



**UNIVERSIDADE FEDERAL DE CAMPINA GRANDE
CENTRO DE ENGENHARIA ELÉTRICA E INFORMÁTICA
UNIDADE ACADÊMICA DE SISTEMAS E COMPUTAÇÃO
CURSO DE BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO**

GUSTAVO LUIZ BISPO DOS SANTOS

**CIDADE INTELIGENTE:
IMPLEMENTAÇÃO DE UM SISTEMA DE SOFTWARE
PARA GESTÃO URBANA**

CAMPINA GRANDE - PB

2023

GUSTAVO LUIZ BISPO DOS SANTOS

**CIDADE INTELIGENTE:
IMPLEMENTAÇÃO DE UM SISTEMA DE *SOFTWARE*
PARA GESTÃO URBANA**

**Trabalho de Conclusão Curso
apresentado ao Curso Bacharelado em
Ciência da Computação do Centro de
Engenharia Elétrica e Informática da
Universidade Federal de Campina
Grande, como requisito parcial para
obtenção do título de Bacharel em Ciência
da Computação.**

Orientador: Professor Dr. João Arthur Brunet Monteiro.

CAMPINA GRANDE - PB

2023

GUSTAVO LUIZ BISPO DOS SANTOS

**CIDADE INTELIGENTE:
IMPLEMENTAÇÃO DE UM SISTEMA DE *SOFTWARE*
PARA GESTÃO URBANA**

**Trabalho de Conclusão Curso
apresentado ao Curso Bacharelado em
Ciência da Computação do Centro de
Engenharia Elétrica e Informática da
Universidade Federal de Campina
Grande, como requisito parcial para
obtenção do título de Bacharel em Ciência
da Computação.**

BANCA EXAMINADORA:

**Professor Dr. João Arthur Brunet Monteiro
Orientador – UASC/CEEI/UFCG**

**Professor Dr. Francisco Vilar Brasileiro
Examinador – UASC/CEEI/UFCG**

CAMPINA GRANDE – PB

2023

CIDADE INTELIGENTE: IMPLEMENTAÇÃO DE UM SISTEMA DE SOFTWARE PARA GESTÃO URBANA

Gustavo Luiz Bispo dos Santos
gustavo.santos@ccc.ufcg.edu.br
Universidade Federal de Campina Grande
Campina Grande, Paraíba, Brasil

João Arthur Brunet Monteiro
joao.arthur@computacao.ufcg.edu.br
Universidade Federal de Campina Grande
Campina Grande, Paraíba, Brasil

RESUMO

Softwares de gestão são meios altamente eficazes de administrar um ambiente. Na sociedade moderna, em que as coisas ocorrem em alta velocidade e frequência, não é incomum deixar de anotar algo importante por falta de conhecimento da situação e, se tratando de cidades, mão de obra fiscalizadora. Em uma de suas falas ao público, o prefeito de Campina Grande, Bruno Cunha Lima, mencionou não haver registro na prefeitura sobre ruas calçadas, asfaltadas ou qualquer registro relacionado a isso. Este trabalho de conclusão de curso propõe um software para web que irá facilitar a gestão pública, registrando, de forma colaborativa, problemas encontrados na cidade. Ao final deste trabalho, esperamos possuir um ecossistema composto pela sociedade, sistema e poder público, capaz de garantir melhores condições de administração para o governo local e tornar-se replicável para outras municipalidades, beneficiando a sociedade através da transparência e agilidade aplicada aos governantes com o software.

PALAVRAS-CHAVE

Gestão pública, Colaborativa, Software, Transparência.

1. INTRODUÇÃO

O problema. Software de gestão são usualmente adotados por empresas dos mais variados setores, desde pequenos mercados e papelarias à grandes empresas como *Amazon* e *Microsoft*. A razão é uma: aumentar a eficiência do estabelecimento. Isso tem se provado ao vermos, cada vez mais, o abandono do "caderninho", em que o controle de dados poderia ser de custosa organização, lento e suscetível a falhas humanas, para a adesão de computadores e softwares, com o objetivo de se antepor aos erros do passado.

Ao tratar-se de centros urbanos, a situação é mais complexa e delicada: não há como uma prefeitura gerenciar tudo que acontece na cidade em tempo real, é inviável fazer um levantamento de quantas ruas possuem buracos, pois, certamente, ao final, o valor já não será o mesmo, não há funcionários suficientes na administração da cidade para fazer o mapeamento ou a correção dos problemas encontrados.

Dessa forma, é possível tomar como exemplo Ricardo Nunes, prefeito da grande cidade de São Paulo, onde existem milhares de ruas distribuídas entre 99 bairros e por seus mais de 1500 km², com seus mais de 31 milhões de veículos circulando no estado. Ricardo teria de descobrir, por meio de pesquisas, visitas e

levantamentos dele mesmo ou dos funcionários da prefeitura, os problemas que o município enfrenta quanto à infraestrutura e gestão, o que seria inviável, considerando que grandes centros urbanos estão em constante expansão e desenvolvimento.

Nesta perspectiva, percebe-se quão complexa é a gestão infraestrutural de um município sem mão de obra capaz de levantar as informações o quanto antes e sem um mecanismo em que se possa armazenar todas as informações de forma organizada e clara. Neste momento, é evidente a necessidade de um software colaborativo capaz de lidar com todas essas demandas técnicas para o funcionamento eficaz de uma gestão pública.

Objetivos. Este trabalho tem como objetivo principal desenvolver um sistema web de gestão colaborativa para manutenção e evolução de municípios, que proporcionará uma administração mais eficiente e simplificada aos governantes de cada cidade. Dessa forma, o gestor coordenará sua equipe de acordo com os anseios e necessidades de sua população.

Nesta perspectiva, o governante poderá ter sua cidade inserida no sistema, habilitando a gestão colaborativa de sua região através da plataforma desenvolvida. Posteriormente, poderá obter os dados sobre o que sua população tem reportado, averiguar a necessidade e viabilidade da implantação do pedido e comunicar a efetivação (ou não) da solução.

Ainda na plataforma, será possível acompanhar os status das solicitações feitas, adicionar comentários em cada requisição, verificar os responsáveis pelas solicitações (civil, fiscalizador e gestor), compartilhar via rede social, adicionar reputação aos usuários (governantes e civis) e exportar relatórios.

2. METODOLOGIA

Durante o desenvolvimento e análise do tema, foram levantados requisitos através da experiência vivenciada na cidade de Campina Grande, de conversas com ex-companheiros de trabalho experientes na área, e análise de um produto similar desenvolvido para solução de temática similar, neste caso, SeeClickFix. Os requisitos levantados foram descritos através das User Stories a seguir e documentadas em um quadro Kanban (Trello).

2.1. User Stories

2.1.1. Implementar cadastro e login e permissões de usuário.

Como usuário da plataforma, eu gostaria de ser capaz de criar uma conta, fazer login e ter permissões adequadas para visualizar o conteúdo relevante ao meu nível de acesso. Isso me permitirá ter uma experiência personalizada e segura ao usar a plataforma. A

atividade deve ser concluída em duas semanas e representa um risco baixo, pois o escopo da tarefa já está bem definido e a implementação não é vista como altamente complexa. O resultado esperado é ter as ações de login e cadastro plenamente funcionais, permitindo que os usuários criem e acessem suas contas com segurança e privacidade adequadas.

2.1.2. Implementar criação de cidades e gerenciamento de usuários

Como administrador da plataforma, eu gostaria de ser capaz de adicionar cidades ao sistema e adicionar usuários gestores à cidade. Além disso, os gestores de cada cidade deverão ser capazes de gerenciar outros usuários, concedendo ou revogando acesso e cargos. Isso me permitirá gerenciar e controlar melhor as operações em cada cidade da plataforma. A atividade deve ser concluída em duas semanas e representa um risco baixo, pois o escopo da tarefa já está bem definido e a complexidade de execução é baixa, não devendo interferir além do já previsto no cronograma. O resultado esperado é ter uma página de criação de cidade e uma página onde possam ser dados e revogados acessos e cargos, permitindo que os administradores e gestores de cada cidade possam gerenciar os usuários da plataforma de forma eficiente.

2.1.3. Implementar cadastro de problemas da cidade

Como um residente da cidade, eu gostaria de ser capaz de criar solicitações de resolução de problemas, descrevendo a categoria do problema, localização e descrição do ocorrido. Além disso, um dos usuários gestor da cidade deverá ser capaz de se atribuir como responsável pela resolução. Isso me permitirá reportar problemas e solicitar sua resolução de forma eficiente e rastreável. A atividade deve ser concluída em duas semanas e representa um risco médio, pois a implementação de um mapa pode não ocorrer adequadamente, nesse caso, a localização do problema será desenvolvida em campos de endereço como em um formulário. O resultado esperado é ter uma página para criação e visualização de um problema da cidade, onde os usuários possam reportar problemas e os gestores possam se atribuir como responsáveis pela sua resolução, aumentando a eficiência do processo de resolução de problemas na cidade.

2.1.4. Implementar dashboard

Como um residente ou gestor da cidade, eu gostaria de poder visualizar os problemas relacionados à minha cidade em um dashboard. Isso me permitirá acompanhar e monitorar de forma eficiente as solicitações de resolução de problemas na cidade. A atividade deve ser concluída em uma semana e representa um risco baixo, pois a listagem dos problemas será feita de forma simples, apenas exibindo alguns dados de cada solicitação realizada. O resultado esperado é ter uma página onde seja possível visualizar todos os problemas já cadastrados para a cidade, permitindo que os usuários possam monitorar e acompanhar a resolução dos problemas reportados.

2.1.5. Implementar status e resolução de problema e validação de terceiro

Como usuário relator ou validador de um problema, eu gostaria de ter a opção de acompanhar o andamento do processo e marcar o problema como resolvido, aumentando a confiabilidade e transparência da plataforma. Além disso, gostaria de poder atribuir-me como validador caso não haja ainda um validador para o problema. A atividade deve ser concluída em duas semanas e representa um risco baixo, uma vez que o escopo da tarefa foi minimizado a fim de que seja possível implementar de forma simples. O resultado esperado é ter na página de cada solicitação o seu status atual, para o relator, um botão de marcar o problema como resolvido e, para demais usuários civis, caso não haja ainda um validador, a opção de se auto-atribuir como validador do problema. Isso permitirá que os usuários possam monitorar o andamento da resolução dos problemas reportados, aumentando a confiabilidade e transparência da plataforma.

2.1.6. Implementar comentários de problema

Como usuário da plataforma, eu quero poder adicionar comentários em uma solicitação de problema para que eu possa fornecer mais informações ou feedback sobre a situação. A implementação deve permitir que qualquer usuário possa adicionar comentários na página de cada solicitação e que esses comentários sejam exibidos em uma lista. O tempo estimado para a implementação é de 1 semana e o risco é baixo, pois será uma adição simples de campo e listagem de comentários. O resultado esperado é ter um campo de adição de comentários e listagem dos mesmos na página de cada solicitação de problema.

2.1.7. Implementar compartilhamento de problema via rede social

Como usuário da plataforma, eu quero poder compartilhar solicitações de problemas via rede social para que eu possa aumentar a visibilidade e conscientização sobre a situação. A implementação deve adicionar um botão de compartilhamento nas páginas de cada solicitação de problema, que permita aos usuários compartilhá-las facilmente em suas redes sociais. O tempo estimado para a implementação é de 1 semana e o risco é baixo, já que o escopo da tarefa é simples e já está definido. O resultado esperado é ter um botão de compartilhamento visível na página de cada solicitação de problema.

2.1.8. Implementar exportação de relatório.

Como gestor da plataforma, eu quero poder exportar um relatório contendo informações sobre as solicitações feitas para que eu possa ter um registro das atividades realizadas. A implementação deve permitir que eu selecione um período de tempo para a geração do relatório e que ele contenha dados relevantes sobre as solicitações, como tipo, data de abertura, data de fechamento (se houver), status atual, entre outros. O tempo estimado para a implementação é de 1 semana e o risco é alto, pois pode faltar tempo para a conclusão da tarefa. O resultado esperado é ter a opção de exportar um relatório a partir do dashboard contendo os dados selecionados pelo gestor.

3. A SOLUÇÃO: FIXED MY CITY

Nesta seção, apresentaremos a solução proposta neste trabalho, que consiste em um software web de gestão colaborativa para registro e resolução de problemas encontrados na cidade

(disponível em <https://fixed-my-city.shop>). Esta seção será dividida em duas subseções. Na primeira, iremos descrever a arquitetura da solução, apresentando sua estrutura e funcionamento, detalhando as tecnologias utilizadas e explicando cada uma das suas funcionalidades. Na segunda subseção, abordaremos a usabilidade da plataforma, apresentando as principais telas e fluxos de uso do sistema, bem como sua interface intuitiva e de fácil utilização, permitindo que os usuários possam realizar o registro e acompanhamento dos problemas encontrados na cidade de forma rápida e eficiente.

3.1 Arquitetura

A arquitetura da solução desenvolvida é fundamentada em uma abordagem modular, utilizando padrões e tecnologias modernas para garantir a escalabilidade, flexibilidade e eficiência do sistema. A solução adota a arquitetura MVC (Model-View-Controller) como base para a estruturação e organização dos componentes, como proposto na imagem a seguir, onde a View exibe os dados para o usuário, o Controller processa a lógica e controla o fluxo de dados ligado ao servidor e atua como middleware entre a View e o Model. Por fim, o Model faz o processamento de dados que serão enviados ou que chegaram do banco de dados.

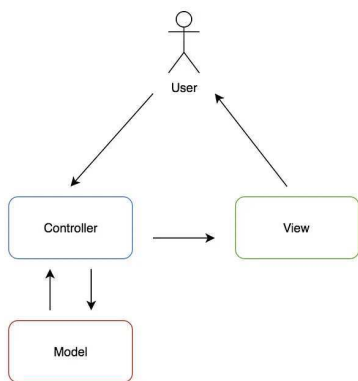


Figura 1: Arquitetura MVC

No que diz respeito à camada de interface do usuário, o front-end foi implementado utilizando tecnologias como React, TypeScript e Tailwind CSS. O React, um framework JavaScript amplamente utilizado no desenvolvimento de interfaces de usuário, permite a construção de componentes reutilizáveis e de alto desempenho. A utilização do TypeScript, uma linguagem de programação tipada, contribui para a melhoria da manutenção e escalabilidade do código. Além disso, o Tailwind CSS foi empregado para facilitar a estilização e o design responsivo da aplicação. A solução também é configurada como Progressive Web Application (PWA), possibilitando que os usuários acessem o sistema através de navegadores web e dispositivos móveis, com recursos offline e notificações push.

No backend, o sistema utiliza o framework Express.js em conjunto com o runtime Node.js e o TypeScript. O Express.js é um framework minimalista e flexível que facilita o

desenvolvimento de aplicativos web e APIs. Já o Node.js, um ambiente de execução JavaScript, permite que a solução seja altamente escalável e eficiente, tornando-a adequada para lidar com uma grande quantidade de dados e solicitações concorrentes. A utilização do TypeScript no backend proporciona benefícios como tipagem estática e autocompletar no código, tornando-o mais robusto e confiável. Para a comunicação com o banco de dados, a solução utiliza o Prisma ORM, que simplifica a interação com o banco de dados MySQL hospedado na PlanetScale. O Prisma ORM oferece uma interface intuitiva e poderosa para a execução de consultas e manipulação de dados, garantindo a segurança e integridade das informações.

Quanto à arquitetura implementada no backend, trata-se de uma API RESTful com uma abordagem MVC. A arquitetura REST (Representational State Transfer) é um estilo arquitetural amplamente utilizado para o desenvolvimento de APIs. Ela enfatiza a comunicação entre o cliente e o servidor por meio de operações HTTP, utilizando os verbos HTTP (GET, POST, PUT, DELETE, etc.) para manipular recursos. A abordagem MVC mencionada se aplica aqui da seguinte forma: os controladores (abstraídos para dentro das rotas da API) atuam como o intermediário entre as requisições e o modelo, processando as solicitações recebidas e fornecendo as respostas adequadas. Os modelos, através do Prisma ORM, interagem com o banco de dados MySQL para realizar operações de persistência de dados, conforme o diagrama abaixo.

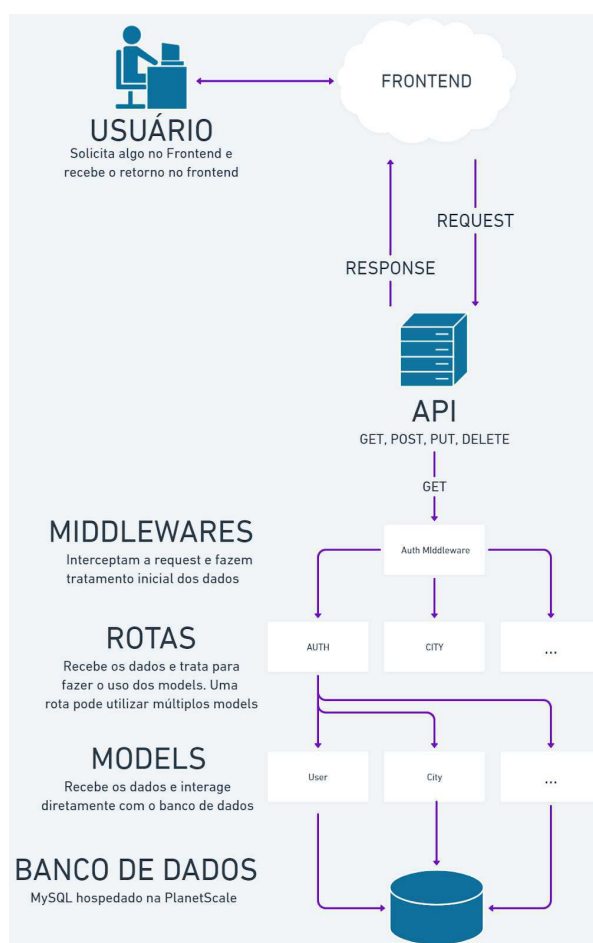


Figura 2: Arquitetura da API



Figura 3: Diagrama de classes UML

3.2 Utilizando o Fixed My City

O Fixed My City é uma plataforma de gestão colaborativa que foi projetada para resolver os problemas mencionados acima. Primeiramente, quando qualquer usuário faz a autenticação no sistema, ele irá se deparar com uma tela semelhante à seguinte, onde ele verá os dados referentes a sua cidade, incluindo um mapa de calor com as informações do período filtrado.

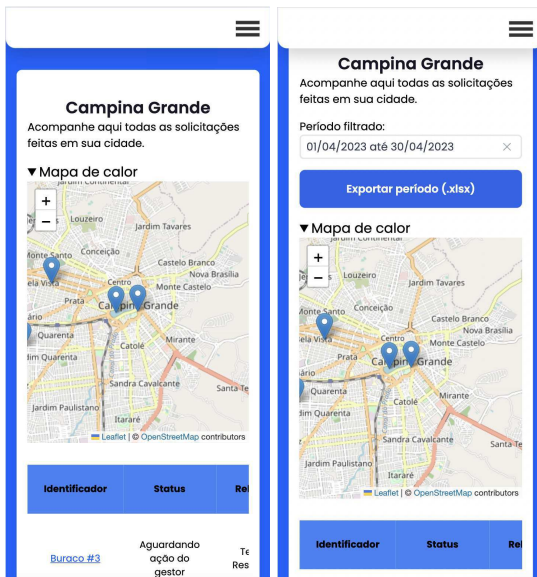


Figura 4: Visão do residente (à esquerda) e do gestor ou owner (à direita) na tela inicial

Uma solicitação (ou requisição), é um pedido registrado na plataforma feito aos gestores e owners de uma cidade. Isso pode ser registrado da seguinte forma:

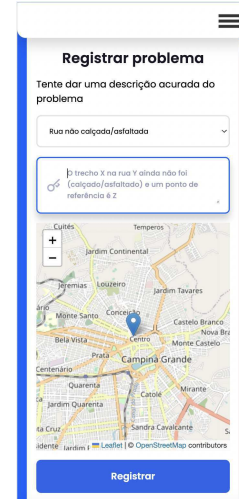


Figura 5: Tela de registro de problema

De tal forma como o usuário pode ver todas as requisições feitas à cidade, também é possível que qualquer pessoa consiga listar todas as suas solicitações:

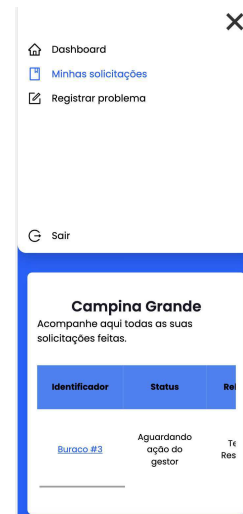


Figura 6: Tela de solicitações feitas pelo usuário

Dentro de uma solicitação, é possível obter diversas informações, como o relator, o fiscal, se atribuir a um dos cargos, compartilhar através das redes sociais, ver quando houveram as últimas atualizações, além de poder acompanhar através do campo de comentários o andamento da situação. Somente o relator pode marcar um problema como resolvido.

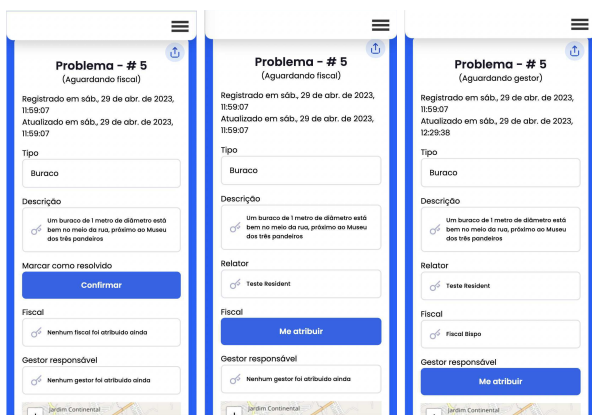


Figura 7: Visão do relator (à esquerda), do usuário comum (ao centro) e do gestor (à direita) na tela de uma solicitação

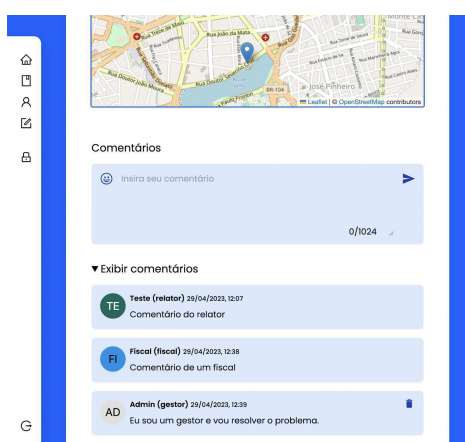


Figura 8: Visão do campo de comentários na tela de uma solicitação

Como gestor ou owner de uma cidade, é possível definir quem receberá elevação de cargos, respeitando a hierarquia, onde um gestor não pode atribuir um owner, nem mudar seu cargo. Isso permite que a cidade seja gerenciada mais facilmente, dando poderes de gestão a mais pessoas que ficarão responsáveis por cuidar da cidade.

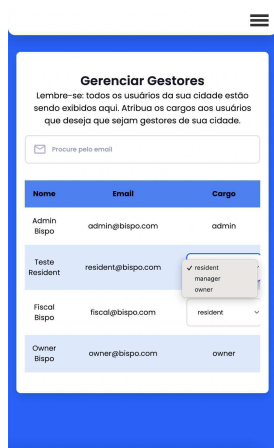


Figura 9: Visão da tela de gerência de gestores (acessível apenas por gestores, owners ou administradores da plataforma)

4. DISCUSSÃO E TRABALHOS FUTUROS

Aqui trataremos das limitações da aplicação desenvolvida e trabalhos futuros os quais pretendemos realizar.

Considerando as possibilidades de melhoria e expansão do software de gestão de cidades apresentado neste trabalho, algumas recomendações para trabalhos futuros foram definidas. A primeira delas é a implementação de notificação de mudança de status, que permitirá aos usuários acompanharem em tempo real as atualizações do andamento de suas solicitações. Essa funcionalidade irá agilizar o processo de solução dos problemas, proporcionando maior transparência e credibilidade ao sistema, além de melhorar a experiência do usuário.

Outra possibilidade de aprimoramento é a inclusão da reputação do usuário. Essa funcionalidade permitirá que cada usuário tenha uma nota de reputação, que poderá ser vista por todos os usuários da plataforma. Essa reputação será baseada em critérios como avaliações positivas das soluções encontradas, colaboração com outros usuários e iniciativas tomadas para resolução ou levantamento de um problema. Essa funcionalidade irá incentivar os usuários a serem mais participativos e engajados no processo de solução de problemas.

Por fim, uma outra recomendação para trabalhos futuros é a inclusão de um registro de ocorrências policiais. Com essa funcionalidade, os usuários poderão informar sobre crimes e outras situações de violência ocorridas na cidade, permitindo uma visão mais ampla e detalhada da segurança do município. Além disso, será possível criar gráficos e mapas de calor, facilitando a visualização dos locais mais problemáticos da cidade e permitindo ações mais efetivas por parte das autoridades competentes.

5. AGRADECIMENTOS

A realização desse trabalho só foi possível devido ao apoio de várias pessoas que estiveram comigo em todas as etapas desse grande processo e a elas agradeço publicamente.

A Deus, pela graça de conseguir chegar até aqui e finalizar o curso depois de tantas dificuldades ao longo da caminhada.

A minha noiva, Déborah Diniz, que me estimulou e esteve ao meu lado nos momentos mais difíceis da minha vida e me encorajou a prosseguir no curso até o fim.

A minha família, que me apoiou e promoveu meios e se esforçaram ao máximo para que eu tivesse todo o tempo necessário para estudar e alcançar cada vez mais longe.

Ao meu irmão, Sem. André Bispo, que enxergou o que eu não via e me ajudou a escolher o curso de computação da UFCG.

A minha sogra, Me. Rubia Karine, que me apoiou, aconselhou, orientou e esteve sempre ao meu lado.

Ao meu sogro, Gutemberg Medeiros, que me orientou quanto a minha carreira, tirou muitas dúvidas quanto a diversos assuntos e é uma de minhas referências profissionais.

Ao meu ex-líder e amigo, Leandro Mangini, que me ensinou muito quanto a diversos assuntos na área de computação e me capacitou a ser o profissional que sou hoje.

Ao meu amigo, Dr. Cícero Fellipe, que sempre me aconselhou quanto a vários assuntos na área acadêmica.

Aos meus ex-companheiros de equipe Sean Sawyer e Gary Wright que me transmitiram conhecimento que me permitiu criar uma plataforma robusta inspirada em um produto que desenvolvemos juntos e que funciona em boa parte dos Estados Unidos da América.

6. REFERÊNCIAS

- [1] Garagem360, 2022, Carros e motos em SP; descubra quantos milhões circulam, conforme pesquisa. Acessada em Out, 2022 de <https://garagem360.com.br/carros-e-motos-em-sp-descubra-quantos-milhoes-circulam-conforme-pesquisa>
- [2] Zap imóveis, 2022, Quantos bairros tem em São Paulo?. Acessada em Out, 2022 de <https://www.zapimoveis.com.br/blog/guia-de-cidades/quantos-bairros-tem-em-sao-paulo/>
- [3] Daniel Deutsch, 2017, Understanding MVC Architecture with React. Acessada em Mai, 2023 de <https://medium.com/createdd-notes/understanding-mvc-architecture-with-react-6cd38e91fef9>
- [4] Sean Sawyer, Lead Software Engineer na empresa CivicPlus. Disponível em <https://www.linkedin.com/in/sean-sawyer-deveng/>
- [5] Gary Wright, Software Development Manager na empresa CivicPlus. Disponível em <https://www.linkedin.com/in/gary-wright-03a043a/>
- [6] SeeClickFix, Software desenvolvido e aplicado nos EUA e que serviu de inspiração para a criação da solução descrita neste documento. Disponível em <https://seeclickfix.com>