

Simulador de aplicações de Inteligência Artificial das Coisas para monitoramento em tempo real

Allan Jheyson R. Gonçalves¹, Ana Beatriz N. de Castro¹, Brenda Evilly¹,
Erico Meneses Leão¹, Jose R. Torres Neto¹, Romuere R. V. Silva¹,
Antonio Oseas C. Filho¹, Ricardo A. L. Rabelo¹,

¹Universidade Federal do Piauí (UFPI)

jheyson@ufpi.edu.br, {anabeatriznovais31,brendaevilly014}@gmail.com,
{ericoleao,jtorres,romuere,antoniooseas,ricardoalr}@ufpi.edu.br

Abstract. *The advancement of the Internet of Things (IoT) and Artificial Intelligence (AI) technologies has opened up new possibilities for applications in several areas, including real-time monitoring. This work presents the development of an Artificial Intelligence of Things (AIoT) applications simulator to monitor rural areas using Unmanned Aerial Vehicles (UAVs). The proposal integrates an edge/fog/cloud architecture, where UAVs equipped with cameras and AI algorithms perform real-time animal detection. The system distributes the processing load between the edge devices and the fog server, optimizing the efficiency and accuracy of detections. The developed graphical interface allows visualization and management of simulations, facilitating analysis and decision-making. The results demonstrate the viability and effectiveness of the system for monitoring difficult-to-access environments, contributing to efficient resource management and rapid response to application events.*

Resumo. *O avanço das tecnologias de Internet das Coisas (Intelligence of Things – IoT) e Inteligência Artificial (IA) abriu novas possibilidades de aplicações em diversas áreas, incluindo monitoramento em tempo real. Este trabalho apresenta o desenvolvimento de um simulador de aplicações de Inteligência Artificial das Coisas (Artificial Intelligence of Things – AIoT) para monitoramento de áreas rurais utilizando Veículos Aéreos Não Tripulados (VANTs). A proposta integra uma arquitetura edge/fog/cloud, onde VANTs equipados com câmeras e algoritmos de IA realizam a detecção de animais em tempo real. O sistema distribui a carga de processamento entre os dispositivos de borda e o servidor fog, otimizando a eficiência e a precisão das detecções. A interface gráfica desenvolvida permite a visualização e gerenciamento de simulações, facilitando a análise e a tomada de decisões. Os resultados demonstram a viabilidade e eficácia do sistema para monitoramento de ambientes de difícil acesso, contribuindo para uma gestão eficiente de recursos e resposta rápida a eventos da aplicação.*

1. Introdução

A tecnologia envolvendo redes sem fio tem se tornado comum para a sociedade na indústria e comércio, agregando valor às atividades realizadas. De acordo com [Stewart

et al. 2020], uma dessas tecnologias foi a Internet das Coisas (*Internet of Things – IoT*), que amplia os sentidos humanos por meio de dispositivos. Com a popularização da Inteligência Artificial (IA), muitas possibilidades surgiram, dando origem à Inteligência Artificial das Coisas (*Artificial Intelligence of Things – AIoT*). Aplicações de AIoT possibilitam a descentralização na tomada de decisões em tempo real para todas as camadas da aplicação, tornando a solução mais precisa e eficiente.

Uma das bases da IoT é a arquitetura de rede *edge/fog/cloud*. De acordo com Minoli et al. 2017, essa arquitetura adota uma abordagem orientada a nós em que o modelo é dividido em três camadas que compreendem diferentes tipos de recursos e é escalável para um grande número de dispositivos *edge* e *fog* interconectados e permite a utilização eficiente do poder de processamento que eles oferecem. Nesse sentido, a utilização desses recursos permite aumentar a eficiência das aplicações da IoT.

As aplicações de IoT foram inicialmente desenvolvidas para ambientes urbanos, onde a conectividade e a energia são mais acessíveis. Em áreas remotas, a IoT enfrenta desafios significativos, como a falta de conectividade e recursos limitados, especialmente em dispositivos como VANTs [Zhu et al. 2022]. A implantação de sensores em cada sujeito da rede pode ser onerosa, mas a utilização de VANTs em AIoT pode melhorar a mobilidade e a comunicação em áreas de difícil acesso [Geraci et al. 2022]. A gestão dessas redes exige estratégias para garantir a cooperação eficaz entre nós terrestres e aéreos, e a integração de processamento na camada *edge* pode reduzir a latência e aumentar a eficiência das aplicações [Wang et al. 2022].

Realizar experimentos de IoT em ambientes reais é custoso e demorado, tornando os simuladores uma alternativa eficaz para avaliar novas soluções. Contudo, muitos simuladores exigem alta capacidade computacional e configuração complexa, além de falharem na integração da IA com IoT. Este trabalho propõe um simulador web que integra mobilidade, IA, e a arquitetura *edge/fog/cloud*, visando a detecção de objetos em tempo real em ambientes remotos.

Este trabalho está organizado da seguinte maneira: a Seção 2 discute os trabalhos relacionados, fornecendo uma revisão da literatura e situando este estudo no contexto da pesquisa atual. A Seção 3 descreve o desenvolvimento do simulador de aplicações de AIoT, detalhando a arquitetura do software, as tecnologias utilizadas e os algoritmos de detecção de objetos. A Seção 4 apresenta os resultados e discussões, analisando a efetividade do sistema desenvolvido. Finalmente, a Seção 5 conclui o trabalho, destacando as principais contribuições e sugerindo direções para futuras pesquisas e aprimoramentos.

2. Trabalhos Relacionados

A literatura sobre simuladores de redes IoT integrados com VANTs revela várias abordagens que exploram diferentes aspectos da simulação e integração de tecnologias emergentes. Por exemplo, o FlyNetSim [Baidya et al. 2018] é um simulador de código aberto que oferece uma plataforma flexível para a modelagem de operações de VANTs e a dinâmica de comunicação em redes IoT. O simulador permite a análise e avaliação de redes de VANTs em diferentes cenários, embora enfrente desafios como a complexidade da integração e atrasos na comunicação em caso de grande fluxo de dados.

Outra proposta relevante é a extensão de um ambiente de simulação IoT que suporta nós móveis, conforme apresentado por Barriga et al. 2023. Este trabalho foca na

criação de um ambiente de simulação IoT que acomoda todas as camadas da arquitetura *edge/fog/cloud*. Apesar da eficácia demonstrada em dois estudos de caso, a limitação notável reside nos protocolos e na conectividade disponíveis, o que pode restringir a aplicabilidade em cenários reais.

Uma análise comparativa de simuladores para aplicações IoT em computação em nuvem e névoa foi realizada por Bajaj et al. 2022, com o objetivo de identificar os simuladores mais adequados para as demandas atuais da IoT. A análise conclui que os simuladores de nuvem e névoa são mais adaptados às demandas modernas, especialmente pela capacidade de evolução e adaptação desses cenários, embora também existam limitações em termos de escalabilidade e resposta em tempo real.

Pereira et al. 2021 apresentam uma ferramenta de simulação voltada para ambientes IoT-Fog-Cloud que permite a análise de políticas de gerenciamento de recursos em ambientes IoT-Fog, sendo eficaz para a simulação de diferentes configurações de rede. No entanto, a complexidade na modelagem de cenários dinâmicos e heterogêneos foi identificada como uma limitação importante. Já os trabalhos de Kertész et al. 2019 e Del-Pozo-Puñal et al. 2023 se destacam ao apresentar simuladores que integram computação em névoa e nuvem com suporte para mobilidade em aplicações sensíveis à latência. As soluções são eficazes na gestão de dispositivos IoT móveis, mas enfrentam desafios na simulação de interações em tempo real entre dispositivos e recursos distribuídos.

Os trabalhos mencionados evidenciam a necessidade de aprimoramentos nos simuladores existentes, principalmente em termos de suporte adequado para a mobilidade dos dispositivos IoT e a migração de serviços, crucial para aplicações em ambientes dinâmicos e distribuídos. Restrições na escalabilidade e na capacidade de resposta impactam a eficiência das simulações em larga escala e impossibilitam a integração de IA nos experimentos. O presente artigo apresenta uma solução de software web que integra AIoT sobre a arquitetura *edge/fog/cloud* no cenário de ambientes remotos para detecção de objetos em tempo real.

3. Simulador de aplicações de AIoT para monitoramento em tempo real

Para o desenvolvimento do simulador, foram estabelecidas algumas etapas de construção, baseadas em princípios da Engenharia de Software para manutenção da qualidade e documentação do projeto. O software implementado é um sistema simulador de IoT executado em navegador *web* capaz de realizar o reconhecimento de objetos em campo por meio de visão computacional. A proposta é possibilitar a visualização e experimentação de uma arquitetura *edge/fog/cloud* que utiliza algoritmos de IA embarcados em VANTs na detecção de objetos por meio de imagens.

Na camada *edge*, os VANTs são equipados com um algoritmo de detecção leve e simplificado, de baixo custo computacional e energético; na camada *fog*, o servidor executa um algoritmo de detecção mais robusto, que busca máxima acurácia. Todas as ocorrências de simulações são registradas na base de dados em nuvem.

A arquitetura, tecnologias e metodologias utilizadas foram baseadas primordialmente no cenário de aplicação. Tratando-se de uma proposta voltada para ambientes rurais, a ideia de construção surgiu sobre o cenário de fazendas para a detecção de animais, onde muitas vezes é necessário realizar a procura de animais de produção desgarrados,

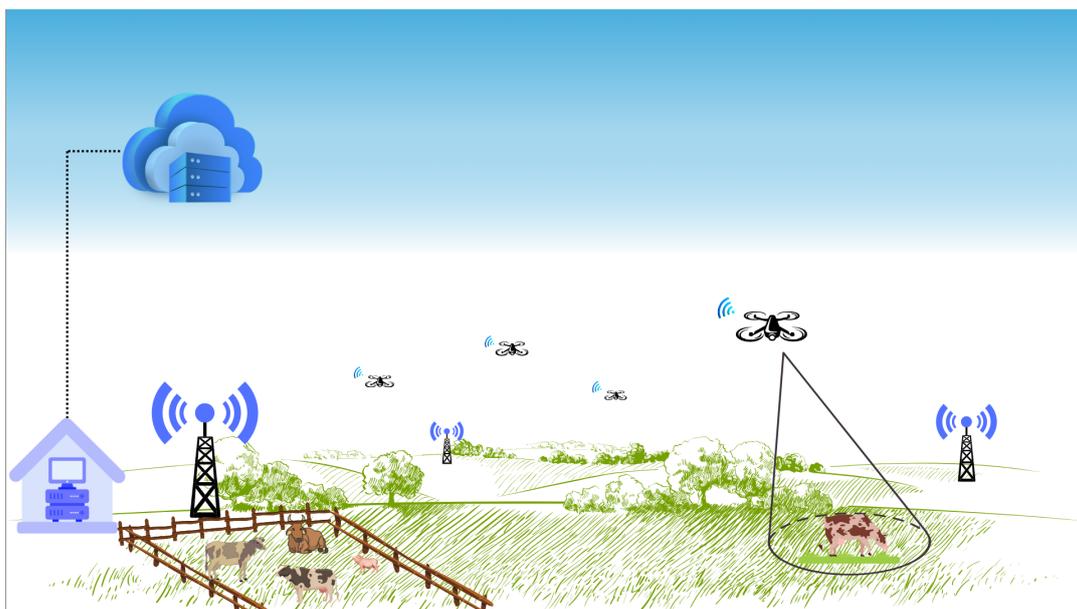


Figura 1. Representação do cenário de aplicação

perdidos, feridos ou ainda de animais indesejados que possam oferecer algum risco a outros animais ou aos humanos. A Figura 1 representa o ambiente, onde é possível visualizar a infraestrutura da proposta; VANTs sobrevoando a região e detectando animais; uma casa abrigoando o servidor *fog* para provisão dos serviços locais e a nuvem, responsável por armazenar todos os dados coletados.

3.1. Definição da Arquitetura do Software

O conjunto de tecnologias para o software foi definido visando atender a necessidade de simulação de VANTs equipados com câmeras sobrevoando vastas regiões com algoritmos de IA embarcados, capazes de detectar animais por meio de visão computacional. A Figura 2 apresenta as tecnologias empregadas no *frontend*, *backend*, protocolos de comunicação e linguagens de programação.

O Angular Framework foi utilizado para criação e integração de todos os componentes de interface gráfica, como os VANTs, animais, mapa, configurações e controles das simulações [Google 2024]. Em complemento, foram utilizados HTML, SCSS e JavaScript [Network 2024] para layout e aparência, além de algumas funcionalidades complementares ao Angular. A linguagem de programação TypeScript foi utilizada tanto no *frontend* quanto no *backend*, nos *frameworks* Angular e NestJS.

No *backend*, o NestJS foi a base de toda a infraestrutura do simulador, integrando os dados das simulações realizadas pelo usuário com a base de dados na nuvem [Myśliwiec 2024]. O Prisma ORM [Prisma 2024] foi integrado para otimizar o desenvolvimento e manutenção da infraestrutura de banco de dados além da criação das funcionalidades *Create*, *Update*, *Read*, *Delete* (CRUD). Em nuvem, o PostgreSQL [Group 2024] armazena e disponibiliza todos os dados das simulações para consultas e análises posteriores.

Para a documentação da *Application Programming Interface* (API), foi utilizada a ferramenta Swagger, facilitando o entendimento e uso da API pelo desenvolvedor e

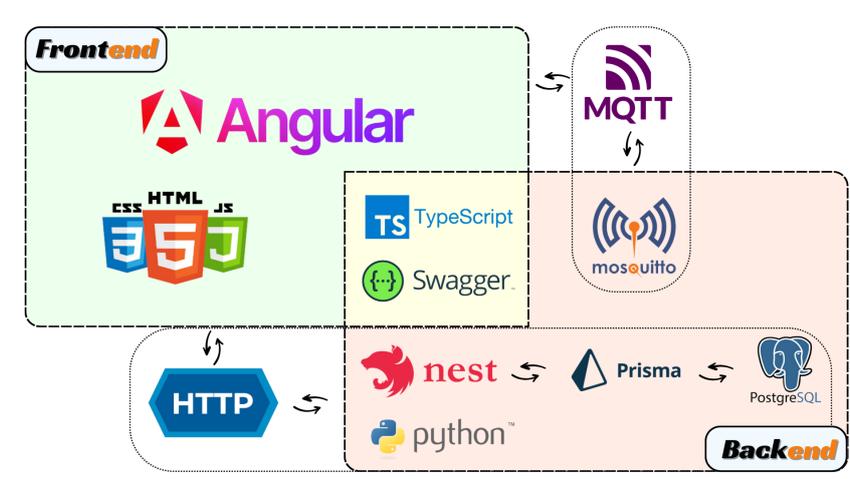


Figura 2. Arquitetura do sistema

aplicativos [SmartBear 2024]. O *Hypertext Transfer Protocol* (HTTP) [Fielding et al. 1999] foi empregado para a comunicação entre VANTs e o servidor *fog*. O *Message Queuing Telemetry Transport* (MQTT) gerencia a comunicação entre VANTs e o envio de mensagens do administrador, permitindo que um VANT notifique os demais sobre eventos, evitando duplicação de esforços [OASIS 2024]. O broker MQTT Mosquitto [Eclipse 2024] suporta a comunicação entre os VANTs e a disseminação de mensagens de controle enviadas pelo administrador.

O servidor *fog* possui uma infraestrutura *backend* capaz de processar imagens para a detecção de objetos e manter o Mosquitto funcionando para a troca de mensagens entre os VANTs. Também hospeda a aplicação de gerenciamento dos VANTs, permitindo o envio de comandos específicos, como o cancelamento de buscas. A linguagem Python foi utilizada para o processamento de imagens no *backend*, devido à sua simplicidade e legibilidade, sendo amplamente utilizada em desenvolvimento web, análise de dados e inteligência artificial.

3.2. Algoritmos de Detecção de Objetos em Imagens

O simulador utiliza o método *You Only Look Once* (YOLO), embarcado nos VANTs e a arquitetura *Residual Networks* (ResNet), para uma detecção mais aprimorada no servidor *fog*. A camada *edge* utiliza a YOLOv9 [Wang and Liao 2024], onde os VANTs realizam a detecção em tempo real e a camada *fog* utiliza ResNet-50 [Carion et al. 2020], realizando um processamento mais robusto para maiores acurácias, ambos pré-treinados com o conjunto de dados *Common Objects in Context* (COCO) 2017 [Lin et al. 2015].

A escolha do YOLOv9 para detecção de objetos se justifica por sua superioridade em velocidade e precisão em comparação com outras abordagens. O YOLOv9 utiliza uma arquitetura de rede neural convolucional que permite a detecção de objetos em uma única passagem pela rede, tornando-o extremamente rápido e eficiente para aplicações em tempo real, como vigilância por vídeo e veículos autônomos [Wan et al. 2022].

ResNet-50 foi escolhido devido à sua eficácia comprovada em diversas tarefas de reconhecimento de imagem, beneficiando-se da introdução de conexões residuais que permitem o treinamento de redes mais profundas sem o problema de degradação. Esta

ID	Nome	Longo Alcance	Bateria	Status	Ações
1	Drone 1	true	100%	OCIOSO_BASE	
2	Drone 2	true	100%	OCIOSO_BASE	
3	Drone 3	true	100%	OCIOSO_BASE	
7	Drone 4	false	100%	OCIOSO_BASE	
8	Drone 5	false	100%	OCIOSO_BASE	
9	Drone 6	false	100%	OCIOSO_BASE	

Figura 3. Tela de Lista de dispositivos cadastrados

arquitetura é particularmente eficaz para tarefas complexas de reconhecimento de imagem, como reconhecimento facial e classificação de objetos, onde a profundidade da rede permite uma melhor extração de características e uma maior precisão [Obaid et al. 2023].

Com a YOLOv9 implementada nos VANTs, a detecção de objetos em tempo real é precisa e permite respostas imediatas aos eventos. A integração com a ResNet-50 no servidor *fog* adiciona uma camada de processamento mais detalhada, garantindo análise precisa das imagens. Essa combinação aumenta a capacidade de detecção e reconhecimento de objetos, otimizando o uso de recursos computacionais distribuídos entre as camadas *edge* e *fog*, resultando em uma solução eficaz no monitoramento em tempo real.

4. Resultados e Discussões

A interface gráfica do simulador desenvolvido possibilita a visualização dos animais no campo, da movimentação dos VANTs e do servidor *fog*, todos representados por elementos gráficos sobre um mapa interativo. Também contempla um menu lateral para navegação e gerência dos dados e um Painel de Controle para acompanhamento das simulações em tempo real. Há ainda uma área de *chat*, onde podem ser vistas as mensagens registradas pelo Mosquitto entre usuários e VANTs ativos. A Figura 3 apresenta a tela de dispositivos cadastrados e seus respectivos dados. Além disso, é possível:

- **Gerenciar dispositivos:** Cadastrar, modificar ou excluir VANTs para as simulações. Visualizar o status de conexão, nível de bateria e ações em execução em tempo real. Enviar comandos para controle de ações em campo durante uma busca.
- **Gerenciar objetos:** Cadastrar, modificar ou excluir animais ou os objetos de interesse que devem ser buscados no ambiente da aplicação.
- **Visualizar registros:** Acompanhar o registro de buscas, tópicos MQTT e eventos, todos estes gerados e alimentados automaticamente pelo simulador após o início de uma busca por objetos.

O mapa interativo é o local onde as simulações podem ser acompanhadas e gerenciadas. Após selecionar o tipo de objeto (animal) a ser adicionado à simulação, o usuário pode selecionar o local do mapa onde deseja inserir e aparecerá um círculo pequeno e colorido no local; essa inserção irá selecionar aleatoriamente uma imagem de vista aérea de um objeto do tipo selecionado, assim, a aproximação do VANT a uma certa distância possibilita que ele detecte e analise esta imagem. Um clique no círculo mostra

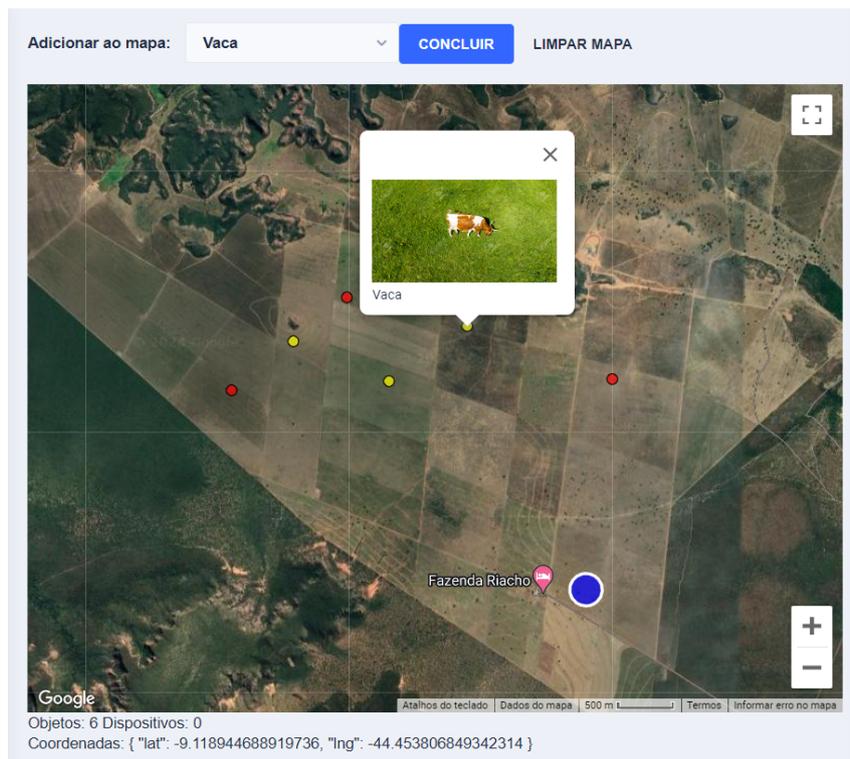


Figura 4. Animais inseridos no mapa pelo usuário

a imagem vinculada ao animal, conforme pode ser visto na Figura 4, onde os círculos amarelos são vacas e os vermelhos são cavalos. Este processo pode ser repetido quantas vezes for necessário, para todos os animais cadastrados na base de dados. O servidor *fog* é representado pelo círculo maior na cor azul, local de onde os VANTs iniciam as buscas.

Ao concluir a inserção dos animais da simulação, novas opções são exibidas para configuração dos animais a serem procurados e, dentre os dispositivos (VANTs) cadastrados, quais deles participarão da busca. Alguns VANTs podem ser configurados como de "longo alcance", desta maneira, estes passam a ter 50% mais autonomia de voo do que um VANT simples e recarga de bateria 2x mais rápido, assim, podem alcançar regiões mais distantes da base, local onde está sempre conectado à fonte de recarga.

De acordo com a extensão da área do mapa, o tempo de simulação pode tornar-se demasiadamente longo, pois o simulador busca retratar uma velocidade de locomoção dos VANTs de aproximadamente 20km/h. Por esta razão, foi adicionada uma opção de seleção da velocidade da simulação, onde os VANTs podem ampliar sua velocidade em até 16x. É importante ressaltar que a esta alteração na velocidade não afeta os demais parâmetros da simulação além da velocidade de locomoção e o consequente consumo energético dos VANTs.

Uma simulação em andamento é apresentada na Figura 5. Nela, há 3 vacas no perímetro da fazenda e 3 VANTs foram selecionados para a busca (1,3 e 4). Os VANTs de longo alcance (1 e 3) são representados por círculos brancos com bordas azuis, os demais (Drone 4) são totalmente brancos e um pouco maiores. No painel de controle pode-se notar que os VANTs em atividade possuem um símbolo verde logo abaixo da sua sigla, além de outras informações.

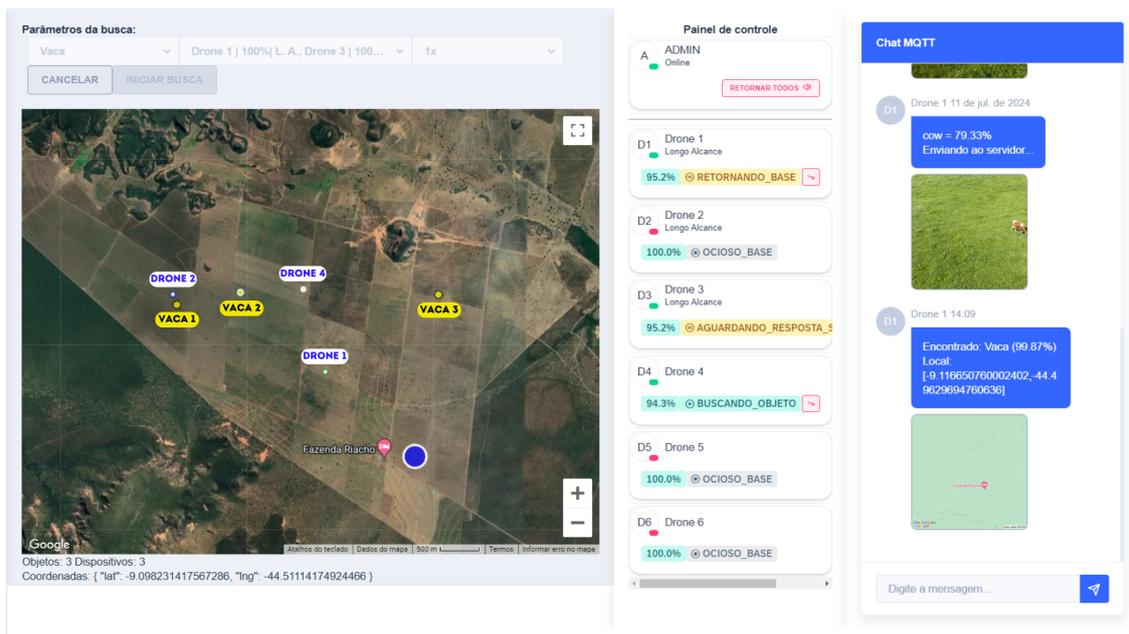


Figura 5. Simulação em andamento

O Drone 1, por exemplo, está online, possui 95,2% de bateria disponível e seu status é "retornando para a base". Observando no mapa, o Drone 1 está com a borda verde, o que significa que ele localizou um objeto e está retornando. No Chat MQTT pode-se perceber que o Drone 1 localizou uma vaca com 79,33% de acurácia, o que é insuficiente para o limiar de 90% estabelecido nos experimentos. A imagem capturada é enviada ao servidor *fog*, este a processa e retorna a resposta ao Drone 1, que confirma a detecção da vaca com 99,87% de acurácia e realiza o procedimento de retorno para a base. Todas as atualizações de status são registrados na base de dados em nuvem. Para os objetos, cada tipo recebe uma cor aleatória na inserção, uma borda branca nestes representa que o objeto foi localizado por um VANT (vaca 2). O esquema de cores dos VANTs no mapa é:

- **Branco:** simples.
- **Branco com borda azul:** longo alcance.
- **Branco com borda verde:** retorno à base por sucesso na busca.
- **Branco com borda amarela:** retorno à base por solicitação do usuário.
- **Branco com borda vermelha:** retorno à base por bateria baixa.

O painel de controle disponibiliza botões para solicitar o retorno dos VANTs para a base, cancelando a busca em andamento. Outros comandos podem ser enviados diretamente pelo Chat MQTT, onde o broker realiza a disseminação da mensagem para os VANTs destinatários. A interpretação das mensagens é feita com base em strings padronizadas, que possuem nos 3 primeiros caracteres o ID do VANT que deve interpretar a mensagem. Também é possível enviar mensagens para um grupo de VANTs por tipo ou status atual. As mensagens deste tipo geralmente servem para alteração nos parâmetros de uma busca em andamento, como adicionar ou modificar os objetivos da busca.

5. Conclusões

Neste trabalho foi desenvolvido um simulador de aplicações de AIoT que integra VANTs em redes IoT aeroterrestres para monitoramento em tempo real. A principal contribuição

é a possibilidade de utilização de algoritmos avançados de IA, como YOLOv9 e ResNet-50, distribuídos em uma arquitetura *edge/fog/cloud*. Essa abordagem permite a detecção precisa e rápida de objetos, com o processamento inicial realizado pelos VANTs e uma análise mais robusta executada no servidor *fog*. Além disso, foi construída uma interface gráfica interativa web que facilita a visualização e o gerenciamento das simulações, tornando a ferramenta eficaz para a análise e tomada de decisões em tempo real com auxílio da IA sem a necessidade de configurações adicionais no ambiente do usuário. A eficiência do sistema foi demonstrada em um cenário de monitoramento de áreas rurais, onde a conectividade é limitada e a necessidade de gestão eficiente de recursos é crítica.

Para aprimorar e expandir este trabalho, além de contribuir para os avanços na experimentação científica na área da IoT, podem ser realizados alguns aprimoramentos, como a integração de algoritmos de detecção variados, tanto no *frontend* quanto no *backend*. Isso permitirá ao usuário escolher e visualizar simulações com diferentes modelos de IA, coletando dados de desempenho comparativos. Além disso, a arquitetura poderá ser estendida para suportar múltiplos servidores *fog*, melhorando a distribuição de carga e a resiliência do sistema. Também poderão ser implementados modelos de movimentação para os VANTs, simulando comportamentos de busca e patrulha mais realistas.

O desenvolvimento deste simulador representa um passo significativo na aplicação de tecnologias AIoT para o monitoramento em tempo real. A combinação de IA e IoT em uma arquitetura *edge/fog/cloud* mostrou-se promissora, oferecendo uma solução eficaz e escalável para desafios de monitoramento em áreas rurais e de difícil acesso. Os resultados obtidos incentivam a continuidade e expansão de softwares nesta linha de pesquisa, com o objetivo de aprimorar cada vez mais as capacidades e aplicações das redes AIoT.

Referências

- Baidya, S., Shaikh, Z., and Levorato, M. (2018). Flynetsim: An open source synchronized uav network simulator based on ns-3 and ardupilot. In *Proceedings of the 21st ACM International Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems*, pages 37–45.
- Bajaj, K., Sharma, B., and Singh, R. (2022). Comparative analysis of simulators for iot applications in fog/cloud computing. In *2022 International Conference on Sustainable Computing and Data Communication Systems (ICSCDS)*, pages 983–988. IEEE.
- Barriga, J. A., Clemente, P. J., Pérez-Toledano, M. A., Jurado-Málaga, E., and Hernández, J. (2023). Design, code generation and simulation of iot environments with mobility devices by using model-driven development: Simulateiot-mobile. *Pervasive and Mobile Computing*, 89:101751.
- Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A., and Zagoruyko, S. (2020). End-to-end object detection with transformers. *CoRR*, abs/2005.12872.
- Del-Pozo-Puñal, E., García-Carballeira, F., and Camarmas-Alonso, D. (2023). A scalable simulator for cloud, fog and edge computing platforms with mobility support. *Future Generation Computer Systems*, 144:117–130.
- Eclipse, F. (2024). *Mosquitto Documentation*. Accessed: 2024-06-24.

- Fielding, R., Gettys, J., Mogul, J., Frystyk, H., Masinter, L., Leach, P., and Berners-Lee, T. (1999). Hypertext transfer protocol – http/1.1. Technical report, IETF. Accessed: 2024-07-10.
- Geraci, G., López-Pérez, D., Benzaghta, M., and Chatzinotas, S. (2022). Integrating terrestrial and non-terrestrial networks: 3d opportunities and challenges. *ArXiv*, abs/2207.10385.
- Google (2024). *Angular Documentation*. Accessed: 2024-07-10.
- Group, P. G. D. (2024). *PostgreSQL Documentation*. Accessed: 2024-07-10.
- Kertész, A., Pflanzner, T., and Gyimóthy, T. (2019). A mobile iot device simulator for iot-fog-cloud systems. *Journal of Grid Computing*, 17:529–551.
- Lin, T.-Y., Maire, M., Belongie, S., Bourdev, L., Girshick, R., Hays, J., Perona, P., Ramanan, D., Zitnick, C. L., and Dollár, P. (2015). Microsoft coco: Common objects in context.
- Minoli, D., Sohraby, K., and Occhiogrosso, B. (2017). Iot considerations, requirements, and architectures for smart buildings—energy optimization and next-generation building management systems. *IEEE Internet of Things Journal*, 4(1):269–283.
- Myśliwiec, K. (2024). *NestJS Documentation*. Accessed: 2024-07-10.
- Network, M. D. (2024). *Web Documentation*. Accessed: 2024-07-10.
- OASIS (2024). *MQTT Protocol*. Accessed: 2024-07-10.
- Obaid, A., Shawkat, A. S., and Abdulhussein, N. S. (2023). Exploring the potential of a-resnet in person-independent face recognition and classification. *International Journal of Advanced Network, Monitoring and Controls*, 8:12–19.
- Pereira, R. S., Prazeres, C. V. S., Barbosa, M. T. M., Barros, E. B. C., and Peixoto, M. L. M. (2021). Iotfogsim: Um simulador orientado a eventos para avaliação de aplicações baseadas em iot-fog-cloud. In *Anais Estendidos do XXXIX Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos*, pages 25–32. SBC.
- Prisma, I. (2024). *Prisma Documentation*. Accessed: 2024-07-10.
- SmartBear (2024). *Swagger Documentation*. Accessed: 2024-07-01.
- Stewart, J. C., Davis, G. A., and Igoche, D. A. (2020). Ai, iot, and aiot: Definitions and impacts on the artificial intelligence curriculum. *Issues in Information Systems*, 21(4).
- Wan, C., Pang, Y., and Lan, S. (2022). Overview of yolo object detection algorithm. *International Journal of Computing and Information Technology*.
- Wang, C.-Y. and Liao, H.-Y. M. (2024). YOLOv9: Learning what you want to learn using programmable gradient information.
- Wang, H., Li, K., and Xu, C. (2022). A new generation of resnet model based on artificial intelligence and few data driven and its construction in image recognition model. *Computational Intelligence and Neuroscience*, 2022.
- Zhu, S., Ota, K., and Dong, M. (2022). Green AI for IIoT: Energy efficient intelligent edge computing for industrial internet of things. *IEEE Transactions on Green Communications and Networking*, 6:79–88.