

Desenvolvimento de um Aplicativo Kanban com React Native para Gerenciamento de Tarefas

Lucas Freire Sêmeler¹

¹Instituto Federal de Rondônia - IFRO - Campus Ji-Paraná

lucasemeler@gmail.com

Resumo. Este artigo detalha a concepção e o desenvolvimento de um aplicativo de quadro Kanban multiplataforma, construído com React Native e Expo. O objetivo principal do projeto é oferecer uma ferramenta intuitiva para o gerenciamento de tarefas pessoais ou de pequenas equipes, permitindo aos usuários organizar atividades em colunas de "A Fazer", "Em Andamento" e "Concluído". O aplicativo implementa funcionalidades essenciais como a criação, edição, exclusão e pesquisa de tarefas, além de um sistema de arrastar e soltar (drag-and-drop) para facilitar a movimentação de cartões entre as colunas. A arquitetura do projeto é baseada em componentes reutilizáveis e no gerenciamento de estado local através de React Hooks, demonstrando uma abordagem moderna para o desenvolvimento de aplicativos móveis.

1. Introdução

O método Kanban é um sistema visual de gerenciamento de fluxo de trabalho que visa otimizar a eficiência e a produtividade. Sua simplicidade e apelo visual o tornam uma escolha popular para o gerenciamento de projetos em diversas áreas. Diante da crescente demanda por ferramentas de organização pessoal e profissional, este projeto propõe o desenvolvimento de um aplicativo móvel de quadro Kanban, utilizando tecnologias modernas e de código aberto para criar uma experiência de usuário fluida e responsiva. O aplicativo "Meu Quadro Kanban" foi projetado para ser uma solução leve e funcional, acessível em dispositivos móveis.

2. Tecnologias Utilizadas

Para a construção do aplicativo, foram selecionadas as seguintes tecnologias e bibliotecas:

- **React Native:** Framework principal para o desenvolvimento de aplicativos móveis multiplataforma com JavaScript e React.
- **Expo:** Plataforma e conjunto de ferramentas que simplificam o desenvolvimento e a implantação de aplicativos React Native, gerenciando o ambiente de build e o acesso a APIs nativas.
- **React Native Drax:** Biblioteca utilizada para implementar a funcionalidade de arrastar e soltar (drag-and-drop), essencial para a movimentação de tarefas entre as colunas do quadro Kanban.
- **React Hooks (useState, useMemo):** Utilizados para o gerenciamento de estado local, como a lista de tarefas, textos de entrada e o controle de visibilidade de modais, garantindo uma interface reativa e performática.

3. Principais Funcionalidades

O núcleo do aplicativo reside no componente `KanbanBoard.android.js`. Ele encapsula toda a lógica e a interface do quadro. As funcionalidades implementadas são:

- **Estrutura de Colunas:** O quadro é dividido em três colunas: "A Fazer" (`todo`), "Em Andamento" (`inProgress`) e "Concluído" (`done`), permitindo uma visualização clara do status de cada tarefa.
- **Gerenciamento de Tarefas:**
 - **Adição:** Um campo de texto permite que o usuário insira novas tarefas, que são adicionadas à coluna "A Fazer".
 - **Edição:** As tarefas podem ser editadas através de um modal que é aberto ao tocar na opção "Editar", permitindo a alteração do texto.
 - **Exclusão:** Cada tarefa possui uma opção para ser removida, com um alerta de confirmação para evitar exclusões acidentais.
- **Funcionalidade de Arrastar e Soltar:** Utilizando a biblioteca `react-native-drag`, os usuários podem pressionar longamente uma tarefa e movê-la entre as colunas. A lógica do `onReceiveDragDrop` atualiza o estado do aplicativo, refletindo a mudança de status da tarefa.
- **Pesquisa de Tarefas:** Um campo de busca permite filtrar as tarefas exibidas em todas as colunas em tempo real, facilitando a localização de itens específicos.

4. Conclusão e Trabalhos Futuros

O projeto "Meu Quadro Kanban" demonstra a viabilidade de construir aplicativos de gerenciamento de tarefas eficientes e interativos com React Native. A combinação de componentes bem estruturados e bibliotecas especializadas, como a `react-native-drag`, resultou em uma experiência de usuário sólida e intuitiva.

Como próximos passos, o projeto pode ser expandido para incluir funcionalidades como:

- Persistência de dados localmente usando `AsyncStorage` ou um banco de dados como o `SQLite`.
- Sincronização em nuvem para uso em múltiplos dispositivos.
- Autenticação de usuários para quadros privados e compartilhados.
- Adição de mais detalhes às tarefas, como datas de entrega, etiquetas e responsáveis.

A base de código atual serve como um excelente ponto de partida para essas e outras melhorias, consolidando o aplicativo como uma ferramenta de produtividade completa.