Desenvolvimento de um aplicativo para planejamento da colheita mecanizada de arroz irrigado

Rogério R. de Vargas[1], Alexandre Russini[1], Luis D. N. Martins[1], Daniel Ciro de Souza[1], Cezar V. L. Halim[1]

[1]Laboratório de Sistemas Inteligentes e Modelagem (LabSIM)
Universidade Federal do Pampa (Unipampa)
CEP 97650-000 – Itaqui – RS – Brasil

rogeriovargas@unipampa.edu.br, alexandrerussini@unipampa.edu.br, luisdavidmartins@outlook.com, danielciro6@gmail.com, lagohalim1@gmail.com

Resumo. *As máquinas agrícolas representam praticamente a metade do custo de produção, sendo a operação de colheita a que mais impacta nos custos, respondendo por aproximadamente 11% do custo total. Neste sentido, o presente trabalho tem por objetivo apresentar o desenvolvimento de um aplicativo denominado ColheArroz, destinado ao planejamento e estimativa do custo horário na colheita mecanizada de arroz irrigado. O aplicativo foi desenvolvido utilizando o framework Flutter, uma ferramenta open source para criação de aplicativos híbridos com a linguagem de programação Dart. Portanto, o aplicativo se mostrou uma importante ferramenta de auxílio aos produtores referente a tomada de decisão de colheita do arroz irrigado.*

Abstract. *Agricultural machines account for almost half of the production cost, with the harvesting operation having the biggest impact on costs, accounting for approximately 11% of the total cost. In this sense, the present work aims to present the development of an application called ColheArroz, intended for planning and estimating the hourly cost of mechanized harvesting of irrigated rice. The application was developed using the Flutter framework, an open source tool for creating hybrid applications with the Dart programming language. Therefore, the application proved to be an important aid tool for farmers regarding irrigated rice harvest decision making.*

1. Introdução

A colheita do arroz irrigado é uma das etapas mais importantes do processo de produção e quando é conduzida de forma inadequada pode ocasionar a perda de grãos e comprometer os esforços e investimentos dedicados à cultura (SILVA, 2015). A atividade sofre com o aumento de custo de produção a cada ano, bem como a estagnação do preço do produto, o que leva alguns produtores a abandonar a cultura ou buscar novas alternativas tecnológicas para otimizar o processo de produção (RUSSINI, 2014).

O desenvolvimento de novas tecnologias tem auxiliado produtores no manejo de suas áreas, visando a otimização de recursos, redução dos custos de produção e posteriormente no aumento do rendimento da cultura. Desenvolver e adotar essas novas tecnologias pode auxiliar na prevenção e na identificação de problemas que ajudarão a tomada de decisão (BARBIERI et al., 2015). Na busca pela otimização do gerenciamento na coleta de amostras de trabalho na Agricultura de Precisão, Leonan (2017) desenvolveu

um aplicativo para armazenamento dos dados levantados em campo, facilitando a organização da grande quantidade de dados e minimizando possíveis perdas. Ainda, essa aplicação aumenta a segurança dos dados e reduz o tempo gasto nas etapas de coletas executadas em campo.

As inovações tecnológicas adotadas na área agrícola caracterizadas pelo uso dos aplicativos móveis surgem como uma inovação revolucionária para o desenvolvimento rural. Os aplicativos agrícolas demonstram ser uma ferramenta eficaz quanto ao uso para soluções tecnológicas e inovadoras em todos os processos de produção. Atualmente estão sendo utilizados aplicativos, por exemplo, para identificação e monitoramento de pragas das culturas, interpretação de análises de solo, recomendação de irrigação baseado nos parâmetros atmosféricos e da cultura, tecnologia de aplicação de produtos fitossanitários, sensoriamento remoto, dentre outros. O desenvolvimento de uma aplicação móvel para estas tecnologias envolve um longo processo de criação, indo desde sua projeção até sua versão final para distribuição aos usuários. Atualmente existem diversas plataformas que permitem o desenvolvimento de aplicativos móveis e com isso, conhecimentos da área agrícola estão cada vez mais sendo aplicadas.

Desta forma, visando criar tecnologias para auxiliar o produtor e demais profissionais no acompanhamento e tomada de decisão, este trabalho tem por objetivo o desenvolvimento de um aplicativo híbrido para dispositivos móveis para auxílio no planejamento da colheita mecanizada de arroz irrigado.

2. Referencial Teórico e Trabalhos Relacionados

A multiplicidade de plataformas e linguagens de programação para a criação de aplicativos móveis está cada vez maior e por consequência disto os desenvolvedores precisam estar constantemente atualizados quanto às atualizações e novas linguagens de desenvolvimento. Atualmente, os sistemas operacionais mais usuais são o Android usando a linguagem de programação *Kotlin* e *Java* e para iOS tem-se o *Swift* e *Object-C* conforme Manning (2018).

A linguagem de programação *Kotlin* é uma linguagem estaticamente tipada, ou seja, utiliza variáveis com tipos específicos, sendo desenvolvida pela *JetBrains*, mesma empresa desenvolvedora do Android Studio. Possui uma sintaxe muito simples e agradável que é compilada para executar na máquina virtual do *Java*, portanto proporciona interoperabilidade total com *Java* (GLAUBER, 2019).

Segundo Mendes (2009) a linguagem *Java* é considerada simples porque permite o desenvolvimento de sistemas em diferentes sistemas operacionais e arquitetura de *hardware*, sem que o programador tenha que se preocupar com detalhes de infraestrutura. A plataforma Android utiliza o *Java* para o desenvolvimento de aplicativos móveis usando a API do *Java* fornecida pelo Google I/O. Diferente da linguagem de programação *Java*, o *Javascript* é uma linguagem de *script* que permite implementar funcionalidades mais complexas em páginas *web* (PRESCOTT, 2016).

Uma alternativa para o desenvolvimento de aplicativos para o sistema operacional iOS é o *Objective-C*, que é uma linguagem derivada da linguagem C tradicional e da *Smalltalk*. Sua sintaxe que inclui pré-processamento, expressões, declaração e chamadas de funções foi herdada da linguagem C, já os aspectos de orientação à objetos foi desenvolvida para habilitar passagem no estilo *Smalltalk*. A empresa Apple Inc. em 2014 lançou a linguagem *Swift* com uma linguagem simples e moderna, sua maior característica

é a possibilidade de criar códigos em conjunto com o *Objective-C*, pois classes do *Swift* podem chamar código escrito em *Objective-C* normalmente e retém conceitos chave como protocolos, cláusulas e categorias, porém, frequentemente substituindo a sintaxe por versões limpas e permite aplicações desses conceitos em diversas outras estruturações da linguagem (LECHETA, 2016).

Um aplicativo nativo é desenvolvido exclusivamente para uma plataforma específica como por exemplo: *Java* desenvolvido no Andoid e *Swift* em iOS. Cada uma das plataformas apresenta suas ferramentas específicas e elementos de interface. O *webapp* diferente da aplicação nativa necessita de conexão com a internet para ser acessado e não possuem integração com o sistema operacional. O aplicativo híbrido é uma alternativa que mistura um *webapp* e um aplicativo nativo permitindo a construção de aplicativos para dispositivos móveis usando *Javascript, HTML5 e CSS3*. Assim, um único aplicativo pode ser executado tanto em Android como para iOS. Silva et al. (2015) descreve os tipos de linguagens para dispositivos móveis e suas vantagens e desvantagens conforme Quadro 1.

Quadro 1: Tipos de aplicativos móveis. Fonte: Silva el al. (2015).

	Principais Vantagens	**Principais Desvantagens**
WebApps	Executados pelo *browser*, proporcionando; Atualização e distribuição rápida e abrangente, não precisam ser baixados ou atualizados; Acesso rápido e fácil, os usuários têm acesso imediato pelo *Smartphone*.	Pouca ou quase nenhuma integração com o *hardware* do dispositivo em que está sendo executado; mais lentos, dependendo da conexão com a internet; Interação entre o usuário e o aplicativo menos rica em funcionalidades.
Nativo	Interação entre o usuário e o aplicativo mais rica em funcionalidades e recursos; Velocidade na execução. Independente da internet.	Uma nova aplicação escrita para cada plataforma diferente; Distribuição e atualização dependentes de lojas *online*; (Apple Store, Goople Play).
Híbrido	Compartilhamento de boa parte do código entre plataformas; Possibilidade do uso de recursos da plataforma com código nativo; pode ser distribuído em lojas *online*; (Appe Store, Google Play)	Performance. Limitação de design.

Como exemplos de linguagens híbridas têm-se o *Ionic e React Native*. O *Ionic* é um *framework open source* para o desenvolvimento de aplicativos móveis multiplataforma que possibilita a implementação do aplicativo utilizando tecnologias empregadas na construção do *front-end* para soluções *web* em *HTML, CSS* e *Javasctript*. O *React Native* usa os mesmos blocos de construção da interface do usuário fundamentais do Android e iOS comuns, em vez de usar *Swift*, *Kotlin*, ou *Java*, os blocos de construção são colocados usando *Javascript* e *React.* Traz um conceito de componentes que possibilita modular a interface, criando comportamentos e atributos de cada elemento de acordo com cada plataforma utilizada.

A empresa Google I/O em 2018 disponibilizou um *framework* para o desenvolvimento de aplicativos nativos para dispositivos móveis, *web* e *desktop* a partir de uma única base de código chamado *Flutter*. O *Flutter* é um *framework* que facilita o desenvolvimento em multiplataforma utilizando a linguagem *Dart* e utiliza *widgets* para incorporar todas as diferenças críticas de plataforma, como rolagem, navegação, ícones e fontes, fornecendo um desempenho nativo completo no iOS e no Android. O *Flutter* inovou ao trabalhar exclusivamente com *Widgets* que facilitam o desenvolvimento do

aplicativo, deixando assim mais estruturado e com um rápido desempenho. O maior atrativo dele é o *Hot Reload* dita como atualização em poucos segundos, com isso o desenvolvedor consegue visualizar seu aplicativo a cada mudança no código facilmente, podendo evitar bugs, erros estruturais ou erros de compilação.

3. Metodologia

O aplicativo denominado ColheArroz foi criado utilizando o ambiente de desenvolvimento *VScode* disponibilizado pela *Microsoft*. Para o desenvolvimento em multiplataforma utilizou-se o *framework Flutter* com a linguagem de programação *Dart.* Através do *Flutter* o desenvolvimento surge dos *widgets* que são as estruturas do aplicativo, tornando assim mais prático o seu desenvolvimento. O aplicativo é uma relação direta entre usuário e aplicativo onde este usuário insere dados e o aplicativo calcula e apresenta em tela os resultados obtidos. Utilizou-se uma base de dados referentes ao desempenho de colhedoras em quatro safras consecutivas abrangendo cultivares de arroz e condições de colheitas diferentes.

A base de dados é utilizada para o desenvolvimento de modelos matemáticos que relacionam as perdas de grãos com a velocidade de deslocamento da colhedora, plataforma de corte e sistema de trilha. Os modelos de estimativas e perdas são baseados nas principais cultivares de arroz irrigado cultivadas na Fronteira Oeste do Rio Grande do Sul (Martins, 2018). A representação estrutural da relação entre usuário e aplicativo está apresentada na Figura 1.

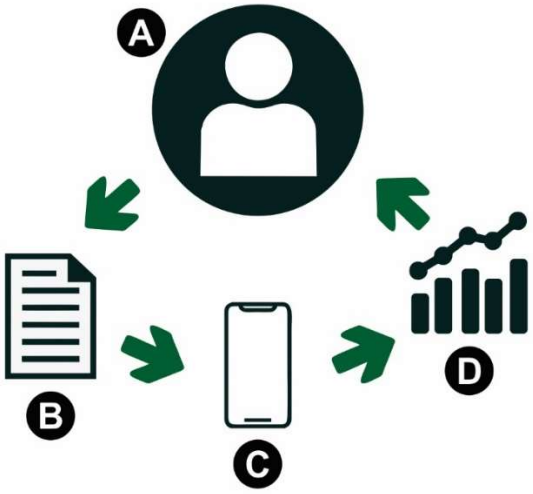

Figura 1: Relação usuário e aplicativo.

Fonte: Os autores.

- **A. Usuário:** Responsável pela inserção das informações no aplicativo, geralmente técnicos e responsáveis da área.
- **B. Dados:** Informações obtidas em campo inseridas pelo usuário no aplicativo.
- **C. Aplicativo móvel:** Aplicação móvel desenvolvida em multiplataforma para calcular e gerar relatórios em tela a partir dos dados informados pelo usuário.
- **D. Resultados:** Relatório gerado pelo aplicativo com resultados detalhados.

O aplicativo está dividido em duas funcionalidades: Planejamento e Custo Horário. No planejamento, o usuário pode fazer estimativas das perdas de grãos durante

a colheita, levando em consideração parâmetros técnicos desta operação, além de informar a capacidade operacional e o tempo estimado para concluir a operação, bem como determinar a largura da plataforma necessárias para o atendimento do planejamento.

Em relação ao Custo Horário, o usuário pode fazer um comparativo e análise de diferentes máquinas agrícolas, visualizando informações de depreciação de cada máquina e seus diversos custos, assim o usuário pode identificar a área mínima que cada máquina com suas respectivas especificações pode apresentar para a viabilizar sua aquisição.

4. Resultados

O usuário ao abrir o aplicativo se depara com uma interface simples e intuitiva. A tela inicial possui um menu com informações gerais da aplicação e desenvolvedores. Na área inferior da tela se encontram três botões: Botão *Home*, Planejamento e Custo horário, assim o usuário pode se movimentar pelas telas de acordo com a funcionalidade desejada.

A navegação de telas foi desenvolvida através de um *BottomNavigationBar* que torna a transição entre telas mais simples facilitando o manejo dos dados. Essa navegação vem de plugins que o *Framework* disponibiliza como uma extensão de seus arquivos aberto ao público, mas sempre desenvolvida na linguagem *Dart.* A Figura 2 apresenta a tela principal da aplicação.

Figura 2: Tela inicial do aplicativo.

Fonte: Os autores.

Para evitar que o usuário insira informações erradas e em formatos diferentes da realidade implementou-se uma validação em cada campo que indica ao usuário a forma correta para inserir os valores e se há campos em branco. As telas com campos para inserção dos dados referentes ao planejamento e de custo horário estão apresentadas nas Figuras 3 e 4 respectivamente.

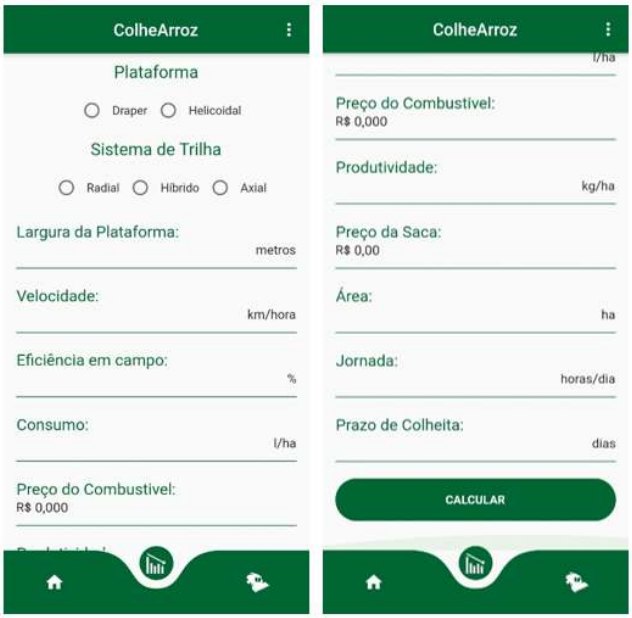

Figura 3: Telas para inserção de dados realizar as estimativas.

Fonte: Os autores.

Figura 4: Telas para inserção dados referentes ao custo horário.

Fonte: Os autores

Os resultados das estimativas podem ser visualizados em três etapas: Dados Colhedora, Estimativas e Perdas. A etapa Dados Colhedora mostra resultados da capacidade operacional, custos com combustível e consumo. Por outro lado, Estimativas mostram os resultados estimados das colhedoras e na colheita, além disso, a terceira etapa Perdas mostra resultados de perdas na plataforma, trilha e prejuízos, conforme apresentado na Figura 5. Os resultados para a funcionalidade de Custo Horário estão mostrados na Figura 6.

Figura 5: Relatório de resultados da estimativa.

Fonte: Os autores.

Figura 6: Relatório de resultados do custo horário.

Fonte: Os autores.

5. Conclusão

O aplicativo surge como uma ferramenta para auxiliar os produtores e profissionais do setor orizícola, apresentando para os usuários informações que podem ser utilizadas na tomada de decisão buscando a redução das perdas na colheita e planejamento, visando o correto dimensionamento resultando desta forma numa maior otimização do uso das máquinas na operação de colheita mecanizada de arroz irrigado.

O usuário também pode fazer análise da viabilidade na compra de uma colhedora através da área mínima que viabilize sua aquisição. A aplicação de modo geral permite uma interação entre o usuário e o aplicativo entregando uma interface de fácil manuseio onde o usuário insere os dados para a base de cálculos e os resultados são apresentados em um relatório em tela. O aplicativo está em fase final de testes e futuramente será disponibilizado para as plataformas Android e iOS.

Referências

Manning, J., Buttfield-Addison, P., Nugent, T. Learning Swift: Building Apps for MacOS, IOS and Beyond. 3ª edição. In: O'Reilly Media, Inc. 2018, Sebastopol. 378 p.

Barbieri, L. W. et al. O comportamento do custo do arroz no comparativo entre dois métodos de cultivo distintos. In: Congresso de Contabilidade. 2015. Disponível em: http://dvl.ccn.ufsc.br/congresso_internacional/anais/6CCF/65_15.pdf. Acesso em: 20 de abril de 2019.

Leonan, F. et al. Aplicativo para gerenciamento de coleta de amostras em agricultura de precisão. In: Congresso Brasileiro de Engenharia Agrícola – CONBEA. 2017, Maceió, AL. p 4.

Martins, L. D. N. et al. Ajuda Importante. In: Cultivar Máquinas. 2018. v. 181, p. 33-35.

Russini. A. et al. Velocidade Certa. In: Cultivar Máquinas. 2014. v. 148, p. 26-29.

Silva, J.G. da., Custódio, D. P. C. Arroz do plantio à colheita. In: UFV. 2015, Viçosa, MG. p.220-242.

Glauber, N. Dominando o Andoid com Kotlin. In: Novatec Editora. 2019, São Paulo. 1064 p.

Mendes, D. R. Programação Java com Ênfase em Orientação a Objetos. In: Novatec Editora. 2009, São Paulo. 456 p.

Prescott, P. Programando em JavaScript. In: Babelcube Inc. 2016. 35 p.

Silva, L. L. B. Pires, D. F. Neto, S. C. Desenvolvimento de aplicações para dispositivos móveis: Tipos e exemplo de aplicação na plataforma iOS. II Workshop de Iniciação Científica em Sistemas de Informação. Goiás: Goiânia, 2015. 4 p.

Lecheta, R. Desenvolvendo para iPhone e iPad. In. Novatec Editora. 2016. 640 p.