

✅ Stack Recomendada #1 – Fullstack TypeScript com Expo + tRPC + SQLite/SyncAdapter

🔧 Frontend e Mobile (Web, Android, iOS, Desktop via Electron)

- **Expo (React Native) + React Native Web** → escreve uma vez, roda em tudo
- **Tailwind CSS (via NativeWind)** → estilo consistente e moderno
- **Electron (opcional)** → empacotamento para desktop

🌐 Backend (API Server, quando online)

- **Node.js + tRPC** (TypeScript Remote Procedure Call)
 - elimina boilerplate REST/GraphQL
 - validações de tipos de ponta a ponta
- **Express.js** (opcional, se precisar lidar com rotas externas não-tRPC)

📦 Banco de Dados

- **SQLite local via WatermelonDB ou Expo SQLite** (modo offline)
- **PostgreSQL + Prisma ORM** (modo online, sincronizável)
- **Replicação com tools como RxDB ou custom adapters** para sync bidirecional

🕸 Gerenciamento de rede e conectividade

- **React Query / TanStack Query** com detecção de status de rede
- **Background sync com mutex de conflitos** usando mecanismos como **Dexie.js** ou **Replicache**

📱 Distribuição e Build

- Web: Vite + React Native Web

- Mobile: Expo Go / EAS Build
 - Desktop: Electron com bundle da mesma base
-

✅ Stack Recomendada #2 – Flutter + Supabase + Drift

📱 Frontend

- **Flutter** (Dart)
 - Suporte nativo a iOS, Android, Web e Desktop
 - UI altamente customizável
 - Excelente desempenho, ideal para apps “nativos”

🔄 Backend como serviço + Self-hosted opcional

- **Supabase** (PostgreSQL, Auth, Realtime, Storage)
 - Facilita o CRUD, autenticação, sincronização de dados
 - Pode ser auto-hospedado se quiser mais controle

📦 Banco de Dados Local + Sync

- **Drift** (Dart ORM para SQLite)
- **Isar** ou **Hive** (alternativas NoSQL para persistência rápida)
- Sincronização via Supabase + lógica custom de merge

🔌 Gerenciamento de estado e rede

- **Riverpod** ou **Bloc** para estado
- **Connectivity Plus** para detecção de rede
- **WorkManager** / **BackgroundFetch** para sync offline/online

Arquitetura modular

- Fácil de migrar para arquiteturas distribuídas (e.g., mesh, p2p) com packages como **Socket.IO**, **WebRTC**, etc.