Exercícios Flutter e API NodeJs

Exercício 1: Gerenciador de Tarefas

Criar um aplicativo de gerenciamento de tarefas

API Node.js:

Criar as seguintes rotas:

GET /tarefas: Retorna todas as tarefas

POST /tarefas: Adiciona uma nova tarefa

PUT /tarefas/:id: Atualiza uma tarefa (ex: marcar como concluída)

DELETE /tarefas/:id: Deleta uma tarefa

Utilize um arquivo JSON para armazenar as tarefas (salvo no servidor)

Aplicativo Flutter:

Crie uma tela para exibir a lista de tarefas

Use o pacote http para consumir as rotas da API

Adicione um botão para adicionar novas tarefas e um para deletar

Exercício 2: Catálogo de Produtos

Criar um aplicativo de catálogo de produtos com capacidade de visualização offline.

API Node.js:

Crie uma rota GET /produtos que retorna uma lista de produtos (nome, preço, descrição)

Utilize um arquivo Sqlite para armazenar os produtos (salvo no servidor)

Aplicativo Flutter:

Ao iniciar o aplicativo, carregue os produtos cadastrados

Use o pacote sqflite para salvar os produtos em um banco de dados SQLite local

Ofereça na listagem dos produtos botões para alterar e excluir o produto

Exercício 3: Gerenciador de Contatos

Construir um aplicativo de gerenciamento de contatos utilizando uma API

API Node.js:

Crie rotas para GET, POST, PUT, DELETE para gerenciar contatos

Salve os dados em um JSON (no servidor)

Aplicativo Flutter:

Crie uma tela para listar os contatos

Ao abrir o aplicativo, sincronize os contatos da API (JSON) e salve-os no SQLite local

Qualquer alteração (adição, edição, exclusão) no aplicativo deve:

1. Atualizar o banco de dados SQLite local

2. Chamar a rota correspondente na API para manter os dados sincronizados

Implemente um sistema de tratamento de erros e um indicador de carregamento para as operações de sincronização

Exercício 4: Aplicativo de Notícias

Criar um aplicativo de notícias que busca artigos e permite ao usuário salvar favoritos offline

API Node.js:

Crie uma rota GET /noticias que retorna uma lista de artigos de notícias (título, conteúdo, autor)

Use um arquivo JSON para os dados

Aplicativo Flutter:

Exiba a lista de artigos em uma Listagem

Crie um botão de Favoritar para cada artigo

Quando um usuário favorita um artigo, salve-o no banco de dados SQLite local

Crie uma tela separada Meus Favoritos que exibe os artigos salvos no SQLite