

# Transferência de Dados entre Agentes em um Sistema Multiagente no Domínio do Gerenciamento de Energia em Veículos Elétricos

Carlos E. da Veiga<sup>1</sup>, Ronaldo S. Mello<sup>2</sup>, Carlos Ramos<sup>3</sup>, Juan Manuel Corchado<sup>4</sup>,  
Carina F. Dorneles<sup>2</sup>

<sup>1</sup>Instituto Federal de Santa Catarina(IFSC)  
Av. Mauro Ramos, 950 – 88.020-300 – Florianópolis – SC – Brazil

<sup>2</sup>Universidade Federal de Santa Catarina (UFSC)  
Cx.P. 476 – 88.040-370 – Florianópolis – SC – Brazil

<sup>3</sup>Polytechnic of Port  
R. Dr. António Bernardino de Almeida, 431 – 4249-015 – Porto – Portugal

<sup>4</sup>University of Salamanca  
Calle Espejo s/n, Edificio I+D+i – 37007 – Salamanca – Spain

cveiga@ifsc.edu.br, {r.mello, carina.dorneles}@ufsc.br,

csr@isep.ipp.pt, jm@corchado.net

**Abstract.** *Electric vehicle energy management comprises a set of procedures and activities whose ultimate focus is to optimize energy use. In this scenario, in recent years, much research has been developed to provide frameworks to assist these procedures and activities through Machine Learning techniques and multi-agent systems. However, these works do not detail the data management involved in these environments, not even how the data models are structured, including considering the issue of secrecy of sensitive data. This paper proposes a multi-agent system model, and details the transfer and sharing of data between the involved agents. The tests carried out demonstrate the efficiency of the adopted model and the security in the exchange of data between agents.*

**Resumo.** *A gestão energética dos veículos elétricos compreende um conjunto de procedimentos e atividades cujo foco final é a otimização do uso de energia. Neste cenário, nos últimos anos, muitas pesquisas foram desenvolvidas para fornecer estruturas que auxiliem esses procedimentos e atividades através de técnicas de Aprendizado de Máquina e sistemas multiagentes. Porém, esses trabalhos não detalham o gerenciamento de dados envolvido nesses ambientes, nem mesmo como os esquemas de dados são estruturados, inclusive considerando a questão do sigilo dos dados sensíveis. Este artigo propõe um modelo de sistema multiagente e detalha a transferência e compartilhamento de dados entre os agentes envolvidos. Os testes realizados comprovam a eficiência do modelo adotado e a segurança na troca de dados entre agentes.*

## 1. Introdução

Muitos países têm incentivado a fabricação de veículos elétricos (VE) para substituir os veículos a combustão interna, porém, isso não determina o sucesso na sua adoção. O

trabalho de Bryła [Bryła et al. 2023] destaca partes da Ásia, a Europa e EUA & Canadá como grande fontes de pesquisa sobre a adoção de VE. Um dos fatores que provoca mais rejeição na adoção é a dificuldade em gerenciar a energia elétrica, gerando ansiedade ao usuário. Para mitigar esta dificuldade no gerenciamento da energia elétrica em VE, vários pesquisadores têm realizado trabalhos que utilizam técnicas de aprendizado de máquina ("*Machine Learning*" - ML) [Almaghrebi et al. 2021, Shahriar et al. 2021, Zhai et al. 2019, Alqahtani et al. 2022], sistemas multiagentes ("*Multi-Agent System - MAS*") e métodos de otimização [Shahriar et al. 2020]. Entretanto, para que essas técnicas sejam aplicadas, faz-se necessário um volume de dados muito grande e confiável [Calearo et al. 2021], sendo necessário transmitir estes dados de um agente para outro de forma confiável e segura.

Um MAS possui vários agentes, trazendo a necessidade de gerência dos dados distribuídos pois, em alguns casos, os mesmos dados deverão estar em diferentes agentes, uma vez que são aplicadas técnicas de ML em cada agente. O resultado da aplicação destas técnicas gera novas informações que são transferidas entre os agentes. Este processo exige um tratamento dos dados diferente de um sistema de gerenciamento de banco de dados (SGBD) convencional, sendo necessário, em alguns casos, o uso de diferentes SGBDs para cada agente, dificultando o controle dos dados. Outro fato, por conta de posições geográficas diferentes e da dinâmica no deslocamento espacial de alguns agentes, podem ocorrer falhas ou perdas na comunicação entre eles.

Os agentes no contexto de um MAS são independentes [Palanca et al. 2020], promovendo um volume de informações exclusivas para cada agente. Na gestão de energia elétrica de VE, os novos dados gerados são referentes à coleta de informações dos elementos envolvidos, como o VE e a rede elétrica. Os dados, ou parte deles, pertencentes a um agente devem ser tratados e transferidos para outros agentes para a realização da gestão de energia do sistema. A transmissão dos dados deve ser realizada por meio de mensagens utilizando o protocolo "*eXtensible Messaging and Presence Protocol*" (XMPP) e na especificação definida pela "*Foundation of Intelligent Physical Agents*"<sup>1</sup> (FIPA), sendo necessário um controle adicional para garantir que dados foram transferidos integralmente ao outro agente.

Este artigo traz a proposta de um modelo de transferência de dados para um MAS que considera o armazenamento de dados distribuídos entre os agentes, onde cada agente possui um conjunto de informações estruturadas e tem-se algumas prerrogativas que devem ser garantidas da mesma forma que em um SGBD, que são disponibilidade, consistência, controle de acesso e sigilo da informação. Os dados armazenados são coletados de diversas fontes, como geração fotovoltaica, dados do VE, dados de viagens, valores da energia elétrica e dados do clima, entre outros. Estes dados são utilizados pela solução de ML adotada, responsável pelo gerenciamento da energia elétrica consumida pelo VE, permitindo auxiliar o proprietário/usuário na manutenção da carga de energia do VE. As principais contribuições do presente trabalho são: (i) uma proposta de um modelo de MAS para gestão de energia de veículos elétricos; (ii) um ambiente de gestão de dados em um MAS; e (iii) um procedimento de transferência de dados em um MAS.

A proposta descrita neste artigo se diferencia de outros trabalhos relacionados,

---

<sup>1</sup><http://www.fipa.org/>

pois compreende um ambiente onde os agentes possuem independência na gestão dos dados. A transferência de dados entre agentes se dá por meio de mensagens com especificações FIPA, uma vez que um determinado agente não pode acessar o BD de outro agente. Ainda, os dados que são coletados nos agentes são sensíveis e é necessário observar a Lei Geral de Proteção de Dados Pessoais (LGPD) [BRASIL 2018], utilizando de artifícios para anonimizar as informações para evitar que ocorra a exposição de dados privados.

Este artigo está organizado conforme segue. A Seção 2 descreve os trabalhos relacionados. A Seção 3 apresenta o modelo de MAS para gestão de energia de VE. Os procedimentos de transferência de dados entre os agentes são apresentados na Seção 4. A metodologia desenvolvida para os testes é apresentada na Seção 5. Na Seção 6 é apresentada a implementação do modelo em um MAS, e os resultados obtidos com testes. Finalmente, as conclusões e trabalhos futuros são descritos na Seção 7.

## 2. Trabalhos relacionados

A proposta apresentada em [Korovin et al. 2020] descreve um esquema para armazenamento de dados distribuídos em agentes, sendo mantidos os dados nos agentes próximos à aplicação que necessita deles. O estudo descrito em [ALSHAMMARI and EASSA 2018] aborda o uso de agentes móveis que consultam BDs em nuvem e permitem o deslocamento entre vários pontos de conexão. O trabalho de [Pirani et al. 2022], por sua vez, propõe a arquitetura de um Sistema Multiagente de Modelo Relacional ("*Relational-model multi-agent system*" - RMAS) para definir a padronização do relacionamento entre os dados em um MAS. Entretanto, estas abordagens enfatizam funcionalidades de um SGBD na forma de um MAS, sendo os dados dependentes entre os agentes e não possui aplicação direta para o propósito deste estudo, pois não aborda independência entre os agentes.

A pesquisa desenvolvida por Bansal [Bansal et al. 1998] aborda uma forma de replicar informações entre agentes de um MAS para garantir a disponibilidade da informação. Ele utiliza mensagens entre os agentes, porém não detalha questões do sigilo e troca parcial de dados entre os agentes. Outra abordagem, descrita em [Ramasubramian and Kannan 2004], utiliza um MAS para garantir a segurança de um BD através de uma estrutura de rede neural que prevê ataque de *hackers*, mas há um relacionamento dos agentes com um único BD. Ainda há a aplicação de MAS para gestão de dados distribuídos utilizando agentes em um sistema de análise forense [Khan 2019], onde os agentes coletam *log* de atividades em partes diferentes da rede e as transferem para um agente responsável pelo armazenamento das informações. Nestes trabalhos não há detalhamento de como se garante a transferência dos dados entre os agentes.

## 3. Modelo de sistema multiagente

Os agentes de um MAS possuem características definidas por comportamentos (*behaviours*), que podem ser cíclicos, de ciclo único, periódicos, de tempo limite e de máquina de estado finito. Além disso, há outras funcionalidades que permitem combinar estes comportamentos (detalhados em [Palanca et al. 2020]), utilizando o XMPP para permitir ao agente se conectar com o MAS e enviar mensagens. A autenticação e o gerenciamento

do agente são realizados por um servidor de XMPP, como o *Prosody*<sup>2</sup> e criptografia TLS/SSL (*Transport Layer Security/Secure Sockets Layer*) para garantir a segurança.

Para a gestão de energia de um VE, é necessária a interação entre vários elementos da rede elétrica, pois todo o sistema funciona de forma interligada e toda a ação de qualquer elemento afetará os demais na rede elétrica. Para representar este ambiente em um sistema que permita esta interdependência, mas também a autonomia dos elementos envolvidos, foi elaborado um modelo de MAS mostrado na Figura 1, que representa cada elemento, ou grupo de elementos, da rede como um agente, conforme descrito a seguir.

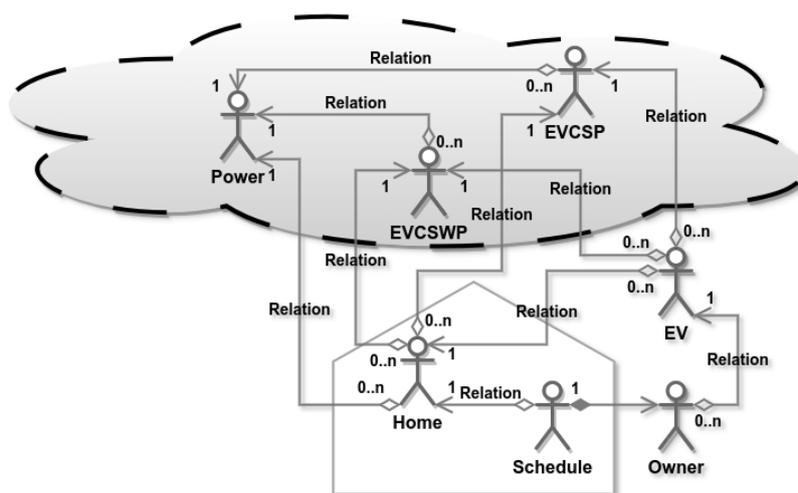


Figura 1. Modelo de MAS para gestão da energia de VE.

Os **agentes móveis** representam os elementos que podem se deslocar no ambiente definido, como uma cidade ou uma região limitada. O modelo proposto aborda somente viagens curtas que compreendem percursos entre casa, local de trabalho e lazer, e não aborda viagens longas. São definidos dois agentes móveis: o agente que representa o veículo elétrico (EV) e o agente que representa o proprietário do VE (Owner).

Os **agentes da casa** são dois: a própria casa (Home) e a agenda do proprietário do VE (Schedule). Como este modelo está direcionado ao usuário doméstico, o agente *Home* concentra a maior parte das rotinas e realiza várias comunicações com os demais agentes para coletar e atualizar dados. Eles estão contidos dentro de um desenho com linha contínua na Figura 1.

Existem ainda os **agentes na nuvem**. A nuvem representa a região ou cidade onde estará sendo executado o MAS, permitindo utilizar elementos agregadores, que são agentes que representam vários elementos do sistema e que possuem funções similares. O uso do agente agregador para grupos de consumidores é apresentado em um MAS hierárquico cooperativo para agendamento de carregamento de VE [Saner et al. 2022] e também é utilizado no MAS em [Nizami et al. 2021] para o gerenciamento coordenado de VE em uma rede residencial, com um único ponto de conexão para o cliente. Nesta representação foram considerados três agentes agregadores, sendo o primeiro agente agregador das estações de carregamento para VE públicas (EVCS), o segundo agente agre-

<sup>2</sup><https://prosody.im/>

gador das estações de carregamento em edificações privadas (*EVCSWP*), como estacionamento no local de trabalho e o terceiro agente agrega os elementos do sistema de energia elétrica (*Power*). Eles estão contidos na área com linha tracejada na Figura 1.

Em cada agente há um conjunto próprio de dados para atender às funções desempenhadas no MAS, porém alguns dados são similares entre agentes diferentes. A principal característica deste MAS é a independência entre os agentes, permitindo que a tecnologia de SGBD seja específica para cada agente, não sendo possível a comunicação direta entre os SGBDs. Para exemplificar, é mostrado o esquema lógico de dados do agente *EV* na Figura 2, que é composto por informações básicas, histórico de manutenção e de carregamento do VE, sendo relacionados pelo identificador do VE na tabela de informações básicas. A tabela de histórico de carregamento possui tabelas auxiliares, sendo uma para intervalos de carregamento e a outra para localização dos pontos de carga.

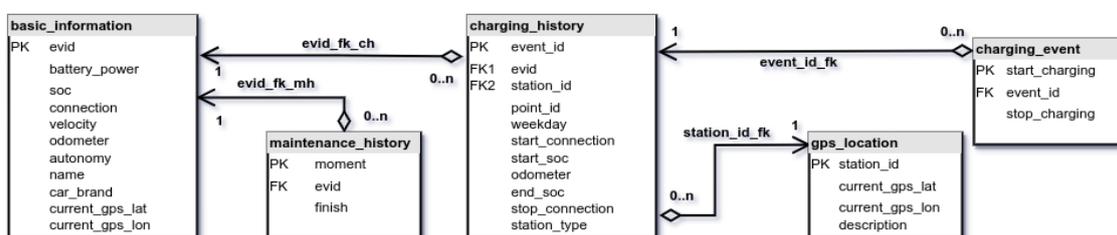


Figura 2. Esquema lógico de dados no agente *EV*.

Cada agente manipula diferentes informações que são definidas conforme a função do agente no MAS e exige uma estrutura de dado própria, conforme descrito na Tabela 1. Os agentes *Home*, *EVCSWP*, *EVCSWP* e *Power* possuem algoritmos de ML e exigem informações de outros agentes para realizar esta função. As informações comuns entre agentes, e que são transferidas por mensagens, estão representadas em negrito.

Tabela 1. Informações organizadas em cada agente.

Agente	Informações
EV	<b>Configurações, localização</b> , manutenção e carregamento.
Owner	<b>Identificação, localização</b> , eventos, viagens e VE.
Schedule	Eventos e <b>localização</b> dos pontos de eventos.
Home	Configurações, <b>localização</b> , estação de carregamento da casa, <b>preço da energia, configurações do VE</b> , consumo da casa e do VE, <b>proprietário</b> , geração fotovoltaica, clima e resultados de aprendizagem realizadas.
EVCSWP	Configurações do agregador e das estações de carregamento, <b>preço da energia</b> , pontos de carregamento, controle de reservas e filas para carregamento do VE, consumo, limites da energia elétrica e resultados de aprendizagem realizadas.
EVCSWP	Configurações do agregador e das estações de carregamento, <b>preço da energia</b> , pontos de carregamento, controle de reservas, filas para carregamento, consumo, limites da energia elétrica e resultados de aprendizagem realizadas.
Power	Configurações do agregador e das estações de carregamento, <b>preço da energia</b> , preço marginal, pontos de carregamento, controle de reservas, filas para carregamento do VE, consumo, limites da energia elétrica e resultados de aprendizagem realizadas.

#### 4. Procedimentos para Transferência de Dados entre os Agentes

A Figura 3 mostra o fluxo das informações de um agente, que é composto de três comportamentos. No primeiro, o agente fica na escuta das solicitações de agentes externos e realiza os procedimentos para atender a solicitação. No segundo comportamento são realizados a coleta, tratamento e armazenamento das informações proveniente de fontes externas ao agente ou de outros agentes. Por fim, o terceiro comportamento executa o processo de ML. Após a execução é armazenado o resultado no BD e distribuídas as informações que são enviadas a outros agentes. O ML é executado em máquinas que hospedam os agentes *Home*, *EVCS*, *EVCSWP* e *Power*.

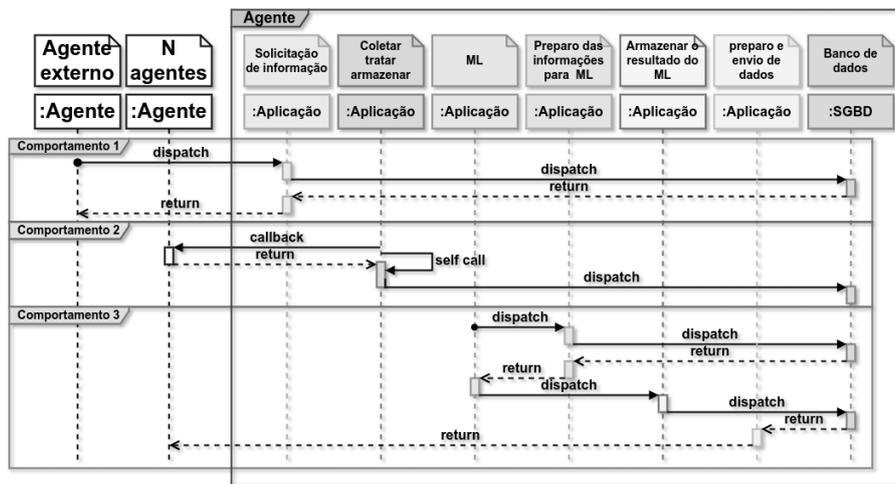


Figura 3. Fluxo de dados no agente.

A comunicação entre os agentes em um servidor XMPP utiliza endereços *Jabber IDs* ou *JIDs*, que são semelhantes ao endereço de e-mail. Estes endereços são trocados entre os agentes no início do processo e criam um relacionamento entre os agentes envolvidos. O envio de mensagem utiliza o formato JSON (*JavaScript Object Notation*) encapsulado no corpo da mensagem encaminhada pelo agente. Conforme mostra a Figura 4, o agente A envia uma mensagem do tipo *FIPA - Inform* utilizando o servidor XMPP. O corpo da mensagem possui como primeira informação o "tipo", que indica qual a função da mensagem enviada: "send", "data error" ou "received data". A segunda informação é o identificador dos dados e a terceira informação é o indicador da entidade de origem dos dados. Todas essas três informações são comuns a todas as mensagens trocadas durante o envio de dados.

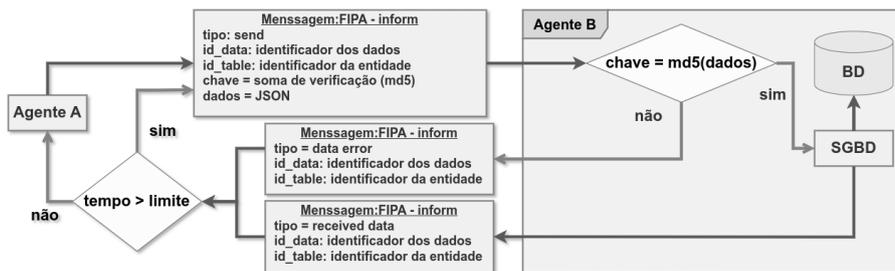


Figura 4. Processo de envio de dados entre agentes através de mensagens

A informação "tipo" permite três funções diferentes: *send*, *received data* e *data error*. Quando "tipo"="send", indica-se a primeira etapa do envio da mensagem, que é o envio dos dados. Neste momento é enviado uma chave MD5<sup>3</sup> e os dados estruturados no formato JSON. Para gerar a chave é calculado um valor com o conjunto dos dados no formato JSON enviados, utilizando a função *md5* da biblioteca *hashlib* em Python, que neste processo não possui a função de criptografia para proteção dos dados. Ela apenas verifica a consistência dos dados. Para a proteção dos dados é utilizada a criptografia TLS/SSL.

Uma mensagem enviada por um agente **A** chega a um agente **B** e é processada, sendo inicialmente verificada a consistência dos dados pela comparação com a chave MD5. Caso os dados estejam íntegros, eles são salvos no BD do agente **B** e é enviada uma resposta do "tipo"="received data". Nesta situação, o processo é registrado no agente **A** e encerrado. Por outro lado, caso ocorra diferença entre a chave e a soma de verificação no agente **B**, é enviada uma mensagem ao agente **A** com o "tipo"="data error". Quando o agente **A** recebe uma mensagem deste tipo ou a resposta não retorna em um limite de tempo controlado pelo emissor, é realizado um procedimento para reenviar a mensagem. O controle do processo é feito por uma variável no formato de dicionário, cuja chave é constituída pelo destinatário e identificador da mensagem, permitindo assim controlar o número de tentativas de reenvio e o tempo de espera de resposta no agente emissor.

Com esses procedimentos de mensagens e conexões entre agentes, é gerado um conjunto de dados que estão em locais diferentes, sem um controle direto entre os SGBDs, mas há consistência dos dados replicados. Este procedimento permite um isolamento completo das informações em cada agente, mas, por outro lado, permite compartilhar as informações necessárias para a gestão dos dados de forma sincronizada.

## 5. Metodologia para os Testes

A metodologia desenvolvida envolve a comunicação entre agentes que estão interligados pela rede de comunicação e, alguns agentes, se deslocam na região de estudo. Neste ambiente podem ocorrer falhas de conexão, dados corrompidos, travamento da aplicação, entre outros. Sendo todo o controle no envio ou reenvio de dados realizado pelo agente emissor, portanto as situações emuladas nos testes se referem ao controle existente no agente emissor. Os erros são emulados no agente receptor dos dados e não há controle sobre isso pelo agente emissor. Para a simulação foi estabelecido o número de reenvio dos dados em quatro tentativas.

Dois tipos de erros envolvendo a comunicação são considerados: um considera a falha na comunicação e outro considera dados corrompidos. Para tanto, foram elaboradas quatro situações possíveis cujos resultados são apresentados na Tabela 2, conforme detalhado a seguir.

**Tipo de situação 1:** Não são inseridos erros e o principal objetivo deste teste é para verificar se o tempo limite de resposta para identificar uma falha, é suficiente para que não resulte um falso positivo, considerando as ações do agente;

**Tipo de situação 2:** Com um erro de 5% das mensagens com falha no envio de resposta do receptor, ou ainda, a falha na coleta da resposta no emissor. O objetivo

---

<sup>3</sup>MD5 - Função hash que produz um valor de hash de 128 bits expresso em 32 caracteres.

desta situação é verificar se o agente emissor consegue detectar a falha de envio e executar os procedimentos para tratamento desta falha;

**Tipo de situação 3:** Com um erro de 5% das mensagens que são coletadas no receptor com os dados corrompidos e, o receptor irá responder com uma mensagem ao emissor solicitando o reenvio dos dados. Esta situação verifica a reação do receptor e do emissor para corrigir a falha de dados corrompidos;

**Tipo de situação 4:** Por fim, são consideradas ambas as situações 2 e 3 em um mesmo momento, possibilitando verificar o comportamento do emissor e do receptor tratando falhas simultâneas.

## 6. Implementação do modelo em um MAS

A função principal do modelo do MAS é a gestão de energia de VE em uma residência utilizando IA no agente *Home* para identificar o momento e local ideal para carregar o VE, sendo necessário um conjunto de dados em cada agente e a realização de transferências de informações entre os agentes de forma segura, com o objetivo de comprovar a eficiência do esquema de dados adotado e a segurança na troca de dados.

### 6.1. Planejamento dos Testes

Para realizar os testes no modelo do MAS, há um agente de cada tipo e executado um modelo de MAS sem as aplicações de ML. Os dados utilizados são informações randômicas geradas e armazenadas no banco de dados de cada agente, para evitar expor os dados reais. A troca de informações entre agentes é realizada com a transferência de 1.000 registros com três atributos cada. O envio foi repetido em quatro situações diferentes conforme descrito na seção anterior. Os erros foram implementados utilizando uma distribuição pseudo aleatória da função `Random`<sup>4</sup> do Python.

A modelagem do MAS é implementada utilizando a biblioteca *Spade* na versão 3.2.3, sendo utilizado o SQLite como SGBD e o algoritmo de IA é implementado em *Python* na versão 3.9.16. Todo o desenvolvimento da modelagem foi realizada em um Laptop Avell High Performance C62 MOB, com 32 GB de memória RAM, processador 11th Gen Intel® Core™ i7-11800H × 16, SSD 2TB, sistema operacional Ubuntu 23.04.

### 6.2. Resultados

Os principais resultados dos experimentos realizados são mostrados na Tabela 2, onde foram enviadas 1.000 mensagens em 4 situações diferentes e contabilizado o número de mensagens enviadas entre os agentes por tipo de erro, detalhadas a seguir.

No tipo de Situação 1, foi realizado o envio das mensagens sem inserir qualquer tipo de erro e não ocorreu nenhuma falha no envio de mensagens, sendo registrado como todas entregues. Nos tipos de Situação 2, 3 e 4, também foram entregues todas as mensagens, porém houve a necessidade de reenviar as mensagens para as simulações de erros de envio de mensagens (Tempo excedido) e erros de corrupção de dados (Dados com erro), sendo simulado 5% de erro em três combinações.

As mensagens reenviadas também possuem a mesma probabilidade de erro. O reenvio de mensagens está relacionado com o tipo de erro simulado, ou seja, se ocorreu

---

<sup>4</sup><https://docs.python.org/pt-br/3.9/library/random.html>

**Tabela 2. Resultado de troca de 1.000 mensagens entre agentes.**

Tipo de sit.	Núm. de tent.	Simulação de erro (%)		Mensagens entregues		Mensagens reenviadas por	
		Dados	Envio	Sim	Não	Dados	Tempo
1	4	0	0	1.000	0	0	0
2	4	0	5	1.000	0	0	42
3	4	5	0	1.000	0	40	0
4	4	5	5	1.000	0	37	49

erro relacionado somente a corrupção dos dados o reenvio é contabilizado como erro nos dados. Porém, nos casos em que ocorrem erros relacionados a resposta não ser efetivada no limite de tempo estipulado, pode ter ocorrido também o erro de corrupção de dados e, assim, o número de reenvios por conta de falha na resposta é maior de que o erro por corrupção de dados. Já em situações de simulação de somente um dos tipos de erros, como nos tipos 2 e 3, o número de reenvios é similar.

Considera-se que os resultados dos testes aplicados ao modelo foram bons, conforme mostra a Tabela 2, uma vez que todos os dados são entregues. O modelo atingiu o objetivo proposto neste trabalho, garantindo a transferência dos dados entre os agentes.

## 7. Conclusão

Este trabalho visa facilitar a gestão de informações no gerenciamento da energia de VE, permitindo a gestão da energia de forma fácil e intuitiva. A forma como é proposta a distribuição dos dados permite um controle de privacidade por isolamento dos agentes e criptografia TLS/SSL na transmissão de dados. Ela também permite a disponibilidade das informações necessárias para cada agente e garante o envio de dados corretos ao destino. Além disso, permite que o agente emissor se certifique que o receptor recebeu a informação correta. A gestão de usuários no servidor XMPP é similar ao de mensagens eletrônicas ou e-mail, sendo uma questão natural para a população em geral.

Para trabalhos futuros, há uma necessidade de rever os padrões FIPA para que possam garantir a troca de dados entre os agentes de forma segura, pois os padrões foram definidos com foco no relacionamento entre os agentes, envolvendo negociações, colaboração e comandos, sem considerar uma transferência de dados estruturados. Além disso, o modelo MAS proposto aborda somente viagens curtas, sendo necessário, portanto, investigar o perfil de viagens longas.

## Referências

- Almaghrebi, A., al Juheshi, F., James, K., Aljuhaishi, N., and Alahmad, M. (2021). Pevs idle time prediction at public charging stations using machine-learning methods. In *2021 IEEE Transportation Electrification Conference & Expo (ITEC)*, pages 1–5.
- Alqahtani, M., Scott, M. J., and Hu, M. (2022). Dynamic energy scheduling and routing of a large fleet of electric vehicles using multi-agent reinforcement learning. *Computers & Industrial Engineering*, 169:108180.
- ALSHAMMARI, S. T. and EASSA, F. (2018). Designing a Flexible Architecture based on mobile agents for Executing Query in Cloud Databases. In *2018 21st Saudi Computer Society National Computer Conference (NCC)*, pages 1–6.

- Bansal, A. K., Ramohanarao, K., and Rao, A. (1998). Distributed storage of replicated beliefs to facilitate recovery of distributed intelligent agents. In *Intelligent Agents IV Agent Theories, Architectures, and Languages*, pages 77–91, Berlin, Heidelberg. Springer.
- BRASIL (2018). Lei nº 13.709, de 14 de agosto de 2018. *Lei Geral de Proteção de Dados Pessoais (LGPD)*.
- Bryła, P., Chatterjee, S., and Ciabiada-Bryła, B. (2023). Consumer adoption of electric vehicles: A systematic literature review. *Energies*, 16(1).
- Calearo, L., Marinelli, M., and Ziras, C. (2021). A review of data sources for electric vehicle integration studies. *Renewable and Sustainable Energy Reviews*, 151:111518.
- Khan, M. N. A. (2019). Multi-agent Based Forensic Analysis Framework for Infrastructures Involving Storage Networks. *Proceedings of the National Academy of Sciences, India Section A: Physical Sciences*, 89(2):291–309.
- Korovin, I. S., Ivanov, D. Y., and Semenisty, S. A. (2020). The Algorithm of Distributed Database for Multi-Agent Interaction. In *ACAI '19, ACAI '19*, pages 303–307, New York, NY, USA. Association for Computing Machinery.
- Nizami, M. S. H., Hossain, M. J., and Mahmud, K. (2021). A Coordinated Electric Vehicle Management System for Grid-Support Services in Residential Networks. *IEEE Systems Journal*, 15(2):2066–2077.
- Palanca, J., Terrasa, A., Julian, V., and Carrascosa, C. (2020). Spade 3: Supporting the new generation of multi-agent systems. *IEEE Access*, 8:182537–182549.
- Pirani, M., Bonci, A., and Longhi, S. (2022). Towards a formal model of computation for RMAS. In Longo, F., Affenzeller, M., and Padovano, A., editors, *ISM 2021*, volume 200 of *Procedia Computer Science*, pages 865–877. ELSEVIER SCIENCE BV.
- Ramasubramian, P. and Kannan, A. (2004). Intelligent Multi-Agent based Back-Propagation Neural Network forecasting model for statistical database anomaly prevention system. In *ICISIP*, pages 108–113. IEEE.
- Saner, C. B., Trivedi, A., and Srinivasan, D. (2022). A Cooperative Hierarchical Multi-Agent System for EV Charging Scheduling in Presence of Multiple Charging Stations. *IEEE Transactions on Smart Grid*, 13(3):2218–2233.
- Shahriar, S., Al-Ali, A. R., Osman, A. H., Dhou, S., and Nijim, M. (2020). Machine Learning Approaches for EV Charging Behavior: A Review. *IEEE Access*, 8:168980–168993.
- Shahriar, S., Al-Ali, A. R., Osman, A. H., Dhou, S., and Nijim, M. (2021). Prediction of ev charging behavior using machine learning. *IEEE Access*, 9:111576–111586.
- Zhai, Z., Su, S., Liu, R., Yang, C., and Liu, C. (2019). Agent-cellular automata model for the dynamic fluctuation of ev traffic and charging demands based on machine learning algorithm. *Neural Computing and Applications*, 31:4639–4652.