

INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DO
RIO GRANDE DO SUL
CAMPUS CANOAS
CURSO SUPERIOR DE TECNOLOGIA EM ANÁLISE E DESENVOLVIMENTO
DE SISTEMAS

FELIPE EMERIM LEAL

**Análise e Desenvolvimento de um sistema de mapeamento
criminal colaborativo**

Canoas, 6 de setembro de 2021.

FELIPE EMERIM LEAL

**Análise e Desenvolvimento de um sistema de mapeamento
criminal colaborativo**

Trabalho de Conclusão de Curso
apresentado como requisito parcial para
obtenção do grau de Tecnólogo em Análise
e Desenvolvimento de Sistemas pelo
Instituto Federal de Educação, Ciência e
Tecnologia do Rio Grande do Sul – Campus
Canoas.

Prof. Igor Lorenzato Almeida
Orientador

Canoas, 6 de setembro de 2021.



Ministério da Educação
Secretaria de Educação Profissional, Científica e Tecnológica
Instituto Federal de Educação, Ciência e Tecnologia do Rio Grande do Sul
Campus Canoas

ATA DE DEFESA PÚBLICA DO TRABALHO DE CONCLUSÃO DE CURSO

Aos **24** dias do mês de **agosto** de **2021**, às **9 horas e 40 minutos**, em sessão pública realizada através do Google Meet, na presença da Banca Examinadora presidida pelo Professor:

Ígor Lorenzato Almeida e

composta pelos examinadores:

1. **Profa. Denise Regina Pechmann,**

2. **Prof. Gustavo Neuberger,**

o aluno **Felipe Emerim Leal** apresentou o Trabalho de Conclusão de Curso intitulado: **Análise e desenvolvimento de um sistema de mapeamento criminal colaborativo** como requisito curricular indispensável para a integralização do Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas. Após reunião em sessão reservada, a Banca Examinadora deliberou e decidiu pela **APROVAÇÃO** do referido trabalho, divulgando o resultado formalmente ao aluno e demais presentes e eu, na qualidade de Presidente da Banca, lavrei a presente ata que será assinada por mim, pelos demais examinadores e pelo aluno.

Ígor Lorenzato Almeida
Digitally signed by Ígor Lorenzato Almeida
DN: c=BR, OU=Campus Canoas, O=IFRS, CN=Ígor Lorenzato Almeida,
E=igor.almeida@canoas.ifrs.edu.br
Reason: Eu sou o autor deste documento.
Location: Porto Alegre, RS
Date: 2021.08.26 00:09:17
Foxit Reader Version: 9.6.0

Presidente da Banca Examinadora

Denise Regina Pechmann
Assinado digitalmente por Denise Regina Pechmann
DN: c=BR, OU=IFRS Campus Canoas, O=IFRS, CN=Denise Regina Pechmann, E=denise.pechmann@canoas.ifrs.edu.br
Reason: Eu sou o autor deste documento
Localização: sua localização de assinatura aqui
Data: 2021.08.27 18:44:44-03'00'
Foxit Reader Versão: 10.1.1

Examinador 01

Gustavo Neuberger
Assinado digitalmente por Gustavo Neuberger
DN: c=BR, OU=IFRS Campus Canoas, O=Gustavo Neuberger, CN=Gustavo Neuberger,
E=gustavo.neuberger@canoas.ifrs.edu.br
Reason: Eu sou o autor deste documento
Localização: sua localização de assinatura aqui
Data: 2021.08.26 11:36:54-03'00'
Foxit Reader Versão: 10.1.0

Examinador 02

Felipe Leal
Digitally signed by Felipe Leal
DN: C=BR, CN=Felipe Leal,
E=lp.emerim@gmail.com
Reason: I agree to specified parts of this document
Location: Canoas
Date: 2021.09.06 21:50:20-03'00'
Foxit PDF Reader Version: 11.0.1

Aluno

AGRADECIMENTOS

Gostaria de agradecer a minha mãe pelo suporte emocional nos momentos difíceis e por nunca me deixar desistir dos meus objetivos ao longo do curso e deste trabalho. Ela foi a pessoa que me trouxe conforto e energia para seguir adiante e finalizar mais esta etapa da minha vida. Mãe, te agradeço do fundo do coração por estar ao meu lado em todos os momentos.

Outra pessoa especial a quem eu gostaria de deixar meu agradecimento é a minha irmã Carina. Ela esteve junto comigo em todos os momentos ao longo do curso e do trabalho, me acompanhou e me apoiou no desenvolvimento deste trabalho.

Agradeço também ao meu cunhado Henrique pelo auxílio principalmente na idealização e melhorias das telas do sistema, e às várias discussões relevantes que ajudaram a aprimorar o trabalho.

Meu pai também me apoiou ao longo de todo curso, principalmente me levando e buscando da faculdade em dias frios e chuvosos, contribuindo para que eu pudesse chegar até esta etapa e concluí-la. Obrigado pai.

Agradeço aos professores que tive ao longo do curso, em especial ao professor Rodrigo Noll por sempre me incentivar e trazer experiências em sala de aula que ajudaram a me preparar para a vida profissional. Deixo mais um agradecimento especial à professora Denise por me ajudar a refinar a ideia inicial deste trabalho, sugerindo várias funcionalidades e melhorias relevantes. Por último agradeço ao meu orientador Igor Almeida pelos conselhos e direcionamento na construção deste trabalho.

RESUMO

Sabe-se que a criminalidade está aumentando e as pessoas acabam ficando temerosas em sair de suas residências para se deslocar a certos locais em determinadas horas. Este trabalho propõe oferecer um mapa de crimes, que será alimentado de maneira colaborativa pela própria população. A ideia é que, de maneira rápida e fácil, um usuário possa acessar a aplicação e verificar os pontos onde foram cadastradas ocorrências como roubos, furtos, uso de drogas e assim evitar se dirigir até estes locais. Este trabalho também informa sobre as motivações que levaram o autor à construção da ideia, contém uma introdução sobre o assunto, uma análise realizada sobre plataformas semelhantes a que foi desenvolvida, as tecnologias que foram utilizadas, seus objetivos geral e específicos e também informações sobre o desenvolvimento e testes realizados.

Palavras-Chave: Mapeamento Criminal. Colaborativo. Aplicativo Web.

ABSTRACT

It is known that the crime is increasing, and people end up becoming afraid of going out of their residences to go to certain places at specific times. The goal of this paper is to offer a crime map which will have its data supplied by the population itself. The idea is that, in a fast and easy way, a user can access the application and check the spots where occurrences like robbery, theft, drug use have been registered and therefore, avoid going to said spots. This paper also informs the reader about the motivations which led the author to build the idea, contains an introduction about the subject, an analysis of applications like the one that was developed, the technologies that were used, its general and specific purposes, and information about its development and testing.

Palavras-Chave: Criminal Mapping. Collaborative. Web App.

LISTA DE FIGURAS

Figura 1 - Tela para cadastro de recebimento de alertas no Crimemapping	17
Figura 2 - Relatórios do Onde Tem Tiroteio	18
Figura 3 - Arquitetura Cliente-Servidor.	21
Figura 4 - Componentes Spring Boot	24
Figura 5 - Trecho de código utilizando Thymeleaf	25
Figura 6 - Integração do Hibernate Search através do ORM	26
Figura 7 - Funcionamento dos contêineres dentro de um SO	27
Figura 8 - Ambiente utilizando máquinas virtuais	28
Figura 9 - Docker X Docker Compose	29
Figura 10 - Diagrama de Casos de Uso CrimeSpot	34
Figura 11 - Diagrama ER CrimeSpot	36
Figura 12 - Tela Inicial CrimeSpot	39
Figura 13 - Filtros na página inicial	40
Figura 14 - Tela de login	41
Figura 15 - Formulário de cadastro de usuário	42
Figura 16 - Tela inicial com usuário autenticado	43
Figura 17 - Edição de cadastro de usuário	44
Figura 18 - Tela de Alteração de senha	44
Figura 19 - Tela de listagem de ocorrências	45
Figura 20 - Tela de detalhes de ocorrência	46
Figura 21 - Formulário de criação de ocorrência	47
Figura 22 - Denunciar ocorrência no mapa de crimes	48
Figura 23 - Modal de denúncia de ocorrência	48
Figura 24 - Tela inicial do moderador	49
Figura 25 - Lista de usuários	50
Figura 26 - Lista de ocorrências pendentes	50
Figura 27 - Modal de confirmação de início de revisão	51
Figura 28 - Lista de ocorrências em análise	52
Figura 29 - Parte de comentários do moderador	52
Figura 30 - Tela inicial Administrador	53
Figura 31 - Lista de ocorrências denunciadas	54
Figura 32 - Denúncias realizadas por usuários em uma ocorrência	54
Figura 33 - Lista de usuários apresentada ao Administrador	55
Figura 34 - Detalhes de ocorrência apresentado ao Administrador	56
Figura 35 - Gráfico que representa os resultados dos objetivos dos usuários	57
Figura 36 - Gráfico representando as respostas da segunda pergunta	57
Figura 37 - Gráfico representando as respostas da terceira pergunta	58
Figura 38 - Gráfico representando as respostas da pergunta 4	58
Figura 39 - Gráfico representando as respostas da pergunta 5	59

Figura 40 - Gráfico representando os resultados da pergunta 6	59
Figura 41 - Gráfico representando as respostas da pergunta 7	60
Figura 42 - Botão Limpar Filtros.....	61
Figura 43 - <i>Tooltip</i> nas ocorrências verificadas	61
Figura 44 - <i>Dropdown</i> de usuário estilizado	62
Figura 45 - Botão para centralizar o mapa na localização do usuário	62

LISTA DE TABELAS

Tabela 1 - Comparativo entre Sistemas	19
Tabela 2 - Entidades, chaves e relacionamentos	37

LISTA DE ABREVIATURAS E SIGLAS

ACID	Atomicidade, Consistência, Isolamento e Durabilidade
API	<i>Application Programming Interface</i>
AWS	<i>Amazon Web Services</i>
BO	Boletim de Ocorrência
CD	<i>Continuous Delivery</i>
CI	<i>Continuous Integration</i>
DNS	<i>Domain Name System</i>
EC2	<i>Amazon Elastic Compute Cloud</i>
ER	Entidade-Relacionamento
HTML	<i>HyperText Markup Language</i>
HTTP	<i>Hypertext Transfer Protocol</i>
HTTPS	<i>Hypertext Transfer Protocol Secure</i>
LAN	<i>Local Area Network</i>
ORM	<i>Object-Relational Mapping</i>
SGBD	Sistema de Gerenciamento de Banco de Dados
SO	Sistema Operacional
SQL	<i>Structured Query Language</i>
URL	<i>Uniform Resource Locator</i>
WAN	<i>Wide Area Network</i>
YAML	<i>Ain't Markup Language</i> (Acrônimo Recursivo)

SUMÁRIO

RESUMO	5
ABSTRACT	6
LISTA DE FIGURAS	7
LISTA DE TABELAS	9
LISTA DE ABREVIATURAS E SIGLAS	10
1 INTRODUÇÃO	13
1.1 MOTIVAÇÃO.....	13
1.2 OBJETIVOS	14
1.2.1 Objetivo Geral	14
1.2.2 Objetivos Específicos.....	14
2 ESTADO DA ARTE	16
2.1 CRIMEMAPPING	16
2.2 ONDE TEM TIROTEIO	17
2.3 COMPARAÇÃO ENTRE OS SISTEMAS.....	18
3 REVISÃO BIBLIOGRÁFICA	20
3.1 ARQUITETURA CLIENTE-SERVIDOR.....	20
3.1.1 Web Service.....	21
3.1.2 Servidor	21
3.1.3 Servidor Web	22
3.1.4 Server-Side Render	22
3.2 APLICAÇÃO WEB	23
3.2.1 Java	23
3.2.2 Spring Boot.....	23
3.2.3 Thymeleaf.....	24
3.2.4 PostgreSQL	25
3.2.5 Hibernate Search	25
3.3 DOCKER.....	26
3.3.1 Docker Compose	28
4 METODOLOGIA	30
5 DESENVOLVIMENTO	32
5.1 MODELAGEM	32
5.1.1 Documento de Visão	32
5.1.2 Diagrama de Casos de Uso.....	33
5.1.3 Diagrama ER	35
5.2 O CRIMESPOT	39

5.3	TESTES DE ACEITAÇÃO.....	56
6	CONCLUSÃO.....	64
6.1	TRABALHOS FUTUROS	64
	REFERÊNCIAS	66
	APÊNDICE A – DOCUMENTO DE VISÃO	70
	APÊNDICE B – FORMULÁRIO DE AVALIAÇÃO	74

1 INTRODUÇÃO

Atualmente, na era digital, o compartilhamento de informações está mais fácil. Com o crescente uso de redes sociais as pessoas estão se conectando e quebrando as barreiras que a distância criava, utilizando-se das diversas vantagens que a internet oferece (OLIVEIRA, 2018).

Com esse acesso facilitado às informações, notícias, tanto boas quanto ruins, chegam até o público com maior velocidade e quantidade. Neste contexto é fácil perceber o quanto a criminalidade tem aumentado. (OLIVEIRA, 2018)

Regiões onde se sabia serem mais tranquilas, hoje sofrem com crimes e violência e se torna cada vez mais difícil sentir-se seguro, independentemente do local ou horário (GAUCHAZH, 2018).

A população sofre com a falta de segurança e, devido a isso, apela para todos os possíveis aparatos que possam deixá-las ao menos um pouco mais tranquilas, como cerca elétrica, segurança privada e grupos de monitoramento entre vizinhos.

Contar apenas com a polícia para segurança não está sendo suficiente. Muitas vítimas de violência, assaltos e demais, acabam por não registrarem boletim de ocorrência nas delegacias devido a uma existência de maior número de crimes, do que policiais para investigarem todos os casos. (DIÁRIO GAÚCHO, 2019)

Para auxiliar no mapeamento de crimes e determinar áreas de risco, a polícia já dispõe de mecanismos e sistemas que a auxiliam, porém a população não tem acesso a estes dados e acaba por permanecer indefesa. Seria interessante poder ter informações sobre locais perigosos e sobre locais em que ocorrem determinados tipos de crimes (GAUCHAZH, 2016).

A proposta do sistema CrimeSpot é oferecer uma ferramenta colaborativa que reúna as situações de violência vividas ou observadas por usuários, onde eles possam cadastrá-las e assim gerar um mapeamento de áreas de risco para auxiliar todos os usuários que acessarem-na.

1.1 MOTIVAÇÃO

Apenas vendo um telejornal, por exemplo, já é possível verificar os casos de violência que ocorrem diariamente. A população fica insegura no meio de todos estes crimes. Os índices de violência aumentaram muito em todo Brasil e vários fatores podem influenciar neste aumento como: aumento da taxa de desemprego, o crescimento da

população, o empobrecimento da população, baixo poder aquisitivo, etc. Talvez haja hoje maior transparência na divulgação desse fenômeno (MÁXIMO, 2004).

Conforme análise realizada por Campos (2015, p.180) quanto às causas para aumento da criminalidade realizando uma análise de dois pontos de vista diferentes, “[...] o crime vem aumentando e as ações dos criminosos são cruéis, o Estado é corrupto, omissivo e seus métodos ineficientes e, para se combater a crueldade dos criminosos, as leis devem ser endurecidas”. Também, segundo essa análise, existe uma parcela da população que acredita que a alteração das leis as endurecendo irá diminuir os índices criminais. Porém no mesmo artigo, uma outra ótica é apresentada ao leitor, informando que “o combate ao crime deve envolver discussões e adoção de medidas de prevenção que envolvam diferentes órgãos da sociedade (igrejas, prefeituras, estado, união, polícia, comunidade), além de medidas policiais” (CAMPOS, 2015).

Muitas discussões podem ser levantadas sobre o tema criminalidade, porém o autor verifica uma emergência dos cidadãos para que ações efetivas de combate à violência sejam tomadas. No círculo social do autor, ele pôde notar o medo das pessoas e verificar que muitos deste grupo já sofreram algum tipo de violência ou ameaça.

O conhecimento dos dados atuais sobre a criminalidade, através de um mapeamento bem elaborado, permitirá o controle e o desenvolvimento de ações preventivas contra o crime e auxiliará a população a evitar locais sabidamente mais perigosos.

Este trabalho propõe disponibilizar os dados de ocorrências sofridas por usuários para que as demais pessoas que acessarem a plataforma possam identificar possíveis áreas de risco e locais perigosos e, assim, os evitem.

A premissa é ser colaborativo, portanto, os próprios usuários alimentam o banco de dados. O sistema processa estes dados cadastrados e os devolve de uma maneira organizada e que faz sentido a quem estiver recebendo a informação.

1.2 OBJETIVOS

1.2.1 Objetivo Geral

Desenvolver um sistema de mapeamento criminal colaborativo como forma de facilitar o acesso a informações sobre pontos de criminalidade na cidade Canoas.

1.2.2 Objetivos Específicos

- Pesquisar sistemas semelhantes;
- Desenvolver o sistema utilizando a linguagem de programação Java com o framework Spring;
- Realizar a automação dos principais fluxos do sistema utilizando Selenium com o auxílio do FluentLenium;
- Documentar os fluxos do sistema utilizando diagramas da Engenharia de Software;

- Realizar testes de aceitação com usuários;
- Disponibilizar os dados para que os usuários que acessarem a plataforma possam identificar possíveis áreas de risco e locais perigosos.

Neste capítulo foi apresentada a introdução e a motivação deste trabalho explicando o porquê de o autor ter escolhido este tema e quais objetivos a serem alcançados com o desenvolvimento deste TCC. No próximo capítulo serão apresentados sistemas semelhantes ao CrimeSpot, que foi desenvolvido neste trabalho.

Nos próximos capítulos serão apresentados sistemas semelhantes ao CrimeSpot, metodologia utilizada no desenvolvimento, revisão bibliográfica, diagramas criados para auxiliar no desenvolvimento e telas do sistema. Na próxima seção será apresentado o estado da arte que traz sistemas semelhantes ao CrimeSpot para que possa ser realizada uma análise e posterior comparação entre essas plataformas.

2 ESTADO DA ARTE

Para auxiliar na organização das ideias desta pesquisa, foi realizada a análise de dois sistemas semelhantes ao que foi desenvolvido neste trabalho.

Para que fosse possível localizá-los, foi utilizado o termo “*crime mapping system*” no sistema de buscas *Google*, onde um dos resultados apresentados trouxe o sistema Crimemapping (2021).

Um outro aplicativo analisado e que contribuiu para execução deste trabalho foi o aplicativo *mobile* OTT – Onde tem tiroteio (2021), que foi localizado através da indicação de um usuário, e pode ser encontrado na *Play Store* ou *App Store*. Este aplicativo é disponibilizado no Brasil, enquanto que o Crimemapping abrange apenas os Estados Unidos.

A disseminação de notícias e dados falsos na internet é uma realidade que precisa ser encarada por todos os usuários da rede mundial de computadores.

Conforme noticiado pela Rede Brasil Atual (2017):

No início deste mês, o casal de fotógrafos [...] foi espancado em Araruama, no Rio de Janeiro, após um boato de que eles estariam sequestrando crianças viralizar no Whatsapp. O texto incluía fotos dos dois e do veículo deles, inclusive com a placa de identificação do carro. Em maio de 2014, [...] foi morta em um linchamento no Guarujá, litoral paulista. Ela também vítima de um boato, que dizia que ela sequestrava crianças e fazia rituais de magia negra (Rede Brasil Atual, 2017).

Ao analisar os sistemas que serviriam de ponto de partida para o CrimeSpot, essa preocupação com a verificação das informações foi identificada facilmente.

2.1 CRIMEMAPPING

O sistema Crimemapping (2021) permite a inserção de ocorrências apenas por delegacias devidamente identificadas e disponibiliza esses dados para que os usuários possam visualizar estatísticas ou criar alertas de acordo com seus interesses.

Conforme informações retiradas do próprio site do aplicativo, as delegacias que desejam fazer parte do sistema precisam entrar em contato e solicitar a análise por parte dos moderadores do sistema. Um dado inicial que precisa ser enviado nessa solicitação é o código da delegacia.

Não é possível realizar o cadastro de um usuário no Crimemapping (2021), pois nele são cadastradas as delegacias em si.

Com os dados de ocorrências cadastrados, a ferramenta apresenta ao usuário dois gráficos do resumo semanal dos crimes, onde é possível observar a quantidade de cada tipo de ocorrência, como uso de drogas, uso de álcool e abuso sexual.

Uma desvantagem é o fato de poderem ser visualizados apenas os índices semanais, não tendo opção de analisar dados anteriores a sete dias.

O Crimemapping (2021) aceita cadastros apenas das regiões dos Estados Unidos, não sendo possível utilizá-lo no Brasil.

Conforme dito anteriormente, é possível realizar o cadastro e configuração de recebimento de alertas sobre ocorrências próximas a um endereço específico, na Figura 1 consta a tela onde um usuário pode cadastrar alertas para receber.

Figura 1 - Tela para cadastro de recebimento de alertas no Crimemapping.

Fonte: Crimemapping, 2021.

2.2 ONDE TEM TIROTEIO

Um sistema que também foi analisado é o OTT – Onde Tem Tiroteio. Uma plataforma mobile que aceita cadastros no Brasil, porém apenas na região do Rio de Janeiro e São Paulo (OTT – Onde Tem Tiroteio, 2021).

Conforme a própria página do aplicativo informa, ele foi criado para proteger os cidadãos e preveni-los sobre balas perdidas, arrastões e falsas blitz.

A principal funcionalidade deste sistema é enviar alertas sobre crimes, que são cadastrados pelos próprios usuários. Quando o alerta é cadastrado, ele primeiramente

passa por uma equipe de moderação que atesta a autenticidade e só após o dispara aos demais usuários.

O OTT também oferece uma página de relatórios que pode ser acessada via *web*, conforme apresentado na Figura 2.



Fonte: OTT – Onde Tem Tiroteio, 2021.

Com essa funcionalidade de relatórios é possível acessar dados históricos para verificar o comportamento do crime em determinadas localidades, porém é possível ter acesso apenas aos meses do ano vigente.

O aplicativo foi criado por um grupo de voluntários e, para garantir a confiabilidade dos dados, os usuários que cadastram os alertas são de confiança dos moderadores, não é qualquer pessoa que pode se cadastrar. O grupo também mantém regras rígidas de funcionamento.

2.3 COMPARAÇÃO ENTRE OS SISTEMAS

A comparação entre sistemas semelhantes ao que está sendo apresentado neste trabalho foi importante para identificar funcionalidades que possam ser mais interessantes aos usuários deste tipo de sistema, bem como para definir algumas restrições com objetivo

de manter a confiabilidade dos dados e preservar a identidade das pessoas que irão cadastrar as ocorrências.

No sistema Crimemapping, o primeiro analisado, pode-se notar a preocupação com a veracidade das informações de modo que apenas delegacias devidamente cadastradas e aprovadas por moderação podem inserir as ocorrências.

Já o segundo sistema analisado, Onde Tem Tiroteio, permite que usuários cadastrados insiram ocorrências, porém mantém um forte esquema de moderação com intuito que informações inverídicas sejam excluídas da plataforma o mais rápido possível.

Na Tabela 1 é possível verificar um comparativo entre os sistemas citados.

Tabela 1 - Comparativo entre Sistemas

Funcionalidades	Crimemapping	OTT	CrimeSpot
Colaborativo	X	X	X
Controle de Entrada de dados		X	X
Permite inserção de vários tipos de crime	X	X	X
Possibilidade dos usuários se cadastrarem		X	X
Usuários cadastrados do sistema cadastram os dados		X	X
Login Google		X	
Notificação de novos crimes	X	X	X
Representação gráfica das informações	X	X	
Parceria com delegacias	X		
Opção de filtrar ocorrências no mapa	X		X
Visualização de ocorrências no mapa	X	X	X
Suporte à utilização no Rio Grande do Sul			X

Fonte: Próprio autor.

O sistema CrimeSpot se difere dos demais devido a focar na colaboração entre os usuários para montar sua base de ocorrências. Existe a possibilidade de cadastrar ocorrências com ou sem BO, para que possa abranger uma maior gama de crimes, pois em algumas situações não é possível registrar um Boletim de Ocorrência. Um exemplo é quando o usuário apenas presencia um crime ocorrendo.

Outro ponto de atenção do sistema é referente a veracidade dos dados apresentados aos usuários, haverá um fluxo de moderação que abrangerá todos os cadastros de ocorrências da plataforma, ou seja, todas as ocorrências que forem apresentadas ao usuário primeiramente serão analisadas por um time de moderadores.

Nesta seção comparamos dois sistemas já consolidados no mercado com a plataforma desenvolvida nesse trabalho, o CrimeSpot. No próximo capítulo será apresentada a revisão bibliográfica, explicando as tecnologias utilizadas para o desenvolvimento deste trabalho.

3 REVISÃO BIBLIOGRÁFICA

O *software* desenvolvido neste trabalho tem como objetivo fornecer dados relevantes referentes a crimes, que são cadastrados na plataforma pelos próprios usuários. Como, em alguns casos, não é necessário boletim de ocorrência para realizar esse cadastramento, uma equipe de moderadores é responsável por revisá-los e então disponibilizar os dados para acesso pelos demais usuários. Foi utilizada a linguagem *Java* para o desenvolvimento da plataforma *web*, e a comunicação com o servidor ocorre por meio do modelo Cliente-servidor.

Uma instância na AWS foi criada com sistema operacional *Ubuntu Server*, onde constam a aplicação e seu banco de dados PostgreSQL.

3.1 ARQUITETURA CLIENTE-SERVIDOR

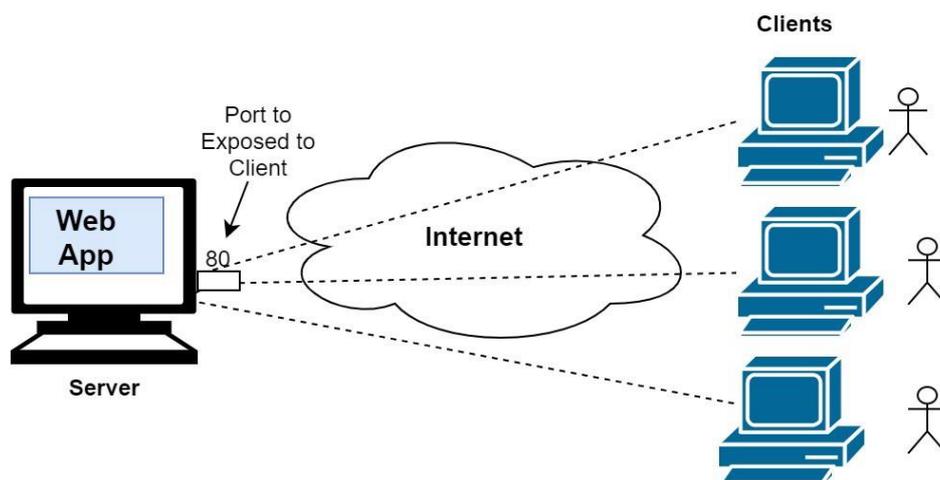
A arquitetura cliente-servidor foi utilizada neste trabalho por oferecer maior segurança devido aos seus serviços serem centralizados. Conforme tradução realizada pelo autor no site Britannica (2021)

“A arquitetura cliente-servidor é uma arquitetura onde muitos clientes fazem requisições e recebem serviço de um servidor centralizado. Os computadores cliente possuem uma interface que os permite realizar requisições ao servidor e disponibilizar os resultados retornados. O servidor fica constantemente aguardando requisições de clientes e, após receber alguma, envia uma resposta. [...] Desta forma, muitos clientes podem acessar recursos simultaneamente e, o servidor pode realizar suas tarefas internas sem interferir nos clientes (Britannica, 2021).”

Este modelo possui dois lados, o cliente e o servidor. O lado do cliente é responsável pelo envio das requisições e podem ser navegadores *web*, por exemplo, já o lado do servidor é responsável por enviar as respostas das requisições solicitadas pelo cliente. O lado do servidor realiza operações como autenticação e acesso a dados no banco de dados.

Na Figura 3 é possível visualizar o modelo cliente-servidor, onde vários clientes fazem requisições a um mesmo servidor.

Figura 3 - Arquitetura Cliente-Servidor.



Fonte: Medium, 2020.

3.1.1 Web Service

Conforme Hansel *Et Al.* (2020),

“Um Web Service é independente de implementação ou plataforma [...] e pode ser descoberto por um mecanismo padrão. Portanto, pode ser acessado através de uma Application Program Interface (API) através da rede, construída ao redor de outros serviços.”

Os *Web services* utilizados neste trabalho são o Google Maps Javascript API e o Google Geocode API. Conforme o *site Developers* (2021), “A Google Maps Javascript API permite criar um mapa com conteúdo customizado para ser exibido em uma página *web* ou dispositivos móveis.” Esta tecnologia é utilizada no mapa de ocorrências do sistema CrimeSpot e permite que sejam inseridos marcadores nas localizações de ocorrência, além de exibir os detalhes ao clicar nos marcadores.

A Google Geocode API possibilita obter as coordenadas geográficas a partir de um endereço ou vice-versa. Neste trabalho ela é utilizada para converter os endereços informados pelo usuário em coordenadas geográficas. Estes dados serão utilizados posteriormente no mapa, e para cálculos de distância no sistema (Developers, 2020).

3.1.2 Servidor

O conceito de servidor pode ser utilizado para referenciar tanto a máquina física quanto o *software*. De acordo com Jaswal (2020) em tradução realizada pelo autor, “um servidor é um programa ou *hardware* cujo trabalho é aceitar requisições de clientes e retornar os dados ou serviço como resposta desta requisição através da rede (*LAN* ou *WAN*).”

Analisando a definição acima é possível perceber que o termo servidor é um pouco genérico, e pode representar vários tipos de servidores que realizam tarefas diferentes.

Neste trabalho será utilizado o termo servidor para referenciar à máquina (*hardware*) que irá hospedar a aplicação e para outros tipos de servidores serão

respeitadas suas definições referentes a *software* pelos seus nomes mais específicos, por exemplo, servidor *web*, servidor DNS, servidor de banco de dados, etc.

Para a hospedagem do servidor foi utilizado a plataforma de computação em nuvem Amazon Web Services (AWS) que é “a plataforma de nuvem mais adotada e abrangente do mundo” (Amazon, 2020).

É utilizado o tipo de hospedagem Amazon Elastic Computer Cloud (EC2) que possui como vantagem além das altas possibilidades de customização da instância, uma interface web para gerenciamento do servidor e um firewall integrado para bloquear requisições indesejadas.

O sistema operacional (SO) utilizado é o Ubuntu *server*. De acordo com o site oficial é um sistema operacional leve e robusto, com excelente custo-benefício, que possui suporte com melhorias de *hardware* e manutenção por 5 anos após o lançamento de uma versão e uma enorme comunidade de usuários que contribui com sugestões e soluções para problemas. Ainda há outros pontos de destaque como a facilidade de integração com Docker e os pacotes *snap* que facilitam a manutenção e reduzem as tarefas administrativas (Techrepublic, 2017).

3.1.3 Servidor Web

De acordo com o *site* do Mozilla, um servidor *web* contém vários componentes que controlam como os usuários acessam os arquivos hospedados (armazenados para disponibilização), e um servidor HTTP. O servidor HTTP é um software que entende endereços *web* (URL) e HTTP (o protocolo que o navegador utiliza para visualizar páginas web).

O servidor *web* utilizado é o Tomcat que vem por padrão embutido em aplicações spring boot. Apesar de possuir maior custo de performance em relação a um Tomcat totalmente separado, ao utilizá-lo de forma embutida é possível deixar aplicações mais independentes, por exemplo, é possível desligar ou reiniciar uma aplicação sem afetar outras. Ainda é possível dedicar recursos de forma mais específica, dependendo das necessidades da aplicação, ambos os casos não são possíveis em um Tomcat completamente separado (ROUSE, 2019).

Devido à alta independência providenciada pelo Tomcat embutido, ele pode ser facilmente movido para contêineres Docker onde é possível aumentar a escala da aplicação com custo reduzido em relação a versão separada (ROUSE, 2019).

3.1.4 Server-Side Render

É a habilidade de uma aplicação poder apresentar uma página *web* no servidor ao invés de renderizá-la no *browser*. (EDUCATIVE, 2021, tradução do autor).

O principal ganho ao utilizar essa arquitetura é a velocidade de carregamento das páginas web que proporciona uma melhor experiência ao usuário. Mesmo se o usuário estiver utilizando uma máquina mais antiga, ou quando ele possui uma internet não tão veloz, é possível observar uma melhor performance ao se utilizar o SSR.

Uma outra vantagem que pode ser observada é a maior relevância da aplicação dentro dos motores de busca como Google, por exemplo, que dão preferência nos resultados de pesquisas a páginas com carregamento mais rápido.

3.2 APLICAÇÃO WEB

Uma aplicação *web* é um sistema que pode ser acessado através de um *browser* conectado à internet.

Conforme SEIBEL (2008) explica, “aplicações *web* são sistemas que executam em ambientes distribuídos, onde as partes do sistema podem executar em máquinas diferentes comunicando-se via protocolo HTTP ou HTTPS.”

As aplicações *web* rodam em um servidor que gera conteúdo HTML e é apresentado através dos *browsers* ou navegadores (SEIBEL, 2008).

3.2.1 Java

Java é uma linguagem de programação criada por James Gosling da empresa Sun em 1991. O objetivo da linguagem é escrever um programa apenas uma vez e então rodar este programa em múltiplos sistemas operacionais.

O Java deve rodar independente de plataforma. Programas Java rodam na máquina virtual Java e realizam acesso direto ao sistema operacional. Isto permite que o mesmo programa rode em vários sistemas (VOGELLA, 2017).

A linguagem é orientada a objetos, com exceção dos tipos primitivos, todos os elementos em Java são objetos. Ela também é estática e fortemente tipada, isso significa que os tipos de variáveis devem ser predefinidos e as conversões entre tipos são limitadas, devendo ser feitas na maioria das vezes pelo próprio programador (VOGELLA, 2017).

O código Java é compilado para bytecode e então interpretado pela máquina virtual Java, seu gerenciamento de memória é automático e todas as operações de alocação e liberação de memória são gerenciadas pelo próprio Java (VOGELLA, 2017).

As características mencionadas fazem do Java uma linguagem robusta e performática. Por mais que a verbosidade torne o desenvolvimento mais lento, o código escrito é mais inteligível e previsível, facilitando a análise e identificação de problemas.

3.2.2 Spring Boot

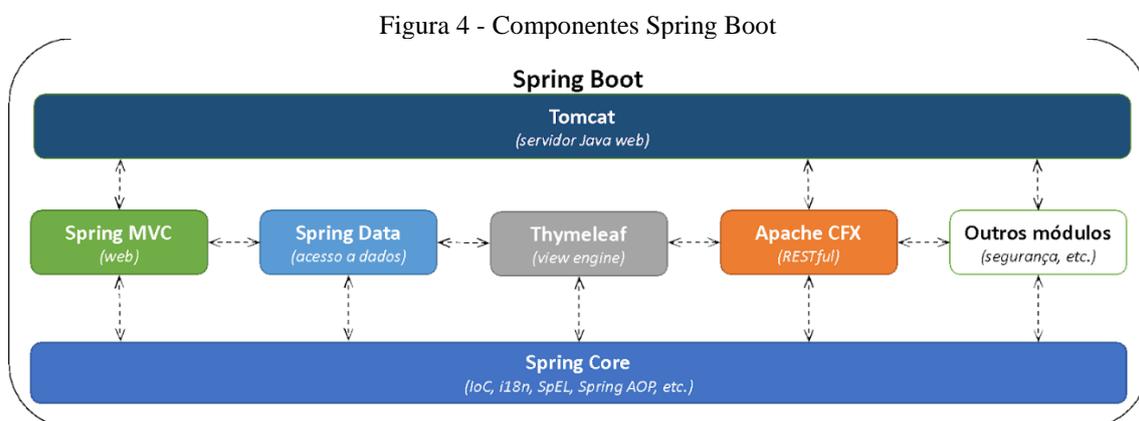
O *framework* escolhido para desenvolvimento foi o Spring boot, que possibilita iniciar de forma rápida o desenvolvimento da aplicação, enquanto providencia ferramentas para auxiliar desde o desenvolvimento, até os testes e publicação da aplicação.

De acordo com a documentação oficial, o *framework* Spring possibilita a construção rápida e sem dificuldade de aplicações *web* removendo grande parte de código *boilerplate* e configurações.

A documentação ainda diz que o Spring boot é construído em volta do Spring e busca subir e rodar uma aplicação de forma rápida e sem esforço. Isto é feito através da

inserção de um servidor *web* embutido, como o Tomcat e a inclusão de pacotes de terceiros que são automaticamente configurados e estão prontos para uso (SPRING, 2021).

A Figura 4 representa os componentes englobados e configurados automaticamente pelo Spring boot.



Fonte: Geek Hunter, 2019.

3.2.3 Thymeleaf

Para auxiliar na geração de páginas do sistema com conteúdo dinâmico vindo do servidor *web*, foi utilizado o motor de *templates* Thymeleaf.

De acordo com a documentação oficial do Expressjs, um motor de *templates* permite que sejam utilizados arquivos de *template* em uma aplicação. Esse arquivo contém uma representação de uma página HTML, com partes estáticas e, possivelmente algumas variáveis.

É responsabilidade do motor de *templates*, em tempo de execução, converter os arquivos mencionados acima para arquivos HTML e substituir as variáveis pelos valores passados como parâmetro pelo servidor *web*. O resultado destas operações é a página HTML que será enviada para o cliente (EXPRESSJS, 2020).

O Thymeleaf é um motor de *templates* para Java que possui o objetivo de criar *templates* elegantes e naturais para o *workflow* de desenvolvimento de uma aplicação. Diferente dos demais motores, o Thymeleaf possibilita a criação de arquivos de *template* bem similares ao seu produto final em HTML, de forma que é possível utilizar o arquivo, mesmo antes de ser processado pelo Thymeleaf, como um protótipo para visualização e/ou discussão de uma tela do sistema (THYMELEAF, 2018).

A Figura 5 mostra um exemplo de arquivo de *template* escrito para ser processado pelo Thymeleaf.

Figura 5 - Trecho de código utilizando Thymeleaf

```

1 <table>
2   <thead>
3     <tr>
4       <th th:text="#{msgs.headers.name}">Name</th>
5       <th th:text="#{msgs.headers.price}">Price</th>
6     </tr>
7   </thead>
8   <tbody>
9     <tr th:each="prod: ${allProducts}">
10      <td th:text="${prod.name}">Oranges</td>
11      <td th:text="${#numbers.formatDecimal(prod.price, 1, 2)}">0.99</td>
12    </tr>
13  </tbody>
14 </table>

```

Fonte: Thymeleaf, 2018.

Na Figura 5, é possível notar o conceito de naturalidade do Thymeleaf pois, as customizações do *template* foram feitas sem adicionar estruturas não contidas na linguagem HTML de forma que, o arquivo poderá ser utilizado como um protótipo para discussão.

3.2.4 PostgreSQL

Para armazenar os dados do sistema, é utilizado o Sistema de Gerenciamento de Banco de Dados (SGBD) PostgreSQL, também conhecido como Postgres.

Este SGBD foi criado com o objetivo de ser altamente extensivo e respeitar os padrões SQL. O Postgres é capaz de realizar diversas tarefas de forma eficiente graças a sua implementação de Controle de Concorrência Multiversão que garante atomicidade, consistência, isolamento e durabilidade (ACID) em suas transações (Ostezer e Drake, 2019).

Além de garantir a integridade dos dados, conforme Ostezer e Drake (2019) informam, “o Postgres consegue utilizar múltiplos processos para realizar suas consultas o que, combinado ao forte suporte para concorrência de escrita o faz uma boa escolha para operações complexas.”

Neste trabalho o Postgres é utilizado principalmente para escrita e armazenamento dos dados. Também é utilizado para consultas de leitura mais simples como, por exemplo, recuperar uma entidade pelo seu identificador.

3.2.5 Hibernate Search

Para entender o Hibernate Search é necessário primeiro conhecer o Apache Lucene que é “uma biblioteca de busca em texto completo, de alta performance, *open source*, escrita inteiramente em Java” (LUCENE, 2021, tradução do autor).

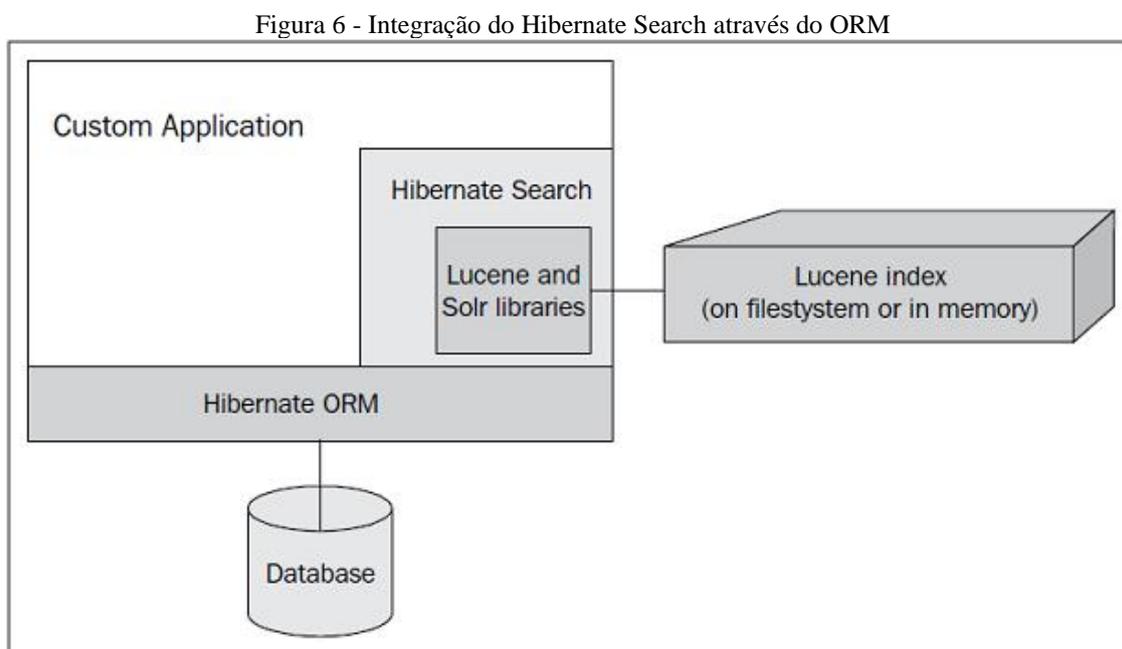
“Esta biblioteca possui poucos requerimentos de hardware e consegue realizar indexação de forma rápida e escalável. Quanto as consultas, possui algoritmos de busca poderosos, precisos e eficientes, entre eles, busca em campos de uma entidade, ordenação por qualquer campo da busca, muitos tipos poderosos de busca como, por exemplo,

buscas de proximidade, buscas em intervalo, busca em frases, etc.” (LUCENE, 2021, tradução do autor).

Apesar de todas as suas vantagens, a manutenção dos índices do Lucene é uma tarefa complexa, principalmente na questão da sincronização com o SGBD. Por isso foi utilizado o Hibernate Search que permite utilizar todas as funcionalidades do Lucene e ainda realiza a manutenção dos índices de maneira automática através de anotações *Java*.

Devido a sua excelente performance e facilidade de utilização, o Hibernate Search é utilizado para todas as consultas de leitura mais complexas do sistema, onde é necessário realizar buscas em várias entidades e realizar cálculos de distância ou intervalo como, por exemplo, o mapa de crimes, e as buscas em listas de entidades.

A Figura 6 mostra a integração do Hibernate Search em uma aplicação através da ORM do Hibernate.



Fonte: DevMedia, 2021.

3.3 DOCKER

De acordo com a documentação oficial, um contêiner é uma unidade padrão de software que contém o código e as dependências de uma aplicação de forma que esta possa rodar de maneira rápida e confiável em vários ambientes de computação. Contêineres isolam o software do ambiente e garantem que ele irá funcionar da mesma forma independente do ambiente em que está configurado (DOCKER, 2021).

O Docker é a ferramenta responsável por criar, rodar e utilizar os contêineres. Ele é similar a uma máquina virtual, porém ao invés de criar um novo sistema operacional para rodar uma aplicação, ele utiliza o mesmo *Kernel* que o sistema em que está rodando,

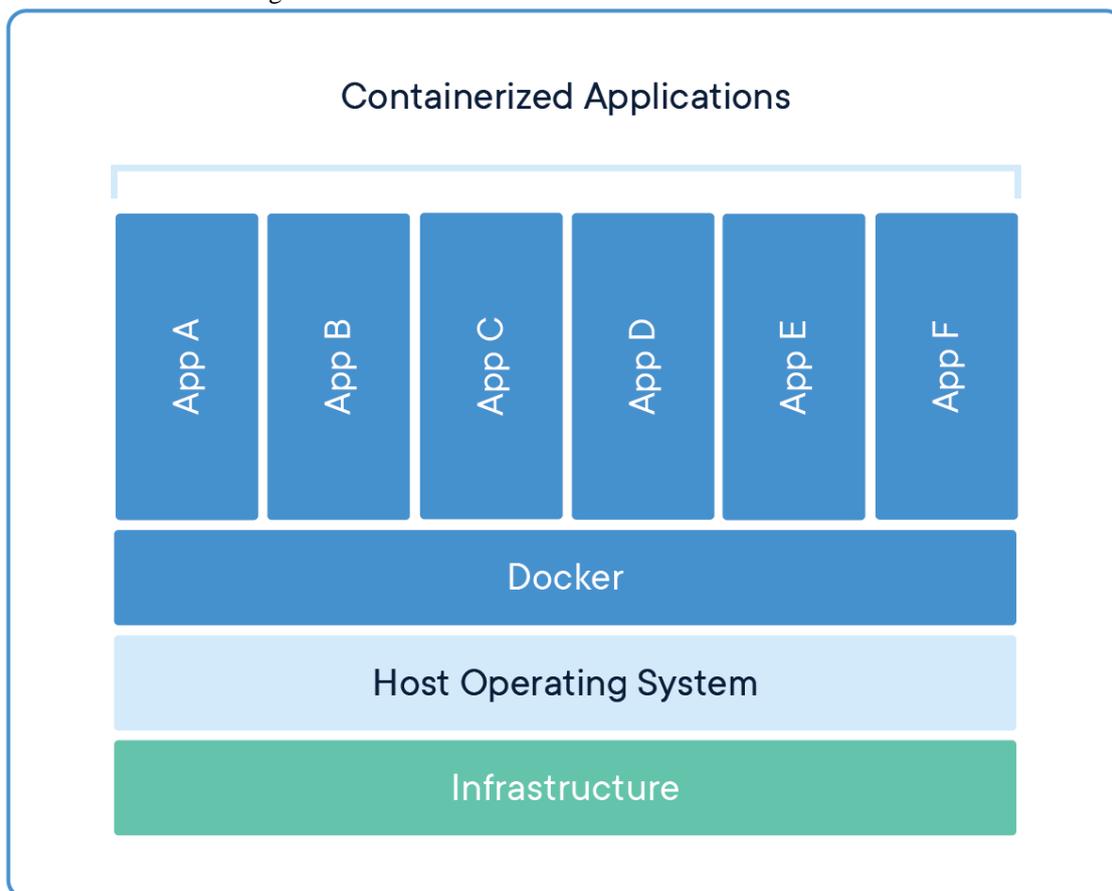
gerando um aumento de performance e tamanho reduzido da aplicação (OPENSOURCE, 2021).

Um exemplo de caso de uso muito comum que o *Docker* resolve é o famoso 'na minha máquina funciona'. Este problema normalmente é gerado por diferenças entre os ambientes de desenvolvimento, homologação e produção porém, ao utilizar o *Docker*, o ambiente onde roda uma aplicação é isolado, eliminando estas diferenças e consequentemente os problemas gerados por elas (DOCKER, 2021).

Neste trabalho foi utilizado Docker para prevenir que problemas relacionados a ambiente não interfiram no funcionamento da aplicação e também possibilitar a integração com sistemas de Integração Contínua (CI) de forma fácil e com pouca configuração.

A Figura 7 representa a forma que os contêineres e o Docker funcionam dentro de um sistema operacional.

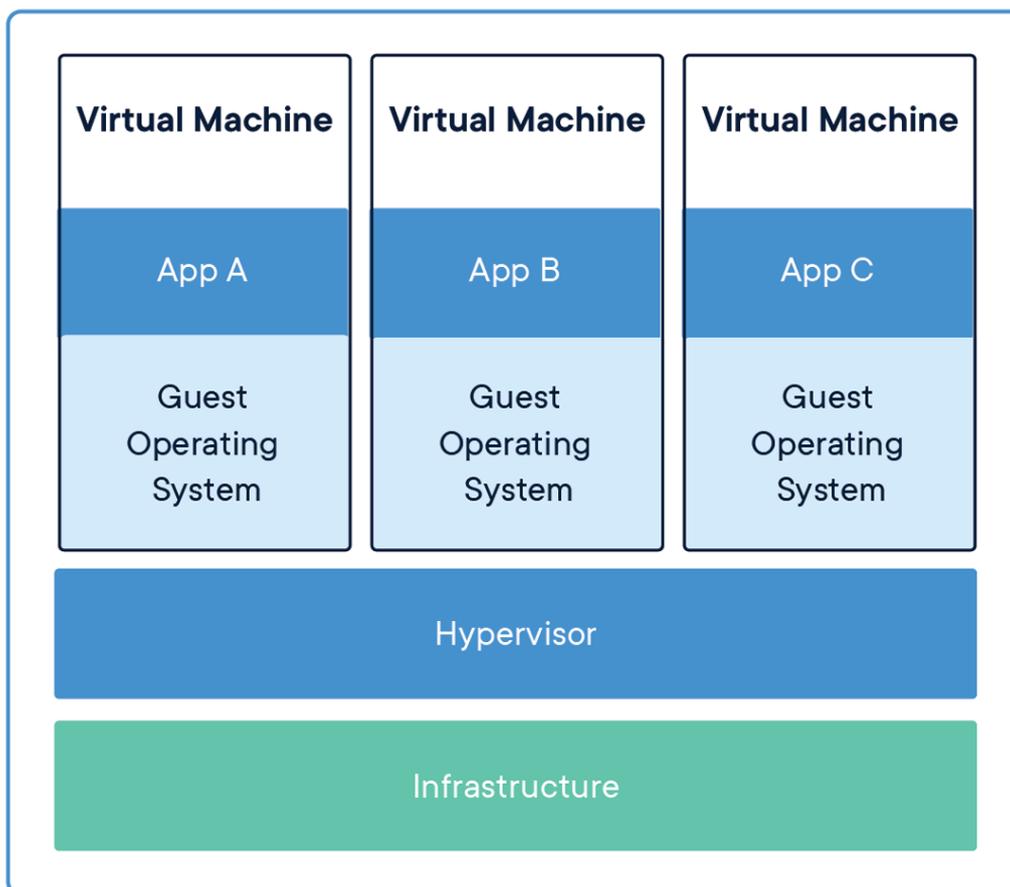
Figura 7 - Funcionamento dos contêineres dentro de um SO



Fonte: DOCKER, 2021.

A Figura 8 representa um ambiente similar utilizando máquina virtual.

Figura 8 - Ambiente utilizando máquinas virtuais



Fonte: DOCKER, 2021.

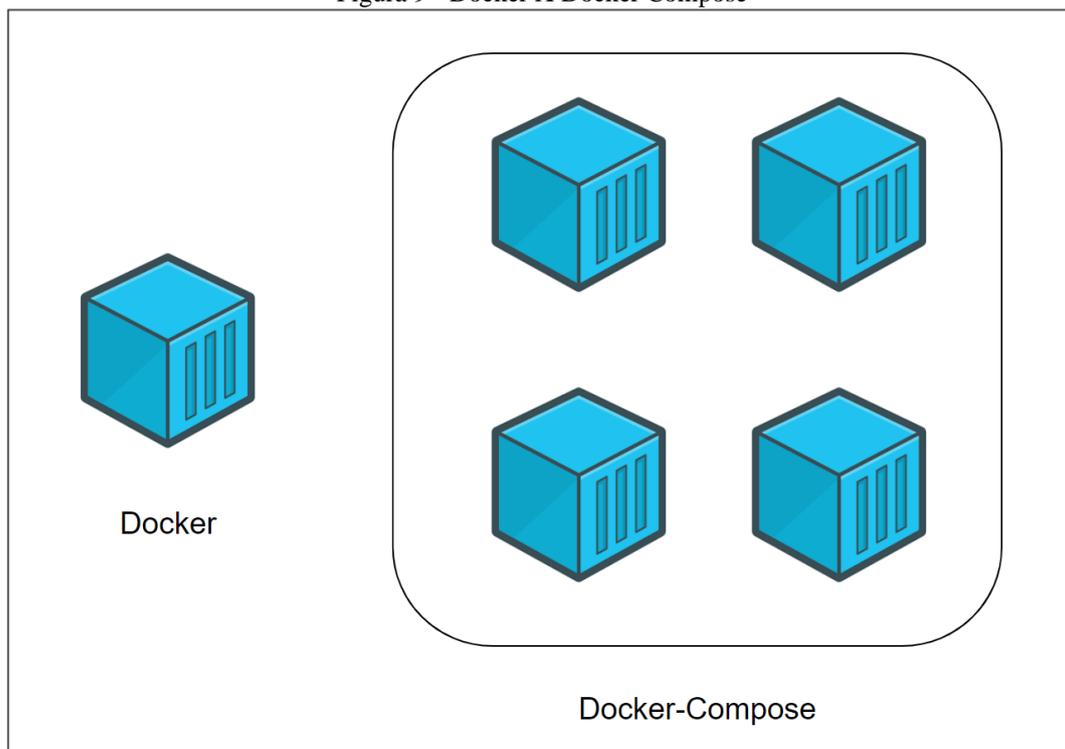
3.3.1 Docker Compose

“Compose é uma ferramenta para definir e rodar aplicações Docker de múltiplos contêineres. Basta utilizar um arquivo YAML para configurar os serviços da aplicação e então, com um comando, todos os serviços desta configuração são criados e iniciados” (DOCS.DOCKER, 2021, tradução do autor).

Um dos casos de uso do Compose, que é o motivo por ele ser utilizado neste trabalho, é a "redução do tempo de configuração do ambiente de desenvolvimento. Com apenas um comando é possível ter um ambiente de desenvolvimento pronto para uso (DOCKER, 2021).

A Figura 9 representa a estrutura do Docker Compose em relação ao Docker.

Figura 9 - Docker X Docker Compose



Fonte: MEDIUM, 2021.

De forma convencional, toda vez que um novo ambiente de desenvolvimento fosse criado, seria necessário rodar uma série de passos como instalar o Java, configurar o Java, instalar dependências etc. O Compose faz isso automaticamente de forma que o desenvolvedor não necessita conhecer todos os conceitos de configuração para iniciar seu projeto.

Neste capítulo foram mostradas as tecnologias utilizadas para o desenvolvimento do CrimeSpot. No capítulo a seguir será apresentado o tipo de pesquisa utilizado, a metodologia escolhida para o desenvolvimento e como as tecnologias foram aplicadas.

4 METODOLOGIA

Para guiar este trabalho inicialmente foi realizada uma pesquisa exploratória em trabalhos e sistemas semelhantes ao que foi desenvolvido neste trabalho. Esta pesquisa foi importante para auxiliar o autor a entender o que os usuários de um sistema de mapeamento criminal necessitam e quais funcionalidades poderiam lhes oferecer maior benefício. Nesta etapa inicial também foi importante definir qual a metodologia de desenvolvimento seria utilizada e optou-se por adotar a versão do *Scrum solo*, que é uma adaptação do *Scrum* para poder ser utilizado por apenas uma pessoa.

O *Scrum* é uma metodologia ágil para gestão e planejamento de projetos, comumente presente no setor de desenvolvimento de software e possui como característica ser iterativo. Com esta metodologia os projetos são separados em ciclos, denominados Sprints e costumam durar entre uma semana a um mês, permitindo entregas de partes do projeto (PAGOTTO *et al*, 2016).

O *Scrum solo* é versão do *Scrum* adaptada para que apenas um programador possa trabalhar no sistema e manter alguns conceitos e cerimônias do *Scrum* em sua rotina de desenvolvimento. Alguns dos conceitos e cerimônias do *Scrum* que podem ser seguidos quando se escolhe seguir o *Scrum solo* são:

- *Product Backlog*: Lista de funcionalidades a serem desenvolvidas.
- *Sprint Backlog*: Itens do *Product Backlog* que foram selecionados para serem desenvolvidos dentro da *Sprint*.
- *Sprint*: Período definido para um ciclo de trabalho. A convenção mostra que duas semanas é o tempo ideal de duração de uma *Sprint*.
- *Sprint Retrospective*: É a cerimônia que ocorre ao final de uma *Sprint* onde se avaliam os resultados entregues e como foi o andamento da *Sprint*.

O diagrama de casos de uso foi utilizado para mostrar os atores e funcionalidades do sistema, para que fosse possível entender como iriam se relacionar. O diagrama entidade relacionamento foi criado para apresentar de maneira mais detalhada as entidades, seus relacionamentos e como os dados do sistema são persistidos na base de dados.

A linguagem escolhida para o desenvolvimento do sistema é o Java com *framework* Spring Boot, baseado no padrão *Model, View e Controller*, que separa as camadas da aplicação auxiliando na organização do projeto.

Foi utilizado também o motor de templates Thymeleaf, estes templates são escritos, em sua maioria, com código HTML5 e tem boa integração com o Spring framework, além da capacidade de processar código CSS e JavaScript (AlgaWorks, 2019).

A IDE *IntelliJ IDEA*, uma interface de desenvolvimento completa da *JetBrains* serve como base para escrita do código. Já para o versionamento foi utilizado o *Git*, um sistema de controle de versões que permite guardar o histórico de todas as alterações de um código, sendo possível recuperar versões antigas em caso de problemas.

Para a escrita de testes funcionais da aplicação foi utilizado o *framework* *FluentLenium* que se trata de uma estrutura de automação de *sites* que estende o Selenium para escrever testes funcionais da UI legíveis, reutilizáveis, confiáveis e resilientes (FluentLenium, 2019).

Após a finalização do desenvolvimento do sistema, ele foi hospedado no servidor para que fosse possível realizar os testes de aceitação com os usuários. Estes testes auxiliaram a garantir a qualidade do sistema e também foram fonte importante de *feedbacks* que resultaram em melhorias na aplicação.

Assim que as últimas correções foram finalizadas, a nova versão da aplicação foi publicada na Amazon (AMAZON, 2021).

Neste capítulo foram apresentadas as metodologias de pesquisa e desenvolvimento aplicadas neste trabalho, e também as tecnologias utilizadas. No próximo capítulo é apresentada a modelagem do sistema, suas funcionalidades e detalhes sobre testes de aceitação.

5 DESENVOLVIMENTO

Este capítulo tem por objetivo apresentar todas as etapas do desenvolvimento do sistema CrimeSpot, desde sua modelagem, até seu desenvolvimento e testes de aceitação. Serão apresentadas todas as telas da aplicação e suas funcionalidades, bem como os resultados dos testes realizados com usuários.

5.1 MODELAGEM

Quando o trabalho CrimeSpot começou a ser idealizado, logo percebeu-se a necessidade de documentar essas ideias para que fosse possível entender as necessidades do sistema. Assim foi iniciada a escrita do Documento de Visão que fornece uma visão de alto nível do produto. Após a escrita deste documento, o Diagrama de casos de uso foi criado para fornecer um melhor entendimento dos atores do sistema, as funcionalidades e como esses atores se relacionam com as funcionalidades. Outro diagrama que auxiliou na especificação do projeto foi o diagrama ER, que ilustra todos os dados que serão persistidos na aplicação.

5.1.1 Documento de Visão

O Documento de Visão apresenta uma descrição sobre o objetivo do sistema, seus usuários, funcionalidades e regras de negócio. Reunindo estes dados em um documento foi possível ajustar alguns fluxos que não estavam muito claros no momento da idealização do trabalho.

A partir da escrita do Documento de Visão a descrição do projeto, seus usuários-alvo e diferenciais em relação aos demais do mercado se tornaram mais claros para todos os envolvidos.

O sistema CrimeSpot foi identificado como uma possível solução para o problema da insegurança das pessoas ao transitar pela cidade devido a ser um sistema de mapeamento criminal com foco na questão da colaboração entre a comunidade, que alimenta a base de dados e fornece informações para todos os usuários que acessarem a aplicação. Contando, assim, com situações passadas pela própria comunidade para auxiliar a prevenir que outras pessoas passem pelo mesmo problema.

Com o conhecimento de possíveis pontos perigosos, a intenção do sistema é evitar que as pessoas transitem por locais inseguros em determinados horários, conforme dados apresentados no CrimeSpot. E também disponibilizar dados confiáveis para a comunidade sobre ocorrências registradas próximas a sua localização. No Documento de

visão foram definidas as partes interessadas no desenvolvimento de um sistema criminal colaborativo, são elas:

- **Usuário:** Utilizam o sistema para visualizar e filtrar ocorrências no mapa de ocorrências.
- **Usuário autenticado:** Este usuário possui cadastro no sistema e será responsável por alimentar a base de dados com ocorrências.
- **Moderador:** Deverá alterar status de ocorrências cadastradas por usuários autenticados e gerenciar estes usuários.
- **Administrador:** O Administrador é responsável por gerenciar moderadores e, como o moderador, alterar o status de ocorrências registradas por usuários autenticados. Além disso, o administrador pode desativar ou revogar aprovação de qualquer ocorrência.
- **Time de desenvolvimento:** Será responsável por elaborar os documentos referentes ao sistema e realizar todo o desenvolvimento e testes da aplicação.

No Documento de Visão também consta a descrição do produto, informando o que ele é, o que ele faz e como se diferencia de um de outro aplicativo semelhante. Uma característica do CrimeSpot que o diferencia do CrimeMapping, por exemplo, é a redução da burocracia para inserção e acesso aos dados, porém com um sistema de moderação que reduz a possibilidade de serem exibidos dados inverídicos.

Além destes dados já citados, o documento de visão também descreve as funcionalidades do sistema, seus atores, restrições funcionais e tecnológicas. O documento pode ser verificado no Apêndice A deste trabalho.

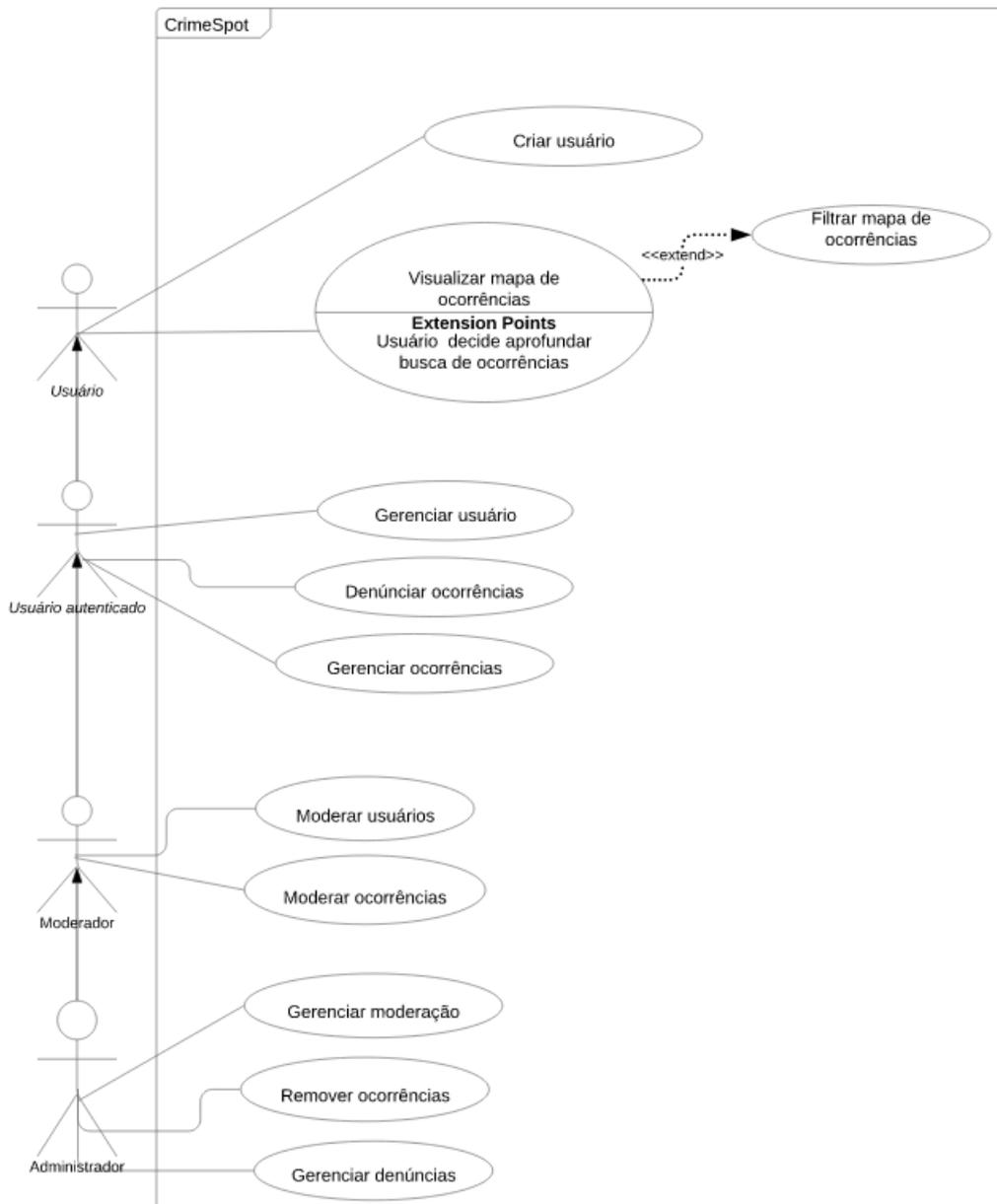
5.1.2 Diagrama de Casos de Uso

Uma descrição visual sobre os atores do sistema e as funcionalidades foi desenhada utilizando o aplicativo de diagramação inteligente LucidChart (LucidChart, 2021). Com o auxílio desta ferramenta ficou mais simples entender como cada usuário interage dentro do sistema e com quais funcionalidades ele interage.

A notação UML foi utilizada para traduzir as necessidades do projeto em usuários e funcionalidades. Conforme o *site* The Club (2021), o UML “[...]É uma linguagem de diagramação ou notação para especificar, visualizar e documentar modelos de softwares orientados a objetos”. Ainda conforme o The Club (2021) “O diagrama de casos de uso é o mais utilizado dentre os demais diagramas da UML, seu objetivo é ilustrar uma funcionalidade ou requisito do sistema do ponto de vista do usuário” e por isso esse diagrama foi escolhido para auxiliar na modelagem do sistema.

Na Figura 10 é possível verificar o Diagrama de Casos de Uso criado para o aplicativo *web* CrimeSpot.

Figura 10 - Diagrama de Casos de Uso CrimeSpot

Diagrama de casos de uso CrimeSpot

Fonte: Próprio autor.

Conforme é possível verificar pelo diagrama apresentado na Figura 10, existem quatro atores no sistema. Os atores usuários podem acessar o sistema para visualizar e filtrar ocorrências conforme suas necessidades, já os usuários autenticados são aqueles que criaram cadastro no CrimeSpot para poderem popular a base de dados com ocorrências. Os moderadores analisam as ocorrências criadas pelos usuários autenticados e podem decidir se estas ocorrências serão disponibilizadas na aplicação ou se serão removidas. Já o ator Administrador é responsável por gerenciar os usuários e suas permissões e pode desativar qualquer ocorrência.

O caso de uso Denunciar ocorrências permite que os usuários reportem ocorrências com conteúdo impróprio, porém apenas se estiverem autenticados no sistema. Essas

denúncias serão avaliadas pelo usuário administrador, cujo fluxo está representado no caso de uso Gerenciar Denúncias. O Administrador poderá definir se a ocorrência denunciada será removida do sistema ou mantida no ar.

O caso de uso Gerenciar Moderação envolve conceder ou remover a permissão de moderação e também revogar uma aprovação de ocorrência realizada anteriormente. É possível transformar um usuário autenticado em um moderador ou um moderador pode voltar a ser um usuário autenticado, perdendo assim alguns privilégios dentro do sistema. Remover ocorrências abrange o privilégio do administrador em poder desativar qualquer ocorrência independente do status.

Moderar ocorrências envolve a análise de uma ocorrência para definir se ela será publicada no sistema, ou se será removida. Ocorrências removidas não poderão mais ser acessadas ou publicadas. Já o caso de uso Moderar Usuários envolve a desativação ou reativação de cadastros de usuários, conforme necessidade. Este caso de uso é importante para casos em que usuários possam estar inserindo dados indevidos na plataforma repetidamente.

O caso de uso Gerenciar Usuário é o responsável pela criação, edição e remoção do cadastro pelo usuário que o criou e o caso de uso Gerenciar Ocorrência permite realizar as mesmas ações do Gerenciar Usuário, porém voltadas as ocorrências. Será possível criar, editar ou excluir uma ocorrência. As ocorrências cadastradas no sistema possuem status e o usuário poderá editar apenas ocorrências que possuem o status Aguardando Análise. A partir do momento que um moderador inicia a análise, a ocorrência não pode mais ser alterada, apenas excluída.

Criar Usuário envolve a criação de cadastro por usuários que ainda não realizaram este procedimento. O caso de uso Visualizar Mapa de Ocorrências possui um ponto de extensão para o Filtrar Ocorrências para que seja possível direcionar as buscas de ocorrências abrangidas pelo caso de uso de Visualizar Mapa de Ocorrências de acordo com os critérios do usuário.

5.1.3 Diagrama ER

Após a identificação dos atores do sistema e das funcionalidades, foi necessário entender como os dados seriam persistidos na base de dados e como seriam seus relacionamentos. Para auxiliar a organizar estes relacionamentos, foi construído o Diagrama ER utilizando a ferramenta LucidChart, também utilizada para criação do Diagrama de Casos de Uso.

Com o Diagrama ER é possível visualizar todas as entidades do sistema e seus atributos. Cada entidade possui uma lista de atributos que juntos formam uma tabela da base de dados. O diagrama mostra os relacionamentos entre essas tabelas e quais atributos, chaves primárias e estrangeiras, são utilizados nesses relacionamentos. Na Figura 11 é possível visualizar o Diagrama ER do sistema CrimeSpot e um exemplo de relacionamento pode ser observado entre as tabelas user e occurrence. As linhas que ligam estas duas entidades mostram que um usuário (tabela user) pode ter nenhuma, uma ou várias ocorrências criadas, e cada ocorrência (tabela occurrence) pertence a um user.

Tabela 2 - Entidades, chaves e relacionamentos

Nome da entidade	Chaves	Função no sistema	Relacionamentos
address	id PK	Armazena os endereços de usuários e ocorrências do sistema	Nenhum
File	id PK	Armazena os arquivos de ocorrências do sistema	Nenhum
occurrence	id PK user_id FK moderator_id FK address_id FK occurrence_document_id FK occurrence_status_id FK	Armazena as ocorrências do sistema	Possui relação com a tabela occurrence_status para recuperar o status da ocorrência. Possui relação com a tabela occurrence_type para recuperar o tipo da ocorrência. Possui dois relacionamentos com a tabela user, um para indicar o usuário que criou a ocorrência e outro para indicar o moderador daquela ocorrência. Possui um relacionamento com address para recuperar o endereço da ocorrência. Possui um relacionamento opcional com file para recuperar o arquivo que comprova a veracidade da ocorrência.
occurrence_status	id PK	Armazena os possíveis status de ocorrência do sistema	Nenhum
occurrence_type	id PK	Armazena os possíveis tipos de ocorrência do sistema	nenhum

Role	id PK	Armazena os possíveis papéis do sistema	Possuí um relacionamento através de user_roles para vincular papéis a usuários
User	id PK address_id FK	Armazena os usuários do sistema	Possui um relacionamento com address para recuperar o endereço do usuário. Possui um relacionamento através de user_roles para vincular papéis a usuários
user_roles	id PK roles_id PK FK	Armazena a associação de usuários e papéis do sistema, indicando quais papéis cada usuário possui	Possui um relacionamento com user e role para poder realizar o vínculo entre as duas tabelas
Reports	id PK user_id FK occurrence_id FK	Armazena as denúncias de ocorrências	Possui relacionamento com a tabela user e a tabela occurrence para vincular os dados de quem denunciou cada ocorrência.

Fonte: Próprio autor.

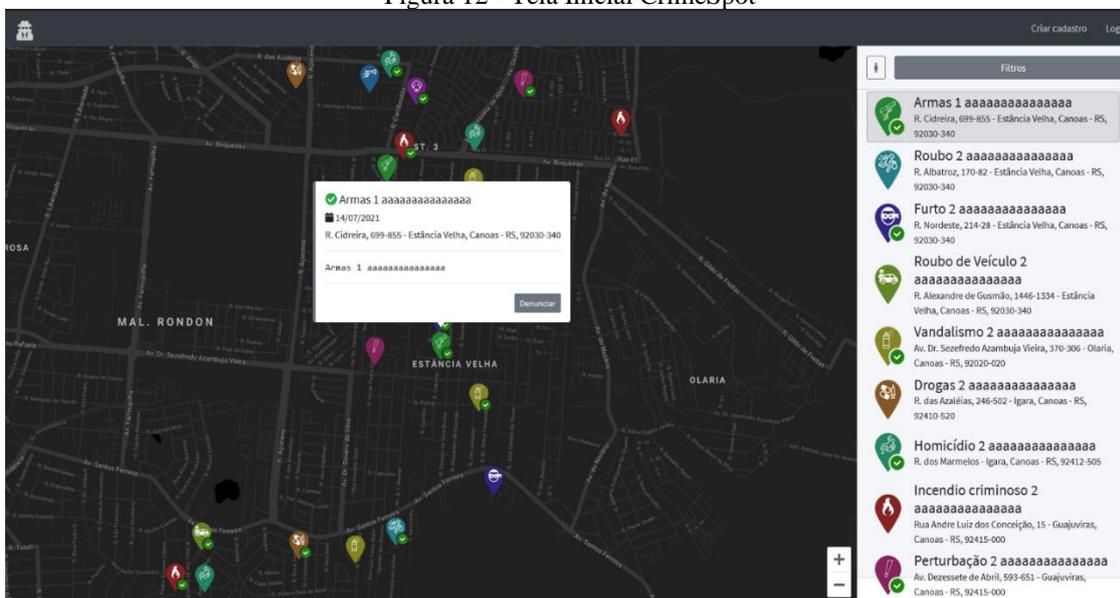
5.2 O CRIMESPOT

O CrimeSpot foi criado com a intenção de ser uma ferramenta de mapeamento criminal colaborativo onde os próprios usuários alimentam a base de dados e moderadores asseguram que os dados publicados sejam confiáveis. Nesta seção serão apresentadas as telas do sistema e explicações sobre seu funcionamento.

Ao acessar o sistema, a primeira tela que o usuário vê é a tela inicial apresentada na Figura 12 e que não necessita de login para ser visitada. Nesta tela é apresentado mapa de ocorrências contendo as ocorrências já verificadas pela moderação. Cada tipo de ocorrência possui um ícone diferente no mapa e as ocorrências que possuem boletim de ocorrências anexado mostram um *check* verde para diferenciá-las das demais.

O mapa apresenta 200 ocorrências simultâneas, caso esse limite seja atingido é necessário filtrar as ocorrências ou aumentar o *zoom*. Este limite foi definido durante o desenvolvimento para evitar “poluir” muito o mapa de ocorrências e acabar comprometendo a visualização das informações.

Figura 12 - Tela Inicial CrimeSpot



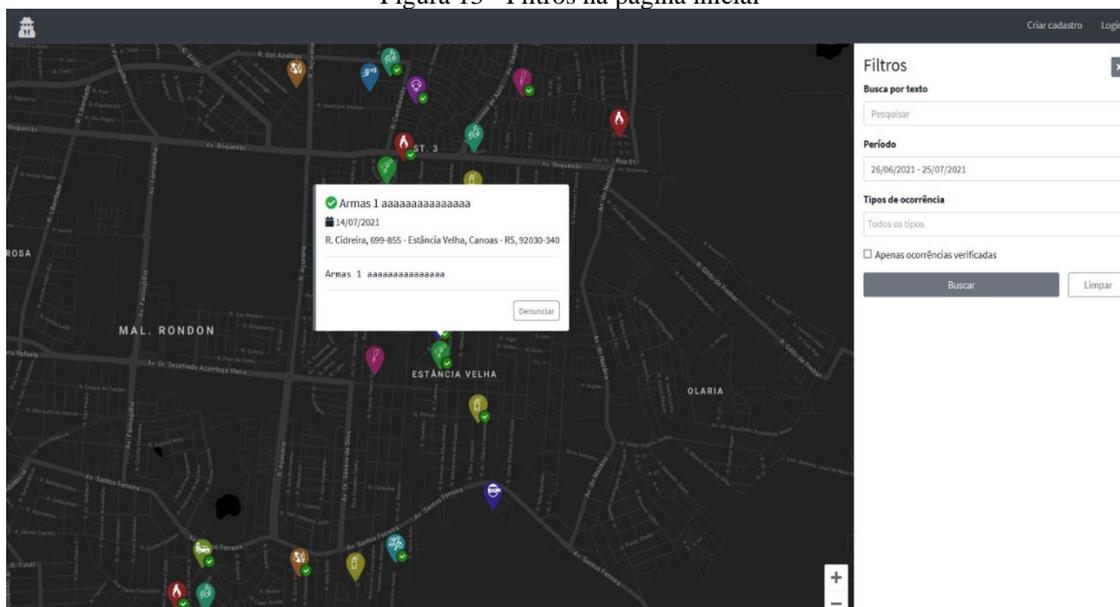
Fonte: Próprio autor.

Neste mapa é possível clicar com o *mouse* e arrastá-lo para as regiões que se deseja visualizar, cada tipo de ocorrência terá um ícone diferente para que seja simples identificá-lo no mapa. Os tipos de ocorrência que podem ser cadastrados no sistema são: roubo, furto, vandalismo, roubo de veículo, drogas, homicídio, incêndio criminoso, perturbação, tiroteio, maus tratos a animais e armas.

No lado direito da tela são apresentados alguns detalhes sobre as ocorrências apresentadas no mapa, clicando sobre elas seu ponto no mapa será evidenciado e mais algumas informações serão apresentadas. Outra maneira de visualizar mais detalhes sobre uma ocorrência é clicando sobre um ponto do mapa. Também do lado direito da tela é

possível verificar o botão de filtros, que possibilita que o usuário possa buscar as ocorrências por tipo, período de tempo ou busca textual. Outra opção interessante é a de restringir para apresentar apenas ocorrências verificadas, ou seja, aquelas que possuem boletim de ocorrência. Ao finalizar as escolhas nos filtros, basta clicar em **Buscar** e então o mapa apresentará as ocorrências conforme critérios informados pelo usuário. As opções de filtros estão sendo mostradas na Figura 13.

Figura 13 - Filtros na página inicial



Fonte: Próprio autor.

É interessante reforçar que para realizar os filtros no mapa de ocorrências não é necessário estar autenticado no sistema. Basta clicar no botão de filtros a partir da tela inicial e será possível acessar esta funcionalidade.

No menu superior direito consta a opção de *Login*. Quando o usuário clica neste local, ele é redirecionado para a tela de *login* do sistema, que está sendo apresentado na Figura 14.

Esta tela contém os campos para inserção de Usuário e Senha para o usuário inserir os dados e clicar em **Entrar** e acessar mais funcionalidades do sistema. Caso o usuário ainda não possua cadastro, ele pode clicar em “Não possui Conta? Criar Perfil” e será redirecionado para o formulário de criação de cadastro de usuário.

Figura 14 - Tela de login

A imagem mostra a tela de login do sistema CrimeSpot. No topo, há um ícone de um homem com um chapéu e uma máscara, seguido pelo nome "CrimeSpot". Abaixo, há um formulário branco com o título "Faça login para iniciar a sessão". O formulário contém dois campos de entrada: "Usuário" com um ícone de pessoa e "Senha" com um ícone de cadeado. Abaixo dos campos, há um botão cinza com o texto "Entrar". Na base do formulário, há um link azul que diz "Não possui conta? Criar Perfil".

Fonte: Próprio autor.

Caso credenciais incorretas sejam inseridas nos campos de *login*, uma mensagem de erro será apresentada e será possível inserir os dados novamente, caso estejam corretos o usuário será redirecionado para a tela inicial, e no canto superior direito constará seu nome de usuário ao invés da opção de Entrar.

Quando selecionar a opção de criar um novo perfil, o usuário é direcionado para o formulário de criação de cadastro que é mostrado na Figura 15. Este formulário apresenta os dados necessários para que um novo usuário possa ser registrado na base de dados do sistema e ter acesso as funcionalidades de usuário autenticado. Os campos obrigatórios estão marcados com um asterisco para que o usuário entenda quais dados são obrigatórios e quais dados são opcionais.

Após o usuário inserir as informações solicitadas no formulário e clicar em Salvar, seu cadastro será criado e ele poderá realizar login no sistema e criar ocorrências, por exemplo. Caso algum dado incorreto seja inserido nos campos do formulário e o usuário clica em Salvar, uma mensagem de erro será apresentada, o cadastro não será criado e ele poderá corrigir as informações e clicar em Salvar novamente.

Figura 15 - Formulário de cadastro de usuário

Dados de usuário	
Nome*	Nome de usuário*
<input type="text" value="John Doe"/>	<input type="text" value="john.doe"/>
Email*	Data de Nascimento*
<input type="text" value="john.doe@gmail.com"/>	<input type="text"/>
CPF*	Raio de Busca de Ocorrências(KM)
<input type="text"/>	<input type="text"/>
Senha*	Confirmar Senha*
<input type="text"/>	<input type="text"/>

Dados de endereço	
CEP*	País*
<input type="text" value="99999999"/>	<input type="text"/>
Estado*	Cidade*
<input type="text"/>	<input type="text"/>
Rua*	Número
<input type="text"/>	<input type="text"/>
Complemento	
<input type="text"/>	

Fonte: Próprio autor.

Também será possível acessar o formulário de criação de usuário ao clicar na opção Criar Cadastro do menu superior.

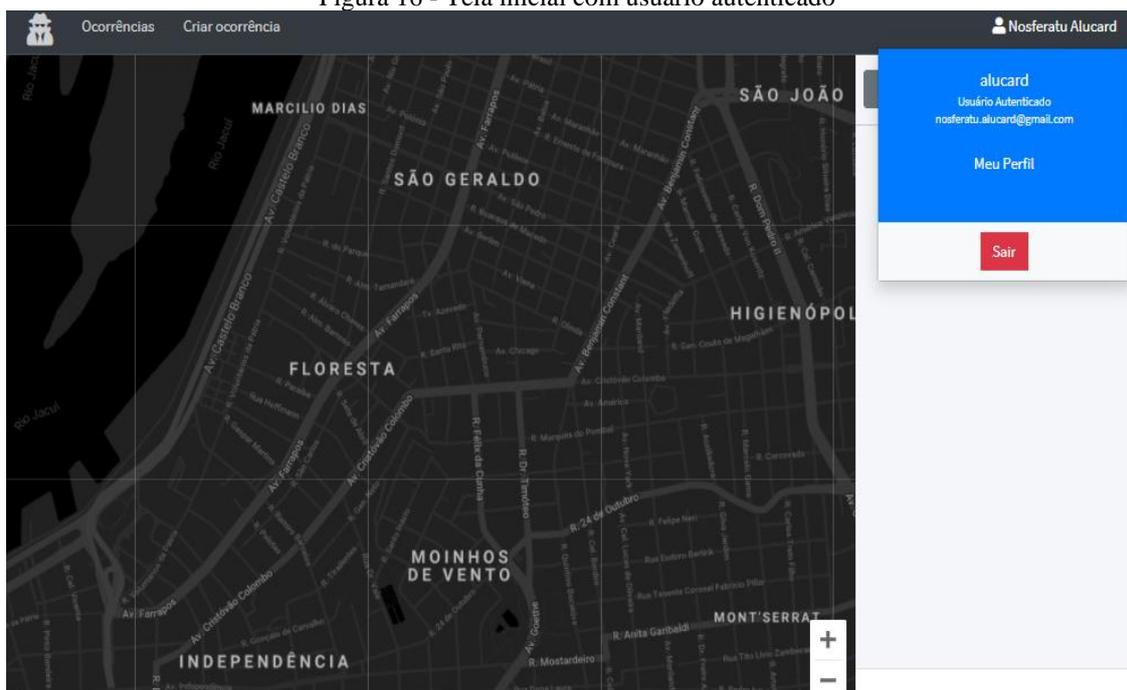
Após a criação do cadastro e autenticação no sistema, o usuário poderá clicar em seu nome no canto superior direito e algumas informações serão apresentadas, como o perfil do usuário (moderador, administrador ou usuário autenticado), o *e-mail* que o usuário inseriu no momento do seu cadastro, uma opção para acessar o seu cadastro e o botão Sair para realizar *logout* da aplicação.

No momento da criação do cadastro, após inserir o cep, o endereço é buscado na API da google. Este endereço precisa ser geolocalizado na API da google para que possa ser identificado e apresentado no mapa de ocorrências.

Essas opções podem ser verificadas na Figura 16, que mostra a visão de um usuário autenticado na tela principal do sistema. É possível notar, no menu superior, a opção de Ocorrências que é apresentada quando o usuário realiza login na aplicação.

O mapa de ocorrências é apresentado levando em consideração o endereço inserido pelo usuário no momento de seu cadastro.

Figura 16 - Tela inicial com usuário autenticado



Fonte: Próprio autor.

Quando o usuário clica na opção Meu Perfil dentro das opções de seu nome de usuário, ele é redirecionado para um formulário semelhante ao de criação de cadastro de usuário, porém todos os campos já constam populados com seus dados e ele poderá realizar alterações conforme a necessidade.

Na Figura 17 é apresentada a tela de edição de cadastro, e também as opções de Alterar Senha e Desativar perfil. Na opção de alteração de senha, o usuário poderá mudar sua senha sempre que veja necessidade, já a opção de desativar perfil permite que o usuário remova seu cadastro se assim julgar necessário, porém suas informações permanecerão na base de dados do sistema e o cadastro poderá ser reativado pelo administrador.

Figura 17 - Edição de cadastro de usuário

Dados de usuário		Alterar senha	Desativar perfil
Nome*	<input type="text" value="Nosferatu Alucard"/>	Nome de usuário*	<input type="text" value="alucard"/>
Email*	<input type="text" value="nosferatu.alucard@gmail.com"/>	Data de Nascimento*	<input type="text" value="26/06/1998"/>
CPF*	<input type="text" value="15861939020"/>	Raio de Busca de Ocorrências(KM)	<input type="text" value="3"/>
Dados de endereço			
CEP*	<input type="text" value="90220021"/>	País*	<input type="text" value="Brasil"/>
Estado*	<input type="text" value="Rio Grande do Sul"/>	Cidade*	<input type="text" value="Porto Alegre"/>
Rua*	<input type="text" value="Rua Almirante Barroso"/>	Número	<input type="text" value="735"/>
Complemento	<input type="text" value="Sala 799"/>		
<input type="button" value="Cancelar"/>		<input type="button" value="Salvar"/>	

Fonte: Próprio autor.

Na Figura 18 é apresentada a tela de alteração de senha, onde é possível modificar a senha do usuário logado sempre que necessário.

Figura 18 - Tela de Alteração de senha

Alterar senha		
Senha*	Nova Senha*	Confirmar nova senha*
<input type="text"/>	<input type="text"/>	<input type="text"/>
<input type="button" value="Cancelar"/>		<input type="button" value="Salvar"/>

Fonte: Próprio autor.

Outra opção disponibilizada ao usuário autenticado é o menu Ocorrências que está apresentado na Figura 19. Ao clicar nesta opção, o usuário é redirecionado para a tela de listagem de ocorrências com todas as ocorrências criadas por ele paginadas e listadas da mais recente para mais antiga. O usuário pode alterar o tipo de ordenamento caso necessário, uma opção muito interessante para os moderadores e administradores é o

ordenamento "mais próximas a mim" que ordena as ocorrências por proximidade ao endereço do moderador ou administrador.

Antes da lista é possível verificar as opções de filtros do sistema onde é possível definir mais critérios além de busca textual e tipo de ocorrência. Também é possível restringir a busca para retornar apenas ocorrências com um determinado status ou alterar a ordenação para apresentar a lista a partir das ocorrências mais antigas.

Os status que podem ser buscados através destes filtros são:

- **Aguardando análise:** Quando a ocorrência está aguardando ser analisada por um moderador ou administrador.
- **Em análise:** Quando o moderador ou administrador está realizando a análise dos dados inseridos na ocorrência.
- **Aprovada:** A ocorrência foi analisada e aprovada na análise do administrador ou moderador.
- **Recusada:** A ocorrência foi analisada, porém devido a conter algum dado indevido ou suspeito foi recusada pelo moderador e não será exibida no mapa de crimes.
- **Recusada pelo Administrador:** A ocorrência foi analisada e aprovada anteriormente pelo moderador, porém o administrador decidiu recusá-la e não será mais exibida no mapa de crimes.
- **Denúncia Aceita:** Quando a ocorrência foi denunciada por um ou mais usuários e o administrador aceitou as denúncias.

Figura 19 - Tela de listagem de ocorrências

The screenshot shows a web interface for listing occurrences. At the top right is a button labeled "Criar ocorrência". Below it are three filter sections: "Busca por texto" with a search input field containing "Pesquisar"; "Status da ocorrência" with a dropdown menu set to "Todos os status"; and "Ordenar por" with a dropdown menu set to "Mais recentes". Below these is a "Tipos de ocorrência" section with a dropdown menu set to "Todos os tipos" and a "Pesquisar" button. The main content area displays two occurrence cards. The first card is titled "Ocorrência em análise" and contains the text "Uma ocorrência em análise show de bola, tudo de bom." followed by details: "Criada por: alucard", "Cidade: Canoas", "Tipo: Furto", "Criada em: 22/mai/2021 22:41:32", "Moderador: admin", "Data da ocorrência: 22/mai/2021", "Rua: Avenida Farroupilha", "Status: Em análise", and "Atualizada em: 22/jun/2021 22:41:32". The second card is titled "Homicídio 4 aaaaaaaaaaaaaaaaa" and contains the text "Homicídio 4 aaaaaaaaaaaaaaaaa" followed by details: "Criada por: alucard", "Cidade: Canoas", "Tipo: Homicídio", "Criada em: 14/mai/2021 22:41:32", "Moderador: moderator", "Data da ocorrência: 14/mai/2021", "Rua: Rua Dona Rafaela", "Status: Aprovada", and "Atualizada em: 22/jun/2021 22:41:32". Each card has a "Ver detalhes" button at the bottom right.

Na tela de listagem, cada ocorrência é apresentada com algumas informações preliminares, e é possível ter mais detalhes clicando no botão Ver detalhes. Este botão redireciona o usuário para a tela de detalhes de ocorrência, apresentada na Figura 20, onde mais informações são mostradas como o comentário do moderador, por exemplo. Esta informação é útil para os casos em que a ocorrência é recusada na moderação, pois o usuário poderá ter acesso ao motivo da recusa.

O campo Verificada será preenchido quando o usuário anexar um boletim de ocorrência válido em sua ocorrência e ela for aprovada pela moderação. Este documento comprova a veracidade da situação e diferencia a ocorrência de outras que não possuam o BO, esta diferenciação é apresentada na tela principal, no mapa de ocorrências. Sobre os ícones das ocorrências verificadas consta um *check* verde.

Outra opção disponível na tela de detalhes de ocorrência é o botão Desativar, em vermelho com um xis branco, no canto superior direito. Ao clicar neste botão, o autor da ocorrência poderá desativá-la a qualquer momento, caso julgue necessário.

Figura 20 - Tela de detalhes de ocorrência

i Esta ocorrência está em análise por um moderador, caso aprovada, será exibida no mapa de crimes

Dados de ocorrência ✕

Título
Ocorrência em análise

Descrição
Uma ocorrência em análise show de bola, tudo de bom.

Tipo	Status	Boletim de ocorrência
Furto	Em análise	
Criada por	Moderador	Data da ocorrência
alucard	admin	29/mai/2021
Criada em	Atualizada em	Verificada
28/mai/2021 21:47:14	28/jun/2021 21:47:14	

Fonte: Próprio autor.

Uma questão pensada durante o desenvolvimento do trabalho foi não expor os dados do usuário que criou a ocorrência para o público em geral devido a serem dados sensíveis e que poderiam comprometê-lo. Desta maneira, apenas os moderadores e administradores terão acesso a dados pessoais de usuários.

Ainda na tela de ocorrências, consta o botão Criar Ocorrência, que redireciona o usuário ao formulário de criar ocorrência com todos os campos necessários para a criação. Se algum campo for preenchido incorretamente e o usuário tentar criar o cadastro mesmo assim, um erro será exibido e ele poderá corrigir os dados incorretos. O campo de Boletim de Ocorrência permite o envio de um arquivo .PDF, JPEG ou PNG de no máximo 5MB

de tamanho, porém este campo não é obrigatório. Apesar de ser possível a criação de ocorrências sem boletim de ocorrência, as ocorrências que possuem este documento são consideradas como mais confiáveis dentro da aplicação.

Após a correta criação da ocorrência, ela é direcionada ao moderador ou administrador do sistema para que passe por uma avaliação de seu conteúdo. Este fluxo foi desenvolvido para evitar que existam dados inverídicos que possam gerar confusão entre os usuários do CrimeSpot. Uma ocorrência sem BO poderá ser inativada pela moderação caso contenha algum texto impróprio, informação inválida ou algum dado pessoal nos campos de texto que possa expor o criador da ocorrência.

Na Figura 21, a tela de criação de ocorrência pode ser verificada.

Figura 21 - Formulário de criação de ocorrência

Após submetida, a ocorrência será analisada pela moderação, caso aprovada, será exibida no mapa de crimes

Dados de ocorrência

Título*

Descrição*

Data da ocorrência* Tipo*

Boletim de ocorrência

Dados de endereço

CEP* País*

Estado* Cidade*

Rua* Número

Complemento

Cancelar Criar

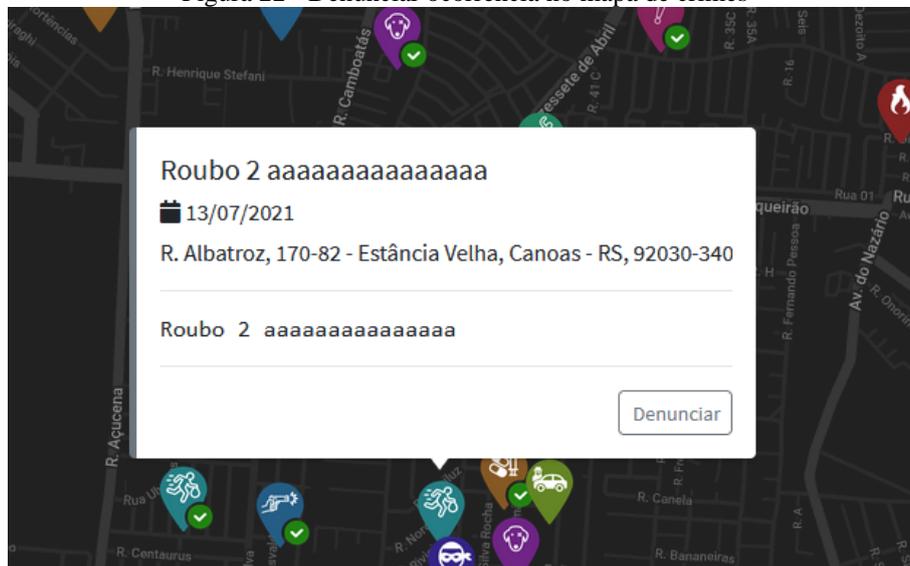
Fonte: Próprio autor.

Outra maneira de acessar o formulário de criação de ocorrência é clicando na opção Criar ocorrência no menu superior. Assim como no cadastro de usuário, na criação de ocorrência o endereço precisa ser georreferenciado, para que seja possível localizá-lo no mapa de ocorrências.

Caso um usuário visualize ocorrências e perceba algum dado falso ou conteúdo impróprio, será possível realizar a denúncia desta ocorrência ao clicar sobre ela no mapa e então clicar no botão denunciar, dentro da modal de informações da denúncia.

A ocorrência denunciada ainda será apresentada no mapa de crimes e outros usuários poderão denunciá-la também. Quando ela for denunciada uma primeira vez, será apresentada ao usuário moderador, apenas para visualização, e para o usuário administrador definir se será retirada do ar ou se a denúncia será rejeitada. Na Figura 22 é possível observar o botão de denunciar ocorrência.

Figura 22 - Denunciar ocorrência no mapa de crimes



Fonte: Próprio autor.

Quando o usuário não estiver logado e clicar em Denunciar, ele será direcionado para realizar login no sistema, caso já esteja autenticado, a modal de inserção do motivo da denúncia será apresentada. A modal está sendo mostrada na Figura 23.

Figura 23 - Modal de denúncia de ocorrência

Criar denúncia de ocorrência ✕

Motivo da denúncia

* O motivo da denúncia precisa conter entre 15 e 1000 caracteres

Cancelar Confirmar

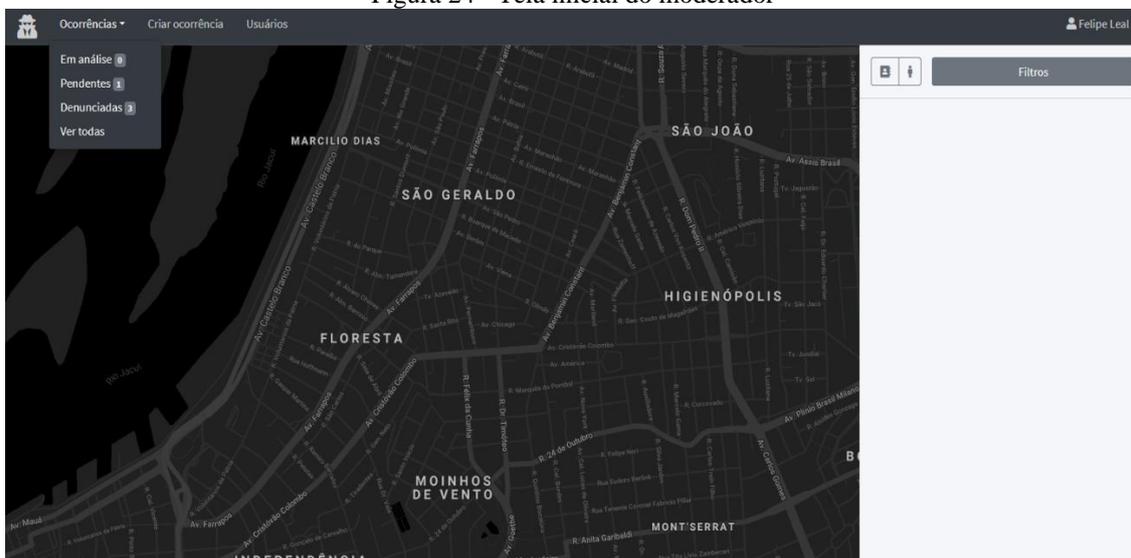
Fonte: Próprio autor.

Conforme dito anteriormente, todas as ocorrências criadas são enviadas para a moderação ou administração analisarem seu conteúdo. Na Figura 24 é possível ver a tela inicial que um usuário moderador visualiza ao realizar login no sistema.

No menu superior algumas opções são apresentadas como Ocorrências, onde ele poderá ver as ocorrências de todos os usuários do sistema, bem como realizar buscas com filtros. Também consta a opção de Usuários, que redireciona a uma lista com todos os usuários cadastrados no sistema (ativos e inativos). Os dados dos usuários são apresentados na lista e consta a opção de filtros onde é possível realizar uma busca textual, por perfil usuário autenticado, moderador ou administrador e também por status ativo ou inativo.

Ainda no menu superior do moderador consta a opção de Ocorrências onde, ao clicar, são apresentados menus com ocorrências em análise, que é uma lista de ocorrências que estão sendo verificadas pelo moderador, e o menu de ocorrências pendentes, uma lista com as ocorrências que ainda precisam ser analisadas para publicação na aplicação ou rejeição e deleção. Ainda consta o menu de ocorrências denunciadas, onde o moderador poderá visualizar as ocorrências que foram denunciadas por usuários autenticados e ainda não foram verificadas pelo administrador.

Figura 24 - Tela inicial do moderador



Fonte: Próprio autor.

O usuário moderador possui acesso a lista de todos os usuários do sistema e pode inativar um cadastro de usuário caso julgue necessário. Ao lado de cada item da lista de usuário é exibido um ícone de cadeado vermelho por onde é possível desativar um usuário. Esta funcionalidade permite que a moderação impeça um usuário de realizar ações dentro do sistema como criação de ocorrências inválidas ou denúncias indevidas. Na lista também é possível reativar um usuário que foi desativado anteriormente. A tela de listagem de usuários é apresentada na Figura 25.

Figura 25 - Lista de usuários

Busca por texto	Tipo de perfil	Status											
<input type="text" value="Pesquisar"/>	<input type="text" value="Todos perfis"/>	<input type="text" value="Todos status"/>	<input type="button" value="Pesquisar"/>										
<table border="1"> <tr> <td>accelerator</td> <td></td> </tr> <tr> <td>Nome: Accelerator</td> <td>Email: accelerator@gmail.com</td> </tr> <tr> <td>Data de Nascimento: 26/jun/1998</td> <td>CPF: 48615711038</td> </tr> <tr> <td>Raio de busca: 3.0KM</td> <td>Perfil: Usuário Autenticado</td> </tr> <tr> <td>Criado em: 22/jun/2021 22:41:32</td> <td>Atualizado em: 22/jun/2021 22:41:32</td> </tr> </table>				accelerator		Nome: Accelerator	Email: accelerator@gmail.com	Data de Nascimento: 26/jun/1998	CPF: 48615711038	Raio de busca: 3.0KM	Perfil: Usuário Autenticado	Criado em: 22/jun/2021 22:41:32	Atualizado em: 22/jun/2021 22:41:32
accelerator													
Nome: Accelerator	Email: accelerator@gmail.com												
Data de Nascimento: 26/jun/1998	CPF: 48615711038												
Raio de busca: 3.0KM	Perfil: Usuário Autenticado												
Criado em: 22/jun/2021 22:41:32	Atualizado em: 22/jun/2021 22:41:32												
<table border="1"> <tr> <td>alucard</td> <td></td> </tr> <tr> <td>Nome: Nosferatu Alucard</td> <td>Email: nosferatu.alucard@gmail.com</td> </tr> <tr> <td>Data de Nascimento: 26/jun/1998</td> <td>CPF: 15861939020</td> </tr> <tr> <td>Raio de busca: 3.0KM</td> <td>Perfil: Usuário Autenticado</td> </tr> <tr> <td>Criado em: 22/jun/2021 22:41:32</td> <td>Atualizado em: 22/jun/2021 22:41:32</td> </tr> </table>				alucard		Nome: Nosferatu Alucard	Email: nosferatu.alucard@gmail.com	Data de Nascimento: 26/jun/1998	CPF: 15861939020	Raio de busca: 3.0KM	Perfil: Usuário Autenticado	Criado em: 22/jun/2021 22:41:32	Atualizado em: 22/jun/2021 22:41:32
alucard													
Nome: Nosferatu Alucard	Email: nosferatu.alucard@gmail.com												
Data de Nascimento: 26/jun/1998	CPF: 15861939020												
Raio de busca: 3.0KM	Perfil: Usuário Autenticado												
Criado em: 22/jun/2021 22:41:32	Atualizado em: 22/jun/2021 22:41:32												
<table border="1"> <tr> <td>carina.emerim</td> <td></td> </tr> <tr> <td>Nome: Carina Emerim</td> <td>Email: carina.emerim@gmail.com</td> </tr> <tr> <td>Data de Nascimento: 26/jun/1998</td> <td>CPF: 40564269026</td> </tr> <tr> <td>Raio de busca: 3.0KM</td> <td>Perfil: Usuário Autenticado</td> </tr> <tr> <td>Criado em: 22/jun/2021 22:41:32</td> <td>Atualizado em: 22/jun/2021 22:41:32</td> </tr> </table>				carina.emerim		Nome: Carina Emerim	Email: carina.emerim@gmail.com	Data de Nascimento: 26/jun/1998	CPF: 40564269026	Raio de busca: 3.0KM	Perfil: Usuário Autenticado	Criado em: 22/jun/2021 22:41:32	Atualizado em: 22/jun/2021 22:41:32
carina.emerim													
Nome: Carina Emerim	Email: carina.emerim@gmail.com												
Data de Nascimento: 26/jun/1998	CPF: 40564269026												
Raio de busca: 3.0KM	Perfil: Usuário Autenticado												
Criado em: 22/jun/2021 22:41:32	Atualizado em: 22/jun/2021 22:41:32												

Fonte: Próprio autor.

No lado direito do menu superior constam duas opções para visualizar a lista de ocorrências criadas por usuários e que ainda não foram publicadas. Na opção Ocorrências pendentes, estão todas as ocorrências que foram criadas, mas ainda não estão sob a análise de um moderador ou administrador, portando nesta lista estão as ocorrências com status “Aguardando análise”.

A Figura 26 apresenta a tela de listagem de ocorrências pendentes.

Figura 26 - Lista de ocorrências pendentes

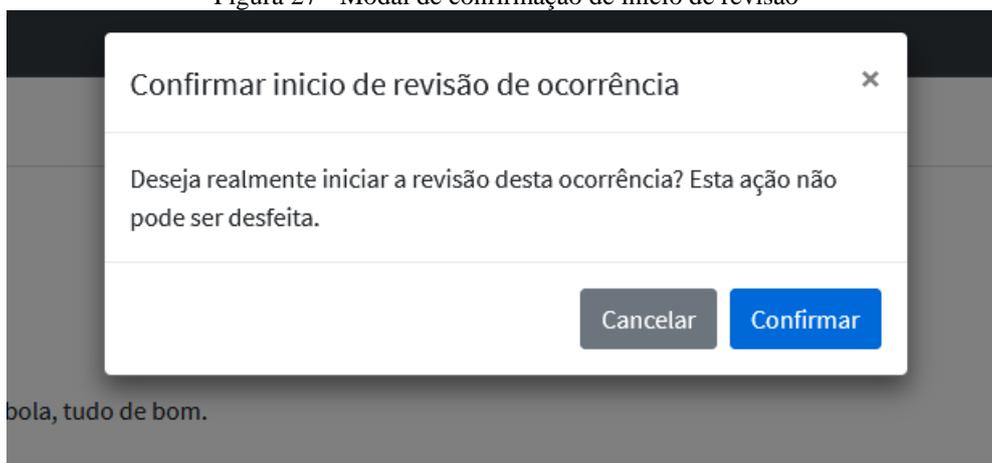
Busca por texto	Status da ocorrência	Ordenar por																			
<input type="text" value="Pesquisar"/>	<input type="text" value="Aguardando análise"/>	<input type="text" value="Mais recentes"/>																			
<table border="1"> <tr> <td>Tipos de ocorrência</td> <td><input type="checkbox"/> Apenas relacionadas a mim</td> <td><input type="button" value="Pesquisar"/></td> </tr> <tr> <td><input type="text" value="Todos os tipos"/></td> <td></td> <td></td> </tr> </table>				Tipos de ocorrência	<input type="checkbox"/> Apenas relacionadas a mim	<input type="button" value="Pesquisar"/>	<input type="text" value="Todos os tipos"/>														
Tipos de ocorrência	<input type="checkbox"/> Apenas relacionadas a mim	<input type="button" value="Pesquisar"/>																			
<input type="text" value="Todos os tipos"/>																					
<table border="1"> <tr> <td colspan="2">Ocorrência legal</td> <td></td> </tr> <tr> <td colspan="3">Uma ocorrência divertida show de bola, tudo de bom.</td> </tr> <tr> <td>Criada por: admin</td> <td>Data da ocorrência: 22/jun/2021</td> <td></td> </tr> <tr> <td>Cidade: Porto Alegre</td> <td>Rua: Rua Almirante Barroso</td> <td></td> </tr> <tr> <td>Tipo: Roubo</td> <td>Status: Aguardando análise</td> <td></td> </tr> <tr> <td>Criada em: 22/jun/2021 22:41:32</td> <td>Atualizada em: 22/jun/2021 22:41:32</td> <td><input type="button" value="Ver detalhes"/></td> </tr> </table>				Ocorrência legal			Uma ocorrência divertida show de bola, tudo de bom.			Criada por: admin	Data da ocorrência: 22/jun/2021		Cidade: Porto Alegre	Rua: Rua Almirante Barroso		Tipo: Roubo	Status: Aguardando análise		Criada em: 22/jun/2021 22:41:32	Atualizada em: 22/jun/2021 22:41:32	<input type="button" value="Ver detalhes"/>
Ocorrência legal																					
Uma ocorrência divertida show de bola, tudo de bom.																					
Criada por: admin	Data da ocorrência: 22/jun/2021																				
Cidade: Porto Alegre	Rua: Rua Almirante Barroso																				
Tipo: Roubo	Status: Aguardando análise																				
Criada em: 22/jun/2021 22:41:32	Atualizada em: 22/jun/2021 22:41:32	<input type="button" value="Ver detalhes"/>																			
<p>« 1 »</p>																					

Fonte: Próprio autor.

Para que um moderador ou administrador possa iniciar a análise de uma ocorrência é necessário acessar os detalhes dela e clicar na opção “Iniciar Revisão”, assim ela irá alterar o status para Em Análise e evitará que seja analisada simultaneamente por duas pessoas ou que a ocorrência sofra alterações enquanto a análise ocorre. Assim que o moderador ou administrador clica em Confirmar na modal de confirmação de início de revisão de ocorrência, seu status é alterado e ela não será mais exibida na listagem de ocorrências pendentes.

A Figura 27 mostra a modal apresentada ao moderador ou administrador quando eles estão no processo de pegar uma ocorrência para analisarem.

Figura 27 - Modal de confirmação de início de revisão



Fonte: Próprio autor.

Ao confirmar a modal da Figura 25, a ocorrência passará a ser apresentada na lista de Ocorrências em Análise e permanecerá nesse status até que a moderação ou administração finalize a verificação.

Na Figura 28 é apresentada a tela de lista de ocorrências em análise.

Figura 28 - Lista de ocorrências em análise

Busca por texto

Status da ocorrência

Ordenar por

Tipos de ocorrência Apenas relacionadas a mim

Ocorrência legal
Uma ocorrência divertida show de bola, tudo de bom.

Criada por: admin	Data da ocorrência: 22/jun/2021
Cidade: Porto Alegre	Rua: Rua Almirante Barroso
Tipo: Roubo	Status: Em análise
Criada em: 22/jun/2021 22:41:32	Atualizada em: 26/jun/2021 01:37:04
Moderador: moderator	

Fonte: Próprio autor.

Na lista de ocorrências em análise, quando o moderador ou administrador clica em Continuar revisão, todos os dados da ocorrência são apresentados para que possa ser realizada a verificação destes dados. Caso exista boletim de ocorrência anexado, é possível baixá-lo.

Na Figura 29 é mostrado campo de comentários do moderador que é apresentado juntamente com os detalhes da ocorrência. Neste campo o moderador pode inserir os comentários pertinentes e definir pela recusa ou aprovação da ocorrência. O autor da ocorrência terá acesso a este comentário na tela de listagem das suas ocorrências, independentemente do status da ocorrência.

Figura 29 - Parte de comentários do moderador

Moderação

Comentários do moderador

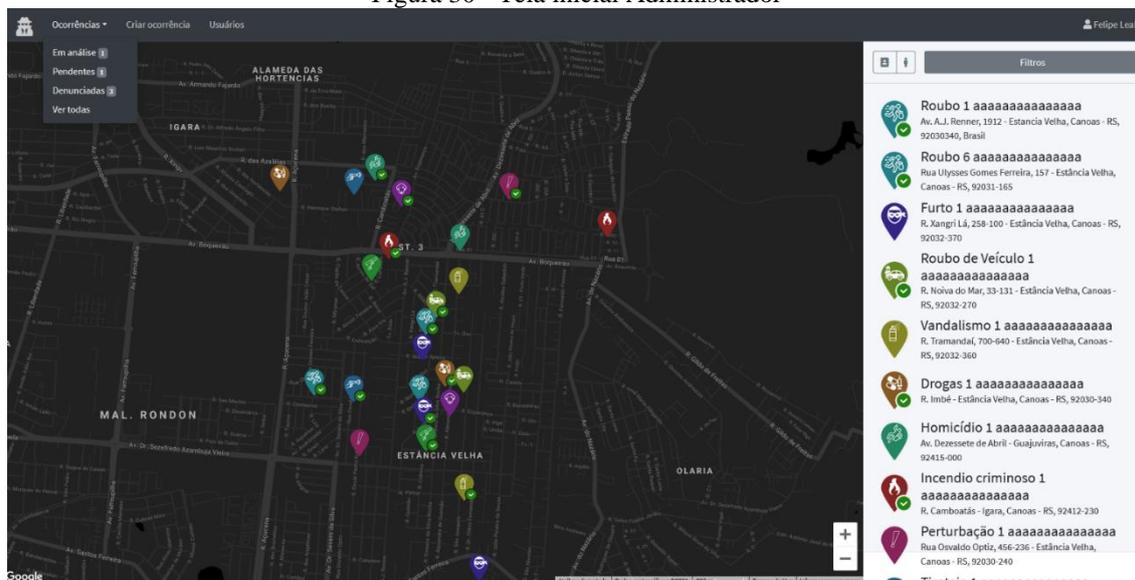
Fonte: Próprio autor.

Quando a ocorrência é aprovada, ela será apresentada no mapa de ocorrências e todos os usuários que acessarem a aplicação terão acesso.

A tela inicial apresentada quando um usuário administrador se autentica no sistema é semelhante a tela do moderador, porém este usuário possui privilégios a mais. O Administrador pode remover qualquer ocorrência do sistema, mesmo aquelas que já foram analisadas e aprovadas pela moderação. Ele também possui todas as permissões de moderador, e pode pegar ocorrências da lista de ocorrências pendentes para moderar.

A Figura 30 mostra a tela inicial apresentada ao administrador quando ele realiza login na aplicação.

Figura 30 - Tela inicial Administrador



Fonte: Próprio autor.

Além dos privilégios já citados, o administrador também pode tornar outros usuários moderadores ou retirar a permissão de um moderador e transformá-lo em um usuário autenticado. Outra ação exclusiva do administrador é a análise de denúncias de ocorrências. As ocorrências denunciadas podem ser acessadas diretamente no menu Ocorrências > Denunciadas, onde será apresentada a lista de ocorrências com filtros aplicados que facilitam a busca. Esta lista está sendo apresentada na Figura 31.

Figura 31 - Lista de ocorrências denunciadas

Busca por texto Status da ocorrência Ordenar por

Pesquisar Aprovada Mais denunciadas

Tipos de ocorrência

Todos os tipos Apenas relacionadas a mim Pesquisar

Roubo de Veículo 1 aaaaaaaaaaaaaaaaa
Roubo de Veículo 1 aaaaaaaaaaaaaaaaa

Criada por: eraldo.lead	Data da ocorrência: 22/jul/2021
Cidade: Canoas	Rua: Rua Noiva do Mar
Tipo: Roubo de Veículo	Status: Aprovada
Criada em 21/jul/2021 22:59:00	Atualizada em 24/jul/2021 22:59:00
Denúncias: 3	Moderador: admin

[Ver detalhes](#)

Furto 1 aaaaaaaaaaaaaaaaa
Furto 1 aaaaaaaaaaaaaaaaa

Criada por: maria.amelia	Data da ocorrência: 23/jul/2021
Cidade: Canoas	Rua: Rua Xangri-lá
Tipo: Furto	Status: Aprovada
Criada em 22/jul/2021 22:59:00	Atualizada em 24/jul/2021 22:59:00
Denúncias: 2	Moderador: admin

[Ver detalhes](#)

Fonte: Próprio autor.

Ao clicar para ver os detalhes de uma ocorrência que possui denúncias, o administrador terá acesso a todas as denúncias realizadas para aquela ocorrência e poderá definir se esta ocorrência será removida do mapa ou se continuará sendo exibida. As informações de denúncia estão sendo apresentadas na Figura 32.

Figura 32 - Denúncias realizadas por usuários em uma ocorrência

Denúncias 3

[Aceitar denúncias](#) [Recusar denúncias](#)

Criada por: [eraldo.lead](#) Criada em: 24/jul/2021 22:59:00

Motivo da denúncia

Denúncia de ocorrência 6

Criada por: [maria.amelia](#) Criada em: 24/jul/2021 22:59:00

Motivo da denúncia

Denúncia de ocorrência 5

Criada por: [romero.britto](#) Criada em: 24/jul/2021 22:59:00

Motivo da denúncia

Denúncia de ocorrência 4

Fonte: Próprio autor.

Quando o administrador acessa a opção usuários no menu superior, a lista de usuários é apresentada a ele, porém diferentemente do moderador, um ícone adicional de “aperto de mãos” está disponível para que o administrador possa gerenciar as permissões dos usuários. Na Figura 33 é apresentado o ícone de adicionar ou retirar permissão de moderador de um usuário.

Figura 33 - Lista de usuários apresentada ao Administrador

luna.lovegood	<p>Nome: Luna Lovegood Data de Nascimento: 26/jun/1998 Raio de busca: 3.0KM Criado em: 22/jun/2021 22:41:32</p>	<p>Email: luna.lovegood@gmail.com CPF: 14456052015 Perfil: Usuário Autenticado Atualizado em: 22/jun/2021 22:41:32</p>	 
maria.amelia	<p>Nome: Maria Amélia Data de Nascimento: 26/jun/1998 Raio de busca: 3.0KM Criado em: 22/jun/2021 22:41:32</p>	<p>Email: maria.amelia@gmail.com CPF: 83040770004 Perfil: Usuário Autenticado Atualizado em: 22/jun/2021 22:41:32</p>	 
melissa.leal	<p>Nome: Melissa Leal Data de Nascimento: 26/jun/1998 Raio de busca: 3.0KM Criado em: 22/jun/2021 22:41:32</p>	<p>Email: melissa.leal@gmail.com CPF: 08462633001 Perfil: Usuário Autenticado Atualizado em: 22/jun/2021 22:41:32</p>	 
moderator	<p>Nome: Felipe Leal</p>	<p>Email: moderator@gmail.com</p>	 

Fonte: Próprio autor.

Outra diferença entre o usuário moderador e o administrador é que o administrador pode remover qualquer ocorrência do sistema, independentemente de seu status. Ao acessar os detalhes de uma ocorrência, é apresentado um botão vermelho com um xis branco no centro, onde ao clicar, será possível remover a ocorrência. A Figura 34 apresenta a tela de detalhes de ocorrência através da visão de um usuário administrador.

Figura 34 - Detalhes de ocorrência apresentado ao Administrador

Dados de ocorrência		
Título		
Ocorrência legal		
Descrição		
Uma ocorrência divertida show de bola, tudo de bom.		
Tipo	Status	Boletim de ocorrência
Roubo	Em análise	
Criada por	Moderador	Data da ocorrência
admin	moderator	22/jun/2021
Criada em	Atualizada em	Verificada
22/jun/2021 22:41:32	26/jun/2021 01:37:04	
Comentários do moderador		

Fonte: Próprio autor.

5.3 TESTES DE ACEITAÇÃO

Para testar a usabilidade do sistema e estabelecer a confiança em sua integridade, verificando se está apto para utilização, foi realizado o teste de aceitação. O teste do CrimeSpot consistiu em divulgar o sistema para um grupo de usuários juntamente a um questionário de satisfação. Este grupo de usuários foi formado por profissionais de tecnologia da informação e pessoas não relacionadas a esta área. Do grupo selecionado houve 13 pessoas que utilizaram o sistema e preencheram o formulário.

Uma sequência de passos foi pensada com intuito de entender se tarefas básicas de usuário e funcionalidades estavam funcionando como esperado e se os usuários conseguiriam finalizá-los sem erros. Todas essas informações foram enviadas para os usuários de teste por *e-mail*.

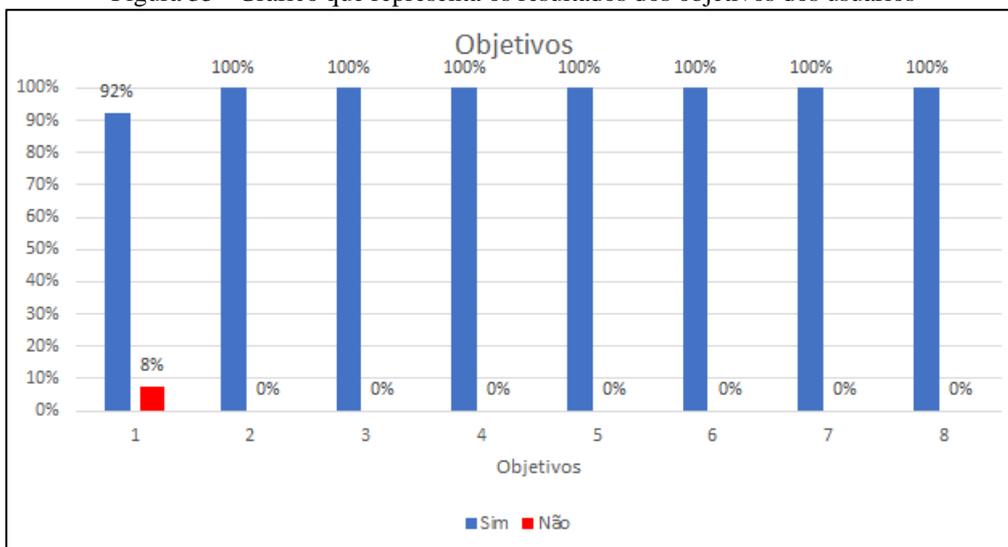
Nestes passos o usuário deveria criar um cadastro, criar uma ocorrência, editar seu cadastro, editar uma ocorrência criada por ele e visualizar ocorrências disponibilizadas no mapa de ocorrências com e sem filtros. Também foram adicionados campos abertos no questionário para reporte de erros e/ou sugestões. O questionário completo, bem como as instruções estão no Apêndice B.

A Figura 35 representa o gráfico com as respostas obtidas referentes aos objetivos dos passos requisitados aos usuários, sendo a coluna esquerda para opção “sim” e a coluna a esquerda para a opção “não”. As seguintes perguntas foram realizadas:

- **Número 1:** Você conseguiu visualizar os detalhes de uma ocorrência no mapa de crimes?
- **Número 2:** Você conseguiu realizar uma busca com filtros no mapa de crimes?
- **Número 3:** Você conseguiu criar seu cadastro no sistema?

- **Número 4:** Você conseguiu criar uma ocorrência?
- **Número 5:** Você conseguiu editar seu cadastro?
- **Número 6:** Você conseguiu visualizar a lista de ocorrências que você criou?
- **Número 7:** Você conseguiu visualizar os detalhes de uma ocorrência que você criou?
- **Número 8:** Você conseguiu editar uma ocorrência que você criou?

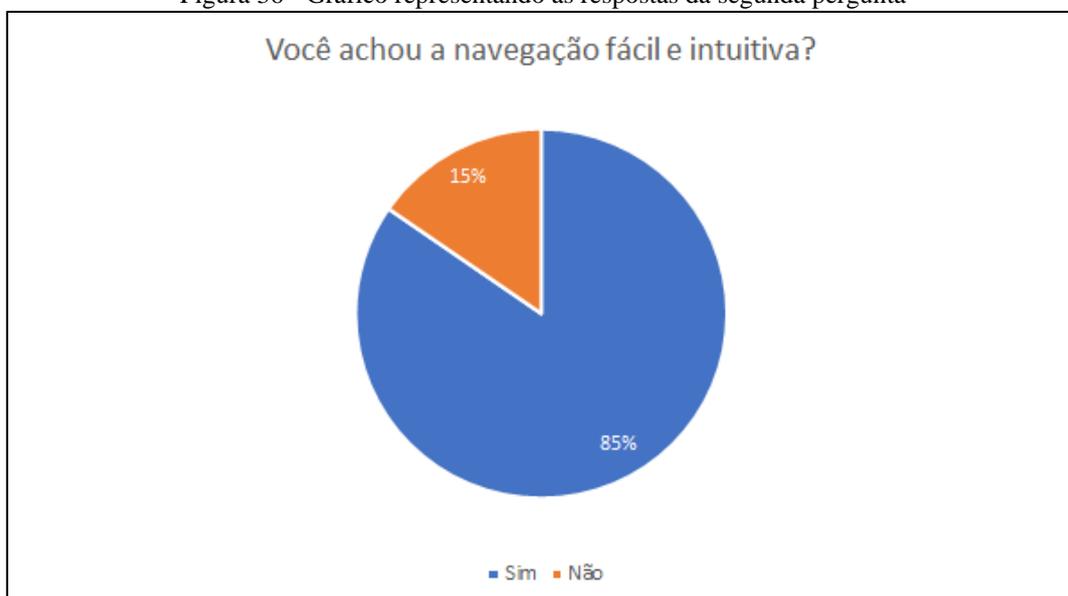
Figura 35 - Gráfico que representa os resultados dos objetivos dos usuários



Fonte: Próprio autor.

A Figura 36 mostra o gráfico que representa as respostas da segunda pergunta “Você achou a navegação fácil e intuitiva?”. O resultado foi que 15% dos usuários informaram “não” e 85% informaram que “sim”.

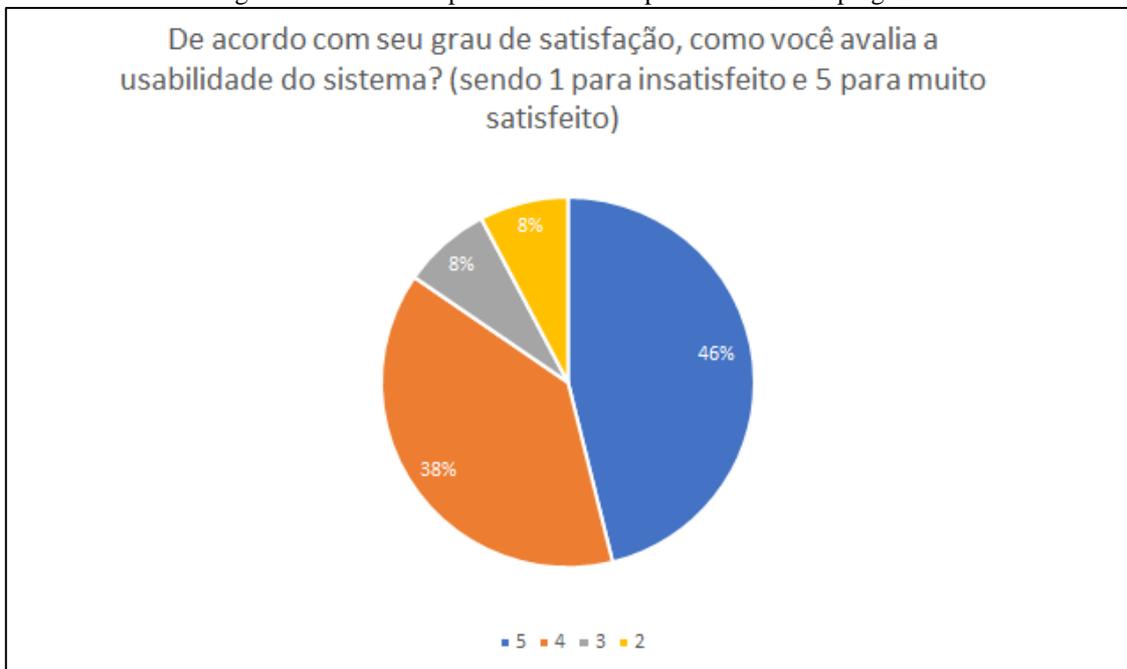
Figura 36 - Gráfico representando as respostas da segunda pergunta



Fonte: Próprio autor.

A Figura 37 mostra o gráfico que representa as respostas da terceira pergunta “De acordo com seu grau de satisfação, como você avalia a usabilidade do sistema? (Sendo 1 para insatisfeito e 5 para muito satisfeito)”. Houve 8% dos usuários que selecionaram 2, 8% selecionaram 3, 38% escolheram 4 e 46% escolheram 5.

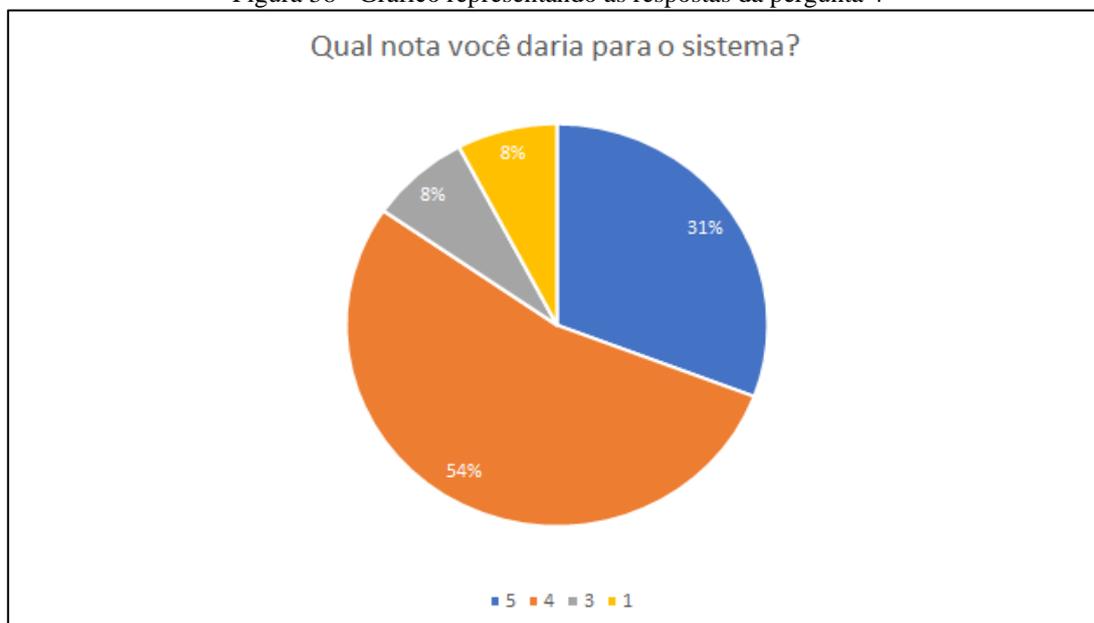
Figura 37 - Gráfico representando as respostas da terceira pergunta



Fonte: Próprio autor.

Na Figura 38 é possível observar o gráfico que representa as respostas da quarta pergunta “Qual nota você daria para o sistema?”. O resultado foi que 8% dos usuários selecionaram 1, 8% escolheram 3, 54% escolheram 4 e 31% informaram 5.

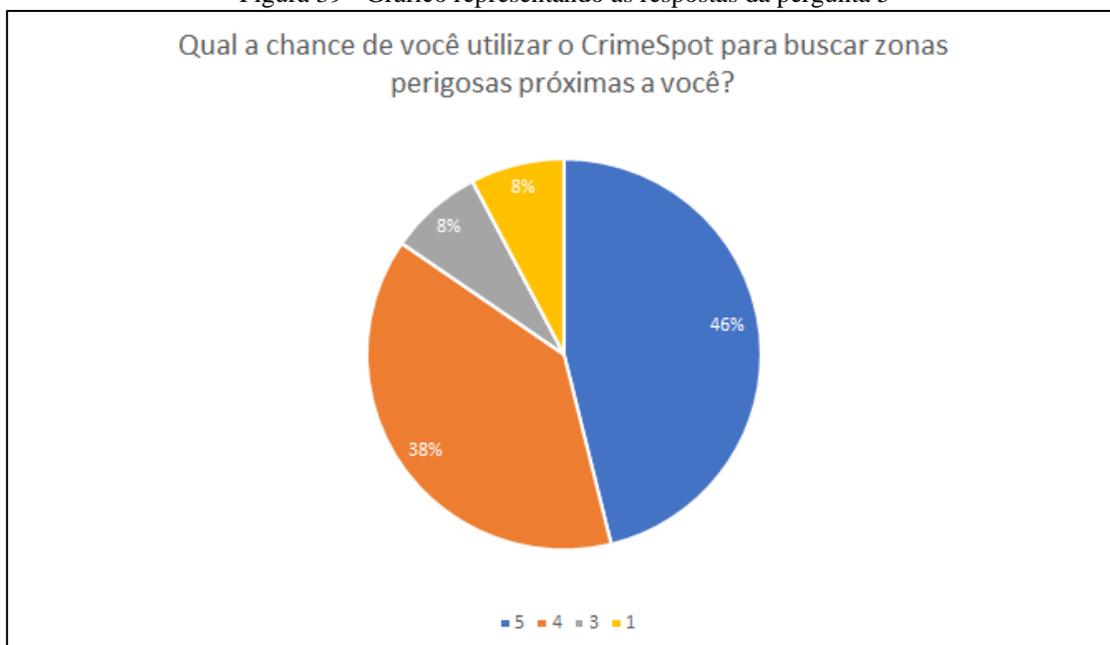
Figura 38 - Gráfico representando as respostas da pergunta 4



Fonte: Próprio autor.

A figura 39 apresenta o gráfico de resultados da pergunta 5 “Qual a chance de você utilizar o CrimeSpot para buscar zonas perigosas próximas a você?”. Para esta pergunta os seguintes resultados foram identificados: 8% dos usuários escolheram 1, 8% escolheram 3, 38% selecionaram 4 e 46% selecionaram 5.

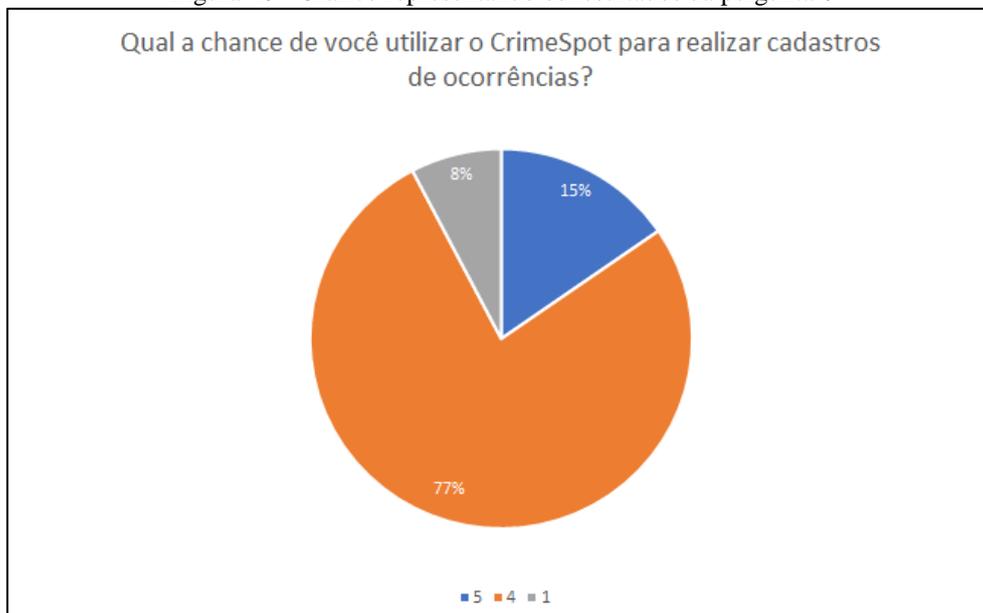
Figura 39 - Gráfico representando as respostas da pergunta 5



Fonte: Próprio autor.

Na Figura 40 é apresentado o gráfico representando as respostas da sexta pergunta “Qual a chance de você utilizar o CrimeSpot para realizar cadastros de ocorrências?”. O resultado foi que 8% dos usuários selecionaram 1, 77% selecionaram 4 e 15% escolheram 5.

Figura 40 - Gráfico representando os resultados da pergunta 6



Fonte: Próprio autor.

A Figura 41 mostra o gráfico representando as respostas da sétima pergunta: “Você recomendaria o CrimeSpot a um amigo?”. Os resultados foram que 8% dos usuários escolheram “não” e 92% escolheram “sim”.

Figura 41 - Gráfico representando as respostas da pergunta 7



Fonte: Próprio autor.

Nos testes de aceitação foi possível verificar que o sistema teve uma boa avaliação, em sua maioria com notas altas. Dois campos abertos foram inseridos no questionário, uma delas perguntando se houve erros ou algo inesperado ocorreu durante a navegação. Um problema relatado por um dos usuários foi que ao digitar o CEP, a ferramenta não preencheu o campo “rua” corretamente. Este problema foi mitigado ao implementar a possibilidade de os usuários realizarem *input* manual dos dados de endereço quando ocorrem erros no lado da API da Google.

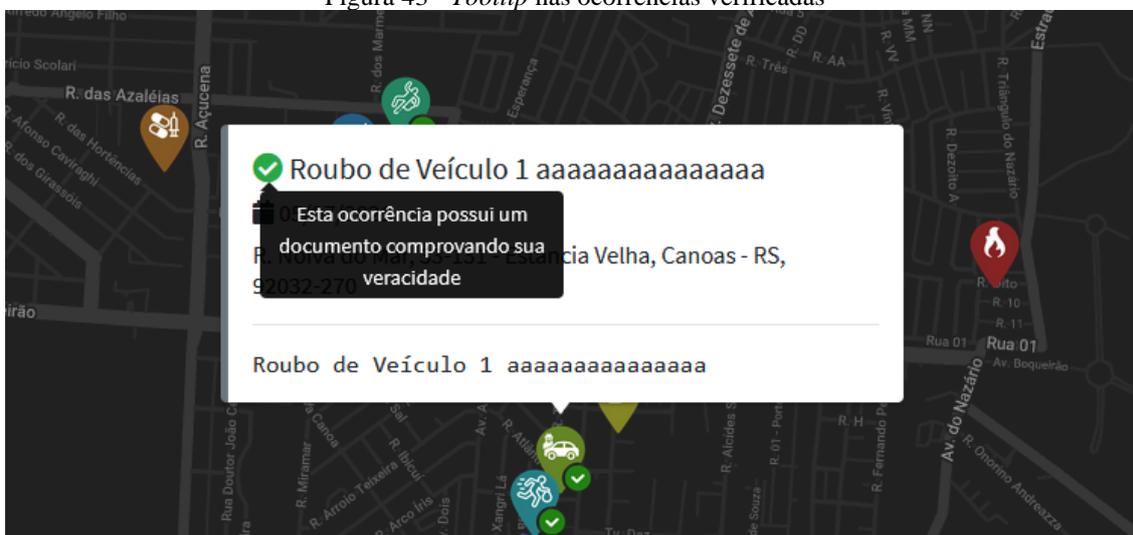
Houve retornos positivos nesta pergunta, como um dos usuários acrescentando que “Isso poderia se tornar um bom produto para análise de perigo e segurança, tendo o cidadão como principal fiscalizador”, outro usuário informou que achou uma ideia “muito boa” e outros usuários informaram que não houve erros nos passos realizados.

No campo aberto para sugestões, foram recebidas algumas ideias bem interessantes. Algumas foram aplicadas como melhorias no sistema e outras podem ser aplicadas em trabalhos futuros. Uma das melhorias sugeridas e que foi implementada foi a criação de um botão para limpeza dos filtros, que está apresentado na Figura 42.

Figura 42 - Botão Limpar Filtros

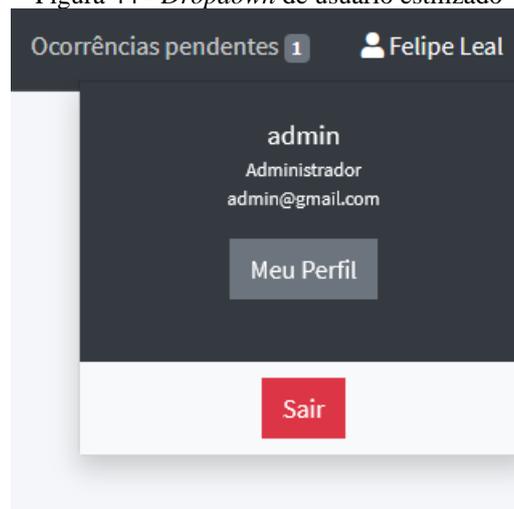
Fonte: Próprio autor.

Um dos usuários de teste informou que não compreendeu a questão de ocorrência verificada e esperava que as ocorrências verificadas fossem aquelas que passaram pela moderação. Para minimizar essa possível confusão que outros usuários possam ter foi implementada uma *tooltip* nas ocorrências verificadas indicando que elas possuem um documento comprovando sua autenticidade e, por isso, são verificadas. Essa *tooltip* pode ser visualizada na Figura 43.

Figura 43 - *Tooltip* nas ocorrências verificadas

Fonte: Próprio autor.

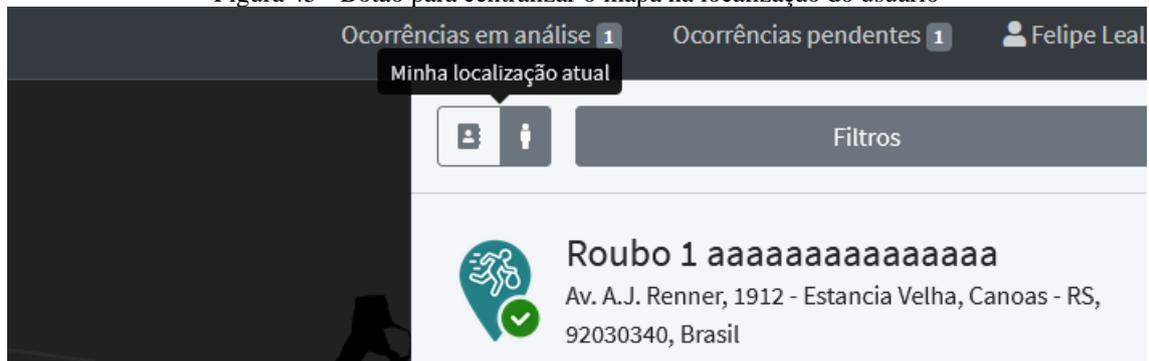
Através das sugestões dos usuários, o *dropdown* de usuário foi alterado para combinar com o restante do tema do sistema. Isso está sendo mostrado na Figura 44.

Figura 44 - *Dropdown* de usuário estilizado

Fonte: Próprio autor.

A Figura 45 mostra outra melhoria realizada no sistema: um botão que centraliza o mapa de crimes de acordo com a localização do usuário. Isso é muito interessante para usuários *mobile* que podem ter acesso rápido ao mapa de crimes da região onde estão.

Figura 45 - Botão para centralizar o mapa na localização do usuário



Fonte: Próprio autor.

Outras melhorias que foram realizadas com os feedbacks dos usuários do teste de aceitação:

- Correção em espaçamentos incorretos no *layout* do sistema;
- Adicionado campo bairro ao cadastro de endereço;
- Adicionada máscara no campo CEP;
- Adicionado botão para centralizar o mapa no endereço cadastrado pelo usuário;
- Erro 403 quando um usuário autenticado tenta acessar *login* ou criação de cadastro tratado.

Algumas sugestões podem ser estudadas para implementação em trabalhos futuros como: adicionar validações de campo do lado do cliente com Javascript, adicionar página com instruções e explicações sobre fluxos do sistema, adicionar geolocalização interativa de endereços, possibilitando que o usuário informe no mapa a localização de seu endereço

ao realizar um cadastro, adicionar a localização da ocorrência no mapa ao fluxo de revisão para o moderador poder verificar a consistência do endereço informado *versus* a localização informada.

Os resultados do teste de aceitação mostraram que o sistema está apto para utilização, com fluxos consistentes e relevantes. Os usuários de teste em sua grande maioria aprovaram o CrimeSpot e contribuíram com melhorias significantes. No próximo capítulo será apresentada a conclusão deste trabalho.

6 CONCLUSÃO

O crescente índice de criminalidade observado diariamente motiva as pessoas a procurarem maneiras de defesa para não se tornarem alvos dos criminosos. E foi isso que motivou o desenvolvimento deste trabalho.

Ao longo do processo de pesquisa em sistemas semelhantes e amadurecimento da ideia, foi possível entender melhor as consequências da violência para a população, o medo que se instala na mente dos cidadãos quando necessitam passar em algum local desconhecido e até mesmo pessoas que evitam sair de casa em determinados horários por medo de serem vítimas de alguma situação.

A partir da criação dos diagramas de casos de uso e ER, as funcionalidades e atores do sistema ficaram mais claros e assim foi possível iniciar o desenvolvimento deste trabalho.

O sistema que foi desenvolvido visa auxiliar os seus usuários apresentando dados sobre a criminalidade de uma certa região, com objetivo de que se evite acessar estes locais caso seja possível.

Para assegurar que os principais fluxos e funcionalidades da aplicação estão funcionando corretamente foram realizados testes funcionais e, posteriormente, estes testes foram automatizados utilizando Selenium. Outro objetivo importante para conclusão deste trabalho foram os testes de aceitação realizados com um grupo de usuários com objetivo de garantir que o sistema está apto para utilização.

Para que os testes de aceitação pudessem ser realizados e para que os usuários pudessem acessar o sistema, ele foi publicado na internet e está disponível para o público em geral. Os resultados dos testes mostraram que a grande maioria dos usuários aprovou o sistema e considera que ele pode auxiliar os usuários a evitarem áreas perigosas, que é o objetivo principal deste trabalho.

6.1 TRABALHOS FUTUROS

Uma das principais melhorias que podem ser realizadas em trabalhos futuros para o sistema CrimeSpot é a automatização da moderação de ocorrências, mesmo que parcialmente. Com esta melhoria o processo de disponibilização ou remoção das ocorrências seria otimizado. Também poderia ser criado um sistema de notificações por e-mail que envia um alerta ao usuário (caso ele assim configure) sempre que uma nova ocorrência for publicada em sua região, ou quando uma ocorrência sua for publicada pela moderação.

Outra melhoria que visa aumentar a segurança seria uma verificação por *e-mail* no momento do cadastro do usuário. Um *e-mail* de confirmação poderia ser enviado para o usuário, e ele confirmaria que o endereço de *e-mail* fornecido é real e pertence a ele. Isto reduziria o índice de perfis de usuário falsos.

A hora também pode ser adicionada nas ocorrências para que seja possível entender em qual momento do dia cada área é mais perigosa. Para facilitar a visualização das ocorrências no mapa, um pequeno desvio aleatório nas coordenadas pode ser aplicado em um trabalho futuro com objetivo de evitar que duas ocorrências ocupem a mesma localização deixando os marcadores sobrepostos.

REFERÊNCIAS

Amazon. Disponível em: <<https://aws.amazon.com/pt/what-is-aws/>>. Acesso em: 10 de Jun. 2021.

Britannica. Client-Server Architecture. Disponível em: <<https://www.britannica.com/technology/client-server-architecture>>. Acesso em: 14 de Jun. 2021.

CAMPOS, Carla Leila Oliveira. Crimes e punições: representações da criminalidade e do papel do estado em seu combate na mídia impressa brasileira. **Revista Línguas & Letras** – Unioeste – Vol. 16 – Nº 32 – 2015. Disponível em: <<http://e-revista.unioeste.br/index.php/linguaseletras/article/view/11701/8694>>. Acesso em: 10 de Jun. 2021.

Crimemapping. Disponível em: <<https://www.crimemapping.com>>. Acesso em: 27 de Ago. 2021.

Diário Gaúcho. Polícia Civil conclui apenas 17% das ocorrências de roubo registradas no RS. Disponível em: <<http://diariogaucha.clicrbs.com.br/rs/policia/noticia/2019/05/policia-civil-conclui- apenas-17-das-ocorrencias-de-roubo-registradas-no-rs-10941735.html>>. Acesso em: 22 de Ago. 2021.

Developers. Geocode API Get Started. Disponível em: <<https://developers.google.com/maps/documentation/geocoding/start>>. Acesso em: 14 de Jun. 2021.

Developers. Maps Javascript API Overview. Disponível em: <<https://developers.google.com/maps/documentation/javascript/tutorial>>. Acesso em: 14 de Jun. 2021.

Devmedia. Hibernate Search. Disponível em: <https://arquivo.devmedia.com.br/artigos/Higor_Medeiros/hibernate-search/image1.png>. Acesso em: 02 de Abr. 2021.

Docker. Containerized Applications. Disponível em: <https://www.docker.com/sites/default/files/d8/2018-11/docker-containerized-appliction-blue-border_2.png>. Acesso em: 03 de Abr. 2021.

Docker. How Docker Helps Development Teams. Disponível em: <<https://www.docker.com/use-cases>>. Acesso em: 03 de Abr. 2021.

Docker. What is a container? Disponível em: <<https://www.docker.com/resources/what-container>>. Acesso em: 03 de Abr. 2021.

Docs.Docker. Overview of Docker Compose. Disponível em: <<https://docs.docker.com/compose/>>. Acesso em: 03 de Abr. 2021.

Educative. What is server-side rendering? Disponível em: <<https://www.educative.io/edpresso/what-is-server-side-rendering>>. Acesso em: 02 de Abr. 2021.

Eu faço Programas. Diagramas UML – Resumo rápido. Disponível em: <<https://eufacoprogramas.com/diagramas-uml-resumo>>. Acesso em: 10 de Jun. de 2021.

Expressjs. Using template engines with Express. Disponível em: <<https://expressjs.com/en/guide/using-template-engines.html>>. Acesso em: 14 de Jun. 2021.

Fluentlenium. Disponível em: <<https://fluentlenium.com/>>. Acesso em: 10 de Jun. de 2021.

Gaucha ZH. Com alta de 58% na taxa de homicídios em 10 anos, RS piora duas posições em ranking de violência. Disponível em: <<https://gauchazh.clicrbs.com.br/seguranca/noticia/2018/06/com-alta-de-58-na-taxa-de-homicidios-em-10-anos-rs-piora-duas-posicoes-em-ranking-de-violencia-cji1n48tz0dqs01qo5lp1zbow.html>>. Acesso em: 10 de Jun. de 2021.

Gaucha ZH. Três exemplos do uso de Tecnologia para combater a criminalidade. Disponível em: <<https://gauchazh.clicrbs.com.br/seguranca/noticia/2016/09/tres-exemplos-do-uso-de-tecnologia-para-combater-a-criminalidade-7505304.html>>. Acesso em: 10 de Jun. de 2021.

HANSEN, Roseli Persson *et al.* Web Services: An architecture overview. Disponível em: <<http://projeto.unisinos.br/webcomposej/Artigos/webservices.pdf>>. Acesso em: 14 de Jun. 2021.

JASWAL, Shivam. What is a server: Definition and how does it actually work? Disponível em: <<https://www.thecoderpedia.com/blog/what-is-a-server/>>. Acesso em: 14 de Jun. 2021.

Lucene. Apache Lucene Core. Disponível em: <<https://lucene.apache.org/core/>>. Acesso em: 02 de Jun. 2021.

MÁXIMO, Alexandre Alves. A importância do mapeamento da criminalidade utilizando-se de tecnologia de sistema de informação geográfica para auxiliar a segurança pública no combate a violência. Disponível em: <<https://core.ac.uk/download/pdf/30367818.pdf>>. Acesso em: 10 de Jun. de 2021.

Medium. Docker X Docker Compose. Disponível em: <https://miro.medium.com/max/2568/1*LHq5mhynSjYBIhfgY3czkQ.png>. Acesso em: 03 de Abr. 2021.

Mozilla. O que é um servidor web? Disponível em: <https://developer.mozilla.org/pt-BR/docs/Learn/Common_questions/o_que_e_um_web_server>. Acesso em: 14 de Jun. 2021.

OLIVEIRA, Sara Mendonça Poubel de. Disseminação das Informações na era das *Fake News*. V **Encontro Regional dos Estudantes de Biblioteconomia, Documentação, Gestão e Ciência da Informação das Regiões Sudeste, Centro-Sul e Sul**. Minas Gerais, 2018. Disponível em: <<http://portaldeperiodicos.eci.ufmg.br/index.php/moci/article/viewFile/3771/2161>>. Acesso em: 10 de Jun. de 2021.

LucidChart. Disponível em: <<https://www.lucidchart.com/pages/pt>>. Acesso em: 23 de Jun. 2021.

Opensource. What is Docker? Disponível em: <<https://opensource.com/resources/what-docker>>. Acesso em: 03 de Abr. 2021.

OSTEZER; DRAKE, Mark. SQLite vs MySQL vs PostgreSQL: A comparison of relational database management systems. Disponível em: <<https://www.digitalocean.com/community/tutorials/sqlite-vs-mysql-vs-postgresql-a-comparison-of-relational-database-management-systems>>. Acesso em: 14 de Jun. 2021.

OTT – Onde Tem Tiroteio. Disponível em: <<https://www.ondetemtiroteio.com.br>>. Acesso em: 27 de Ago. 2021.

PAGOTTO, Tiago *et al.* Scrum Solo, Processo de Software para Desenvolvimento Individual. Disponível em: <<https://engenhariasoftware.files.wordpress.com/2016/04/scrum-solo.pdf>>. Acesso em: 10 de Jun. de 2021.

Rede Brasil Atual. Divulgação de notícias falsas nas redes sociais pode ter consequências graves. Disponível em: <<https://www.redebrasilatual.com.br/revistas/2017/04/divulgacao-de-noticias-falsas-nas-redes-sociais-pode-ter-consequencias-graves/>>. Acesso em: 20 de Ago. 2021.

ROUSE, Margareth. Embedded Tomcat. Disponível em: <<https://www.theserverside.com/definition/embedded-Tomcat>>. Acesso em: 14 de Jun. 2021.

SEIBEL, Tiago. O que são aplicações web? Disponível em: <<http://www.tiagoseibel.com.br/2008/02/o-que-so-aplicaes-web.html>>. Acesso em: 14 de Jun. 2021.

Spring. Web Applications. Disponível em: < <https://spring.io/web-applications> >. Acesso em: 14 de Jun. 2021.

The Club. UML Diagrama de casos de uso. Disponível em: <<http://www.theclub.com.br/restrito/revistas/201305/umld1305.aspx>>. Acesso em: 24 de Jun. 2021.

Thymeleaf. Disponível em: <<https://www.thymeleaf.org/index.html>>. Acesso em: 23 de Jun. 2021.

Thymeleaf. Thymeleaf VS JSP. Disponível em: <<https://www.thymeleaf.org/doc/articles/thvsjsp.html>>. Acesso em: 10 de Jun. 2021.

Ubuntu. Disponível em: <<https://ubuntu.com/server>>. Acesso em: 23 de Jun. 2021.

WALLEN, Jack. Ubuntu Server. A cheat sheet. Disponível em: <<https://www.techrepublic.com/article/ubuntu-server-the-smart-persons-guide/>>. Acesso em: 23 de Jun. 2021.

Vogella. Introduction to Java programming – Tutorial. Disponível em: <<https://www.vogella.com/tutorials/JavaIntroduction/article.html>>. Acesso em: 10 de Jun. 2021.

APÊNDICE A – DOCUMENTO DE VISÃO

Documento de Visão

Histórico da Revisão

Data	Versão	Descrição	Autor
24/05/2020	1.0	Criada primeira versão do documento.	Felipe Emerim Leal
24/05/2020	2.0	Atualização do documento para se adequar as novas necessidades do projeto.	Felipe Emerim Leal
22/06/2021	2.1	Atualização do documento para se adequar as novas necessidades do projeto.	Felipe Emerim Leal
02/07/2021	3.0	Alterações nos usuário e pequenas correções.	Felipe Emerim Leal

1. Objetivo

O propósito deste documento é coletar, analisar e definir as necessidades de alto-nível e características do sistema, focando nas potencialidades requeridas pelos afetados e usuários-alvo, e como estes requisitos foram abordados no sistema. A visão do sistema documenta o ambiente geral de processos desenvolvidos para o sistema, fornecendo a todos os envolvidos uma descrição compreensível deste e suas macro-funcionalidades. O Documento de Visão documenta as necessidades e funcionalidades do sistema.

2. Descrição do Problema

O problema de	insegurança das pessoas em transitarem por alguns pontos da cidade em determinados horários
afeta	a comunidade
cujo o impacto é	ocorrência de crimes devido ao não conhecimento da população sobre possíveis pontos de risco da sua cidade
uma boa solução seria	uma plataforma que apresente de forma organizada os dados históricos de ocorrências de crimes, onde as próprias vítimas ou pessoas que visualizaram as situações possam cadastrá-las para que os demais possam visualizar e dessa maneira terem a possibilidade de evitar lugares com periculosidade alta, consequentemente reduzindo o índice de crimes ao evitar andar por áreas com conhecido número elevado de ocorrências.

3. Definição das Partes Interessadas

Usuários

Nome	Descrição	Responsabilidades
Usuário	Usuário do sistema antes de se cadastrar	Este usuário poderá filtrar e visualizar o mapa de ocorrências.
Usuário Autenticado	Usuário cadastrado no sistema	Este usuário será responsável por alimentar o sistema com os cadastros de ocorrências e denunciar conteúdos impróprios.

Moderadores

Nome	Descrição	Responsabilidades
Moderador	Usuário do sistema com privilégio de moderador	Desativar usuários autenticados e alterar status de ocorrências.

Administrador

Nome	Descrição	Responsabilidades
Administrador	Usuário do sistema com privilégio de administrador	Gerenciar usuários moderadores, alterar status de ocorrências, desativar ocorrências e gerenciar denúncias.

Time de Desenvolvimento

Nome	Descrição	Responsabilidades
Felipe Emerim Leal	Analista de sistemas e desenvolvedor do projeto	Elaboração do documento de visão e diagramas de casos de uso e Entidade-Relacionamento. Desenvolvimento e gerenciamento do projeto.

4. Descrição do Produto

Para	todas as pessoas
Que	que necessitam se informar sobre pontos com alta taxa de criminalidade
O	CrimeSpot
É um	sistema de mapeamento criminal colaborativo
Que	irá coletar dados fornecidos por usuários para criar um mapa da criminalidade que pode ser visualizado por qualquer pessoa, a qualquer momento. Ainda será possível parametrizar a visualização de forma a mostrar no mapa apenas o que traz valor para o usuário.
Ao contrário	do CrimeMapping.com
Nosso produto	planeja reduzir a complexidade de navegação, de forma que qualquer pessoa pode criar um cadastro no sistema e incluir ocorrências. Também existirá a possibilidade de registro de crimes sem boletim de ocorrência para casos onde o usuário apenas visualizou a situação mas não esteve envolvido diretamente.

5. Necessidades e Funcionalidades do Produto

1. Gerenciar Usuários

- **Benefício:** Crítico
- **Funcionalidades:**
 1. Cadastrar usuário
 - Atores Envolvidos: Usuário
 2. Editar Cadastro de usuário
 - Atores Envolvidos: Usuário autenticado, Moderador, Administrador
 3. Desativar usuário
 - Atores envolvidos: Moderador, Administrador

2. Gerenciar ocorrências

- **Benefício:** Crítico
- **Funcionalidades:**
 1. Cadastrar ocorrências
 - Atores Envolvidos: Usuário autenticado, Moderador, Administrador
 2. Desativar ocorrências
 - Atores Envolvidos: Usuário autenticado, Moderador, Administrador

3. Manter mapa de crimes

- **Benefício:** Crítico
- **Funcionalidades:**
 1. Visualizar mapa
 - Atores Envolvidos: Usuário, Usuário autenticado, Moderador, Administrador
 2. Filtrar mapa
 - Atores Envolvidos: Usuário, Usuário autenticado, Moderador, Administrador

4. Gerenciar permissões de usuário

- **Benefício:** Importante
- **Funcionalidades:**
 1. Gerenciar permissão moderador
 - Atores envolvidos: Administrador

5. Moderar ocorrências

- **Benefício:** Crítico
- **Funcionalidades:**
 1. Moderar ocorrência
 - Atores envolvidos: Moderador, Administrador
 2. Gerenciar moderação
 - Atores envolvidos: Administrador

6. Denunciar ocorrências

- **Benefício:** Crítico
- **Funcionalidades:**
 1. Criar denúncia de ocorrência
 - Atores envolvidos: Usuário autenticado
 2. Gerenciar denúncias de ocorrências
 - Atores envolvidos: Administrador

6. Proposta de Solução Tecnológica Escolhida

Será desenvolvido um sistema Web utilizando a linguagem [Java](#) para back-end. Utilizaremos o framework Java [Spring MVC](#) que é muito bem conceituado para desenvolvimento back-end de aplicações web. O gerenciador de banco de dados será uma base dados relacional utilizando [PostgreSQL](#) com a qual nos comunicaremos através do [Java Persistence API \(JPA\)](#) que é um conjunto de métodos para banco de dados do Java utilizando o padrão objeto-relacional. o motor de templates do lado servidor será o [Thymeleaf](#) que irá auxiliar a montar as telas com os dados necessários além de facilitar a manipulação dos mesmos. Para o desenvolvimento front-end será utilizado [HTML](#), [CSS](#) e [JavaScript](#). Serão incluídos diversos frameworks de JavaScript como o [jQuery](#) para facilitar operações comuns como, por exemplo, [requisições AJAX](#). Também será necessária uma API de mapas que será o [GoogleMaps](#) para mostrar o mapa de crimes e converter um endereço ou CEP em coordenadas geográficas para poder alimentar este mapa. Para realização dos testes iremos usar o [FluentLenium](#) que é uma ferramenta que nos auxilia a automatizar os testes do sistema.

7. Restrições

Funcionais/Negócio

Usuário:

- Não poderá criar cadastro com cpf, nome de usuário ou e-mail que já existam cadastrados no sistema.

Usuário Autenticado:

- Poderá excluir apenas seus próprios anúncios.
- Não poderá visualizar dados do usuário criador de uma ocorrência.

Moderador:

- Poderá recusar ou aprovar Ocorrências cadastradas.
- Não poderá conceder ou retirar permissão de moderador.

Administrador:

- Poderá conceder ou retirar permissão de moderador.
- Existe apenas um administrador no sistema.
- Poderá desativar ocorrências em qualquer status.

Tecnológicas

- O sistema exige conexão à internet para poder ser utilizado.
- O sistema não garante responsividade para dispositivos móveis.

APÊNDICE B – FORMULÁRIO DE AVALIAÇÃO

Avaliação do CrimeSpot

Sistema web: crimespot.tcc.dev.br

Olá! Preciso que você teste e avalie o CrimeSpot, meu trabalho de conclusão de curso (TCC). O CrimeSpot é uma aplicação web de mapeamento criminal colaborativo. O objetivo é armazenar uma base de dados de ocorrências cadastradas pelos usuários do sistema e apresentar estes dados em um mapa para identificação de possíveis áreas perigosas.

O sistema foi pensado para utilização em um computador, portanto recomendo que você o acesse a partir de um PC ou Notebook.

Como funciona:

Pessoas acessam o sistema com o objetivo de identificar possíveis pontos perigosos em sua região. Ao realizar o cadastro, os usuários se tornam aptos a colaborarem com a aplicação inserindo cadastros de ocorrências com ou sem BO. Para os usuários autenticados, o mapa de ocorrências da tela inicial irá apresentar as ocorrências conforme o endereço cadastrado.

Para controlar conteúdos impróprios dentro da aplicação, todas as ocorrências passarão por uma análise da moderação antes de serem publicadas.

Para contribuir com o meu projeto preciso que realize os seguintes passos:

- Selecione uma ocorrência no mapa de crimes para visualizar seus detalhes
- Realize uma busca com filtros no mapa de crimes
- Crie um cadastro no sistema
- Crie uma ocorrência no sistema
- Edite seu cadastro
- Visualize a lista de ocorrências que você criou
- Visualize os detalhes de uma ocorrência que você criou
- Edite uma ocorrência que você criou

***Obrigatório**

1. Objetivos *

Marcar apenas uma resposta por linha

	Sim	Não
Você conseguiu visualizar os detalhes de uma ocorrência no mapa de crimes?	<input type="radio"/>	<input type="radio"/>
Você conseguiu realizar uma busca com filtros no mapa de crimes?	<input type="radio"/>	<input type="radio"/>
Você conseguiu criar seu cadastro no sistema?	<input type="radio"/>	<input type="radio"/>
Você conseguiu criar uma ocorrência?	<input type="radio"/>	<input type="radio"/>
Você conseguiu editar o seu cadastro?	<input type="radio"/>	<input type="radio"/>
Você conseguiu visualizar a lista de ocorrências que você criou?	<input type="radio"/>	<input type="radio"/>
Você conseguiu visualizar os detalhes de uma ocorrência que você criou?	<input type="radio"/>	<input type="radio"/>
Você conseguiu editar uma ocorrência que você criou?	<input type="radio"/>	<input type="radio"/>

2. Você achou a navegação fácil e intuitiva? *

Marque apenas uma resposta

Sim

Não

3. De acordo com seu grau de satisfação, como você avalia a usabilidade do sistema? (sendo 1 para insatisfeito e 5 para muito satisfeito) *

1	2	3	4	5
<input type="radio"/>				

4. Qual nota você daria para o sistema? *

1	2	3	4	5
<input type="radio"/>				

5. Qual a chance de você utilizar o CrimeSpot para buscar zonas perigosas próximas a você? *

Marque apenas uma resposta

1	2	3	4	5
<input type="radio"/>				

6. Qual a chance de você utilizar o CrimeSpot para realizar cadastros de ocorrências? *

Marque apenas uma resposta

1	2	3	4	5
<input type="radio"/>				

7. Você recomendaria o CrimeSpot a um amigo? *

Marque apenas uma resposta

- Sim
- Não

8. Houve algum erro nos passos ou algo não ocorreu conforme o esperado? *

Sua resposta

9. Gostaria de deixar alguma sugestão?

Sua resposta
