

## Desenvolvimento da interface gráfica com o usuário para dispositivos móveis

### Development of the graphical user interface for mobile devices

#### RESUMO

**Bruna Quispe Oruchi**  
[brunaoruchi@alunos.utfpr.edu.br](mailto:brunaoruchi@alunos.utfpr.edu.br)  
Universidade Tecnológica Federal do Paraná, Cornélio Procópio, Paraná, Brasil

**Gabriel Costa Silva**  
[gabrielcosta@utfpr.edu.br](mailto:gabrielcosta@utfpr.edu.br)  
Universidade Tecnológica Federal do Paraná, Cornélio Procópio, Paraná, Brasil

**Hamad Musa Ali Neto**  
[hamadmusaali@gmail.com](mailto:hamadmusaali@gmail.com)  
Universidade Tecnológica Federal do Paraná, Cornélio Procópio, Paraná, Brasil

O Instituto Paranaense de Assistência Técnica e Extensão Rural (EMATER), parceiro no desenvolvimento do projeto de inovação, ajuda sojicultores paranaenses na aplicação da tecnologia de Manejo Integrado de Pragas (MIP). Essa tecnologia permite a otimização no controle de pragas, reduzindo significativamente o uso de pesticidas e contribuindo para o desenvolvimento sustentável. Contudo, o EMATER enfrenta algumas dificuldades na execução do MIP. Para auxiliá-los, foi desenvolvida uma aplicação Web, que torna mais eficiente o processo de coleta e análise de dados do MIP. Este trabalho apresenta o desenvolvimento de um protótipo da versão para dispositivos móveis da aplicação existente. Para o desenvolvimento foi necessário compreender o funcionamento do *framework React Native* e as funcionalidades da aplicação existente. Foram desenvolvidas as telas que dão acesso às funcionalidades do aplicativo. Para dar funcionalidade às telas, o aplicativo *JSON Server* foi usado para simular a API da aplicação. A avaliação do protótipo mostrou que o desenvolvimento da aplicação para dispositivos móveis é viável e que o *React Native* é uma tecnologia apropriada.

**PALAVRAS-CHAVE:** Aplicativos móveis. Desenvolvimento de software. JavaScript.

#### ABSTRACT

Instituto Paranaense de Assistência Técnica e Extensão Rural (EMATER), partner in the development of the innovation project, it helps soybean farmers in Paraná in the application of Integrated Pest Management (MIP) technology. This technology allows optimization in pest control, significantly reducing the use of pesticides and contributing to sustainable development. However, EMATER faces some difficulties in implementing the MIP. To assist them, a Web application was developed, which makes the MIP data collection and analysis process more efficient. This work presents the development of a prototype for the mobile version of the existing application. For the development it was necessary to understand the functioning of the *React Native* framework and the functionalities of the existing application. The screens that provide access to the application's functionalities were developed. To give functionality to the screens, the *JSON Server* application was used to simulate the application's API. The evaluation of the prototype showed that the development of the application for mobile devices is feasible and that *React Native* is an appropriate technology.

**KEYWORDS:** Mobile apps. Software development. JavaScript.

**Recebido:** 19 ago. 2020.

**Aprovado:** 01 out. 2020.

**Direito autoral:** Este trabalho está licenciado sob os termos da Licença Creative Commons-Atribuição 4.0 Internacional.



## INTRODUÇÃO

O Manejo Integrado de Pragas (MIP) consiste em otimizar o controle de pragas. As lavouras comerciais que adotam o MIP, realizam um acompanhamento semanal referente aos níveis populacionais das pragas de soja. Por meio desse acompanhamento são obtidos dados para apoiar a aplicação de agrotóxicos no momento correto. Essa tecnologia tem mostrado uma redução nos custos de inseticidas nas safras, assim reduzindo impactos ambientais nas lavouras de soja (Silva, 2020).

O Instituto Paranaense de Assistência Técnica e Extensão Rural (EMATER) utiliza as técnicas MIP em lavouras associadas, porém o mesmo tem enfrentado dificuldades no processo de coleta e análise dos dados. As coletas de dados são realizadas pelos técnicos da EMATER, registrados em papel e tabulados em planilhas eletrônicas. Contudo, a análise dos dados pode ser prejudicada caso tenha algum dado inconsistente. Outra dificuldade está no tempo de execução da análise dos dados, que tem a duração de até seis meses para ser concluído (Silva, 2020).

Para auxiliar o EMATER nesta situação, surgiu a parceria para um projeto de inovação com a UTFPR, com a finalidade de desenvolver uma aplicação para a coleta e análise de dados da tecnologia do MIP, chamado manejo.app. O manejo.app permite registrar os dados coletados em campo, permite que ocorra uma padronização na entrada de dados e permite visualizar de forma mais eficiente os dados das lavouras (Silva, 2020).

Entretanto, o manejo.app possui apenas uma interface Web. Dessa forma, a coleta de dados continua sendo registrada em papéis e inseridas na aplicação posteriormente. Devido às condições de acesso à rede de internet em áreas rurais, levar um notebook ou celular para registrar e visualizar os dados em campo não é tão eficaz. Logo se torna viável a versão da aplicação para dispositivos móveis sem necessidade de ter acesso à internet e tornando mais eficiente a coleta e análise de dados.

Esse projeto elaborou um protótipo da interface gráfica para a aplicação móvel. Para cada funcionalidade foi necessário compreender a distribuição dos elementos e os componentes que estão presentes na tela, assim proporcionando maior usabilidade ao usuário. E por fim, visualizar o funcionamento da aplicação com o uso do *JSON Server* para simular a API da aplicação.

## MATERIAL E MÉTODOS

O *React Native* é um *framework* de código livre para aplicações em dispositivos móveis. Foi desenvolvido pela equipe do Facebook desde 2015. Utiliza o conceito de multiplataforma, portanto, um único código pode ser executado nos sistemas operacionais Android e iOS (Native, 2020). Diferente do conceito de interface híbrida, o *React Native* realiza a conversão do código para a interface nativa, proporcionando uma alta performance no seu desenvolvimento. Com o *React Native* é possível alcançar maior produtividade e implementações mais rápidas (Codificar, 2018).

As seguintes atividades foram realizadas na execução do projeto:

- a) Entender o funcionamento do *React Native*: o *framework* consiste em diversos componentes que, em conjunto, permitem o desenvolvimento da aplicação;
- b) Desenvolver funcionalidade de pulverização: a pulverização consiste em visualizar e registrar as informações sobre os custos, a operação e os produtos aplicados no local pelo responsável técnico.

## RESULTADOS E DISCUSSÃO

Para o desenvolvimento da aplicação foi utilizado como base o aplicativo já existente na versão desktop. Este artigo apresenta a funcionalidade da pulverização que foi desenvolvida.

A Figura 1 mostra a tela de pulverização. Essa tela permite a visualização das ações de pulverização nas unidades de referência (propriedades rurais). Adicionalmente, a tela permite o acesso a outras funcionalidades da aplicação por meio do menu principal. O menu principal está localizado na parte superior da Figura (1). Outras ações possíveis são acessíveis por botões logo abaixo das informações locais (2). As ações possíveis são: (i) operações de pulverização/aplicação (3), que permite visualizar as operações de pulverização existentes para a unidade de referência; e (ii) nova operação de pulverização (4), que permite inserir dados para criar uma nova operação de pulverização.

Figura 1 – Tela de pulverização



Fonte: Print retirado da aplicação desenvolvida (2020).

Ao selecionar o botão de operações de pulverização/aplicação, o usuário é direcionado para a tela apresentada na Figura 2. Essa tela permite a visualização dos registros das pulverizações por meio de datas de quando a coleta foi realizada (2). A seta no canto superior esquerdo (1) permite retornar para a tela anterior. Para ver mais detalhes da operação, é necessário selecionar a data de interesse (2).

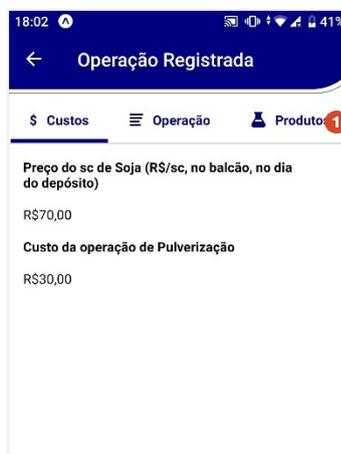
Figura 2 – Tela da operação de pulverização



Fonte: Print retirado da aplicação desenvolvida (2020).

A Figura 3 mostra o detalhamento de dados de uma operação – resultado de clicar na data 01-09-2016, apresentada na Figura 2. Foi utilizado o componente de abas (1) com o intuito de otimizar a visualização dos dados. A tela possui três abas: (i) Custos, que lista dados relacionado ao custo da operação de pulverização; (ii) Operação, que mostra a data e estádio da cultura; e (iii) Produtos, que lista os produtos usados na operação de pulverização.

Figura 3 – Tela da operação registrada



Fonte: Print retirado da aplicação desenvolvida (2020).

A Figura 4 mostra um trecho de código que implementa a tela na Figura 3. Ao pressionar o botão de salvar os dados, os dados inseridos na tela são armazenados em um objeto chamado data (linhas 68-85). O trecho de código nas linhas 86 a 95 valida se todos foram preenchidos. O trecho da linha 97 a 99 submete os dados para registro no servidor, além de confirmar o sucesso do registro da operação para o usuário.

Figura 4 – Código da Tela da nova operação de pulverização

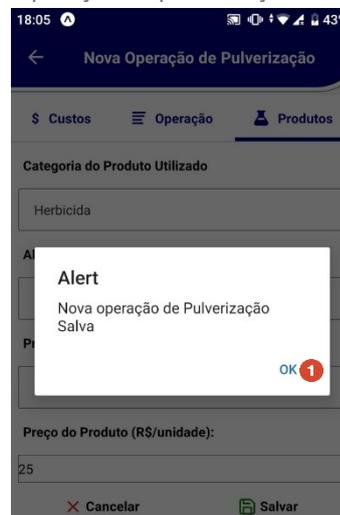
```

67 salvarDadosPulverizacao = async () => {
68   const data = {
69     data: dataHoje,
70     custo: {
71       precoScSoja: this.state.precoScSoja,
72       custoOperacao: this.state.custoOperacao,
73     },
74     operacao: {
75       dataOperacao: this.state.dataOperacao,
76       estadioCultura: this.state.estadioCultura,
77       volumeCalda: this.state.volumeCalda
78     },
79     produto: {
80       categoria: this.state.categoria,
81       alvoPrincipalouFuncao: this.state.alvoPrincipalouFuncao,
82       nomeProduto: this.state.nomeProduto,
83       precoProduto: this.state.precoProduto
84     }
85   };
86   if(this.state.precoScSoja != '' &&
87     this.state.custoOperacao != '' &&
88     this.state.dataOperacao != '' &&
89     this.state.estadioCultura != '' &&
90     this.state.volumeCalda != '' &&
91     this.state.categoria != '' &&
92     this.state.alvoPrincipalouFuncao != '' &&
93     this.state.nomeProduto != '' &&
94     this.state.precoProduto != ''
95   ) {
96     try {
97       const res = await axios.post('http://localhost:3000/operacaoRegistrada', data);
98       alert('Nova operação de Pulverização Salva');
99       return res.data;
100     } catch(error) {
101       alert('Erro do tipo: ', error);
102     }
103   }
104   else {
105     alert("Insira todos os dados");
106   }
107 }
  
```

Fonte: Código retirado da aplicação desenvolvida (2020).

Como o aplicativo atualmente em uso não está preparado para receber e enviar dados por meio de chamadas remotas, o aplicativo *JSON Server* foi usado para simular a API da aplicação. A Figura 5 mostra que a nova operação de pulverização foi salva com sucesso (1). A Figura 6 mostra o resultado da interação entre o aplicativo e o *JSON Server*.

Figura 5 – Tela da nova operação de pulverização com API em funcionamento



Fonte: Print retirado da aplicação desenvolvida (2020).

A Figura 6 mostra a primeira chamada ao método *get*, que ocorreu ao carregar as informações referente a tela da operação de pulverização da Figura 2. O método *post* ocorreu ao ser criado um novo objeto e enviado ao servidor. O último método *get* foi chamado para visualizar novamente a tela de operação de pulverização, que verifica se realmente foi registrado no servidor, como mostra a Figura 7.

Figura 6 – *Json Server* em execução

```

C:\> Selecionar C:\Windows\System32\cmd.exe - json-server --watch db.json
db.json has changed, reloading...

Loading db.json
db.json has changed, reloading...
db.json has changed, reloading...
db.json has changed, reloading...
Done

Resources
http://localhost:3000/pesquisa
http://localhost:3000/ur
http://localhost:3000/mipmid
http://localhost:3000/operacaoRegistrada

Home
http://localhost:3000

GET /operacaoRegistrada 200 9.911 ms - -
POST /operacaoRegistrada 201 191.456 ms - 376
GET /operacaoRegistrada 200 5.557 ms - -
  
```

Fonte: Print retirado do aplicativo *JSON Server* em funcionamento (2020).

Ao comparar com a Figura 2 com a Figura 7, pode-se observar que houve a inclusão de um novo objeto que já pode ser acessado na aplicação (1). Assim, a finalidade da demonstração foi alcançada para mostrar que a interface da aplicação pode se comunicar com uma API.

Figura 7 – Tela da operação de pulverização após um novo registro



Fonte: Print retirado da aplicação desenvolvida (2020).

## CONCLUSÃO

O objetivo deste trabalho foi desenvolver um protótipo para dispositivos móveis do manejo.app. Para isso foi utilizado o *React Native* que possibilitou a construção das telas da aplicação móvel com qualidade. Os componentes utilizados são intuitivos e adequados para cada funcionalidade, trazendo elementos que proporcionam maior usabilidade para o usuário. Com a popularidade do *framework* na comunidade de desenvolvimento minhas dúvidas foram solucionadas com rapidez.

Os resultados deste trabalho mostram que o desenvolvimento da versão para dispositivos móveis da aplicação manejo.app é viável, aumentando a eficácia na coleta e análise de dados. Os próximos passos deste projeto consistem em evoluir o protótipo desenvolvido e adaptar a aplicação Web existente para receber e enviar dados para o aplicativo para dispositivos móveis.

O projeto desenvolvido permitiu que os autores deste trabalho agregassem muito aprendizado, sendo por terem obtido o conhecimento de uma nova tecnologia, como *React Native*, conhecimento em chamadas REST e a gestão de tempo necessária para concluir o protótipo na data esperada. Além disso, obtiveram conhecimento da tecnologia de manejo integrado de pragas e doenças, seus objetivos e o modo que aplicação irá facilitar durante o processo de uma etapa relevante.

## AGRADECIMENTOS

O presente trabalho foi realizado com o apoio da Universidade Tecnológica Federal do Paraná/Brasil.

## REFERÊNCIAS

CODIFICAR. **O que é React Native? O futuro do desenvolvimento híbrido.** 2018.

Disponível em: <https://codificar.com.br/o-que-e-react-native-o-futuro-desenvolvimento-hibrido/>. Acesso em: 05 ago. 2020.

NATIVE, R. **A framework for building native apps using React.** 2020. Disponível em: <https://reactnative.dev/>. Acesso em: 05 ago. 2020.

SILVA, G. C., **Desenvolvimento De Aplicativos Para Coleta E Análise De Dados da Tecnologia De Manejo Integrado De Pragas Da Cultura Da Soja.** Cornélio Procópio, 2020.