - valor_total_cotado_item: valor total de cotação que um participante realizou para uma unidade do item;
 - vencedor?: indicativo de vitória (1) ou derrota (0) do participante na cotação correspondente;
 - id_participantelicitacaocotacao: índice do procedimento licitatório no qual a cotação foi realizada;

- id_itemlicitacao: índice do item do procedimento licitatório sobre o qual a cotação foi realizada.
- estabelecimentos: tabela contendo as principais informações a respeito de um estabelecimento comercial.
 - cnpj: cnpj do estabelecimento;
 - nome_fantasia: nome do estabelecimento;
 - tipo_logradouro: tipo do logradouro no qual o estabelecimento está localizado;
 - logradouro: logradouro do endereço do estabelecimento;
 - numero: número do endereço do estabelecimento;
 - complemento: complemento do endereço do estabelecimento, caso necessário;
 - bairro: bairro do endereço do estabelecimento;
 - cep: CEP do endereço do estabelecimento;
 - uf: estado onde está localizado o estabelecimento;
 - municipio: município onde está localizado o estabelecimento;
 - ddd_1: DDD do número de telefone do estabelecimento;
 - telefone_1: número de telefone do estabelecimento;
 - email: e-mail do estabelecimento;
- socios: tabela contendo as principais informações a respeito dos sócios de um estabelecimento.
 - cnpj: cnpj do estabelecimento do qual o sócio faz parte;
 - nome_socio: nome do sócio;
 - cpf_cnpj_socio: CPF do sócio, havendo os primeiros três e os últimos dois dígitos censurados na forma de '*'.
- alertas: tabela responsável por armazenar os alertas gerados pelo sistema.
 - id_alerta: índice do alerta gerado;
 - nome: nome do alerta gerado, representado pelo nome da hipertipologia;
 - nivel: nível de gravidade do alerta gerado, variando entre 1 e 3;
 - data_ultima_execucao: data de execução do sistema, na qual foi gerado o alerta;
 - id_categoria: índice de uma categoria de um alerta;
 - descricao_curta: descrição curta a respeito do alerta gerado, tratando-se as tipologias detectadas;

- descricao_longa: descricao longa a respeito do alerta gerado, tratando-se de um texto com as principais informações a respeito do procedimento licitatório/empresa e dos indicativos de fraude identificados.
- categoria_alertas: tabela contendo as principais categorias de fraude para um alerta.
 - id_categoria: índice de uma categoria de alerta;
 - nome: nome da categoria de fraude do alerta, representada pelo nome de uma hipertipologia;
 - descrição: descrição a respeito do que se trata a categoria, apresentando as tipologias que contidas nela.

Para validar o sistema desenvolvido, as tabelas criadas foram populadas com diversos dados diferentes, sendo adicionados quinze processos licitatórios, havendo sessenta e nove itens de licitação distribuídos entre eles. Em relação aos participantes, a tabela armazena um total de cinquenta e dois participações, resultando em duzentos e sessenta e uma cotações para cobrir todos os itens e licitações. Vale ressaltar que, entre essas participações, há um total de vinte estabelecimentos, possuindo sessenta e seis sócios distribuídos entre eles. Os dados utilizados foram elaborados pelo autor, com a inclusão intencional de cenários fraudulentos, com o propósito de validar o funcionamento das consultas SQL utilizadas na detecção dessas irregularidades.

Entre esses dados, foram pré-definidas as seguintes fraudes:

- Licitação 1: a Empresa 5 é uma vencedora contra a empresa perdedora contumaz Empresa 11; a Empresa 11 é perdedora contumaz; a Empresa 4 e a Empresa 5 compartilham o endereço (porém não identificável no alerta).
- Licitação 2: não há fraude.
- Licitação 3: a Empresa 5 é uma vencedora contra as empresas perdedora contumaz Empresa 10 e Empresa 11; a Empresa 10 e a Empresa 11 são perdedoras contumaz; a Empresa 5 e a Empresa 10 compartilham um sócio em comum; a Empresa 10 e a Empresa 11 compartilham mesmo endereço, mesmo número de telefone e mesmo e-mail.
- Licitação 4: a Empresa 1 e a Empresa 2 compartilham um sócio em comum; a Empresa 4 e a Empresa 5 compartilham mesmo endereço (porém não identificável no alerta).
- Licitação 5: não há fraude.
- Licitação 6: não há fraude.

- Licitação 7: a Empresa 12 e a Empresa 13 compartilham mesmo endereço e mesmo e-mail; a Empresa 12 e a Empresa 13 apresentam propostas idênticas.
- Licitação 8: a Empresa 1 e a Empresa 2 compartilham um sócio em comum; a Empresa 1 e a Empresa 3 compartilham um sócio em comum; há uma alternância de vitória entre a Empresa 1, a Empresa 2 e a Empresa 3.
- Licitação 9: a Empresa 7 e a Empresa 15 apresentam propostas simétricas.
- Licitação 10: a Empresa 5 é uma vencedora contra a empresa perdedora contumaz Empresa 11; a Empresa 11 é perdedora contumaz.
- Licitação 11: não há fraude.
- Licitação 12: a Empresa 6 e a Empresa 7 compartilham um sócio em comum; a Empresa 6 e a Empresa 15 compartilham um sócio em comum; a Empresa 6, a Empresa 7 e a Empresa 15 apresentam propostas simétricas.
- Licitação 13: a Empresa 1 e a Empresa 2 compartilham um sócio em comum; a Empresa 1 e a Empresa 3 compartilham um sócio em comum; há uma alternância de vitória entre a Empresa 1, a Empresa 2 e a Empresa 3.
- Licitação 14: a Empresa 1 e a Empresa 2 compartilham um sócio em comum; a Empresa 1 e a Empresa 3 compartilham um sócio em comum; há uma alternância de vitória entre a Empresa 1, a Empresa 2 e a Empresa 3.
- Licitação 15: não há fraude.

Entretanto, espera-se a obtenção de mais alertas. Isso porque, nesse estágio de desenvolvimento, as análises são simplistas, baseadas em apenas consultar uma base de dados com SQL e exibir o resultado, sem haver muito raciocínio sobre o dado retornado. Um exemplo é a hipertipologia C, que consiste em disparar um alerta caso as propostas possuam menos de 3% de diferença de valor. Entretanto, por ser uma disputa competitiva e por os valores de cotação do conjunto de dados serem baixos, há alta probabilidade de eles serem bem semelhantes, sem necessariamente significar fraude. Além disso, pelo mesmo motivo, espera-se a falha de algumas análises, as quais foram propositalmente adicionadas no conjunto de dados para exibir o comportamento. Um exemplo disso é o endereço em que variações de nome como ‘Doutor’ e ‘Dr’ para uma rua não são considerados, sendo tratados como dados diferentes e, portanto, não identificando o indicativo de fraude.

4 RESULTADOS

Ao executar o sistema, é possível observar os seguintes resultados, baseados nas fraudes pré-definidas na seção anterior:

- Licitação 1:
 - Tipologia 6: não identificada, como previsto, devido ao formato distinto do dado do endereço.
 - Tipologia 8: não identificada.
 - Tipologia 9: não identificada.
 - Tipologia 11: como previsto na seção anterior, corretamente identificado, porém não se tratando de uma fraude intencional.
- Licitação 2:
 - Tipologia 11: como previsto na seção anterior, corretamente identificado, porém não se tratando de uma fraude intencional.
- Licitação 3:
 - Tipologia 1: identificada corretamente.
 - Tipologia 6: identificada corretamente.
 - Tipologia 7: identificada corretamente.
 - Tipologia 8: não identificada.
 - Tipologia 9: não identificada.
 - Tipologia 11: como previsto na seção anterior, corretamente identificado, porém não se tratando de uma fraude intencional.
- Licitação 4:
 - Tipologia 1: identificada corretamente.
 - Tipologia 6: não identificada, como previsto, devido ao formato distinto do dado do endereço.
 - Tipologia 8: identificada de forma incorreta ou falso-positivo.
 - Tipologia 9: identificada de forma incorreta ou falso-positivo.
 - Tipologia 10: identificada de forma incorreta ou falso-positivo.
 - Tipologia 11: como previsto na seção anterior, corretamente identificado, porém não se tratando de uma fraude intencional.
- Licitação 5:

- Tipologia 11: como previsto na seção anterior, corretamente identificado, porém não se tratando de uma fraude intencional.
- Licitação 6:
 - Tipologia 11: como previsto na seção anterior, corretamente identificado, porém não se tratando de uma fraude intencional.
- Licitação 7:
 - Tipologia 6: identificada corretamente.
 - Tipologia 7: identificada corretamente.
 - Tipologia 10: identificada de forma incorreta ou falso-positivo.
 - Tipologia 11: como previsto na seção anterior, corretamente identificado, porém não se tratando de uma fraude intencional.
 - Tipologia 12: não identificada.
- Licitação 8:
 - Tipologia 1: identificada corretamente.
 - Tipologia 10: identificada corretamente.
 - Tipologia 11: como previsto na seção anterior, corretamente identificado, porém não se tratando de uma fraude intencional.
- Licitação 9:
 - Tipologia 8: identificada de forma incorreta ou falso-positivo.
 - Tipologia 9: identificada de forma incorreta ou falso-positivo.
 - Tipologia 10: identificada de forma incorreta ou falso-positivo.
 - Tipologia 11: identificada corretamente.
- Licitação 10:
 - Tipologia 8: não identificada.
 - Tipologia 9: não identificada.
 - Tipologia 11: como previsto na seção anterior, corretamente identificado, porém não se tratando de uma fraude intencional.
- Licitação 11:
 - Tipologia 11: como previsto na seção anterior, corretamente identificado, porém não se tratando de uma fraude intencional.

dentro de uma licitação específica, sem considerar a recorrência da situação. Portanto, basta realizar um ajuste no SQL "embutido" nos agentes para alterar a consulta realizada e, dessa forma, obter o resultado desejado.

Além disso, vale ressaltar que, como mencionado na seção anterior, há falha na detecção da tipologia 6 nos casos em que endereços idênticos são representados em formatos distintos. Isso porque, neste trabalho, a análise de endereço é feita apenas comparando se dois valores retornados são iguais, não prevendo nem tratando variações para representar um mesmo nome. Assim como há detecção da tipologia 11 em basicamente todos os casos apresentados, indicando falsos-positivos. Isso ocorre porque, como mencionado anteriormente, foram utilizados valores de propostas semelhantes nos dados do estudo de caso, com o intuito de simular a competitividade. Dessa forma, apesar de os alertas serem gerados corretamente - pois a diferença dos valores é, de fato, menor do que 3% -, os casos identificados não seriam situações em que estaria ocorrendo uma "fraude real".

5 CONCLUSÃO E TRABALHOS FUTUROS

Este trabalho se propôs a avaliar a viabilidade de um sistema multiagente na detecção de indicativos de fraudes em uma licitação. O objetivo principal era o desenvolvimento de uma versão inicial de um sistema multiagente capaz de, a partir de consultas a uma base de dados com as informações necessárias, identificar se houve algum indicativo de fraude - baseado nas hipertipologias investigadas neste escopo -, gerando um alerta informativo ao usuário.

Para atingir esse objetivo foi, inicialmente, realizado um estudo teórico a respeito do tema, tanto de sistemas multiagente e de JaCaMo - usado para implementar a aplicação - quanto de como funciona o processo de investigação de fraudes em uma licitação. Com esse conhecimento, foi possível implementar uma arquitetura completa, com organização bem definida, capaz de, de maneira organizada, realizar uma consulta ao banco de dados e retornar o resultado da análise de fraude - de acordo com cada tipologia - ao seu agente “responsável”, que irá receber esses dados e lidar com o gerenciamento deles, criando os respectivos alertas. Dessa forma, foi possível proporcionar uma boa divisão de tarefas, fazendo com que cada agente ficasse responsável por uma função específica e, assim, focasse em realizar seu objetivo de forma eficiente.

Este trabalho, portanto, demonstrou que a abordagem de sistemas multiagente se trata de uma abordagem viável e promissora no processo de automatização de investigação de fraudes em procedimentos licitatórios, sendo um grande auxílio na tomada de decisão e na redução da carga de dados a serem analisados. Dessa forma, a principal contribuição deste trabalho foi a implementação de uma ferramenta de auxílio para promotores capaz de automatizar a busca por dados, o que pode ser demorado e trabalhoso, e contribuir na investigação e na tomada de decisões de situações sutis que podem passar despercebidas para um humano. Seu funcionamento foi validado a partir do estudo de caso apresentado, que demonstrou o desempenho e a eficácia do sistema.

Apesar disso, este trabalho possui algumas limitações. Primeiramente, há bastante tipologia que não foi possível obter dados para ser analisada, tornando a investigação menos completa do que o planejado. Outra situação se trata de, como mencionado nas seções anteriores, o agente possuir um raciocínio simplificado, racionalizando sobre o dado obtido sem definir uma sequência de investigações a serem feitas antes de definir uma determinada situação como potencial fraude, o que acaba gerando diversos falsos-positivos e diminuindo sua eficácia. Por fim, há a dificuldade de lidar com dados que possam estar em diferentes formatos - como o exemplo do endereço mencionado na seção 3.3 - ou incompletos - como a censura de dados sensíveis, por exemplo o CPF.

A partir das conclusões e limitações identificadas, sugere-se, primeiramente, a expansão do sistema, adicionando mais hipertipologias ou categorias de fraudes para serem analisadas pelos agentes, o que permite um aumento do escopo de irregularidades identi-

cáveis, e implementando um *trigger* no banco de dados, responsável por informar ao agente sempre que um novo edital licitatório for adicionado ao sistema e, assim, automatizando a investigação sobre ele. Além disso, sugere-se oferecer mais robustez ao sistema, implementando uma linha de raciocínio lógico ao agente. Isso permitiria que os agentes, ao invés de somente exibir as informações encontradas na consulta à base de dados, racionalizassem sobre elas, seguindo uma “trilha de investigação” ao simular o “passo a passo” de como um promotor pensa para definir se determinada situação se trata de uma fraude. Ademais, para aprimorar a análise de similaridade entre dados em formatos distintos, sugere-se a implementação de métodos de cálculo da distância de Hamming ou de Levenshtein, possibilitando a identificação de divergências ortográficas, variações de escrita e inconsistências. Por fim, sugere-se a generalização e refatoração de alguns trechos de código, como os dos agentes *hipertipologia*, para evitar repetições desnecessárias.

REFERÊNCIAS

ALEXANDRE, Claudio; Balsa, João. A Multi-Agent System Based Approach to Fight Financial Fraud: An Application to Money Laundering. **Preprints**, 2018.

BOISSIER, Olivier *et al.* Multi-agent oriented programming with JaCaMo. **Science of Computer Programming**, 2013.

BRASIL. **Lei nº 14.133, de 1º de Abril de 2021**. [S.l.: s.n.], 2021. Disponível em: https://www.planalto.gov.br/ccivil_03/_ato2019-2022/2021/lei/l14133.htm.

FUNDAÇÃO ESCOLA NACIONAL DE ADMINISTRAÇÃO PÚBLICA. **Conceitos, Princípios e Boas Práticas da Licitação Pública Aplicadas à SFTI**. Brasília, 2014.

GUTOWSKA, Anna. **What is a multiagent system?** [S.l.: s.n.], 2024. Disponível em: <https://www.ibm.com/think/topics/multiagent-system>.

HÜBNER, Jomi F.; BOISSIER, Olivier *et al.* **Multi-Agent Oriented Programming: Programming Multi-Agent Systems Using Jacamo**. [S.l.]: MIT Press, 2020.

HÜBNER, Jomi F.; BORDINI, Rafael H. **Jason, a BDI agent programming language**. [S.l.: s.n.]. Disponível em: <https://jason-lang.github.io/>.

HÜBNER, Jomi F.; BORDINI, Rafael H.; WOOLDRIDGE, Michael. **Programming Multi-Agent Systems in Agentspeak Using Jason**. [S.l.]: Wiley-Interscience, 2007.

JUCHEM, Murilo; BASTOS, Ricardo Melo. Engenharia de Sistemas Multiagentes: Uma Investigação sobre o Estado da Arte. **Faculdade de Informática - PUCRS**, 2001.

KRAUSBURG, Tabajara *et al.* SMART-JaCaMo: an Organisation-Based Team for the Multi-Agent Programming Contest. **The Multi-Agent Programming Contest**, 2018.

MASCENA, Emanuel. Licitação: Conceito e Finalidade. **Jusbrasil**, 2017.

NUNES, Ingrid Oliveira de. Implementação do Modelo e da Arquitetura BDI. **Departamento de Informática - PUC-Rio**, 2007.

ORGANISATION FOR ECONOMIC COOPERATION AND DEVELOPMENT. **Preventing Corruption in Public Procurement**. [S.l.], 2016.

PORTAL DA TRANSPARÊNCIA. **Licitações e Contratações**. [S.l.: s.n.]. Disponível em: <https://portaldatransparencia.gov.br/entenda-a-gestao-publica/licitacoes-e-contratacoes>.

RALHA, Célia Ghedini; SILVA, Carlos Vinícius Sarmento. A multi-agent data mining system for cartel detection in Brazilian government procurement. **Expert Systems with Applications**, 2012.

RODRIGUES, Neila Carla Silva; FILHO, Raimundo Nonato Lima. Modalidades Licitatórias e o Risco de Ocorrência de Fraudes nos Municípios Baianos Fiscalizados pela Controladoria Geral da União. **CCCCS Contribuciones a Las Ciencias Sociales**, 2017.

SANTA CATARINA, Universidade Federal de. **CÉOS - Inteligência Artificial em benefício da sociedade**. [S.l.: s.n.], 2025. Disponível em: <https://ceos.ufsc.br/>.

SILVA, João Victor Souza da; OLIVEIRA, Rilker Dutra de. Fraude em licitações: uma perspectiva da lei nº 14.133/2021. **Revista Científica COGNITIONIS**, 2024.

TRANSPARÊNCIA BRASIL. **Métodos de Detecção de Fraude e Corrupção em Contratações Públicas**. [S.l.], 2024.

TRIBUNAL DE CONTAS DA UNIÃO. **Licitações e Contratos - Orientações e Jurisprudência do TCU**. [S.l.], 2024.

WÜST, Andre *et al.* **Céos: Inteligência de Dados para a Sociedade: Auxílio à Tomada de Decisão Inteligente em Domínios Complexos do Setor Público**. Florianópolis, 2024.

Apêndices

APÊNDICE A – ARTIGO

Sistema Multiagente para Verificação de Indício de Fraude em Licitações

Mariana Mazzo Heitor¹, Jerusa Marchi¹

¹Departamento de Informática e Estatística – Universidade Federal de Santa Catarina (UFSC)
Florianópolis – SC – Brasil

mariana.mazzo.heitor@grad.ufsc.br, jerusa.marchi@ufsc.br

Abstract. *When there is a demand for contracting works, services or purchases by a public agency, the Brazilian Public Administration must necessarily open a bidding process. Although this is a procedure that aims to ensure fair competition among bidders, it is subject to corruption, and fraud can occur during any of its stages. As a result, prosecutors are tasked with investigating possible fraud, and there are several methodologies that seek to combat and prevent it. However, this verification can become complex due to the extensive amount of data to be analyzed. To this end, Multi-Agent System technology seeks, through a set of intelligent agents, to automate this process by simulating the behavior of prosecutors in the investigation of the bidding process, perceiving information from the bidding document and exchanging information among themselves. In this context, this work aims to present a multi-agent system capable of identifying evidence of irregularities in a group of hyper-typologies - concept representing a set of checks for evidence of fraud within a category of irregularity - based on the main categories of bidding fraud. The multi-agent system developed contains several intelligent agents, grouped into an organization, divided into a group by hyper-typology. Thus, each group is responsible for searching for signs of fraud present in the hyper-typology for which it is responsible, returning an alert if detected. The developed system was validated based on a case study prepared using fictitious data, demonstrating, despite some flaws and limitations, a satisfactory result and, thus, proving to be a viable and promising alternative in the process of automating fraud investigation in bidding procedures.*

Resumo. *Ao haver demanda de contratação de obras, serviços ou compras por parte do órgão público, a Administração Pública deve necessariamente abrir um processo licitatório. Apesar de se tratar de um procedimento que visa garantir uma competição justa entre os licitantes, ele está sujeito à corrupção, podendo ocorrer fraude durante qualquer uma de suas etapas. Como consequência disso, é atribuída aos promotores a função de exercer a apuração de possíveis fraudes, havendo diversas metodologias que buscam garantir seu combate e sua prevenção. Entretanto, essa verificação pode se tornar complexa com a extensa quantidade de dados a serem analisados. Para isso, a tecnologia de Sistema Multiagente busca, por meio de um conjunto de agentes inteligentes, automatizar esse processo ao simular o comportamento dos promotores na investigação da licitação, percebendo informações do documento e trocando informações entre si. Nesse contexto, este trabalho pretende apresentar*

um sistema multiagente capaz de identificar indícios de irregularidades de um grupo de hipertipologias - conceito que representa um conjunto de verificações de indício de fraude dentro de uma categoria de irregularidade - baseado nas principais categorias de fraude licitatória. O sistema multiagente desenvolvido contém diversos agentes inteligentes, agrupados em uma organização, divididos em um grupo por hipertipologia. Dessa forma, cada grupo é responsável por buscar os indicativos de fraude presentes na hipertipologia pela qual é responsável, retornando um alerta caso seja detectado. O sistema desenvolvido foi validado a partir de um estudo de caso elaborado por meio de dados fictícios, demonstrando, apesar de algumas falhas e limitações, um resultado satisfatório e, com isso, revelando-se uma alternativa viável e promissora no processo de automatização da investigação de fraudes em procedimentos de licitação.

1. Introdução

De acordo com as definições apresentadas por [tcu 2024] e [da Silva and de Oliveira 2024], licitação é o processo administrativo realizado pela Administração Pública com o objetivo de, baseado em condições estabelecidas previamente em um edital, selecionar a proposta mais vantajosa para a contratação e aquisição de bens e serviços, enquanto garante o caráter competitivo, preserva igualdade de condições entre os licitantes e promove o interesse público de forma eficiente.

Corrupção é um fenômeno social, político e econômico que consiste em práticas de suborno e propina, fraude, apropriação indébita ou qualquer outro desvio de recursos por parte de um agente público. Ela é derivada do infringimento de lei por parte do corrupto e do corruptor, trazendo riscos ao desenvolvimento do Estado e da própria sociedade [Rodrigues and Filho 2017]. De acordo com a OECD [oec 2016], as compras públicas são uma das atividades governamentais mais vulneráveis à corrupção, sendo apontado pelo OECD Foreign Bribery Report que mais da metade dos casos ocorre em licitações e quase dois terços dos casos ocorre em setores profundamente associados a elas.

Nesse contexto, tem-se a fraude licitatória como uma vertente da corrupção que compromete a transparência, eficiência, qualidade e economia na utilização dos recursos públicos ao realizar acordo prévio entre licitantes e agentes públicos ou manipular editais licitatórios visando beneficiar, financeiramente e administrativamente, empresas ou indivíduos [da Silva and de Oliveira 2024]. Com isso, nota-se sua responsabilidade em impactar negativamente na eficiência de aplicação e integridade dos recursos públicos, subvertendo a finalidade dos processos licitatórios e, com isso, gerando prejuízos econômicos e sociais à população. Além disso, compromete a transparência e equidade nas contratações públicas, intensificando a desconfiança institucional e governamental, e a qualidade dos serviços públicos. Para lidar com isso, a Lei nº 14.133/2021 estabelece sanções, prevendo penalidades severas de acordo com a gravidade do ato e o prejuízo causado, e medidas preventivas, como a exigência de transparência em todas as etapas do processo licitatório [da Silva and de Oliveira 2024].

Observa-se, portanto, dois principais objetivos da fraude em licitação: o comprometimento da competitividade, que é essencial em um processo licitatório, a fim de garantir controle do resultado do certame; e a apropriação de recursos públicos, por meio da contratação com superfaturamento, visando um benefício financeiro. Vale ressaltar

que, apesar de comumente as fraudes estarem associadas à apropriação de recursos, pode ocorrer o interesse em ingressar no mercado de obras públicas, não visando o lucro por meio do superfaturamento. Nessa situação, devido à exigência de documentação comprovando qualificação e experiência prévia, a área de atuação da fraude costuma ser a simulação ou falsificação desses documentos [det 2024].

Entretanto, mesmo com conhecimento acerca dos principais tipos de fraude e de suas características, o processo de análise para identificá-la não é tão simples. Devido à grande quantidade de dados que podem ser extraídos de editais, documentos e participantes, o tratamento dessa base de conhecimento em busca de informação útil pode ser um trabalho complexo para ser realizado pelos promotores, podendo entrar em questões técnicas que necessitem de especialistas. Não só isso, como a busca manual por demandas específicas torna esse processo de análise demorado, tornando essa atividade mais propensa a não ser examinada de forma eficiente e, devido a isso, pode-se deixar passar detalhes específicos do processo licitatório que podem indicar fraude, principalmente aqueles que são difíceis de serem identificados individualmente. Além das dificuldades técnicas mencionadas, essa investigação demanda uma boa quantidade de tempo do promotor na busca completa de dados para avaliar a validade de uma denúncia, podendo haver desperdício desse tempo ao se tratar de investigações improdutivas ou de repetições de verificações que já foram feitas [Wüst et al. 2024].

Portanto, torna-se interessante a implementação de um sistema multiagente - devido a possibilidade de adicionar conhecimento de especialista ao agente, permitindo a racionalização a respeito dos dados obtidos e, com isso, promovendo tomadas de decisões mais precisas - capaz de auxiliar na identificação de indícios de fraude em um processo licitatório ao modelar uma situação análoga à problemática analisada. Isso permite que os agentes simulem o comportamento dos promotores agindo sobre as licitações - representadas pelo ambiente - em busca das informações necessárias para a investigação de uma possível fraude, realizando troca de informações com outros agentes e percebendo elementos do documento licitatório e participantes da licitação analisados.

Este trabalho focará na análise de indicativos de fraude encontrados dentro de uma hipertipologia - conjunto de verificações de indício de fraude dentro de uma categoria de irregularidade em um procedimento licitatório - composta de diversas investigações - representadas como tipologias - que, analisadas em conjunto, irão verificar a possibilidade de haver uma fraude licitatória baseada na investigação das categorias apresentadas na tabela 1 e, assim, definindo uma gravidade baseada nos indícios descobertos.

2. Sistema Multiagente

A principal tecnologia utilizada neste projeto é um Sistema Multiagente, uma subárea da Inteligência Artificial que consiste em modelar e simular sistemas complexos como uma organização de agentes autônomos situados em um ambiente lógico compartilhado, comunicando-se e interagindo entre si e com o próprio ambiente, a fim de alcançar objetivos comuns de forma cooperativa [Hübner et al. 2020].

Um Sistema Multiagente (MAS) pode ser dividido em três diferentes abstrações de dimensões, que serão explicadas nas próximas seções: agente, ambiente e organização. Para implementá-las, será utilizada a tecnologia JaCaMo, que combina a linguagem Jason com os frameworks CArtaGO e Moise.

Tabela 1. Tabela de hipertipologias analisadas.

I. Análise da relação entre empresas licitantes	Hipertipologia A Empresas participantes de uma mesma licitação compartilham características	01	Empresa licitante com sócio em comum com outra empresa licitante
		02	Empresa licitantes com ex-sócio em comum com outra empresa licitante
		03	Familiares consanguíneos de primeiro grau em comum entre sócios de empresas licitantes
		04	Sócio de uma empresa licitante é empregado de outra empresa licitante
		05	Empresas licitantes com contador em comum
		06	Empresas licitantes estão sediadas no mesmo endereço
		07	Empresas licitantes possuem mesmo telefone, e-mail ou grupo econômico
	Hipertipologia B Empresas participantes de uma mesma licitação combinam de uma sempre perder (perdedora contumaz) para outra ganhar, ou alternam vitórias	08	Empresa licitante é perdedora contumaz
		09	Empresa licitante é vencedora contra empresa perdedora contumaz
		10	Alternância de vitórias entre empresas licitantes
	Hipertipologia C Empresas participantes de uma mesma licitação apresentam propostas com proporcionalidade de valor/preço	11	Empresa apresenta propostas simétricas acima de patamar definido
	Hipertipologia D Propostas idênticas entre Licitantes em Unidades Gestoras Diferentes	12	Empresa apresenta propostas idênticas acima de patamar definido
	Hipertipologia E Conluio já identificado	13	Empresas que já tiveram conluio comprovado

2.1. Agente

De acordo com [Hübner et al. 2007] e [Hübner et al. 2020], o conceito de agente é uma abstração para programar entidades reativas que exibem um certo grau de autonomia. Portanto, ao atribuímos uma tarefa ou missão, ele é capaz de determinar por si mesmo como concluí-la, tomando decisões com flexibilidade e dinamismo ao reagir a eventos enquanto proativamente direciona seu comportamento para satisfazer suas metas. Uma de suas principais características é o fato de estar situado em um ambiente, sendo capaz de percebê-lo e se comunicar com ele, bem como realizar ações capazes de alterar seu estado. Dessa forma, por meio de interações com o ambiente e outros agentes, ele é capaz de atingir seus objetivos de forma autônoma, racionalizando, em tempo de execução, sobre quais metas irá perseguir e qual é a melhor ação a ser realizada para alcançá-las.

Um agente possui três conceitos fundamentais que regem seu funcionamento: *goal*, *belief* e *action* [Hübner et al. 2020]. Além disso, para um funcionamento adequado, espera-se que um agente inteligente possua autonomia, proatividade, reatividade e habilidade social [Hübner et al. 2007].

Neste trabalho, será utilizada a linguagem de programação orientada a agentes Jason para implementar os agentes do Sistema Multiagente. Os agentes implementados com essa linguagem são baseados no modelo *BDI* (Belief-Desire-Intention) - uma arquitetura baseada na noção de estados mentais para descrever o comportamento do agente, com foco no papel das intenções no raciocínio prático -, representando seus estados de informação, motivação e deliberação [Krausburg et al. 2018].

2.2. Ambiente

A principal característica de um ambiente é o seu dinamismo, permitindo que ele evolua e se adapte de acordo com as necessidades do agente. Essa dimensão é responsável por situar os agentes dentro de um determinado cenário, fornecendo-lhes os meios para afetar o ambiente em que se encontram, alterando certas propriedades, e captar estados e eventos observáveis nele, permitindo-lhes perceber o ambiente e atualizar suas crenças com base nas mudanças. Seu principal objetivo, portanto, é encapsular no sistema recursos e ferramentas em que os agentes operam a fim de receber auxílio para atingir suas metas e

completar seus objetivos [Hübner et al. 2020]. Os principais conceitos atribuídos a essa dimensão são *workspace* e *artifact*.

Em síntese, um ambiente é uma representação do “mundo” sob o qual os agentes operam e estão situados, encapsulando serviços, recursos e ferramentas no Sistema Multi-agente. Sua principal diferença em relação à dimensão dos agentes é a falta de autonomia e proatividade nas entidades que o compõem, considerando que elas apenas reagem às ações realizadas pelos agentes, não tendo a capacidade de racionalizar e tomar decisões por si mesmas [Hübner et al. 2020].

Neste trabalho, será utilizado o framework CArtAgO para implementar os ambientes nos quais os agentes estarão situados. Um agente que foca em um determinado artefato terá, então, percepção de suas *propriedades observáveis* e será capaz de agir sobre ele com base nas operações disponibilizadas pelo artefato [Krausburg et al. 2018].

2.3. Organização

A função da organização é estruturar, coordenar e regulamentar agentes trabalhando em conjunto dentro de um ambiente compartilhado, auxiliando em suas decisões e interações enquanto garante um estado global coerente ao sistema. Dessa forma, ela garante suporte para que múltiplos agentes, dependentes entre si, trabalhem de maneira coordenada e para o controle de sua autonomia dentro do sistema, com base em regras sociais que devem ser obedecidas. Vale ressaltar que, mesmo fazendo parte de uma organização, agentes podem se coordenar entre si sem depender dela. Entretanto, caso o comportamento auto coordenado do agente não esteja de acordo com o que foi especificado na organização, é considerado uma violação. Dessa forma, como a função da organização é impor diretrizes e estabelecer normas, ela poderá aplicar uma punição ao agente transgressor, dependendo da gravidade da infração [Hübner et al. 2020].

A definição das especificações e entidades dessa dimensão pode ser dividida em três abstrações: *structural*, *functional* e *normative*. Embora não seja obrigatório, o uso de especificações estruturais e funcionais é vantajoso em termos computacionais. Isso porque, enquanto a falta de especificação estrutural torna necessário que os agentes negociem entre si a liderança e as possíveis relações e cooperações, devido à falta de conhecimento sobre quem está envolvido em um determinado grupo; a falta de especificação funcional faz com que os agentes precisem raciocinar e definir planos toda vez que forem agir em conjunto, devido à possibilidade de perder aqueles desenvolvidos anteriormente por falta de memória organizacional, e se coordenar entre si [Hübner et al. 2020].

Neste trabalho, será utilizado o framework Moise para implementar as organizações compostas por agentes. A configuração da organização é feita por meio de um arquivo XML que é constituído pelas especificações estrutural, funcional e normativa [Krausburg et al. 2018].

3. Projeto CéOS

O projeto CéOS é um projeto de pesquisa em parceria entre o Departamento de Informática e Estatística (INE) da Universidade Federal de Santa Catarina (UFSC) e o Ministério Público de Santa Catarina (MPSC) com o objetivo de automatizar a análise e extração do conhecimento de grandes volumes de dados, por meio de sua coleta,

integração, organização e processamento, para auxiliar e aprimorar a tomada de decisões em domínios complexos do setor público e, assim, contribuir com a prevenção e o combate à corrupção [de Santa Catarina 2025].

Com o aumento da quantidade de dados disponível para processamento, a extração de informações de editais e documentos se torna uma tarefa complexa e demorada, tornando essencial o uso de tecnologias para administrá-la. O surgimento do projeto decorreu, portanto, da demanda de técnicas de gerenciamento e análise desses dados, possibilitando seu tratamento e manipulação com o intuito de transformá-los em informação útil e relevante aos promotores e, assim, auxiliá-los na otimização da inspeção e análise de licitações e indicadores de possíveis fraudes licitatórias [Wüst et al. 2024].

Este trabalho busca, no âmbito do projeto CÉOS, desenvolver um Sistema Multiagente que será integrado em um painel inteligente de licitações - uma interface de usuário contendo informações a respeito de processos licitatórios e alertas identificados associados à eles que poderão ser acessadas, sem a necessidade de consultar diferentes bases de informação, para proporcionar uma análise integrada - com o objetivo de verificar, de forma inteligente e autônoma, se há indícios de fraude nos processos licitatórios, agilizando o processo de investigação [Wüst et al. 2024].

4. Trabalhos Relacionados

Para o fortalecimento do embasamento teórico e prático a respeito da problemática deste trabalho, foi realizada uma revisão do estado da arte, focada nas áreas de Sistema Multiagente e de Fraude Licitatória/Monetária, em repositórios como o Google Scholar e o DBLP para a busca de trabalhos relacionados.

Essa seção, portanto, consiste na apresentação de dois artigos encontrados que possuem relação com o tema deste trabalho, percorrendo a respeito de seu conteúdo e do diferencial da proposta apresentada pelos autores.

4.1. A multi-agent data mining system for cartel detection in Brazilian government procurement

Em [Ralha and Silva 2012], a detecção de cartéis no processo de detecção de corrupção é reconhecida como um problema complexo, devido ao grande volume de dados disponível e necessário para correlacionar as informações e as diversas estratégias distintas utilizadas para esconder as operações fraudulentas.

Devido a isso, os autores propuseram o desenvolvimento do AGMI, um *agent-mining tool* que integra as estratégias de mineração de dados distribuída (*DDM*), descoberta de conhecimento em bancos de dados (*KDD*) e sistema multiagente (*MAS*), se tratando de um sistema de mineração de dados multiagente projetado para auxiliar na detecção, prevenção e monitoramento de cartéis que atuam em processos de compras públicas.

O principal diferencial da abordagem proposta, dessa forma, se trata ao implementar um agente inteligente em conjunto com a análise tradicional de mineração de dados, possibilitando a inclusão de heurísticas baseadas no conhecimento de especialistas.

O trabalho [Ralha and Silva 2012], portanto, se destaca como uma solução de abordagem híbrida, demonstrando a importância da adição de conhecimento especialista

na estratégia para superar limitações da mineração clássica na detecção de irregularidades, como a detecção de cartelização em compras públicas, e aumentar eficácia e desempenho.

4.2. A Multi-Agent System Based Approach to Fight Financial Fraud: An Application to Money Laundering

Em [Alexandre and Balsa 2018], é reconhecida a ineficiência do processo de investigação semi-automatizado, envolvendo regras fixas, de identificação de crimes de lavagem de dinheiro e de auxílio à tomada de decisões, devido ao crescente volume de transações e a constante publicação de novas regulamentações.

Devido a isso, os autores propuseram uma abordagem baseada em sistemas multiagente para analisar e sinalizar transações suspeitas, auxiliando o analista humano na investigação de comportamentos suspeitos. Para isso, é realizada a criação de um perfil de comportamento transacional dos clientes, utilizando tecnologias de mineração de dados para buscar um histórico de operações.

O principal diferencial da abordagem proposta, dessa forma, se trata da utilização desses dados extraídos para compor uma base de conhecimento de um sistema multiagente, implementado na plataforma JaCaMo. A arquitetura, então, é composta por agentes especializados em detectar suspeitas e por agentes que, além de sinalizar os resultados obtidos, atua como um assistente do analista humano, podendo aprender com as decisões tomadas por ele.

O trabalho [Alexandre and Balsa 2018], portanto, se destaca ao demonstrar a capacidade da abordagem em identificar comportamentos irregulares complexos de forma mais eficaz e eficiente, identificando, inclusive, casos de lavagem de dinheiro que não haviam sido previamente detectados por sistemas tradicionais baseados em regras estáticas.

5. Metodologia

A modelagem do sistema proposto é realizada por meio de trilhas de investigação, compostas por diversas tipologias que, em conjunto, formam uma hipertipologia responsável por representar uma categoria de fraude. O sistema, então, utilizará diversos agentes, agrupados em uma organização e trabalhando de forma coordenada e cooperativa, a fim de verificar cada uma dessas tipologias e, a partir da apuração obtida, estabelecer a gravidade da fraude - caso seja encontrado algum indício -, que será exibida ao usuário por meio de um alerta.

O sistema proposto neste projeto, dessa forma, é responsável por averiguar uma análise da relação entre empresas licitantes, analisando as seguintes hipertipologias: A (“empresas participantes de uma mesma licitação compartilham características”), B (“empresas participantes de uma mesma licitação combinam de uma sempre perder para outra ganhar ou alternam vitórias”), C (“empresas participantes de uma mesma licitação apresentam propostas com proporcionalidade de valor/preço”), D (“propostas idênticas entre licitantes em Unidades Gestoras diferentes”) e E (“conluio já identificado”).

A organização do sistema multiagente será composta por:

- um agente gerente: responsável por gerenciar os outros agentes, criando um grupo para cada hipertipologia e definindo papéis aos agentes responsáveis por coordenar um respectivo grupo, e por buscar na base de dados os editais de licitação que serão analisados, selecionando-os e repassando-os aos agentes de hipertipologia;

- cinco agentes hipertipologia: responsável por criar um grupo para a respectiva hipertipologia, definindo papéis aos agentes responsáveis por realizar a análise das tipologias, por repassar os editais selecionados aos agentes de tipologia, os quais são enviados de forma sequencial, um por vez, até a finalização da análise do que está com investigação em curso, e por criar o alerta da respectiva hipertipologia - ao receber o resultado de todos os agentes de tipologia pelos quais está responsável -, definindo o seu nível de gravidade - o maior valor de gravidade entre as tipologias detectadas ou o nível de gravidade da hipertipologia, caso todas tenham sido identificadas - e gerando uma breve descrição a respeito das tipologias identificadas - a partir dos dados a respeito dessas tipologias e dos dados salvos no documento JSON - para, dessa forma, notificar as principais informações a respeito da fraude identificada ao usuário;
- treze agentes tipologia: responsável por realizar consultas na base de dados a respeito da tipologia da qual é encarregado, verificando se há algum indício de fraude no edital licitatório que está sendo investigado e, em caso positivo, armazenando informações relevantes a respeito da irregularidade e da empresa/licitação em um documento JSON, e retornar ao seu respectivo agente de hipertipologia o resultado da análise, sendo representado como uma lista com *false* em caso de regularidade e com informações a respeito da tipologia - número da tipologia, nome da tipologia e gravidade da tipologia - em caso de irregularidade.

Esses agentes serão divididos em dois tipos de grupos distintos. O primeiro, chamado *groupHipertipologia*, contém o agente gerente e todos os agentes hipertipologia. O segundo, chamado *groupHipertipologiaN* - onde N representa os valores A, B, C, D e E, dependendo da hipertipologia abordada no grupo -, contém o agente hipertipologia correspondente ao grupo e seus respectivos agentes tipologia. A figura 1 apresenta uma representação da arquitetura descrita em relação aos grupos *groupHipertipologia* e *groupHipertipologiaB* - considerando que os outros *groupHipertipologiaN* possuem estrutura semelhante - e aos artefatos que estão observando.

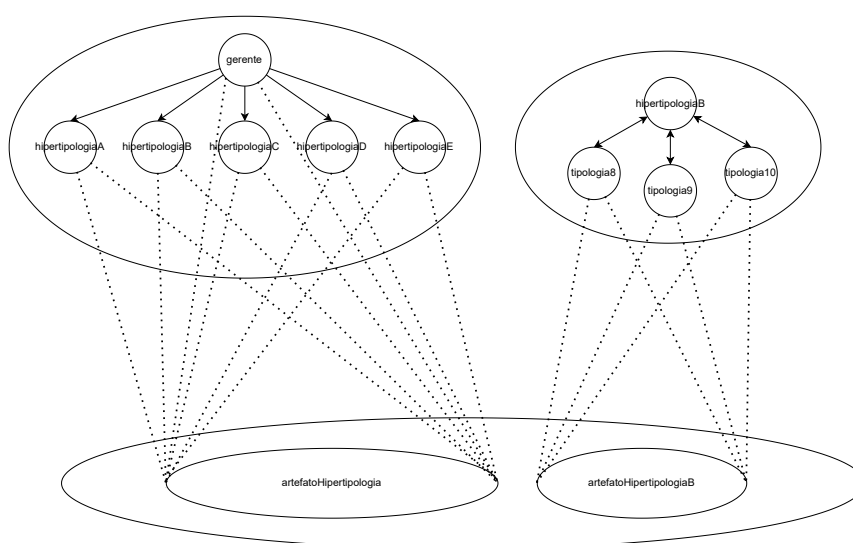


Figura 1. Modelagem da arquitetura dos grupos da organização.

O grupo *groupHipertipologia* possui o *schemeHipertipologia*, que é separado em

duas *missions* diferentes. A primeira consiste na criação e gerenciamento dos grupos de cada hipertipologia. Por outro lado, a última consiste na busca pelos editais licitatórios na base de dados e seu posterior repasse aos agentes *hipertipologia*, sendo uma obrigação do agente *gerente*. O funcionamento geral do grupo pode ser observado na figura 2.

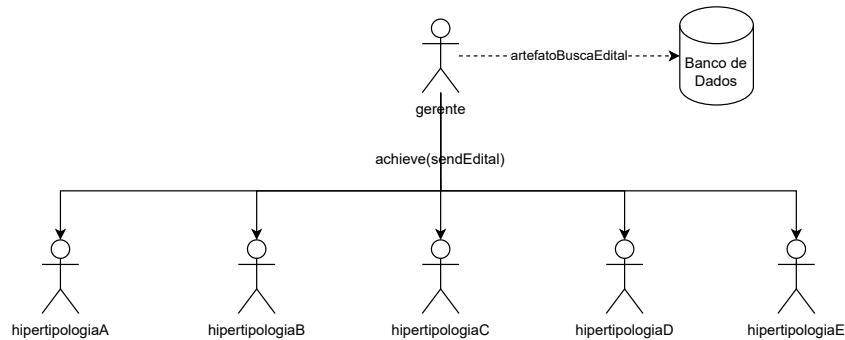


Figura 2. Modelagem da funcionalidade do grupo *groupHipertipologia*.

Em relação ao grupo *groupHipertipologiaN*, haverá um *scheme* por hipertipologia, sendo denominados como *schemeHipertipologiaN*. Eles funcionam de forma semelhante, sendo separados em uma *mission* por tipologia presente na hipertipologia. Cada *mission*, então, consiste em realizar a busca na base de dados a respeito da tipologia investigada, retornando o resultado da análise do indicativo de fraude. Essas *missions* serão atribuídas, de forma obrigatória, aos agentes *tipologia*, sendo determinadas a partir do *role* da tipologia correspondente. A figura 3 apresenta o funcionamento geral do grupo *groupHipertipologiaB*, se tratando de um exemplo que representa uma execução de ações semelhante a observada no resto dos grupos *groupHipertipologiaN*.

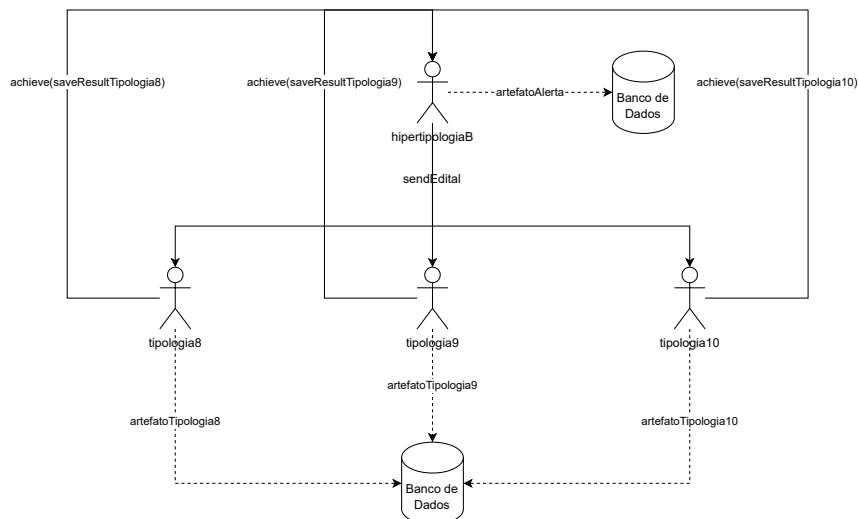


Figura 3. Modelagem da funcionalidade do grupo *groupHipertipologiaB*.

O sistema multiagente inicia com o agente *gerente* executando o *plan start*, o qual é responsável por criar a organização e o artefato correspondente a ela, se juntando a ela em sequência. Em seguida, ele será responsável por criar o grupo *groupHipertipologia* e

criar o artefato que será observado e utilizado pelos membros desse grupo. Com o grupo criado, ele enviará, para cada agente presente em *membrosHipertipologia* - uma lista em sua base de crenças contendo os agentes que farão parte do *groupHipertipologia* -, um *achieve* para que eles se unam à organização e, em sequência, adotem um *role* nela. Por fim, ao receber a confirmação da formação completa do grupo, ele criará o *scheme* do grupo.

Ao finalizar esse conjunto de ações definido no *plan start*, ele irá executar a *mission missionBuscaEdital*, que consiste em buscar na base de dados os editais de licitação que serão analisados. Isso é feito a partir de uma consulta SQL ao banco de dados, armazenando o resultado como um array de string em uma propriedade observável do artefato, denominada *resultEdital*. Essa propriedade, inicializada no artefato com o valor *false*, assume um valor no formato `[id.procedimento.licitatório1, id.procedimento.licitatório2, ..., id.procedimento.licitatórioN]`, sendo N a quantidade de editais encontradas na base de dados.

O agente *gerente*, ao observar uma mudança na propriedade *resultEdital*, irá buscar a lista de membros do *groupHipertipologia* e enviar a todos o array de editais licitatórios selecionados, a partir de um *achieve* para que eles executem o *plan sendEdital*. Seu funcionamento pode ser observado no fluxo de interação apresentado na figura 4.

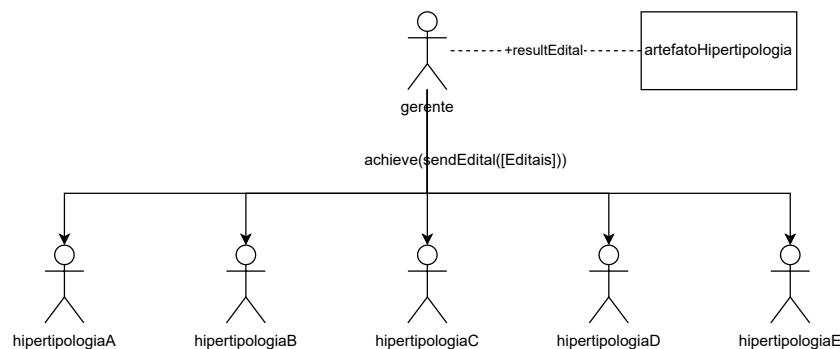


Figura 4. Fluxo de interação dos agentes em *resultEdital* do agente *gerente*.

Os agentes *hipertipologia*, por outro lado, inicializam e aguardam o agente *gerente* informar que é para eles se unirem à organização e começarem a focar nos artefatos da organização e do grupo ao qual pertencem, adotando o papel correspondente à letra da hipertipologia ao qual pertencem em sequência.

Esses agentes, ao entrarem no grupo, irão executar a *mission missionHipertipologia*, que funciona de forma semelhante ao *goal start* do agente *gerente*, criando um *workspace* para o grupo específico da respectiva tipologia ao invés da organização e os grupos, artefatos e *schemes* correspondentes a esse grupo. Além disso, vale ressaltar que, ao invés de usar uma lista de membros na base de crenças para definir os agentes que serão membros do grupo, é utilizada uma concatenação de *tipologiaX* - sendo X a letra correspondente a do agente *hipertipologia* - com um valor N - tendo N variando de 1 até a quantidade de tipologias presente na hipertipologia em questão.

Em seguida, ao receber um *achieve* do agente *gerente*, eles irão executar o *plan*

sendEdital, que consiste em selecionar o primeiro edital do array e enviar para todas as tipologias pelas quais são responsáveis. Após realizar o envio, os agentes *hipertipologia* atualizam sua crença *editalAtual* para o número do edital licitatório que foi enviado e sua crença *editalRestante* para o resto do array de editais. A definição do *plan sendEdital* pode ser observada a partir do fluxo de interação da figura 5.

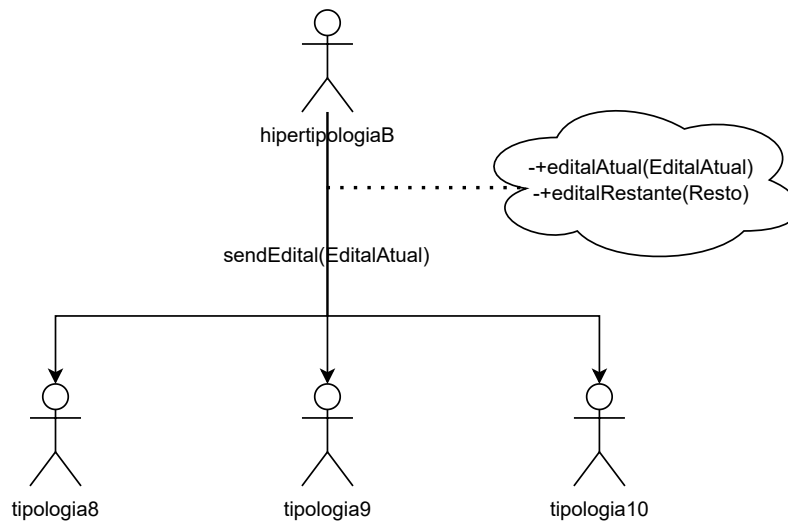


Figura 5. Fluxo de interação dos agentes em *sendEdital* do agente *hipertipologiaB*.

Os agentes *hipertipologia*, então, passam a aguardar o resultado da análise dos agentes *tipologia* pelos quais são responsáveis. Isso é feito a partir do valor de *hypertipologia* em sua base crenças, definida inicialmente como *hypertipologia([null,null,...,null])*, sendo um array que tem N posições - N representando a quantidade de tipologias que cada hipertipologia possui - definidas como *null*.

Quando houver uma alteração ao valor da crença *hypertipologia*, os agentes *hipertipologia* irão reagir a isso, realizando uma sequência de ações apenas ao receber todos os resultados de análise das tipologias, o que é verificado ao checar se todos os valores do array é diferente de *null*. Com isso, seu próximo passo é realizar algumas verificações para definir que “caminho” seguir. A primeira é se houve algum indício de fraude ou não, analisando as informações recebidas dos agentes *tipologia* para verificar se o primeiro valor da lista de todas as informações recebidas são *false*, indicando que não foi detectada nenhuma fraude em caso positivo. Nessa situação, os agentes *hipertipologia* não irão criar nenhum alerta, apenas voltar a crença *hypertipologia* ao valor inicial e enviar o próximo procedimento licitatório a ser analisado.

A segunda situação se trata de quando há indicativo de fraude, porém somente de uma parte da hipertipologia, ou seja, apenas algumas tipologias detectaram indícios de fraude, enquanto outras estão regulares. Para checar isso, de forma semelhante a situação anterior, verifica o primeiro valor da lista de todas as informações recebida, analisando se algum deles é igual a *false*, indicando que pelo menos um agente *tipologia* não detectou nenhuma fraude em caso positivo. Nessa situação, será criado um alerta. Para isso, entretanto, será necessário primeiro definir qual será o nível de gravidade dele. Os

agentes *hipertipologia*, para isso, definem qual tipologia possui o maior valor, utilizando o resultado para definir a gravidade do alerta como um todo.

Ao obter esse valor, eles buscam qual é o edital que está sendo analisado no momento e as informações a respeito da hipertipologia para construir o alerta, que consiste em, inicialmente, verificar os resultados de fraude que foram passados aos agentes *hipertipologia*, buscando o documento JSON com as informações da fraude caso o resultado não seja *false*. Ao identificar o documento, o conteúdo é lido e convertido em um JSONArray, que será usado para a construção da descrição longa do alerta. Essa descrição é gerada a partir de um template, que possui uma formatação pré-definida de acordo com a hipertipologia e com a tipologia, utilizando as informações obtidas do JSON para completar as frases geradas. Em seguida, é executado um SQL para adicionar o alerta e sua respectiva categoria - caso não tenha sido previamente armazenada - no banco de dados, com base nas informações obtidas. Por fim, a crença *hipertipologia* é redefinida para seu valor inicial e um novo procedimento licitatório é enviado para ser analisado. A figura 6 onstra o fluxo de funcionamento da geração de alertas.

A última situação se trata de quando foi detectado indício de fraude para todas as tipologias presentes na hipertipologia, sendo caracterizada como uma "fraude completa". Essa situação irá funcionar de forma semelhante a anterior, buscando as informações da hipertipologia e, a partir dela, criando um alerta. Entretanto, ao invés de calcular o valor da gravidade, será utilizado o nível padrão da hipertipologia, definido em *info_hiptipologia*.

Os agentes *tipologia*, por fim, inicializam e aguardam o agente *hipertipologia* correspondente informar que é para eles se unirem à organização e focarem nos respectivos artefatos da organização e do grupo. Para definir o *role* que o agente irá assumir no grupo, será verificado se um determinado *role* do grupo ao qual pertence já foi adotado por outro agente, contando quantos agentes assumiram esse *role* e somente adotá-lo caso nenhum outro agente tenha assumido ele. Ao definir um *role*, além de adotá-lo, o agente armazena o nome do respectivo *role* e as informações da tipologia correspondente, no formato *info_tipologia(NumeroTipologia, NomeTipologia, GravidadeTipologia)*, na base de crença.

Esses agentes, ao entrarem no grupo, irão executar a *mission missionTipologiaN* - sendo N o valor numérico que representa o número da tipologia pela qual o agente é responsável -, que é responsável por realizar a consulta da tipologia sobre o edital licitatório que está sendo investigado. Para isso, os agentes, inicialmente, buscam as informações a respeito da tipologia - obtidas a partir da crença *info_tipologia* - e esperam o agente *hipertipologia* responsável repassar o edital que será verificado.

Ao receber o edital, eles irão realizar uma consulta SQL no banco de dados, armazenando o resultado em uma propriedade observável *resultTipologiaN* como um array de arrays de tuplas, em que cada array dentro do "array principal" equivale a um resultado da consulta e os elementos desses arrays exibem as informações obtidas a partir da consulta, sendo armazenados no formato *nomeAtributo(valorAtributo)*. Além disso, a função é responsável por, caso tenha pelo menos um valor de retorno, salvar os dados em um arquivo JSON. Por fim, eles irão indicar que o edital terminou de ser analisado, removendo *num_edital* de sua base de crenças e esperando pelo próximo edital.

Ao mesmo tempo, os agentes irão reagir quando observarem uma mudança

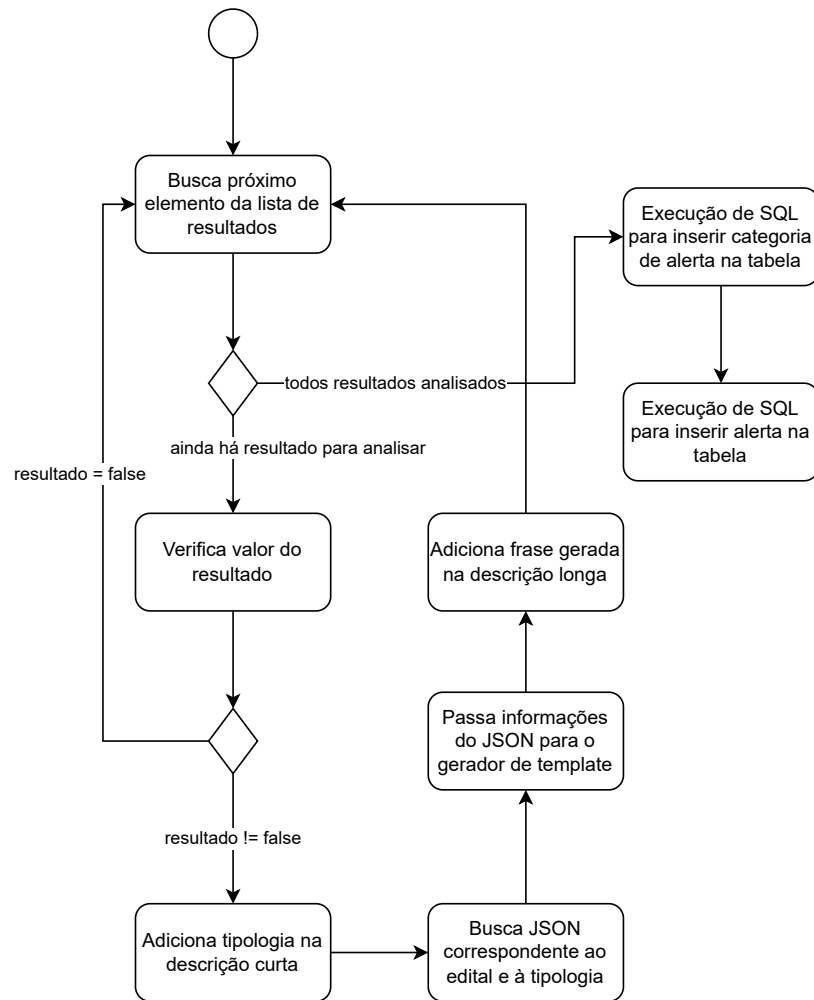


Figura 6. Fluxo de geração de alertas do artefato *ArtefatoHipertipologia*.

de valor na propriedade observável *resultTipologiaN*, verificando se o retorno da análise é uma lista vazia, o que indica que não foi identificado nenhum indicativo de fraude. Em caso afirmativo, será enviado ao agente *hipertipologia* responsável um *achieve saveResultTipologiaN* - *plan* responsável por guardar o resultado da investigação - com o valor `[false]`, indicando que nenhum indício foi detectado. Em caso negativo, por outro lado, serão novamente buscadas as informações a respeito da tipologia, sendo enviado ao agente *hipertipologia* responsável um *achieve saveResultTipologiaN* com os dados da tipologia no formato `[NumeroTipologia, NomeTipologia, GravidadeTipologia]`, indicando que foi detectado indicativo de fraude. É apresentado, tendo a tipologia 8 como referência, o fluxo de retorno do resultado da consulta por meio da figura 7.

6. Resultados

Devido aos dados utilizados na análise serem sigilosos, estando protegidos por um contrato de confidencialidade, optou-se pela criação de um estudo de caso com dados fictícios

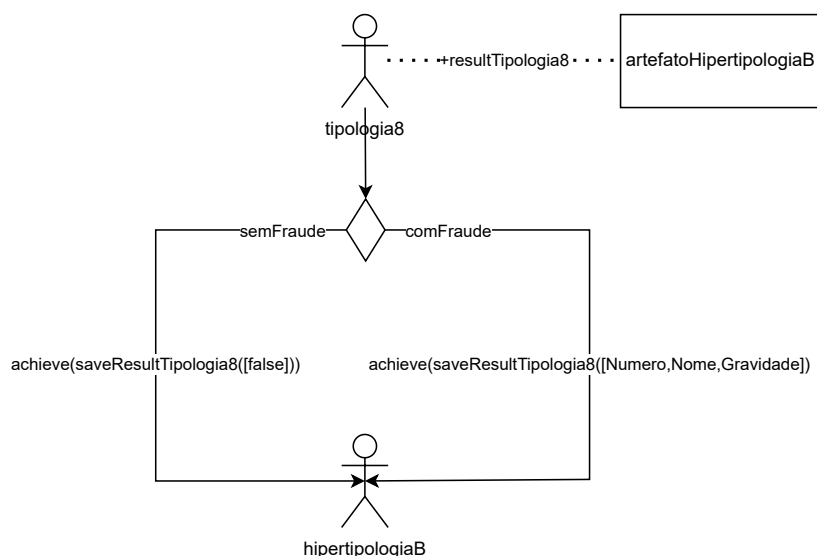


Figura 7. Fluxo de interação dos agentes no retorno de resultado do agente *tipologia8*.

para demonstrar o funcionamento do trabalho. Para isso, foram projetadas tabelas em postgresSQL com uma arquitetura semelhante a original.

Para validar o sistema desenvolvido, as tabelas criadas foram preenchidas com diversos dados diferentes, sendo adicionados quinze processos licitatórios, havendo sessenta e nove itens de licitação distribuídos entre eles. Em relação aos participantes, a tabela armazena um total de cinquenta e dois participações, resultando em duzentos e sessenta e uma cotações para cobrir todos os itens e licitações. Vale ressaltar que, entre essas participações, há um total de vinte estabelecimentos, possuindo sessenta e seis sócios distribuídos entre eles. Os dados utilizados foram elaborados pelo autor, com a inclusão intencional de cenários fraudulentos, com o propósito de validar o funcionamento das consultas SQL utilizadas na detecção dessas irregularidades.

Entre esses dados, foram pré-definidas as seguintes fraudes:

- Licitação 1: tipologias 6, 8 e 9.
- Licitação 2: sem fraude.
- Licitação 3: tipologias 1, 6, 7, 8 e 9.
- Licitação 4: tipologias 1 e 6.
- Licitação 5: sem fraude.
- Licitação 6: sem fraude.
- Licitação 7: tipologias 6 e 7.
- Licitação 8: tipologias 1 e 10.
- Licitação 9: tipologia 11.
- Licitação 10: tipologias 8 e 9.
- Licitação 11: sem fraude.
- Licitação 12: tipologias 1 e 11.
- Licitação 13: tipologias 1 e 10.
- Licitação 14: tipologias 1 e 10.

- Licitação 15: sem fraude.

Vale ressaltar que, embora o sistema tenha sido projetado para suportar todas as hipertipologias e tipologias dentro do escopo definido, não foi possível obter todos os dados necessários para a análise. Devido a isso, algumas tipologias - 2, 3, 4, 5 e 13 - não serão analisadas, retornando o resultado como se nenhuma fraude tivesse sido encontrada e adicionando um log “Dados insuficientes”.

Ao executar o sistema, no entanto, podem ser observados os seguintes resultados:

- Licitação 1: tipologia 11.
- Licitação 2: tipologia 11.
- Licitação 3: tipologias 1, 6, 7 e 11.
- Licitação 4: tipologias 1, 8, 9, 10 e 11.
- Licitação 5: tipologia 11.
- Licitação 6: tipologia 11.
- Licitação 7: tipologias 6, 7, 10 e 11.
- Licitação 8: tipologias 1, 10 e 11.
- Licitação 9: tipologias 8, 9, 10 e 11.
- Licitação 10: tipologia 11.
- Licitação 11: tipologia 11.
- Licitação 12: tipologias 1 e 11.
- Licitação 13: tipologias 1, 8, 9, 10 e 11.
- Licitação 14: tipologias 1, 10 e 11.
- Licitação 15: sem fraude.

Baseado nos resultados obtidos, pode-se observar diversas situações. A primeira é a dificuldade em lidar com dados que possam estar em formatos diferentes. Isso ocorre nas investigações da tipologia 6, que consistem em analisar empresas sediadas no mesmo endereço. Devido a isso, quando há variações em nomes, como “Doutor” e sua abreviação “Dr.”, para identificar um local, o agente os tratará como dados diferentes, não considerando isso como uma possível variação da mesma palavra e, portanto, deixando de identificar o indício de fraude.

Além disso, nas investigações relacionadas à tipologia 11, que envolve propostas entre empresas distintas com uma diferença de valor inferior a 3%, existem alguns falsos positivos. Isso porque, no estudo de caso desenvolvido, considerou-se que, por se tratar de uma disputa competitiva, os valores das propostas seriam quase sempre bem semelhantes entre si. Dessa forma, mesmo que o agente tenha identificado corretamente o indicativo de fraude - considerando que as propostas possuem, de fato, uma diferença inferior a 3% -, não se trata de um caso intencional de fraude. Portanto, embora a análise em si esteja sendo realizada corretamente, seria necessário realizar uma investigação mais aprofundada antes de ser sinalizado uma possível fraude. Nesse cenário, portanto, a abordagem ideal seria conduzir uma “trilha de investigação”, verificando outras informações, antes de gerar um alerta.

Por fim, há uma falha na investigação das tipologias 8, 9 e 10. Analisando os alertas gerados, entretanto, observa-se que o erro possivelmente decorre do fato de que o SQL utilizado para a análise está considerando apenas os dados dentro do escopo de um único procedimento licitatório na investigação, ignorando o histórico completo de vitórias e derrotas de uma empresa - uma informação importante a ser considerada para

as tipologias analisadas. Portanto, basta um pequeno ajuste no SQL dessas tipologias para que a verificação funcione corretamente, gerando os alertas esperados.

7. Conclusão

Este trabalho se propôs a avaliar a viabilidade de um sistema multiagente na detecção de indicativos de fraudes em uma licitação. O objetivo principal era o desenvolvimento de uma versão inicial de um sistema multiagente capaz de, a partir de consultas a uma base de dados com as informações necessárias, identificar se houve algum indicativo de fraude - baseado nas hipertipologias investigadas neste escopo -, gerando um alerta informativo ao usuário.

Para atingir esse objetivo foi utilizada a tecnologia de sistemas multiagente e de JaCaMo - Jason, CArtaGO e Moise - para implementar uma arquitetura completa, com organização bem definida, capaz de, de maneira organizada, realizar uma consulta ao banco de dados e retornar o resultado da análise de fraude - de acordo com cada tipologia - ao seu agente “responsável”, que irá receber esses dados e lidar com o gerenciamento deles, criando os respectivos alertas. Dessa forma, foi possível proporcionar uma boa divisão de tarefas, fazendo com que cada agente ficasse responsável por uma função específica e, assim, focasse em realizar seu objetivo de forma eficiente.

Este trabalho, portanto, demonstrou que a abordagem de sistemas multiagente se trata de uma abordagem viável e promissora no processo de automatização de investigação de fraudes em procedimentos licitatórios, sendo um grande auxílio na tomada de decisão e na redução da carga de dados a serem analisados. Dessa forma, a principal contribuição deste trabalho foi a implementação de uma ferramenta de auxílio para promotores capaz de automatizar a busca por dados, o que pode ser demorado e trabalhoso, e contribuir na investigação e na tomada de decisões de situações sutis que podem passar despercebidas para um humano. Seu funcionamento foi validado a partir do estudo de caso apresentado, que demonstrou o desempenho e a eficácia do sistema.

Apesar disso, este trabalho possui algumas limitações. Primeiramente, há diversos dados que não foi possível obter, tornando a investigação menos completa do que o planejado. Outra situação se trata de o agente possuir um raciocínio simplificado, racionalizando sobre o dado obtido sem definir uma sequência de investigações a serem feitas antes de definir uma determinada situação como potencial fraude, o que acaba gerando alguns falsos-positivos e diminuindo sua eficácia. Por fim, há a dificuldade de lidar com dados que possam estar em diferentes formatos - como endereços iguais escritos de formas distintas - ou incompletos - como a censura de dados sensíveis.

A partir das conclusões e limitações identificadas, sugere-se, primeiramente, a expansão do sistema, adicionando mais hipertipologias ou categorias de fraudes para serem analisadas pelos agentes, o que permite um aumento do escopo de irregularidades identificáveis, e implementando um *trigger* no banco de dados, responsável por informar ao agente sempre que um novo edital licitatório for adicionado ao sistema e, assim, automatizando a investigação sobre ele. Além disso, sugere-se oferecer mais robustez ao sistema, implementando uma linha de raciocínio lógico ao agente. Isso permitiria que os agentes, ao invés de somente exibir as informações encontradas na consulta à base de dados, racionalizassem sobre elas, seguindo uma “trilha de investigação” ao simular o “passo a passo” de como um promotor pensa para definir se determinada situação se trata de uma

fraude. Ademais, para aprimorar a análise de similaridade entre dados em formatos distintos, sugere-se a implementação de métodos de cálculo da distância de Hamming ou de Levenshtein, possibilitando a identificação de divergências ortográficas, variações de escrita e inconsistências. Por fim, sugere-se a generalização e refatoração de alguns trechos de código, como os dos agentes *hipertipologia*, para evitar repetições desnecessárias.

Referências

- (2016). *Preventing Corruption in Public Procurement*. Organisation for Economic Cooperation and Development.
- (2024). *Licitações Contratos - Orientações e Jurisprudência do TCU*. Tribunal de Contas da União.
- (2024). *Métodos de Detecção de Fraude e Corrupção em Contratações Públicas*. Transparência Brasil.
- Alexandre, C. and Balsa, J. (2018). A multi-agent system based approach to fight financial fraud: An application to money laundering. *Preprints*.
- da Silva, J. V. S. and de Oliveira, R. D. (2024). Fraude em licitações: uma perspectiva da lei nº 14.133/2021. *Revista Científica COGNITIONIS*.
- de Santa Catarina, U. F. (2025). CéOs - inteligência artificial em benefício da sociedade.
- Hübner, J. F., Boissier, O., Bordini, R. H., and Ricci, A. (2020). *Multi-Agent Oriented Programming: Programming Multi-Agent Systems Using Jacamo*. MIT Press.
- Hübner, J. F., Bordini, R. H., and Wooldridge, M. (2007). *Programming Multi-Agent Systems in Agentspeak Using Jason*. Wiley-Interscience.
- Krausburg, T., Cardoso, R. C., Damasio, J., Peres, V., Farias, G. P., Engelmann, D. C., Hübner, J. F., and Bordini, R. H. (2018). Smart-jacamo: an organisation-based team for the multi-agent programming contest. *The Multi-Agent Programming Contest*.
- Ralha, C. G. and Silva, C. V. S. (2012). A multi-agent data mining system for cartel detection in brazilian government procurement. *Expert Systems with Applications*.
- Rodrigues, N. C. S. and Filho, R. N. L. (2017). Modalidades licitatórias e o risco de ocorrência de fraudes nos municípios baianos fiscalizados pela controladoria geral da união. *CCCSS Contribuciones a Las Ciencias Sociales*.
- Wüst, A., Dorneles, C. F., Tyska, J., Marchi, J., Castro, M., Fileto, R., Mello, R., and Werner, S. S. (2024). CéOs: Inteligência de dados para a sociedade. Technical report, Universidade Federal de Santa Catarina, Florianópolis.

APÊNDICE B – CÓDIGO FONTE

O código fonte completo deste trabalho, incluindo os agentes em Jason, artefatos em CArTAgO e organização em Moise, está acessível publicamente no link <https://codigos.ufsc.br/mariana.mazzo.heidtor/sma-deteccao-fraude>. Para executar o sistema, siga as instruções disponíveis no README.md do repositório.

O repositório está organizado da seguinte forma:

- **src/agt:** Contém os códigos fonte dos agentes (.asl).
- **src/env/example:** Contém os artefatos do ambiente (.java).
- **src/org:** Contém as configurações da organização (.xml).
- **dados_estudo_caso:** Contém o backup do banco de dados utilizado no estudo de caso.