

# SISTEMA DE MONITORAMENTO DE RISCOS OU CATÁSTROFES POR RANKING DE USUÁRIOS

Cecília Chinaglia Kliemke, Gustavo Grilla Barroso

Prof. Dr. Mauricio Caldora

Departamento de Engenharia da Computação -

Universidade Anhembi Morumbi (UAM)

**Resumo.** Considerando os riscos eminentes que as grandes cidades apresentam, tais como violência, trânsito, acidentes, entre outros, percebeu-se a necessidade de uma rede de apoio, onde os próprios cidadãos possam se ajudar. O projeto tem como objetivo a criação de um aplicativo para reports que serão feitos pelos usuários que testemunharem um ocorrido e o considerarem impactante para o bom funcionamento da cidade, assim alertando outros usuários próximos e, caso seja necessário, enviando um alerta para as autoridades responsáveis. O aplicativo contará com recurso para avaliação dos usuários, garantindo a veracidade do acontecimento reportado. A partir deste recurso, será criado um ranking entre os usuários.

**Summary.** Considering the imminent risks that cities present, such as violence, traffic, accidents, among others, the need for a support network was perceived, where citizens can help themselves. The project aims to create an application for reports that will be made by users who witness an event and consider it impactful for the proper functioning of the city, then, alerting other nearby users and, if necessary, sending an alert to the responsible authorities. The application will have a resource for user evaluation, then ensuring the veracity of the reported event. From this feature, a ranking will be created between users.

## 1. Introdução

Morar em uma grande cidade possui muitas vantagens, pois inúmeros eventos ocorrem nelas, proporcionando grandes oportunidades de trabalho, opções de lazer, opções de educação e moradia. Entretanto, as grandes cidades podem apresentar, da mesma maneira, inúmeros riscos, entre elas: violência; poluição; trânsito; tumulto; estresse; entre outros.

As Cidades Inteligentes, também chamadas “Smart Cities”, têm como um de seus pilares proporcionar qualidade de vida através da eficiência e conectividade entre seus

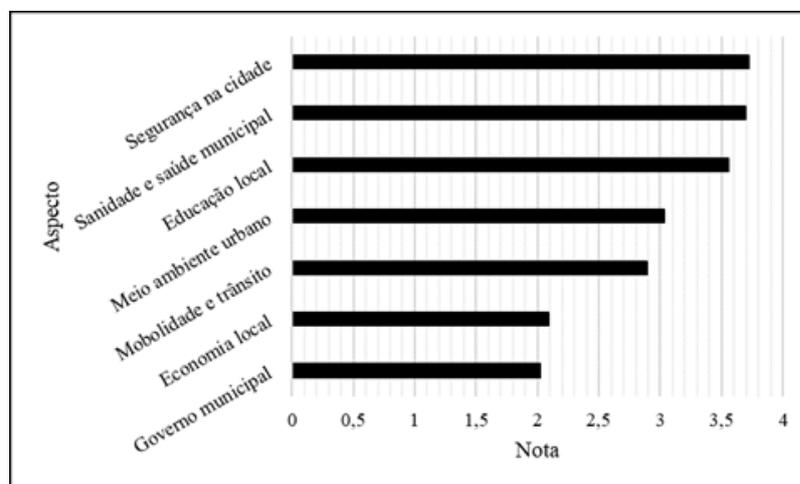
moradores e seus serviços. Dessa forma, acredita-se que é possível tornar uma cidade mais segura utilizando-se dos princípios das Smart Cities.

Este projeto tem como principal finalidade auxiliar os cidadãos que habitam em metrópoles e utilizam rotas perigosas, isto é, expostas a riscos provenientes de crimes e catástrofes, através de aplicativo mobile que proporcione eficiência na comunicação entre a sociedade civil e autoridades responsáveis por manter a ordem, na intenção de tornar o cotidiano das pessoas mais seguro.

O sistema tem como princípio ser algo extremamente intuitivo e acessível a todos pois, quanto maior o número de usuários, maior seria a eficiência, atrelada ao ranking de confiabilidade dos usuários, que possui o objetivo de garantir a autenticidade do report, visto que ele funciona a partir da contribuição dos usuários, que serão avaliados com uma nota de 0,0 a 5,0, que será dada a partir da integridade da informação e do usuário.

## 2. Metodologia

A segurança é o aspecto visto como o mais importante para os cidadãos (3,72), seguida de saúde (3,69) e educação (3,55) (Figura 1), sendo a escala de 0 (nada importante) a 4 (muito importante).



**Figura 1 – Quais as notas que você dá aos aspectos da gestão da sua cidade? Escala de 0 (nada importante) a 4 (muito importante).**

**Fonte: Estudo Smart Brazil Citizens, 2015**

Destaca-se a importância imputada à segurança, para a qual os cidadãos atribuem maior preocupação.

As Smart Cities têm como objetivo, no aspecto de segurança, proporcionar aos cidadãos espaços seguros e protegê-los de ameaças, além de ser capaz de reagir rapidamente a emergências.

Com o objetivo de auxiliar as grandes cidades a alcançar o cenário ideal de segurança, decidiu-se que neste trabalho será desenvolvido um aplicativo para report rápido de catástrofes às quais as cidades estão expostas a sofrer.

No contexto deste aplicativo dois conceitos são fundamentais: ‘catástrofe’ e ‘risco’. A ‘catástrofe’ refere-se a eventos ou situações de grande impacto e severidade que podem causar danos significativos à segurança, à infraestrutura ou à vida das pessoas em uma cidade. Exemplos de catástrofes incluem desastres naturais, como: inundações e tempestades severas ou incidentes humanos graves.

Por outro lado, ‘risco’ é um termo mais amplo que se refere a qualquer situação potencialmente perigosa, que pode variar desde pequenos incidentes até grandes catástrofes. O risco pode ser previsível e gerenciável, enquanto uma catástrofe é geralmente um evento de grande escala e imprevisível.

Partindo da identificação de um problema que pode ser previsível ou não; de ordem natural ou provocada pela intervenção do ser humano busca-se nesse projeto uma solução com a criação do aplicativo “SEG”, inspirado na palavra “segurança”. O aplicativo SEG foi desenvolvido com a intenção de facilitar a comunicação entre a população das metrópoles, com objetivo de auxiliar os usuários a evitarem situações de risco o máximo possível, através de alertas originados dos próprios usuários. A intenção é que o cidadão adquira ciência dos problemas que ocorrem na cidade o mais rapidamente possível, para que o problema seja evitado e os danos sejam reduzidos ao mínimo.

Após analisar situação problema/solução o aplicativo SEG foi desenvolvido com base no framework Flutter, que é um conjunto de códigos genéricos permitindo o desenvolvimento de sistemas e aplicações, funcionando como um modelo. O Flutter é um framework de código aberto desenvolvido pela empresa Google. Atualmente, oferece suporte ao desenvolvimento de aplicações em seis plataformas: iOS, Android, Web, Windows, MacOS e Linux.

Para o desenvolvimento, utilizou-se a linguagem de programação Dart, também desenvolvida pela Google em 2011. Esta linguagem é otimizada para o desenvolvimento de interfaces de usuário e permite criar aplicações rápidas em qualquer plataforma.

Oferece mais produtividade durante o desenvolvimento multiplataforma, combinada com uma flexibilidade que possibilita a criação tanto de aplicações web quanto de aplicativos mobile com o Flutter.

Para tanto, foi empregado o Firebase, uma plataforma digital que facilita o desenvolvimento de aplicativos web ou móveis de maneira efetiva, rápida e simples. Essa plataforma oferece facilidades em serviços como autenticação de usuários, armazenamento de dados e mídia na nuvem, laboratório de testes, entre outros recursos.

O aplicativo é compatível com equipamentos que possuam sistema operacional Windows, MacOS e Linux, bem como com dispositivos móveis (celulares e tablets) que utilizem os sistemas operacionais Android e iOS em suas versões mais recentes. A intenção é proporcionar uma experiência do usuário intuitiva, permitindo a rápida realização de relatórios, considerando a urgência de cada situação.

A utilização do Flutter e Firebase oferece diversas vantagens em comparação com outras ferramentas de desenvolvimento web, proporcionando uma abordagem eficiente e moderna para a criação de aplicativos multiplataforma.

O Firebase é uma plataforma de desenvolvimento de aplicativos desenvolvida pela Google que oferece uma variedade de serviços para facilitar o desenvolvimento, o crescimento e a manutenção de aplicativos para dispositivos móveis e web. Ele inclui uma gama de ferramentas e serviços que abrangem desde o armazenamento de dados até a autenticação de usuários e a análise de aplicativos.

Foram adicionadas as dependências necessárias do Firebase ao arquivo 'pubspec.yaml' no projeto. Este arquivo é usado para definir as dependências do mesmo, configurar metadados e fornecer informações sobre o projeto Flutter em si. São essas dependências que permitirão que o aplicativo se conecte ao Firebase.

No console do Firebase, é possível ter acesso a diversas visões e métricas relacionadas à utilização do aplicativo e de seus recursos. Conforme ilustrado na Figura

2, é viável consultar, por exemplo, a quantidade de gravações de dados no banco referente ao último mês.

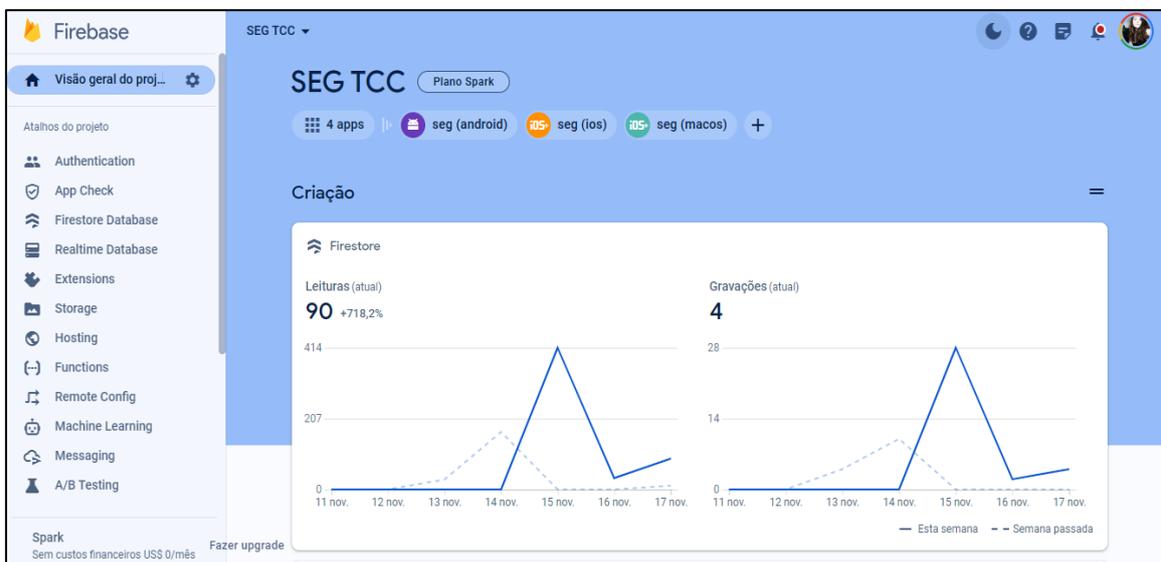


Figura 2 - Console do Firebase.

#### a. Firestore Database e Storage

O *Firestore Database* é um banco de dados hospedado na nuvem. Os dados são armazenados como JSON e sincronizados em tempo real para cada cliente conectado. O *Firestore Database* é um banco de dados *NoSQL* e, por isso, tem otimizações e funcionalidades diferentes de um banco de dados relacional. Em vez da estrutura tabular típica de um banco de dados relacional, os bancos de dados *NoSQL* armazenam dados em uma estrutura de dados. Como esse design de banco de dados não relacional não requer um esquema, ele oferece escalabilidade rápida para gerenciar conjuntos de dados grandes e normalmente não estruturados.

Já o *Storage* para *Firestore* foi criado com base na infraestrutura rápida e segura do *Google Cloud* para desenvolvedores de apps que precisam armazenar e disponibilizar conteúdo gerado pelo usuário, como fotos ou vídeos.

Utilizou-se ambos os recursos para armazenamento dos dados referentes aos *reports* enviados pelos usuários, como: descrição, tipo de incidente e localização. O *Storage* será utilizado para armazenar em nuvem as fotos que os usuários poderão realizar *upload* junto aos reports.

## **b. Firebase Authentication**

O Firebase Authentication é um serviço do Firebase que permite aos desenvolvedores de aplicativos autenticar os usuários com diferentes métodos, como e-mail e senha, números de telefone, provedores de identidade federados (como Google, Facebook e Twitter) e outros. O Firebase Authentication armazena os dados dos usuários na nuvem com criptografia e segurança, protegendo-os de ataques maliciosos.

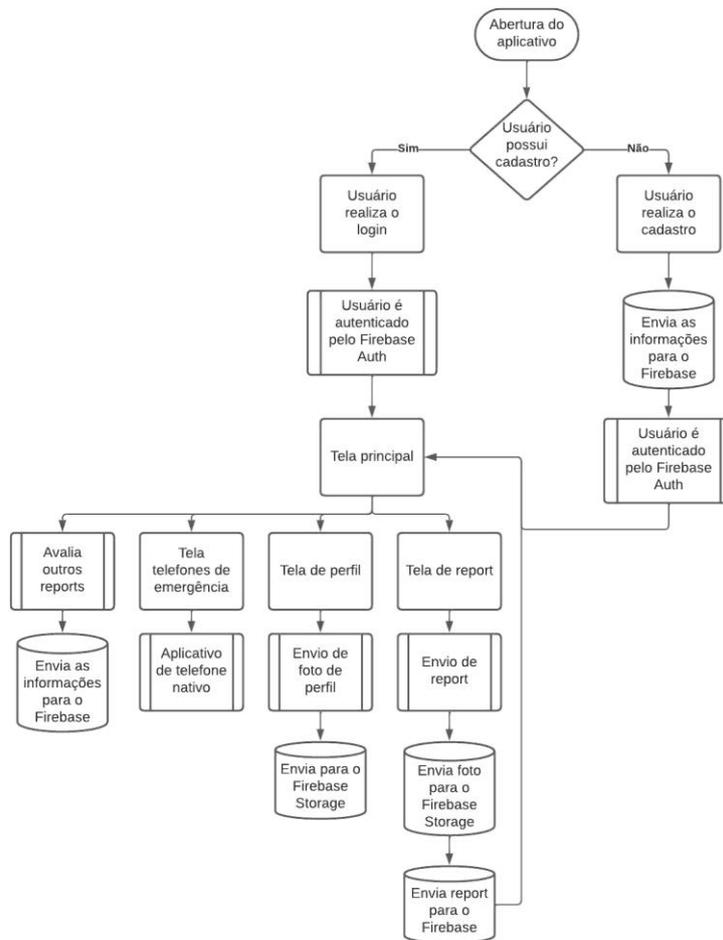
## **c. IDE - Integrated Development Environment**

Foi utilizado o Android Studio, o ambiente de desenvolvimento integrado (IDE) oficial para o desenvolvimento de aplicativos Android. O referido ambiente foi anunciado em 16 de maio de 2013 durante a conferência Google I/O. O Android Studio fundamenta-se no IntelliJ IDEA, um software desenvolvido pela JetBrains, sendo concebido especificamente para o desenvolvimento destinado à plataforma Android.

## **3. Desenvolvimento do aplicativo**

Neste tópico, serão abordados detalhadamente os recursos de cada tela do aplicativo e suas funções, bem como as bibliotecas e recursos utilizados. Todas as telas foram planejadas de maneira a se apresentarem visualmente intuitivas e com usabilidade simplificada, considerando a amplitude do público-alvo.

Na Figura 3 é representado, em forma de fluxograma, o funcionamento do aplicativo integrado com o Firebase. O fluxograma ilustra a interação entre um aplicativo e o Firebase, ressaltando os passos cruciais do processo. Primeiramente, o aplicativo começa a comunicação estabelecendo um vínculo com o Firebase. Posteriormente, o fluxograma mostra a troca de dados entre o aplicativo e o banco de dados do Firebase, onde atividades como autenticação de usuários, armazenagem de dados e administração de arquivos podem ser realizadas.



**Figura 3 - Fluxograma: envio de informações para o Firebase.**

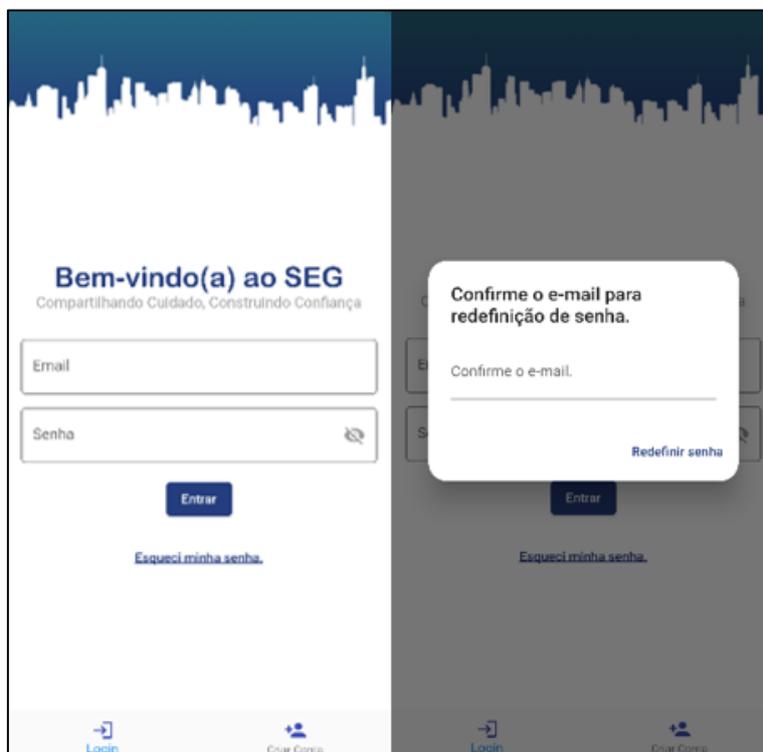
#### 4. Tela de login

A tela de login, conforme Figura 4, do aplicativo apresenta campos para e-mail e senha, permitindo que o usuário autentique-se utilizando as informações registradas durante o cadastro. Esta tela é a primeira a ser exibida ao usuário após a instalação e abertura do aplicativo. Além disso, a tela inclui uma barra de navegação inferior para facilitar o acesso à tela de cadastro.

Na tela de login, também é disponibilizada o recurso "Esqueci minha senha", que permite ao usuário alterar a senha de autenticação em caso de esquecimento. Ao selecionar essa opção, o usuário deve fornecer o e-mail registrado, resultando no envio de um e-mail para redefinição da senha ao endereço informado.

Recomenda-se não realizar o logout do aplicativo após o uso, uma vez que o usuário poderá encontrar-se em uma situação de risco e/ou catástrofe aonde não haverá

tempo para utilizar este recurso, ademais, a funcionalidade de manter o usuário sempre logado foi incorporada para proporcionar maior agilidade. A menos que o usuário faça o logout ou reinstale o aplicativo, permanecerá autenticado. Isso significa que, uma vez conectado, o usuário poderá realizar tarefas e reportar qualquer incidente de forma rápida.



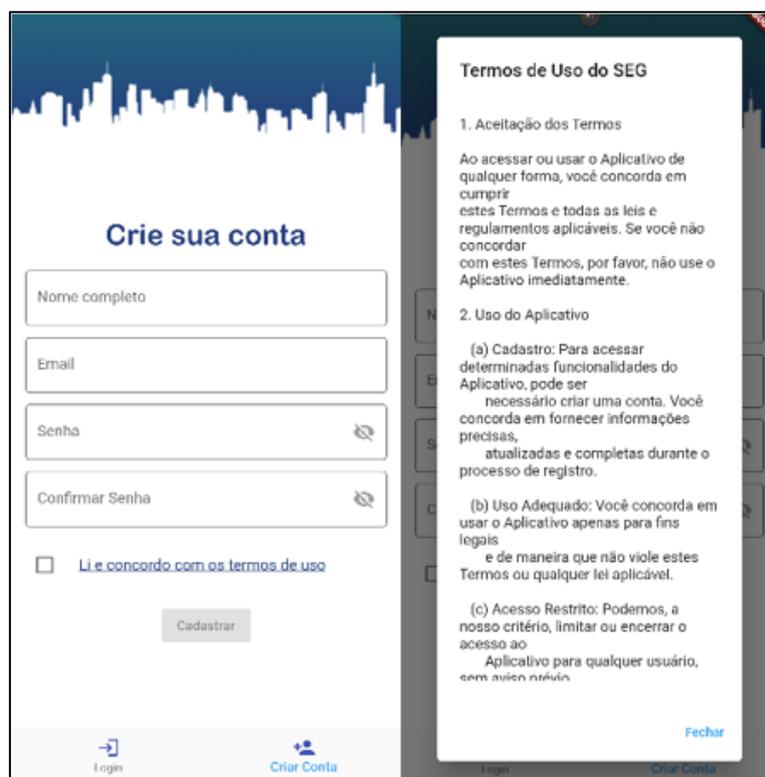
**Figura 4 - Tela de login.**

#### **4.1. Tela de cadastro**

Na tela de cadastro (Figura 5), o usuário tem a possibilidade de efetuar seu registro no aplicativo preenchendo um formulário que inclui campos para nome completo, e-mail, senha e confirmação de senha. É importante observar que as senhas devem ser idênticas e conter mais de 6 caracteres.

Para possibilitar a autenticação no aplicativo, utilizou-se o serviço do Firebase Authentication. Neste projeto, a autenticação foi implementada por meio de e-mail e senha.

Adicionalmente, na tela de cadastro, o usuário tem acesso aos termos de uso do aplicativo, sendo necessário concordar com esses termos para prosseguir com o cadastro no aplicativo. Nos termos constam condições acerca do uso do aplicativo, propriedade intelectual, privacidade, limitação de responsabilidade, alteração nos termos e lei aplicável.



**Figura 5 - Tela de cadastro.**

## **4.2. Tela principal**

Após efetuar a autenticação no aplicativo, o usuário será redirecionado para a tela principal, também conhecida como linha do tempo (time line). Nesta interface, o usuário terá a capacidade de visualizar os reports realizados por outras pessoas.

Conforme Figura 6, as informações serão apresentadas em formato de cards, exibindo o nome do usuário responsável pelo report e sua respectiva avaliação, o tipo de incidente, a descrição correspondente, uma possível foto associada (caso o usuário a tenha carregado), a localização, o horário do report e um campo que permite ao usuário avaliar o responsável pelo report através de um sistema de "estrelas", visando uma experiência mais intuitiva.

As publicações serão organizadas por tempo, ou seja, os reports que ocorreram recentemente terão prioridade para aparecer no topo da linha do tempo. Na tela principal, encontra-se um botão no canto inferior direito que direciona o usuário à tela de reports, possibilitando o envio de um novo report.



*Figura 6 - Tela principal.*

### **4.3. Tela de report**

Ao clicar no ícone "+" na tela principal, o usuário é redirecionado para a tela de Report, conforme Figura 7, onde poderá criar reports inserindo informações essenciais no campo de descrição. Além disso, poderá escolher um tipo de incidente e efetuar o carregamento de uma foto, seja através da câmera ou da galeria. Para a seleção de mídia, foi utilizada a biblioteca Image Picker, um plugin que oferece uma maneira facilitada de obter imagens e vídeos do rolo da câmera ou da galeria do dispositivo.

A localização será preenchida automaticamente com base na localização atual do usuário que está realizando o report. Para isso, foi utilizado a biblioteca Geolocator para Dart, um plugin desenvolvido pela Baseflow. Essa biblioteca proporciona uma API multiplataforma (iOS, Android) para funções genéricas de localização (GPS, etc.). Esse processo é realizado criando uma instância do Geolocator e chamando a função `getCurrentPosition`.

É aconselhável efetuar reports anonimamente, resultando na substituição do nome do usuário responsável pelo report pelo texto "Report anônimo" na tela principal. Este recurso foi implementado para assegurar que o usuário não seja identificado ao relatar informações sensíveis, como denúncias de assaltos, onde há a possibilidade de o agressor estar cadastrado no aplicativo.

Ao selecionar "Publicar", o usuário enviará as informações inseridas ao banco de dados, as quais serão posteriormente exibidas na tela principal para que todos os usuários possam visualizá-las e avaliá-las.

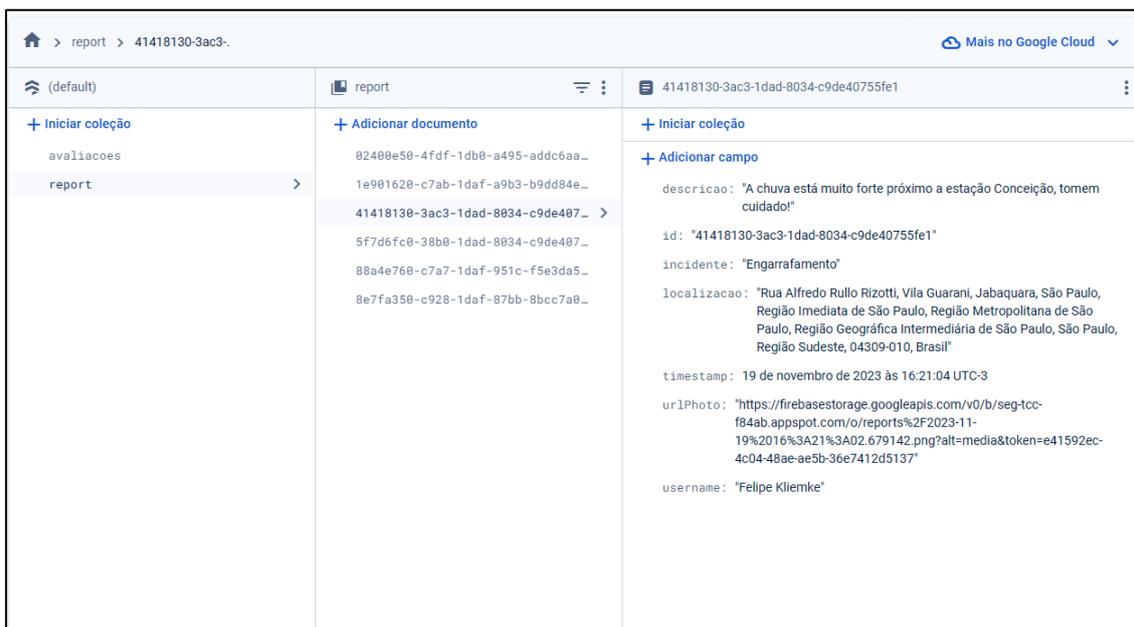


**Figura 7 - Tela de report.**

As informações que compreendem um report no banco de dados incluem:

- ID (identificador único) do report;
- Nome do usuário que realizou o report (em caso de reports não anônimos) e seu UID (identificador único);
- Descrição;
- Tipo de incidente selecionado;
- Geolocalização contendo latitude e longitude;
- URL da foto, que é gerada pelo Firebase Storage durante o envio de mídia.

Todos os reports podem ser consultados no Cloud Firestore, conforme Figura 8.



**Figura 8 - Console do Cloud Firestore.**

Após clicar em “Publicar” o usuário retornará automaticamente para a tela principal do aplicativo.

#### **4.4. Tela de perfil**

Na tela de perfil, é possível acessar algumas informações do usuário autenticado, incluindo o nome do usuário, o e-mail de cadastro, a foto de perfil e a nota no aplicativo, conforme Figura 9.

É nessa tela também que o usuário tem a capacidade de atualizar sua foto de perfil. Esse processo é realizado por meio do serviço do Firebase Storage e da biblioteca Image Picker, que opera de maneira semelhante à tela de report. A atualização da visualização da foto ocorre automaticamente após o upload da nova imagem.



**Figura 9 - Tela de perfil.**

As imagens enviadas para o Firebase Storage são armazenadas seguindo uma estrutura semelhante à do Cloud Firestore, organizadas em uma estrutura de pastas. O Storage está dividido em duas categorias principais, sendo uma denominada "profiles", destinada ao armazenamento das fotos de perfil, e a outra denominada "reports", responsável por armazenar as imagens carregadas nos reports, demonstrado na Figura 10.



**Figura 10 - Console Firebase Storage.**

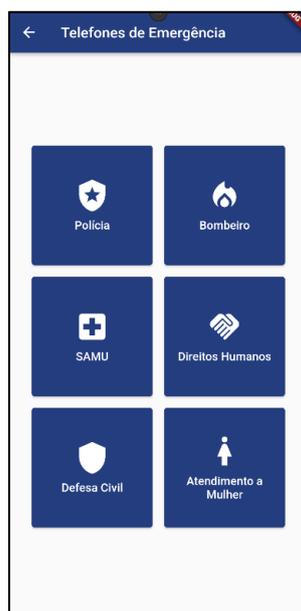
#### **4.5. Tela telefones de emergência**

Foi incorporada uma tela no aplicativo que permite ao usuário efetuar chamadas telefônicas para os principais órgãos de apoio. Ao clicar no ícone relacionado ao órgão desejado, conforme Figura 11, o usuário é redirecionado para o aplicativo de telefone nativo do dispositivo, com o número do órgão já discado. Nesse processo, é necessário apenas que o usuário confirme a ligação.

Foram disponibilizados os números dos seguintes órgãos:

- Polícia;
- Bombeiros;
- SAMU;
- Direitos Humanos;
- Defesa Civil;
- Atendimento à Mulher.

Para viabilizar essa funcionalidade, foi utilizada a biblioteca Url Launcher, um plugin que facilita o lançamento de URLs em diversas aplicações, incluindo navegadores da web, aplicativos de mapas, aplicativos de discagem, aplicativos de e-mail, entre outros.



*Figura 11 - Tela telefones de emergência.*

## **5. Melhorias futuras**

Considerando o contínuo aprimoramento do aplicativo SEG, vislumbram-se oportunidades significativas para a expansão de suas funcionalidades, alinhadas às diversas demandas dos usuários, duas melhorias fundamentais estão sendo contempladas para futuras implementações.

### **a. Reports em Outras Localizações:**

Aprimorar a capacidade do aplicativo para permitir que os usuários realizem reports em localizações distintas da atual. Isso proporcionará uma flexibilidade essencial, permitindo que incidentes ou riscos em outras áreas geográficas sejam prontamente comunicados. Essa funcionalidade é particularmente relevante para usuários que desejam relatar situações de risco em locais que não correspondem à sua posição atual, mas que foram testemunhadas anteriormente e, por algum motivo, como por exemplo falta de conexão à internet, não puderam ser reportadas no exato momento, expandindo assim a utilidade do aplicativo em diversas situações e cenários urbanos.

### **b. Opções de Reordenação na Tela Principal:**

Introduzir duas opções de reordenação na tela principal, permitindo que os usuários personalizem a visualização de reports com base em critérios específicos. A

primeira opção será reordenar os reports com base na proximidade da localização atual do usuário, mantendo o foco nas ocorrências mais relevantes para sua posição geográfica imediata. A segunda opção possibilitará a reordenação com base na proximidade de um ponto escolhido pelo usuário. Isso oferecerá maior controle e adaptabilidade, permitindo que os usuários priorizem informações relevantes com base em suas necessidades e preferências específicas.

Essas melhorias refletem o compromisso contínuo em otimizar a utilidade e a adaptabilidade do aplicativo SEG, buscando sempre atender às expectativas, necessidades e demandas dinâmicas de uma comunidade urbana em constante evolução.

## **6. Conclusão**

A crescente urbanização e a vida em grandes cidades apresentam tanto vantagens quanto desafios significativos para seus habitantes. Este projeto teve como objetivo central abordar um dos desafios mais prementes, que é a segurança dos cidadãos em ambientes urbanos complexos. Diante dos riscos associados à vida metropolitana, o desenvolvimento de um aplicativo inovador, denominado SEG, fundamentado nos princípios das Smart Cities, tornou-se essencial.

O aplicativo SEG é concebido como uma ferramenta eficaz para aprimorar a comunicação entre os cidadãos. O aplicativo permite que usuários relatem incidentes iminentes, contribuindo para a criação de uma rede de alerta e apoio colaborativa.

A metodologia de pesquisa adotada permitiu uma compreensão aprofundada dos desafios enfrentados pelas grandes cidades, bem como dos conceitos fundamentais das Smart Cities. Essa abordagem forneceu uma base sólida para o desenvolvimento do aplicativo e uma compreensão aprofundada do cenário urbano contemporâneo.

As especificações técnicas do aplicativo refletem a escolha estratégica do framework Flutter e da plataforma Firebase. Essas tecnologias não apenas possibilitam o desenvolvimento eficiente e atraente do aplicativo, mas também oferecem uma solução robusta e escalável. A compatibilidade multiplataforma e a integração com serviços em nuvem proporcionam uma experiência de usuário fluída e eficaz.

Ao considerar a importância atribuída à segurança pelos cidadãos a iniciativa do aplicativo SEG, em união com a colaboração dos usuários, está alinhada com as necessidades reais e prioritárias da população urbana. O aplicativo não apenas aborda a

preocupação com a segurança, mas também busca criar uma comunidade colaborativa onde a confiabilidade das informações é garantida por meio de avaliações dos usuários.

## 7. Referências

Ahmed, Y. Launching URLs in Flutter with url\_launcher. Disponível em: <[https://blog.logrocket.com/launching-urls-flutter-url\\_launcher/](https://blog.logrocket.com/launching-urls-flutter-url_launcher/)>. Acesso em: 21 nov. 2023.

Cunha, Maria Alexandra, et al. Smart cities: transformação digital de cidades. Programa Gestão Pública e Cidadania, 2016.

Curso de formação em A partir do zero: crie projetos em Dart, a linguagem utilizada no Flutter. Disponível em: <<https://www.alura.com.br/formacao-dart>>. Acesso em: 11 nov. 2023.

Firebase: o que é e para que serve essa ferramenta Google? Disponível em: <<https://rockcontent.com/br/blog/firebase/>>. Acesso em: 11 nov. 2023.

Firestore. Disponível em: <<https://firebase.google.com/docs/firestore?authuser=0&hl=pt>>. Acesso em: 11 nov. 2023.

Geolocator. Disponível em: <<https://pub.dev/packages/geolocator>>. Acesso em: 20 nov. 2023.

O que é Flutter? – Explicação sobre a aplicação Flutter – AWS. Disponível em: <<https://aws.amazon.com/pt/what-is/flutter/>>. Acesso em: 19 set. 2023.

O que são bancos de dados NoSQL? Disponível em: <<https://www.ibm.com/br-pt/topics/nosql-databases>>. Acesso em: 12 nov. 2023.

Palankar, R. The easiest way to implement image picker in flutter using library example. Disponível em: <<https://protocoderspoint.com/implement-image-picker-in-flutter-using-library-with-example/>>. Acesso em: 20 nov. 2023.

Sérgio De Lima, R.; Bueno, S.; Mingardi, G. Estado, polícias e segurança pública no Brasil STATE, POLICE AND PUBLIC SAFETY IN BRAZIL. Disponível em: <<https://www.scielo.br/j/rdgv/a/k8Cfd9XbDpJ8vzyfJqXP3qN/?format=pdf&lang=pt>>. Acesso em: 8 nov. 2022.

Wikipedia Contributors. Android Studio. Disponível em:  
<[https://pt.wikipedia.org/w/index.php?title=Android\\_Studio&oldid=66173914](https://pt.wikipedia.org/w/index.php?title=Android_Studio&oldid=66173914)>.  
Acesso em: 12 nov. 2023.

## **8. Apêndice**

Apêndice 1: Link para o repositório no GitHub:  
<<https://github.com/GustavoBarroso/seg>>