



Documentação Técnica

Sistema de Reporte Urbano Inteligente – Fala Triângulo

Descrição do Projeto

O Fala Triângulo é uma plataforma web interativa desenvolvida para facilitar o reporte, o acompanhamento e a resolução de problemas urbanos na região do Triângulo Mineiro. O sistema permite que cidadãos relatem ocorrências como buracos nas ruas, falhas de iluminação pública, acúmulo de lixo, entre outros, utilizando geolocalização e imagens. A gestão pública acessa essas informações via um painel administrativo centralizado, otimizando a resposta e promovendo a transparência.

A aplicação é responsiva, acessível, construída com React, Firebase e Chakra UI, e busca promover cidades mais inteligentes e colaborativas.

Motivação

A administração pública enfrenta desafios significativos para coletar e priorizar problemas urbanos. Canais tradicionais, como ligações ou e-mails, geram ruídos e dificultam a rastreabilidade. O Fala Triângulo surge como resposta a essa necessidade, oferecendo:

- Canal digital estruturado e rastreável
- Redução de ruídos na comunicação com o poder público
- Dados organizados e georreferenciados para tomada de decisões em tempo real

Tecnologias Utilizadas

Frontend: React, TypeScript, Chakra UI, Vite, Framer Motion, React Router DOM, Leaflet & React Leaflet, Fuse.js, Date-fns, UUID

Backend & Infraestrutura: Firebase (Auth, Realtime DB, Hosting, Storage)

Funcionalidades e Telas

1. Página Inicial (`Home.tsx`)

- Painel com destaques dos problemas mais recentes
- Acesso rápido às áreas principais: Mapa, Reportar e Perfil
- Integração com autenticação (exibe saudação personalizada)
- Componentes responsivos com Chakra UI

2. Formulário de Reporte (`ReportForm.tsx`)

- Campos obrigatórios: categoria, descrição, localização e imagem
- Geolocalização automática via GPS
- Upload com preview e envio assíncrono
- Armazenamento de dados e imagens no Firebase

3. Mapa Interativo (`MapView.tsx`)

- Visualização geográfica dos reportes
- Clusterização de marcadores
- Filtros por categoria, status e data
- Popups com resumo e acesso ao detalhe

4. Detalhes do Problema (`ProblemDetails.tsx`)

- Imagem, descrição, localização, status e histórico
- Comentários por usuários autenticados
- Atualizações e resolução por parte da prefeitura

5. Listagem de Problemas (`ProblemList.tsx`)

- Lista geral com ordenação e filtros
- Ações rápidas: editar, excluir, alterar status
- Visual diferenciada para usuários comuns e administradores

6. Perfil do Usuário (`Perfil.tsx` / `Profile.tsx`)

- Dados pessoais e estatísticas de participação
- Histórico completo de reportes realizados
- Configurações de conta

7. Área Administrativa (`Admin.tsx`)

- Dashboard com métricas e indicadores em tempo real
- Moderação de problemas e usuários
- Geração de relatórios
- PrimeiroAdmin.tsx: configuração inicial e atribuição de permissões

Scripts e Execução

Clonar o repositório

```
git clone [URL_DO_REPOSITORIO]
```

Instalar dependências

```
npm install
```

Configurar variáveis de ambiente do Firebase

(crie o arquivo .env com as chaves fornecidas)

Rodar localmente

```
npm run dev
```

Segurança

- Autenticação via Firebase Authentication
- Regras de segurança no Realtime Database
- Validação de dados em múltiplas camadas
- Proteção de rotas críticas via roles

Responsividade

A aplicação foi construída com Chakra UI, garantindo compatibilidade com diversos tamanhos de tela, incluindo desktops, tablets e smartphones.

Funcionalidades-Chave

- 📍 Reporte de problemas com geolocalização
- 🔍 Busca por filtros e categorias
- 🗺️ Visualização em mapa interativo
- 🖼️ Upload de imagens com preview
- 📊 Dashboard administrativo com estatísticas
- 👤 Perfis personalizados com histórico
- 🛡️ Sistema de moderação de ocorrências

Como Contribuir

1. Realize um Fork do projeto
2. Crie uma nova branch: `git checkout -b feature/SuaFuncionalidade`
3. Faça commits com mensagens claras: `git commit -m "feat: nova funcionalidade"`
4. Suba as alterações: `git push origin feature/SuaFuncionalidade`
5. Abra um Pull Request para revisão

Repositório do Projeto

Acesse o código-fonte completo no GitHub:
<https://github.com/VitorFigueiredoDev/TCC.git>