UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ

GABRIEL LEONARDO MARTINS DE OLIVEIRA

SISTEMA MULTIAGENTE PARA NEGOCIAÇÃO DE ATIVOS DIGITAIS

PONTA GROSSA

2024

GABRIEL LEONARDO MARTINS DE OLIVEIRA

SISTEMA MULTIAGENTE PARA NEGOCIAÇÃO DE ATIVOS DIGITAIS

Multi-Agent System for Digital Asset Trading

Trabalho de Conclusão de Curso de Graduação apresentado como requisito para obtenção do título de Bacharel em Ciência da Computação do Curso de Bacharelado em Ciência da Computação da Universidade Tecnológica Federal do Paraná.

Orientador: Prof. Gleifer Vaz Alves

Coorientador: Prof. Nilson Mori Lazarin

PONTA GROSSA

2024

4.0 Internacional

Esta licença permite compartilhamento, remixe, adaptação e criação a partir do trabalho, mesmo para fins comerciais, desde que sejam atribuídos créditos ao(s) autor(es). Conteúdos elaborados por terceiros, citados e referenciados nesta obra não são cobertos pela licença.

GABRIEL LEONARDO MARTINS DE OLIVEIRA

SISTEMA MULTIAGENTE PARA NEGOCIAÇÃO DE ATIVOS DIGITAIS

Trabalho de Conclusão de Curso de Graduação apresentado como requisito para obtenção do título de Bacharel em Ciência da Computação do Curso de Bacharelado em Ciência da Computação da Universidade Tecnológica Federal do Paraná.

Data de aprovação: 19/dezembro/2024

Gleifer Vaz Alves
Doutor
Universidade Tecnológica Federal do Paraná

Nilson Mori Lazarin
Doutor
Centro Federal de Educação Tecnológica do Rio de Janeiro

André Pinz Borges
Doutor
Universidade Tecnológica Federal do Paraná

Carlos Eduardo Pantoja
Doutor
Centro Federal de Educação Tecnológica do Rio de Janeiro

PONTA GROSSA

2024

AGRADECIMENTOS

A conclusão deste trabalho representa o encerramento de uma importante etapa da minha vida, e isso não seria possível sem o apoio de pessoas que estiveram ao meu lado durante toda essa jornada.

À minha mãe Ivania que acreditou no meu potencial, me dando toda oportunidade possível para crescer na minha vida pessoal, acadêmica e profissional, sendo minha maior base e meu maior motivo para não desistir e correr atrás dos meus sonhos. Ao meu pai Duani, que mostrou confiança em mim, aconselhando e me mantendo no caminho certo, agradeço por demonstrar este orgulho que tem de mim. Seja nos momentos de dúvida ou nas conquistas, sou eternamente grato à vocês dois, suas palavras de incentivo e amor foram o alicerce para que eu pudesse superar os desafios e chegar até aqui.

À minha namorada Thais, que também encarou o desafio que era morar longe, fomos um para o outro um refúgio, onde pudemos nos consolar e encontrar forças para seguir nossa jornada e encarar os desafios do dia a dia juntos. Sou feliz e grato por ter você ao meu lado, esta conquista também é sua.

Ao meu falecido avô, cuja memória carrego com carinho, infelizmente o senhor não me viu formado, mas sei que esteve e sempre estará torcendo por mim lá do céu, espero te encher cada vez mais de orgulho. Às minhas avós, que sempre demonstraram felicidade e amor em tudo que faço, mesmo com muitas saudades, dedico este trabalho como forma de honrar tudo o que me ensinaram e os valores que transmitiram.

Aos meus tios e tias, que de diversas maneiras contribuíram para o meu crescimento, seja com conselhos, risadas ou apoio nos momentos mais difíceis, o meu sincero obrigado.

Aos meus amigos na minha cidade natal, estiveram me apoiando mesmo à distância, em viagens e reencontros que me fizeram ter mais fé em mim e força para continuar, me incentivando e me alegrando durante todo o tempo. Vocês foram fundamentais para que eu mantivesse a motivação e a sanidade ao longo do caminho.

Por fim, aos meus novos amigos que fiz nesta jornada e também à meus professores, com quem compartilhei discussões, trabalhos em grupo e o desejo mútuo de nos tornarmos melhores a cada dia, meu reconhecimento e gratidão. Juntos, enfrentamos muitos obstáculos e celebramos nossas conquistas.

A cada um de vocês, dedico este trabalho como um reflexo do apoio, carinho e inspiração que sempre me proporcionaram. Muito obrigado!

RESUMO

Este trabalho apresenta o desenvolvimento de um sistema multiagente BDI integrado à tecnologia blockchain, utilizando o *middleware* Velluscinum para facilitar negociações descentralizadas de ativos digitais em um ambiente de Smart Parking. O objetivo foi criar uma solução que combinasse as capacidades de agentes BDI, caracterizados por suas crenças, desejos e intenções, com a segurança e transparência proporcionadas pela *blockchain*. Para alcançar esse objetivo, foram analisadas negociações existentes, definidos e implementados mecanismos de negociação, e realizados testes que validaram o desempenho do sistema em diferentes cenários. Os resultados demonstraram a funcionalidade do sistema e sua viabilidade prática, destacando o uso de NFTs como representação digital das vagas de estacionamento. No entanto, limitações foram observadas, como a centralização do gerenciamento das transações e a dependência de conectividade de rede, que impactaram o desempenho em cenários mais complexos. Apesar disso, o sistema contribui para avanços no uso de *blockchain* e agentes BDI, com perspectivas de melhorias futuras, como a descentralização do modelo e integração com outros componentes de *Smart Cities*.

Palavras-chave: agentes bdi; blockchain; sistema multiagentes; smart parking; ativos digitais.

ABSTRACT

This work presents the development of a BDI multi-agent system integrated with blockchain technology, utilizing the Velluscinum middleware to enable decentralized digital asset negotiations in a Smart Parking environment. The objective was to create a solution combining the capabilities of BDI agents, characterized by their beliefs, desires, and intentions, with the security and transparency offered by blockchain. To achieve this, existing negotiations were analyzed, negotiation mechanisms were defined and implemented, and tests were conducted to validate the system's performance in various scenarios. The results demonstrated the system's functionality and practical feasibility, highlighting the use of NFTs as digital representations of parking spots. However, some limitations were identified, such as the centralization of transaction management and the reliance on network connectivity, which affected performance in more complex scenarios. Despite these challenges, the system contributes to advancements in the application of blockchain and BDI agents, with future improvements envisioned, such as decentralizing the model and integrating it with other Smart City components.

Keywords: bdi agents; blockchain; multi-agent system; smart parking; digital assets.

LISTA DE FIGURAS

Figura 1 – Diagrama Geral do Sistema. **23**
Figura 2 – Diagrama Uso Imediato. **32**
Figura 3 – Diagrama Compra de Reserva. **33**
Figura 4 – Diagrama Uso de Reserva. **34**
Figura 5 – Diagrama Venda e Compra de Reserva. **35**
Figura 6 – Mensagens por tipo de Conexão. **38**
Figura 7 – Negociações feitas por tipo de Conexão. **39**
Figura 8 – Mensagens Enviadas. **39**
Figura 9 – Taxa de Sucesso de Negociações 1. . . . . . **40**
Figura 10 – Diagrama Agente Banco **46**
Figura 11 – Diagrama Agente Gerente **47**
Figura 12 – Diagrama Agente Motorista **48**

LISTA DE TABELAS

Tabela 1 – Especificações da máquina. **37**
Tabela 2 – Configurações dos Experimentos. **37**
Tabela 3 – Resultados dos experimentos **38**
Tabela 4 – Resultados dos experimentos **50**

LISTAGEM DE CÓDIGOS FONTE

Listagem 1 – Exemplo de requisição Velluscinum **26**
Listagem 2 – Crença do caminho para Servidor **26**
Listagem 3 – Exemplo loadWallet **27**
Listagem 4 – Exemplo walletContent **27**
Listagem 5 – Função auxiliar findToken **28**
Listagem 6 – Exemplo deployToken **28**
Listagem 7 – Exemplo transferToken **28**
Listagem 8 – Exemplo deployNFT **29**
Listagem 9 – Exemplo transferNFT **30**
Listagem 10 – Exemplo stampTransaction **30**
Listagem 11 – Exemplo tokenInfo **30**

LISTA DE ABREVIATURAS E SIGLAS

Siglas

DLT	Tecnologia de Registro Distribuido, do inglês *Distributed Ledger Technologies*
NFT	Token Não-Divisível, do inglês *Non-Fungible Token*

SUMÁRIO

1 INTRODUÇÃO 11

1.1 Objetivos 11

1.2 Justificativa 12

1.3 Organização 13

2 REFERENCIAL TEÓRICO 14

2.1 Agentes e Sistemas Multiagentes 14

2.1.1 Arquitetura BDI 14

2.1.2 Comunicação 15

2.1.3 Protocolos de Negociação 15

2.2 Blockchain 16

2.2.1 Ativos Digitais 17

2.2.2 DLT Autorizada e Não Autorizada 17

2.3 Integração por *Middleware* 17

2.4 Considerações Finais 18

3 TRABALHOS RELACIONADOS 19

3.1 Sistemas Multiagentes e Arquitetura BDI 19

3.2 Aplicações de SMA em Estacionamentos Inteligentes 19

3.3 Blockchain e Negociação de Ativos Digitais 19

3.4 Propostas Integradoras 20

3.5 Lacunas de Pesquisa 21

4 DESENVOLVIMENTO DO SISTEMA MULTIAGENTE 22

4.1 Agentes 22

4.1.1 Banco 24

4.1.2 Gerente 24

4.1.3 Motorista 24

4.1.4 Criador 25

4.2 Integração Sistema/Velluscinum 25

4.2.1 Load Wallet 25

4.2.2 Wallet Content 27

4.2.3 Deploy Token 27

4.2.4 Transfer Token 27
4.2.5 Deploy NFT 29
4.2.6 Transfer NFT 29
4.2.7 Stamp Transaction 29
4.2.8 Token Info 29
4.3 Artefatos 30
4.3.1 ParkControl 30
4.3.2 Utils 31
4.4 Cenários 31
4.4.1 Uso imediato - Compra 32
4.4.2 Compra de Reserva - Motorista/Gerente 32
4.4.3 Uso de Reserva 32
4.4.4 Venda e Compra de Reserva - Motorista/Motorista 33
4.5 Negociações 34
5 RESULTADOS 36
5.1 Configurações dos Experimentos 36
5.1.1 Métricas Analisadas 36
5.1.2 Parâmetros Variáveis dos Cenários 36
5.1.3 Detalhes Adicionais 37
5.2 Experimentação 37
5.3 Análise e Comparação 38
6 CONCLUSÃO 41
REFERÊNCIAS 43
APÊNDICE A DIAGRAMAS DOS AGENTES 49

1 INTRODUÇÃO

A crescente dependência da sociedade em sistemas inteligentes descentralizados tem transformado significativamente os hábitos cotidianos. Dispositivos como *smartphones*, *wearables*, veículos e eletrodomésticos têm evoluído em suas capacidades computacionais e de comunicação, influenciando diretamente as interações humanas e o modo de vida (CALVARESI *et al.*, 2019). Nesse cenário, os sistemas multiagentes emergem como uma solução promissora para problemas complexos, especialmente aqueles que requerem colaboração entre agentes autônomos.

Na área de Inteligência Artificial, os Sistemas Multiagentes (SMA) buscam criar entidades autônomas capazes de atuar de forma adaptativa em ambientes dinâmicos. Esses ambientes frequentemente incluem múltiplos agentes, muitas vezes projetados por diferentes partes e operando em redes distribuídas (PAPI; HüBNER; de Brito, 2022). Nesse contexto, a teoria dos agentes BDI (*Belief-Desire-Intention*) (BRATMAN, 1987) desempenha um papel central, sendo inspirada na psicologia cognitiva. O modelo BDI descreve agentes que possuem crenças (*Beliefs*) sobre o ambiente, desejos (*Desires*) a serem alcançados e intenções (*Intentions*) que orientam suas ações. Essa abordagem tem sido amplamente aplicada tanto na modelagem de processos cognitivos humanos quanto em sistemas multiagentes distribuídos.

Com o crescimento de ativos digitais, como criptomoedas e tokens, novas oportunidades surgem para negociação entre agentes autônomos. Esses ativos oferecem um meio descentralizado e seguro para representar e transferir valor, eliminando intermediários tradicionais. No entanto, a integração de ativos digitais em sistemas multiagentes ainda é um campo pouco explorado, mas promissor, especialmente para aplicações em transações automatizadas (OLIVEIRA; ALVES; LAZARIN, 2023).

Os SMA proporcionam um conjunto robusto de ferramentas para o desenvolvimento de aplicações que envolvem negociações de ativos digitais (PAPI; HüBNER; de Brito, 2022). Neste trabalho, serão empregadas técnicas de modelagem BDI, análise de protocolos de negociação e mecanismos como o sistema de grau de confiança entre agentes (CASTRO; ALVES; BORGES, 2017) e o Contract Net Protocol (CNP) (SMITH, 1980), amplamente reconhecido como uma abordagem eficaz para precificação em Smart Parking (ALVES *et al.*, 2019). Para garantir a segurança e a descentralização das requisições na *blockchain*, será adotado o *middleware* Velluscinum (LAZARIN *et al.*, 2023) para execução de transações digitais. Casos de uso reais e aplicações práticas também serão investigados, culminando no desenvolvimento de um sistema para negociação de ativos digitais em ambientes de Smart Parking.

1.1 Objetivos

O objetivo geral do trabalho é estruturar e implementar um sistema multiagentes BDI integrado a um ambiente de Smart Parking, no qual os agentes realizam negociações por meio

do *middleware* Velluscinum com foco na viabilização de negociações descentralizadas e seguras de ativos digitais. Para alcançar esse propósito, os seguintes objetivos específicos foram definidos:

- Analisar negociações de agentes que utilizam ativos digitais em sistemas similares.
- Definir mecanismos de negociação entre os agentes que sejam compatíveis com o uso de ativos digitais.
- Implementar a arquitetura BDI nesses mecanismos de negociação.
- Comparar diferentes mecanismos de negociação.
- Realizar testes para validar o sistema implementado.

1.2 Justificativa

Como observado por Zanella et al. (2014, tradução nossa), "embora ainda não exista uma definição clara e formalmente aceita para o conceito de Cidade Inteligente, o principal objetivo é maximizar o uso dos recursos públicos, aprimorar a qualidade dos serviços oferecidos aos cidadãos e, simultaneamente, reduzir os custos operacionais da administração pública". Nesse escopo, surgem aplicações em menor escala, como o Smart Parking, que possibilita um gerenciamento mais eficiente das vagas de estacionamento. Esse sistema permite monitoramento em tempo real, otimização do tempo necessário para estacionar e até mesmo a reserva antecipada de vagas, proporcionando mais comodidade aos motoristas.

A implementação de um sistema de *Smart Parking* pode ser aprimorada com o uso de sistemas multiagentes, nos quais agentes inteligentes tomam decisões com base em informações disponíveis e no ambiente ao seu redor. No caso de agentes baseados no modelo BDI (Beliefs, Desires, Intentions), essas informações são estruturadas em crenças, desejos e intenções, permitindo que os agentes tomem decisões de forma mais racional e adaptativa.

Quando essa abordagem é integrada a tecnologias inovadoras, como a *blockchain*, que oferece transações seguras e transparentes, o potencial para soluções mais eficazes e confiáveis torna-se evidente. A combinação dessas tecnologias possibilita melhorias significativas tanto na gestão operacional quanto na experiência do usuário.

Buscando preencher uma lacuna na pesquisa atual sobre sistemas multiagentes BDI que incorporam ativos digitais (OLIVEIRA; ALVES; LAZARIN, 2023). Este trabalho propõe a integração de agentes inteligentes BDI com a tecnologia *blockchain* para aumentar a eficiência e a transparência em sistemas de estacionamento inteligente.

Neste trabalho, foram empregadas diversas ferramentas que foram fundamentais para o desenvolvimento do Sistema Multiagente. Entre elas, destaca-se o Velluscinum, um *middleware* utilizado para estabelecer a comunicação com a *blockchain*, através do BigchainDB (GMBH,

2018). Além disso, foi utilizado o *framework* JaCaMo (BOISSIER; HüBNER; RICCI, 2016), que permite a programação e execução do Sistema Multiagente. Vale ressaltar que a execução do JaCaMo requer a instalação do ambiente Java, essencial para seu funcionamento adequado.

1.3 Organização

Este documento está organizado em capítulos que exploram os diferentes aspectos da pesquisa e implementação deste trabalho. O Capítulo 2 apresenta a fundamentação teórica, detalhando conceitos sobre sistemas multiagentes, arquitetura BDI e ativos digitais, além de explorar tecnologias relacionadas, como *blockchain* e o *middleware* Velluscinum. No Capítulo 3 descreve os trabalhos relacionados, analisando pesquisas anteriores e casos de uso que inspiraram e guiaram o desenvolvimento deste projeto. No Capítulo 4, detalha-se o processo de desenvolvimento do sistema multiagente, incluindo a arquitetura implementada, os mecanismos de negociação e a integração com o *middleware* Velluscinum. O Capítulo 5 apresenta os resultados obtidos, avaliando o desempenho do sistema em cenários simulados e comparando os diferentes cenários de negociação testados. Por fim, o Capítulo 6 encerra o trabalho com as conclusões, destacando as contribuições realizadas, as limitações encontradas e sugestões para futuras pesquisas.

2 REFERENCIAL TEÓRICO

Este capítulo mostra todo o referencial teórico necessário para o desenvolvimento deste trabalho. Na Seção 2.1 é apresentado os conceitos de agentes inteligentes, explicando seu uso e suas definições, apresentando também a arquitetura BDI que será usada ao decorrer do documento e alguns outros conceitos como protocolos de negociação. A Seção 2.2 mostra uma base sobre *Blockchain*, apontando propriedades e versatilidade que pode-se alcançar com esta tecnologia. Por último, a Seção 2.3 descreve o que é um *middleware* e como ele pode ser utilizado para integração entre *blockchain* e Sistemas Multiagentes.

2.1 Agentes e Sistemas Multiagentes

Um agente inteligente é um sistema computacional autônomo que opera de forma independente para atingir seus objetivos, eliminando a necessidade de instruções explícitas (WOOLDRIDGE, 2009). Em um sistema multiagente, vários desses agentes coexistem, interagindo entre si e facilitando a troca de informações por meio de uma infraestrutura de rede. Esse paradigma promove uma abordagem descentralizada, permitindo que os agentes tomem decisões autonomamente e contribuam para um ambiente dinâmico de cooperação ou competição.

No âmbito da Inteligência Artificial, e mais especificamente nos Sistemas Multiagentes (SMA), a ênfase está no desenvolvimento de entidades computacionais autônomas capazes de ações flexíveis e independentes em contextos dinâmicos, muitas vezes envolvendo agentes desenvolvidos por diferentes partes e distribuídos em uma rede (PAPI; HüBNER; de Brito, 2022).

2.1.1 Arquitetura BDI

Dentro dos Sistemas Multiagentes, a teoria dos agentes BDI (*Belief-Desire-Intention*), inspirada na psicologia cognitiva, ganha destaque (BRATMAN, 1987). Essa teoria descreve as atitudes mentais dos agentes, incluindo crenças (*Beliefs*), desejos (*Desires*) e intenções (*Intentions*). O modelo BDI, originalmente utilizado para modelar a cognição humana, é aplicado com sucesso em ambientes de Sistemas Multiagentes (RAO, 1996).

Esse tipo de agente mantém um conjunto de crenças sobre o mundo ao seu redor. Crenças que são informações que o agente acredita serem verdadeiras, baseadas em observações, experiências ou comunicação com outros agentes e elas podem ser atualizadas dinamicamente à medida que ele interage com o ambiente, permitindo que o agente ajuste suas crenças com base em novas informações ou mudanças nas circunstâncias.

O agente BDI também possui objetivos, ou metas, que expressam suas preferências e intenções. Esses desejos representam o que o agente aspira alcançar ou realizar em um determinado contexto. Os desejos do agente podem ser organizados em uma hierarquia, refle-

tindo a importância relativa de diferentes metas, que permite que o agente tome decisões mais ponderadas quando confrontado com conflitos entre desejos.

Quando o agente decide agir para atingir um determinado objetivo, ele forma uma intenção que envolve a seleção de um plano de ação ou estratégia para realizar a meta desejada. As intenções do agente podem ser adaptadas dinamicamente com base em mudanças no ambiente ou na disponibilidade de recursos e caso as circunstâncias mudarem, o agente pode ajustar sua intenção para continuar a buscar seus objetivos.

2.1.2 Comunicação

As capacidades de negociação e argumentação são fundamentais para a habilidade de um agente em alcançar um acordo (WOOLDRIDGE, 2009). Em sistemas multiagentes baseados no modelo BDI, a comunicação desempenha um papel crucial para facilitar a colaboração eficiente entre os agentes autônomos. A troca de informações é fundamental para a construção e atualização das crenças de cada agente sobre o ambiente compartilhado. Os agentes comunicam suas intenções e desejos, permitindo uma compreensão mútua dos objetivos individuais e coletivos. Além disso, a comunicação desempenha um papel importante na resolução de conflitos, pois os agentes podem compartilhar suas intenções e, assim, coordenar suas atividades para evitar interferências ou colisões de objetivos.

A comunicação em sistemas multiagentes BDI também é essencial para a adaptação dinâmica. À medida que as crenças dos agentes evoluem com a aquisição de novas informações, a comunicação permite a propagação eficiente dessas atualizações, garantindo uma visão comum do estado do ambiente (WOOLDRIDGE, 2009). Estratégias de comunicação eficazes, como a argumentação, podem ser empregadas para persuadir outros agentes a adotarem determinadas crenças ou ações. Em suma, a comunicação em sistemas multiagentes BDI é um componente-chave para promover a cooperação, resolver conflitos e garantir uma coordenação efetiva em ambientes dinâmicos e complexos.

Conflitos e incertezas podem ocorrer nesta relação, sendo essencial garantir mecanismos para que todos os agentes cooperem, assegurando a confiabilidade na informação adquirida e nos recursos utilizados (DEMAZEAU, 1996).

2.1.3 Protocolos de Negociação

Um problema óbvio, relacionado à questão da cooperação, é o de alcançar acordos em uma sociedade de agentes auto-interessados. No mundo multiagente em que todos nós habitamos todos os dias, frequentemente somos obrigados a interagir com outros indivíduos com os quais pode ser que não compartilhemos objetivos comuns (WOOLDRIDGE, 2009).

Protocolos de negociação entre agentes inteligentes são conjuntos de regras e procedimentos que regulam a interação entre entidades autônomas durante o processo de negociação. Esses protocolo desempenham um papel crucial na facilitação de acordos e na tomada de decisões, permitindo que os agentes alcancem acordos mutuamente benéficos em ambientes dinâmicos.

A complexidade da negociação entre agentes inteligentes é influenciada por diversos fatores. Os objetivos específicos de cada agente ou comunidade de agentes podem não ser compatíveis, surgindo assim a necessidade de estratégias de negociação eficientes. Além disso, as dependências entre as tarefas e a presença de informações incompletas podem impactar significativamente o processo de negociação, requerendo adaptação contínua (ALVES *et al.*, 2019).

Essa complexidade é principalmente dependente de três parâmetros: Objetivos, cada entidade (agente) ou comunidade de entidades tem objetivos específicos, que podem ou não ser compatíveis; Dependências de Tarefas, a execução de tarefas é inter-relacionada e no processo de negociação devem ser consideradas as dependências e precedências entre as tarefas; Informações incompletas, em alguns casos é necessário que ocorra a negociação sem saber de toda informação disponível sobre um problema (ALVES *et al.*, 2019).

2.2 Blockchain

A *blockchain* é um sistema descentralizado e transparente que possibilita transações seguras e imutáveis, tornando-se uma solução atraente para questões de confiança e integridade de dados em ambientes digitais (KUMAR *et al.*, 2023). Esse sistema utiliza a tecnologia de Tecnologia de Registro Distribuido, do inglês *Distributed Ledger Technologies* (DLT), onde as informações são compartilhadas e validadas por uma rede de participantes, eliminando a necessidade de uma autoridade central. A segurança e a autenticidade das transações são garantidas por algoritmos criptográficos, tornando praticamente impossível a adulteração dos dados. A criptografia, por meio de métodos como funções hash, assinaturas digitais e criptografia de chave pública, é essencial para proteger a integridade e a confidencialidade das informações na *blockchain*. As funções hash, por exemplo, geram saídas de comprimento fixo que funcionam como uma impressão digital, garantindo a autenticidade dos dados (KUMAR *et al.*, 2023).

Além disso, os mecanismos de consenso são responsáveis por garantir que todos os participantes da rede cheguem a um acordo sobre o estado atual do registro, promovendo a confiança no sistema. Os contratos inteligentes, por sua vez, são programas autônomos que executam automaticamente cláusulas previamente acordadas assim que as condições estipuladas são atendidas, proporcionando automação e transparência aos processos. Essas características tornam a *blockchain* uma tecnologia fundamental para diversas aplicações, desde transações financeiras seguras até cadeias de suprimentos transparentes e contratos digitais confiáveis.

2.2.1 Ativos Digitais

A ascensão dos ativos digitais, como criptomoedas e Token Não-Divisível, do inglês *Non-Fungible Token* (NFT), têm proporcionado novas possibilidades para a negociação entre agentes autônomos. Esses ativos digitais permitem a representação e transferência de valor de forma descentralizada e segura, sem a necessidade de intermediários tradicionais (MINARSCH *et al.*, 2022). Ativos digitais em uma DLT, como *blockchain*, referem-se a representações digitais de valor ou propriedade que são registradas e gerenciadas de forma descentralizada. Esses ativos são frequentemente tokenizados e registrados em uma *blockchain*, permitindo a transferência segura e transparente de propriedade. Eles podem abranger uma ampla variedade de ativos, desde criptomoedas como Bitcoin e Ethereum até ativos tokenizados, como títulos, imóveis e arte.

Ativos digitais podem ser categorizados como divisíveis ou indivisíveis. Ativos divisíveis, como criptomoedas, podem ser fracionados em unidades menores, permitindo transações precisas em frações, por exemplo, um Bitcoin pode ser dividido em unidades menores, como satoshis. Em contraste, ativos indivisíveis, como títulos ou certificados de propriedade, não podem ser fracionados, cada unidade representa a totalidade do ativo, e a transferência de propriedade ocorre na unidade completa. A capacidade de representar ativos de ambas as formas contribui para a versatilidade e amplitude de aplicações da DLT em diversos setores econômicos.

2.2.2 DLT Autorizada e Não Autorizada

Uma DLT pode ser classificada em duas categorias de controle de acesso, sendo elas autorizada (*permissioned*) ou não autorizada (*permissionless*) (RAJASEKARAN; AZEES; ALTURJMAN, 2022). No contexto *permissioned*, o acesso à rede e as permissões para validar transações são concedidos apenas a usuários autorizados, resultando em maior controle e privacidade. Por outro lado, a DLT *permissionless* abraça uma abordagem completamente descentralizada, permitindo que qualquer pessoa participe, valide transações e contribua para o consenso. Exemplificado pelo *blockchain* do Bitcoin, esse modelo promove a inclusão aberta e a ausência de controle central, oferecendo transparência e autonomia para os participantes. Cada abordagem atende a diferentes necessidades e contextos, com as DLTs com autorizadas enfocando controle e segurança em ambientes específicos, e as DLTs não autorizadas abraçando a descentralização total e a acessibilidade generalizada.

2.3 Integração por *Middleware*

No contexto dos sistemas multiagentes, onde entidades autônomas interagem para atingir objetivos comuns, a importância do *middleware* se destaca. Essa categoria de software,

conhecida como *middleware*, assume um papel fundamental ao facilitar a comunicação e a colaboração entre agentes autônomos em um ambiente distribuído. É notável que sistemas distribuídos frequentemente adotam o *middleware* para suportar a heterogeneidade de computadores e redes, oferecendo uma visão coesa do sistema. Essa camada intermediária, situada entre usuários e aplicativos de um lado e sistemas operacionais do outro, desempenha um papel crucial na facilitação eficiente das interações entre agentes autônomos, proporcionando integração e compatibilidade em sistemas multiagentes (TANENBAUM; STEEN, 2007).

Portando, considerando os conceitos de SMA e *blockchain*, é valido a busca de algum *middleware* que faça o intermédio desses dois componentes, fazendo com que seja possível uma comunicação de agentes autônomos utilizando esse *middleware*. O Velluscinum (LAZARIN *et al.*, 2023) é um *middleware* que está presente neste âmbito, buscando uma proposta de integrar agentes BDI com a tecnologia de registros distribuídos (DLT), que compõe a *blockchain* e permite que os agentes possam criar e transferir ativos digitais e também gerenciar carteiras digitais em uma estrutura autorizada da *blockchain* utilizando BigchainDB.

2.4 Considerações Finais

Este capítulo apresentou o referencial teórico fundamental para o desenvolvimento deste trabalho, abordando conceitos essenciais para a compreensão da integração entre agentes inteligentes e *blockchain* por meio de um *middleware*. Inicialmente, discutimos os Sistemas Multiagentes e a arquitetura BDI, destacando sua capacidade de tomada de decisão autônoma baseada em crenças, desejos e intenções. Também exploramos a importância da comunicação entre agentes, enfatizando os protocolos de negociação como ferramentas fundamentais para interações eficientes em um ambiente distribuído.

Na sequência, foi discutida a tecnologia *blockchain*, suas propriedades de segurança e transparência, bem como sua aplicação na representação e transferência de ativos digitais. A distinção entre DLTs autorizadas e não autorizadas foi apresentada, evidenciando os diferentes níveis de controle e acessibilidade que essa tecnologia pode proporcionar. Por fim, foi explorado o conceito de *middleware*, ressaltando seu papel na integração entre agentes inteligentes e *blockchain*, permitindo a comunicação eficiente entre essas tecnologias.

Este trabalho tem como proposta a implementação de um sistema que une agentes inteligentes e *blockchain* por meio de um *middleware*, viabilizando interações seguras, autônomas e descentralizadas. A abordagem visa otimizar processos de negociação e tomada de decisão em ambientes multiagentes, garantindo maior confiabilidade e rastreabilidade das transações.

3 TRABALHOS RELACIONADOS

Neste capítulo, serão apresentados os principais conceitos e estudos relacionados ao desenvolvimento deste trabalho. Na seção 3.1, será explorada a arquitetura BDI e sua aplicação em Sistemas Multiagentes (SMA). Em seguida, na seção 3.2, serão discutidas soluções baseadas em SMA no contexto de Smart Parking. A seção 3.3 abordará como a integração de *blockchain* com SMA potencializa transações e negociações entre agentes. Por fim, nas seção 3.4 e seção 3.5, serão discutidas iniciativas recentes que combinam esses elementos, bem como as oportunidades e os desafios ainda existentes, justificando a proposta deste trabalho.

3.1 Sistemas Multiagentes e Arquitetura BDI

Sistemas Multiagentes são amplamente utilizados para a resolução de problemas complexos em ambientes dinâmicos. Eles consistem em múltiplos agentes autônomos que interagem e colaboram entre si para atingir objetivos comuns ou individuais. Dentre as arquiteturas propostas para modelar esses agentes, destaca-se a abordagem BDI (*Belief-Desire-Intention*), que se baseia na cognição humana para representar crenças, desejos e intenções dos agentes (BRATMAN, 1987). Estudos recentes, como os de Calvaresi et al. (2019), reforçam que a combinação de SMA com tecnologias emergentes, como *blockchain*, pode fornecer soluções inovadoras e eficazes para diversos domínios.

3.2 Aplicações de SMA em Estacionamentos Inteligentes

No contexto de Cidades Inteligentes, a aplicação de SMA em sistemas de Smart Parking tem sido uma área de pesquisa promissora. Souza et al. (2017) propuseram o uso de agentes com graus de confiança para alocação de vagas de estacionamento, destacando sua eficácia na melhora da experiência do usuário. De forma complementar, Alves et al. (2019) exploraram o uso do protocolo de negociação CNP (*Contract Net Protocol*) para resolver problemas de consenso em sistemas de Smart Parking, evidenciando que este protocolo é altamente eficiente em termos de precificação.

3.3 Blockchain e Negociação de Ativos Digitais

A ascensão da tecnologia *blockchain* tem possibilitado novas formas de representação e transação de valor em sistemas distribuídos. Estudos como os de Papi et al. (2022) demonstram que a integração de *blockchain* com SMA pode oferecer maior transparência e segurança em transações de ativos digitais. No contexto de negociação entre agentes, a utilização de

middlewares específicos, como o Velluscinum, desenvolvido por Lazarin et al. (2023), tem se mostrado uma solução robusta para implementar essas interações em sistemas multiagentes.

3.4 Propostas Integradoras

Trabalhos recentes procuram unir as vantagens da arquitetura BDI, *blockchain* e SMA em aplicações práticas. O trabalho de Oliveira et al. (2023) demonstra a realização um mapeamento sistemático visando identificar lacunas de pesquisa e reunir conteúdos sobre a utilização de ativos digitais na negociação entre agentes. Este trabalho busca e analisa artigos dentro desse escopo trazendo os seguintes artigos que foram analisados:

- "A Blockchain integration to support transactions of assets in multi-agent systems" (PAPI; HüBNER; de Brito, 2022);
- "Trusted Registration, Negotiation, and Service Evaluation in Multi-Agent Systems throughout the Blockchain Technology" (CALVARESI *et al.*, 2018a);
- "The good, the bad, and the ethical implications of bridging blockchain and multi-agent systems" (CALVARESI *et al.*, 2019);
- "A Distributed Electricity Trading System in Active Distribution Networks Based on Multi-Agent Coalition and Blockchain" (LUO *et al.*, 2019);
- "Enterprise Platform of Logistics Services Based on a Multi-Agents Mechanism and Blockchains" (LIU; HENNEQUIN; ROY, 2021);
- "Trust-Based Smart Contract for Automated Agent to Agent Communication" (MHAMDI *et al.*, 2022);
- "Resilience Network Controller Design for Multi-Domain SDN: A BDI-based Framework" (SONG *et al.*, 2022);
- "Reputation Management in Multi-Agent Systems Using Permissioned Blockchain Technology" (CALVARESI *et al.*, 2018b);
- "Velluscinum: A Middleware for Using Digital Assets in Multi-Agent Systems" (LAZARIN *et al.*, 2023).

Como conclusão, Oliveira et al. (2023) mostra que há diversos estudos recentes exploram a integração de SMA com *blockchain*, mas poucos utilizam DLTs autorizadas em conjunto com agentes BDI. A análise dos artigos revela que a maioria emprega contratos inteligentes para a interação entre agentes, além de uma tendência à descentralização. Trabalhos como os de Calvaresi et al. (2018a, 2019) abordam a integração completa de SMA e *blockchain* com

DLTs autorizadas, enquanto Papi et al. (2022) investiga um modelo centralizado com agentes BDI. O estudo de Lazarin et al. (2023) destaca-se por propor um *middleware* para auxiliar na negociação entre agentes BDI utilizando DLT. No contexto de SMA BDI aplicados a ambientes de *Smart Parking*, identifica-se uma lacuna de pesquisa, a qual motiva o sistema desenvolvido neste trabalho.

3.5 Lacunas de Pesquisa

Apesar dos avanços, algumas lacunas ainda persistem. A maioria dos trabalhos foca em aspectos isolados, como protocolos de negociação ou frameworks de implementação. Não há uma abordagem consolidada que integre completamente as arquiteturas BDI, *blockchain* e SMA com foco em aplicações específicas, como Smart Parking. Além disso, a análise comparativa entre diferentes mecanismos de negociação ainda é limitada, evidenciando a necessidade de estudos mais abrangentes.

4 DESENVOLVIMENTO DO SISTEMA MULTIAGENTE

O Sistema Multiagente BDI foi implementado utilizando o framework JaCaMo, baseado no modelo proposto por Oliveira, Alves e Lazarin (2023), com adaptações em alguns cenários e agentes para alcançar resultados mais eficazes. O sistema abrange cenários que exemplificam a troca de mensagens entre agentes e sua interação com a *blockchain* através do *middleware* Velluscinum. Disponível em [1].

Para a implementação e desenvolvimento do Sistema Multiagente (SMA) proposto, foi necessário o uso de um conjunto de ferramentas que possibilitam desde programação dos agentes até a integração com tecnologias externas, como o *middleware* Velluscinum. Essas ferramentas desempenharam papéis fundamentais em diferentes etapas do desenvolvimento, proporcionando suporte para a simulação, o gerenciamento de transações digitais e a comunicação entre os agentes.

O desenvolvimento do sistema multiagentes foi feito interamente através o framework JaCaMo (BOISSIER; HüBNER; RICCI, 2016) pela distribuição JaCaMo CLI na versão 1.2, disponível em [2], esta versão conta com Jason 3.2, Moise 1.0, SAI 0.5.3 e Cartago 3.1. Também para utilização deste framework é necessário Java 17.

A Figura 1 apresenta uma visão geral do funcionamento do sistema, destacando como ele gerencia negociações de forma segura e eficiente por meio de requisições ao Velluscinum, operando de maneira descentralizada. A figura também ilustra os agentes principais do sistema, detalhados na seção 4.1, pontuando as requisições às quais cada um tem acesso para se comunicar com a *blockchain*. Cada requisição mostrada será explicada futuramente na seção 4.2.

4.1 Agentes

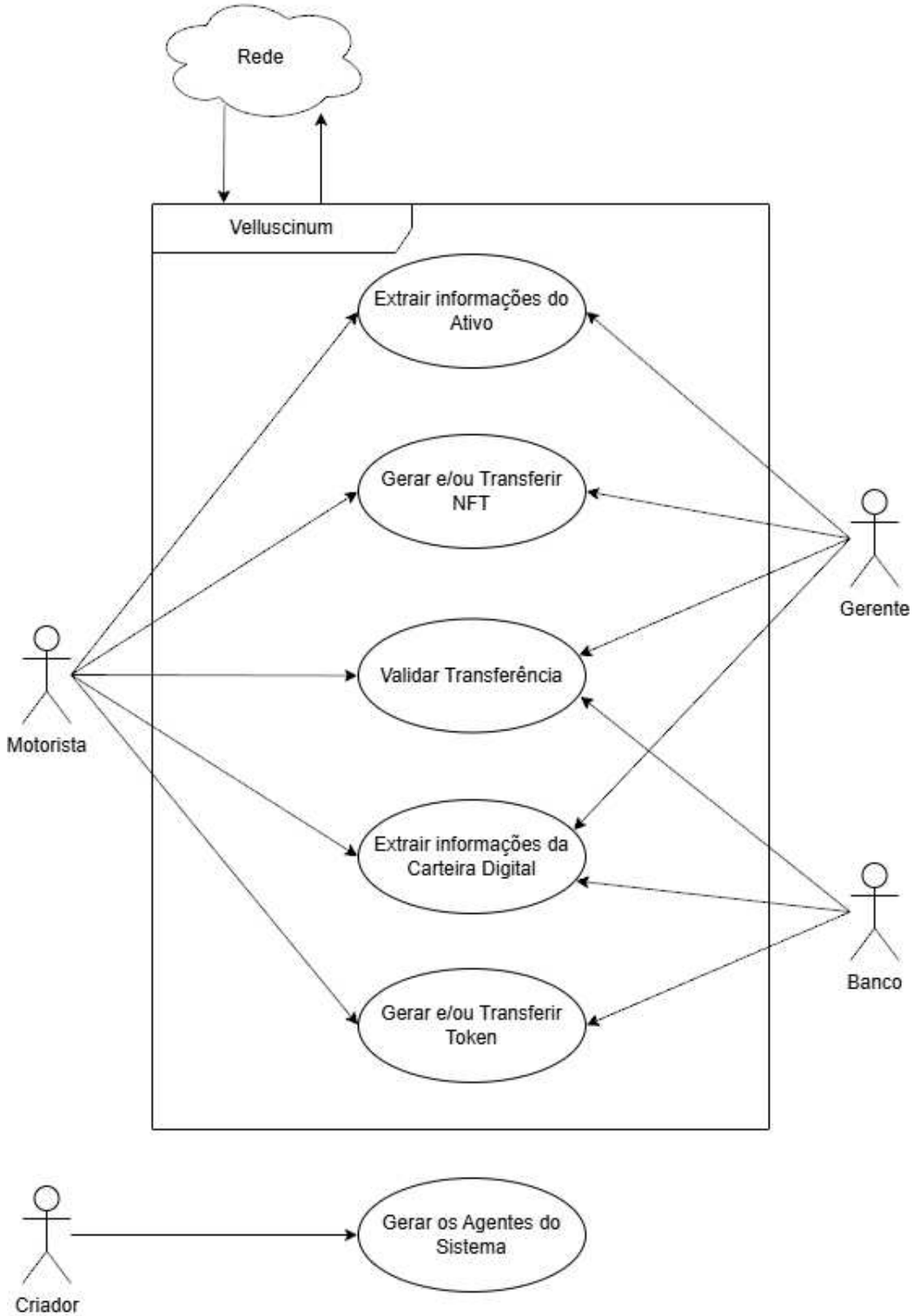
O sistema multiagente é composto por quatro tipos principais de agentes: Banco, Gerente, Motorista e Criador. Cada agente desempenha um papel específico na simulação, contribuindo para o funcionamento do *Smart Parking* descentralizado.

Quando o sistema é iniciado, o criador gera os agentes que estarão presentes e o agente Banco cria a moeda (um token divisível) que será utilizada nas transações, distribuindo-a por meio de empréstimos aos agentes Motoristas. Por sua vez, Motoristas e Gerentes criam ou carregam suas carteiras digitais. O agente Gerente também é responsável pela organização das vagas no estacionamento, representadas como NFTs.

[1] https://github.com/laca-is/smartParking-velluscinum

[2] https://github.com/jacamo-lang/jacamo/releases/tag/v1.2

Figura 1 – Diagrama Geral do Sistema.



Fonte: Autoria própria (2025).

4.1.1 Banco

O agente Banco desempenha um papel fundamental na criação do token divisível que serve como moeda para as transações no sistema. Ele fornece empréstimos aos Motoristas, permitindo que esses agentes realizem negociações envolvendo as vagas do *Smart Parking* e transações entre si, como apresentado na Figura 10, Apêndice A.

Além disso, o Banco mantém uma crença referente à rota para o servidor, essencial para que as requisições realizadas pelo *middleware* Velluscinum funcionem corretamente (LAZARIN *et al.*, 2023). Essa rota é compartilhada com os outros agentes, facilitando a integração com a *blockchain*.

4.1.2 Gerente

O agente Gerente é responsável pela criação dos NFTs que representam as vagas de estacionamento. Cada vaga possui um ID único e um tipo específico, como:

- Vagas curtas;
- Vagas curtas cobertas;
- Vagas longas;
- Vagas longas cobertas.

Além disso, o Gerente executa operações relacionadas à gestão do *Smart Parking*, incluindo a venda de vagas, validação de transferências e criação de reservas. Ele também valida transações e se comunica diretamente com os agentes Motoristas para realizar essas operações como apresentado na Figura 11, Apêndice A.

4.1.3 Motorista

O agente Motorista simula um indivíduo interessado em utilizar o *Smart Parking*. Ele possui diversas ações que refletem diferentes cenários do sistema, como:

- Comprar uma vaga para uso imediato;
- Reservar uma vaga com o Gerente;
- Comprar uma reserva de outro Motorista;
- Vender uma reserva para outro Motorista;
- Utilizar uma reserva armazenada em sua carteira.

Cada Motorista possui crenças específicas e gerencia sua própria carteira digital, permitindo a realização de transações e interações com outros agentes e com o Gerente. Ver Figura 12, Apêndice A.

4.1.4 Criador

O agente Criador é um agente auxiliar responsável por automatizar a criação dos agentes Motoristas. Cada Motorista criado é único, com suas respectivas carteiras digitais e crenças predefinidas, garantindo diversidade e individualidade entre os agentes.

Este conjunto de agentes e suas interações formam a base para o funcionamento do *Smart Parking*, permitindo a simulação de um sistema descentralizado integrado à *blockchain*, conforme descrito nos cenários detalhados.

4.2 Integração Sistema/Velluscinum

O *middleware* Velluscinum oferece aos agentes inteligentes a capacidade de criar e transferir ativos digitais, validar transações e gerenciar suas carteiras diretamente na dimensão do agente (LAZARIN *et al.*, 2023). No contexto do *Smart Parking*, os agentes Banco, Motorista e Gerente utilizam as funcionalidades disponibilizadas pelo Velluscinum, o que assegura a descentralização das negociações que ocorrem no sistema. Essa abordagem possibilita que todos os agentes mantenham pleno conhecimento das transações realizadas, dos valores envolvidos e das validações efetuadas na *blockchain*, promovendo transparência e confiabilidade nas operações.

Para configurar e integrar o Velluscinum ao sistema multiagente desenvolvido, o pacote foi importado no arquivo de configuração do SMA (arquivo .jcm), conforme ilustrado na Listagem 1. Em seguida, foi criada uma crença no agente Banco para armazenar o caminho da porta do servidor, como apresentado na Listagem 2. Essa chave do servidor é então transmitida para todos os agentes do sistema, utilizando o mecanismo de *broadcast*, permitindo que os agentes acessem o *middleware*.

Após a configuração inicial, os agentes conectados ao servidor podem realizar as requisições presentes no *middleware*, os agentes Banco, Motorista e Gerente usufruem das requisições apresentadas nas subseções seguintes.

4.2.1 Load Wallet

Responsável por criar ou carregar a carteira digital do agente, no exemplo mostrado pela Listagem 3 é mostrado que alguns valores são retornados dentro de uma crença pela requisição, tais valores sendo PrK uma chave privada e PuK uma chave pública, ambos vinculados à

Listagem 1 – Exemplo de requisição Velluscinum

```
mas smartParking {
    agent manager: manager.asl {
        focus: network.parkControl, network.utils
    }

    agent creator {
        focus: network.utils
    }

    agent bank {
        focus: network.utils
    }

    workspace network {
        artifact parkControl: ParkControl
        artifact utils: Utils
    }

    uses package: velluscinum "com.github.chon-group:Velluscinum:+"
}

```

Fonte: Autoria própria (2025).

Listagem 2 – Crença do caminho para Servidor

```
{ include("$jacamo/templates/common-cartago.asl") }
{ include("$jacamo/templates/common-moise.asl") }

// chainServer("http://testchain.chon.group:9984/"). // remoto
chainServer("http://localhost:9984/"). // local

...

!carregarCarteira.

+!carregarCarteira: chainServer(Server) <-
    .broadcast(tell, chainServer(Server));

...
```

Fonte: Autoria própria (2025).

carteira do agente, servem para que o agente possa efetuar as transações dentro da *blockchain* através do Velluscinum.

Listagem 3 – Exemplo loadWallet

```
1 +!criarCarteira : not myWallet(PrK,PuK) <-
2     .print("Obtendo carteira digital...");
3     .velluscinum.loadWallet(myWallet);
4     .wait(myWallet(PrK, PuK));
5     +driverWallet(PuK);
```

Fonte: Autoria própria (2025).

4.2.2 Wallet Content

Responsável por extrair os ativos digitais presentes na carteira digital do agente, pelo exemplo mostrado pela Listagem 4, nota-se que uma das ações do agente Motorista, onde ele carrega o conteúdo presente na sua carteira através da requisição, passando por parâmetro a chave do servidor, as chaves, privada e pública (da carteira do agente) e uma crença para que seja retornado os valores pelo Velluscinum. Após isso, o objetivo *findToken*, apresentado na Listagem 5, percorre a lista de conteúdo retornada e preenche as crenças dos ativos presentes na carteira.

Listagem 4 – Exemplo walletContent

```
1 +!obterConteudoCarteira : chainServer(Server) & myWallet(PrK, PuK)
2                 & cryptocurrency(Coin) <-
3     .print("Obtendo conteudo da carteira...");
4     .velluscinum.walletContent(Server, PrK, PuK, content);
5     .wait(content(Content));
6     !findToken(Coin, set(Content));
7     !findToken(nft, set(Content)).
```

Fonte: Autoria própria (2025).

4.2.3 Deploy Token

Responsável por criar um ativo digital divisível (criptomoeda), no caso deste SMA desenvolvido esta requisição é feita apenas pelo agente Banco, criando a moeda usada para as negociações, ele cria esta moeda como mostrado na Listagem 6, passando por parâmetro na requisição a chave do servidor, as chaves da carteira, a descrição para aquele ativo, a quantidade desejada e uma crença para receber o id do ativo criado.

4.2.4 Transfer Token

Responsável por Transferir uma quantidade de tokens para outra carteira digital, a Listagem 7 apresenta um exemplo de como o agente Motorista usa esta requisição para pagar o uso da vaga ao gerente, passando por parâmetro as chaves do servidor e da carteira, o id da moeda

Listagem 5 – Função auxiliar findToken

```
+!findToken(Term,set([Head|Tail])) <-
    !compare(Term,Head,set(Tail));
    !findToken(Term,set(Tail)).

+!compare(Term,[Type, AssetId, Qtd],set(V)) : (Term == AssetId) <-
  +coinBalance(Qtd);
    .print("Saldo atual: ", Qtd).

+!compare(Term,[Type,AssetId,Qtd],set(V)) : (Term == Type)
                & listaNFTs(Lista) <-
    -+listaNFTs([AssetId|Lista]).

+!compare(Term,[Type,AssetId,Qtd],set(V)) : (Term == Type)
                & not listaNFTs(Lista) <-
    .concat(AssetId, Lista);
    +listaNFTs([Lista]).

-!compare(Term,[Type,AssetId,Qtd],set(V)).

-!findToken(Type,set([    ])).
```

Fonte: Autoria própria (2025).

Listagem 6 – Exemplo deployToken

```
+!criarMoeda: chainServer(Server) & myWallet(PrK, PuK) <-
  .print("Criando moeda");
  .velluscinum.deployToken(Server, PrK, PuK, "name:cryptocurrency",
       100000, cryptocurrency);
  .wait(cryptocurrency(Coin)).
```

Fonte: Autoria própria (2025).

que deseja transferir, a chave pública da carteira do destinatário, a quantidade da moeda que deseja tranferir e por fim uma crença para receber o id referente à transação realizada.

Listagem 7 – Exemplo transferToken

```
+!pagarUso(Valor) : cryptocurrency(Coin) & chainServer(Server)
            & myWallet(PrK,PuK) & managerWallet(Manager)
            & coinBalance(Balance)<-
    .print("Pagamento em andamento...");
    ?idVaga(Id);
    .velluscinum.transferToken(Server,PrK,PuK,Coin,Manager,Valor,payment);
    .wait(payment(TransactionId));
    ...
```

Fonte: Autoria própria (2025).

4.2.5 Deploy NFT

Responsável por Criar um ativo digital não divisível, como apresentado pela Listagem 8, que é um exemplo de como um agente Gerente cria um NFT reserva referente a uma vaga do estacionamento. Para essa criação é necessário as chaves do servidor e da carteira do agente, uma string contendo os dados imutáveis do NFT, uma string contendo os dados Mutáveis e uma crença para receber o id do ativo digital criado.

Listagem 8 – Exemplo deployNFT

```
1  +!criarReservaNFT(IdVaga, Data, Tempo,
2          DriverAgent)[source(self)] : chainServer(Server) & myWallet(PrK, PuK)
3      & driverWallet(DriverW)[source(Agente)] & Agente = DriverAgent  <-
4
5      ...
6
7    .concat("idVaga:", IdVaga, ";descricao:reserva de vaga;tipo:",
8            TipoAtual, ";data:", Data, ";duracao:", Tempo, Dados);
9    .concat("descricao:reserva de vaga;tipo:", Descricao);
10   .velluscinum.deployNFT(Server, PrK, PuK, Dados, Descricao, nft);
11   .wait(nft(ReservaId));
```

Fonte: Autoria própria (2025).

4.2.6 Transfer NFT

Responsável por Tranferir um NFT para outra carteira digital, seguindo a mesma função da subseção anterior temos o uso desta requisição, mostrada Listagem 9, para executá-la é semelhante à requisição *transferToken*, portanto é necessário passar os parâmetros de chaves, de servidor e da carteira do agente, o id do NFT, a chave pública da carteira de destino, uma descrição para a transação e uma crença para receber o id da transferência realizada.

4.2.7 Stamp Transaction

Responsável por validar uma transferência feita dentro da rede, o exemplo desta requisição, mostrado pela Listagem 10, mostra que para validar uma transferência é necessário passar as chaves do servidor, a chave privada e pública do agente e o id da transação, esta requisição caso não seja bem sucedida, a tranferência não for validada, o plano do agente falha.

4.2.8 Token Info

Responsável por extrair os dados de um ativo digital requisitado, exemplo mostrado pela Listagem 11, nesta requisição é necessário enviar a chave do servidor, o id do ativo digital que

Listagem 9 – Exemplo transferNFT

```
+!criarReservaNFT(IdVaga, Data, Tempo,
    DriverAgent)[source(self)] : chainServer(Server) & myWallet(PrK, PuK)
    & driverWallet(DriverW)[source(Agente)] & Agente = DriverAgent  <-

    ...

    .print("Enviando reserva: ", ReservaId, " para: ", DriverAgent);

  .concat("reservaId:", ReservaId, ";tipo:", TipoAtual, ";
           data:", Data, ";duracao:", Tempo, DescricaoReserva);
  .velluscinum.transferNFT(Server, PrK, PuK, ReservaId, DriverW,
           DescricaoReserva, transfer);
  .wait(transfer(TransferId));
```

Fonte: Autoria própria (2025).

Listagem 10 – Exemplo stampTransaction

```
+!stampProcess(Transfer)[source(self)] : chainServer(Server)
           & myWallet(PrK,PuK) <-
  .velluscinum.stampTransaction(Server,PrK,PuK,Transfer);
```

Fonte: Autoria própria (2025).

deseja extrair os dados, um parâmetro para qual tipo de dado que é o requerido (podendo ser data, metadata ou all) e uma crença para receber o conteúdo presente no ativo.

Listagem 11 – Exemplo tokenInfo

```
+!liberarVaga(Id) : chainServer(Server) & myWallet(PrK,PuK) <-
  .velluscinum.tokenInfo(Server, Id, metadata, content);
  .wait(content(Content));
```

Fonte: Autoria própria (2025).

4.3 Artefatos

O sistema criado possui alguns artefatos que auxiliam durante a execução do sistema, fazendo algumas funções auxiliares do estacionamento inteligente e utilitários no geral, como por exemplo uma função que contabiliza as mensgens trocadas pelos agentes durante a execução do SMA.

4.3.1 ParkControl

O ParkControl é um artefato desenvolvido para gerenciar vagas de estacionamento e suas reservas no sistema multiagente. Ele auxilia com operações como verificação de disponibilidade, registro e busca de reservas, além do cálculo de valores de uso. No geral, são funciona-

lidades que são feitas no artefato pela facilidade de trabalhar com a estrutura que o Velluscinum retorna as informações sobre os ativos digitais. As principais funcionalidades incluem:

- **Verificação de Compra e Reserva:** valida se uma vaga está disponível para um tipo específico e se há conflitos de datas com reservas existentes.
- **Registro e Busca de Reservas:** permite registrar novas reservas e localizar reservas existentes, associando informações relevantes como identificadores e períodos de uso.
- **Consulta de Informações de Vagas:** acessa e exibe o status e o tipo de uma vaga.
- **Cálculo de Valores:** determina o custo proporcional do uso da vaga com base no tempo e nas regras de precificação.

A integração com métodos auxiliares e manipulação de dados estruturados possibilita um controle eficiente e flexível, suportando a automação do gerenciamento de vagas.

4.3.2 Utils

O Utils é um artefato criado para oferecer utilitários auxiliares para o gerenciamento de operações no sistema multiagente, incluindo contagem de mensagens, conversão de strings para números, e cálculo de tempo de estacionamento baseado em orçamento. Suas principais funcionalidades são:

- **Contador de Mensagens:** registra e exibe periodicamente o número de mensagens trocadas pelos agentes, utilizando um agendador interno que imprime o valor a cada 20 segundos.
- **Conversão de String para Número:** converte uma string para um número de ponto flutuante, com tratamento de erros para entradas inválidas.
- **Cálculo de Tempo Estacionado:** gera um tempo aleatório de uso (mínimo de 20 minutos) e verifica se o custo estimado não excede o orçamento máximo fornecido, considerando a tarifa por hora.

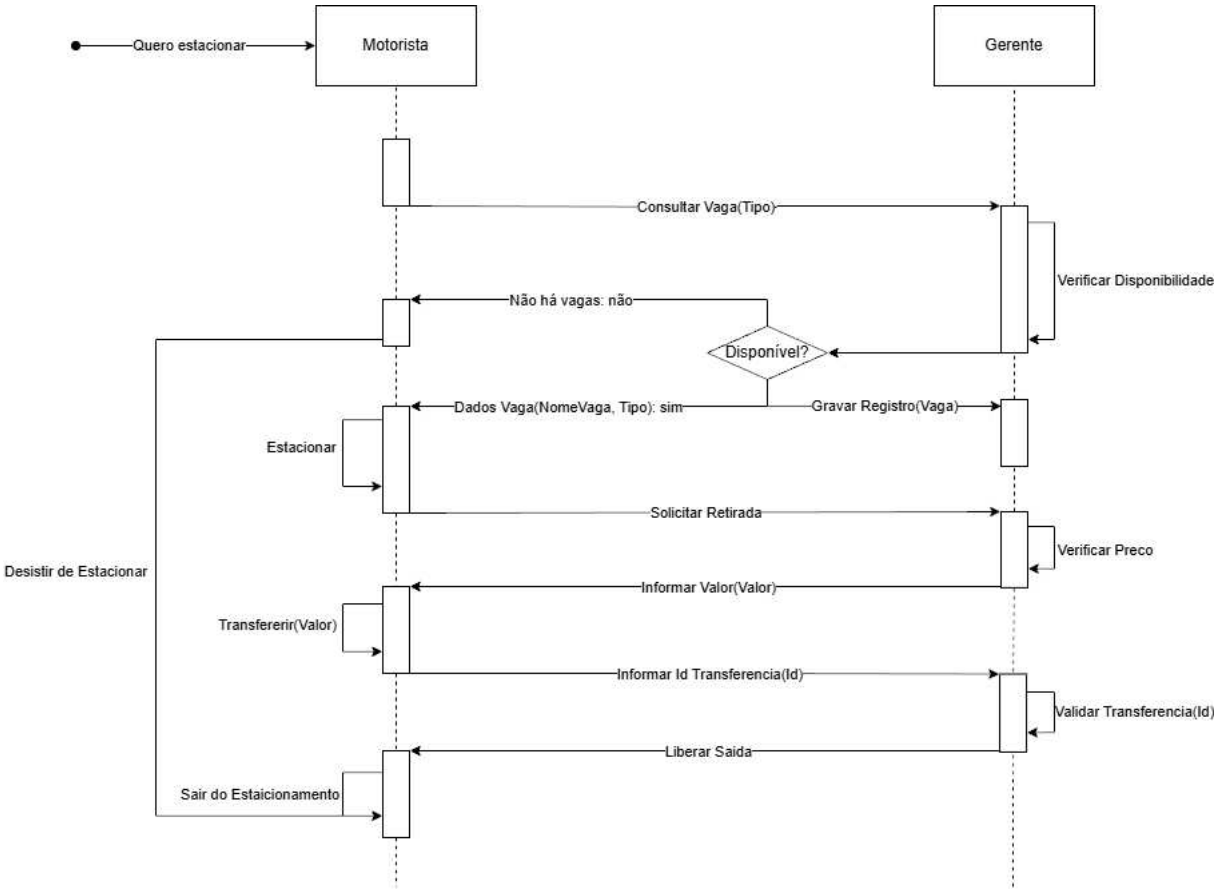
4.4 Cenários

Os cenários abordados pelo sistema englobam comunicações e transações típicas de um *Smart Parking*, simulando interações entre motoristas, interessados em estacionar seus veículos, e gerentes, focados em vender e reservar vagas.

4.4.1 Uso imediato - Compra

Conforme ilustrado na Figura 2, este cenário retrata a situação em que o motorista deseja estacionar seu veículo de forma imediata. A partir dessa necessidade, são realizadas as comunicações e transações necessárias para viabilizar o processo de maneira ágil e eficiente.

Figura 2 – Diagrama Uso Imediato.



Fonte: Adaptado de Oliveira, Alves e Lazarin (2023).

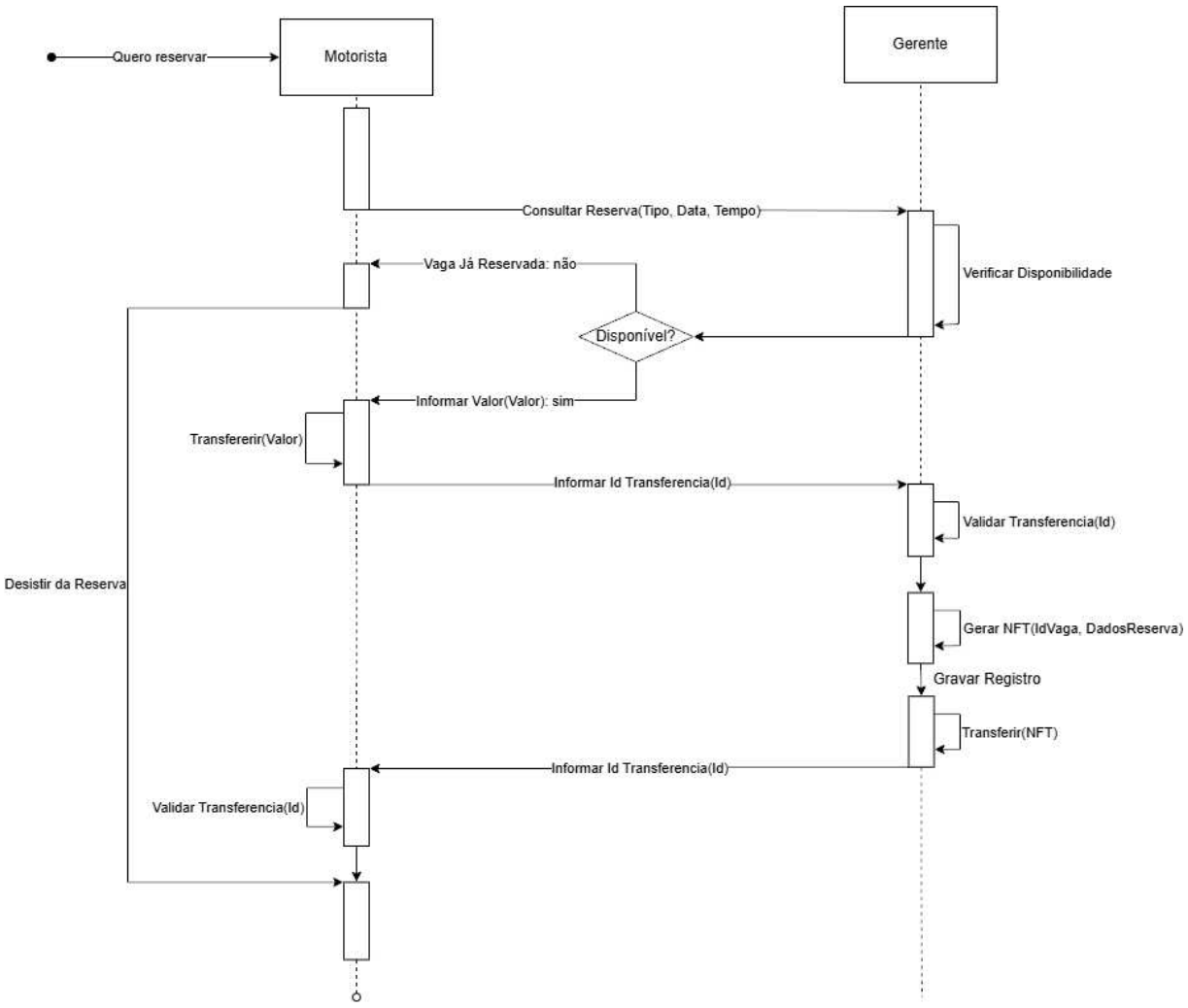
4.4.2 Compra de Reserva - Motorista/Gerente

Neste segundo cenário, representado pela Figura 3, o agente Motorista manifesta o interesse em reservar uma vaga para uso futuro. O processo envolve a compra da reserva, com a emissão e transferência de um NFT referente à ela como comprovante digital de posse e validação da reserva.

4.4.3 Uso de Reserva

O terceiro cenário apresenta uma das duas opções disponíveis para o agente Motorista ao possuir uma reserva de estacionamento no SMA. Neste caso, o motorista utiliza o NFT

Figura 3 – Diagrama Compra de Reserva.



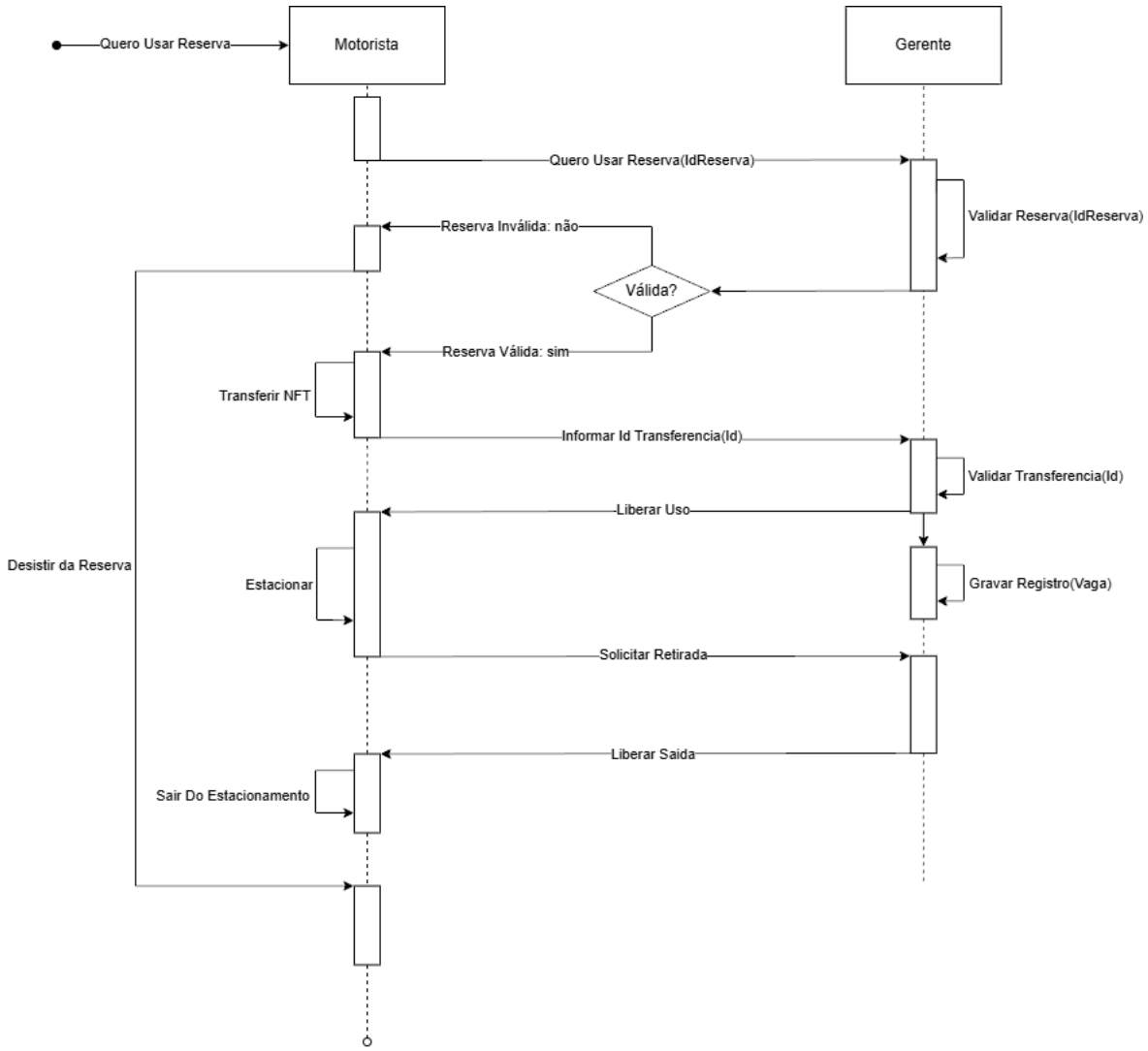
Fonte: Adaptado de Oliveira, Alves e Lazarin (2023).

adquirido como comprovante da reserva para acessar a vaga. A Figura 4 ilustra o motorista manifestando sua intenção de utilizar a reserva e descreve todo o procedimento realizado para concluir o processo.

4.4.4 Venda e Compra de Reserva - Motorista/Motorista

Por fim, este cenário explora o segundo uso possível para o NFT de reserva adquirido por um agente Motorista: a venda para outro motorista. Conforme ilustrado na Figura 5, trata-se de uma negociação direta entre dois motoristas, onde um deseja vender sua reserva enquanto o outro demonstra interesse em comprá-la. Este caso evidencia de forma clara as vantagens do uso da *blockchain*, destacando a segurança e a transparência de transações realizadas sem a necessidade de agentes intermediários.

Figura 4 – Diagrama Uso de Reserva.



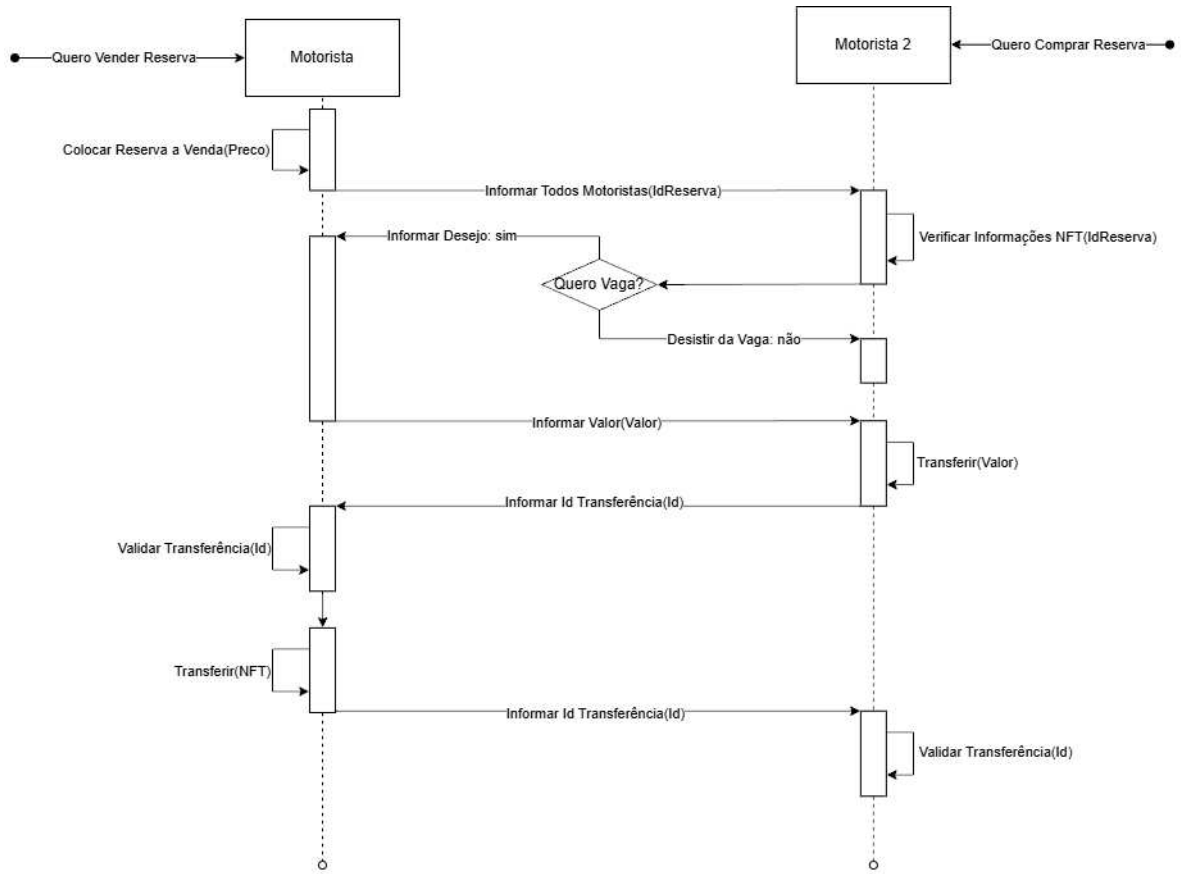
Fonte: Adaptado de Oliveira, Alves e Lazarin (2023).

4.5 Negociações

As negociações no SMA desenvolvido ocorrem de forma híbrida, sendo separadas em duas modalidades, dependendo do caso analisado. A primeira modalidade abrange cenários em que o Motorista deseja utilizar a vaga diretamente, conforme apresentado na Seção 4.4.1, ou quando ele pretende realizar uma reserva com o Gerente ou utilizar uma vaga do estacionamento, conforme descrito nas Seções 4.4.2 e 4.4.3. Essas situações caracterizam um processo de negociação centralizado, no qual o Gerente detém o poder de criar e transferir os NFTs correspondentes às vagas, sendo indispensável para que o processo de negociação aconteça.

A segunda modalidade, por sua vez, caracteriza-se como descentralizada, destacando a interação direta entre os agentes Motoristas nas transações de compra e venda de reservas já em posse de um Motorista, conforme discutido na Seção 4.4.4. Assim, embora apenas o Gerente tenha autoridade para executar ações diretamente relacionadas ao estacionamento, o

Figura 5 – Diagrama Venda e Compra de Reserva.



Fonte: Adaptado de Oliveira, Alves e Lazarin (2023).

Motorista, ao adquirir um NFT referente ao estacionamento, passa a ter a capacidade de realizar ações no mercado.

Em particular, a venda de reservas, representada na Figura 5, exemplifica como o uso de tecnologias como *blockchain* e NFTs proporcionam simplicidade, segurança e transparência. Neste processo, o agente Motorista que detém uma reserva manifesta sua intenção de vendê-la, enquanto outro agente interessado em adquiri-la inicia a negociação diretamente, sem a necessidade de intermediários. A transação ocorre mediante validação e transferência do NFT correspondente, que atua como um comprovante digital da posse e validação da reserva, assegurando a confiabilidade e rastreabilidade do processo.

5 RESULTADOS

Nesta seção, são descritos os cenários experimentais utilizados para validar o sistema multiagente proposto. Apresentam-se os parâmetros de configuração e os resultados obtidos, com ênfase na análise das interações entre agentes BDI em um ambiente de *Smart Parking*, utilizando o *middleware* Velluscinum e ativos digitais.

5.1 Configurações dos Experimentos

Os experimentos foram conduzidos exclusivamente no SMA, monitorando as comunicações e transações realizadas pelos agentes. O tempo foi contabilizado a partir da execução do SMA. Processos como a criação de carteiras digitais, moedas (tokens divisíveis) e vagas no *Smart Parking* foram realizados previamente por um sistema externo, acessível também através do repositório disponível em [1].

5.1.1 Métricas Analisadas

Para avaliar o desempenho do sistema, foram utilizadas as seguintes métricas principais:

- **Mensagens trocadas:** Número total de mensagens enviadas entre os agentes durante os experimentos.
- **Transações realizadas:** Quantidade de transações processadas pelo *middleware* Velluscinum.
- **Taxa de sucesso:** Contagem de negociações concluídas com êxito em relação às negociações iniciadas.
- **Tempo de execução:** Tempo de execução do sistema.

5.1.2 Parâmetros Variáveis dos Cenários

Os experimentos foram configurados para permitir ajustes nos seguintes parâmetros:

- **Número de motoristas:** Quantidade de agentes motoristas simulados.
- **Número de vagas:** Total de vagas disponíveis no *Smart Parking*.
- **Cenários simulados:** Conjunto de casos possíveis que definem os comportamentos e interações dos motoristas.

[1] https://github.com/laca-is/smartParking-velluscinum

- **Conexão:** Configuração de servidor local ou público do BigchainDB fornecido pelo Velluscinum para a execução das requisições na *blockchain*.

5.1.3 Detalhes Adicionais

Os experimentos foram realizados em um ambiente controlado para assegurar a consistência nos resultados. Alterações nos parâmetros dos cenários permitiram avaliar a escalabilidade do sistema e o desempenho da comunicação com o *middleware* Velluscinum frente a diferentes configurações. Os experimentos foram executados em uma máquina com as configurações apresentadas na Tabela 1.

Tabela 1 – Especificações da máquina.

Componentes	**Especificações**
Processador	Intel(R) Core(TM) i5-12400F
Placa de Vídeo	NVIDIA GeForce RTX 2060 Super
Memória RAM	16GB
SSD	512GB
Sistema Operacional	Windows 11 com WSL
Versão Linux do WSL	Ubuntu 22.04.5 LTS

Fonte: Autoria própria (2025).

5.2 Experimentação

Os resultados foram divididos em diferentes experimentos, com variações nas configurações de motoristas, vagas e tipos de conexões. O número de vagas é variado de acordo com o número de motoristas, porém, sempre sendo múltiplo de 4 para que tenham sempre a mesma quantidade para cada tipo de vaga (Curta, Longa, Curta Coberta, Longa Coberta). A Tabela 2 apresenta as configurações do sistema para a experimentação.

Tabela 2 – Configurações dos Experimentos.

Experimentos	**N° de Motoristas**	**N° de Vagas**	**Tipo de Conexão**
A	100	100	Local
B	100	100	Remota
C	100	52	Local
D	100	24	Local
E	50	52	Local
F	50	52	Remota
G	30	32	Local
H	30	32	Remota
I	30	16	Local

Fonte: Autoria própria (2025).

Já a Tabela 3, demonstra os resultados obtidos para cada experimento, evidenciando a quantidade de mensagens por agente, a quantidade de transações feitas pelo Velluscinum e a taxa de sucesso, que é a taxa medida de negociações sucedidas por negociações feitas.

Tabela 3 – Resultados dos experimentos

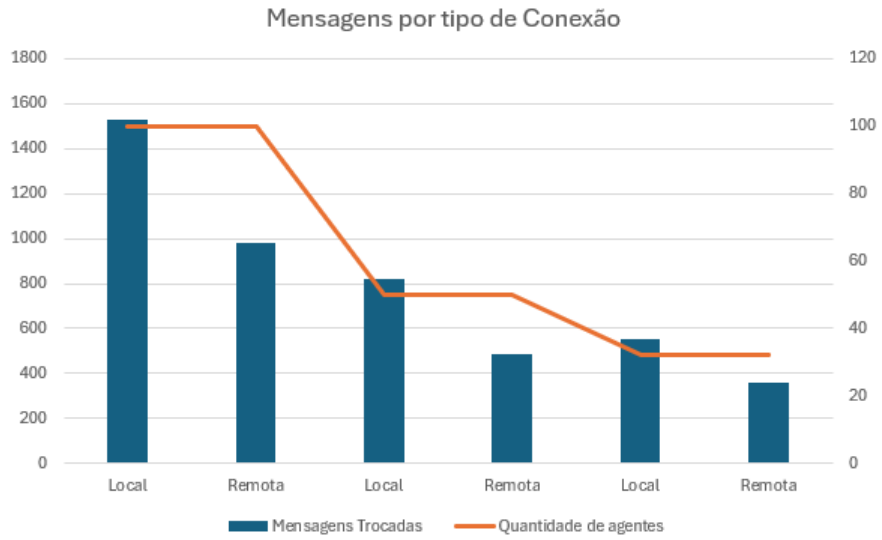
Experimentos	Msg. Gerente	Msg. Motorista	Transações Velluscinum	Taxa de Sucesso
A	214	1317	622	30/35
B	87	895	559	27/30
C	190	1262	569	18/24
D	250	1384	660	20/30
E	143	678	549	23/32
F	81	408	523	28/35
G	143	411	503	28/31
H	84	277	497	32/36
I	114	322	528	33/42

Fonte: Autoria própria (2025).

5.3 Análise e Comparação

As mensagens trocadas pelos agentes foi diretamente influenciada pelo tipo de conexão, conforme mostra a Figura 6, evidenciando que em situações de muitos ou poucos agentes, a tendência é a conexão local ter mais mensagens trocadas do que na conexao remota. Os experimentos tratados nesta comparação são respectivamente A, B, E, F, G e H.

Figura 6 – Mensagens por tipo de Conexão.

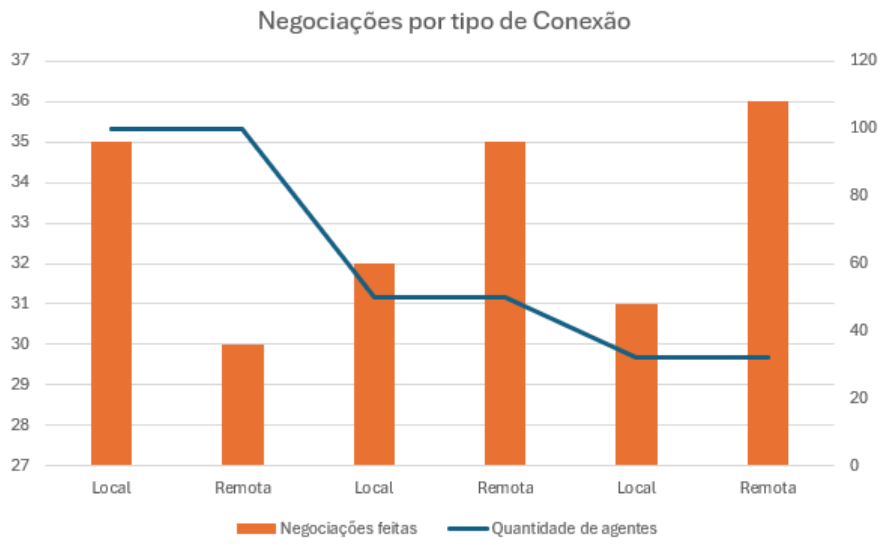


Fonte: Autoria própria (2025).

Porém para a quantidade de negociações a análise é diferente, tendo em vista que em dois casos o resultado foi de mais negociações feitas remotamente do que localmente, como

apresentado na Figura 7. Os experimentos tratados nesta comparação são os mesmos da figura anterior.

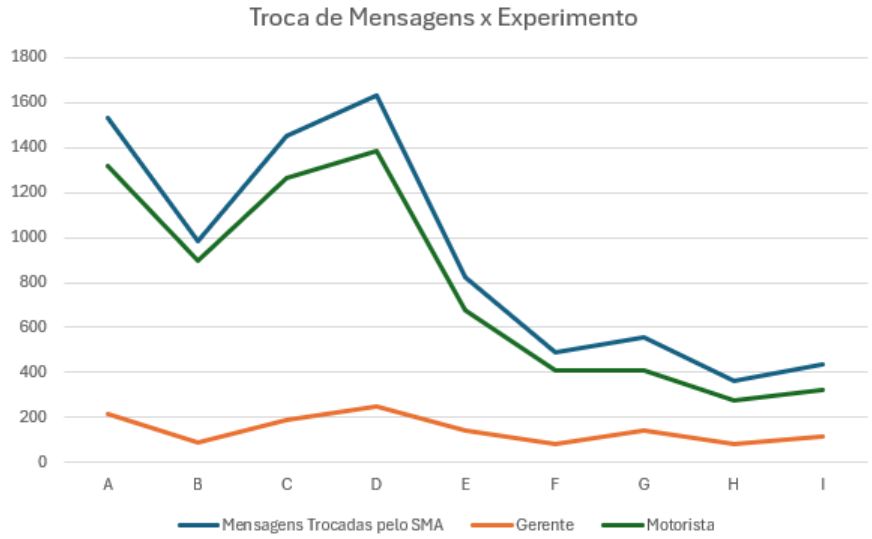
Figura 7 – Negociações feitas por tipo de Conexão.



Fonte: Autoria própria (2025).

Em relação à quantidade de mensagens trocadas por experimento, podemos analisar através da Figura 8, a relação entre mensagens enviadas por motoristas e mensagens enviadas pelos gerentes através dos experimentos feitos. Isso mostra que durante os experimentos, os gerentes foram pouco afetados pela mudança de quantidade de motoristas, evidenciando uma possível sobrecarga, e mal desempenho nos cenários onde existiam muitos motoristas no SMA.

Figura 8 – Mensagens Enviadas.

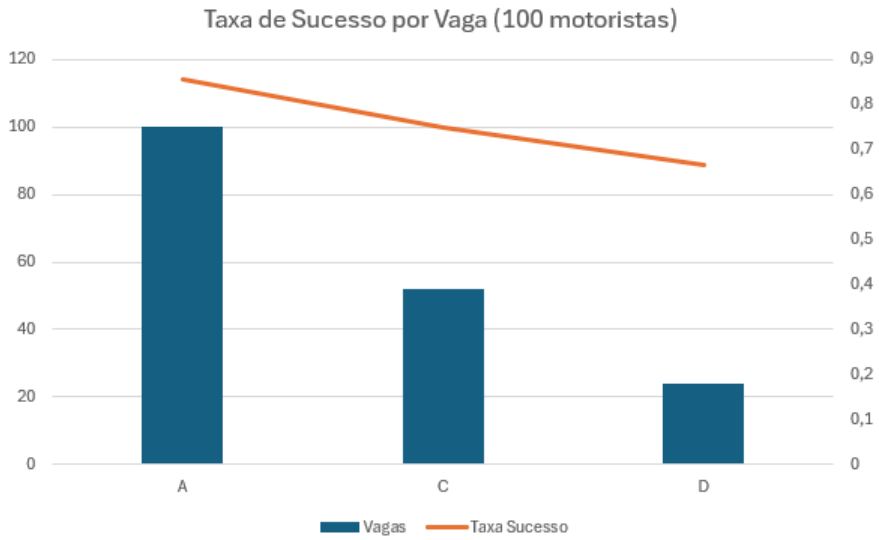


Fonte: Autoria própria (2025).

Também pode ser notado que no experimento D há um pico de mensagens enviadas pelo SMA, neste caso é evidenciado que quanto maior a diferença da quantidade de motoristas

e quantidade de vagas a tendência é que maior seja a quantidade de mensagens trocadas, observando que neste experimento citado, existem 100 Motoristas para 24 vagas apenas. Isso pode ser resultado de requisições dos motoristas sendo negadas ou não entrando no estágio de negociação, reforçado pela Figura 9.

Figura 9 – Taxa de Sucesso de Negociações 1.



Fonte: Autoria própria (2025).

Agora analisando as taxas de sucesso em relação as negociações feitas e comparando com a quantidade de vagas disponíveis no sistema, apresentado na Figura 9 e representando os experimentos A, C e D. Nota-se que a taxa de sucesso é proporcional à quantidade de vagas, mostrando que quanto mais os motoristas disputam pelas vagas, menor é a chance deles obterem o desejado, portanto a taxa de sucesso cai. Por isso, com maior taxa de sucesso de 0,85 é o experimento A, tendo 100 vagas para 100 motoristas, enquanto o menor é experimento D, com uma taxa de 0,66 de sucesso.

6 CONCLUSÃO

Este trabalho teve como objetivo principal o desenvolvimento de um sistema multiagente BDI integrado com a tecnologia *blockchain*, utilizando o *middleware* Velluscinum para execução de negociações descentralizadas de ativos digitais em um ambiente de Smart Parking. Para alcançar esse objetivo, foram definidos objetivos específicos, como a análise de negociações de ativos digitais em sistemas similares, a definição de mecanismos de negociação compatíveis com a utilização de ativos digitais, a implementação da arquitetura BDI nos mecanismos de negociação, a comparação entre diferentes protocolos de negociação e a validação do sistema implementado através de testes.

Com base nos resultados obtidos, pode-se concluir que os objetivos foram, em grande parte, atingidos. O sistema desenvolvido mostrou-se funcional, permitindo a interação de agentes BDI com ativos digitais em um ambiente descentralizado, como o proposto. Além disso, os testes realizados validaram o desempenho do sistema em cenários simulados, o que reforça a aplicabilidade da solução em contextos reais de gestão de estacionamento inteligente. No entanto, algumas limitações foram identificadas, como a centralização do gerenciamento dos ativos digitais em um único agente, o que gerou gargalos em cenários com um maior número de agentes, de acordo com a Tabela 3, além da dependência de uma conexão de rede estável para a execução das transações, fator que pode impactar a performance do sistema.

Apesar das contribuições, pode-se notar algumas limitações através do resultado obtido. Uma delas é que embora as negociações envolvendo os ativos digitais e a *blockchain* sejam descentralizadas, o SMA desenvolvido é centralizado, pois apenas um agente (Gerente) detém o poder de criar e gerenciar os NFTs referentes à vagas do estacionamento. Em cenários extremos e com maior complexidade o sistema demonstra um gargalo por conta da sobrecarga de requisições ao Gerente, pode-se notar isso quando ao olhar os resultados do Capítulo 5, onde o número de negociações não tem um crescimento considerável mesmo com quantidades diferentes de agentes Motoristas.

Também por ser um sistema que depende de conexão à uma rede para executar as transações pela *blockchain*, instabilidades de conexão, podem atrasar estas transações, o que pode impactar na experiência dos usuários.

Estas limitações apresentadas devem ser consideradas para futuras melhorias e ajustes no sistema, com a possível implementação de um SMA com arquitetura descentralizada para resolver os problemas de escalabilidade observados. Além disso, novas formas de negociação poderiam ser exploradas em um sistema descentralizado, ampliando as possibilidades de comunicação entre os agentes e permitindo a simulação de outros cenários, como por exemplo leilões de vagas durante horários de pico.

O *Smart Parking* é apenas um dos muitos componentes das *Smart Cities*. Assim, a integração do SMA desenvolvido neste trabalho com outros sistemas em um cenário mais abrangente permitiria uma melhor compreensão do funcionamento de cidades inteligentes. Por meio

de simulações, seria possível identificar melhorias e propor ajustes no planejamento urbano, beneficiando cidades que pretendem implementar esse conceito no futuro.

A combinação de sistemas multiagentes BDI com *blockchain* através de um *middleware* demonstrou ser uma abordagem promissora para resolver problemas complexos para negociações de forma descentralizada. Este trabalho contribuiu para a consolidação desse campo de estudo, promovendo alternativas e soluções mais seguras, transparentes e escaláveis para a gestão de recursos em cidades inteligentes. As perspectivas futuras reforçam o potencial de expandir o impacto desta pesquisa, integrando tecnologias emergentes em novos contextos e aplicações.

REFERÊNCIAS

ALVES, B. R. *et al.* Experimentation of Negotiation Protocols for Consensus Problems in Smart Parking Systems. *In*: MAříK, V. *et al.* (Ed.). **Industrial Applications of Holonic and Multi-Agent Systems**. Cham: Springer International Publishing, 2019. (Lecture Notes in Computer Science), p. 189–202. ISBN 978-3-030-27878-6. DOI: 10.1007/978-3-030-27878-6_15.

BOISSIER, O.; HüBNER, J. F.; RICCI, A. The JaCaMo Framework. *In*: ALDEWERELD, H. *et al.* (Ed.). **Social Coordination Frameworks for Social Technical Systems**. Cham: Springer International Publishing, 2016. v. 30, p. 125–151. ISBN 978-3-319-33568-1 978-3-319-33570-4. Series Title: Law, Governance and Technology Series. DOI: 10.1007/978-3-319-33570-4_7. Disponível em: http://link.springer.com/10.1007/978-3-319-33570-4_7.

BRATMAN, M. **Intention, plans, and practical reason**. [*S.l.*]: CSLI, 1987.

CALVARESI, D. *et al.* The good, the bad, and the ethical implications of bridging blockchain and multi-agent systems. **Information**, v. 10, n. 12, p. 363, 2019. DOI: 10.3390/info10120363.

CALVARESI, D. *et al.* Trusted registration, negotiation, and service evaluation in multi-agent systems throughout the blockchain technology. *In*: **2018 IEEE/WIC/ACM International Conference on Web Intelligence (WI)**. [*S.l.*: *s.n.*], 2018. p. 56–63. DOI: 10.1109/WI.2018.0-107.

CALVARESI, D. *et al.* Reputation management in multi-agent systems using permissioned blockchain technology. *In*: **2018 IEEE/WIC/ACM International Conference on Web Intelligence (WI)**. [*S.l.*: *s.n.*], 2018. p. 719–725. DOI: 10.1109/wi.2018.000-5.

CASTRO, L. F. S. D.; ALVES, G. V.; BORGES, A. P. Using trust degree for agents in order to assign spots in a Smart Parking. **ADCAIJ: Advances in Distributed Computing and Artificial Intelligence Journal**, v. 6, n. 2, p. 45–55, jun. 2017. ISSN 2255-2863. DOI: 10.14201/ADCAIJ207624555. Disponível em: http://revistas.usal.es/index.php/2255-2863/article/view/ADCAIJ2017624555.

DEMAZEAU, Y. When can knowledge-based systems be called agents? 11 1996.

GMBH, B. **BigchainDB 2.0 The Blockchain Database**. 2018. https://www.bigchaindb.com/whitepaper/.

KUMAR, K. P. V. *et al.* Blockchain technology: A comprehensive review and future directions. *In*: **2023 Second International Conference on Augmented Intelligence and Sustainable Systems (ICAISS)**. [*S.l.*: *s.n.*], 2023. p. 1362–1368.

LAZARIN, N. M. *et al.* Velluscinum: A middleware for using digital assets in multi-agent systems. *In*: MATHIEU, P. *et al.* (Ed.). **Advances in Practical Applications of Agents, Multi-Agent Systems, and Cognitive Mimetics. The PAAMS Collection**. Cham: Springer Nature Switzerland, 2023. p. 200–212. ISBN 978-3-031-37616-0. DOI: 10.1007/978-3-031-37616-0_17.

LIU, S.; HENNEQUIN, S.; ROY, D. Enterprise platform of logistics services based on a multi-agents mechanism and blockchains. **IFAC-PapersOnLine**, v. 54, n. 1, p. 825–830, 2021. DOI: 10.1016/j.ifacol.2021.08.097.

LUO, F. *et al.* A distributed electricity trading system in active distribution networks based on multi-agent coalition and blockchain. **IEEE Transactions on Power Systems**, v. 34, n. 5, p. 4097–4108, 2019. DOI: 10.1109/tpwrs.2018.2876612.

MHAMDI, H. *et al.* Trust-based smart contract for automated agent to agent communication. **Computational Intelligence and Neuroscience**, v. 2022, p. 1–11, 2022. DOI: 10.1155/2022/5136865.

MINARSCH, D. *et al.* Autonomous economic agent framework. *In*: ALECHINA, N.; BALDONI, M.; LOGAN, B. (Ed.). **Engineering Multi-Agent Systems**. Cham: Springer International Publishing, 2022. p. 237–253. ISBN 978-3-030-97457-2. DOI: 10.1007/978-3-030-97457-2_14.

OLIVEIRA, G. L.; ALVES, G. V.; LAZARIN, N. M. Um mapeamento de sistemas multiagentes e ativos digitais com uma proposta de aplicaçao em smart parking. *In*: **Anais do XVII Workshop-Escola de Sistemas de Agentes, seus Ambientes e Aplicações- WESAAC2023**. Pelotas, Brasil: [*s.n.*], 2023. p. 32–43.

PAPI, F. G.; HüBNER, J. F.; de Brito, M. A blockchain integration to support transactions of assets in multi-agent systems. **Engineering Applications of Artificial Intelligence**, v. 107, p. 104534, 2022. ISSN 0952-1976. DOI: 10.1016/j.engappai.2021.104534. Disponível em: https://www.sciencedirect.com/science/article/pii/S0952197621003821.

RAJASEKARAN, A. S.; AZEES, M.; AL-TURJMAN, F. A comprehensive survey on blockchain technology. **Sustainable Energy Technologies and Assessments**, v. 52, p. 102039, 2022. ISSN 2213-1388. Disponível em: https://www.sciencedirect.com/science/article/pii/S2213138822000911.

RAO, A. S. AgentSpeak(L): BDI agents speak out in a logical computable language. *In*: CARBONELL, J. G. *et al.* (Ed.). **Agents Breaking Away**. Berlin, Heidelberg: Springer Berlin Heidelberg, 1996. v. 1038, p. 42–55. ISBN 978-3-540-60852-3 978-3-540-49621-2. Series Title: Lecture Notes in Computer Science. DOI: 10.1007/BFb0031845. Disponível em: http://link.springer.com/10.1007/BFb0031845.

SMITH. The contract net protocol: High-level communication and control in a distributed problem solver. **IEEE Transactions on Computers**, C-29, n. 12, p. 1104–1113, 1980.

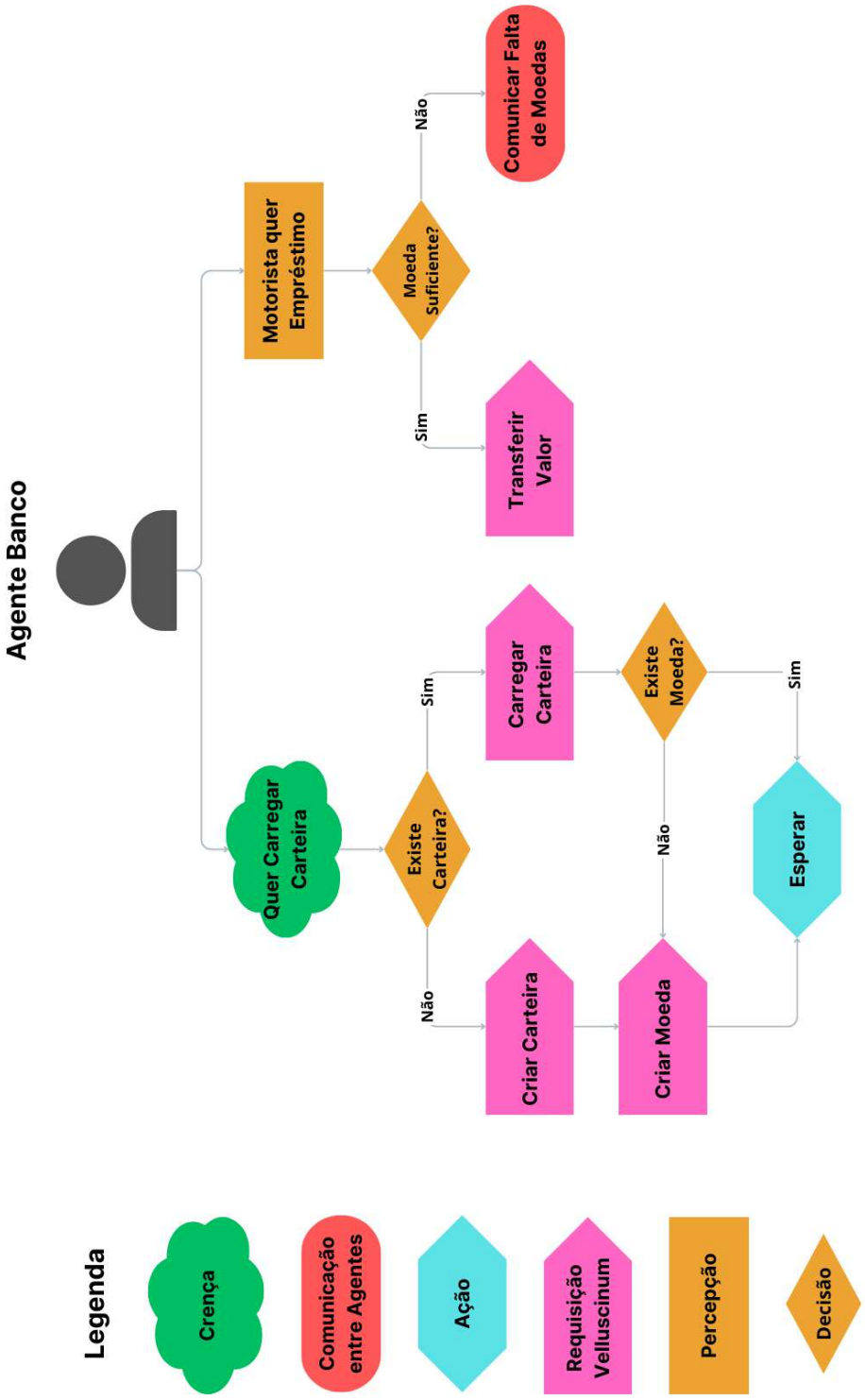
SONG, Y. *et al.* Resilience network controller design for multi-domain sdn: A bdi-based framework. *In*: **2022 IEEE 95th Vehicular Technology Conference: (VTC2022-Spring)**. [*S.l.*: *s.n.*], 2022. p. 1–5. DOI: 10.1109/VTC2022-Spring54318.2022.9860710.

TANENBAUM, A. S.; STEEN, M. van. **Distributed Systems: Principles and Paradigms**. 2. ed. Upper Saddle River, NJ: Pearson Prentice Hall, 2007. ISBN 978-0-13-239227-3.

WOOLDRIDGE, M. **An Introduction to MultiAgent Systems**. 2nd. ed. [*S.l.*]: Wiley Publishing, 2009. ISBN 0470519460.

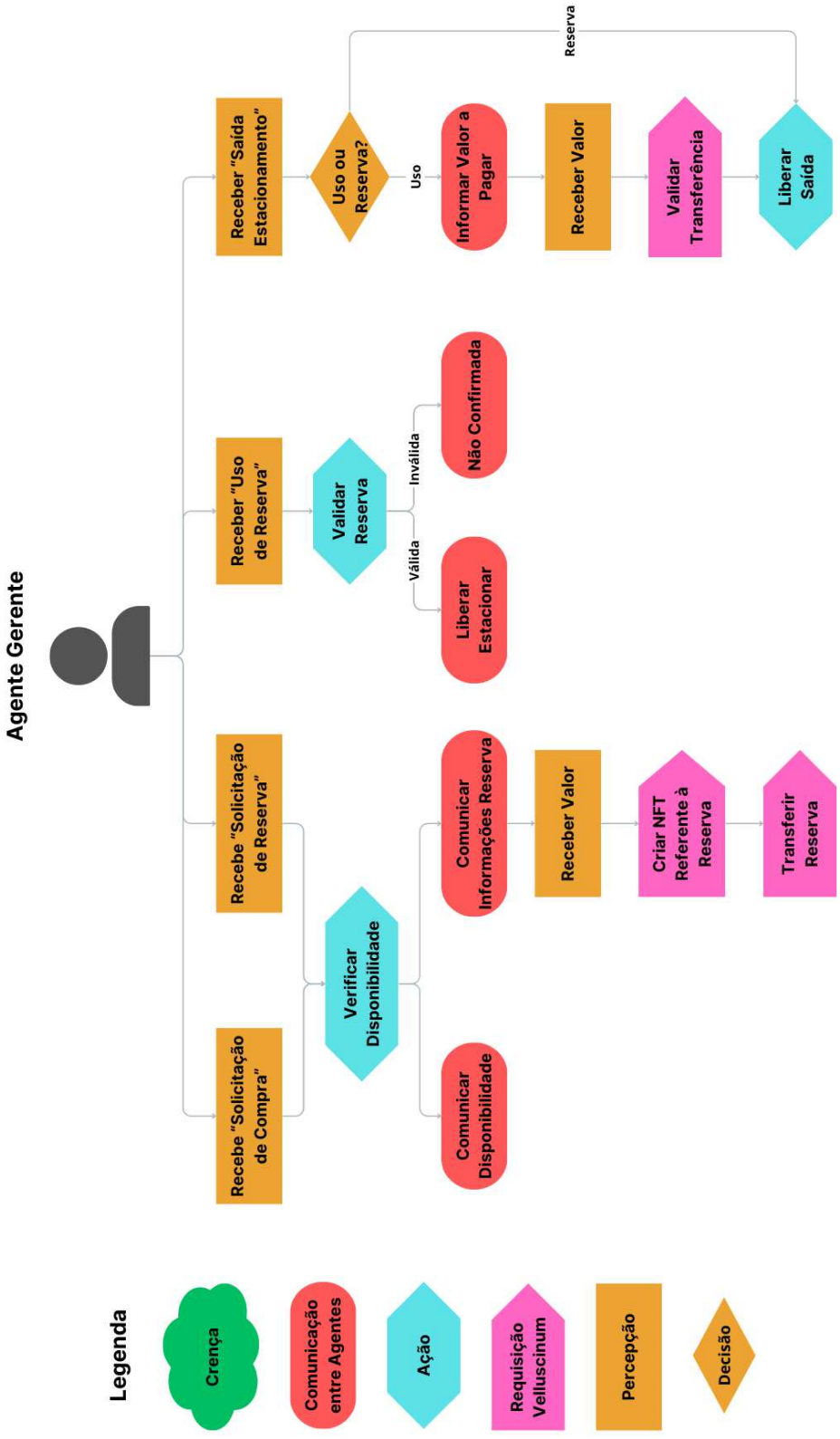
APÊNDICE A – Diagramas dos Agentes

Figura 10 – Diagrama Agente Banco



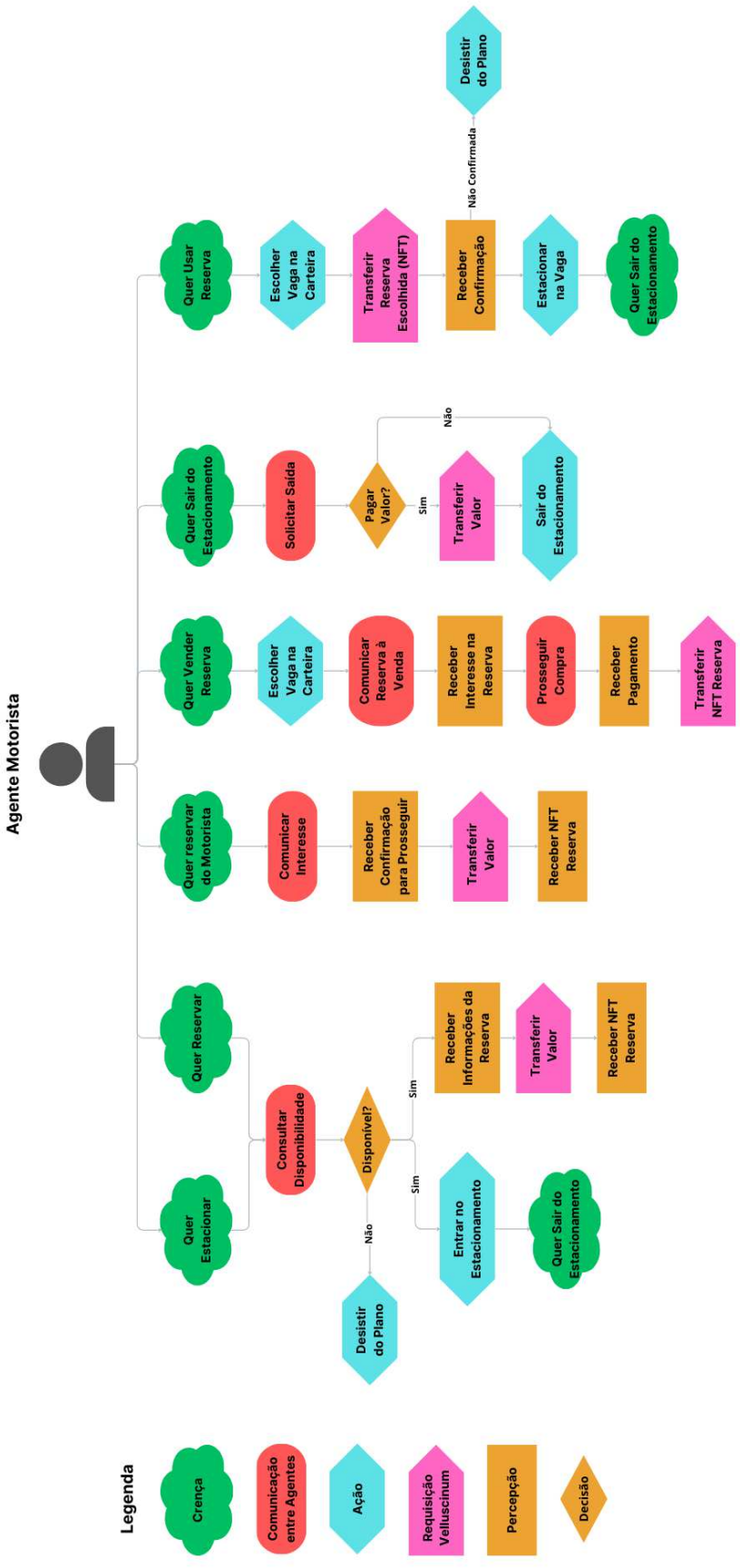
Fonte: Autoria própria (2025).

Figura 11 – Diagrama Agente Gerente



Fonte: Autoria própria (2025).

Figura 12 – Diagrama Agente Motorista



Fonte: Autoria própria (2025).