

Aplicativo Móvel para Gestão de Operações em Áreas Agrícolas

Bruno Eduardo F. Cordeiro*, Jakeline da S. Andrade*, Claudio Leones Bazzi*, Kelyn Schenatto[†], Ricardo Sobjak[‡]

*Universidade Tecnológica Federal do Paraná (UTFPR)

Medianeira, Brasil

Email: brunocordeiro@alunos.utfpr.edu.br, jakelineandradeds@gmail.com, bazzi@utfpr.edu.br, kschenatto@utfpr.edu.br
ricardosobjak@gmail.com

Abstract—Agriculture is crucial for human life and global economic growth, but it faces challenges such as environmental impacts and the need to increase production for a growing population. This study developed a mobile application to improve agricultural management, focusing on process optimization. Figma was used for prototyping, and tools such as Android Studio, GitHub, Visual Studio Code, and Flutter were employed for development, applying Clean Architecture to ensure code quality. The results show that the application has great potential to enhance agricultural property management, offering an efficient and intuitive solution.

Keywords—Agriculture; Optimization; Productivity.

Resumo—A agricultura é crucial para a vida humana e o crescimento econômico global, mas enfrenta desafios como impactos ambientais e a necessidade de aumentar a produção para uma população crescente. Como auxílio a esse cenário, este estudo desenvolveu um aplicativo móvel para melhorar a gestão agrícola, focando na otimização de processos. Utilizou-se Figma para prototipagem e ferramentas como Android Studio, GitHub, Visual Studio Code e Flutter para o desenvolvimento, aplicando a Arquitetura Limpa para garantir a qualidade do código. Os resultados mostram que o aplicativo tem grande potencial para aprimorar a gestão de propriedades agrícolas, oferecendo uma solução eficiente e intuitiva.

Palavras-chave—Agricultura; Otimização; Produtividade.

I. INTRODUÇÃO

A agricultura, sendo uma das práticas mais antigas e essenciais da humanidade, desempenha um papel central na sustentação da vida humana e no impulso do crescimento econômico global [1]. Em países desenvolvidos, essa atividade pode contribuir com até 4% no PIB (Produto Interno Bruto), enquanto em nações em desenvolvimento, sua participação pode alcançar 25% [2].

Contudo, apesar de sua importância econômica, as práticas agrícolas nem sempre têm efeitos benéficos e podem provocar impactos negativos significativos sobre o meio ambiente físico, como a erosão do solo e a poluição da água e do ar [3]. Além disso, as pastagens e as terras cultiváveis já ocupam cerca de 50% das áreas habitáveis do planeta [2], e essa proporção

pode aumentar ainda mais, considerando a previsão de uma população global de 9,5 bilhões de pessoas até 2050 [4].

Nesse contexto, a criação de ferramentas para o auxílio na gestão de recursos agrícolas surge como uma solução promissora, integrando tecnologias para promover uma maior produção sem a necessidade de expandir a área de cultivo, reduzir o uso de defensivos agrícolas, otimizar o aproveitamento dos recursos disponíveis e diminuir os impactos ambientais [2], [5].

Diversos estudos investigam a aplicação de ferramentas de gestão agrícola, empregando diferentes metodologias. Um desses estudos é conduzido por [6], que busca avaliar o potencial de um sistema de gestão agrícola em aprimorar o desempenho administrativo de uma propriedade rural localizada em Castro, Paraná (PR). Para isso, foi empregada uma pesquisa qualitativa para avaliar o impacto do software. Os resultados demonstram que a adoção do sistema contribuiu para uma maior eficiência nos processos administrativos.

Em [7], é apresentado o desenvolvimento de um sistema de gerenciamento de benefícios agrícolas baseado na web, implementado para a Prefeitura Municipal de Marques de Souza. O principal objetivo do sistema é otimizar processos e cálculos que anteriormente eram realizados de forma manual. A estrutura do sistema inclui componentes como: login, cadastro de usuário, guia de localidades, guia de talão, guia de informações do talão e guia de cálculo de benefícios.

No estudo conduzido por [8], foi criado um sistema para o controle de pecuária, desenvolvido utilizando as tecnologias React.js e Node.js. A metodologia proposta visa criar uma interface intuitiva que facilita a utilização do software por usuários com diferentes níveis de familiaridade tecnológica. O estudo ressalta a importância desse sistema para a gestão eficaz dos recursos pecuários, evidenciando sua capacidade de evitar perdas significativas na produtividade e de aprimorar o controle operacional.

Os exemplos apresentados evidenciam que a adoção de ferramentas para a gestão de recursos agrícolas tem um impacto po-

sitivo significativo, auxiliando tanto para a organização quanto para a otimização dos recursos. Com o objetivo de contribuir para esta área de pesquisa, este estudo constitui a primeira fase no desenvolvimento de um aplicativo móvel destinado à gestão de recursos agrícolas. O aplicativo tem como foco o gerenciamento de áreas, safras, maquinários, funcionários, operações e demais aspectos relacionados.

II. MATERIAIS E MÉTODOS

As metodologias empregadas neste estudo para o desenvolvimento das interfaces do software de gerenciamento de dados agrícolas, voltado ao apoio na organização e otimização de recursos, estão detalhadas no fluxograma apresentado na Fig. 1.

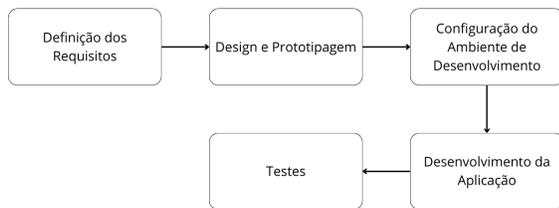


Fig. 1. Autoria Própria - Fluxograma de atividades

O primeiro passo consistiu na coleta de informações essenciais para o desenvolvimento do aplicativo, envolvendo a identificação das principais necessidades dos agricultores. Esse processo foi realizado tomando como referência a aplicação desenvolvida por [5].

No que diz respeito à etapa de design e prototipagem, utilizou-se a ferramenta Figma¹, um programa de desenvolvimento vetorial e design baseado em web, no qual oferece diversas vantagens, destacando-se por ser um software baseado em nuvem, com suporte tanto para web quanto para uso em desktop. Além disso, permite a colaboração em tempo real entre os membros da equipe e inclui a funcionalidade *Figma Mirror*, que possibilita visualizar os designs criados no desktop em vários dispositivos Android [9], [10].

Na etapa de configuração do ambiente de desenvolvimento, foram utilizadas as ferramentas Android Studio², GitHub³ e Visual Studio Code⁴. O Android Studio é um ambiente de desenvolvimento integrado (IDE) específico para a plataforma Android, disponível tanto em versões web quanto desktop, ambas com acesso ao Gemini, a inteligência artificial desenvolvida pela Google. Além disso, o Android Studio permite a criação de

layouts dinâmicos com o *Jetpack Compose* e oferece um editor de código inteligente, que possibilita a escrita de códigos mais limpos, promovendo maior fluidez e produtividade [11].

O GitHub é uma plataforma de hospedagem de código na nuvem que usa o Git para gerenciar projetos de software de forma colaborativa. Permite repositórios públicos ou privados e também funciona como uma "rede social" para desenvolvedores, facilitando o compartilhamento de dúvidas e a colaboração [12] [13].

Complementando essas ferramentas, o Visual Studio Code é um editor de código-fonte desenvolvido pela Microsoft, compatível com sistemas Windows, Linux e macOS. A escolha dessa ferramenta foi motivada por sua interface intuitiva, ampla compatibilidade com diversas linguagens de programação e pela disponibilidade de uma vasta gama de extensões e plugins que otimizam o fluxo de trabalho e aumentam a produtividade dos desenvolvedores.

Dentre as extensões oferecidas pelo Visual Studio Code, destaca-se o Flutter, um framework de desenvolvimento multiplataforma para dispositivos móveis criado pela Google, que foi utilizado na fase de desenvolvimento da aplicação. O Flutter é uma estrutura de código aberto, com a principal funcionalidade de criar aplicativos multiplataforma compilados nativamente, atendendo tanto à criação de aplicativos móveis quanto web, utilizando como linguagem de programação o Dart, que é reconhecido por sua acessibilidade, portabilidade e produtividade [14].

A arquitetura adotada para a estruturação do aplicativo foi a Arquitetura Limpa ou *Clean Architecture*, uma abordagem essencial para aumentar a produtividade e a qualidade do software. Essa arquitetura organiza o código em camadas distintas, que incluem Entidades, Casos de Uso, Interface de Usuário, Interfaces e Adaptadores, além de *Frameworks* e *Drivers* [15].

A Arquitetura Limpa oferece diversas vantagens, como a facilidade de manutenção do código, permitindo que itens que necessitam de reestruturação para garantir a funcionalidade sejam facilmente identificados; flexibilidade para mudanças, evitando a necessidade de modificar todo o código para implementar alterações; e a facilidade na realização de testes, devido ao isolamento da lógica de negócios [16].

Por fim, a validação do aplicativo foi realizada por meio de testes de campo, focando especificamente nas funcionalidades relacionadas ao GPS. Esses testes consistiram em usar o aplicativo em condições reais para garantir que as funções de rastreamento e localização estivessem operando corretamente no ambiente em que seriam utilizadas.

III. RESULTADOS

Inicialmente, tornou-se necessário estabelecer um sistema de cadastro de usuários e autenticação. Caso o usuário já esteja

¹<https://www.figma.com/pt-br/>

²<https://developer.android.com/studio?hl=pt-br>

³<https://github.com/>

⁴<https://code.visualstudio.com/>

registrado, ele poderá acessar diretamente a área de login, em que, ao inserir suas credenciais (Fig. 2a), poderá acessar o aplicativo (Fig. 2c). Caso contrário, será necessário criar uma conta, preenchendo os campos obrigatórios, como nome, e-mail, número de celular, data de nascimento e senha (Fig. 2b).

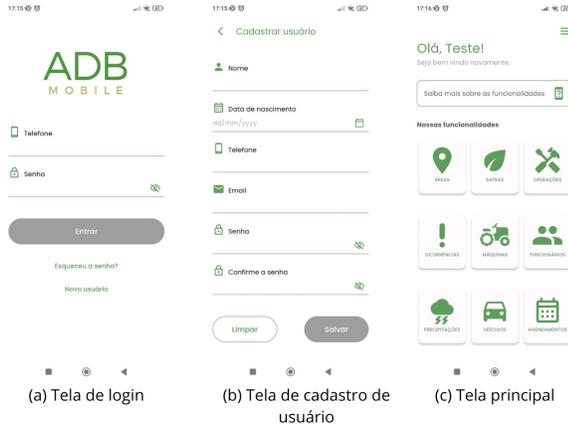


Fig. 2. Autoria Própria - Representação do processo de login, cadastro de usuários e tela principal

Entre as demais funcionalidades abordadas que compõem o aplicativo estão o gerenciamento de áreas, safras, operações, ocorrências, máquinas, funcionários, precipitações, veículos e agendamentos.

No gerenciamento de áreas, o produtor poderá realizar o cadastro de duas maneiras (Fig. 3a): utilizando a função *touch-screen*, que permite desenhar polígonos na tela para delimitar a área (Fig. 3b), ou utilizando o modo percurso, no qual o produtor caminha ao redor da área enquanto o aplicativo realiza a demarcação automaticamente (Fig. 3d). Nesta opção, é possível configurar o temporizador (Fig. 3c), que define o intervalo, em segundos, para a coleta de dados relativos à posição via GPS. Após a conclusão e o cadastro da área, o usuário pode visualizar, editar, excluir ou buscar uma área específica a qualquer momento.

No gerenciamento de safras, o usuário pode registrar o tipo de cultura, a data de início e fim da mesma. Já em operações, o usuário pode informar a área em que a operação será realizada, o tipo de operação, a safra associada e as datas de início e fim.

No que diz respeito às ocorrências, o produtor pode utilizar o mapa para demarcar a área em que foi identificada a ocorrência, que pode ser caracterizada, por exemplo, pelo surgimento de ervas daninhas ou pela falta de nutrientes em determinadas áreas. No gerenciamento de máquinas, é possível inserir informações como placa e ano de cada maquinário cadastrado, processo semelhante ao realizado no gerenciamento de veículos.

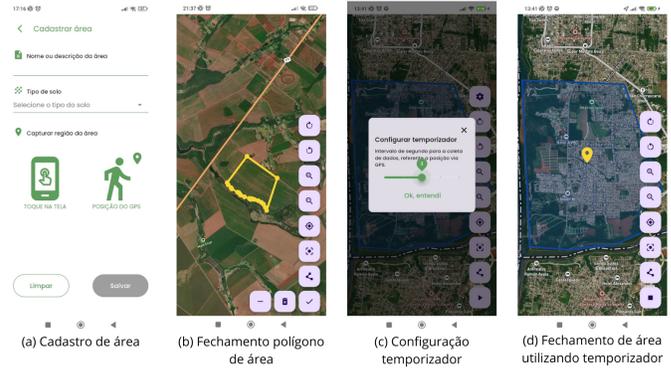


Fig. 3. Autoria Própria - Representação do processo de cadastro de área, fechamento polígono de área, configuração do temporizador e fechamento de área utilizando o temporizador

Para o gerenciamento de funcionários, é possível registrar o nome de cada colaborador. Em precipitações, podem ser armazenados dados como a quantidade de milímetros de chuva, a área em que a precipitação ocorreu, a estação e as datas de início e fim.

No gerenciamento de agendamentos, é possível programar operações detalhadas para uma área previamente cadastrada, permitindo a associação da safra correspondente, a definição da data de início das atividades, bem como a inclusão de observações pertinentes.

Além das funcionalidades mencionadas, o menu do aplicativo permite o cadastro prévio de itens como cultura, tipo de solo, tipo de operação e tipo de ocorrência. Esses registros facilitam o cadastro dos itens principais do aplicativo e auxiliam na organização, tornando o processo mais ágil e eficiente (Fig. 4).

Para auxiliar os usuários, foi desenvolvido um tutorial que explica todas as funcionalidades disponíveis no aplicativo. Um exemplo do funcionamento desse tutorial pode ser visto na Fig. 5.



Fig. 4. Autoria Própria - Menu com a opção de cadastro prévio de itens como cultura, tipo de solo, tipo de operação e tipo de ocorrência

IV. CONCLUSÃO

Embora o aplicativo ainda não esteja disponível em plataformas como a AppStore, os resultados e testes realizados indicam



Fig. 5. Autoria Própria - Tutorial para o auxílio de usuários

que ele possui grande potencial para se tornar uma ferramenta essencial para produtores que buscam maior controle e eficiência na gestão de suas propriedades. Além disso, o sistema desenvolvido oferece uma solução robusta e intuitiva para o gerenciamento agrícola, atendendo de forma eficaz às principais necessidades do setor.

Dessa forma, o sistema não apenas cumpre os objetivos propostos, mas também demonstra um potencial significativo para aprimorar a gestão e otimização de processos agrícolas, proporcionando uma experiência do usuário mais integrada e produtiva.

AGRADECIMENTOS

Agradeço à Universidade Tecnológica Federal do Paraná - UTFPR e o apoio contínuo de professores e colegas que contribuíram significativamente para a realização deste trabalho.

REFERÊNCIAS

- [1] G. Sunkad, "The importance of agriculture in present world," *Research-Gate*, 2020, Shri Kalidas Degree College Badami.
- [2] Grupo Banco Mundial, "Agricultura e alimentação," 2024, acesso em agosto de 2024. [Online]. Available: <https://www.worldbank.org/en/topic/agriculture/overview>
- [3] S. Waithaka, "Effects of agriculture on the environment," *International Journal of Agriculture*, vol. 8, no. 1, pp. 10–20, 2023, acesso em agosto de 2024.
- [4] Nações Unidas Brasil, "População mundial deve chegar a 9,7 bilhões de pessoas em 2050, diz relatório da onu," 2023, acesso em agosto de 2024. [Online]. Available: <https://brasil.un.org/pt-br/83427-populacao-mundial-deve-chegar-97-bilhoes-de-pessoas-em-2050-diz-relatorio-da-onu>
- [5] C. L. Bazzi, E. P. Jasse, P. S. G. Magalhães, E. G. de Souza, K. Schenatto, and R. Sobjak, "Agdatabox api – integração de dados e software em agricultura de precisão," *SoftwareX*, vol. 10, p. 100327, July 2019, acesso aberto, disponível em Português. [Online]. Available: <https://doi.org/10.1016/j.softx.2019.100327>
- [6] L. Almeida, A. Baptista, and C. K. Junior, "Software de gestão agrícola: A agilidade do escritório, também no campo," in *IX Congresso Brasileiro de Engenharia de Produção*. Associação Brasileira de Engenharia de Produção, December 2019, 4 a 6 de dezembro de 2019.
- [7] G. Dadall, "Sistema de gerenciamento de benefício agrícola para produtores rurais do município de marques de souza." 2012, orientador: Leandro Krug Wives. [Online]. Available: <http://hdl.handle.net/10183/54140>
- [8] W. Antônio dos Santos, J. Eduardo Ribeiro, and F. Florian, "Desenvolvimento de um software para gestão agropecuária," *RECIMA21 - Revista Científica Multidisciplinar*, vol. 4, no. 1, p. e414696, 2023. [Online]. Available: <https://doi.org/10.47820/recima21.v4i12.4696>
- [9] A. R. Isnain, Q. J. Adrian, and A. D. Putra, "Digital printing training for design at students of smk budi karya natar," *Journal of Engineering and Information Technology for Community Service (JEIT-CS)*, vol. 1, no. 3, pp. 137–141, 2023, received 22-December-2022, Accepted 26-December-2022, Published 1-January-2023. [Online]. Available: <http://jurnal.teknokrat.ac.id/index.php/JEIT-CS>
- [10] B. F. Howell, A. L. Hemming, G. Kilbourn-Barber, and S. Y. Christensen, "Explorando o impacto dos métodos de apresentação lineares e não lineares em um curso de história do design," in *Anais da Conferência Internacional sobre Educação em Engenharia e Design de Produto (E&PDE 2023)*, ser. E&PDE, L. Buck, H. Grierson, and E. Bohemia, Eds., Estados Unidos da América, 2023.
- [11] J. F. DiMarzio, *Beginning Android Programming with Android Studio*. Indianapolis, IN, USA: John Wiley & Sons, Inc., 2017, published simultaneously in Canada.
- [12] S. Peng, E. Kalliamvakou, P. Cihon, and M. Demirer, "The impact of ai on developer productivity: Evidence from github copilot," *arXiv preprint arXiv:2302.06590*, 2023, available at <https://arxiv.org/abs/2302.06590>.
- [13] A. d. C. R. Samapaio, "Criação de um guia de teste de software a partir da análise de repositórios do github," 2021, orientadora: Prof. Dra. Jacilane de Holanda Rabelo.
- [14] J. R. d. S. Faria, "Extensão para auxiliar na acessibilidade para aplicativos feitos em flutter utilizando vscode," 2023, orientador: Prof. Henrique Rebêlo.
- [15] A. C. Beltrão, F. d. A. Farzat, and G. H. Travassos, "Technical debt: A clean architecture implementation," in *Anais Estendidos do XI Congresso Brasileiro de Software: Teoria e Prática (CBSOFT 2020)*. Rio de Janeiro, RJ, Brazil: PESC – COPPE – Universidade Federal do Rio de Janeiro (UFRJ), 2020.
- [16] C. E. d. O. Bueno, "Desenvolvimento de um aplicativo utilizando o framework flutter e clean architecture," 2021, orientador: Rafael Leal Martins.