

Roadmap Tecnológico - Aplicativo de Mapa de Calor e Relatos Urbanos

Front-end (Interface do usuário)

- React Native: criar app nativo para Android e iOS.
- Alternativas: Flutter ou desenvolvimento nativo.
- Web: React.js ou Vue.js, Leaflet.js ou Mapbox GL JS, Tailwind CSS ou Bootstrap.

Mapas e Geolocalização

- Google Maps Platform: mapas, geocoding e rotas.
- Mapbox: personalização e custo acessível.
- OpenStreetMap: base aberta e gratuita.
- Leaflet: biblioteca JS simples para mapas.

Back-end (lógica e processamento)

- Node.js + Express.js: APIs REST para relatos, autenticação e banco de dados.
- Alternativas: Laravel (PHP), Django (Python), Spring Boot (Java).
- WebSockets: notificações em tempo real.
- Autenticação: JWT ou Firebase Authentication.

Banco de Dados

- PostgreSQL com PostGIS: ideal para dados geoespaciais.
- Alternativas: MongoDB ou Firebase Firestore.

Serviços em Nuvem

- Firebase: autenticação, banco de dados, armazenamento e notificações.
- AWS, Google Cloud ou Azure: infraestrutura escalável.
- Vercel ou Netlify: deploy fácil da versão web.

APIs e Serviços Complementares

- Geocoding API: Google ou OpenCage.
- Reverse Geocoding: transformar coordenadas em endereços.
- Push Notifications: FCM ou OneSignal.
- Mapas de Calor: Google Heatmap Layer, Mapbox Density Maps, ou Leaflet plugins.

Segurança

- HTTPS obrigatório.
- Autenticação JWT.
- Controle de CORS na API.

- Validação e sanitização de dados.

DevOps e Monitoramento

- Git + GitHub/GitLab: versionamento.
- CI/CD: GitHub Actions ou GitLab CI.
- Sentry ou Firebase Crashlytics: monitoramento de erros.
- Docker: isolamento e deploy.

Resumo em Blocos

Front-end: React Native ou Flutter

Mapas: Mapbox, Google Maps ou OpenStreetMap

Back-end: Node.js + Express, ou Django/Laravel

Banco de Dados: PostgreSQL + PostGIS, ou Firebase

Autenticação: Firebase Auth, JWT

Notificações: FCM ou OneSignal

Hospedagem: AWS, Google Cloud, Vercel

Monitoramento: Sentry, Firebase Crashlytics

Etapas recomendadas

1. Validar a ideia: definir MVP.
2. Escolher stack: React Native ou Flutter?
3. Modelar dados: relatos com local, categoria e descrição.
4. Desenhar a UI: wireframes ou mockups.
5. Começar pelo mapa: carregar mapa, marcar pontos e enviar relatos.