

## Direção assistida por agentes cognitivos: Uma análise de viabilidade para o controle de velocidade

Ricardo de Monteiro e Tavares<sup>1</sup>, Carlos Eduardo Pantoja<sup>2</sup>,  
Nilson Mori Lazzarin<sup>1,2</sup>, Flávia Cristina Bernadini<sup>1</sup>

<sup>1</sup>Instituto de Computação - Universidade Federal Fluminense - Niterói, RJ – Brasil

<sup>2</sup>Centro Federal de Educação Tecnológica Celso Suckow da Fonseca (Cefet/RJ)  
Rio de Janeiro, RJ – Brasil

ricardotavares@id.uff.br

**Abstract.** *This paper presents a study on the feasibility of using Embedded Multi-Agent Systems for vehicle autonomous speed control. The system is implemented in a prototype equipped with Raspberry Pi and Arduino boards, using RFID tags to read traffic signs. The prototype reads the tags and adjusts the speed according to the data received. The experimental results demonstrate the viability of the proposed model despite some technical limitations, suggesting that future improvements may include more powerful RFID readers and the incorporation of encryption for greater security.*

**Resumo.** *Este trabalho apresenta um estudo sobre a viabilidade do uso de Sistemas Multiagente Embarcados para o controle autônomo de velocidade em veículos. Utilizando tags RFID para leitura de sinalizações de trânsito, o sistema é implementado em um protótipo equipado com um Raspberry Pi e placas Arduino. O protótipo lê as tags e ajusta a velocidade conforme os dados recebidos. Os resultados experimentais demonstram a viabilidade do modelo proposto, apesar de algumas limitações técnicas, sugerindo que futuras melhorias podem incluir leitores RFID mais potentes e a incorporação de criptografia para maior segurança.*

### 1. Introdução

Atualmente a fiscalização da velocidade veicular é feita por radares, seja de forma fixa ou móvel. Os radares fixos ficam instalados em um determinado ponto, para medir a velocidade em um trecho da via, para medir a velocidade média no trecho. Já os radares móveis, são agentes de trânsito ou policiais rodoviários que vão para um determinado ponto da via e realizam o monitoramento por meio de ondas eletromagnéticas por meio de uma pistola de medição [Lara 2020]. Porém, a fiscalização de velocidade veicular por meio de radares torna-se ineficiente quando há uma grande quilometragem de vias para fiscalizar, porque necessitaria de um número considerável de radares e isto tem impacto nos custos de operacionalização [Redação Mobilidade 2022]. Diante deste cenário, este trabalho visa analisar a viabilidade do uso de tecnologias baseadas em agentes, para permitir que o próprio veículo faça, de forma autônoma e/ou colaborativa, este controle de velocidade a partir da leitura das sinalizações presentes nas vias.

Agentes são entidades autônomas da inteligência artificial distribuída, capazes de perceberem seu ambiente ao redor e, de forma autônoma e proativa, tomarem decisões baseados em seus próprios objetivos e percepções, que coexistem em um sistema multiagente (SMA), podendo ter objetivos comuns, conflitantes ou sistêmicos [Wooldridge 2009]. Agentes e SMAs vem sendo utilizados em soluções embarcadas para prover autonomia a protótipos de veículos autônomos terrestres [Barros. et al. 2014, Souza de Castro et al. 2018] através do uso de extensões do Jason [Bordini et al. 2007] para programação de agentes em sistemas embarcados [Brandão et al. 2021, Lazarin et al. 2024] e simulados [Heijmeijer and Vaz Alves 2018, Souza de Castro et al. 2022]. O JasonEmbedded [Pantoja et al. 2023] é uma versão do Jason com potencial de uso em sistemas embarcados, permitindo o uso de arquitetura específicas de agentes embarcados capazes de gerenciar microcontroladores, que adota o modelo cognitivo Belief-Desire-Intention (BDI) [Bratman et al. 1988], adicionando uma camada de cognição próxima ao pensamento humano em sistemas embarcados.

O objetivo deste trabalho é analisar a viabilidade do uso de SMA embarcados para no controle de velocidade de forma que um veículo, de forma autônoma, consiga ler os dados das sinalizações presentes nas vias e realize o controle da velocidade veicular de forma condizente com os dados lidos da sinalização. Para tal, será criado um protótipo de um veículo terrestre autônomo utilizando-se componentes eletrônicos, um computador de placa única Raspberry Pi e duas placas Arduino. O protótipo terá um SMA embarcado utilizando o framework JasonEmbedded e suas extensões para sistemas embarcados. As placas de trânsito serão representadas por tags RFID (*Radio Frequency Identification*) indicando a velocidade permitida pela via. Serão realizados experimentos para garantir a leitura da tag RFID, a comunicação entre os agentes e o processo de controle de velocidade. Os experimentos buscam mostrar a viabilidade, em ambiente experimental, do potencial da abordagem.

Este trabalho está dividido da seguinte forma: na Seção 2 será a apresentada a metodologia proposta do trabalho; na Seção 3 serão apresentados o estudo de caso e os experimentos realizados; por fim, na Seção 4 será apresentada a conclusão.

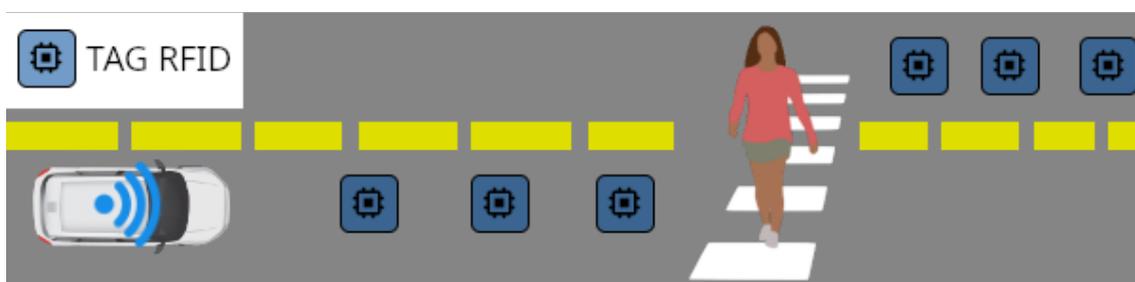
## **2. Metodologia proposta para controle de velocidade usando RFID**

O controle de velocidade em estradas é um procedimento adotado em diversos países. Atualmente, os radares de velocidade identificam em pontos estáticos ao longo da estrada se um determinado veículo passou ou não de uma velocidade pré-determinada, a fim de regular a velocidade praticada. Contudo, a aderência ou não a velocidade pré-determinada depende da disponibilidade do condutor e da visibilidade da sinalização. Além disso, a identificação do veículo pode ser burlada como a troca da placa ou a partir da inserção de objeto que dificultem a identificação da placa.

Ao adotar um mecanismo de direção assistida, a transferência da responsabilidade de controle de velocidade é transferida para o sistema veicular. Neste trabalho, visamos aplicar agentes cognitivos embarcados, que através de suas percepções do mundo físico, onde podem ser utilizadas etiquetas RFID, por exemplo, o agente pode auxiliar o motorista, emitindo alerta, ou até mesmo, reduzir a velocidade do veículo de forma autônoma.

A tecnologia RFID já é utilizada em diversas aplicações, tais como controle Pa-

rimonial e logístico, utilização em pedágios, e autenticação de funcionários. As sinalizações de trânsito serão confeccionadas respeitando as regras atuais do Código de Trânsito Brasileiro (CTB). As tags RFID conterão as mesmas informações que a placa apresenta visualmente e a instalação poderia ser feita no asfalto da via. As tags serão do tipo passivo, o que permite com que os sistemas, inicialmente, não dependam de baterias ou acesso à energia elétrica, diminuindo assim o custo de implantação. A cada vez que um veículo identificar uma sinalização, ela encaminhará a percepção ao SMA, que ficará responsável por atender ao especificado pela sinalização através do controle de velocidade e do limitador de frenagem. O leitor de RFID, faz a captura dos dados contidos nas tags através das ondas de radiofrequência, permitindo a leitura em qualquer posição, desde que, esteja no raio de leitura [Alecrim and Marques 2023]. A Figura 1 exemplifica como ficaria no mundo físico.



**Figura 1. Ilustração de instalação das tags RFID nas vias urbanas.**

Existem dois tipos de chip RFID, o passivo e o ativo. O chip passivo, só irá transmitir os dados quando estiver no raio do leitor, pois o mesmo se aproveita das ondas de rádio frequência, gerado pela antena do leitor, para se energizar. Já os chips ativos, funcionam semelhantemente, porém possui uma bateria acoplada neles para haver uma transmissão constante e unidirecionalmente. A distância para a leitura varia conforme o tipo e a frequência utilizada. O modelo passivo, a distância da leitura, pode chegar a 15 metros, enquanto o ativo, pode chegar a 300 metros. Para o desenvolvimento deste sistema, utilizamos o chip RFID do tipo passivo, devido ao não uso de baterias. O que removeria os custos necessários para manter a bateria com carga.

Na Figura 2 é apresentada a arquitetura do sistema embarcado no veículo. A arquitetura física do sistema proposto é composto pelas seguintes unidades: Unidade Leitora (UL); Unidades Externas (UEX) e Unidade Controladora (UC).

- A UL é composta pelo leitor de RFID, que faz a leitura dos dados presentes nas tags RFID instaladas no asfalto.
- A UC é composta por um Arduino e um Raspberry Pi. O Arduino fará o papel de comunicação entre as UEX, a UL e o Raspberry Pi. A comunicação feita entre o Raspberry Pi e o Arduino é realizado por meio de comunicação serial usando o Javino [Lazarin and Pantoja 2015], que envia as informações para o SMA em execução pelo JasonEmbedded, localizado no Raspberry Pi.
- As UEX é composta por uma ou mais dispositivos, sensores e atuadores, que estão instalados nos veículos que percebem o ambiente e enviam para os agentes embarcados na UC, ou atuam no ambiente, conforme deliberação dos agentes na UC.

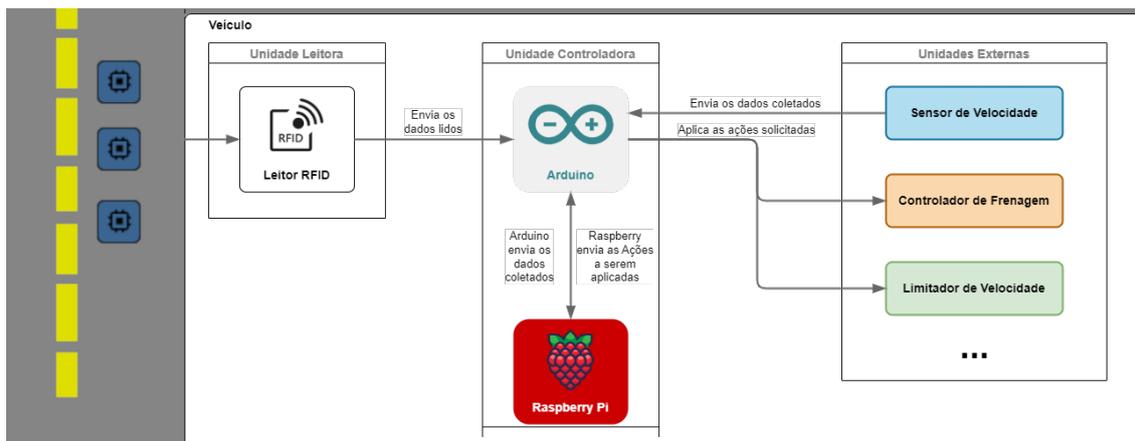


Figura 2. A arquitetura física de um veículo considerando a necessidade do controle de velocidade autônoma.

### 3. Estudo de Caso

O estudo de caso consiste em reproduzir em um ambiente controlado uma estrada com diferentes limites de velocidades (Baixa, Média e Alta) e conforme o veículo for trafegando, este deve obedecer à sinalização. Para isso foram realizados três experiências. A primeira experiência compreende a validação da leitura dos dados do chip RFID pelo agente. A segunda experiência, compreende a comunicação entre agentes e a realização de ações do agente para o protótipo após a leitura do chip RFID. A terceira, e última, experiência compreende do protótipo ler vários chips RFID ao longo de um circuito e realizar as ações de adequação de velocidade.

Para a execução do estudo de caso foi utilizado um *Cognitive Hardware On Network Basic Prototype (ChonBot) 2WD* [Lazarin et al. 2023], o qual foi adaptado, adicionando um Arduino Uno R3, um Hub USB, um leitor de RFID modelo MFRC522. O protótipo é apresentado na Figura 3.



Figura 3. O Protótipo (ChonBot 2WD - adaptado).

O leitor RFID modelo MFRC522 foi escolhido por ser de baixo custo e dimensões pequenas. O leitor possui frequência de 13.56MHz e pode ler tags RFID em até 4 centímetros de distância. Um dos Arduinos é dedicado ao leitor RFID e o outro para o circuito de controle das funções básicas de movimentação, iluminação e sensoriamento de distância e luminosidade. O Hub USB permite a comunicação entre o RaspberryPi e os Arduinos. O circuito é descrito na Figura 4.

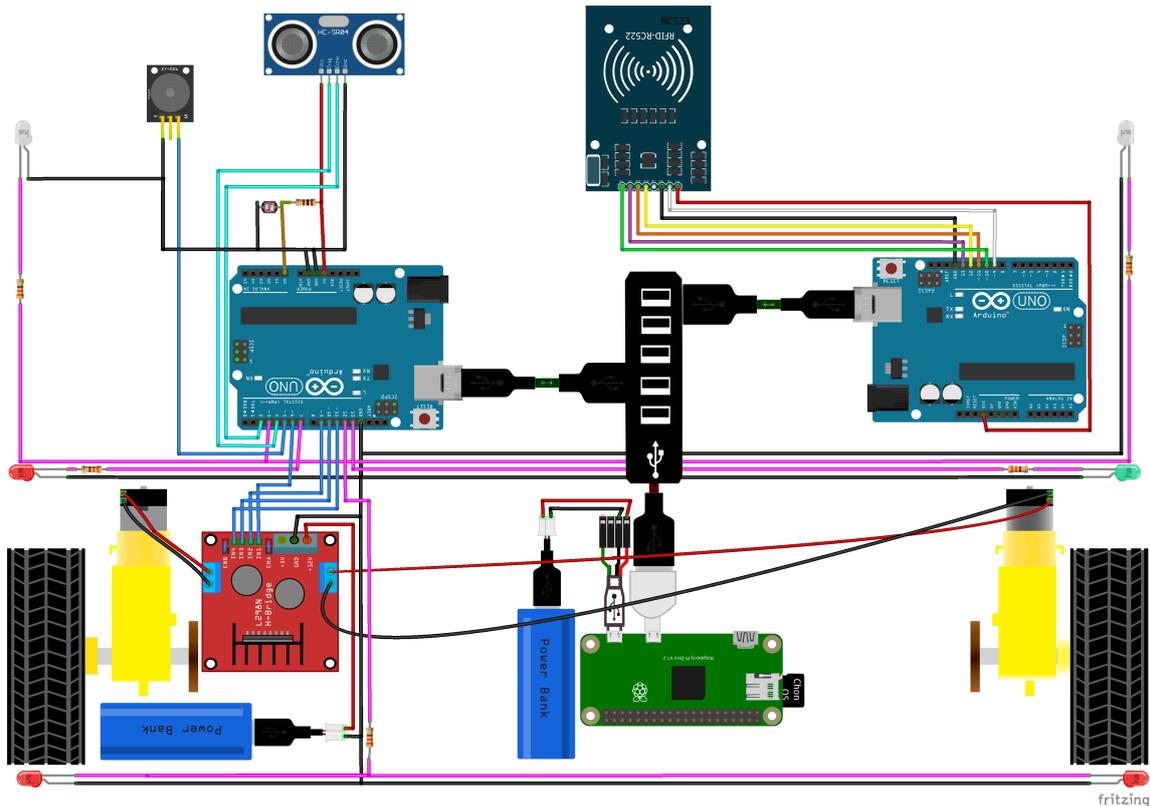


Figura 4. O Circuito do protótipo.

O Sistema multiagente desenvolvido contém dois agentes do tipo ARGO [Pantoja et al. 2016] — o agente *pilot* e o *copilot*. O *pilot* é responsável por enviar os comandos para serem executados pelas UEx, já o agente *copilot* faz as solicitações de informações providas das UEx e da UL e informa para o agente *pilot* a informação atual do ambiente. Os códigos do raciocínio dos agentes é apresentado abaixo:

```

1 !start.
3 +!start <- .port(ttyACM0); .percepts(open).
5 +!stopAction[source(copilot)]<-
   .act(stop); .act(breakLOn); .act(lightOff).
7
9 +!speedLimiterAction(X)[source(copilot)]
: X == med <-
   .act(speedM); .act(lightOn); .act(goAhead); .act(breakLOff).
11
13 +!speedLimiterAction(X)[source(copilot)]
: X == low <-
   .act(speedL); .act(lightOnL); .act(goAhead); .act(breakLOff).
15
17 +!speedLimiterAction(X)[source(copilot)]
: X == high <-
   .act(speedH); .act(lightOnH); .act(goAhead); .act(breakLOff).

```

Código-fonte 1. Raciocínio do agente *pilot*.

```

1 !start.
2
3 +!start <- .port(ttyACM1); .percepts(open); .send(pilot, achieve, speedLimiterAction(high)).
4
5 +roadAction(spdhigh) [source(percept)] <- .send(pilot, achieve, speedLimiterAction(high)).
6
7 +roadAction(spdmed) [source(percept)] <- .send(pilot, achieve, speedLimiterAction(med)).
8
9 +roadAction(spdlow) [source(percept)] <- .send(pilot, achieve, speedLimiterAction(low)).
10
11 +roadAction(stop) [source(percept)] <- .send(pilot, achieve, stopAction).

```

### Código-fonte 2. Raciocínio do agente *copilot*.

O código do firmware da unidade leitora de RFID adicionada ao protótipo é apresentado abaixo:

```

1 #include <Javino.h> /* https://github.com/chon-group/javino2arduino/releases/latest */
2 #include <SPI.h> /* https://www.arduino.cc/reference/en/language/functions/communication/spi/ */
3 #include <MFRC522.h> /* https://github.com/miguelbalboa/rfid/releases/latest */
4 #define RFIDresetPin 9 /* https://github.com/miguelbalboa/rfid#pin-layout */
5 #define RFIDsdPin 10 /* https://github.com/miguelbalboa/rfid#pin-layout */
6 #define BLOCKREAD 2
7
8 Javino javino;
9 MFRC522 rfid(RFIDsdPin, RFIDresetPin); // Instancia da classe do leitor
10 MFRC522::MIFARE_Key key={0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF};
11
12 void setup() {
13     javino.start(9600); /* Inicia a comunicacao Serial */
14     SPI.begin(); /* Inicia a comunicacao SPI */
15     rfid.PCD_Init(); /* Inicia o MFRC522 */
16 }
17
18 void serialEvent(){ javino.readSerial(); }
19
20 void loop(){
21     if(javino.availableMsg() && javino.requestPercepts()){javino.sendPercepts();}
22     lerCartao();
23 }
24
25 void lerCartao(){
26     if(rfid.PICC_IsNewCardPresent() && rfid.PICC_ReadCardSerial()){ /*Funcao que verifica se alguma tag foi lida*/
27         MFRC522::PICC_Type tag = rfid.PICC_GetType(rfid.uid.sak);
28         if(tag == MFRC522::PICC_TYPE_MIFARE_MINI || tag == MFRC522::PICC_TYPE_MIFARE_1K || tag == MFRC522::
29             PICC_TYPE_MIFARE_4K){
30             byte bufferLen = 18;
31             byte readBlockData[18];
32             MFRC522::StatusCode auth = rfid.PCD_Authenticate(MFRC522::PICC_CMD_MF_AUTH_KEY_A, BLOCKREAD, &key, &(rfid.uid));
33             MFRC522::StatusCode read = rfid.MIFARE_Read(BLOCKREAD, readBlockData, &bufferLen);
34             if(auth == MFRC522::STATUS_OK && read == MFRC522::STATUS_OK){
35                 String strValue = String((char *)readBlockData);
36                 javino.addPercept("roadAction("+strValue+"");
37             }
38             rfid.PICC_HaltA();
39             rfid.PCD_StopCrypto1();
40         }
41     }

```

### Código-fonte 3. Firmware do leitor RFID.

## 3.1. Experimentos

O objetivo do primeiro experimento é verificar se o agente recebe os dados lidos de uma tag. Para realizar o experimento, foram desenvolvidos o código para a leitura das tags RFID, e um agente ARGO, programado na ChonIDE [Souza de Jesus et al. 2023], que solicita ao Arduino, por meio do Javino, a leitura da placa e informa uma mensagem na tela informando qual placa foi lida. Neste experimento, todas as leituras realizadas e mensagens informadas pelo agente, por intermédio do Arduino, corresponderam, fielmente, aos dados presentes nos chips.

O objetivo do segundo experimento é verificar se é possível realizar a comunicação entre os agentes *pilot* e *copilot*, e se o agente *pilot* consegue controlar o protótipo. Para realizar este experimento, será adicionado um novo agente ao código do experimento anterior, que passará a receber as informações do agente já desenvolvido, realizando ações

de controle de intensidade do motor, ajustando a intensidade de luminosidade e desligando os faróis dianteiros e ligando e desligando as luzes de freio. O agente *copilot*, desenvolvido na experiência anterior, sofrerá modificações para permitir a comunicação com o novo agente desenvolvido. O agente *pilot* conseguia controlar o protótipo por meio das ações, já previamente configuradas. Neste experimento, a comunicação entre os agentes foi realizado com sucesso, sendo necessário apenas algumas correções no código do agente *copilot* para o uso do método adequado de comunicação para o *pilot*, e todas as ações feitas pelo *pilot* foram executadas com sucesso.

Para o terceiro experimento, o protótipo passa a trafegar em um circuito em que consiste em uma via reta com algumas tags RFID dispostos uniformemente ao longo da via. As tags foram inseridas na superfície da via e a sequência de leitura foi proposto da seguinte forma: lê-se primeiramente a placa de Velocidade Média, depois lê-se as placas de Velocidade Alta, Velocidade Baixa e de Pare respectivamente. Neste experimento, o sistema conseguiu ler e aplicar as ações determinadas no programa, porém houve alguns problemas durante as execuções destes experimentos. Os problemas foram causados devido a uma inconsistência nas rotações dos motores do protótipo. Devido a isto, o carrinho não andava em linha reta, mas levemente inclinado à esquerda, causando problemas de leitura das tags RFID instalados no circuito. Pois quando o protótipo passa pelo chip RFID a alguns milímetros ao lado do leitor, devido à baixa potência do leitor RFID, o sistema não conseguia realizar a leitura do chip e, conseqüentemente, a não aplicação das ações configuradas para o trecho do circuito.

Na Figura 5 são apresentadas algumas imagens da realização dos experimentos, também disponíveis<sup>1</sup> em vídeo.



Figura 5. Imagens da realização dos experimentos.

<sup>1</sup><https://papers.chon.group/WESAAC/2024/direcaoAssistida/>

## 4. Conclusão

Este trabalho apresentou um estudo de viabilidade do uso de SMA Embarcado no controle de velocidade veicular por meio de um protótipo e tags RFID. As experiências realizadas mostraram que o modelo apresentado, apesar dos problemas ocorridos durante a experiência número 3, produziu resultados gerais satisfatórios. Logo, é possível que um veículo, de forma autônoma, consiga ler os dados das sinalizações presentes nas vias e realize o controle da velocidade veicular de forma condizente com os dados lidos da sinalização.

Para trabalhos futuros, se faz necessário a utilização de um leitor de RFID mais potente ou um chip RFID com antena que permita com que os veículos realizem a leitura correta dos dados inseridos, independentemente da posição do veículo na via. Também deve-se atentar para a questão da segurança, com a aplicação de criptografia no chip RFID, para permitir apenas os agentes autorizados façam a reescrita dos dados presentes nos cartões.

Pode-se avaliar também a incorporação de sistemas colaborativos no sistema atual mediante interações entre os veículos conectados e as sinalizações de trânsito, por meio de um sistema de contratos e recompensas para informar aos motoristas sobre algum incidente que tenha ocorrido a metros ou quilômetros à frente, deste modo evitando acidentes e/ou engarrafamentos indesejados. Outra linha de pesquisa seria utilizar o JaCaMo [Boissier et al. 2013] e adotar o modelo organizacional Moise para fazer o gerenciamento das normativas de velocidade de trânsito.

## Referências

- Alecrim, E. and Marques, A. (2023). O que é RFID? Saiba como funciona essa tecnologia de conexão | Tecnoblog. Disponível em <https://tecnoblog.net/responde/o-que-e-rfid-entenda-como-funciona-essa-tecnologia/>. Acesso em 08/07/2024.
- Barros., R. S., Heringer., V. H., Pantoja., C. E., Lazarin., N. M., and de Moraes., L. M. (2014). An Agent-oriented Ground Vehicle's Automation using Jason Framework. In *Proceedings of the 6th International Conference on Agents and Artificial Intelligence - Volume 2: ICAART*, pages 261–266. INSTICC, SciTePress. DOI: <https://doi.org/10.5220/0004917102610266>.
- Boissier, O., Bordini, R. H., Hübner, J. F., Ricci, A., and Santi, A. (2013). Multi-agent oriented programming with JaCaMo. *Science of Computer Programming*, 78(6):747–761. Special section: The Programming Languages track at the 26th ACM Symposium on Applied Computing (SAC 2011) Special section on Agent-oriented Design Methods and Programming Techniques for Distributed Computing in Dynamic and Complex Environments. DOI: <https://doi.org/10.1016/j.scico.2011.10.004>.
- Bordini, R., Hübner, J., and Wooldridge, M. (2007). *Programming Multi-Agent Systems in AgentSpeak using Jason*. Wiley Series in Agent Technology. Wiley.
- Brandão, F. C., Lima, M. A. T., Pantoja, C. E., Zahn, J., and Viterbo, J. (2021). Engineering Approaches for Programming Agent-Based IoT Objects Using the Resource Management Architecture. *Sensors*, 21(23). DOI: <https://doi.org/10.3390/s21238110>.

- Bratman, M. E., Israel, D. J., and Pollack, M. E. (1988). Plans and resource-bounded practical reasoning. *Computational Intelligence*, 4(3):349–355. DOI: <https://doi.org/10.1111/j.1467-8640.1988.tb00284.x>.
- Heijmeijer, A. V. H. and Vaz Alves, G. (2018). Development of a Middleware between SUMO simulation tool and JaCaMo framework. *ADCAIJ: Advances in Distributed Computing and Artificial Intelligence Journal*, 7(2):5–15. DOI: <https://doi.org/10.14201/ADCAIJ201872515>.
- Lara, R. (2020). Fiscais das ruas: radares de velocidade são mesmo precisos? | Tilt UOL. Disponível em <https://www.uol.com.br/tilt/noticias/redacao/2020/11/19/fiscais-das-ruas-como-funcionam-os-radares-de-velocidade.htm>. Acessado em 22/07/2023.
- Lazarin, N., Pantoja, C., and Viterbo, J. (2023). Towards a Toolkit for Teaching AI Supported by Robotic-agents: Proposal and First Impressions. In *Anais do XXXI Workshop sobre Educação em Computação*, pages 20–29, Porto Alegre, RS, Brasil. SBC. DOI: <https://doi.org/10.5753/wei.2023.229753>.
- Lazarin, N. M. and Pantoja, C. E. (2015). A robotic-agent platform for embedding software agents using Raspberry Pi and Arduino Boards. In *Proceedings of the 9th Workshop-School on Agents, Environments, and Applications (WESAAC 2015)*, pages 13–20, Niteroi. UFF. <http://www2.ic.uff.br/~wesaac2015/Proceedings-WESAAC-2015.pdf>.
- Lazarin, N. M., Pantoja, C. E., and Viterbo, J. (2024). Dealing with the unpredictability of physical resources in real-world multi-agent systems. In Rocha, A. P., Steels, L., and van den Herik, J., editors, *Agents and Artificial Intelligence*, pages 48–71, Cham. Springer Nature Switzerland. DOI: [https://doi.org/10.1007/978-3-031-55326-4\\_3](https://doi.org/10.1007/978-3-031-55326-4_3).
- Pantoja, C. E., Jesus, V. S. d., Lazarin, N. M., and Viterbo, J. (2023). A Spin-off Version of Jason for IoT and Embedded Multi-Agent Systems. In Naldi, M. C. and Bianchi, R. A. C., editors, *Intelligent Systems*, Lecture Notes in Computer Science, pages 382–396, Cham. Springer Nature Switzerland. [https://doi.org/10.1007/978-3-031-45368-7\\_25](https://doi.org/10.1007/978-3-031-45368-7_25).
- Pantoja, C. E., Stabile, M. F., Lazarin, N. M., and Sichman, J. S. (2016). ARGO: An Extended Jason Architecture that Facilitates Embedded Robotic Agents Programming. In Baldoni, M., Müller, J. P., Nunes, I., and Zalila-Wenkstern, R., editors, *Engineering Multi-Agent Systems*, pages 136–155, Cham. Springer International Publishing. [https://doi.org/10.1007/978-3-319-50983-9\\_8](https://doi.org/10.1007/978-3-319-50983-9_8).
- Redação Mobilidade (2022). Reduzir a velocidade dos carros piora o congestionamento? | Mobilidade Estadão. Disponível em: <https://mobilidade.estadao.com.br/mobilidade-com-seguranca/reduzir-a-velocidade-dos-carros-piora-o-congestionamento/> Acessado em: 22/07/2023.
- Souza de Castro, L., Borges, A., and Vaz Alves, G. (2018). Developing a smart parking solution based on a Holonic Multiagent System using JaCaMo Framework. In *Proceedings of the 12th Workshop-School on Agents, Environments, and Applications (WESAAC 2018)*, Fortaleza. UFC. <https://scholar.google.com.br/scholar?cluster=16564013293278034919>.

- Souza de Castro, L. F., Manoel, F. C. P. B., Souza de Jesus, V., Pantoja, C. E., Pinz Borges, A., and Vaz Alves, G. (2022). Integrating Embedded Multiagent Systems with Urban Simulation Tools and IoT Applications. *Revista de Informática Teórica e Aplicada*, 29(1):81–90. DOI: <https://doi.org/10.22456/2175-2745.110837>.
- Souza de Jesus, V., Mori Lazarin, N., Pantoja, C. E., Vaz Alves, G., Ramos Alves de Lima, G., and Viterbo, J. (2023). An IDE to Support the Development of Embedded Multi-Agent Systems. In Mathieu, P., Dignum, F., Novais, P., and De la Prieta, F., editors, *Advances in Practical Applications of Agents, Multi-Agent Systems, and Cognitive Mimetics. The PAAMS Collection*, pages 346–358, Cham. Springer Nature Switzerland. DOI: [https://doi.org/10.1007/978-3-031-37616-0\\_29](https://doi.org/10.1007/978-3-031-37616-0_29).
- Wooldridge, M. (2009). *An introduction to multiagent systems*. John wiley & sons.