

Relatório

1. Implementações realizadas

Descreva as principais funcionalidades implementadas

- **Navegação entre mais de uma tela:** implementada a navegação da tela de lista (TaskListScreen) para a tela de formulário (TaskFormScreen), permitindo tanto a criação de novas tarefas quanto a edição de tarefas existentes.
- **CRUD completo:** o sisteminha agora suporta a criação (Create), leitura (Read), atualização (Update) e exclusão (Delete) de tarefas, com persistência em banco de dados.
- **Formulário com validação:** foi criado um formulário usando GlobalKey, TextFormField com validação de campos (obrigatório e tamanho mínimo), DropdownButtonFormField para prioridades e SwitchListTile para o status "completo".
- **Feedback ao usuário:** foram adicionados SnackBar para confirmar ações (criação, atualização, erro) e um AlertDialog para confirmar a exclusão de uma tarefa, melhorando a UX.
- **Filtragem de tarefas:** é possível filtrar tarefas através de um botão localizado na parte superior da tela.

Liste os componentes Material Design 3 utilizados

- MaterialApp (com useMaterial3: true)
- Scaffold e AppBar
- Card (e CardThemeData global)
- FloatingActionButton.extended
- TextFormField (com InputDecorationTheme global)
- DropdownButtonFormField
- SwitchListTile
- ElevatedButton e OutlinedButton
- PopupMenuButton e PopupMenuItem
- AlertDialog
- SnackBar
- RefreshIndicator

2. Desafios encontrados

Quais dificuldades teve?

Particularmente, eu tive dificuldade em rodar o Flutter no navegador Google Chrome. Optei por seguir no web por ter mais familiaridade, mas acho que adaptar o `main.dart` pra ficar sem erros no console me deu mais trabalho do que eu esperava. Isso consumiu boa parte do meu tempo em realizar a construção do aplicativo. Mas, no fim, acho que ficou bem legal!

Como resolveu?

Resolvi o problema do Chrome instalando as dependências necessárias que "adaptavam" o projeto pra rodar na web, e alterando o `main.dart` pra comportar essas novas dependências. É possível checar o arquivo `main.dart` atualizado [aqui](#).

3. Melhorias implementadas

Como tive problemas em rodar o projeto, não adicionei nada além do roteiro base. Meu foco principal foi implementar fielmente o roteiro base. Não foram adicionadas funcionalidades "extra" além das customizações propostas (como busca ou temas).

4. Aprendizados

Principais conceitos aprendidos

- Rodar um projeto em Flutter: definitivamente meu principal aprendizado.
- Estrutura dos diretórios e do projeto em Flutter: foi bem bacana entender um pouco mais sobre uma tecnologia nova.
- Gerenciamento de estado simples: como um `StatefulWidget` pode gerenciar seu próprio estado e recarregar dados de fontes assíncronas usando `setState`, algo interessante principalmente se comparado com tecnologias como React.
- Encapsulamento de widgets: aprendi sobre importância de criar widgets customizados (como o `TaskCard.dart`) para reduzir a complexidade, reutilizar código e manter a tela principal (`TaskListScreen`) limpa e legível. Seria algo próximo da componentização em módulos do React, se forms fazer um paralelo.

Diferenças entre Lab 1 e Lab 2

O Laboratório 1 foi uma introdução ao `ListView` e ao `DatabaseService` em uma única tela, com UI básica (ex: `ListTile`). O Laboratório 2 transformou o protótipo em uma aplicação com mais de uma

tela, com uma UI um pouquinho mais avançada (widgets/), navegação, gerenciamento de estado entre telas, validação de dados e uma experiência de usuário (UX) muito mais completa e robusta.

5. Próximos passos

O que gostaria de adicionar?

Eu gostaria de adicionar mais opções de gerenciamento, tipo conseguir agendar tarefas de acordo com dias da semana.

Ideias para melhorar a aplicação

- Implementar as customizações, por exemplo, pra adicionar dark themes na aplicação.
- Implementar datas de vencimento.
- Implementar notificações locais.