

Sincronização em nuvem para a plataforma e-Cattle utilizando *multi-tenant*

Ygo A. Brito¹, Thiago O. Soares¹, Hana K. S. Rubinsztein¹, Camilo Carromeu²

¹Faculdade de Computação – Universidade Federal de Mato Grosso do Sul (UFMS)
79.070-900 – Campo Grande – MS – Brasil

²Empresa Brasileira de Pesquisa Agropecuária (EMBRAPA)
79.106-550 – Campo Grande – MS – Brasil

{ygo1992, thg.ol.so}@gmail.com, hana.rubinsztein@ufms.br,
camilo.carromeu@embrapa.br

Abstract. *Cattle farming is crucial to the Brazilian economy, significantly contributing to the country's GDP. Technological innovations are essential for increasing efficiency and sustainability. The e-Cattle platform emerges as an integrated solution for precision livestock farming, aggregating data from IoT devices managed by the BigBoxx middleware. This paper outlines a solution to overcome the challenges of storing and remotely accessing data collected by farm sensors. A cloud environment is introduced to synchronize e-Cattle data, comprising multi-tenant web applications. Also a memory management module for BigBoxx to address storage space constraints and prevent data overload. These innovations improve operational efficiency and contribute to the sustainability and productivity of the agricultural sector.*

Resumo. *A pecuária bovina é crucial para a economia brasileira, contribuindo significativamente para o PIB do país. Inovações tecnológicas são essenciais para aumentar sua eficiência e sustentabilidade. A plataforma e-Cattle surge como uma solução integrada para a pecuária de precisão, agregando dados de dispositivos IoT gerenciados pelo middleware BigBoxx. Este trabalho detalha uma solução para superar os desafios de armazenamento e acesso remoto aos dados coletados por sensores em fazendas. Introduz-se um ambiente em nuvem para a sincronização de dados da e-Cattle, compreendendo aplicações web multi-tenant, e um módulo de gerenciamento de memória para o Big-Boxx, visando contornar restrições de espaço e evitar excesso de dados. Essas inovações melhoram a eficiência operacional e contribuem para a sustentabilidade e produtividade do setor agropecuário.*

1. Introdução

A Tecnologia da Informação (TI), aliada à gestão do conhecimento, tem exercido um papel de influência positiva no desempenho, competitividade, planejamento e inovação nas instituições modernas (RAMOS; YAMAGUCHI; COSTA, 2020). É importante que as principais atividades econômicas do país adotem estrategicamente a TI para otimizar suas tomadas de decisão. A pecuária bovina, em particular, é fundamental para a economia do país. No ano de 2023, o Brasil contava com aproximadamente 202 milhões de

cabeças de gado, o que representa 12,18% do rebanho mundial (ABIEC, 2023). Como atividade de destaque no setor econômico, a pecuária pode se beneficiar da TI, uma vez que a implementação de tecnologias inovadoras pode aprimorar os processos de gestão e tomada de decisão, aumentando a eficiência em todas as etapas da cadeia produtiva (JUNIOR, 2020).

A Internet das Coisas, do inglês *Internet of Things* (IoT), em conjunto com a computação em nuvem, figura entre as tecnologias capazes de impulsionar a pecuária digital. Estima-se que, até ano de 2030, existam mais de 29 bilhões de dispositivos IoT em operação (VAILSHERY, 2022). Segundo Samuel e Sipes (2019), a IoT pode ser definida como uma coleção de dispositivos interconectados, que comunicam entre si e com serviços externos, utilizando padrões de protocolos de comunicação para compartilhamento de dados.

Com a crescente adoção ao paradigma de IoT, surgiram soluções inovadoras para problemas previamente conhecidos, tais como mensuração de dados climáticos e acompanhamento do desenvolvimento na criação animal. Contudo, essa evolução trouxe consigo diversos desafios, com destaque para as áreas de segurança, integração de múltiplas plataformas e padrões de arquiteturas de hardware (Dudhe et al., 2017). No Brasil cerca de 74,7% das propriedades em áreas rurais possuem acesso à internet, representando um crescimento de 16,9% entre 2019 e 2021 (NERY; BRITTO, 2022).

O e-Cattle é uma plataforma IoT voltada à pecuária de precisão, que visa agregar semanticamente e centralizar os dados coletados pelos dispositivos sensoriais, através de um *middleware* denominado BigBoxx. Essa plataforma foi concebida para ser implantada em propriedades rurais com o mínimo de esforço, ser de baixo custo e código aberto, garantindo sua confiabilidade e evolução contínua. Levando em consideração essas premissas, optou-se pela utilização do hardware Raspberry Pi (CARROMEU, 2019).

Assim, o BigBoxx é instalado fisicamente na propriedade rural e, inicialmente, disponibiliza as informações aos usuários apenas *in loco*, o que cria a necessidade de disponibilizar esses dados na nuvem, para permitir o acesso e gerenciamento remoto por parte dos proprietários e administradores.

Diversas abordagens para a sincronização de dados com a nuvem foram exploradas, cada uma atendendo a requisitos específicos para facilitar o acesso à informação em fazendas com variados níveis de conectividade. Algumas soluções são mencionadas em Dieng et al. (2017), Taneja et al. (2019), e Stólarik (2021). Contudo, nenhuma delas aborda todos os requisitos deste trabalho. Entre as necessidades não atendidas, destacam-se o gerenciamento de memória para dispositivos de baixo custo, como o Raspberry Pi, e a flexibilização da sincronização com a nuvem, permitindo maior controle, como a definição da quantidade de registros ou a priorização de um sensor.

Para enfrentar o desafio de gerenciamento remoto das informações coletadas pela plataforma *e-Cattle*, foi desenvolvido um ambiente em nuvem utilizando a arquitetura *multi-tenant*. Este ambiente possui a capacidade de conectar múltiplos BigBoxx a uma única propriedade e permite acessos simultâneos de diversos usuários, com permissões definidas com base em seus papéis no sistema. Com esta solução, é possível otimizar a gestão de dados na plataforma *e-Cattle*, permitindo o controle centralizado e facilitando o gerenciamento remoto das propriedades.

Foram implementadas aplicações para o gerenciamento e monitoramento remoto das propriedades rurais, disponibilizadas como imagens de *container*. Além disso, foram desenvolvidos mecanismos para a criação de ambientes isolados para cada propriedade, utilizando a arquitetura *multi-tenant*.

2. Fundamentação Teórica

Apresentam-se a seguir as definições e conceitos básicos das tecnologias empregadas no projeto, que incluem o desenvolvimento de software empregando o conceito *multi-tenant*, bem como uma visão geral da plataforma e-Cattle.

2.1. *Multi-tenant*

Uma abordagem de desenvolvimento de software, que é frequentemente utilizada, emprega a arquitetura de três camadas, que é composta por: o *frontend*, responsável pela interação com o usuário; o *backend*, que trata da lógica de negócios; e o armazenamento de dados, onde os dados são mantidos. Essa estrutura é comumente adotada em aplicações monolíticas, nas quais toda a lógica de programação é consolidada em um único pacote de aplicativos, o qual é então distribuído de forma integrada (FOWLER, 2017).

Uma alternativa é a segmentação do pacote de aplicativos em pequenas aplicações, conhecidas como microsserviços (PRASANDY et al., 2020). Fowler (2017) descreve o conceito de microsserviço como uma aplicação pequena e que executa apenas uma única tarefa de maneira eficiente, trazendo vantagens como modularidade, facilidade de manutenção e escalabilidade.

Adicionalmente, há a opção de utilizar uma instância de software que atenda diversos clientes ou sistemas de forma isolada, abordagem esta denominada *multi-tenant* (HAT, 2020). A arquitetura *multi-tenant* é empregada em sistemas *Software-as-a-Service* (SaaS), onde cada item armazenado na infraestrutura do sistema é gerenciado como um inquilino (TSAI; ZHONG; CHEN, 2016). Conforme mostrado na Figura 1, cada inquilino pode interagir com a aplicação como se fosse seu único usuário, sem que tenha acesso ou possibilidade de visualizar os dados de outro inquilino, garantindo assim o isolamento entre eles (MIETZNER; LEYMAN; PAPAZOGLU, 2008).

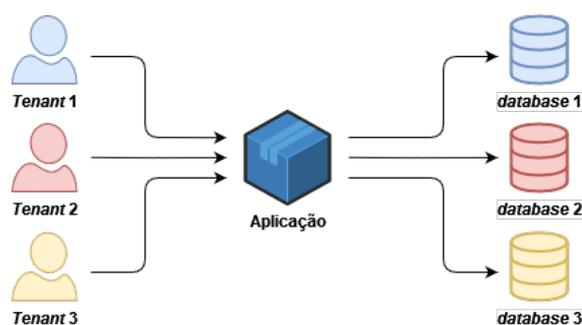


Figura 1. Arquitetura *multi-tenant*

Fonte: Elaborada pelo autor (2023)

As vantagens de desenvolver um sistema *multi-tenant* abrangem desde a segurança até a redução de custos operacionais e de desenvolvimento (KUPPUSAMY; KANIAPPAN; THIRUPATHI, 2015). Especificamente, o conceito de *multi-tenant* aprimora a integração de dispositivos IoT com a nuvem, conhecida como *edge computing* ou

computação em névoa, através do uso de contêineres para estabelecer uma divisão de carga de trabalho e do poder computacional entre os dispositivos e plataformas envolvidas (BUYA; SRIRAMA, 2019).

Diversos estudos destacam a importância da escalabilidade e do modelo *multi-tenant* para gerenciar o crescente volume de dados gerados por dispositivos IoT, com destaque para os trabalhos de Sellami, Kacem e Kacem (2020) e Batista et al. (2022). Adicionalmente, outros estudos reforçam a importância da segurança dos dados armazenados em nuvem dentro da arquitetura *multi-tenant* Masmoudi et al. (2019) e Liu Yun Yang e Xia (2020).

2.2. Plataforma e-Cattle

A parceria entre Empresa Brasileira de Pesquisa Agropecuária (EMBRAPA) e Faculdade de Computação (FACOM) resultou na criação de diversas soluções IoT para a coleta de dados para pecuária de precisão, no entanto, essas pesquisas foram desenvolvidas isoladamente, sem integração entre si. Diante disso, Carromeu (2019) desenvolveu uma plataforma IoT denominada e-Cattle¹, que tem por objetivo consolidar todos esses dados coletados de diferentes dispositivos e disponibilizar por meio de uma interface amigável e intuitiva ao usuário, possibilitando um monitoramento mais preciso da propriedade com informações atualizadas. O e-Cattle é composto por um *gateway* IoT, um dispositivo implantado na borda da rede que integra diversos dispositivos IoT heterogêneos (KRYLOVSKIY, 2015). Atualmente, a arquitetura do e-Cattle, conforme ilustrado na Figura 2, está completamente implementada, incluindo os componentes de sincronização em nuvem, objeto de estudo deste trabalho.

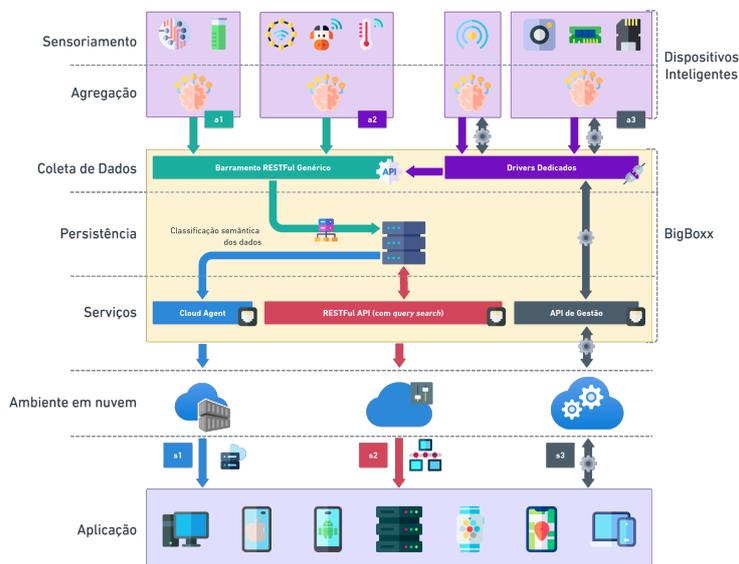


Figura 2. Ambiente em nuvem dentro da arquitetura do e-Cattle

Fonte: Adaptada de (CARROMEU, 2019).

A arquitetura da plataforma e-Cattle (Figura 2), divide-se em seis camadas:

¹<https://github.com/e-cattle>

- **Camada de Sensores:** Inclui artefatos de *hardware* e *software* responsáveis pela coleta de dados. Nessa camada encontram-se as diferentes soluções advindas da parceria EMBRAPA e FACOM, sem uma padronização quanto ao formato ou frequência das informações coletadas;
- **Camada de Agregação:** Responsável por encaminhar os dados coletados na camada anterior, seja por meio de software ou hardware. As camadas de **Sensores** e **Agregação** formam os **Objetos Inteligentes** da arquitetura;
- **Camada de Coleta de Dados:** Composta por um barramento de serviços *RESTful* que recebe requisições com os dados provenientes dos **Objetos Inteligentes**.
- **Camada de persistência:** Encarregada do armazenamento dos dados oriundos da camada de **Coleta de Dados**. A persistência é realizada em banco de dados *Not Only SQL* (NoSQL), após uma organização semântica, proporcionando agilidade e flexibilidade nas consultas aos dados ;
- **Camada de Serviços:** Oferece os dados estruturados às aplicações de alto nível da camada de **Aplicação** por meio de uma *Application Programming Interface* (API) *GraphQL*. Essa abordagem permite buscas com filtros personalizados, otimizando a produtividade (CÁCERES, 2020). As camadas **Coleta de Dados**, **Persistência** e **Serviços** são implementadas no *middleware* BigBoxx, instalado no hardware Raspberry Pi. A visualização dos dados para o proprietário é feita por meio de uma interface gráfica em formato de *dashboard*;
- **Camada de Aplicação:** Composta pelas aplicações de alto nível que utilizam os dados fornecidos pela camada de **Serviços**, auxiliando na tomada de decisões estratégicas na propriedade.

Há várias abordagens voltadas para a integração dos dispositivos sensoriais e a disponibilização de seus dados, tais como ThingSpeak², ThingsBoard³, Dojot⁴, entre outras. No entanto, o e-Cattle se destaca ao organizar semanticamente os dados da pecuária de corte por meio de seu *middleware* BigBoxx, tirando proveito de sua especialização nesse domínio de negócios.

Este trabalho implementou a camada denominada “Ambiente em Nuvem” e o *Cloud Agent* da camada “Serviços” para a persistência em nuvem dos dados armazenados localmente no *middleware* BigBoxx, viabilizando o gerenciamento remoto das propriedades. Conforme destacado na Figura 2, o ambiente em nuvem foi implementado entre as camadas de **Serviços** e **Aplicação** da plataforma e-Cattle, com sistemas orquestrados em *containers* por meio do Docker Swarm⁵.

A arquitetura do Docker Swarm possui dois tipos de nós, o nó *manager* responsável por agendar e distribuir as tarefas, e os nós *workers* que as executam. No ambiente Docker Swarm uma aplicação rodando em um *container* é conhecida por *service* e o conjunto de *services* é chamado de *stack* (SUSE, 2019).

3. Metodologia

Inicialmente, a plataforma e-Cattle era limitada ao uso de um único BigBoxx para cada propriedade rural, o que representava um desafio para sua implantação em fazen-

²<https://thingspeak.com>

³<https://thingsboard.io>

⁴<https://github.com/dojot/dojot>

⁵<https://docs.docker.com/engine/swarm/>

das de grande extensão territorial e com um volume significativo de dados sensoriais. Essa limitação poderia resultar no rápido esgotamento do espaço de armazenamento disponível.

Nesse contexto, propusemos a criação de um ambiente em nuvem para a plataforma e-Cattle, que viabiliza a sincronização com a nuvem das informações armazenadas localmente no *middleware* BigBoxx, além do desenvolvimento de um módulo para gerenciar a memória do BigBoxx. Este ambiente também possibilita a utilização de múltiplos dispositivos BigBoxx em uma mesma propriedade.

3.1. Ambiente em nuvem

Para a implementação do ambiente em nuvem do e-Cattle⁶, optamos pela arquitetura *multi-tenant*, utilizando o Docker Swarm para o gerenciamento dos diversos *containers*.

Conforme ilustrado na Figura 3, o ambiente em nuvem projetado para o e-Cattle compreende vários serviços organizados numa *stack*⁷. Entre esses serviços, destaca-se a aplicação *web* Gestor de Inquilinos⁸, projetada para permitir o gerenciamento de usuários e das propriedades que receberão dados de *middlewares* BigBoxx.

A aplicação *Scheduler Job*⁹ verifica se há fazendas recém criadas no ambiente em nuvem; em caso afirmativo ela é responsável por criar a *stack* parametrizada de serviços para a propriedade. Assim, quando o usuário com devida permissão autorizar o envio dos dados do *middleware* BigBoxx para a sincronização em nuvem, o conjunto de *containers* da propriedade estará apto a receber as requisições e conseqüentemente os dados.

Outra aplicação que compõem a *stack* do ambiente em nuvem é o Portal Web¹⁰, que permite gerenciar o vínculo dos *middlewares* BigBoxx à *stack* de uma determinada propriedade, por meio da aplicação *Cloud API*¹¹.

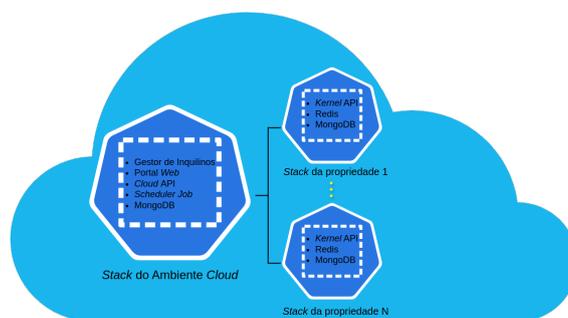


Figura 3. Arquitetura do ambiente em nuvem do e-Cattle

Fonte: Elaborada pelo autor (2023)

Cada propriedade rural registrada no ambiente em nuvem é alocada em um conjunto de *containers* dedicados, responsáveis por armazenar suas informações e disponibilizar os serviços de forma isolada das *stacks* de outras propriedades. Além disso,

⁶<https://github.com/e-cattle/swarm>

⁷*Stack* refere-se ao conjunto de serviços que compõem uma aplicação (PORTAINER, 2023)

⁸<https://github.com/e-cattle/manager>

⁹<https://github.com/e-cattle/farm>

¹⁰<https://github.com/e-cattle/web>

¹¹<https://github.com/e-cattle/cloud>

quando as aplicações web requisitam dados específicos de uma fazenda, elas interagem diretamente com a *stack* correspondente da propriedade envolvida, empregando assim a estratégia *multi-tenant*.

3.2. Sincronização dos Dados Sensoriais

Para realizar a sincronização dos dados com a nuvem, este trabalho implementou o *Cloud Agent* (Figura 2), neste contexto chamado de “Agente de Sincronismo”, um serviço essencial para garantir a persistência na nuvem dos dados coletados localmente por sensores. Essa sincronização ocorre assim que a rede da propriedade rural estabelece conexão com a internet, possibilitando o acesso remoto aos dados da fazenda por meio de aplicativos externos.

A sincronização dos dados é realizada por um agente de sincronização na nuvem, que se comunica com o módulo de sincronização do BigBoxx. Esse processo ocorre apenas em ambientes com acesso à internet, permitindo que dados coletados, especialmente aqueles de maior relevância, sejam sincronizados com a nuvem para uma visualização remota das condições da propriedade.

A Figura 4 ilustra o fluxograma que o gestor da propriedade deve seguir para permitir a sincronização dos dados no ambiente em nuvem do e-Cattle.

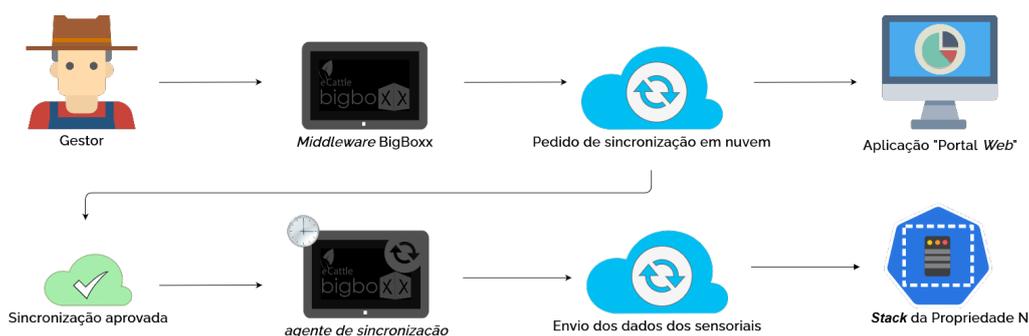


Figura 4. Fluxograma da sincronização no ambiente em nuvem do e-Cattle

Fonte: Elaborada pelo autor (2023)

Inicialmente, o gestor da propriedade precisa indicar no *middleware* BigBoxx o ID da fazenda que deseja realizar a sincronização com a nuvem, aguardando, em seguida, pela aprovação na aplicação Portal Web. Esta etapa de aprovação é necessária para confirmar que o ID fornecido corresponde realmente à fazenda em questão, evitando o risco de inserção de um ID incorreto. A solicitação deve ser aprovada por um usuário que detenha o papel de *Manager* ou *Owner*. Com a aprovação, o Agente de Sincronismo acessa as informações necessárias para a sincronização com a *stack* específica da propriedade, como o ID da Fazenda e o *token* de autenticação.

Para assegurar a integridade e a segurança dos dados a serem sincronizados com a aplicação na nuvem (*Cloud API*), todas as transações requerem autenticação. Essa autenticação é efetuada por meio de um *token JSON Web Token (JWT)*¹², fornecido pelo *kernel* do BigBoxx ao estabelecer conexão com a nuvem. Além do JWT, o ID da fazenda também é fornecido, sendo ambos essenciais para a sincronização dos dados.

¹²O JWT é um padrão (RFC 7519) para a transmissão segura de informações entre sistemas.

Diante do grande volume de dados coletados, existe o risco de esgotamento do espaço de armazenamento no BigBoxx. Para enfrentar esse desafio, foi implementado um módulo denominado *Garbage Collector* (GC). Visando preservar a integridade do BigBoxx, e prevenir a sobrecarga de dados armazenados, o GC é encarregado de remover os dados já sincronizados com a nuvem quando a memória disponível for menor ou igual a 20%. Na ausência de dados sincronizados e com o intuito de manter o desempenho do BigBoxx, estabeleceu-se que a capacidade em disco não deve ser menor que 10%. Caso isso ocorra, os registros mais antigos serão identificados e excluídos, repetindo-se esse procedimento até que se libere ao menos 10% da capacidade de memória.

O Agente de Sincronismo e o GC têm suas configurações armazenadas em um arquivo no formato JSON e podem ser alteradas de acordo com as necessidades e preferências do usuário. Ele contém o número de registros a serem sincronizados ou excluídos a cada execução, um campo usado para especificar quais *collections* devem ser priorizadas, de acordo com a importância definida pelo administrador, dois campos para determinar a porcentagem de memória a ser utilizada como critério inicial e final na exclusão de registros pelo GC, uma chave para indicar se a sincronização está ativada, entre outros aspectos relevantes.

4. Conclusão

A plataforma e-Cattle representa uma inovação para a pecuária de precisão, incorporando tecnologias atuais e adotando uma licença *open-source*. Esta pesquisa propôs uma solução para os desafios relacionados à limitação do espaço de armazenamento de dados persistidos no *middleware* BigBoxx e à restrição de acesso às informações exclusivamente *in loco*, por meio da implementação da sincronização em nuvem. Foram desenvolvidas duas aplicações Web *multi-tenant*, destinadas à criação de fazendas, usuários e ao estabelecimento de vínculos dos *middlewares* BigBoxx com as propriedades. Implementou-se também a aplicação *Scheduler Job*, responsável pela instanciação parametrizada das *stacks* das propriedades, além da aplicação *Cloud API*, criada para disponibilizar os *end-points* acessados pelas demais aplicações do ambiente em nuvem.

Este trabalho incluiu a implementação de um módulo eficaz para a sincronização de dados armazenados localmente, assegurando a padronização desses dados ao serem transferidos para a nuvem. A criação de um módulo dedicado ao gerenciamento da memória do BigBoxx foi uma resposta ao desafio imposto por sua capacidade de armazenamento limitada. Esse módulo não apenas previne a sobrecarga de dados, como também garante a continuidade e a eficiência das operações, essenciais para a sustentabilidade de longo prazo das aplicações na pecuária de precisão.

As principais contribuições deste trabalho englobam a criação do ambiente em nuvem da plataforma IoT e-Cattle utilizando a arquitetura *multi-tenant*, a orquestração de *containers* por meio do Docker Swarm, e a sincronização de múltiplos *middlewares* BigBoxx na mesma propriedade. Essas inovações ampliam o uso do e-Cattle para aplicações remotas, possibilitando o acesso aos dados sensoriais mesmo fora da rede da propriedade e facilitando o gerenciamento e a tomada de decisões à distância.

Essas inovações prometem não apenas melhorar a eficiência operacional das fazendas, mas também contribuir para a sustentabilidade e a produtividade do setor agropecuário.

Atualmente a plataforma e-Cattle está em fase de testes. Uma possível melhoria futura seria estabelecer uma política de atualizações em todos os componentes e artefatos da plataforma. Uma estratégia para a implementação dessa política de atualizações envolve a adoção da versão *Long-term Support (LTS)* mais recente de todas as tecnologias utilizadas, assegurando assim a estabilidade e o suporte prolongado do e-Cattle.

Referências

ABIEC. Perfil da pecuária no brasil - capítulo 3 - 2023. *Associação Brasileira das Indústrias Exportadoras de Carne*, Associação Brasileira das Indústrias Exportadoras de Carne, 2023.

BATISTA, C. et al. *Towards a Multi-Tenant Microservice Architecture: An Industrial Experience*. 2022. 553-562 p.

BUYA, R.; SRIRAMA, S. *Fog and Edge Computing: Principles and Paradigms*. [S.l.]: Wiley, 2019. (Wiley Series on Parallel and Distributed Computing).

CARROMEU, C. *e-Cattle: Uma Plataforma de IoT para Pecuária de Precisão*. Tese (Doutorado) — Universidade Federal de Mato Grosso do Sul, Campo Grande, MS, Brasil, 2019.

CÁCERES, B. d. A. *Uma arquitetura de consulta semântica de dados sensoriais no barramento de serviços IoT na Plataforma e-Cattle*. Campo Grande, MS, Brasil: [s.n.], 2020.

DIENG, O. et al. A study on iot solutions for preventing cattle rustling in african context. In: *Proceedings of the Second International Conference on Internet of Things, Data and Cloud Computing*. New York, NY, USA: Association for Computing Machinery, 2017. (ICC '17). ISBN 9781450347747.

Dudhe, P. V. et al. *Internet of Things (IOT): An overview and its applications*. 2017. 2650-2653 p. Acesso em: 12 dez. 2019.

FOWLER, S. *Microserviços prontos para a produção: Construindo sistemas padronizados em uma organização de engenharia de software*. [S.l.]: Novatec Editora, 2017.

HAT, R. *Arquitetura multitenancy*. [S.l.]: Red Hat, 2020. Disponível em: <https://www.redhat.com/pt-br/topics/cloud-computing/what-is-multitenancy>. Acesso em: 15 de abril de 2023.

JUNIOR, G. B. M. Forças motrizes para a agropecuária brasileira na próxima década: implicações para a agricultura digital. *Agricultura digital: pesquisa, desenvolvimento e inovação nas cadeias produtivas.*, Embrapa Informática Agropecuária, 2020. ISSN 978-65-86056-37-2.

KRYLOVSKIY, A. *Internet of Things gateways meet linux containers: Performance evaluation and discussion*. 2015. 222-227 p.

KUPPUSAMY, S.; KANIAPPAN, V.; THIRUPATHI, D. Design and development of multi-tenant web framework. In: . [S.l.: s.n.], 2015. v. 48, p. 180 – 191. ISSN 1877-0509. International Conference on Computer, Communication and Convergence (ICCC 2015).

LIU YUN YANG, W. G. Z.; XIA, J. A multi-tenant usage access model for cloud computing. *Computers, Materials and Continua*, v. 64, n. 2, p. 1233–1245, 2020. ISSN 1546-2226.

MASMOUDI, F. et al. Accountability management for multi-tenant cloud services. *International Journal of Grid and Utility Computing*, v. 10, p. 141, 01 2019.

MIETZNER, R.; LEYMANN, F.; PAPAZOGLU, M. P. *Defining Composite Configurable SaaS Application Packages Using SCA, Variability Descriptors and Multi-tenancy Patterns*. 2008. 156-161 p.

NERY, C.; BRITTO, V. *Pesquisa Nacional por Amostra de Domicílios*. [S.l.]: IBGE, 2022.

PORTAINER. *Stacks*. [S.l.]: Portainer Documentation., 2023. Disponível em: <https://docs.portainer.io/user/docker/stacks>. Acesso em: 08 de janeiro de 2024.

PRASANDY, T. et al. Migrating application from monolith to microservices. In: *2020 International Conference on Information Management and Technology (ICIMTech)*. [S.l.: s.n.], 2020. p. 726–731.

RAMOS, N.; YAMAGUCHI, C.; COSTA, U. Brazilian journal of development tecnologia da informação e gestão do conhecimento: estratégia de competitividade nas organizações information technology and knowledge management: competitiveness strategy in organizations. *Brazilian Journal of Development*, v. 6, p. 144–161, 01 2020.

Samuel, A.; Sipes, C. *Making Internet of Things Real*. 2019. 10-12 p. Disponível em: <https://ieeexplore.ieee.org/document/8835419>. Acesso em: 12 dez. 2019.

SELLAMI, W.; KACEM, H. H.; KACEM, A. H. Dynamic provisioning of service composition in a multi-tenant saas environment. *Journal of Network and Systems Management*, v. 28, 04 2020.

STOLÁRIK, B. M. A mobile system enabling cloud connectivity of iot devices in smart food industry. 2021.

SUSE. *Kubernetes vs Docker Swarm: Comparison of Two Container Orchestration Tools*. [S.l.]: SUSE, 2019. Disponível em: https://www.suse.com/c/rancher_blog/kubernetes-vs-docker-swarm-comparison-of-two-container-orchestration-tools/. Acesso em: 9 de abril de 2023.

TANEJA, M. et al. Smartherd management: A microservices-based fog computing–assisted iot platform towards data-driven smart dairy farming. *Software: practice and experience*, Wiley Online Library, v. 49, n. 7, p. 1055–1078, 2019.

TSAI, W.-T.; ZHONG, P.; CHEN, Y. Tenant-centric sub-tenancy architecture in software-as-a-service. *CAAI Transactions on Intelligence Technology*, v. 1, n. 2, p. 150 – 161, 2016. ISSN 2468-2322.

VAILSHERY, L. S. *Number of Internet of Things (IoT) connected devices worldwide from 2019 to 2021, with forecasts from 2022 to 2030*. [S.l.]: Statista, 2022.