

Ensaio Teórico

Gestão de Dados Locais em Aplicações Flutter

A gestão de dados locais é essencial em apps móveis para manter informação acessível sem ligação à internet. Em Flutter, combinam-se soluções simples para preferências (SharedPreferences) com bases de dados estruturadas (SQLite) quando é preciso CRUD e consultas. A nossa mini-app demonstra **as duas abordagens em conjunto**: preferências leves (tema e username) via SharedPreferences e notas persistentes via SQLite com padrão DAO.

Porquê duas soluções?

- **SharedPreferences**: chave-valor, ótimo para *estado leve* (ex.: darkMode, username).
 - **SQLite (sqflite) + DAO**: estrutura tabelar, múltiplos registos, operações CRUD fiáveis e escaláveis para as **Notas**.
-

Arquitetura e ficheiros

- **lib/notes/note.dart** – Modelo Note (ex.: id, title, content).
 - **lib/notes/notes_db.dart** – *Singleton* da BD SQLite (criação/versão da tabela).
 - **lib/notes/notes_dao.dart** – Operações **CRUD** sobre Note (insert, query, update, delete).
 - **lib/notes/notes_page.dart** – UI das **Notas** (lista/edição), ligada ao DAO.
 - **lib/notes/splash_page.dart** – **Ecrã inicial** minimalista com ícone e botão **Entrar**.
 - **lib/main.dart** – *Shell* da app:
 - MaterialApp com debugShowCheckedModeBanner: false;
 - **Tema dinâmico** (Material 3): theme, darkTheme e themeMode controlado por darkMode;
 - **Persistência de preferências** (SharedPreferences) para darkMode e username;
 - Navegação: SplashPage → DefaultTabController com **2 tabs**:
 - **Definições (_PrefsPage)**: alterna **Modo Escuro** (Switch) e guarda **Username** (TextField) em SharedPreferences;
 - **Notas**: usa o ecrã existente de SQLite/DAO.
-

Fluxo de UI

SplashPage --(Entrar)--> Home (DefaultTabController)



Decisões de design

- **Separação de responsabilidades:** modelo/BD/DAO isolados da interface.
 - **Material 3 + tema persistente:** experiência consistente claro/escuro entre sessões.
 - **Simplicidade de navegação:** SplashPage dedicado e *shell* com tabs para clareza.
 - **Escalabilidade:** manter o DAO facilita futuras features (pesquisa, categorias, sync).
-

Possíveis extensões

Durante a implementação, foi também valorizada a modularidade do código: cada componente tem uma responsabilidade bem definida, facilitando futuras expansões da aplicação, tais como:

- Filtros nas notas;
 - Categorias/etiquetas;
 - Exportar/Importar BD;
 - Notificações locais para lembretes.
-

Conclusão

Em suma, esta mini-app exemplifica a aplicação prática dos conceitos de gestão de dados locais em Flutter, combinando boas práticas de programação, arquitetura limpa e persistência de informação. O resultado é uma aplicação simples mas funcional, escalável e adaptada a contextos reais de utilização.

Referências

- Flutter Documentation: <https://docs.flutter.dev>
- sqflite Package: <https://pub.dev/packages/sqflite>
- shared_preferences Package: https://pub.dev/packages/shared_preferences
- SQLite Documentation: <https://www.sqlite.org/docs.html>