Aperfeiçoamento da AgDataBox-Data-API para gerenciamento de dados em agricultura digital

Oswaldo Beltrani Neto[1], Ricardo Sobjak[1], Claudio Leones Bazzi[1], Estefani Aparecida Rup Roza[1], Kelyn Schenatto[1], Eduardo Godoy de Souza[2], Pedro Luiz de Paula Filho[1]
[1]Universidade Tecnológica Federal do Paraná, Campus Medianeira - Medianeira, Brasil
[2]Universidade Estadual do Oeste do Paraná, Campus Cascavel - Cascavel, Brasil
E-mail: osvaldobeltrame@alunos.utfpr.edu.br, ricardosobjak@utfpr.edu.br, bazzi@utfpr.edu.br, estefani@alunos.utfpr.edu.br, kschenatto@utfpr.edu.br, eduardo.souza@unioeste.br, pedrol@utfpr.edu.br

***Abstract*—Digital agriculture leverages technologies focused on rural digitization to support decision-making by integrating data from various sources. The Web platform AgDataBox, offered for free, provides APIs and applications to meet the demands of digital agriculture. This work aimed to enhance the data API, enable new features, and adopt current technologies. The development used the Visual Studio Code IDE, Java programming language, Spring Boot framework, and PostgreSQL database with PostGIS extension. The API follows the REST architecture, organized into layers of controllers, services, repositories, models, and data transfer objects (DTOs). The request tests were performed using the REST Client extension. As a result, a new API was created that provides secure HTTP operations for managing data, such as creating, editing, deleting, and retrieving. Authentication mechanisms using JWT tokens and authorization were implemented. Data access is divided into public, available to all users, and private, accessible only to the owner or authorized users. In conclusion, the API is ready to manage agricultural data and enable integration with other applications.**

***Keywords*—Data integration; RESTful API; Spatial data.**

***Resumo*—A agricultura digital utiliza tecnologias voltadas à digitalização rural para apoiar a tomada de decisão, integrando dados de diversas fontes. A plataforma Web AgDataBox, gratuita, oferece APIs e aplicações para atender às demandas da agricultura digital. Este trabalho teve como objetivo aprimorar a API de dados, permitindo novos recursos e a adoção de tecnologias atuais. O desenvolvimento foi realizado com a IDE Visual Studio Code, a linguagem Java, o framework Spring Boot e o banco de dados PostgreSQL com extensão PostGis. A API segue a arquitetura REST, organizada em camadas de controladores, serviços, repositórios, modelos e objetos de transferência de dados (DTO). Os testes de requisições foram realizados com a extensão REST Client. Como resultado, foi criada uma nova API que disponibiliza operações HTTP seguras para gerenciar dados, como criar, editar, deletar e consultar. Mecanismos de autenticação via token JWT e de autorização foram implementados. O acesso aos dados é dividido em públicos, disponíveis para todos os usuários, e privados, acessíveis apenas ao proprietário ou a usuários autorizados. Conclui-se que a API está pronta para gerenciar dados agrícolas e possibilitar a integração com outras aplicações.**

***Palavras-chave*—Integração de dados; RESTful API; Dados espaciais.**

I. Introdução

A crescente demanda por produção alimentícia, aliada à necessidade de otimização de recursos ambientais tem impulsionado transformações na agricultura moderna. Esse cenário de alta competitividade exige a adoção de práticas eficientes e sustentáveis na agricultura contemporânea. As tecnologias emergentes como Internet das coisas (IoT), computação em nuvem, robótica e inteligência artificial têm o potencial de transformar a agricultura com o desenvolvimento de estratégias agrícolas mais sofisticadas, permitindo a automação e otimização de processos produtivos voltados à tomada de decisão [1].

É nesse cenário que a agricultura digital (AD), uma abordagem que integra tecnologias avançadas de comunicação, informação e análise espacial que permitem aos produtores rurais planejar, monitorar e gerenciar as atividades operacionais e estratégicas do sistema produtivo [2]. O objetivo da AD é usar todas as informações e conhecimentos disponíveis para permitir a automação de processos sustentáveis na agricultura. A AD atingiu considerável desenvolvimento nas últimas duas décadas devido à disponibilidade de sensores mais baratos e potentes, de atuadores e microprocessadores, comunicação celular de alta largura de banda, comunicação em nuvem e *Big Data*. Como resultado, o fluxo de informação já não é proveniente apenas do equipamento agrícola usado, mas também de novos serviços que estão sendo oferecidos com novos algoritmos que transformam dados em inteligência útil [3].

Neste paradigma de AD, as tecnologias digitais são inseridas em todas as etapas da cadeia produtiva, para que os interessados obtenham vantagens competitivas e contribuam com benefícios socioambientais [4]. Neste contexto, uma grande quantidade de dados é disponibilizada e o desafio é agregar valor a eles, com a inserção dos portais de dados e plataformas de trabalho. Nos portais, o consumidor final pode visualizar seus dados sem a necessidade de inseri-los manualmente. Já por meio de plataformas, este consumidor poderá transformar os dados em novas e mais poderosas informações [5].

Entre as diversas tecnologias disponíveis para integração de software, destacam-se os serviços *Web*, que se

consolidaram como uma das principais técnicas de interoperabilidade em redes de computadores. Esses serviços permitem que diferentes aplicações, independentemente de suas plataformas ou linguagens de programação, possam se comunicar de maneira simplificada. A troca de informações ocorre por meio do protocolo de transferência de hipertexto (HTTP, *Hypertext Transfer Protocol*), amplamente utilizado na *Web*. Esse protocolo é um padrão que possibilita a transmissão de dados entre servidores e clientes, fornecendo uma interface comum para a troca de mensagens. Assim, uma Interface de Programação de Aplicações (API, *Application Programming Interface*) de serviço *Web* define um formato para a representação das mensagens trocadas, geralmente utilizando *JavaScript Object Notation* (JSON) ou *Extensible Markup Language* (XML) [6]. As APIs desempenham um papel essencial na integração de sistemas, permitindo que diferentes softwares possam se comunicar. Segundo Fielding [7], por meio da arquitetura REST, é possível integrar sistemas, estabelecer uma comunicação através de mensagens e informações, mantendo a semântica dos dados e assegurando uma segurança adequada na transmissão e distribuição das informações.

A plataforma *Web* AgDataBox (ADB) foi desenvolvida com o objetivo de ser uma solução integrada, para centralizar e organizar dados, softwares, procedimentos e metodologias para a Agricultura de Precisão (AP) e AD. Esta plataforma possui a ADB-DATA-API, que permite o compartilhamento de dados entre diferentes dispositivos e aplicações desse ecossistema. Nela são armazenados dados oriundos de diferentes fontes de dados, como análises laboratoriais, de máquina, satélite, sensores, dentre outros. A primeira versão da ADB-DATA-API [8] foi disponibilizada para uso em dezembro de 2017 e está registrada junto ao INPE (BR 51 2018 000899-2). Portanto, o objetivo deste trabalho foi aperfeiçoar a ADB-DATA-API, tornando-a ainda mais eficiente e confiável o gerenciamento de dados e a expansão de softwares integrados.

II. Material e Métodos

A. *Arquitetura da plataforma AgDataBox*

A nova ADB-DATA-API foi integrada na arquitetura de microserviços da plataforma ADB [9]. Existe uma divisão lógica presente em cada parte da plataforma, na camada de *front-end*, o usuário interage com os artefatos de softwares chamados de aplicação, assim como a ADB-Map, *ADB-Mobile* e *ADB-Remote Sensing*. Já os artefatos que fornecem armazenamento e execução de serviços ficam na camada de *back-end*. Por exemplo, microsserviços são disponibilizados para performance de análise de dados estatísticos, limpeza e preparação de dados (ADB Statistics API), análises e operações com dados espaciais (ADB Spatial API), agrupamento de dados e métricas para estimativas de *cluster* (*ADB Clustering API*), retificação de zona de manejo (*ADB Rectification API*) (Fig. 1).

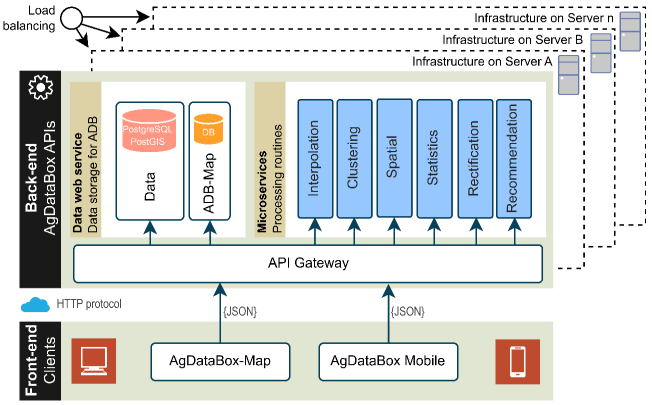

Fig. 1. Representação da arquitetura de microsserviços da plataforma AgDataBox, onde a camada de *back-end* contém os microsserviços, incluindo rotinas e dados, enquanto a camada de *front-end* contém as aplicações que consomem os microsserviços. Fonte: [9].

A AgDataBox foi desenvolvida pensando em permitir a integração de diferentes aplicações de forma centralizada em uma única localização. Isso facilita o trabalho de desenvolvedores, que podem utilizar dos recursos complexos já implementados na API, mostrado na Fig. 2.

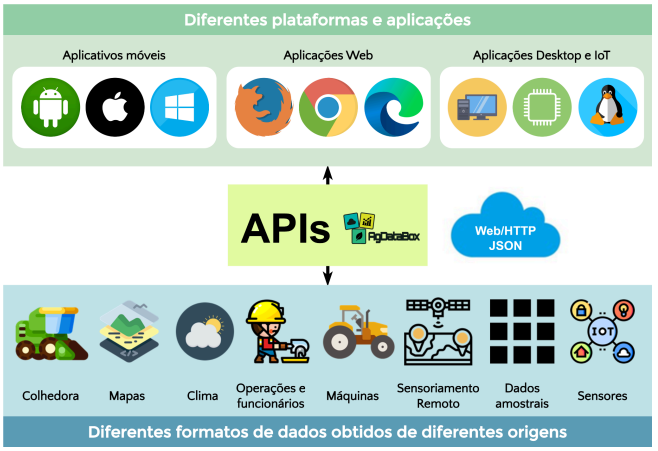

Fig. 2. Estrutura de integração de dados e operações da plataforma AgDataBox. Diferentes tipos de dados são padronizados via JSON e encaminhados utilizando o protocolo HTTP para a API, fornecendo acesso a base de dados. Diferentes tipos de aplicações que podem ser conectadas. Fonte: autoria própria.

A ADB-DATA-API está hospedada no *data center* da UTFPR. O ambiente da plataforma ADB está em uma máquina virtual com 16 núcleos de processador, 32 GB de memória RAM e 225 GB de armazenamento, com sistema operacional Debian e utilização de Docker para gerenciar os contêineres.

B. *Tecnologias utilizadas*

As tecnologias para desenvolvimento de software usadas na construção da API consistiram na linguagem de programação Java, versão 17, com recursos da plataforma *Java Enterprise Edition* (JEE); Maven para gerenciamento de projetos; *framework Spring Boot* para construir recursos REST e organizar o fluxo de solicitações e respostas na API,

substituindo a antiga estrutura de projeto, que utilizava o framework VRaptor, que está a muito tempo ser receber atualizações.

Os dados são armazenados no sistema de gerenciamento de banco de dados (SGBD) PostgreSQL, já que além de ser um projeto livre e de estabilidade reconhecida, possui interfaces de programação nativas para Java, C/C ++, .Net, Perl, Python, Ruby, entre outros, e possui uma extensão geográfica (PostGIS) que por meio de cerca de 1500 funções disponibilizadas permite a execução de diversos tipos de análises da forma geográfica.

Já como abordagem de estrutura dos dados para comunicação o padrão escolhido foi a Notação de Objetos JavaScript (JSON), pois é um formato que permite uma maior compactação dos dados, além de ser uma forma de notação que é mais legível para o usuário.

Para documentar os recursos da ADB-DATA-API foi utilizada a ferramenta Swagger, que é um framework de código aberto, que permite criar documentações interativas para APIs, permitindo que testes práticos integrados sejam disponibilizados para os usuários.

A ADB-DATA-API foi projetada para executar em contêiner da ferramenta Docker, a fim de facilitar a execução em diferentes plataformas.

As solicitações para manipulação de dados no ADB-API são feitas pelo protocolo HTTP, usando um método de solicitação (*get, post, put ou delete*), um identificador uniforme de recursos (URI) e uma representação de dados no formato padronizado JSON. A resposta entregue pelo servidor que hospeda o ADB-DATA-API é uma mensagem contendo algumas informações em seus cabeçalhos, como o status da resposta e o conteúdo principal solicitado, também no formato JSON (Fig. 3).

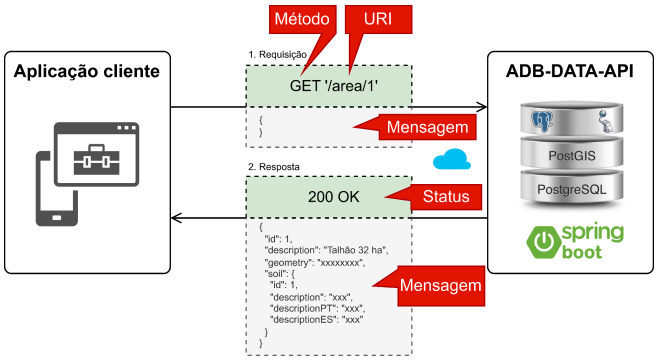

Fig. 3. Representação do processo de requisição e resposta de dados na comunicação entre uma aplicação cliente e o servidor da ADB-API. Fonte: autoria própria.

C. *Estudo de caso*

Dados de uma área agrícola comercial de 20 ha^{-1} localizado no município de Serranópolis do Iguaçu, estado do Paraná, sul do Brasil, com coordenadas centrais −54.01232307º (longitude) e −25.39526307º (latitude) no Datum WGS 1984, foram utilizados para demonstrar algumas das funcionalidades ADB-DATA-API (Fig. 4). A grade amostral é composta por pontos com distâncias irregulares, localizados ao longo de uma linha imaginária entre as curvas de nível seguindo a topografia do terreno. A densidade amostral de 2,6 ha^{-1}, que atende o sugerido de 1 amostra ha^{-1} [10] a 2,5 amostras ha^{-1} [11][12].

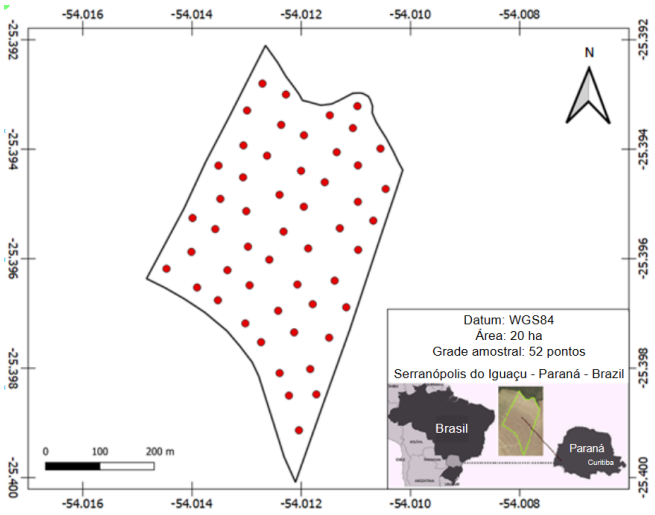

Fig. 4. Localização do experimento e grade amostral de 52 pontos na área de 20 ha em Serranópolis do Iguaçu, no estado do Paraná, Sul do Brasil, delimitada pelo contorno preto. As coordenadas geográficas estão em graus decimais (WGS 1984). A distância mínima e máxima entre os pontos amostrais varia entre 45 e 706 m. Fonte: autoria própria.

Com dados experimentais, foram aplicados testes de execução HTTP que abrange diversos aspectos da comunicação entre cliente e servidor. Inicialmente, foram identificados os endpoints principais da API, e casos de teste foram elaborados para cada um, contemplando cenários normais de execução. Utilizando a extensão *REST Client* do *Visual Studio Code*, foram realizadas requisições HTTP (*GET, POST, PUT, DELETE*) com diferentes parâmetros e payloads em formato JSON. A análise das respostas incluiu a verificação de códigos de status, tempos de resposta e a integridade dos dados retornados.

III. Resultados e Discussão

A ADB-DATA-API foi criada para suprir a necessidade de gerenciamento e integração de dados agrícolas. Assim, é possível fazer com que aplicações diferentes tenham interoperabilidade, facilitando o compartilhamento da informação entre softwares para computador, aplicativos para dispositivos móveis, portais *Web* e dispositivos de IoT. A primeira versão da ADB-DATA-API foi desenvolvida no idioma português. A partir da segunda versão, a representação de alguns recursos foram reestruturados e traduzidos para o idioma inglês. Agora, foi atualizada e disponibilizada como um microsserviço na plataforma ADB.

Os recursos disponibilizados na API foram categorizados da seguinte forma (Fig. 5):

- *Agricultural*: recursos para gerenciamento de dados agrícolas, como o cadastro de áreas, atributos agrícolas, safra, cultura, zonas de manejo, amostragens e grades amostrais, solo e variedade de sementes;
- *Management*: recursos gerais para controle de informações em recursos associados na API;
- *Classification*: determinação de níveis de classificação de atributos disponibilizados por entidades;
- *Spatial*: recursos com informação geográfica associada;
- *News*: recursos para disponibilização de notícias para aplicações clientes;
- *Security*: gerenciamento de segurança e compartilhamento dos dados;
- *Records*: registro de ocorrências na lavoura;
- *IoT*: gerenciamento de dados de estrutura IoT;
- *General*: registros gerais;
- *Operation*: gerenciamento de operações em áreas agrícolas.

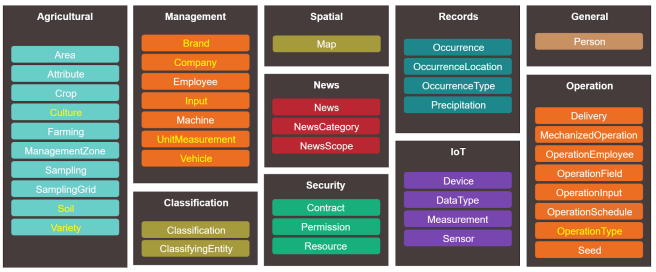

Fig. 5. Recursos disponibilizados na ADB-DATA-API. Fonte: autoria própria.

Alguns recursos permitem o registro de informação associada no contexto geográfico, isto é, com uma coordenada geográfica associada. Uma geometria espacial pode ser especificada por uma representação textual ou binária. Adotamos a linguagem de marcação de texto *Well-Known Text* (WKT) para representar objetos de geometria vetorial em um mapa. Esse formato foi originalmente definido pelo *Open Geospatial Consortium* (OGC) e, portanto, é um padrão conhecido para representação espacial de dados. Um dos motivos de adotar este formato é a simplicidade como a informação é representada, diminuindo a sintaxe em relação a outros formatos, como o GeoJson.

O projeto desta API foi estruturado para ser modular e disponibilizado em diferentes ambientes. Atualmente está disponibilizado em um ambiente de acesso público (https://adb.md.utfpr.edu.br/api/data/v3), que pode ser consumido por diferentes aplicações experimentais. Entretanto, as organizações interessadas podem disponibilizar dentro da infraestrutura interna, com acesso e gerenciamento restrito dos recursos (Fig. 6). A disponibilização é facilitada por utilizar contêineres da plataforma Docker, que podem ser facilmente providos em diferentes infraestruturas de tecnologia da informação.

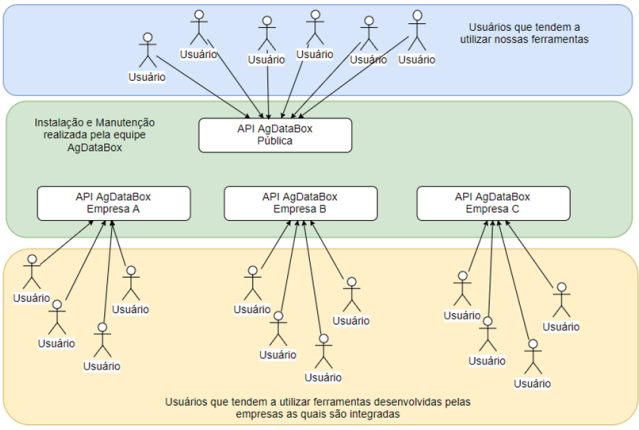

Fig. 6. Demonstração da disponibilização da ADB-DATA-API em diferentes ambientes de execução. Fonte: autoria própria.

Acessos restritos foram implementados na API, tendo como objetivo incorporar segurança, rotinas e protocolos que precisam seguir uma operação correta na plataforma, assim como padrões de comunicação e formatos de dados.

Os recursos da API podem ter diferentes visibilidades: públicos, acessíveis por todos os usuários; privados, acessível somente por pessoas autorizadas. Os recursos como área, amostra, mapa, zona de manejo, dentre outros, são privativos do usuário e podem ser compartilhados com outros usuários. Os recursos como cultura, solo, atributo, unidades de medida, dentro outros,são de acesso público e compartilhados automaticamente entre os usuários.

A. Documentação para desenvolvedores

Como a API permitirá que outras aplicações utilizem os recursos implementados, foi desenvolvido um site documental (https://adb.md.utfpr.edu.br/api/data/v3/index.html, Fig. 7) para atender a esse requisito de maneira dinâmica e funcional. Nesse ambiente, todos os recursos de armazenamento e recuperação de dados são descritos de forma clara e objetiva. O site de documentação também incorpora a própria API.

Fig. 7. Documentação da ADB-API acessível pela URL https://adb.md.utfpr.edu.br/apidata/v3/index.html. Fonte: autoria própria.

B. Testes de consumo dos recursos da API

Todos os recursos foram testados e estão funcionando corretamente, os testes foram realizados para avaliar o desempenho das operações (obtenção, criação, atualização e deleção) em diferentes cenários, garantindo que os serviços da API respondessem de acordo com o esperado em um ambiente de operação normal.

Códigos HTTP que a API pode retornar:

- 200 (OK): A requisição foi processada e obteve sucesso;
- 400 (*Bad Request*): A requisição foi mal formada ou contém parâmetros inválidos;
- 401 (*Unauthorized*): A requisição não foi autorizada, seja por falta de credenciais ou por credenciais inválidas;
- 404 *(Not Found*): O recurso solicitado não foi encontrado no servidor.

Os testes mostrados na Tabela 1 foram executados em um cenário normal de operação, com carga estável e sem sobrecargas que pudessem alterar o desempenho dos serviços. Cada operação foi testada em diferentes recursos de aplicação (Amostragem, Mapa, Zona de Manejo e Área), medindo o tempo médio de execução para operações de leitura (obter registros, criação, atualização e deleção).

TABELA 1
TEMPO MÉDIO DE EXECUÇÃO (MS) DOS RECURSOS EM UM CENÁRIO NORMAL DE OPERAÇÃO

Operação	Recurso			
	Amostragem	Mapa	Zona de Manejo	Área
Obter todos os registros	78 ms	70 ms	85 ms	91 ms
Obter um registro	60 ms	370 ms	375 ms	50 ms
Criar um registro	125 ms	669 ms	641 ms	133 ms
Atualizar um registro	122 ms	738 ms	763 ms	85 ms
Remover um registro	125 ms	348 ms	304 ms	130 ms

Fonte: autoria própria.

Nos testes de consumo dos recursos Tamanho do JSON (em caracteres) e *pixels* de cada requisição:

- No recurso amostragem, há 51 *pixels* com 7842 caracteres em JSON. Estes *pixels* são locais específicos no terreno onde os dados foram coletados, e são utilizados para representar estatísticas ou outras informações;
- Em mapas, temos 3019 *pixels* com 261682 caracteres em JSON. Esses *pixels* são unidades gráficas que descrevem a visualização espacial do mapa no sistema;
- Em Zona de Manejo, temos 3021 *pixels* com 234925 caracteres em JSON associado, representando os dados geográficos da área de gestão;
- Em Área, temos 57 *pixels* com 2002 caracteres em JSON. Esses pontos são usados para delinear a forma da área representada no mapa, podendo variar dependendo do nível de detalhe necessário para definir uma geometria espacial.

Um caso separado é a operação Obter todos os registros, para fins de testes, cada classe que requisita todas as informações, contém 10 registros anteriormente salvos. Assim, Amostras tem 510 *pixels*, com 78420 caracteres, Mapas contém 30210 *pixels* com 2616820 caracteres, Zona de manejo com 30210 *pixels* e 2349250 e em Área são 570 *pixels* com 20020 caracteres.

A Tabela 1, referente ao tempo médio de execução em milissegundos (ms) para cada operação e recurso, evidenciando que operações de leitura (Obter um registro, Obter todos os registros) são mais rápidas em comparação com as operações de escrita (Criar um registro, Atualizar um registro). Demonstrando também uma grande eficiência ao lidar com grandes volumes de dados.

C. Aplicações clientes

A ADB-DATA-API tem o potencial de atender várias aplicações voltadas para a agricultura, sendo algumas delas (Fig. 8):

a) AgDataBox-Mobile (ADB-Mobile) [13] é um aplicativo para dispositivos móveis que auxilia produtores rurais no registro e organização de operações em suas propriedades, mantendo um histórico de safra com dados armazenados localmente e em um servidor. Ele permite operações offline, com sincronização posterior com a ADB-DATA-API.

b) AgDataBox-Admin é uma aplicação *web* em desenvolvimento que visa administrar os recursos da ADB-DATA-API, utilizando tecnologias gratuitas para facilitar sua adoção. A plataforma permitirá visualizar, criar, editar e excluir dados do usuário, além de gerenciar o compartilhamento de informações entre os usuários.

c) AgDataBox-Map (ADB-Map) [14] é uma aplicação web gratuita que possibilita interação com mapas temáticos agrícolas. Através dela, usuários podem gerenciar áreas, dados amostrais e criar mapas utilizando métodos de interpolação como MM, IDP e KO. A aplicação está integrada com a ADB-DATA-API, permitindo a busca e armazenamento de dados do usuário.

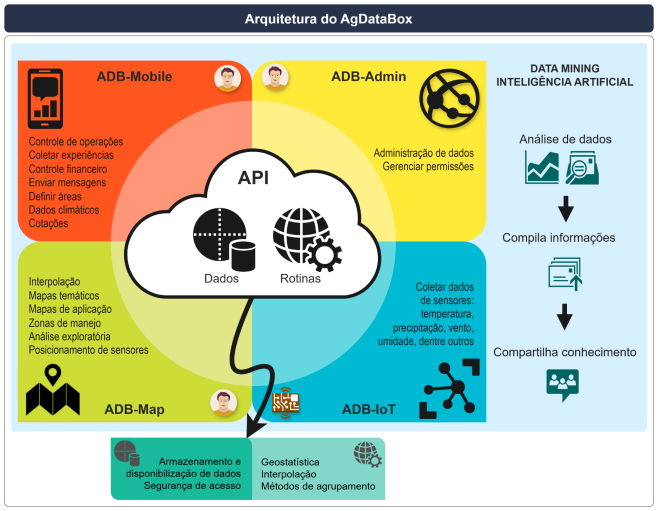

Fig. 8. Representação da plataforma AgDataBox com suas aplicações associadas. Fonte: [15].

IV. Conclusão

A implementação da ADB-DATA-API oferece uma solução para a integração e gerenciamento de dados em agricultura digital, centralizando a manutenção e permitindo que aplicações cliente se conectem sem se preocupar com protocolos complexos. Sua estrutura minimiza falhas, acelera o desenvolvimento de aplicações através de recursos já disponíveis, e facilita a escalabilidade ao permitir a adição de novas aplicações sem grandes reestruturações. Além disso, proporciona liberdade de escolha de ferramentas e linguagens para desenvolvedores e pode ser integrada a ambientes corporativos, tornando-a uma escolha ideal para otimizar processos e aumentar a eficiência.

1 Agradecimentos

Agradecemos ao Conselho Nacional de Desenvolvimento (CNPq), à Universidade Tecnológica Federal do Paraná (UTFPR), à Fundação Araucária e ao Laboratório Multiusuário de Agrotecnologias (AGRILAB-MD) pelo apoio concedido para a realização da pesquisa.

2 Referências

[1] JAVAID, M.; HALEEM, A.; SINGH, R. P.; SUMAN, R. Enhancing smart farming through the applications of Agriculture 4.0 technologies. *International Journal of Intelligent Networks*, v. 3, 2022, p. 150-164, 2022. https://doi.org/10.1016/j.ijin.2022.09.004.

[2] BOLFE, É. L.; JORGE, L. A. D. C.; SANCHES, I. D.; LUCHIARI JÚNIOR, A.; DA COSTA, C. C.; VICTORIA, D. D. C.; INAMASU, R. Y.; GREGO, C. R.; FERREIRA, V. R.; RAMIREZ, A. R. Precision and Digital Agriculture: Adoption of Technologies and Perception of Brazilian Farmers. *Agriculture*, v. 10, n. 12, 653, 2020. https://doi.org/10.3390/agriculture10120653

[3] CEMA - European Agricultural Machinery. *Digital farming:* what does it really mean? Brussels: CEMA, 2017. 9p.

[4] MASSRUHÁ, S. M. F. S.; LEITE, M. A. d. A.; LUCHIARI JUNIOR, A.; EVANGELISTA, S. R. M. *Digital Transformation in the Field Towards Sustainable and Smart Agriculture*. 2023.

[5] SOBJAK, R.; SOUZA, E. G. de.; BAZZI, C. L.. *AgDataBox-Data-API*: documentação da API para gestão de dados agrícolas da plataforma AgDataBox. Medianeira: AGRILAB/LAMAP, 2021.

[6] GRAHL, M.; BLUHM, T.; GRÜN, M.; HENNIG, C.; HOLTZ, A.; KROM, J.G.; KÜHNER, G.; LAQUA, H.; LEWERENTZ, M.; RIEMANN, H.; SPRING, A.; WERNER, A. Archive WEB API: A web service for the experiment data archive of Wendelstein 7-X. *Fusion Engineering and Design*, v. 123, n. 1, p. 1015-1019, 2017.

[7] FIELDING, R. T. *Architectural Styles and the Design of Network-Based Software Architectures*. 2000. 162 f. Tese (Doutorado) - Curso de Computer Science, Departamento de Computer Science, University Of California, Irvine, 2000.

[8] BAZZI, C. L.; JASSE, E. P.; GRAZIANO MAGALHÃES, P. S.; MICHELON, G. K; SOUZA, E. G.; SCHENATTO, K.; SOBJAK, R. AgDataBox API - Integration of data and software in precision agriculture. *SoftwareX*, v. 10, p. 100327, 2019.

[9] SOBJAK, R.; SOUZA, E. G. DE; BAZZI, C. L.; SCHENATTO, K.; BETZEK, N. M.; GAVIOLI, A. Incorporation of computational routines in a Microservice Architecture in AgDataBox platform. *Sustainable Computing: Informatics and Systems*, In press, 101038, 2024.

[10] FERGUSON, R. B.; HERGERT, G. W. Soil sampling for precision agriculture. *Ext. Precis. Agric.,* v. 1, n. 1, p. 1-4. 2009.

[11] JOURNEL, A. G.; HUIJBREGTS, C. J. *Mining Geostatistics*. London, New York, San Francisco: Academic Press. 1978.

[12] DOERGE, T. A. *Site-Specific Management Guidelines*. Potash & Phosphate Institute, Norcross. 2000.

[13] SCHENATTO, K.; SOUZA, E. G.; BAZZI, C. L.; GAVIOLI, A.; MICHELON, G. K. Software de gerenciamento de dados agrícola: AGDATAFIELD_MOBILE. In: Rosalen, D. L., Zerbato, C., Turco, J. E. P (Eds.), A importância da Engenharia Agrícola para a segurança alimentar, 1. Sociedade Brasileira de Engenharia Agrícola, pp. 1-10, 2017.

[14] BORGES, L. G., BAZZI, C. L., SOUZA, E. G., MAGALHÃES, P. S. G., MICHELON, G. K. Web software to create thematic maps for precision

agriculture. *Pesquisa agropecuária brasileira*, v. 55, 2020. https://doi.org/10.1590/S1678-3921.pab2020.v55.00735

[15] HACHISUCA, A. M. M.; SOUZA, E. G.; OLIVEIRA, W. K. M.; BAZZI, C. L.; DONATO, D. G.; MENDES, I. S.; ABDALA, M. C.; MERCANTE, E. AgDataBox-IoT - application development for agrometeorological stations in smart. *MethodsX*, v. 11, 102419, 2023.