

Atividade Prática – Aplicativo de Controle de Abastecimento e Veículos em Flutter

Objetivo:

Desenvolver um aplicativo **exclusivo para controle de abastecimento e gerenciamento de veículos** utilizando **Flutter** integrado ao **Firebase**. O app deve possuir **autenticação**, **armazenamento de dados em Firestore**, **theme personalizado**, e seguir a **estrutura recomendada do Flutter**. A entrega será **avaliada por commits no Git**, correspondentes a cada etapa do desenvolvimento.

Requisitos Técnicos

1. Autenticação (Firebase Authentication)

- Implementar **login e cadastro** com e-mail e senha.
- Validar campos e mensagens de erro.
- Redirecionar o usuário autenticado para a tela inicial.

2. Banco de Dados (Cloud Firestore ou RealTimeDatabase)

- Cada usuário terá uma **coleção exclusiva (por UID)**.
- Criar duas coleções principais:
 - **veiculos**
 - Campos: modelo, marca, placa, ano, tipoCombustivel.
 - **abastecimentos**
 - Campos: data, quantidadeLitros, valorPago, quilometragem, tipoCombustivel, veiculoId, consumo, observacao.
- Permitir **criar, listar e excluir** registros de veículos e abastecimentos.
- Abastecimentos devem ser associados a um veículo previamente cadastrado.

Estrutura do Projeto (recomendada pela comunidade):

Aplicar separação de responsabilidades (ex: MVVM, Provider, Riverpod, ou outro gerenciador de estado).

3. Interface e Navegação:

- **Tela inicial:**
 - Exibir **imagem ilustrativa** (logo, banner, veículo ou Endrick).
 - Incluir **Drawer lateral** com opções:
 - *Meus Veículos*
 - *Registrar Abastecimento*
 - *Histórico de Abastecimentos*
 - *Sair*
- Utilizar **Material 3** e um **ThemeData personalizado** (cores, tipografia, botões e ícones).

4. Funcionalidades Extras (bônus):

- Exibir **gráfico de consumo ou custo** por veículo usando `fl_chart` ou `charts_flutter`.
- Adicionar **animações de transição** entre telas.

Etapas de Desenvolvimento (com avaliação via Git commits)

Etapas	Descrição	Peso	Evidência (Commit Git)
1. Configuração inicial	Criação do projeto Flutter, integração com Firebase (flutterfire configure), e configuração de dependências.	1	Commit: setup: estrutura inicial do projeto
2. Autenticação	Telas de login e cadastro + integração com FirebaseAuth.	2	Commit: feat(auth): autenticação de usuário
3. Cadastro e listagem de veículos	CRUD de veículos e exibição de lista.	2	Commit: feat(veiculo): controle de veículos

4. Registro de abastecimentos	CRUD vinculado a veículos.	2	Commit: feat(abastecimento): registrar e listar abastecimentos
5. Tema e interface	Implementação do tema personalizado e Drawer.	1	Commit: style(theme): tema customizado e navegação
6. Funcionalidades extras (bônus)	Gráficos, animações ou refinamentos visuais.	2	Commit: enhancement(ui): gráficos e animações