

Relatório do Trabalho Final - Programação para Dispositivos Móveis

Matheus Baron Lauritzen¹, Gustavo Baron Lauritzen¹, Gabriel Bósio¹

¹Escola Politécnica – Ciência da Computação – Universidade do Vale do Itajaí (UNIVALI)
Itajaí – Santa Catarina – SC – Brasil

Abstract. *This Financial Education project in React Native aims to help users manage their expenses through efficient categorization. The key functionality developed allows users to create, edit and delete expense categories, assigning a name and limit to each category. The intuitive interface makes financial organization easier, helping users keep their finances in order and achieve their financial goals.*

Resumo. *Este projeto de Educação Financeira em React Native visa auxiliar os usuários no gerenciamento de suas despesas através da categorização eficiente. A funcionalidade-chave desenvolvida permite que os usuários criem, editem e excluam categorias de despesas, atribuindo um nome e um limite a cada categoria. A interface intuitiva facilita a organização financeira, ajudando os usuários a manterem suas finanças em ordem e alcançarem seus objetivos financeiros.*

1. Contexto

O projeto desenvolvido consiste em um aplicativo de Educação Financeira, criado com o objetivo de auxiliar os usuários a gerenciarem suas finanças pessoais de forma eficiente e organizada. O aplicativo, desenvolvido em React Native, oferece uma interface amigável e intuitiva, permitindo que os usuários acompanhem suas despesas e estabeleçam metas financeiras claras.

A funcionalidade-chave escolhida e desenvolvida para este aplicativo é a categorização de despesas. Esta funcionalidade permite aos usuários criar, editar e excluir categorias de despesas, atribuindo um nome e um limite a cada categoria. Por exemplo, um usuário pode criar categorias como "Alimentação", "Transporte" e "Lazer", e definir um limite de gastos para cada uma delas. Esta categorização ajuda os usuários a visualizarem onde estão gastando seu dinheiro e a manterem-se dentro dos limites estabelecidos, promovendo um maior controle e planejamento financeiro.

A implementação dessa funcionalidade foi projetada para ser simples e eficiente. Utilizando componentes como TextInput, Button e FlatList, o aplicativo proporciona uma experiência de usuário fluida, permitindo que as categorias sejam gerenciadas de forma dinâmica e em tempo real.

Além disso, a aplicação inclui validações, como a entrada obrigatória de valores numéricos para os limites das categorias, garantindo a integridade dos dados inseridos pelos usuários. Os detalhes do desenvolvimento dessa funcionalidade, incluindo os desafios enfrentados e as soluções implementadas, serão abordados mais adiante neste relatório.

2. Desenvolvimento

O desenvolvimento do aplicativo de Educação Financeira foi realizado utilizando o framework React Native, que permite a criação de aplicativos móveis nativos para Android e iOS a partir de uma única base de código. Esta escolha tecnológica foi motivada pela necessidade de criar uma aplicação multiplataforma de forma eficiente, mantendo uma experiência de usuário consistente em ambos os sistemas operacionais.

A principal tela do aplicativo é a tela inicial, que dá as boas-vindas aos usuários e fornece um botão para acessar a funcionalidade principal do aplicativo: a categorização de despesas. A tela de categorização foi desenvolvida no arquivo MainApp.js. Funcionalidade-Chave: Categorização de Despesas

A funcionalidade-chave do aplicativo é a categorização de despesas. Esta funcionalidade permite que os usuários criem, editem e excluam categorias de despesas, facilitando o gerenciamento financeiro pessoal. A implementação foi dividida em vários componentes:

Componente Categoria: Este componente exibe as informações de cada categoria, incluindo o nome e o limite de despesas. Componente FormulárioCategoria: Este componente permite que os usuários adicionem novas categorias. Ele inclui campos de entrada para o nome da categoria e o limite de despesas, além de validações para garantir que o limite seja um valor numérico. Tela Principal (MainApp): Esta tela gerencia a lista de categorias, renderiza os componentes de categoria e formulário, e fornece a funcionalidade de edição e exclusão de categorias.

A funcionalidade de categorização de despesas foi implementada com foco em simplicidade e eficiência. Utilizamos componentes essenciais como TextInput, Button e FlatList para oferecer uma interação intuitiva aos usuários. As categorias podem ser criadas, editadas e excluídas dinamicamente, proporcionando uma gestão flexível e em tempo real das despesas. Para garantir a precisão dos dados inseridos, implementamos validações de entrada de dados. Por exemplo, exigimos que os usuários forneçam valores numéricos para os limites das categorias, assegurando a integridade das informações registradas.

A navegação entre a tela inicial e a tela de categorização foi implementada utilizando a biblioteca react-navigation, que facilita a criação de uma navegação robusta e intuitiva no aplicativo. Configuramos um stack navigator para permitir a transição suave entre a tela inicial e a tela de categorização de despesas.

O desenvolvimento do aplicativo de Educação Financeira em React Native focou na criação de uma experiência intuitiva e eficiente para os usuários, destacando-se pela implementação da categorização de despesas. Este projeto fortaleceu nossas habilidades em desenvolvimento mobile e reforçou a importância de interfaces responsivas para facilitar a gestão financeira pessoal.

3. Limitações

Durante o desenvolvimento do aplicativo de Educação Financeira, enfrentamos algumas limitações que impactaram o processo de implementação.

Validação de Entradas: Embora tenhamos implementado validações para garantir

que os limites das categorias sejam numéricos, lidar com diferentes formatos de entrada e garantir que os usuários compreendam as mensagens de erro foi um desafio contínuo.

Navegação: A configuração inicial da navegação entre a tela inicial e a tela de categorização de despesas apresentou dificuldades. Integrar a biblioteca react-navigation e garantir uma transição suave entre as telas exigiu ajustes e testes extensivos.

Gerenciamento de Estado: O gerenciamento eficiente do estado do aplicativo, especialmente com a adição, edição e exclusão de categorias em tempo real, demandou um planejamento cuidadoso. Utilizar o hook useState foi eficaz, mas revelou limitações em cenários mais complexos, sugerindo a necessidade de soluções mais robustas como Redux em projetos futuros.

Enfrentar esses desafios nos permitiu compreender melhor as necessidades de um aplicativo multiplataforma robusto e eficiente, preparando-nos para otimizar processos e implementar soluções mais eficazes em futuros projetos.