



# Desenvolvimento do Módulo de Coleta de Ocorrências de Pragas do **manejo.mobile**

## *Development of the Pest Occurrence Collection Module for the manejo.mobile*

Bruna Quispe Oruchi\*, Gabriel Costa Silva†,

### RESUMO

O Instituto de Desenvolvimento Rural do Paraná (IDR-Paraná) e o Serviço Nacional de Aprendizagem Rural/PR (SENAR), parceiros no desenvolvimento do projeto de inovação, auxiliam na aplicação da tecnologia de Manejo Integrado de Pragas (MIP) em lavouras de soja. Essa tecnologia otimiza as tomadas de decisões para o controle de pragas, reduzindo o uso de agrotóxicos e os impactos ambientais nas lavouras de soja. Entretanto, para o acompanhamento efetivo do MIP, houve a necessidade de desenvolvimento de uma aplicação móvel que funcione *offline* - devido às restrições de acesso à Internet em áreas rurais. Este trabalho apresenta o desenvolvimento do módulo de coleta de ocorrências de pragas do aplicativo móvel *manejo.mobile*. Técnicas tradicionais de desenvolvimento de software foram usadas neste projeto, como teste de software, prototipação e gerenciamento de versão. Adicionalmente, quatro etapas foram realizadas: (i) a escolha da tecnologia para o desenvolvimento, (ii) análise das funcionalidades necessárias, (iii) prototipação da interface gráfica, e (iv) desenvolvimento do módulo proposto. O módulo desenvolvido foi avaliado de forma positiva pelos usuários finais, sugerindo a viabilidade do projeto.

**Palavras-chave:** Aplicativos móveis, Desenvolvimento de Software, JavaScript

### ABSTRACT

The Instituto de Desenvolvimento Rural do Paraná (IDR-Paraná) and the Serviço Nacional de Aprendizagem Rural/PR (SENAR), partners in the development of the innovation project, assist in the application of Integrated Pest Management (IPM) technology in soybean crops. This technology optimizes decision-making for pest control, reducing the use of pesticides and the environmental impacts on soy crops. However, for the effective monitoring of the IPM, there was a need to develop a mobile application that works offline - due to Internet access restrictions in rural areas. This work presents the development of the pest occurrence collection module for the mobile application *manejo.mobile*. Traditional software development techniques were used in this project, such as software testing, prototyping and version management. Additionally, four steps were performed: (i) choosing the technology for development, (ii) analyzing the necessary functionalities, (iii) prototyping the graphical interface, and (iv) developing the proposed module. The developed module was positively evaluated by the end users, suggesting the project's feasibility.

**Keywords:** Mobile apps, Software Development, JavaScript.

## 1 INTRODUÇÃO

O Manejo Integrado de Pragas (MIP) é uma técnica utilizada para o controle de pragas, a fim de otimizar as tomadas de decisões em relação ao uso de agrotóxicos. Utilizado em lavouras comerciais para o monitoramento semanal dos níveis populacionais das pragas de soja, seu propósito é verificar se há necessidade de aplicar agrotóxicos. O impacto causado nas safras por meio dessa tecnologia reduz os custos da safra pela

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

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



redução no uso de agrotóxicos, gerando mais lucro para o produtor e reduzindo os impactos ambientais nas lavouras de soja (SILVA, 2020).

O Instituto de Desenvolvimento Rural do Paraná -IAPAR-EMATER (IDR-Paraná) e o SENAR/PR aplicam o MIP em lavouras de soja. Apesar de proporcionar vantagens nas safras, haviam alguns empecilhos no processo da coleta e análise de dados. Os técnicos do IDR-Paraná vão até lavouras para realizar a coleta de dados, anotando as informações em papel e, posteriormente, registrando os dados em planilhas eletrônicas. A análise de dados é realizada posteriormente, por meio de dados inseridos na etapa de coleta. Porém, no caso de algum dado inconsistente, a consolidação e análise precisa ser interrompida para verificar a inconsistência. Isso pode atrasar a tomada de decisões da safra. A duração do processo de consolidação e análise podia chegar a durar seis meses, ocasionando prejuízo para as safras (SILVA, 2020).

Mediante a situação que o IDR-Paraná e o SENAR/PR se encontravam, surgiu a parceria com UTFPR para um projeto de inovação. A proposta do projeto é desenvolver o *manejo.app*, uma aplicação Web para apoiar a coleta e análise de dados do MIP. Deste modo, os técnicos podem inserir as informações na aplicação e visualizar em tempo real gráficos para compreensão dos dados, otimizando o tempo na análise dos dados da safra.

Apesar do *manejo.app* resolver o problema de consolidação e análise de dados do IDR-Paraná e o SENAR/PR, as coletas ainda são realizadas de modo manual, anotando em papel, e em seguida, inseridas na aplicação. Assim, para complementar o projeto de inovação, se fez necessário desenvolver uma versão para dispositivos móveis do *manejo.app*. A proposta é desenvolver uma aplicação móvel que funcione offline, devido às restrições de acesso a rede de Internet em áreas rurais, facilitando a coleta e análise de dados.

Esse projeto foca no desenvolvimento do módulo de coleta de ocorrência de pragas para o aplicativo móvel *manejo.mobile*. Inicialmente, foi necessário analisar as possíveis tecnologias que poderiam ser usadas, avaliando suas vantagens e desvantagens. Após, houve a necessidade de identificar as funcionalidades do *manejo.app*, com a finalidade de compreender a aplicação e os componentes presentes. Em seguida, para dar início ao desenvolvimento, foi realizado o protótipo de interface gráfica. E, por fim, o desenvolvimento da aplicação móvel.

## 2 MÉTODO (OU PROCEDIMENTOS OPERACIONAIS DA PESQUISA)

O *React Native* é um framework de código aberto para desenvolver aplicações móveis e foi criado pela equipe do Facebook em 2015, usando a licença do MIT (NATIVE, 2021). As principais características do *React Native* são: o conceito de multiplataforma, desenvolvimento utilizando componentes e o *live reload*. O conceito de multiplataforma é a capacidade do único código ser renderizado em sistemas operacionais Android e iOS. O desenvolvimento utilizando componentes, se relaciona em utilizar as *tags* nativas com componentes para facilitar no desenvolvimento da aplicação, proporcionando um alto desempenho. E o *live reload* é uma abordagem para otimizar a lentidão no momento da renderização durante o teste e compilação dos dados das aplicações (JUNIOR, 2020).

As etapas desenvolvidas durante esse projeto, foram: (i) análise de tecnologias para desenvolvimento, (ii) identificação das funcionalidades, (iii) desenvolvimento de protótipo de interface gráfica e (iv) evolução do protótipo.

### 2.1 Análise de tecnologia para desenvolvimento

A princípio duas tecnologias foram analisadas para o desenvolvimento:



- Ionic Vue: Para realizar a análise, foram desenvolvidas duas funcionalidades da aplicação: a ocorrência da (i) pulverização e de (ii) pragas, com a finalidade de conhecer a tecnologia. Apesar das funcionalidades terem sido desenvolvidas com sucesso, com essa tecnologia não foi atingido o nível de compreensão desejado para o desenvolvimento. Além disso, houve dificuldades de encontrar soluções na documentação e na comunidade. Por fim, a percepção obtida não foi positiva para dar continuidade com tal tecnologia;
- React Native: Posteriormente o *framework React Native* foi investigado. O *framework* já havia sido utilizado no primeiro projeto de desenvolvimento de aplicativo móvel e acabou sendo selecionado para dar continuidade no projeto. As principais vantagens são: (i) interage com as APIs nativas, (ii) desempenho similar a uma aplicação nativa e (iii) menor do tempo de desenvolvimento quando comparada as tecnologias nativas (JUNIOR, 2020).

## 2.2 Identificação das funcionalidades

A próxima etapa no projeto foi identificar as funcionalidades da aplicação existente, manejo.app. Há sete funcionalidades principais na aplicação, sendo elas: (i) Painel de Controle: Onde ocorrem as análises do sistema. Alguns dados são representados graficamente para facilitar sua visualização e auxiliar os técnicos para tomada de decisão; (ii) Unidade de Referência (UR): Representam propriedades rurais, no qual contém todas as informações relevantes da propriedade; (iii) Pesquisa: Quando associada a uma UR é possível gerenciar todas as informações registradas por técnicos do IDR-Paraná por meio dos lançamentos, como MIP, MID e Pulverização; (iv) Manejo Integrado de Pragas (MIP): Gerenciamento de pragas, doenças das pragas e inimigos naturais reconhecidas nas URs; (v) Manejo Integrado de Doenças (MID): Gerenciamento de entidades e profissionais para leitura de lâminas; (vi) Pulverização: Gerenciamento do alvo principal ou função e os produtos utilizados nas URs; (vii) Usuários: Gerenciamento dos usuários que utilizam o sistema manejo.app.

Para auxiliar o desenvolvimento nas próximas etapas, foi construída a árvore de componentes da aplicação web. O software *XMind* foi utilizado para fazer o mapa mental para o mapeamento (XMIND, 2021). O foco está na tela de dados da amostragem. Esta tela possui o componente de calendário, para identificação da data da coleta; o componente de listagem, usado para coleta do estágio da cultura; e o componente de entrada de texto, para a identificação da porcentagem de desfolha. Todos esses componentes são usados para entrada de dados.

## 2.3 Desenvolvimento de protótipo de interface

Após identificar as funcionalidades da aplicação manejo.app, foi definido o foco de desenvolvimento no lançamento de ocorrências de pragas. O software de prototipagem *Figma* foi usado para a criação dos layouts. O software foi escolhido por ser intuitivo. Ele também cria um link que permite que usuários acessem os protótipos online para validação. Isso facilita o feedback dos usuários (FIGMA, 2021).

A primeira versão do protótipo de interface gráfica consiste em se aproximar ao máximo da aplicação Web. Dessa forma, o usuário tem facilidade de reconhecer os componentes e a navegação entre as telas é facilitada.

Foram identificadas várias oportunidades de melhorias no primeiro protótipo. Assim, ocorreu a segunda versão do protótipo de interface gráfica. Foi considerado incluir fontes e componentes maiores, com cores que possam ser visíveis em um dia bem ensolarado, além da redução na quantidade de telas. O objetivo dessas mudanças era facilitar o uso do aplicativo pelos técnicos em campo.



Por fim, as seguintes telas foram desenvolvidas: (i) Tela de Dados da Amostragem: Esta tela permite registrar a data da coleta, o estádio da cultura atual e a porcentagem de desfolha, com algumas opções mais previsíveis; (ii) Tela de Flutuação de Pragas: Esta tela permite registrar as pragas encontradas no campo, requer informar o tamanho e a média; (iii) Tela de Doenças das Pragas: Esta tela permite registrar as doenças das pragas observadas no campo, onde para cada doença encontrada requer inserir a média; (iv) Tela de Inimigos Naturais: Esta tela permite registrar os inimigos naturais das pragas, onde para cada inimigo encontrado requer inserir a média; (v) Tela de Revisão: Esta tela permite rever todas as informações do lançamento de pragas antes que o usuário salve os dados.

Para garantir que o projeto está alinhado com a expectativa dos seus usuários, o protótipo foi compartilhado com os técnicos do IDR-Paraná e do SENAR/PR. Eles tiveram 10 dias para avaliar o protótipo e retornar seus comentários.

### 3 RESULTADOS

A Figura 1 mostra as telas desenvolvidas no protótipo de interface.



Fonte: Print retirado do software Figma (2021).

Os comentários recebidos dos técnicos foram avaliados. Porém, foram recebidos apenas comentários positivos. Isso confirmou que os protótipos eram adequados. Dessa forma, foi dado continuidade ao projeto, evoluindo os protótipos para a aplicação proposta.

#### 3.1 Evolução do protótipo

O projeto foi desenvolvido usando o *framework React Native* com a linguagem de programação *Typescript*. Os componentes desenvolvidos são: `buttonIcon`, `buttonText`, `calendarContainer`, `card`, `header`, `input`, `mip`, `pickerContainer`, `pulverization` e `searchInput`. Os principais componentes são `mip` e o `pulverization`, entretanto o foco será no componente `mip` que compõem as telas desenvolvidas na etapa do protótipo de interface gráfica.

O `formMip` é o componente responsável pela navegação entre os componentes do `mip` e armazena os dados fornecidos pelo usuário ao finalizar o lançamento dos dados de ocorrência de pragas. A Figura 2 mostra um trecho de código que exemplifica como os dados são armazenados em estados no componente `formMip`. O primeiro estado (linhas 42 a 46) armazena os dados de amostragem com valores iniciais atribuídos a ele. O segundo estado (linha 47) armazena a lista de ocorrência de pragas. O terceiro estado (linha 48) armazena a

lista de ocorrência das doenças das pragas. O quarto estado (linhas 49 a 51) armazena a lista de ocorrência de inimigos naturais. E o quinto estado (linha 52) armazena o valor da etapa durante o lançamento.

**Figura 2 – Parte do código do componente formMip, responsável pelo armazenamento de dados em estados**

```
42     const [sampleData, setSampleData] = useState<ISampleData>({  
43       sampleDate: new Date(),  
44       growthPhase: 'V5',  
45       desfolha: '0',  
46     });  
47     const [pests, setPests] = useState<IPest[]>([]);  
48     const [pestsDiseases, setPestsDiseases] = useState<IPestDiseases[]>([]);  
49     const [naturalsPredator, setNaturalsPredator] = useState<INaturalPredator[]>(  
50       [],  
51     );  
52     const [step, setStep] = useState(1);
```

Fonte: Código retirado da aplicação desenvolvida (2021).

A Figura 3 mostra o trecho de código responsável pela navegação realizada entre as telas para o lançamento do MIP. Uma estrutura condicional (switch case) é usada em conjunto com o estado do step para verificar a tela que deve ser renderizada. O estado do step é iniciado com o valor um, portanto é executado o case 1 (linhas 76 a 83). Logo é renderizado a tela Dados da Amostragem com o estado de sampleData (linha 79). O método nextStep (linha 80) quando executado atualiza o valor do step para o próximo número, no caso para dois.

O método onChangeHandlerSampleData altera os atributos do estado sampleData (linha 81). Quando o método onChangeHandlerSampleData é acionado, significa que o usuário alterou algum valor na entrada de dados da tela Dados da Amostragem. E, ao finalizar o step 1, o método nextStep é executado, renderizando a próxima tela, o case 2.

No case 2, é renderizado a tela de Flutuação de pragas. Muitos métodos dessa tela são semelhantes ao case 1. O método handleSetPests (linha 88) auxilia para inserção de cada praga. A inclusão da ocorrência da praga ocorre após acionar o método handleSetPests. O estado da lista de ocorrência de pragas da Figura 2 (linha 47) é atualizada a cada inclusão. E o método prevStep (linha 90) quando executado atualiza o valor do step para o número anterior, no caso para um, assim retornando a tela anterior. Os case 3 e 4 são semelhantes aos case 2, a diferença está na ocorrência de doenças das pragas e de inimigos naturais, respectivamente.

No case default é renderizado a tela de revisão, com os valores atualizados dos estados: dados de amostragem (linha 114), a lista de ocorrência de pragas (linha 115), a lista de ocorrência das doenças das pragas (linha 116), a lista de ocorrência de inimigos naturais (linha 117).

**Figura 3 – Código referente a navegação entre os componentes que o mip possui**

```
75     switch (step) {  
76       case 1:  
77         return (  
78           <MipSample  
79             sampleData={sampleData}  
80             nextStep={nextStep}  
81             onChangeHandlerSampleData={onChangeHandlerSampleData}  
82           />  
83         );  
84       case 2:  
85         return (  
86           <PestOccurrence  
87             pests={pests}  
88             handleSetPests={value => setPests(value)}  
89             nextStep={nextStep}  
90             prevStep={prevStep}  
91           />  
92         );  
93       case 3:  
94         return (  
95           <PestDiseasesOccurrence  
96             pestsDiseases={pestsDiseases}  
97             handleSetPestsDiseases={value => setPestsDiseases(value)}  
98             nextStep={nextStep}  
99             prevStep={prevStep}  
100          />  
101        );  
102       case 4:  
103         return (  
104           <NaturalPredatorOccurrence  
105             naturalsPredator={naturalsPredator}  
106             handleSetNaturalsPredator={value => setNaturalsPredator(value)}  
107             nextStep={nextStep}  
108             prevStep={prevStep}  
109           />  
110         );  
111       default:  
112         return (  
113           <Revision  
114             sampleData={sampleData}  
115             pests={pests}  
116             pestsDiseases={pestsDiseases}  
117             naturalsPredator={naturalsPredator}  
118             prevStep={prevStep}  
119           />  
120         );  
121       }  
122     };  
123  
124     export default FormMip;  
125
```



Fonte: Código retirado da aplicação desenvolvida (2021).

#### 4 CONCLUSÃO

O objetivo deste projeto foi desenvolver o módulo de coleta de ocorrência de pragas da aplicação móvel *manejo.mobile*. O módulo foi desenvolvido usando o framework *React Native*, trazendo bons resultados e menor tempo de desenvolvimento. A popularidade do framework na comunidade de desenvolvedores e com a documentação do *React Native* facilitaram no aprendizado da tecnologia e o esclarecimento de dúvidas obtidas no projeto.

Os resultados apresentados mostram que o desenvolvimento inicial da aplicação móvel é viável. A pesquisa e seleção da tecnologia otimizaram as próximas etapas do projeto. Com o protótipo de interface gráfica, as próximas telas a serem desenvolvidas podem se basear no design inicial. E utilizar os componentes já desenvolvidos durante a etapa da evolução do protótipo.

Foi possível agregar muito aprendizado com desenvolvimento do projeto, por meio das pesquisas para selecionar a tecnologia, alguns tópicos relacionados a área de design e a utilização de *frameworks* para desenvolvimento inicial do protótipo móvel. E, por fim, ter obtido o conhecimento da tecnologia MIP, sua finalidade em lavouras associadas e a vantagem que agrega nos plantios.

#### AGRADECIMENTOS

Ao Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq) pela bolsa de estudos para o desenvolvimento deste projeto.

#### REFERÊNCIAS

- FIGMA. **Figma: the collaborative interface design tool**. Disponível em: <https://www.figma.com/>. Acesso em: 18 jun. 2021.
- JUNIOR, Adil Calomeno. **React Native: o que é, quais as funcionalidades e as vantagens desse framework**. 2020. Disponível em: <https://ateliware.com/blog/react-native>. Acesso em: 20 jul. 2021.
- NATIVE, React. **React Native: Learn once, write anywhere**. Disponível em: <https://reactnative.dev/>. Acesso em: 05 ago. 2020.
- SILVA, Gabriel Costa, **Desenvolvimento De Aplicativos Para Coleta E Análise De Dados da Tecnologia De Manejo Integrado De Pragas Da Cultura Da Soja**. 2020.
- XMIND. **XMIND: Mind Mapping Software**. Disponível em: <https://www.xmind.net/>. Acesso em: 18 jun. 2021.