Curso de Sistemas de Informação

Universidade Estadual de Mato Grosso do Sul

DESENVOLVIMENTO DE UM MAPA DE ESTATÍSTICA CRIMINAL USANDO A METODOLOGIA PWA

Diego Afonso de Souza

Dr. Evandro Cesar Bracht (Orientador)

Dourados – MS
2023

DESENVOLVIMENTO DE UM MAPA DE ESTATÍSTICA CRIMINAL USANDO A METODOLOGIA PWA

Diego Afonso de Souza

Este exemplar corresponde à redação final da monografia da disciplina Projeto Final de Curso devidamente corrigida e defendida por Diego Afonso de Souza e aprovada pela Banca Examinadora, como parte dos requisitos para a obtenção do título de Bacharel em Sistemas de Informação.

Dourados, 09 de novembro de 2023.

Prof. Dr. Evandro Cesar Bracht (Orientador)

S714d Souza, Diego Afonso de

Desenvolvimento de um mapa de estatística criminal usando a metodologia PWA / Diego Afonso de Souza. – Dourados, MS: UEMS, 2023.
39 p.

Monografia (Graduação) - Sistemas de Informação – Universidade Estadual de Mato Grosso do Sul, 2023.
Orientador: Prof. Dr. Evandro Cesar Bracht.

1. Mapa estatístico criminal 2. PWA 3. Sistemas de informação I. Bracht, Evandro Cesar II. Título

CDD 23. ed. - 005.3

Ficha Catalográfica elaborada pela bibliotecária da Universidade Estadual de Mato Grosso do Sul (UEMS)
Bruna Peruffo Vieira – CRB 1/2959

Curso de Sistemas de Informação

Universidade Estadual de Mato Grosso do Sul

DESENVOLVIMENTO DE UM MAPA DE ESTATÍSTICA CRIMINAL USANDO A METODOLOGIA PWA

Diego Afonso de Souza

Novembro de 2023

Banca Examinadora:

Prof. Dr. Evandro Cesar Bracht (Orientador)
Área de Computação – UEMS

Prof. Dr. Diogo Fernando Trevisan
Área de Computação – UEMS

Profª. Dra. Raquel Marcia Muller
Área de Computação – UEMS

RESUMO

A evolução da web nos últimos anos possibilitou o surgimento de novas tecnologias e serviços, como os Progressive Web Apps (PWAs), que conseguem entregar ao usuário a mesma experiência de utilização que ele teria em uma aplicação nativa de um dispositivo móvel. A criminalidade é um assunto de grande interesse, tanto por parte da população quanto por parte dos órgãos de segurança pública. Tal interesse inspirou o desenvolvimento de uma ferramenta que possa ser utilizada pela Polícia Militar do Estado de Mato Grosso do Sul, inicialmente limitando os testes à região oeste da cidade de Dourados, com o objetivo de auxiliar o operador na tomada de decisões durante suas atividades. Para isso, a utilização do PWA apresentou-se como uma solução no desenvolvimento dessa ferramenta, que demonstra visualmente ao operador círculos no mapa de tamanhos e intensidades de cor diferentes, de acordo com os índices de criminalidade da região oeste da cidade de Dourados.

PALAVRAS-CHAVES: Mapa Estatístico Criminal; PWA; Dourados.

SUMÁRIO

LISTA DE SIGLAS

API	Application Programming Interface
CSS	Cascading Style Sheets
CIPM	Companhia Independente de Polícia Militar
DOM	Document Object Model
HTTP	HyperText Transfer Protocol
HTML	HyperText Markup Language
IBGE	Instituto Brasileiro de Geografia e Estatística
PWA	Progressive Web App
URL	Uniform Resource Locator
XHR	XMLHttpRequest

LISTA DE FIGURAS

1. INTRODUÇÃO

Com o passar do tempo, o uso de smartphones têm se tornado essencial no dia a dia da sociedade. Esse aparelho se mostra muito eficiente aumentando a praticidade em realizar tarefas cotidianas, facilitando assim a vida de seus usuários. Segundo pesquisa realizada pelo IBGE em 2021 (IBGE, 2021), a quantidade de residências em que havia smartphone alcançou 96,3%. Da população com 10 anos ou mais de idade que utilizou a Internet, o meio de acesso indicado por maior número de pessoas foi, destacadamente, o telefone móvel celular com 98,8%.

Devido a esta demanda crescente na utilização de internet em smartphones, os aplicativos móveis estão alcançando um grande sucesso no mercado, atraindo diversas empresas interessadas em ter seus próprios aplicativos móveis, buscando integrar parte da rotina de seus usuários de forma mais eficiente e produtiva.

No entanto, como no mercado existem diversas marcas e modelos de smartphones, com diferentes Sistemas Operacionais, o desenvolvimento de uma aplicação de forma nativa para cada dispositivo se tornou caro, em contrapartida as tecnologias web sofreram diversas melhorias permitindo mais recursos e capacidade, onde antes só era possível quando se desenvolvia uma aplicação nativa surgem novas possibilidades ao consolidar o desenvolvimento apenas em aplicativos web, que tem como requisito básico apenas um browser.

A criminalidade é um assunto de grande interesse, tanto por parte da população, quanto por parte dos órgãos de segurança pública. Constantemente debatidos na imprensa, o aumento dos índices de criminalidade afeta a segurança e tranquilidade de todos. Visando a resolução desse problema, foi desenvolvido um aplicativo capaz de demonstrar visualmente aos usuários os índices de criminalidade em regiões de uma cidade. Através desta visualização dos dados, será possível alertar a população de pontos críticos, assim como auxiliar a Polícia Militar na decisão de onde podem ser melhor empregados os recursos disponíveis.

Para a realização deste projeto, foram utilizadas ferramentas de desenvolvimento web convencionais, incluindo HTML, CSS, JavaScript e frameworks, além do Open Street Map, que é um projeto colaborativo para a criação de mapas, e o Leaflet, que é uma biblioteca JavaScript de código aberto usado para manipulação de mapas. Para garantir a compatibilidade em diversas plataformas e dispositivos, foi utilizado a metodologia de PWA.

Ao utilizar a metodologia PWA torna a aplicação acessível em diferentes dispositivos, e proporcionaram uma solução eficiente e prática para o problema de segurança pública em Dourados. Espera-se que essa aplicação possa ser um importante instrumento para a Polícia Militar e população local, ajudando a mapear e identificar as áreas mais críticas e contribuindo para a tomada de decisões estratégicas no combate à criminalidade.

1.1. JUSTIFICATIVA

A falta de segurança pública é um problema frequentemente discutido na imprensa. Na cidade de Dourados observa-se um aumento na criminalidade à medida que a cidade cresce e isso afeta a tranquilidade da população. Com a finalidade de ajudar a melhorar a segurança, verifica-se a necessidade de desenvolver uma aplicação para ser uma ferramenta de auxílio à tomada de decisões da Polícia Militar e posteriormente se necessário para outros órgãos da segurança pública. Atualmente, na cidade de Dourados, a qual será a região utilizada para os testes, não existe qualquer aplicação com essa finalidade.

Para o desenvolvimento, foi aplicada a metodologia de PWA, resultando em uma aplicação móvel multiplataforma com tempo de desenvolvimento reduzido. Além disso, a utilização de PWA resolve um dos problemas existentes nas aplicações nativas, que é a necessidade de acessar uma loja de aplicativos para baixar e instalar, o que ocupa espaço de armazenamento do aparelho. Com o PWA, o usuário pode acessar a aplicação diretamente pelo navegador do smartphone. Isso permite uma experiência mais fluida e acessível para os usuários.

1.2. OBJETIVOS

O objetivo deste trabalho foi desenvolver uma aplicação utilizando a metodologia de PWA que proporciona-se aos usuários uma experiência de uso próxima ou superior ao uso de uma aplicação nativa. Além disso, buscou-se realizar uma análise de vantagens e desvantagens do PWA para identificar em quais situações seu uso traz vantagens e, com isso evidenciar a importância, para o desenvolvedor, de aprender novas metodologias, facilitando a resolução de problemas com métodos diferentes.

Bem como, disponibilizar para a 9ª CIPM e população da cidade de Dourados, uma aplicação capaz de apresentar visualmente os locais com alto índice de criminalidade, e para

atualizações futuras, a possibilidade de gerar relatórios estatísticos através dos dados criminalísticos.

Além disso, a aplicação desenvolvida traz diversos benefícios para a 9ª CIPM e a população de Dourados. Por meio dela, é possível realizar uma análise mais precisa e visual dos dados criminais, identificando áreas com alto índice de criminalidade. Isso permite à polícia direcionar seus esforços de forma mais eficiente, reforçando a presença em locais de maior risco e adotando medidas preventivas. Ao mesmo tempo, a população também pode se beneficiar ao ter acesso a informações atualizadas sobre a segurança em sua região, podendo tomar medidas de proteção adicionais, colaborar com as autoridades e contribuir para a construção de uma comunidade mais segura e colaborativa.

Essa aplicação representa um importante avanço no combate à criminalidade e fortalece a parceria entre a 9ª CIPM e a população de Dourados. Ao unir tecnologia, análise de dados e a participação da comunidade na visualização e busca de informações, desta forma promovendo uma abordagem mais eficaz para enfrentar os desafios de segurança em nossa região.

2. PROGRESSIVE WEB APP (PWA)

No ano de 2015, com o objetivo de trazer evolução para as aplicações web e competir com os aplicativos nativos, surgiu um novo conjunto de padrões definido pelo Google para formalizar o desenvolvimento nas plataformas web. Esse conjunto de padrões tem como objetivo trazer recursos de um aplicativo nativo, como suporte offline, gerenciamento de páginas em segundo plano, notificações push, possibilidade de instalação, entre outros, para a plataforma web. Essa metodologia é chamada de Progressive Web App (PWA), uma solução web para plataformas mobile que utiliza a API service worker para entregar novos recursos, o que não era possível com as tecnologias web convencionais (Biørn et al., 2017).

A distribuição de um Progressive Web App (PWA) é facilitada pois independe de uma loja de aplicativos; isto é feito acessando o site da aplicação através de um browser, e a partir desse ponto o usuário tem a opção de realizar a instalação em seu dispositivo, com isso a execução do aplicativo independe da plataforma e sistema operacional que está sendo executado. Os PWAs foram projetados para serem progressivos, responsivos, seguros e instaláveis, assim trazendo para o usuário a mesma experiência como se estivesse em uma aplicação nativa da sua plataforma.

2.1. CONCEITO

A metodologia de Progressive Web App (PWA) consiste na utilização de tecnologias web para desenvolvimento de aplicações que são executadas em navegadores de dispositivos mobile ou desktop multiplataformas (Web Dev, 2023).

Segundo a Rede de Desenvolvedores da Mozilla (MDN, 2021a), a aplicação para ser considerada um Progressive Web App (PWA) precisa atender a alguns padrões, como:

- Detectável: seu conteúdo deve ser encontrado pelos buscadores;
- Instalável: estar disponível na tela inicial do dispositivo;
- Vinculável: capaz de compartilhá-lo enviando um URL;
- Independente de rede: capaz de funcionar sem conexão de rede;
- Progressivo: Pode ser usado em um nível básico em navegadores mais antigos, mas totalmente funcional nos mais recentes;

- Reengajados: para que possa enviar notificações sempre que houver novo conteúdo disponível;
- Responsivo: Possibilita ser usado em qualquer dispositivo com tela e navegador (telefones celulares, tablets, laptops, TVs, etc);
- Seguro: As conexões entre o usuário, o aplicativo e seu servidor são protegidas contra terceiros.

Oferecer esses recursos e fazer uso de todas as vantagens oferecidas pelos aplicativos da web pode criar uma oferta atraente e altamente flexível para seus usuários e clientes.

Para tornar mais claras as diferenças entre PWAs e aplicações web convencionais, aplicativos híbridos e aplicativos nativos, abaixo são apresentadas algumas das características de cada tecnologia:

- Aplicações Web convencional: São aplicações executadas através de navegadores, desenvolvidas com tecnologias web como HTML, JavaScript e CSS (MDN, 2021b). A execução pode ser feita via servidor HTTP ou localmente no dispositivo do usuário.
- Aplicações nativas: São aplicações desenvolvidas a partir da tecnologia específica do sistema operacional em que está sendo executada, como por exemplo Android ou IOS, possibilitando o acesso a todas as funcionalidades do dispositivo. Este tipo de aplicação é disponibilizada em lojas virtuais, para ser baixada e instalada.
- Aplicações híbridas: Também são aplicações desenvolvidas com tecnologia web e tem o comportamento parecido com o de aplicativos nativos. Com o desenvolvimento de apenas um código é possível gerar aplicações para diversos sistemas operacionais onde são executados na WebView dos dispositivos. Sua instalação se dá através do acesso à loja virtual do dispositivo (MADEINWEB, 2019).

Para a implementação de um PWA temos como adição ao método tradicional de desenvolvimento de aplicações web (HTML, javascript, CSS) a adição de um Service Worker e um Web App Manifest. Com o Service Worker é possível interceptar as requisições da aplicação e guardar os resultados no lado do cliente, permitindo um acesso de dados mais rápido e o funcionamento offline. O Web App Manifest tem como finalidade a padronização da aplicação, o que possibilita que o usuário adicione um ícone do site em sua tela inicial sem a necessidade de baixar um aplicativo.

2.2. SERVICE WORKER

O service worker é um arquivo JavaScript que roda em segundo plano na página web do cliente em que está sendo executado. Seu funcionamento é baseado em eventos, assim atuando como um intermediário na comunicação entre o cliente e o servidor (MDN, 2021c). Caso ocorra uma perda da conexão com a internet o service worker é o responsável por manter o funcionamento da aplicação.

A execução de um service worker é feita em um contexto isolado, assim não tem acesso ao DOM (Document Object Model) e roda em uma thread separada. São projetados para serem totalmente assíncronos e por consequência não permitem o acesso à APIs como XHR síncrono e localStorage (MDN, 2021c).

Para evitar problemas com segurança, só é possível utilizar o service worker em sites que utilizam o protocolo HTTPS como meio de transmissão de dados entre cliente e servidor, o qual é necessário para estabelecer uma conexão segura entre o navegador e o servidor web, através da criptografia dos dados que estão sendo trafegados durante a comunicação, assim evitando invasões e modificação dos dados (MDN, 2021c).

O processo de instalação do service worker se inicia através do registro na página, exemplificado na Figura 1, com a finalidade de informar ao navegador a localização do arquivo JavaScript do service worker. Para isso é utilizado o método navigator.serviceWorker.register(). Se não ocorrer nenhum erro, o service worker será descarregado no cliente e então será realizada a tentativa de instalação/ativação para as URLs acessadas pelo usuário sob a origem registrada, ou se preferir, apenas em um subconjunto específico (MDN, 2021d).

```
20  async function registerSW() {
21      if ('serviceWorker' in navigator) {
22          try {
23              const reg = await navigator.serviceWorker.register('./sw.js');
24              console.log('Service Worker registrado com sucesso!', reg);
25          } catch (err) {
26              console.log('Registro do Service Worker falhou!', err);
27          }
28      }
29  }
```

Figura 1: arquivo app.js do código da aplicação

Após o passo de registro, o service worker terá o seguinte ciclo de vida: download, instalação e ativação. No primeiro acesso à página, o service worker já é descarregado no navegador do usuário, e posteriormente será feito novos downloads em um intervalo de tempo de 24 horas (MDN, 2021c). Sempre que for identificado um arquivo novo, será realizado o processo de instalação.

Na Figura 2 é demonstrado através do trecho de código, como é realizado o evento “install” do service worker:

```
5  self.addEventListener('install', (event) => {
6      console.log('Service worker install event!');
7      event.waitUntil(
8          caches.open(cacheName).then((caches) => {
9              caches.addAll([ /* Lista os arquivos que serão pré armazenados em cache */
10                 '/',
11                 '/index.html',
12                 '/home.html',
13                 '/css/styles.css',
14                 '/js/main.js',
15                 '/js/app.js',
16                 '/js/login.js'
17             ])
18         })
19     );
20 });
```

Figura 2: arquivo sw.js do código da aplicação

2.3. WEB APP MANIFEST

O web app manifest é um arquivo JSON que descreve e configura um Progressive Web App (PWA) para ser instalado e executado em dispositivos móveis e desktop. Ele fornece informações importantes ao navegador sobre o aplicativo, como seu nome, ícone e a URL que deve ser aberta quando o aplicativo é iniciado. Além disso, permite que a aplicação seja adicionada à tela principal do dispositivo, proporcionando uma experiência semelhante à de um aplicativo nativo. Essa funcionalidade oferecida pelo web app manifest facilita a instalação, acesso e utilização dos PWAs, contribuindo para uma experiência mais imersiva e integrada aos dispositivos dos usuários. (MDN, 2021e)

Além das informações básicas, é possível especificar as configurações de exibição do aplicativo, como a orientação da tela, as cores de tema preferenciais e a exibição em tela cheia. Isso oferece aos desenvolvedores um maior controle sobre a aparência e o comportamento do aplicativo, permitindo personalizá-lo para se adequar às necessidades e

preferências dos usuários. Com o uso do web app manifest, os PWAs podem fornecer uma experiência consistente e agradável em diferentes dispositivos e plataformas, com isso pode haver um possível aumento na adoção e engajamento dos usuários.

Na Figura 3 é apresentado o código do arquivo manifest.json:

manifest.json > ...

```
{
    "short_name": "Mapa Criminal",
    "name": "Mapa Criminal",
    "start_url": ".",
    "display": "standalone",
    "theme_color": "#ffffff",
    "theme-color": "#ffffff",
    "background_color": "#000000",
    "background-color": "#000000",
    "description": "Demonstração do indice criminal atravez do mapa",
    "icons": [
        {
            "src": "/img/icons/logo-icon.png",
            "type": "image/png",
            "sizes": "256x256"
        }
    ]
}
```

Figura 3: arquivo manifest.json do código da aplicação

Para adicioná-lo ao projeto basta inserir o Código 1 dentro da tag <head> no HTML.

```
<link rel="manifest" href="/manifest.json">
```

Código 1: cria um link para o arquivo manifest.json

2.4. VANTAGENS

Essa metodologia traz algumas vantagens, a mais importante seria o fato de os programadores não precisarem desenvolver aplicações específicas para diferentes plataformas, seja ela mobile ou desktop, pois funcionarão perfeitamente em ambas. E mesmo para aqueles que desenvolvem apenas para iOS e Android, terão o benefício de reduzir o tempo e esforço que teriam se fosse produzir para uma plataforma nativa, pois com pequenas modificações em sites ou aplicações web já implementadas é possível transformá-las em PWA.

As PWAs, diferentemente das aplicações web convencionais, podem ser executadas em modo offline, ou seja, sem a necessidade de uma conexão com a internet, bastando apenas a instalação da aplicação no dispositivo. Nesse processo é exigido uma conexão com internet para que seja possível baixar os arquivos necessários localmente. A partir deste ponto o usuário terá o app em sua tela inicial que deixará de ter um aspecto de site passando a sensação de ser uma aplicação nativa instalada.

O fato de não necessitar acessar uma loja virtual do aparelho para realização da instalação, facilita o acesso do usuário, pois o usuário pode acessar a aplicação através da URL, com isso é possível que haja um aumento no uso da aplicação.

Outra vantagem importante é que o PWA tem foco em performance e otimização, possibilitando que pessoas em zonas de baixa qualidade de conexão consigam acessar a aplicação.

2.5. DESVANTAGENS

Embora os PWAs tenham algumas vantagens em relação aos aplicativos nativos, há algumas desvantagens que precisam ser consideradas.

Uma das principais desvantagens dos PWAs é a sua compatibilidade limitada com os dispositivos mais antigos e navegadores menos atualizados. Isso ocorre porque alguns recursos do PWA, como a API Service Worker, não são suportados em todos os navegadores. Isso pode afetar a experiência do usuário e limitar a adoção de PWAs em certos segmentos do mercado.

Apesar de permitirem uma experiência de usuário mais rápida e fluida em comparação com as aplicações web convencionais, os PWAs ainda podem ter um desempenho ligeiramente inferior em relação aos aplicativos nativos em alguns aspectos, como a velocidade de carregamento de recursos pesados, como vídeos ou animações complexas.

Por fim, embora os PWAs ofereçam muitas vantagens em relação aos aplicativos nativos, ainda não são tão conhecidos do público em geral, o que pode afetar a adoção e a popularidade dessa tecnologia.

3. FIREBASE

O Firebase é uma plataforma de desenvolvimento do Google que oferece aos usuários uma variedade de ferramentas para criar e gerenciar aplicativos web e mobile. Ele funciona como um serviço de back-end na nuvem, conhecido como BaaS (Backend as a Service), que automatiza grande parte do desenvolvimento do back-end. Ao utilizar o Firebase, é possível aproveitar sua infraestrutura na nuvem, o que elimina a necessidade de configurar e gerenciar servidores tradicionais (GOOGLE, 2021a).

Um dos principais motivos em optar pela utilização do Firebase para este projeto, em vez de um servidor back-end convencional, é sua integração com diversas funcionalidades específicas para PWAs. Isso significa que não é necessário desenvolver um servidor back-end separado, permitindo que os desenvolvedores se concentrem principalmente no front-end do aplicativo. Essa abordagem simplificada resulta em um aumento na produtividade e economia de tempo e recursos, já que a responsabilidade de gerenciar os servidores é terceirizada para o Firebase. Vale destacar que a plataforma Firebase oferece um plano gratuito, o que torna ainda mais atrativo para desenvolvedores com recursos limitados.

Dentre os serviços disponíveis no Firebase, foram utilizados o Authentication e o Cloud Firestore para o desenvolvimento deste projeto. O Firebase Authentication oferece recursos de autenticação de usuários, permitindo que os usuários se registrem, façam login e gerenciem suas contas de forma segura. Já o Cloud Firestore é um banco de dados NoSQL que permite armazenar e sincronizar dados em tempo real, fornecendo uma solução escalável e flexível para o gerenciamento de informações na aplicação.

Com a utilização do Firebase Authentication e Cloud Firestore, o projeto se beneficia da segurança, escalabilidade e facilidade de integração oferecidas por esses serviços. Essa combinação permite criar um PWA com recursos de autenticação de usuário e armazenamento de dados eficientes, sem a necessidade de configurar e gerenciar um servidor back-end tradicional.

3.1. AUTHENTICATION

O Firebase Authentication é um serviço de autenticação oferecido pela plataforma Firebase, desenvolvida pelo Google. Ele fornece uma solução completa e fácil de usar para implementar recursos de autenticação em aplicativos, permitindo que os usuários se autentiquem de maneira segura e eficiente (GOOGLE, 2021a).

Uma das principais vantagens do Firebase Authentication é sua ampla gama de métodos de autenticação suportados. Ele oferece suporte a diferentes opções de autenticação, como autenticação por e-mail/senha, autenticação social (usando contas do Google, Facebook, Twitter, etc.), autenticação com telefone, autenticação com provedores de identidade federados (como o OAuth) e até mesmo a possibilidade de autenticação anônima.

O processo de configuração do Firebase Authentication é relativamente simples. Ele pode ser facilmente integrado a diferentes plataformas de desenvolvimento, incluindo Android, iOS e web. O Firebase Authentication fornece SDKs e bibliotecas prontas para uso, que permitem uma integração suave no aplicativo.

Ao usar o Firebase Authentication, é possível lidar com o gerenciamento de usuários de forma eficiente. Ele fornece recursos para criação de contas de usuário, gerenciamento de perfis, redefinição de senhas e verificação de e-mail. Isso garante que os usuários tenham uma experiência segura e personalizada.

Também facilita a implementação de recursos adicionais, como autenticação de dois fatores (2FA) e gerenciamento de permissões de usuário. Esses recursos adicionais podem ser essenciais para melhorar a segurança e a proteção dos dados do aplicativo.

Outra vantagem do Firebase Authentication é a integração com outros serviços do Firebase, como o Firebase Realtime Database e o Cloud Firestore. Isso permite associar informações adicionais aos perfis de usuário autenticados e oferecer uma experiência personalizada com base nas necessidades individuais dos usuários.

Em relação à segurança, o Firebase Authentication utiliza práticas de segurança padrão do setor. Ele protege as informações confidenciais dos usuários, como senhas, por meio de criptografia e técnicas avançadas de proteção de dados. Além disso, é atualizado regularmente pelo Google para fornecer patches de segurança e correções de bugs, garantindo que os aplicativos possam se beneficiar das últimas proteções disponíveis.

Na Figura 4 é visualizado o painel de configuração ao acessar o Firebase Authentication:

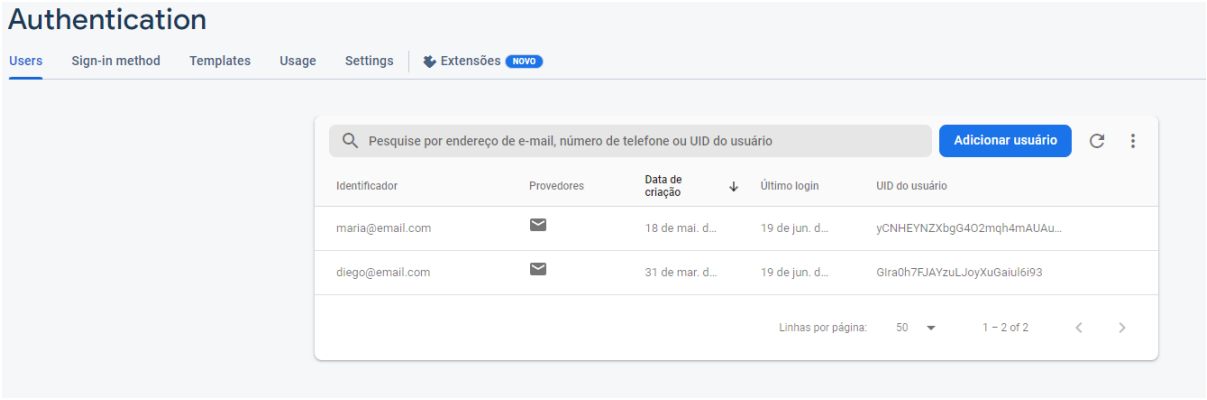

Figura 4: Interface do Authentication

3.2. CLOUD FIRESTORE

O Firebase Cloud Firestore é um banco de dados NoSQL (Não Relacional) que permite armazenar, sincronizar e consultar dados de forma eficiente para aplicações desenvolvidas tanto na plataforma web ou mobile (GOOGLE, 2021a).

É projetado para oferecer um modelo de dados flexível e escalável. Ele armazena dados em coleções e documentos, que podem ser organizados de maneira hierárquica e aninhada. Os dados são armazenados em um formato JSON-like, permitindo um acesso rápido e eficiente.

Uma das principais vantagens do Cloud Firestore é a sincronização em tempo real dos dados. Isso significa que qualquer alteração feita nos dados é automaticamente propagada para todos os clientes conectados, sem a necessidade de atualizações manuais ou solicitações adicionais. Essa capacidade de sincronização em tempo real permite que os aplicativos sejam altamente interativos e ofereçam uma experiência colaborativa para os usuários.

Oferece uma API simples e intuitiva para acessar e manipular os dados armazenados. Os desenvolvedores podem ler, gravar, atualizar e excluir dados de forma eficiente, usando consultas flexíveis para recuperar informações específicas. A API também oferece recursos avançados, como transações e batch writes, para garantir a integridade dos dados e otimizar as operações em lote.

Além disso, possui integração nativa com outros serviços do Firebase, como o Firebase Authentication e o Firebase Storage. Isso permite o desenvolvimento de aplicativos

completos, incluindo autenticação de usuários, armazenamento de arquivos e muito mais, usando uma plataforma unificada.

Em questão de segurança é utilizado regras baseadas em declarações que podem ser definidas pelo desenvolvedor para controlar o acesso aos dados. Essas regras podem ser personalizadas com base em autenticação de usuário, permitindo que seja definido permissões granulares para diferentes operações e níveis de acesso.

Possui ainda recursos de escalabilidade automática, que ajustam automaticamente a capacidade do banco de dados para atender à demanda dos aplicativos. Isso significa que o banco de dados pode lidar com um grande volume de dados e alto tráfego sem problemas de desempenho.

Na Figura 5 é visualizado o painel de configuração ao acessar o Cloud Firestore:

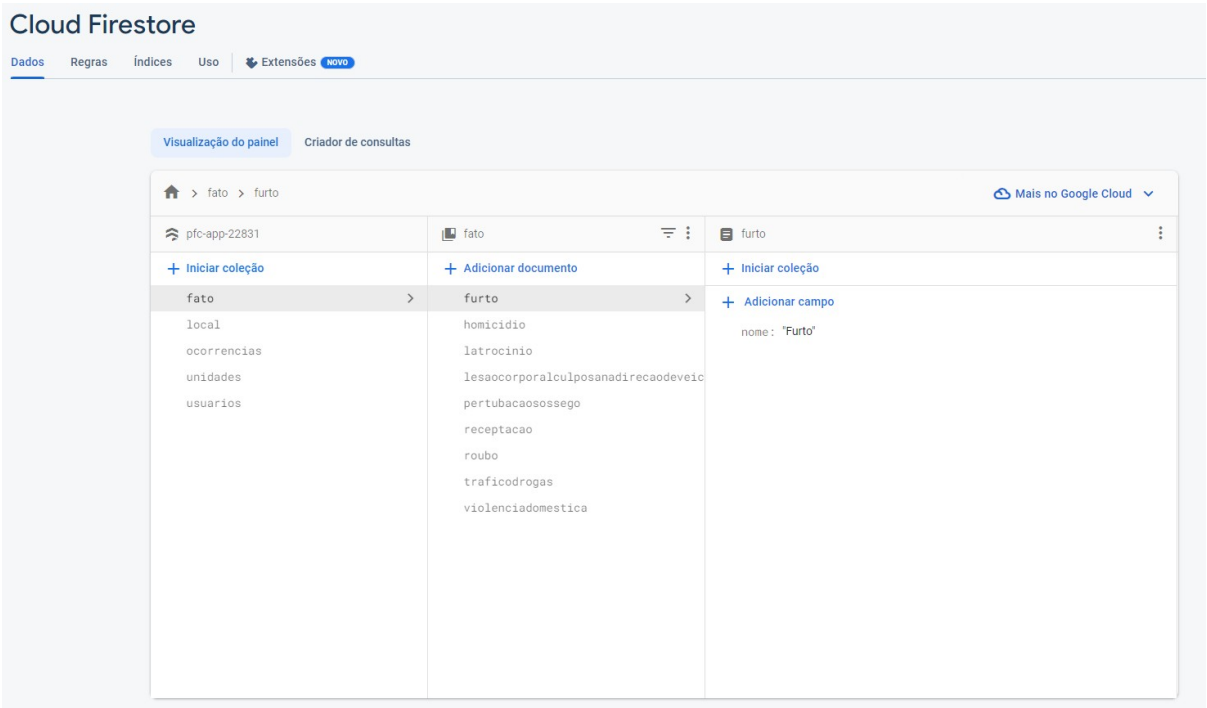

Figura 5: Interface do Cloud Firestore

4. OPEN STREET MAPS

O OpenStreetMap (OSM) é um projeto colaborativo e de código aberto que visa criar e fornecer dados geográficos livres e precisos para uso em diversos aplicativos e serviços. Ele utiliza uma abordagem comunitária, onde voluntários de todo o mundo contribuem com informações geográficas, como estradas, edifícios, pontos de interesse e muito mais (Wiki Open Street Maps, 2023).

Uma das principais vantagens do OpenStreetMap é a sua natureza aberta e acessível. Ao contrário de outras soluções de mapeamento, como por exemplo o google maps, que podem ser restritas ou sujeitas a licenças e restrições de uso, o OSM permite que qualquer pessoa use, compartilhe e melhore os dados disponíveis. Isso resulta em um conjunto de informações geográficas vasto e atualizado, que pode ser utilizado livremente por desenvolvedores e usuários.

No contexto do projeto em questão, o OpenStreetMap está sendo utilizado como a base de mapas para a visualização dos índices de criminalidade. Sua integração permite exibir de forma interativa e personalizada os dados de criminalidade em um mapa, utilizando recursos como marcadores, camadas e ferramentas de zoom.

Outra característica do OpenStreetMap é de oferecer uma API (Interface de Programação de Aplicativos) que permite aos desenvolvedores obter dados geográficos e realizar consultas avançadas, como busca de endereços, cálculo de rotas e muito mais. Essa flexibilidade e facilidade de acesso aos dados geográficos tornam o OpenStreetMap uma escolha popular para projetos que envolvem visualização de mapas e informações espaciais.

Na Figura 6 é visualizado um mapa detalhado da cidade de Dourados, disponibilizado através do OpenStreetMap:

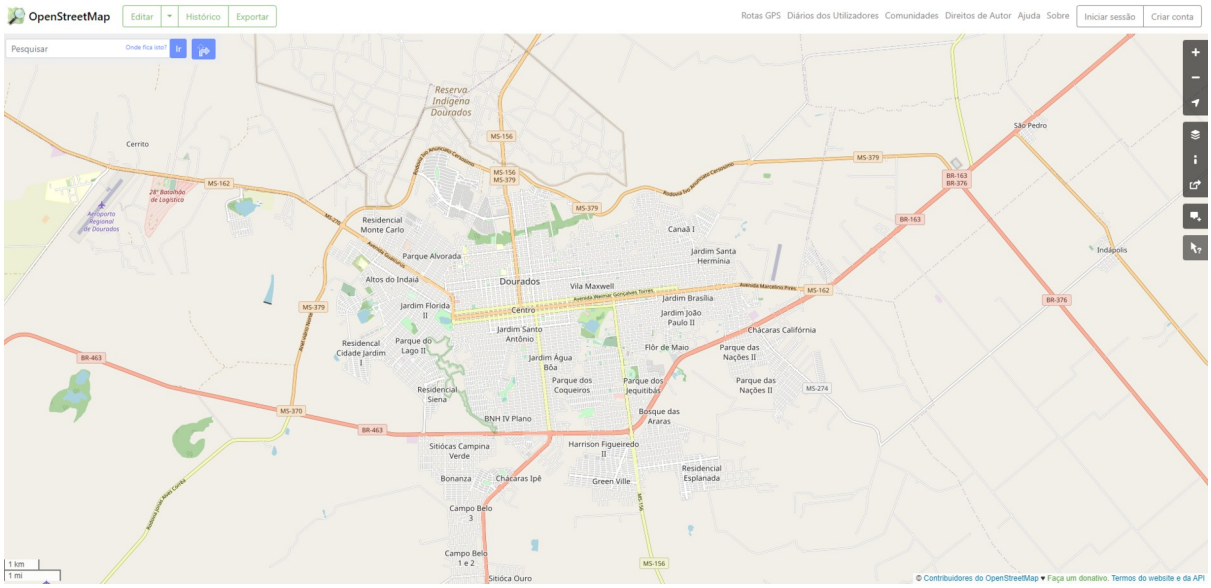

Figura 6: Página inicial do OpenStreetMap

5. LEAFLET

A biblioteca JavaScript Leaflet é uma poderosa ferramenta para a criação de mapas interativos em aplicações web. Ela oferece uma abordagem simples e flexível para a exibição de mapas, permitindo a incorporação de camadas de informação geográfica e interação com o usuário (Leaflet, 2023).

Desenvolvida como uma biblioteca de código aberto, a Leaflet possui uma ampla comunidade de usuários e uma extensa documentação, o que facilita o aprendizado e a adoção dessa tecnologia. Ela é amplamente utilizado em diversas aplicações web que exigem a exibição de informações geográficas, desde mapas simples até aplicações complexas de geolocalização.

Uma das principais características da Leaflet é a sua leveza e eficiência. Com um tamanho compacto, ela é rápida de carregar e possui um desempenho excepcional, mesmo em dispositivos móveis. Isso torna a Leaflet uma escolha popular para o desenvolvimento de aplicações web responsivas, que podem ser acessadas em diferentes dispositivos e navegadores.

A Leaflet oferece uma ampla gama de recursos, como a exibição de mapas em várias projeções cartográficas, suporte a camadas de azulejos (tiles) para exibição de mapas base, adição de marcadores, polígonos e linhas, suporte a eventos de interação com o usuário, como cliques e arrastar, além de controles de zoom e navegação.

Além disso, a Leaflet possui uma arquitetura modular que permite a extensão e personalização de funcionalidades através do uso de plugins. Existem inúmeros plugins disponíveis que adicionam recursos avançados a Leaflet, como visualização de dados geográficos em formatos específicos, ferramentas de roteamento, integração com serviços de geocodificação e muito mais.

Na Figura 7 é visualizado o resultado da utilização da Leaflet na aplicação através dos círculos vermelhos sobre o mapa:

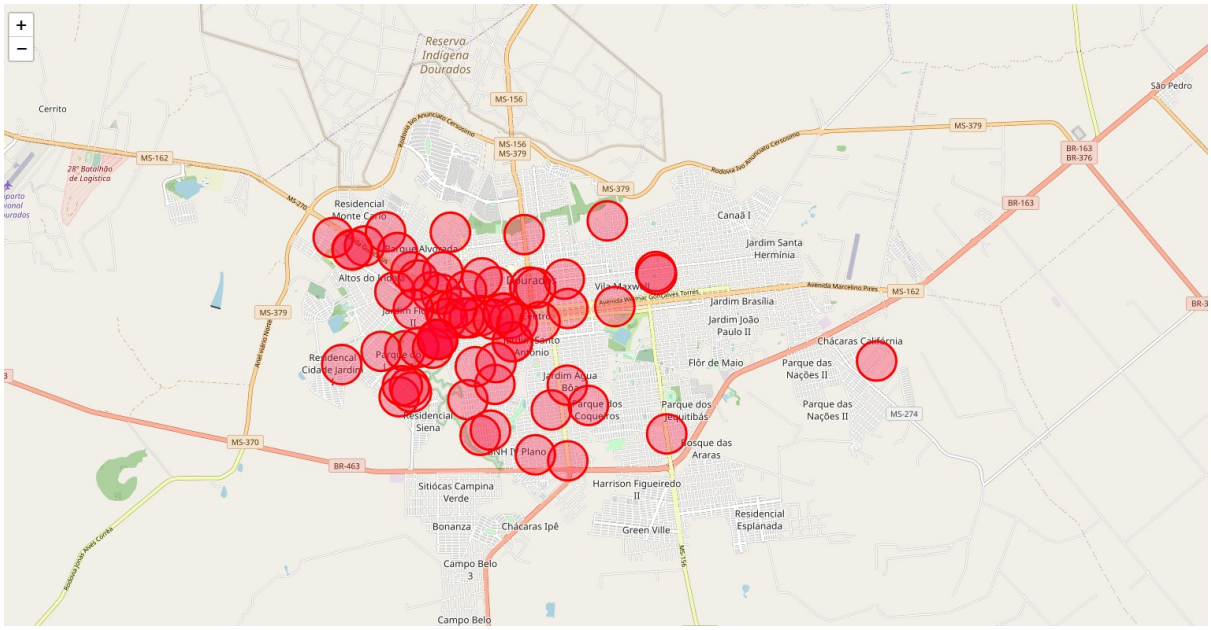

Figura 7: Mapa da aplicação utilizando a Leaflet e OpenStreetMap

6. INTERFACE DO USUÁRIO

A interface do usuário da aplicação mostra visualmente os índices de criminalidade através de círculos no mapa e tem como objetivo ter um design intuitivo, amigável, responsivo e acessível, visando fornecer uma experiência eficiente aos usuários.

Ao acessar a aplicação, os usuários serão apresentados a um mapa interativo, onde os círculos representam os índices de criminalidade em diferentes áreas geográficas. Esses círculos podem ser coloridos ou dimensionados proporcionalmente à intensidade da criminalidade em cada região.

Abaixo da barra de filtros, é possível visualizar a informação do total de ocorrências, que mostrará a quantidade total de ocorrências atualmente sendo exibidas. Essa quantidade pode ser alterada de acordo com as preferências de filtragem personalizadas pelo usuário.

Na Figura 8 é exemplificado a primeira página da aplicação ao ser acessado pelo usuário sem autenticação:

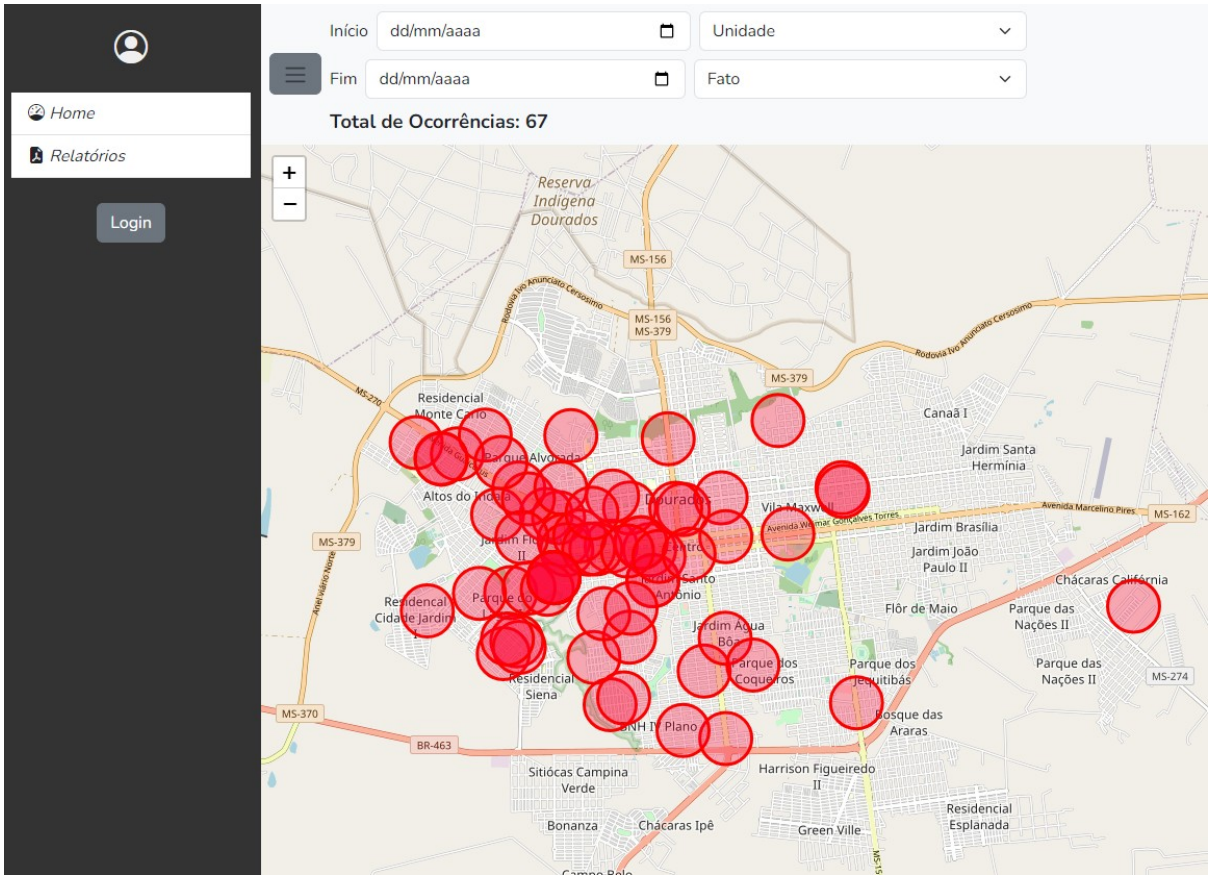

Figura 8: Página inicial com usuário sem autenticação

6.1. FILTRAR DADOS

No topo da interface, uma barra de opções de filtros permite aos usuários personalizarem a visualização dos índices de criminalidade. Os filtros podem incluir a seleção de um período específico, com a data de início e fim, bem como a escolha de unidades ou tipos de crime específicos para exibir ou ocultar. Esses filtros permitem que os usuários adaptem a visualização de acordo com suas necessidades e interesses, conforme visualizado na Figura 9.

A barra lateral, localizada ao lado esquerdo do mapa, é uma parte essencial da interface. Ela pode conter várias seções, como opções de login, relatórios e outras funcionalidades adicionais que podem ser inseridas futuramente. As opções de login permitem que os usuários se autentiquem para acessar recursos adicionais da aplicação.

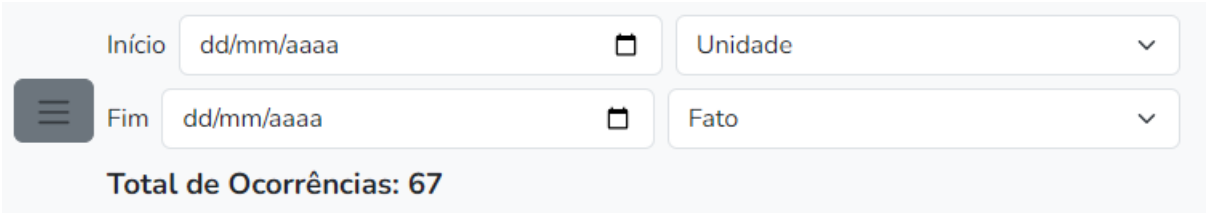

Figura 9: Barra de filtro

6.2. LOGIN

O login do usuário é um componente crucial em muitas aplicações, pois desempenha o papel de autenticar e verificar a identidade do usuário. Por meio desse processo, é possível garantir que apenas usuários autorizados tenham acesso aos recursos e funcionalidades personalizados da aplicação.

A interface de login, visualizado na Figura 10, consiste em campos de entrada para o e-mail de usuário e senha. Os usuários devem inserir suas credenciais corretas para realizar a autenticação e acessar a área restrita da aplicação, e caso seja necessário redefinir uma senha só será possível através da solicitação a um administrador.

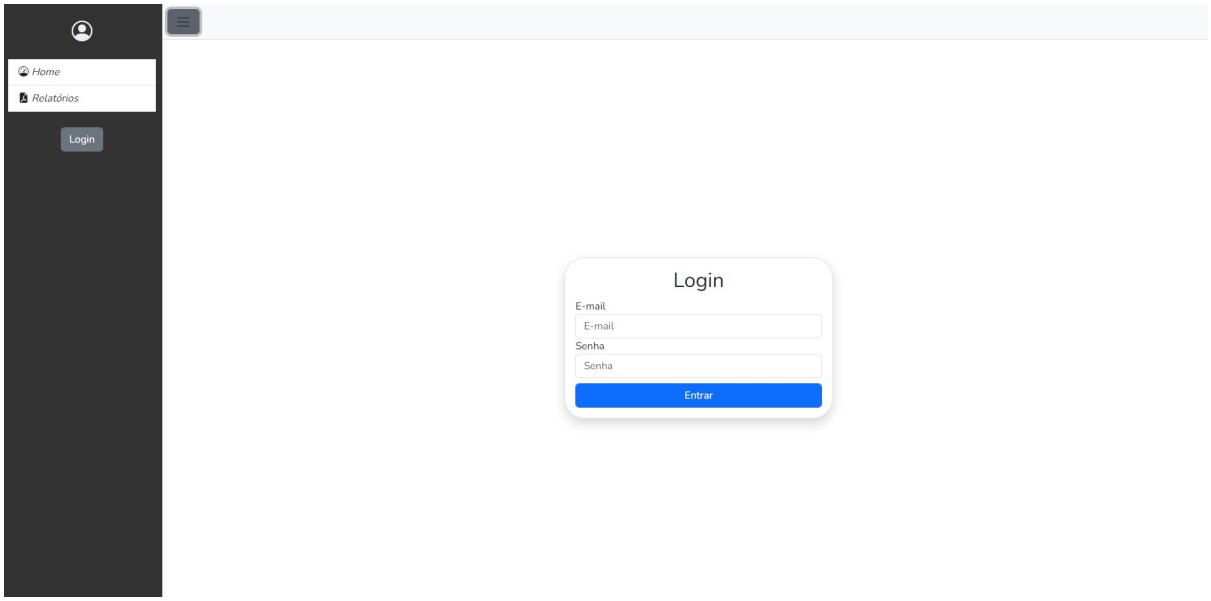

Figura 10: Página de login da aplicação

Na Figura 11 é visualizado como é a interface da aplicação de um usuário que se autenticou com o perfil user:

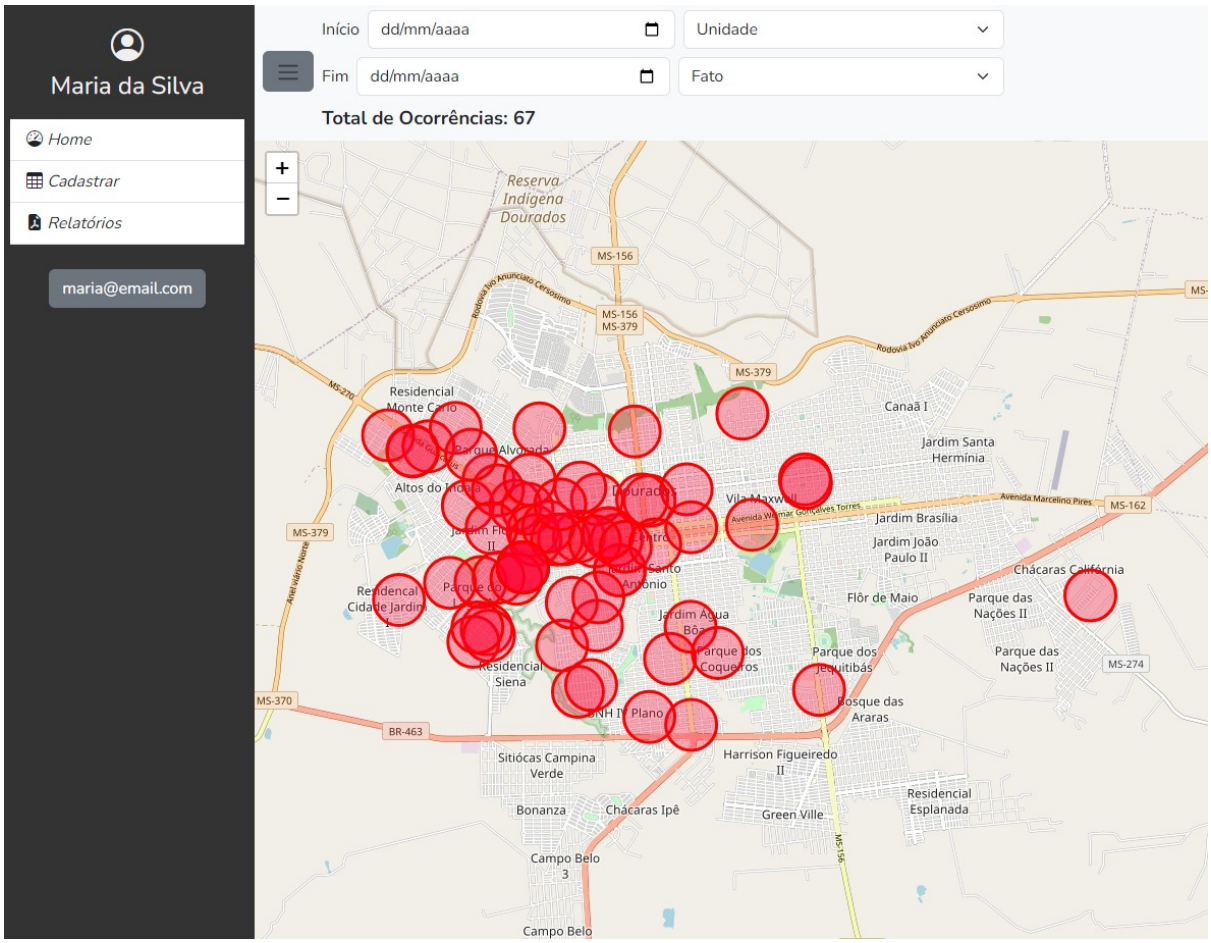

Figura 11: Página inicial com usuário autenticado com perfil user

6.3. CADASTRAR DADOS

Quando um usuário está autenticado, a barra lateral pode exibir funções adicionais, como a capacidade de cadastrar ocorrências. Esse recurso permite que os usuários insiram as ocorrências diretamente na aplicação, fornecendo detalhes relevantes, como data, hora, unidade, tipo de crime e localização, conforme visualizado na Figura 12. Essas informações podem ser usadas para atualizar os dados e os círculos no mapa, fornecendo uma visão em tempo real das ocorrências.

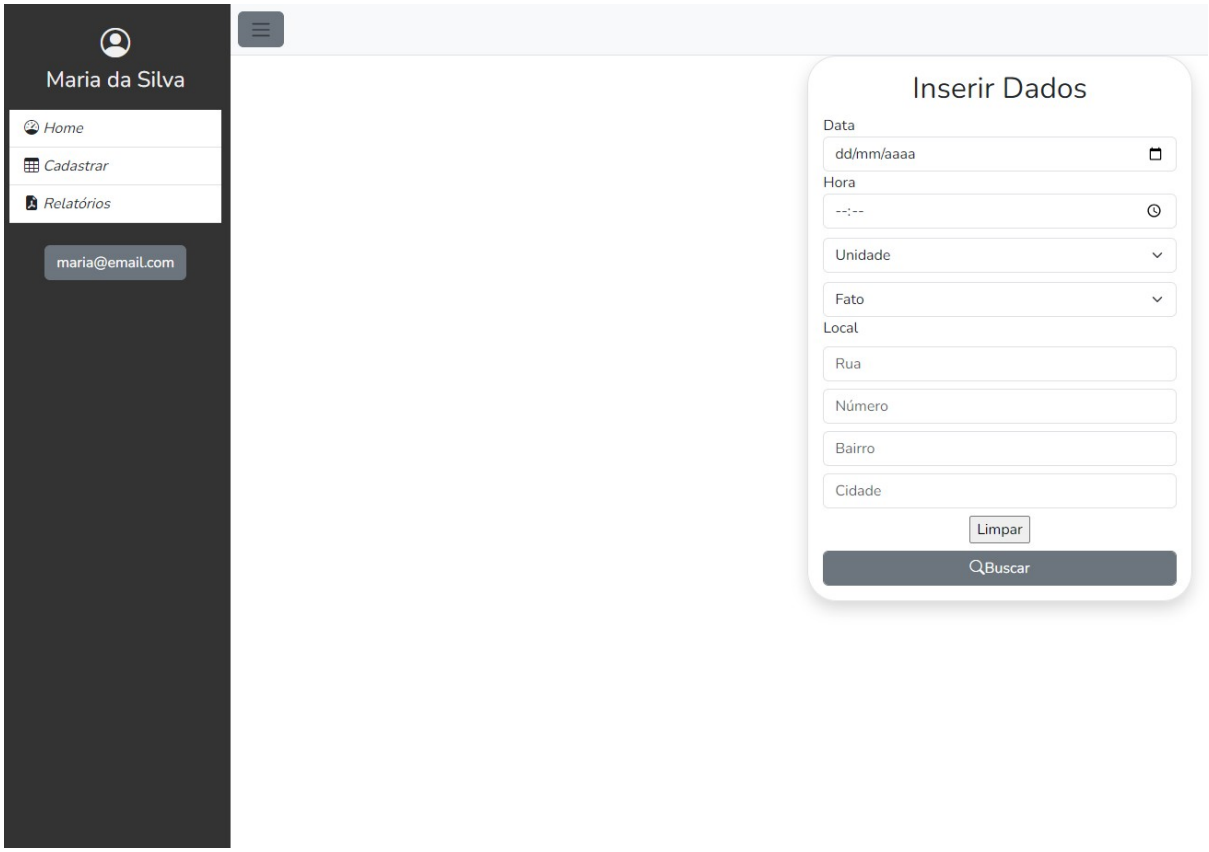

Figura 12: Página de cadastro de ocorrências

6.3.1. API NOMINATIM

A API Nominatim é um serviço de geocodificação e pesquisa de endereços oferecido pelo OpenStreetMap. Ela permite converter informações de localização, como endereços ou nomes de lugares, em coordenadas geográficas (latitude e longitude) e vice-versa. Através da API Nominatim, é possível realizar consultas de busca por endereços, obter detalhes de localização, como nomes de ruas, bairros e cidades, e até mesmo encontrar pontos de interesse próximos a determinadas coordenadas (API nominatim, 2023).

Esta API foi utilizada para realizar a busca de endereço no formulário de cadastro de ocorrência, desempenhando um papel fundamental ao permitir que os usuários informem um endereço para que seja convertido em coordenadas geográficas precisas que serão armazenadas no banco de dados, conforme demonstrado no Código 2:

```
const link = `https://nominatim.openstreetmap.org/search?addressdetails=1&q=${rua},$
{numero},${bairro},${cidade}&format=jsonv2`;

    const data = await fetch(link) /* Armazena os dados do endereço consultado */
        .then(response => response.json())
        .catch((error) => {
            console.log("Erro: ", error);
        });
```

Código 2: Exemplo de utilização da API nominatim

Sua escolha se deu por ser amplamente utilizada em projetos que requerem recursos de geocodificação e pesquisa de endereços, e sua integração com o OpenStreetMap oferece uma fonte de dados rica e atualizada para essas funcionalidades.

6.4. CADASTRAR NOVO USUÁRIO

Também há a possibilidade de um usuário estar autenticado com um perfil de administrador, onde na barra lateral irá aparecer mais uma função adicional, a de “Novo Usuário”, conforme visualizado na Figura 13. Essa função permite que os administradores cadastrem novos usuários pela interface da aplicação, conforme visualizado na Figura 14.

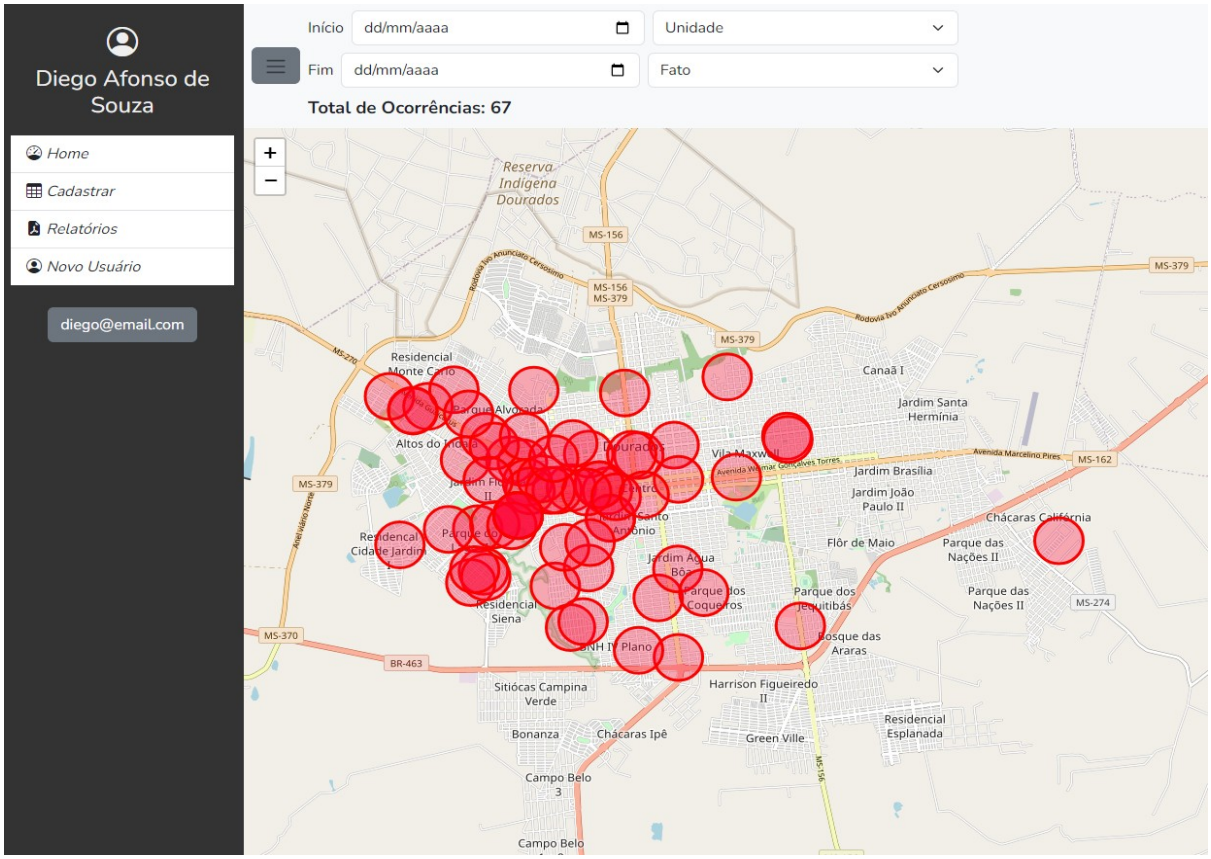

Figura 13: Página inicial com usuário logado como perfil adm

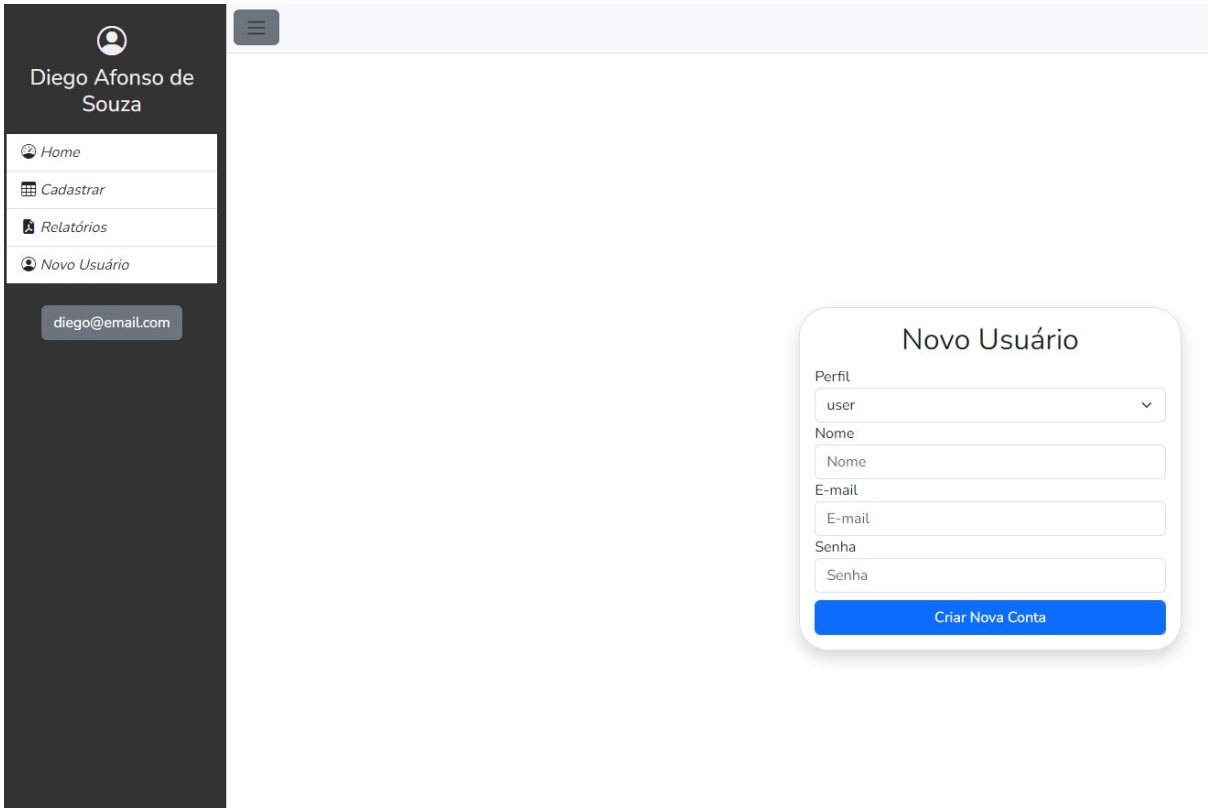

Figura 14: Página para cadastrar novo usuário

7. CONCLUSÃO

Diante da evolução tecnológica e da relevância do tema da criminalidade, este trabalho teve como objetivo explorar a metodologia de Progressive Web Apps (PWAs) e suas características para o desenvolvimento de uma aplicação voltada para a Polícia Militar do Estado de Mato Grosso do Sul. O foco principal foi auxiliar os operadores de segurança pública na tomada de decisões durante suas atividades, por meio da visualização gráfica de índices de criminalidade na área da 9ª CIPM (região oeste da cidade de Dourados).

Ao longo deste estudo, foi possível compreender o potencial das PWAs em fornecer uma experiência de utilização semelhante à de aplicativos nativos de dispositivos móveis, superando as limitações das aplicações web tradicionais. Através da implementação da aplicação, que utiliza recursos como o OpenStreetMap para exibir os índices de criminalidade através de círculos no mapa, foi possível criar uma ferramenta visual e intuitiva para auxiliar os operadores no monitoramento e análise dos dados de segurança.

A utilização da metodologia de PWA permitiu que não houvesse a necessidade de instalação através de uma loja de aplicativos, além de possuir compatibilidade com diferentes plataformas, garantindo uma maior acessibilidade. Além disso, a integração com serviços como o Firebase Authentication e o Firebase Cloud Firestore proporcionou recursos de autenticação de usuário e armazenamento de dados eficientes, contribuindo para o bom funcionamento da aplicação.

É importante ressaltar que este trabalho representa apenas o ponto de partida no desenvolvimento dessa ferramenta, inicialmente sendo utilizado para testes na área de atuação da 9ªCIPM, que abrange a região oeste de Dourados. A partir dos resultados obtidos, surgem novas oportunidades para aprimorar e expandir o sistema, contemplando outras regiões e aperfeiçoando os recursos disponíveis.

Por fim, este estudo contribui para o avanço da aplicação de tecnologias inovadoras no campo da segurança pública, fornecendo uma abordagem moderna e eficaz para o monitoramento e análise de dados de criminalidade. Através da utilização de PWAs e recursos como o OpenStreetMap, Firebase Authentication e Firebase Cloud Firestore, foi possível criar uma aplicação funcional que pode ser uma ferramenta valiosa para auxiliar os profissionais de segurança no combate à criminalidade e na tomada de decisões estratégicas, além de servir a população com essas informações.

7.1. TRABALHOS FUTUROS

Para trabalhos futuros, algumas melhorias e expansões podem ser consideradas para aprimorar ainda mais a aplicação desenvolvida, algumas sugestões são:

- Implementação de geração de relatórios estatísticos: Uma adição importante seria implementar a funcionalidade de gerar relatórios estatísticos em formatos populares, como Excel ou PDF. Isso permitiria aos usuários obter uma visão mais abrangente e analítica dos índices de criminalidade.
- Inclusão de filtro de cidade: Adição de um filtro para permitir aos usuários selecionar a cidade que desejam visualizar os índices de criminalidade. Isso ampliaria a utilidade da aplicação, possibilitando a expansão para outras regiões e cidades, tornando-a mais flexível e adaptável a diferentes contextos geográficos.
- Aperfeiçoamento da interface do usuário: Sempre há espaço para melhorias na interface do usuário, visando torná-la mais intuitiva, amigável e responsiva. Isso pode incluir aprimoramentos na disposição dos elementos, organização das informações, melhorias visuais e a implementação de recursos interativos para facilitar a navegação e a interação dos usuários com a aplicação.
- Integração com outras fontes de dados: A aplicação poderia ser aprimorada com a integração de outras fontes de dados relevantes para enriquecer as informações apresentadas. Isso poderia incluir a incorporação de dados demográficos, informações sobre infraestrutura e outros fatores que possam influenciar os índices de criminalidade.

Referências Bibliográficas

API Nominatim, 2023. Disponível em: https://nominatim.org. Acesso em: 19 jun 2023.

Biørn-Hansen, Andreas & Majchrzak, Tim A. & Grønli, Tor-Morten. (2017). Progressive Web Apps: The Possible Web-native Unifier for Mobile Development. 344-351. 10.5220/0006353703440351. Disponível em: https://www.scitepress.org/Papers/2017/63537/63537.pdf. Acesso em: 19 jun 2023.

GOOGLE: Firebase, 2021. Disponível em: https://firebase.google.com/docs. Acesso em: 19 jun 2023.

IBGE: Acesso à Internet e à televisão e posse de telefone móvel celular para uso pessoal, 2021. Disponível em: https://biblioteca.ibge.gov.br/visualizacao/livros/liv101963_informativo.pdf. Acesso em: 02 ago. 2023.

Leaflet, 2023. Disponível em: https://leafletjs.com/. Acesso em: 19 jun 2023.

MADEINWEB: Aplicativo Nativo X Aplicativo Híbrido: quais as diferenças?, 2019. Disponível em: https://www.madeinweb.com.br/aplicativo-nativo-aplicativo-hibrido-quais-as-diferencas/. Acesso em: 19 jun 2023.

MDN Web Docs: Introduction to progressive web apps, 2021. Disponível em: https://developer.mozilla.org/en-US/docs/Web/Progressive_web_apps/Introduction. Acesso em: 19 jun 2023.

MDN Web Docs: Service Worker API, 2021. Disponível em: https://developer.mozilla.org/en-US/docs/Web/API/Service_Worker_API. Acesso em: 19 jun 2023.

MDN Web Docs: ServiceWorkerContainer, 2021. Disponível em: https://developer.mozilla.org/en-US/docs/Web/API/ServiceWorkerContainer. Acesso em: 19 jun 2023.

MDN Web Docs: Web app manifests, 2021. Disponível em: https://developer.mozilla.org/en-US/docs/Web/Manifest. Acesso em: 19 jun 2023.

MDN Web Docs: Web technology for developers, 2021. Disponível em: https://developer.mozilla.org/en-US/docs/Web. Acesso em: 19 jun 2023.

Web Dev, 2023. Disponível em: https://web.dev/progressive-web-apps/. Acesso em: 20 mar 2023.

Wiki Open Street Maps, 2023. Disponível em: https://wiki.openstreetmap.org. Acesso em: 19 jun 2023.