

Plano de Desenvolvimento - Aplicativo da Prefeitura

1. Planejamento e Levantamento de Requisitos

Tempo: 1 semana

Atividades:

- Reuniões com representantes da prefeitura para entender necessidades.
- Definição das funcionalidades prioritárias.
- Identificação de requisitos técnicos e legais (LGPD).
- Elaboração de um documento de requisitos e backlog inicial.

Ferramentas:

- Miro ou Lucidchart - Mapear fluxo do sistema e funcionalidades.
- Trello ou Jira - Organizar backlog e tarefas.
- Google Meet / Zoom - Reuniões online.

Justificativa: Ferramentas colaborativas e fáceis de usar para manter todos alinhados, mesmo a distancia.

2. Arquitetura e Design da Solução

Tempo: 1 semana

Atividades:

- Definir arquitetura do sistema (frontend, backend, banco de dados, notificacoes).
- Escolher tecnologias e padroes de desenvolvimento.
- Criar prototipo visual (wireframe + UI inicial).

Ferramentas:

- Figma - Design de interfaces responsivas.
- Draw.io - Diagramas de arquitetura.
- Notion - Documentacao tecnica.

Justificativa: Figma facilita prototipagem rapida e colaboracao; Draw.io e otimo para esquemas tecnicos.

3. Configuração do Ambiente e Infraestrutura

Tempo: 1 semana

Atividades:

- Configuracao de repositórios.
- Ambiente de desenvolvimento local e de testes.
- Configuracao inicial de servidor/cloud.
- Integracao com CI/CD.

Ferramentas:

- GitHub - Controle de versao.
- Docker - Padronizacao do ambiente.
- AWS / Azure / GCP - Hospedagem e banco de dados.
- GitHub Actions - CI/CD.

Plano de Desenvolvimento - Aplicativo da Prefeitura

Justificativa: GitHub + Docker garantem padronizacao; cloud permite escalabilidade.

4. Desenvolvimento Backend

Tempo: 5 semanas

Atividades:

- Criacao de API RESTful.
- CRUD de ocorrencias e noticias.
- Integracao com push notifications e e-mail.
- Autenticacao de usuarios (JWT ou OAuth).
- Painel administrativo para prefeitura.

Ferramentas:

- Node.js + NestJS - Backend robusto.
- PostgreSQL - Banco de dados relacional.
- Prisma ORM - Manipulacao de dados simplificada.
- SendGrid / Firebase Cloud Messaging - Notificacoes.

Justificativa: NestJS e rapido e organizado; PostgreSQL e confiavel para dados estruturados.

5. Desenvolvimento Frontend

Tempo: 5 semanas (em paralelo com backend)

Atividades:

- Aplicativo mobile (Android e iOS).
- Versao web responsiva.
- Integracao com API.
- Autenticacao e gestao de sessoes.

Ferramentas:

- React Native - App hibrido.
- React.js - Versao web responsiva.
- Expo - Facilita build e testes de mobile.

Justificativa: React Native + React.js permitem compartilhar logica e acelerar desenvolvimento.

6. Testes e Garantia de Qualidade

Tempo: 2 semanas

Atividades:

- Testes unitarios, de integracao e end-to-end.
- Testes de usabilidade.
- Ajustes de performance e seguranca.

Ferramentas:

Plano de Desenvolvimento - Aplicativo da Prefeitura

- Jest - Testes unitarios.
- Cypress - Testes end-to-end.
- Postman - Testar APIs.
- OWASP ZAP - Verificacao de seguranca.

Justificativa: Ferramentas cobrem testes automatizados, manuais e seguranca.

7. Deploy e Entrega

Tempo: 1 semana

Atividades:

- Publicar app nas lojas.
- Publicar versao web.
- Configuracao de monitoramento.

Ferramentas:

- AWS Amplify / Vercel - Deploy web.
- Firebase App Distribution / TestFlight - Testes internos mobile.
- Google Play Console / App Store Connect - Publicacao.

Justificativa: Ferramentas simplificam deploy continuo e publicacao nas lojas.

8. Treinamento e Entrega para a Prefeitura

Tempo: 1 semana

Atividades:

- Treinar equipe da prefeitura.
- Entregar manuais de uso.
- Criar canal de suporte.

Ferramentas:

- Google Docs - Documentacao.
- Loom - Videos explicativos.
- Slack - Canal de comunicacao.

Justificativa: Treinamento e suporte inicial reduzem erros e duvidas.

9. Manutenção e Evolução

Tempo: Contínuo (pos-lancamento)

Atividades:

- Correcao de bugs.
- Inclusao de novas funcionalidades.
- Monitoramento de performance.

Ferramentas:

Plano de Desenvolvimento - Aplicativo da Prefeitura

- Sentry - Monitoramento de erros.
- Google Analytics - Acompanhamento de uso.
- GitHub Issues - Controle de solicitacoes.

Justificativa: Garantem que o sistema continue estavel e atualizado.

Resumo de Cronograma

Planejamento e requisitos: 1 semana

Arquitetura e design: 1 semana

Configuracao de ambiente: 1 semana

Backend: 5 semanas

Frontend: 5 semanas

Testes: 2 semanas

Deploy: 1 semana

Treinamento: 1 semana

Total: 17 semanas (~4 meses)