



SEI-SICITE 2021

Pesquisa e Extensão para um mundo em transformação

Desenvolvimento do Módulo de Registro de Pulverizações do **manejo.mobile**

Development of the Pulverization Registration Module for manejo.mobile

Bruno Oliveira Galvão de Arruda*, **Gabriel Costa Silva†**

RESUMO

Este trabalho é parte do projeto de inovação em parceria com o Instituto de Desenvolvimento Rural do Paraná (IDR) e Serviço Nacional de Aprendizagem Rural/PR (SENAR/PR). Esse projeto visa aumentar a eficiência na coleta e análise de dados da tecnologia de Manejo Integrado de Pragas (MIP) da cultura da soja. O MIP é uma tecnologia de procedimentos ambientais e econômicos para reduzir o uso de inseticidas por meio do controle efetivo de pragas. Dado o limitado acesso à internet em regiões rurais, existe a necessidade de uma solução que permita registrar, de forma offline, as pulverizações realizadas durante a execução do MIP. O objetivo deste trabalho é desenvolver o módulo de registro de pulverizações, que será integrado ao aplicativo manejo.mobile, e disponibilizado aos técnicos do IDR e instrutores do SENAR. Para o desenvolvimento do presente trabalho, foram usadas técnicas tradicionais da Engenharia de Software, como gerenciamento de versão, teste de software e prototipação. O desenvolvimento foi realizado em quatro fases: (i) análise de tecnologia para desenvolvimento do aplicativo móvel; (ii) identificação das funcionalidades da aplicação; (iii) desenvolvimento do protótipo de interface gráfica; e (iv) evolução e disponibilização do protótipo para testes com usuários. Os resultados deste trabalho mostram a viabilidade do protótipo desenvolvido, contribuindo com uma parte importante do aplicativo manejo.mobile.

Palavras-chave: Smartphones - Programação, Interface de programas aplicativos (Software), Computação móvel, JavaScript (Linguagem de programação de computador).

ABSTRACT

The work is part of the innovation project in partnership with the Instituto de Desenvolvimento Rural do Paraná (IDR) and the Serviço Nacional de Aprendizagem Rural/PR (SENAR/PR). This project aims to increase the efficiency of data collection and analysis of Manejo Integrado de Pragas (MIP) technology for soybean crops. MIP is a technology of environmental and economical procedures to reduce the use of insecticides through effective pest control. Given the limited access to the internet in rural regions, there is a need for a solution that allows recording, offline, the sprayings carried out during the execution of the MIP. The objective of this work is to develop the spraying registration module, which will be integrated with the manejo.mobile application, and made available to IDR technicians and SENAR instructors. For the development of this work, traditional techniques of Software Engineering were used, such as version management, software testing and prototyping. The development was carried out in four phases: (i) technology analysis for mobile application development; (ii) identification of application functionalities; (iii) development of the graphical interface prototype; and (iv) evolution and availability of the prototype for testing with users. The results of this work show the viability of the developed prototype, contributing to an important part of the manejo.mobile application.

Keywords: Smartphones - Programming, Application Program Interface (Software), Mobile Computing, JavaScript (Computer Programming Language).

1 INTRODUÇÃO

* Engenharia de Computação, Universidade Tecnológica Federal do Paraná, Cornélio Procópio, Paraná, Brasil; bruno.ogarruda@gmail.com

† Universidade Tecnológica Federal do Paraná, Campus Cornélio Procópio; gabrielcosta@utfpr.edu.br



O Manejo Integrado de Pragas (MIP) é uma tecnologia de procedimentos ambientais e econômicos para reduzir o uso de inseticidas por meio do controle efetivo de pragas. O uso correto de inseticidas nas lavouras evita a vulnerabilidade do avanço rápido das pragas e o surgimento de novas pragas. Por outro lado, o uso exagerado de inseticidas afeta o controle dos agentes biológicos naturais. O IDR-Paraná, SENAR, Embrapa Soja e os sojicultores paranaenses têm aplicado o MIP com sucesso nas lavouras de soja nos últimos anos. Por exemplo, na safra 2019/2020, o resultado obtido foi de 1,65 aplicações de inseticidas durante o ciclo da cultura nas propriedades que aplicaram o MIP, enquanto nas áreas que não aplicaram o MIP o uso foi de 3,02 (CONTE et al., 2020). Essa redução no uso de inseticidas não só reduz o custo de produção como também colabora significativamente com a preservação do meio ambiente.

Atualmente, o MIP é auxiliado pelo aplicativo *Web manejo.app*. Os técnicos rurais registram na aplicação os dados coletados em campo. Tais dados são validados no momento do registro. Em seguida os dados podem ser visualizados e analisados de maneira imediata (SILVA, 2020).

Porém, o aplicativo *Web* existente depende de conexão constante com a Internet para o lançamento das ocorrências de pragas nas lavouras. Como a disponibilidade de Internet no campo ainda é incomum, o técnico do IDR-Paraná precisa anotar em papel as ocorrências e fazer o lançamento posteriormente no aplicativo. Isso causa eventuais dificuldades, como a necessidade de fazer a mesma anotação duas vezes: uma vez em papel, e outra no aplicativo.

Dessa maneira, existe uma necessidade de uma aplicação móvel que não necessita da Internet para lançar os dados de ocorrências de pragas e/ou pulverizações em propriedades rurais. Esse projeto se concentra em desenvolver o módulo de registro de pulverizações, que deverá integrar o aplicativo para dispositivos móveis *manejo.mobile*.

2 MÉTODO (OU PROCEDIMENTOS OPERACIONAIS DA PESQUISA)

O projeto usou técnicas tradicionais da Engenharia de Software nas suas quatro etapas de desenvolvimento. A primeira etapa, análise de tecnologia para desenvolvimento. Durante essa etapa, as tecnologias Ionic Vue e React Native foram investigadas. Também, dois protótipos foram desenvolvidos para permitir a comparação das tecnologias. A segunda etapa, identificação das funcionalidades, analisou a funcionalidade de pulverização atualmente presente no *manejo.app*, com o objetivo de identificar como adaptar a funcionalidade para o *manejo.mobile*. A terceira etapa, desenvolvimento de protótipo de interface gráfica, desenvolveu uma versão da interface gráfica para o registro de pulverizações do *manejo.mobile*. Ainda nessa etapa, uma pesquisa com os usuários foi realizada para verificar a viabilidade da interface gráfica projetada. A última etapa, evolução do protótipo, desenvolveu as telas de registro de pulverizações do *manejo.mobile*, além dos testes unitários e de aceitação.

3 RESULTADOS

3.1 Análise de tecnologia para desenvolvimento

Nesta etapa foi realizada uma comparação entre as tecnologias Ionic Vue e React Native para o desenvolvimento do aplicativo. O React Native é um framework para desenvolvimento de aplicações multiplataforma (Android e iOS). Criado pelo Facebook em 2015, atualizado até hoje pelo próprio Facebook e também com ajuda de pessoas e empresas pelo mundo todo (REACT NATIVE, 2021). Algumas das características do React Native são: (i) uma base de código para as plataformas Android e iOS, (ii) renderização

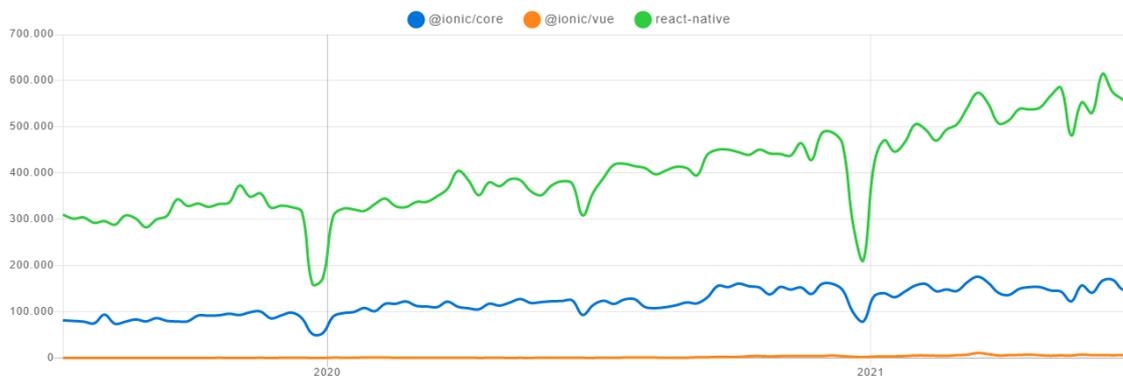


nativa para iOS e para Android, (iii) atualizações rápidas das telas após alterações na aplicação, (iv) grande comunidade e (v) atualizações constantes pelo Facebook e pela comunidade (JAVATPOINT, 2021b).

Criado por Max Lynch, Ben Sperry e Adam Bradley em 2013, o Ionic também é um framework para desenvolvimento de aplicações multiplataforma (Android e iOS) que pode ser utilizado junto com Angular, React ou Vue.js (WIKIPEDIA, 2021). Algumas das características do Ionic são: (i) uma base de código para as plataformas Android e iOS, (ii) baseado em estruturas padrões da Web confiáveis como HTML, CSS e JavaScript, (iii) possibilita construir aplicativos limpo, simples e funcional, (iv) diversos componentes prontos e (v) pode ser testado tanto no browser como em emuladores (JAVATPOINT, 2021a).

Comparando as duas tecnologias, podemos perceber uma preferência pelo React Native. Na Figura 1 podemos ver uma ampla vantagem do React Native sobre Ionic nos downloads com o NPM (Node Package Modules) nos últimos dois anos, e também sobre o Ionic com Vue.js no qual essa vantagem fica ainda mais evidente (POTTER, 2021).

Figura 1 - Número de downloads com NPM das tecnologias React Native, @ionic/core e @ionic/vue.



Fonte: Potter (2021)

A pesquisa feita pelo Raphaël Benitte e Sacha Greif no ano de 2020 apresenta que 68,2% das pessoas, com o total de 21.144 respostas, não conhecem, não estão interessadas ou não usariam novamente a tecnologia Ionic. Enquanto para o React Native, a pesquisa apresenta 31,2% das pessoas, com o total de 21.008 respostas, não conhecem, não estão interessadas ou não usariam novamente a tecnologia (BENITTE e GREIF, 2020). Isso mostra o tamanho da comunidade React Native que resulta em um vasto material disponibilizado pela comunidade na Internet, tanto para tirar dúvidas, como de pacotes e bibliotecas desenvolvidos e mantidos atualizados pela comunidade que podem ser utilizados nos projetos.

Assim, dois protótipos foram desenvolvidos. Ambos protótipos possuem a mesma aparência e funcionalidades. Os protótipos foram usados para comparar (i) a dificuldade de aprendizado, (ii) o tamanho do aplicativo gerado e (iii) o tempo para escrever o código.

A análise das tecnologias mostrou que o tamanho do aplicativo gerado e o tempo para escrever os códigos são parecidos independente da tecnologia utilizada. Contudo, o React Native mostrou mais facilidade de aprendizado, principalmente pelo vasto material disponibilizado pela comunidade na Internet. Dessa forma, o React Native foi a tecnologia escolhida para o desenvolvimento.

3.2 Identificação das funcionalidades



Durante esta segunda etapa, foi identificado as funcionalidades já existentes na aplicação Web e que deveriam estar presentes na aplicação móvel. Foi feito o mapeamento de todas as telas e campos com a ferramenta Xmind (XMIND, 2021). Mediante a análise das telas, foi estabelecido que estariam presentes no aplicativo móvel as seguintes telas: (i) telas de lançamento de dados MIP e de (ii) pulverização, e também das telas do (iii) painel de controle para análise dos dados lançados com a utilização de gráficos e tabelas. Estas que estas telas são as mais utilizadas durante as inspeções dos técnicos nas plantações. Os resultados são apresentados na forma de texto, de figuras, de quadros e de tabelas, respeitando a formatação detalhada a seguir.

3.3 Desenvolvimento de protótipo de interface

A terceira etapa consiste na prototipação da interface gráfica. Para isso, a ferramenta Figma foi utilizada (FIGMA, 2021). O desenvolvimento da interface gráfica foi dividido em três fases. Na primeira fase foi desenvolvido as telas do painel de controle que envolve os gráficos de linha e de coluna dos dados das lagartas, percevejo, inimigos naturais e desfolha, e também os dados tabulados do quadro geral MIP e as telas para coleta dos dados do lançamento MIP. Na fase dois foram desenvolvidas as telas para coleta dos dados do lançamento de pulverização. Na terceira fase foram feitos testes com usuários. Essa seção tem foco nas telas do painel de controle e pulverização.

Na fase 1 houve a realização de duas versões. A principal mudança entre elas foi nos gráficos do painel de controle, tanto o de barra quanto o de linha, deixando mais simples a navegação e visualização das informações.

Na fase 2, foram desenvolvidas as telas para coleta dos dados do lançamento de pulverização. O desafio dessa fase foi transformar um formulário padrão que estava na aplicação Web em um formulário mais fácil e interativo para usar em uma aplicação mobile que será acessada em plantações, em propriedades rurais. Nessa fase foram criadas as seguintes telas: (i) lançamento da operação, (ii) seleção de classe do produto, (iii) seleção de alvo ou função do produto, (iv) seleção do produto, (v) informações do produto, e (vi) revisão.

A Figura 2 mostra as telas de lançamento da operação, de seleção de classe do produto, informações do produto, e revisão.

Figura 2 - Telas de lançamento da operação, de selecionar uma classe do produto, de informações do produto e a de revisão.



Fonte: Autoria própria (2021).



Na terceira fase, foi enviado uma representação da interface gráfica da aplicação para os usuários atuais da aplicação Web. Eles avaliaram o protótipo e responderam com as suas avaliações e opiniões sobre o resultado final. O retorno dos usuários foi positivo. Assim, pode-se dar continuidade no desenvolvimento do protótipo.

3.4 Evolução do protótipo

Inicialmente, foi criado um projeto em React Native com Typescript, para manter uma padronização do código, foram utilizados os frameworks eslint e prettier. Foram criados os componentes buttonIcon, buttonText, calendarContainer, card, header, input, mip, pickerContainer, pulverization e searchInput. Os dois principais e mais complexos componentes são o mip e o pulverization. Esta seção foca no componente pulverization, que implementa a coleta dos dados de pulverização.

O componente pulverization é composto pelos seguintes componentes: formPulverization, insertData, operationData, revision, selectClass, selectProduct e selectTarget. O formPulverization é responsável pela navegação entre todos componentes e também responsável pelo armazenamento das informações preenchidas pelo usuário, quando o usuário pressiona o botão Salvar, na tela de revisão.

A Figura 3 mostra o código que implementa a navegação entre as páginas. A implementação consiste de uma estrutura condicional (switch). Cada case do switch executa um componente que atualiza a tela do aplicativo, na seguinte ordem: lançamento da operação, seleção de classe do produto, seleção de alvo ou função do produto, seleção do produto, informações do produto, e revisão.

Figura 3 - Código responsável pela navegação entre as telas no componente formPulverization.

```
101 let aux = 2;
102
103 switch (step) {
104   case 1:
105     if (type === 'update') aux = 6;
106     else aux = 2;
107     return (
108       <OperationData
109         type={type}
110         nextStep={() => setStep(aux)}
111         pulverization={pulverization}
112         onChangeHandlerPulverization={onChangeHandlerPulverization}
113       />
114     );
115   case 2:
116     if (type === 'new') aux = 6;
117     else aux = 1;
118     return (
119       <SelectClass
120         type={type}
121         nextStep={() => setStep(3)}
122         prevStep={() => setStep(aux)}
123         product={product}
124         onChangeHandlerProduct={onChangeHandlerProduct}
125       />
126     );
127   case 3:
128     return (
129       <SelectTarget
130         nextStep={() => setStep(4)}
131         prevStep={() => setStep(2)}
132         product={product}
133         onChangeHandlerProduct={onChangeHandlerProduct}
134       />
135     );
136   case 4:
137     if (product.classe === 'Acaricida') aux = 2;
138     else aux = 3;
139     return (
140       <SelectProduct
141         nextStep={() => setStep(5)}
142         prevStep={() => setStep(aux)}
143         product={product}
144         onChangeHandlerProduct={onChangeHandlerProduct}
145       />
146     );
147   case 5:
148     return (
149       <InsetData
150         prevStep={() => setStep(4)}
151         nextNo={nextNo}
152         nextYes={nextYes}
153         product={product}
154         onChangeHandlerProduct={onChangeHandlerProduct}
155       />
156     );
157   default:
158     return (
159       <Revision
160         edit={edit}
161         newProduct={newProduct}
162         pulverization={pulverization}
163         products={products}
164       />
165     );
166 }
```

Fonte: A autoria própria (2021).



Após a implementação das telas, foram realizados os testes unitários e os testes de interface. Para o componente formPulverization foram criados 6 testes unitários e 5 testes de interface.

4 CONCLUSÃO

O desenvolvimento do módulo com o React Native foi produtivo. Como o código do aplicativo tem como base o JS, é relativamente simples de utilizar, principalmente comparado com tecnologias nativas. Como possui uma grande comunidade, fica fácil de aprender e tirar dúvidas. O recurso de atualização rápida do React Native permite que, após alterações no código, a atualização no aplicativo seja instantânea, aumentando a produtividade no desenvolvimento.

O principal objetivo do presente trabalho foi desenvolver o módulo de registro de pulverizações da aplicação móvel manejo.mobile. Dessa forma, esse trabalho concluiu esse objetivo pois houve a escolha da tecnologia, a identificação das funcionalidades presentes na aplicação móvel, desenvolvimento da interface gráfica e desenvolvimento do módulo.

Como estudante, com o desenvolvimento do projeto, consegui aprender conceitos de design, conceitos avançados de programação mobile, conceitos de testes unitários e de interface. Além de ter obtido conhecimentos teóricos de como é feito o manejo integrado de pragas e suas vantagens.

REFERÊNCIAS

- BENITTE, Raphaël; GREIF, Sacha. **The State of JavaScript Survey**. 2020. Disponível em: <https://stateofjs.com/>. Acesso em: 06 jul. 2021.
- CONTE, Osmar; POSSAMAI, Edivan José; SILVA, Gabriel Costa; REIS, Eliana Aparecida; GOMES, Emerson Crivelaro; CORRÊA-FERREIRA, Beatriz Spalding; ROGGIA, Samuel; PRANDO, André Mateus. **Resultados do manejo integrado de pragas da soja na safra 2019/2020 no Paraná**. Londrina: Embrapa Soja, 2020.
- FIGMA. **Figma: the collaborative interface design tool**. 2021. Disponível em: <https://www.figma.com/>. Acesso em: 25 jun. 2021.
- JAVATPOINT. **Ionic Framework Features**. 2021a. Disponível em: <https://www.javatpoint.com/ionic-features>. Acesso em: 04 jul. 2021.
- JAVATPOINT. **React Native Tutorial**. 2021b. Disponível em: <https://reactnative-introductioncnative.dev/>. Acesso em: 05 jul. 2021.
- REACT NATIVE. **A framework for building native apps using React**. 2021. Disponível em: <https://reactnative.dev/>. Acesso em: 05 jul. 2021.
- POTTER, John. **Compare package download counts over time**. 2021. Disponível em: <https://www.npmtrends.com/>. Acesso em: 06 jul. 2021.
- SILVA, G. C. **Desenvolvimento de aplicativos para coleta e análise de dados da tecnologia de manejo integrado de pragas da cultura da soja**. 2020.
- WIKIPEDIA. **Ionic (mobile app framework)**. 2021. Disponível em: [https://en.wikipedia.org/wiki/Ionic_\(mobile_app_framework\)](https://en.wikipedia.org/wiki/Ionic_(mobile_app_framework)). Acesso em: 04 jul. 2021.
- XMIND. **Xmind - Mind Mapping Software**. 2021. Disponível em: <https://www.xmind.net/>. Acesso em: 07 jul. 2021.