

Relatório - Laboratório 2: Interface Profissional

Aluno: Lucas Carvalho Peres
Disciplina: Projeto de Software / Flutter
Tema: Task Manager – Interface Profissional com SQLite

1. Implementações Realizadas

Durante o Laboratório 2, foi desenvolvido o aprimoramento da interface do aplicativo Task Manager, transformando-o em uma aplicação com layout profissional, responsivo e completo.

As principais funcionalidades implementadas foram:

- Inclusão de campo de prioridade com `DropDownButtonFormField`.
- Implementação de data de vencimento (`DatePicker`) para cada tarefa.
- Criação do sistema de categorias coloridas e filtro por categoria.
- Adição de filtros por status (todas, completas, pendentes) com `ChoiceChip`.
- Exibição de contador dinâmico de tarefas (total, feitas, pendentes).
- Ordenação automática por data de vencimento, priorizando as tarefas mais próximas.
- Exibição de alerta visual para tarefas vencidas.

Componentes Material Design 3 utilizados: `AppBar`, `Scaffold`, `TextField`, `DropDownButtonFormField`, `ElevatedButton`, `OutlinedButton`, `ChoiceChip`, `Chip`, `ListTile`, `Card`, `Checkbox`, `IconButton`, `SnackBar`, `DatePicker`, `Wrap`.

2. Desafios Encontrados

- Persistência de novas colunas no SQLite: a inclusão dos campos `dueDate` e `categoryId` exigiu o uso do `onUpgrade` e controle de versão do banco de dados.

→ Solução: implementei migrações automáticas no `DatabaseService` com verificação de versão e `ALTER TABLE`.

- Manutenção do estado da tela: ao adicionar filtros e novos dropdowns, o uso de `setState` precisou ser ajustado para evitar rebuilds desnecessários.

→ Solução: reorganizei a lógica de atualização e carregamento de listas (`_loadAll()`).

3. Melhorias Implementadas

Além do roteiro solicitado, foram adicionadas:

- Ícone de alerta vermelho (■) para tarefas vencidas.
- Cores personalizadas nas categorias (`Category.colorHex`).
- Ordenação múltipla por vencimento e data de criação.
- Interface aprimorada com espaçamento, bordas e ícones consistentes.
- Filtros combinados (status + categoria) sem perda de desempenho.

4. Aprendizados

Apreendi a integrar banco de dados local (SQLite) com uma interface reativa em Flutter, além de manipular datas, filtros e componentes dinâmicos. Compreendi a importância de manter arquitetura modular (models, services, screens) e de usar componentes do Material Design 3 para consistência visual. A principal diferença entre o Lab 1 e o Lab 2 foi a evolução da UI simples para

uma interface profissional, com lógica de estado, feedback visual e boas práticas de usabilidade.

5. Próximos Passos

- Adicionar edição de tarefas e notificações locais de vencimento.
- Implementar modo escuro (dark mode).
- Introduzir persistência de preferências de filtro.
- Migrar o gerenciamento de estado para Provider ou Riverpod.
- Expandir o app para integração com API remota (sincronização futura).

Relatório gerado em: 31/10/2025 – PUC Minas