

# HACKATHON 2025

## DOCUMENTAÇÃO DA SOLUÇÃO

### 1. Visão Geral da Solução

**Nome da Solução:** Agenda

**Equipe:** Obelisco da Torneira

**Participantes:**

Kauan Barbosa Machado ([github.com/Kauangithub](https://github.com/Kauangithub)),

Lucas Hirotsu ([github.com/Glark398](https://github.com/Glark398)),

Rafael Ribeiro dos Santos ([github.com/lucashirotsu](https://github.com/lucashirotsu)),

Ygor da Conceição Prado ([github.com/FatalRestart](https://github.com/FatalRestart))

Sistema desenvolvido para o problema da falta de visibilidade de eventos ocorridos na cidade de Caraguatatuba, visando melhorar a identificação e entendimento sobre eventos da cidade e suas datas e locais.

### 2. Problema e Motivação

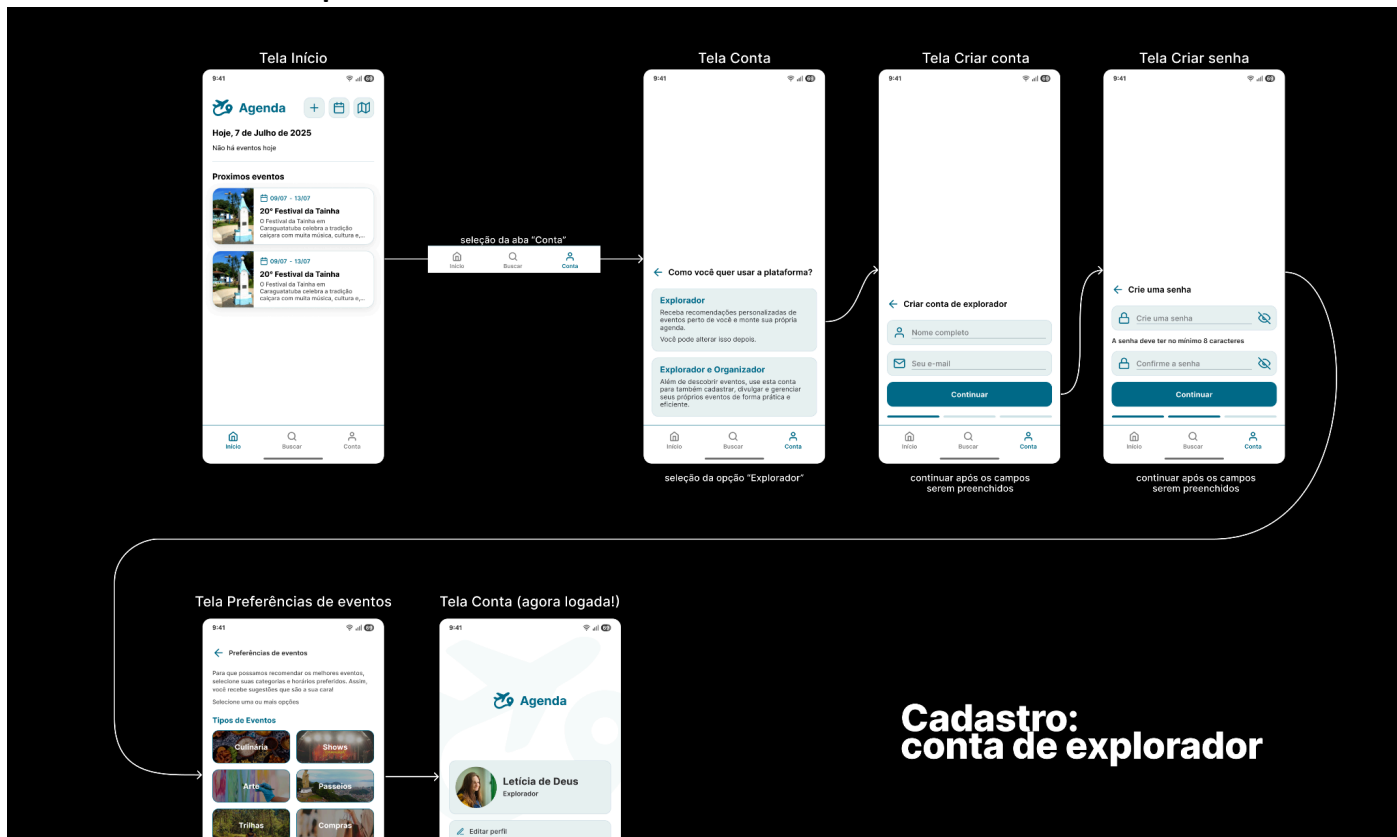
Caraguatatuba é uma cidade com grande foco turístico. Essa atenção oferece muitas oportunidades, que muitas vezes se escondem atrás de problemas. O problema que a Agenda quer solucionar é a falta de alcance de eventos, fazendo com que muitos turistas e até mesmo munícipes não conheçam as oportunidades que a cidade oferece. Esta solução pretende unificar em um local só diversos eventos, públicos ou privados, demonstrando os eventos com mais destaque e um mapa para visualizar todos os eventos que estão acontecendo e/ou devem acontecer.

### 3. Descrição da Solução

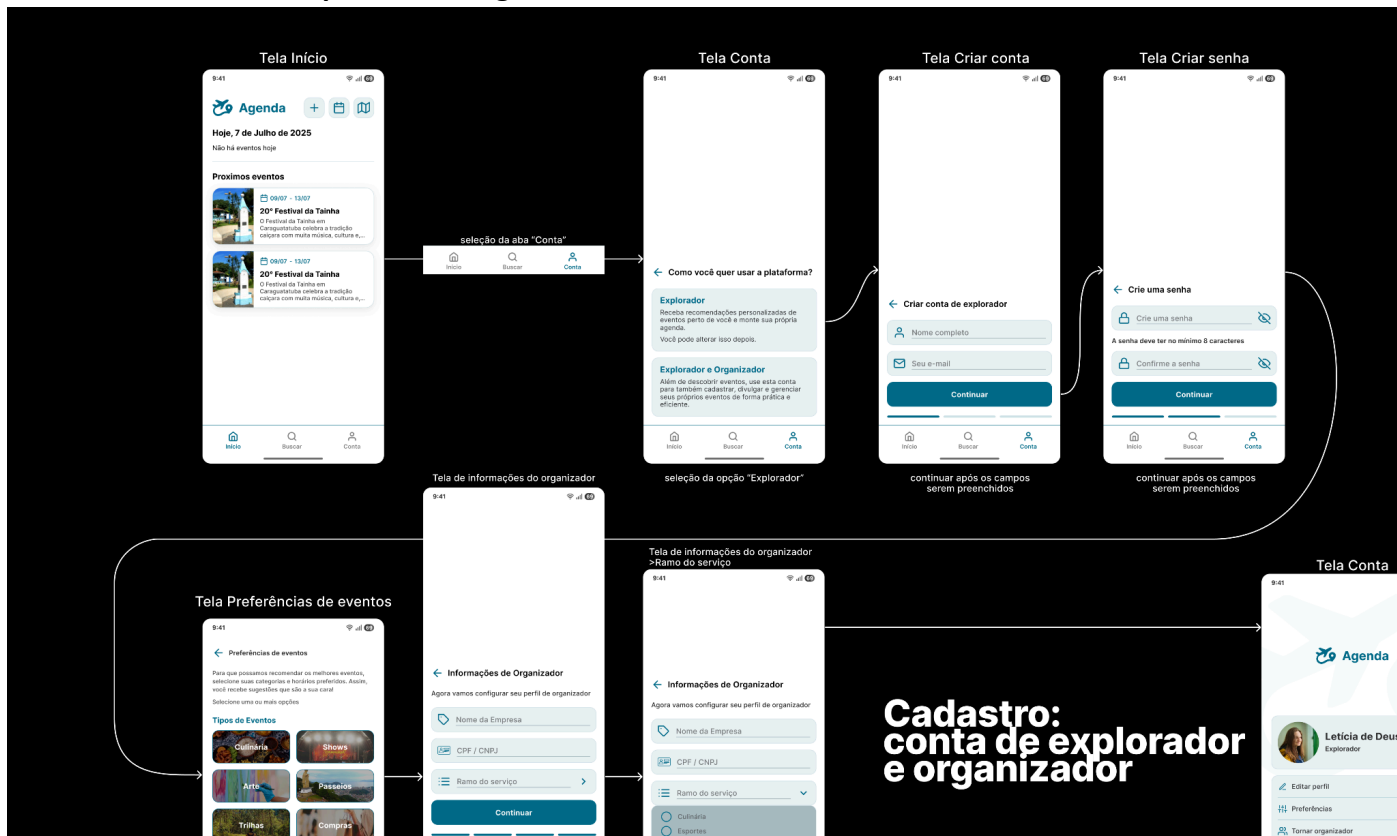
A solução permite que diferentes usuários publiquem e vejam eventos. As funcionalidades principais da solução se destacam nas opções de preferência de quais tipos de evento o usuário vai receber recomendações e no mapa interativo que permite a visualização de todos os eventos da região.

#### 3.2 Fluxo do usuário

### 3.2.1 Cadastro de um explorador

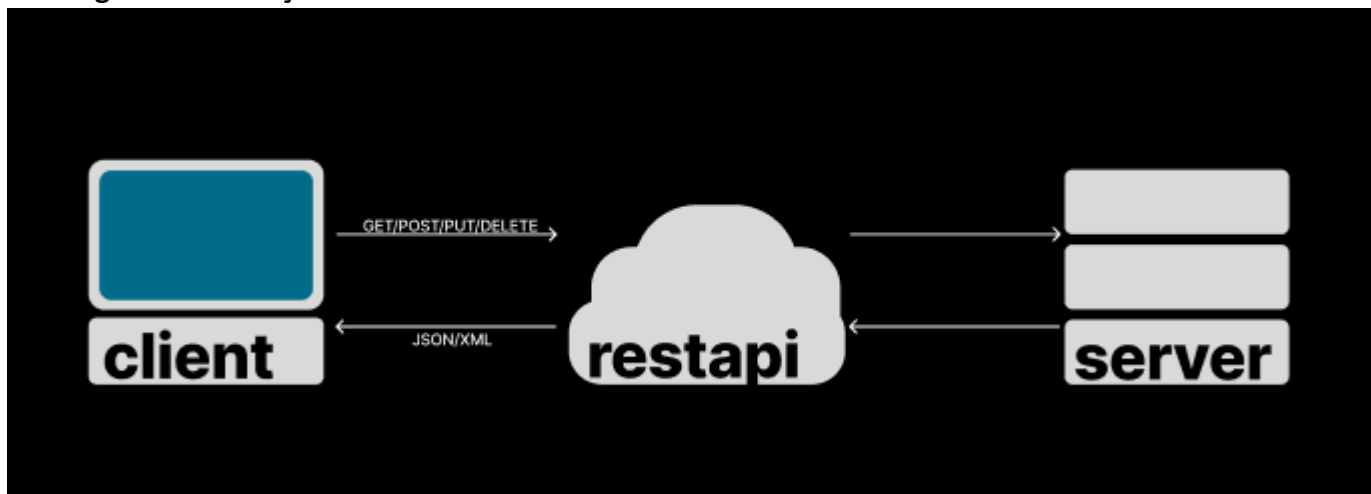


### 3.2.2 Cadastro de um explorador e organizador



## 4. Arquitetura Técnica

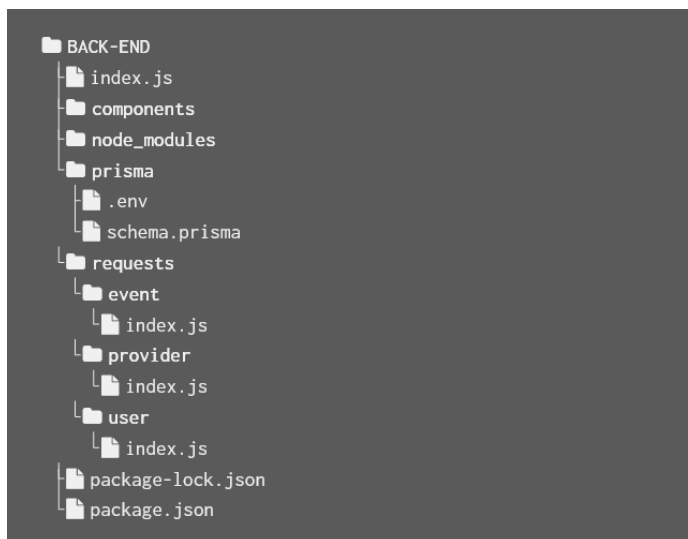
### 4.1 Diagrama da solução



#### 4.2.1 front-end

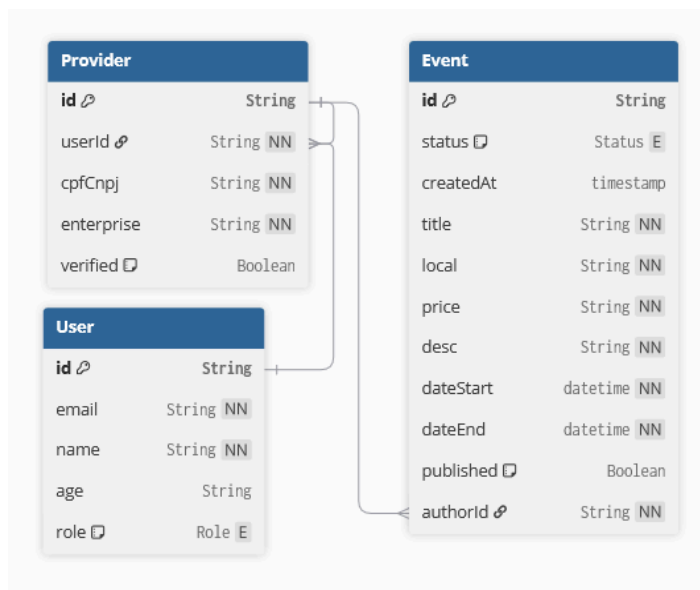
O front-end é feito em React, repassando os requerimentos pro back-end com o uso do Express que utiliza do Prisma para a utilização do Mongodb como banco de dados

#### 4.2.2 back-end



A estrutura do banco de dados é definida no arquivo `schema.prisma`, e a conexão é feita através da variável de ambiente `DATABASE_URL` presente no arquivo `.env`. O banco utiliza o provedor `mongodb`, e os dados são persistidos como documentos.

diagrama do banco de dados:



#### 4.3 Tecnologias usadas

Tecnologia	Versão	Descrição
Node.js	22.17.0	Runtime JavaScript para desenvolvimento do servidor.
Express	5.1.0	Framework minimalista e flexível para criação de APIs em Node.js.
Prisma ORM	6.11.1	ORM para facilitar a comunicação com o banco de dados MongoDB.
MongoDB	Atlas / Local	Banco de dados NoSQL orientado a documentos.
CORS	2.8.5	Permite requisições cross-origin, essencial para integração com front-ends distintos.

As tecnologias utilizadas nesta solução foram pensadas para a criação de um sistema de fácil manutenção, num ambiente pequeno e controlado. Num caso real, a opção seria implementar um banco de dados SQL com maior capacidade de tráfego

#### 5. Guia de Instalação e Execução

Para executar o projeto localmente é necessário a instalação NodeJs e a partir deste utilização do 'npm i', tanto na pasta BACK-END quanto na pasta front-end, para a instalação de todos os pacotes utilizados na produção.

## 5.1 Variáveis de ambiente

É necessária atenção para as variáveis de ambiente localizadas em 'Hackathon\BACK-END\[index.js](#)' como 'CONST PORT = 3333', para o backend, e em 'Hackathon\front-end\src\context\[maincontext.js](#)' como 'const BASE\_URL = '[http://localhost:3000](#)', essas variáveis vão definir onde os endpoints estarão e onde o front-end vai rodar. Também é necessária a criação de um .env contendo um valor de 'DATABASE\_URL' (um dado sensível que vai conter a(s) chave(s) de um banco de dados) em 'Hackathon\BACK-END\prisma\'.

## 6. Testes e Validação

A solução foi testada a partir do ambiente de desenvolvimento local. populando as tabelas via terminal com alguns dados fictícios para os testes, a partir disso foi testado a integração do front-end com o back-end, que não se demonstrou completamente implementada, funcionando apenas na página inicial, sem a opção de filtros e sem a opção de puxar imagens do banco de dados.

## 7. Próximos Passos e Escalabilidade

É esperado um desenvolvimento dos mapas do sistema utilizando de base os dados da OpenStreetMaps utilizando a biblioteca do Leaflet, mantendo escalabilidade e eficiência sem abrir mão do baixo custo. É recomendada a melhora da segurança dos requerimentos da aplicação utilizando de fatores como o OAuth. Por fim, é necessária a implementação de meios para upload de imagens, visto a importância delas para o cadastrar um evento realmente chamativo.

A solução é viável, apresentando custos médios de manutenção (salário de operador(es), manutenção de servidor e banco de dados) e ainda pode evoluir para outros itens além dos eventos, apresentando locais recomendados para visita, como praias, trilhas, etc. Também é importante pontuar a possibilidade de manter usuários verificados como aqueles cadastrados no Cadastur, ou outro critério relevante.

## 8. Anexos

Github:

<https://github.com/Kauangithub/Hackathon>

Figma:

<https://www.figma.com/design/h3Mh9DLIKKCU7EjHwqzmqE/Prot%C3%B3tipo?node-id=1-2&p=f&t=mMn2yyTWq24kdegS-0>