

Intelligent Monitoring of Houseplants Using Embedded AI Agents

Rafael Carvalho, João Penna, Fernanda Pinho, Bruno Freitas, Nilson Lazarin

Bacharelado em Engenharia Elétrica
Centro Federal de Educação Tecnológica Celso Suckow da Fonseca (CEFET/RJ)
Nova Friburgo, RJ — Brasil

rafael.campos@aluno.cefet-rj.br

Abstract. *This paper proposes the development of an embedded artificial intelligence agent for monitoring household plants and communicating their needs to caregivers. By collecting and analyzing variables that are essential to plant health and growth, the system can provide recommendations and alerts in natural language. Furthermore, the interaction between the agent and the user aims to make plant care a more accessible, engaging, and interactive activity, contributing to the promotion of gardening habits in domestic environments.*

Resumo. *Este artigo propõe o desenvolvimento de um agente de inteligência artificial embarcado para o monitoramento de plantas domiciliares e a comunicação de suas necessidades aos cuidadores. Por meio da coleta e análise de variáveis essenciais ao desenvolvimento saudável da planta, o sistema é capaz de fornecer recomendações e alertas em linguagem natural. Além disso, a interação entre o agente e o usuário busca tornar o cuidado com plantas uma atividade mais acessível, engajadora e interativa, contribuindo para a promoção de hábitos de cultivo em ambientes domésticos.*

1. Introdução

Sistemas Multiagentes (SMA) são compostos por múltiplos agentes autônomos capazes de perceber alterações no ambiente, tomar decisões de forma independente e interagir entre si para alcançar objetivos individuais ou coletivos [Wooldridge 2009]. Essa abordagem permite o desenvolvimento de sistemas inteligentes distribuídos, nos quais diferentes agentes cooperam para resolver problemas complexos, sendo amplamente aplicada em áreas como robótica, automação, internet das coisas (IoT) e sistemas embarcados. Atividades como leitura de dados, interpretação de sinais e tomada de decisões baseadas em informações do ambiente já podem ser realizadas por sistemas inteligentes de maneira semelhante à atuação humana. Entretanto, embora a automação proporcione ganhos de eficiência, sua adoção pode reduzir o envolvimento do usuário e tornar a interação menos significativa, especialmente quando o sistema elimina sua participação nas decisões e ações do processo [Norman 2013].

Diferentemente das soluções tradicionais, que se concentram na automação da irrigação e na apresentação de dados ambientais, este trabalho propõe o desenvolvimento de uma arquitetura baseada em agentes inteligentes embarcados para o gerenciamento dos cuidados necessários à manutenção de plantas em ambiente doméstico. A arquitetura busca manter o usuário como parte ativa do processo de cuidado da planta, utilizando

agentes cooperativos para interpretar continuamente variáveis ambientais, como temperatura, umidade e luminosidade, identificar as necessidades da planta e comunicá-las ao usuário por meio de linguagem natural. Nesse contexto, o uso de agentes possibilita que o monitoramento ocorra de forma contínua e autônoma, reduzindo a necessidade de o usuário acompanhar constantemente os dados dos sensores. A partir da interpretação das informações coletadas, o sistema pode identificar situações que demandem atenção e notificar o usuário, por exemplo, por meio do envio de mensagens de alerta. Dessa forma, em vez de apenas disponibilizar os dados ambientais, como ocorreria em uma abordagem baseada somente na aquisição e apresentação das informações pelo Arduino, o sistema busca interpretar essas informações e comunicar ao usuário quando uma intervenção for necessária. Dessa forma, pretende-se tornar a interação mais envolvente, incentivando a participação ativa do usuário e aumentando as chances de sobrevivência e desenvolvimento saudável da planta, que passa a “expressar” suas necessidades por meio do sistema.

Para viabilizar essa arquitetura em um ambiente doméstico, serão empregadas tecnologias de agentes inteligentes embarcados executadas em um módulo Raspberry Pi, plataforma amplamente utilizada em aplicações de monitoramento e agricultura de precisão devido à sua capacidade de processamento local, integração com sensores e suporte à execução de algoritmos inteligentes [Joice et al. 2025]. Sobre essa plataforma, os agentes serão organizados em uma arquitetura de Sistema Multiagentes (SMA), responsável pelo monitoramento, processamento das informações e comunicação com o usuário.

Este artigo está organizado da seguinte forma: a Seção 2 apresenta a metodologia e a arquitetura proposta; a Seção 3 descreve a implementação do sistema; e, por fim, a Seção 4 apresenta as conclusões do trabalho.

2. Metodologia e Arquitetura

Esta seção apresenta a arquitetura proposta para o monitoramento inteligente de plantas em ambiente doméstico. A solução foi desenvolvida segundo uma arquitetura de Sistema Multiagentes (SMA), na qual agentes cooperativos são responsáveis pela aquisição, processamento e interpretação das informações provenientes dos sensores, bem como pela comunicação das necessidades da planta ao usuário. A Figura abaixo apresenta uma visão geral da arquitetura do sistema.

A arquitetura proposta foi implementada em um protótipo funcional, utilizado para validar a integração entre os componentes de hardware e software. A Figura 2 apresenta o sistema desenvolvido, no qual é possível observar o ambiente monitorado, os sensores responsáveis pela aquisição das variáveis ambientais, o Arduino utilizado para coleta das leituras e o Raspberry Pi (com o sistema chonOS [Lazarin et al. 2026]), responsável pela execução do Sistema Multiagente.

No protótipo desenvolvido, foi considerada uma planta genérica, de forma que o sistema pudesse validar a arquitetura proposta independentemente das necessidades específicas de cada espécie.

2.1. Arquitetura do Sistema

Conforme ilustrado na Figura 1, a arquitetura proposta é composta por duas camadas principais: uma camada de aquisição de dados e uma camada de processamento inteligente. A camada de aquisição é responsável pela coleta das variáveis ambientais por

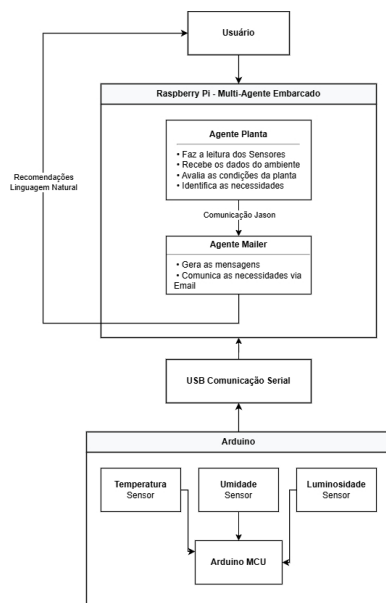


Figura 1. Arquitetura proposta do sistema.

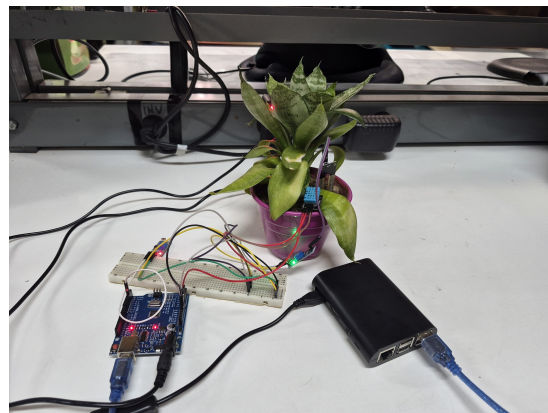


Figura 2. Protótipo desenvolvido para validação do sistema.

meio de sensores conectados ao microcontrolador Arduino. As leituras de temperatura, umidade do solo e luminosidade são transmitidas ao Raspberry Pi por comunicação serial via interface USB, onde ocorre a execução do Sistema Multiagentes. Essa divisão de responsabilidades permite utilizar o Arduino como unidade dedicada à aquisição de dados, enquanto o Raspberry Pi concentra o processamento, a inferência e a execução dos agentes, explorando sua maior capacidade computacional.

O Sistema Multiagentes é composto por dois agentes cooperativos com responsabilidades distintas. O Agente Planta recebe continuamente as percepções provenientes do ambiente, atualiza sua base de crenças e executa regras de inferência para selecionar os planos apropriados de acordo com o estado atual da planta. Além da classificação das condições de temperatura, umidade do solo e luminosidade, esse agente mantém um histórico das leituras realizadas, permitindo a geração de alertas e relatórios baseados não apenas em uma única observação, mas também no comportamento das variáveis ao longo do tempo.

Os alertas gerados pelo Agente Planta são encaminhados ao Agente Mailer por meio da troca de mensagens entre agentes do framework Jason. Esse agente é responsável por transformar os eventos recebidos em mensagens compreensíveis ao usuário, realizando o envio de alertas e relatórios periódicos por correio eletrônico.

Essa divisão de responsabilidades permite que cada agente desempenhe uma função específica dentro do sistema. Enquanto o Agente Planta concentra o monitoramento e a tomada de decisão, o Agente Mailer é responsável por informar o usuário.

2.2. Funcionamento da Arquitetura

O funcionamento do sistema inicia-se com a leitura periódica dos sensores de temperatura, umidade do solo e luminosidade conectados ao Arduino. As informações coletadas

são enviadas ao Raspberry Pi por meio da comunicação serial USB, onde são recebidas pelo Agente Planta.

Sempre que novas percepções são recebidas, o agente atualiza sua base de crenças e executa as regras de inferência definidas para interpretar o estado da planta. Além de analisar as leituras mais recentes, o agente mantém um histórico das variáveis monitoradas, permitindo identificar situações que persistem ao longo do tempo, como solo seco ou encharcado por longos períodos, temperaturas inadequadas ou tempo insuficiente de exposição à luz.

Quando essas condições são identificadas, ou quando chega o momento de gerar o relatório periódico, o Agente Planta envia uma mensagem ao Agente Mailer utilizando o mecanismo de troca de mensagens do framework Jason. O Agente Mailer recebe essas informações e é responsável por enviar ao usuário alertas e relatórios por e-mail ao usuário.

Esse processo é executado continuamente durante o funcionamento do sistema, permitindo acompanhar as condições ambientais em tempo real e informar ao usuário sempre que forem identificadas situações que exijam sua atenção.

3. Implementação do Sistema

A implementação do sistema foi realizada com base no framework Jason, utilizado para o desenvolvimento de Sistemas Multiagentes (SMA). Nesse contexto, foram empregadas as bibliotecas Argo [Pantoja et al. 2023] e Javino [Lazarin and Pantoja 2015], responsáveis por viabilizar a integração entre os agentes e os dispositivos físicos utilizados no sistema. O desenvolvimento dos agentes ocorreu no ambiente ChonIDE [Souza De Jesus et al. 2023], o qual oferece suporte completo à criação, execução e depuração de sistemas baseados em agentes.

Na sequência, são apresentados os principais elementos da implementação, incluindo o código do arduino, a modelagem do conhecimento por meio de crenças, as regras de inferência utilizadas pelo agente e os mecanismos de comunicação entre os componentes do sistema.

3.1. Arduino e aquisição de dados

O Arduino é responsável pela aquisição dos dados provenientes dos sensores utilizados no sistema. A partir das leituras de umidade do solo, temperatura e luminosidade, os dados são enviados à Raspberry Pi por meio da biblioteca Javino. Na Raspberry Pi, os dados recebidos são utilizados como percepções pelos agentes desenvolvidos no ambiente ChonIDE. Parte do código implementado para o Arduino é apresentada no Listing 1, enquanto o código completo está disponível no GitHub, cujo link encontra-se disponibilizado nas referências deste artigo.

```
1 void percebe() {  
  // --- 1. LEITURA DOS SENSORES ---  
3  
  // Umidade do Solo (Analogico no A0)  
5  int umidadeBruta = analogRead(A0);  
  int umidadeSolo = map(umidadeBruta, 1023, 375, 0, 100);  
7  umidadeSolo = constrain(umidadeSolo, 0, 100);  
9  
  // Temperatura do Ar (DHT11 no A1)  
  float tempAr = dht.readTemperature();  
11  
  // Luz (Digital no D2)  
13  int luz = digitalRead(2);
```

```

15 // --- 2. ENVIO DOS DADOS ---
17 javino.addPercept("umidadeSolo(" + String(umidadeSolo) + ")");
    javino.addPercept("temperatura(" + String(tempAr) + ")");
19 javino.addPercept("luz(" + String(luz) + ")");
    }

```

Listing 1. Código do Arduino para leitura dos sensores e comunicação com o Agente Planta

As leituras obtidas pelo Arduino são, então, utilizadas pelos agentes para atualizar suas crenças e realizar a classificação das condições da planta, conforme apresentado na seção a seguir.

3.2. Representação das Crenças

No sistema proposto, as crenças dos agentes representam a interpretação das leituras provenientes dos sensores, utilizando a lógica do framework Jason. Essas crenças são atualizadas de forma dinâmica a partir das percepções recebidas, permitindo que o estado da planta seja representado de maneira simbólica, como solo(seco) ou clima(confortavel), em vez de trabalhar diretamente com os valores numéricos lidos pelos sensores.

As condições do solo são classificadas a partir da variável de umidade, conforme as seguintes regras:

```

2 solo(seco) :- umidadeSolo(U) & U < 20.
  solo(ideal) :- umidadeSolo(U) & U >= 20 & U <= 60.
  solo(encharcado) :- umidadeSolo(U) & U > 60.

```

Listing 2. Regras de inferência para classificação do solo

A exposição à luz solar é tratada de forma acumulativa, permitindo avaliar a luminosidade recebida pela planta ao longo do tempo. Dessa forma, o sistema considera não apenas leituras instantâneas, mas também o histórico de exposição solar para determinar se a planta recebe luminosidade suficiente.

```

1 historico_luz(suficiente) :-
  minutos_sol_acumulados(M) &
  horas_sol_necessarias(H) &
  M >= (H * 60).

```

Listing 3. Regra de inferência para avaliação da luminosidade

A temperatura ambiente é interpretada de forma simbólica pelo sistema, permitindo classificar as condições climáticas em diferentes níveis de conforto para a planta, sendo utilizada como apoio ao processo de inferência do agente.

```

2 clima(frio) :- temperatura(T) & T < 15.
  clima(confortavel) :- temperatura(T) & T >= 15 & T <= 30.
  clima(quente) :- temperatura(T) & T > 30.

```

Listing 4. Regra de inferência para classificação da temperatura

3.3. Inferência e Geração de Alertas

As crenças atualizadas a partir das leituras dos sensores servem de base para o processo de inferência realizado pelo Agente Planta. A cada novo ciclo de monitoramento, o agente avalia as condições do ambiente e verifica se existe alguma situação que demande a intervenção do usuário.

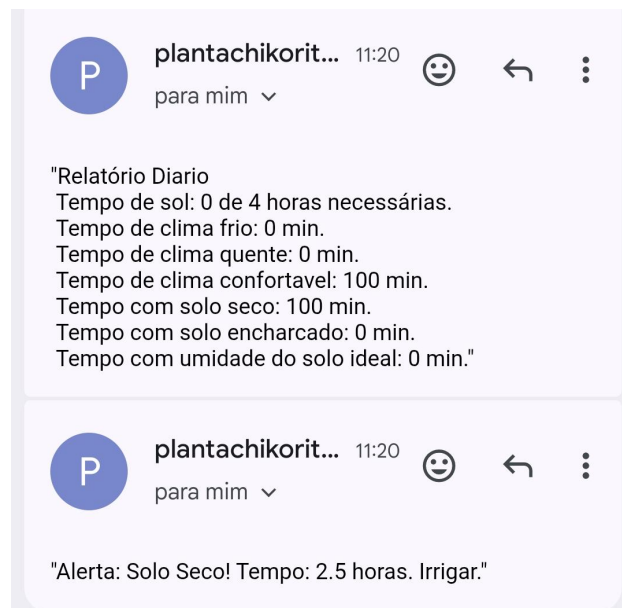


Figura 3. Exemplo de Relatório Diário.

Para evitar alertas provocados por variações momentâneas dos sensores, o agente mantém um histórico das condições observadas ao longo do tempo. Esse histórico permite identificar situações persistentes, como solo seco ou encharcado por períodos prolongados, evitando alertas inconsistentes.

Quando uma dessas condições ultrapassa os limites estabelecidos, o Agente Planta gera um alerta e o encaminha ao Agente Mailer por meio da troca de mensagens do framework Jason. O Agente Mailer foi implementado utilizando a biblioteca Mailer para Jason, que disponibiliza ações internas para configuração do serviço de correio eletrônico e envio de mensagens, permitindo que agentes BDI interajam com usuários por meio do protocolo de e-mail [Lazarin et al. 2024].

A Figura 3 apresenta um exemplo de relatório diário gerado pelo sistema e enviado ao usuário. Além de informar as condições atuais da planta, o relatório reúne informações acumuladas durante o período de monitoramento, fornecendo uma visão geral do ambiente ao longo do dia.

4. Trabalhos Correlacionados

O monitoramento inteligente de plantas tem sido amplamente explorado por meio de soluções baseadas em Internet das Coisas (IoT), principalmente voltadas ao monitoramento ambiental e à automação da irrigação.

Elangovan et al. [Elangovan et al. 2018] apresenta um sistema de monitoramento utilizando Raspberry Pi e sensores para acompanhar variáveis como temperatura, umidade do solo, luminosidade e pH. Os dados coletados são disponibilizados ao usuário para acompanhamento remoto, permitindo a visualização das condições da planta em tempo real. Entretanto, o sistema concentra-se na aquisição e apresentação das informações, sem incorporar mecanismos de raciocínio para interpretar automaticamente os estados monitorados. Essa abordagem faz com que o usuário precise acompanhar as informações disponibilizadas e interpretá-las para identificar quando a planta necessita de algum cui-

dado, o que pode dificultar o acompanhamento contínuo, especialmente em situações nas quais alterações relevantes ocorram entre os momentos de consulta.

Karpagam et al. [Karpagam et al. 2020] propõem um sistema de irrigação automática baseado em Arduino, utilizando sensores de umidade do solo, temperatura e umidade do ar. A irrigação é acionada conforme limites previamente definidos, e notificações são enviadas ao usuário por meio de um módulo GSM. A tomada de decisão, entretanto, permanece centralizada na lógica implementada no microcontrolador. Embora essa abordagem possibilite a automação do cuidado da planta, a ação é realizada diretamente pelo sistema a partir das condições previamente estabelecidas. No contexto doméstico, uma abordagem em que o sistema interpreta as condições da planta e comunica ao proprietário a necessidade de atenção permite manter o usuário como responsável pela decisão sobre a ação a ser realizada, preservando sua autonomia no cuidado da planta.

Em contraste, o sistema proposto neste trabalho adota uma arquitetura baseada em Sistemas Multiagentes desenvolvida no framework Jason. Nessa abordagem, as informações provenientes dos sensores são convertidas em crenças, permitindo que os agentes realizem inferências simbólicas antes da tomada de decisão. Além disso, a separação entre o Agente Planta, responsável pelo monitoramento e raciocínio, e o Agente Mailer, responsável pelo envio de notificações, proporciona uma arquitetura mais modular e facilita a manutenção e a evolução do sistema. Essa organização permite a inclusão de novos agentes e funcionalidades sem alterações significativas nos componentes já existentes.

5. Reprodutibilidade

Com o objetivo de promover a reprodutibilidade dos experimentos apresentados neste trabalho, todo o código-fonte desenvolvido foi disponibilizado em um repositório público no GitHub¹. O repositório contém a implementação dos agentes desenvolvidos em Jason, o código embarcado para o Arduino, a documentação do projeto e instruções para reprodução do protótipo.

6. Conclusão

Este trabalho apresentou o desenvolvimento de uma arquitetura baseada em Sistemas Multiagentes para o monitoramento inteligente de plantas em ambiente doméstico. A solução proposta integra sensores ambientais, um microcontrolador Arduino e um Raspberry Pi, utilizando agentes implementados no framework Jason para interpretar continuamente as condições do ambiente e comunicar ao usuário as necessidades da planta por meio de linguagem natural.

A implementação demonstrou a aplicação de Sistemas multiagentes em um cenário de Internet das Coisas, no qual agentes cooperativos atuam de forma distribuída na aquisição de dados, inferência das condições ambientais e comunicação com o usuário. A utilização de crenças, regras de inferência e troca de mensagens entre agentes permitiu transformar leituras dos sensores em informações compreensíveis, favorecendo uma interação mais ativa do usuário com o processo de cuidado da planta.

¹<https://github.com/LabRedesCefetRJ/carvalho2026>

Como trabalhos futuros, pretende-se ampliar a arquitetura para permitir o monitoramento simultâneo de diferentes plantas, cada uma representada por um agente inteligente com suas próprias características e histórico de cuidados. Essa abordagem possibilita que agentes de diferentes usuários interajam entre si, compartilhando informações e experiências relacionadas ao cultivo. Dessa forma, busca-se criar uma experiência mais colaborativa, na qual as plantas possam compartilhar seu desenvolvimento por meio dos agentes, incentivando a troca de conhecimentos e fortalecendo o envolvimento dos usuários com o cuidado das plantas.

Referências

- Elangovan, R., Santhanakrishnan, N., Rozario, R., and Banu, A. (2018). Tomen: A plant monitoring and smart gardening system using iot. *International Journal of Pure and Applied Mathematics*, 118(20):539–544.
- Joice, A., Tufaique, T., Tazeen, H., Igathinathane, C., Zhang, Z., Whippo, C., Hendrickson, J., and Archer, D. (2025). Applications of Raspberry Pi for Precision Agriculture—A Systematic Review. *Agriculture*, 15(3). DOI: 10.3390/agriculture15030227.
- Karpagam, J., Merlin, I., Bavithra, P., and Kousalya, J. (2020). Smart Irrigation System Using IoT. In *6th ICACCS*. DOI: 10.1109/ICACCS48705.2020.9074201.
- Lazarin, N. and Pantoja, C. (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*, pages 13–20, Porto Alegre, RS, Brasil. SBC. DOI: 10.5753/wesaac.2015.33308.
- Lazarin, N., Pantoja, C., and Viterbo, J. (2026). An operating-system infrastructure for embedded bdi-based multi-agent systems. In *Computational Science and Its Applications – ICCSA 2026*, pages 589–609, Cham. Springer Nature Switzerland. DOI: 10.1007/978-3-032-30494-0_37.
- Lazarin, N. M., de Souza, J. P. B., Pantoja, C. E., Alexandre, T., and Viterbo, J. (2024). Human and BDI-Agent Interaction via KQML Messages over IMAP and SMTP. In *Proceedings of PAAMS 2024*, page 172–183, Berlin, Heidelberg. Springer-Verlag. DOI: 10.1007/978-3-031-70415-4_15.
- Norman, D. (2013). *The design of everyday things: Revised and expanded edition*. Basic books.
- 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*, volume 14195, pages 382–396, Cham. Springer Nature Switzerland. DOI: 10.1007/978-3-031-45368-7_25.
- 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 *Advances in Practical Applications of Agents, Multi-Agent Systems, and Cognitive Mimetics. The PAAMS Collection*, volume 13955, pages 346–358. Springer Nature Switzerland, Cham. DOI: 10.1007/978-3-031-37616-0_29.
- Wooldridge, M. (2009). *An introduction to multiagent systems*. John Wiley & Sons.