

Map Delitos-PHB: Aplicação Web para o Mapeamento de Delitos Criminais (furto e roubo) na Cidade de Parnaíba-PI

Wesley Da Costa Silva Júnior¹, Leinyllson Fontinele Pereira¹

¹Ciência da Computação - Universidade Estadual do Piauí (UESPI)
Parnaíba-PI-Brasil

Abstract. *This paper describes the development of a responsive web platform to mapping criminal occurrences in real time at Parnaíba-PI city. The website had its usability validated by using a SUS forms, with population and police officers assigned to the task. The application, through a REST API developed for this purpose, receive data from a mobile application and is capable of to provide a map with all occurrences in the city and allows police officers to receive occurrence data and update them in real time.*

Resumo. *Este artigo descreve o desenvolvimento de uma plataforma web responsiva para o mapeamento de ocorrências criminais em tempo real na cidade de Parnaíba-PI. O site teve sua usabilidade validada por meio de formulários do SUS, com população e policiais designados para a tarefa. O aplicativo, por meio de uma API REST desenvolvida para esse fim, recebe dados de um aplicativo móvel e é capaz de fornecer um mapa com todas as ocorrências na cidade e permite que os policiais recebam dados da ocorrência e os atualizem em tempo real.*

1. Introdução

A violência urbana é um problema presente em qualquer ambiente que sofre crescimento populacional acompanhado de déficit na infraestrutura. "Violência, de qualquer forma e natureza, existe em qualquer sociedade, independente das orientações socioeconômicas e culturais"[Algaba 2018]. A tecnologia é um meio importante para que se desenvolva ferramentas capazes de detectar cenários e auxiliar na tomada de decisões no mundo real, para que facilite e agilize o processo de combate a violência.

No estado do Piauí, segundo o relatório de criminalidade do ano de 2019 divulgado pela Secretaria de Estado da Segurança Pública do Piauí, Parnaíba é apontada como a cidade do interior com maior número de casos de violência registrados, sendo a segunda maior cidade do estado [Secretaria de Segurança do Piauí 2020]. O relatório encontra-se disponível ao público por meio do *website* do governo do estado do Piauí e se concentra em apresentar diversos tipos de características criminais que foram identificadas em meio aos dados captados no referido período. Tais dados contabilizam os casos totais da cidade, mas não há informações específicas que suporte o entendimento do comportamento de cada região, como por exemplo: identificar a quantidade de ocorrências, qual a situação das ocorrências encontradas e obter dados durante um intervalo de tempo superior ou inferior a um ano.

O departamento de segurança da cidade dispõe do método padrão para o registro de um boletim de ocorrência, através de comparecimento ao local físico destinado ao departamento e preenchimento de um formulário impresso em papel, através de um servidor

público encarregado. Os dados preenchidos não estão disponíveis digitalmente, impossibilitando que o departamento possa executar uma demanda sem investir demasiado tempo na busca manual de boletins relacionados. Há a possibilidade de informar um ocorrido por meio de um formulário *online*, entretanto, isso ainda incide em falta de estruturação e indexação dos dados informados, pois os dados são armazenados em formato de planilha. Dessa forma, mesmo que os dados sejam digitais, o modelo de busca ainda é executado de forma manual.

Atualmente, com a disponibilidade de dispositivos móveis como *smartphones* e *tablets* que utilizam sistemas operacionais, como por exemplo o *Android*, é possível viabilizar esse processo de forma digital, através do desenvolvimento de uma ferramenta que permita o cidadão acessar e cadastrar ocorrências de acordo com a necessidade. A ferramenta pode ser estruturada tanto em forma de *website* como aplicativo móvel, dessa forma a aplicação facilitaria o processo de informar as entidades que um fato ocorreu, não tornando necessário que o cidadão faça algum deslocamento até a entidade de segurança pública. De outro modo, a entidade de segurança pública tomaria conhecimento das ocorrências e obteria o benefício de acesso rápido a informação, capacidade de efetuar buscas efetivas e com atributos direcionados, e automação do *feedback*(retorno de informação) que o cidadão recebe sobre o estado da ocorrência que por ele foi aberta. A aplicação também deve ser capaz de fornecer o mapeamento dos dados, sem expor dados que sejam de acesso restrito, apenas números, para que os usuários tomem conhecimento da situação com base no número de ocorrências registradas, em função de cada região da cidade.

Este trabalho tem como contribuir para a segurança pública da cidade de Parnaíba-PI, fornecendo uma aplicação WEB para o mapeamento gráfico das ocorrências de furto e roubo, aliado com dados estatísticos que podem ser usados para o planejamento de ações de prevenção desses delitos. A usabilidade da aplicação desenvolvida foi avaliada por meio de um questionário aplicado com os usuários, o resultado obtido através do questionário SUS determina que quais pontos a plataforma foi eficaz e quais quesitos podem ou devem ser melhorados em relação a usabilidade.

O restante do artigo está organizado da seguinte forma: a Seção 2 apresenta o referencial teórico utilizado no desenvolvimento do trabalho, a Seção 3 apresenta trabalhos relacionados ao tema deste trabalho, a Seção 4 apresenta os requisitos, tecnologias e processo de desenvolvimento da aplicação, a Seção 5 descreve todo o processo de avaliação da usabilidade da plataforma desenvolvida, a Seção 6 apresenta os resultados coletados por meio de questionários respondidos pelos usuários, a Seção 7 contém as considerações finais a respeito do trabalho desenvolvido, discutindo os resultados, a Seção 8 discorre sobre possíveis trabalhos futuros.

2. Referencial Teórico

2.1. API REST

API é o acrônimo para *Application Programming Interface*, esse acrônimo na computação é utilizado para denominar um conjunto de rotinas, focando apenas na funcionalidade, sem transparecer para o desenvolvedor detalhes da implementação. São as APIs que permitem programadores expandirem as funcionalidades de suas aplicações, sem que seja

necessário projetar, desenvolver e testar a própria funcionalidade individualmente, implicando em: velocidade de desenvolvimento, maior foco na própria ferramenta, possibilidade de integração com diversas tecnologias e maior reuso de código. No dia a dia, é muito comum que usuários façam o uso de diversas funcionalidades de várias APIs mesmo sem ter conhecimento de que aquele serviço é provido por meio de uma. Os tipos de serviços providos por meio de APIs são inúmeros, são exemplos: dados de georreferenciamento, sincronia de data e hora, dados de ruas através de busca por CEP, acesso a bancos de dados.

REST é o acrônimo para para **Representational State Transfer**, é um estilo arquitetural para sistemas distribuídos de hipermídia. De acordo com [Fielding 2000], o REST foi desenvolvido definindo o que o sistema precisa como um todo, em seguida, incrementalmente é definido e aplicado constantes aos elementos do sistema para diferenciar a visão do processo da arquitetura.

2.2. *Single-Page Applications (SPAs)*

De acordo com [Shahzad 2017], *Single-Page Applications (SPAs)* são aplicações para a *web* que carregam uma única página HTML capaz de carregar de forma dinâmica o conteúdo da página de acordo com a interação do usuário com menus e barras laterais. Esse tipo de aplicação oferece uma usabilidade semelhante a usabilidade encontrada em aplicativos móveis nativos, as aplicações criadas são fluidas e responsivas, fazem uso de *Javascript* e da tecnologia *AJAX (Asynchronous Javascript and XML)* para enviar e receber dados do servidor.

Em geral, esse tipo de aplicação requisita ao servidor operações simples de persistência de dados, do inglês, *CREATE-READ-UPDATE-DELETE*, ou simplesmente o acrônimo *CRUD*, faz com que tais aplicações sejam chamadas de aplicações *CRUD*.

Os *frameworks SPAs* são capazes de estender funcionalidades do HTML, fazendo com que a linguagem de marcação inclua comportamentos semelhantes ao de uma linguagem de programação, esses comportamentos são denominados de diretivas. As diretivas permitem com que o desenvolvedor, através do HTML, defina e manipule estados, renderize dados de forma massiva e aplique filtragem de conteúdo, dessa forma, as diretivas incrementam a usabilidade do *website* fazendo com que os formulários sejam atualizados automaticamente e de forma instantânea.

2.3. Banco de dados *NoSQL*

Segundo [Corbellini 2016], bancos de dados *NoSQL* vieram como alternativas de armazenamento, não se baseiam em modelos relacionais. Em geral, um banco de dados *NoSQL* não é estruturado, não utiliza um esquema fixo e são de interface simples, permitindo os desenvolvedores migrarem para o uso de forma rápida.

Esse tipo de banco de dados evita o uso de *joins* em nível de armazenamento, e suas operações tendem a ser mais custosas, dessa forma, o desenvolvedor tem que optar entre realizar as operações de *join* na camada de aplicação ou então desnormalizar os dados.

Muitos desses banco de dados são projetados para aplicação em ambientes distribuídos, permitindo a adição de nós a infraestrutura para ampliar a capacidade de

operação do banco de dados, característica conhecida como escalonamento horizontal. Existem algumas categorias que depreendem a estrutura dos dados armazenados.

2.4. SCRUM

”Scrum é um método de desenvolvimento ágil e é um dos mais utilizados. Argumenta-se que o Scrum funciona melhor em times entre cinco e sete pessoas, mas a maioria das técnicas podem ser utilizadas por um único desenvolvedor”[Blom 2012]. O Scrum é um dos métodos mais conhecidos pela indústria de desenvolvimento de software, é iterativo, leve e incremental. Essa metodologia, como uma metodologia ágil, é capaz de se adaptar caso o cliente mude de idéia a respeito dos objetivos do produto, diferente do modelo legado de desenvolvimento em cascata, permite adaptações sem fazer necessário descartar todo o trabalho já elaborado até o momento.

3. Trabalhos Relacionados

O trabalho de [Liu 2017] consiste em um aplicativo para acionar a segurança interna do campus de universidade, que dispõe de redes *Wi-Fi* e *Bluetooth*, através de um aplicativo para dispositivos móveis que foi desenvolvido para utilização por alunos do campus, o departamento de segurança é acionado em tempo real, em caso de uma ocorrência aberta. Devido a grande extensão do campus, a aplicação resolve o problema de identificar locais de ocorrências através do cálculo da posição do aparelho que solicitou ajuda, utilizando as características das redes sem-fio para executar o posicionamento e informar o local correto ao departamento de segurança, acelerando o processo de atendimento de um caso.

Segundo [Algaba 2018], O ”Laguna Patroller” se trata de um aplicativo móvel construído para conscientização pública sobre diversas manifestações de violência e as leis aplicáveis a isso. O objetivo é ajudar a diminuir os casos de violência na província de Laguna. O *website* e o desenho da aplicação foram desenvolvidos para que a: utilização de GPS, confiabilidade no sistema e a adição de gravação de vídeo sejam funcionalidades previstas para o sistema.

O trabalho realizado por [Tello 2020], trata de um sistema proposto para reportar crimes de roubo, com notificação enviada em tempo real. O usuário pode utilizar o smartphone de forma furtiva através do clique do botão de ligar/desligar do aparelho por uma determinada quantidade de vezes para informar que um novo evento acabou de acontecer. O aplicativo é capaz de ativar um sistema de câmeras que compartilham ao vivo a imagem do local ocorrido e auxilia oficiais de emergência.

4. Projeto e Implementação

Esta seção trata do projeto e implementação da aplicação *web*, sua base de dados e de ferramentas relacionadas. O sistema foi projetado utilizando o Scrum como metodologia ágil escolhida para guiar o desenvolvimento da aplicação. A escolha pelo Scrum foi dirigida focando no aspecto de permitir que pequenas funcionalidades fossem rapidamente desenvolvidas e testadas, e um Mínimo Produto Viável (MVP) fosse alcançado em pouco tempo, para que a aplicação alcançasse a validação. Apesar do projeto contar com apenas um desenvolvedor invés de um time, como geralmente ocorre, os conceitos ainda se aplicam de maneira mais produtiva que a encontrada no método legado em cascata.

o desenvolvimento necessitou ser dividido em algumas partes, inclusive para que atender requisitos básicos inicialmente e diminuir o tempo de desenvolvimento da aplicação *web*. A ordem dos tópicos a seguir indicam a sequência de implementação de cada funcionalidade.

4.1. Definição dos requisitos

Os requisitos básicos da aplicação foram definidos com base em pesquisa sobre modelos de formulários de boletins de ocorrência disponíveis publicamente na internet. A extração de dados dos boletins possibilitou a formulação de um formulário específico para a aplicação, esse formulário engloba atributos que mantém o documento curto e simples. Um conjunto de aplicações foi pensado para atender desde o cenário de registro de uma ocorrência não tão recente, como o cenário onde uma ocorrência acabou de acontecer e necessita de imediatismo ao ser informada. Posteriormente em uma reunião marcada com o delegado da polícia militar da cidade de Parnaíba, foi possível validar os atributos do formulário e adquirir mais informações sobre outros possíveis cenários que as aplicações deveria operar.

Desde o início do projeto, duas aplicações foram concebidas como necessárias para a realização do projeto. Uma para o acesso pela população, e a aplicação *web* que é o objeto deste trabalho. A aplicação *web* contém uma parte pública e uma parte restrita aos usuários responsáveis pelo acesso nas entidades de segurança pública que fizessem uso da mesma, inicialmente, por policiais.

A primeira aplicação foi desenvolvida por um membro interno do projeto, nomeada de DelitosPHB, em referência ao nome da cidade para onde foi projetado. Essa aplicação foi desenvolvida para dispositivos móveis utilizando a ferramenta App Inventor, estando disponível apenas no sistema operacional Android. A aplicação permite que um cidadão crie uma conta de usuário e informe, através de um formulário, os dados de uma nova ocorrência, esse formulário é então enviado a API do servidor da aplicação e armazenado no banco de dados.

A parte pública da aplicação *web* foi desenvolvida para mostrar o mapeamento de ocorrências da cidade, dessa forma, ela contém um mapa da cidade de Parnaíba, um campo para seleção de filtros de dados e uma área onde é gerado, em tempo real, um gráfico dos dados obtidos após a seleção do filtro.

A parte restrita aos policiais permite que o usuário possa fazer login, obter uma lista de ocorrências atreladas a um Cadastro de Pessoa Física (CPF), visualizar dados de uma ocorrência e obter a lista das últimas ocorrências registradas no sistema. Esse último recurso foi adicionado para acelerar o processo de busca de casos mais recentes. Esses foram os requisitos necessários para o desenvolvimento do conjunto de aplicações citado. Ambas as plataformas compartilham da mesma API e banco de dados desenvolvidos por este projeto, a única diferença é que atendem a recursos separados, mantendo a consistência com a arquitetura REST.

4.2. Modelagem do Banco de Dados

O projeto utilizou a ferramenta MongoDB como banco de dados. Este é um banco de dados *NoSQL* orientado a documentos. As características do MongoDB justificam a escolha pois é uma ferramenta de código aberto e multiplataforma, por isso permite que o

lado servidor da aplicação possa executar nos mais diversos tipos de máquinas e sistemas operacionais, incluindo FreeBSD. O MongoDB utiliza documentos semelhantes ao formato JSON, com esquemas, por isso a comunicação com a linguagem Javascript no lado do servidor ocorre de maneira natural quando necessário utilizar consultas e funções de agregação de campos. A ferramenta permite que campos em um documento possam ser indexados com índices primários e secundários, pois cada documento contém um identificador do documento chamado de *document ID*, como ilustrado na Figura 1. Devido as características de um banco *NoSQL*, o MongoDB é escalável, um aspecto importante visto que foi esperado um número de acessos alto na aplicação *web*, já que se trata de uma ferramenta que atende toda a população da cidade.

```
_id: ObjectId("5e1fc608080fac05d43c6adc")
type: "Furto"
date: 2020-11-25T07:00:00.000+00:00
description: "Sem descrição"
cpf: "55622534507"
district: "centro"
latitude: -2.91218
longitude: -41.77826
status: "running"
```

Figura 1. Formato do documento destinado a armazenar dados das ocorrências.

Como definido na etapa de requisitos, as aplicações necessitaram apenas de duas de coleções de documentos, uma para armazenar documentos de usuários (descrito na Figura 1) e outra para armazenar o dados de ocorrências (descrito na Figura 2). Os campos que definem um documento de usuário se resumem a: nome de usuário, senha, nome, sobrenome, data de nascimento, CPF e tipo de usuário. Para a definição do documento de uma ocorrência, os seguintes campos são armazenados: tipo de ocorrência, data do ocorrido, descrição do ocorrido, CPF da vítima, nome do bairro, latitude e longitude do local da ocorrência, estado da ocorrência. Os possíveis estados de uma ocorrência são: aberto (quando é imediatamente cadastrado), em andamento e fechado.

```
_id: ObjectId("5e1fc602080fac05d43c6adb")
username: "Aleph"
password: "mail"
firstName: "Aleph"
secondName: "Jake"
birthday: "1999-08-23T03:00:00"
cpf: "55622534507"
permission: "user"
```

Figura 2. Formato do documento destinado a armazenar dados dos usuários.

4.3. Implementação da API REST

Para a comunicação da aplicação *web* com a persistência de dados, foi desenvolvida uma API seguindo o modelo REST de consumo de dados. A API foi desenvolvida na linguagem Javascript e executa no servidor através do ambiente NodeJS. Para lidar com

requisições e respostas HTTP foi utilizado o framework ExpressJS, um framework minimalista e rápido que permite manipular requisições HTTP no servidor de maneira prática e enxuta, que apesar de ser um framework, não induz ao desenvolvedor nenhum padrão ou arquitetura de projeto para a sua utilização.

A API é capaz de manipular o banco de dados através de um interface criada utilizando o padrão de arquitetura *Model-view-viewmodel (MVVW)*. Esse padrão facilita a separação da interface gráfica das demais funcionalidades. O *Model* representa a camada de acesso ao conteúdo. A *View* representa a camada responsável pela interface gráfica. O *view-model* é a camada intermediária que é um elo de ligação entre *Model* e *View*, dessa forma, os dados da camada de usuário são coletados, representados em seus devidos objetos e transferidos para que o *Model* desempenhe a função de persistência ou acesso aos dados.

No caso da API desenvolvida, a camada de *View* não atende diretamente a uma interface gráfica, invés disso, o próprio *script*, que executa o servidor Node.js e o *framework* Express.js, está na posição de *View* na arquitetura e é responsável por escutar a porta de acesso a API e manipular as requisições HTTP. O *view-model* desempenha o mesmo papel, recebendo os dados da *View*, filtra e estrutura esses dados para que o *Model*, que manipula os *scripts* de persistência do MongoDB, os armazene no banco de dados.

Seguindo a descrição da arquitetura REST, a API fornece acesso aos dados através dos recursos, permitindo o uso de *queries* para recuperação de dados. Foram definidos os recursos de acordo com os requisitos não-funcionais da etapa de definição de requisitos, são os recursos: Recurso de Logins, Criação de usuários, Obter um usuário, Obter dados para o mapa, Obter dados dos gráficos, Inserção de ocorrência, Obter lista de ocorrência de um CPF, Obter ocorrência pela *document ID*, Obter últimas ocorrências e Atualizar estado de uma ocorrência.

Em resumo, a API é pequena e simples, pois apenas recebe as requisições HTTP, executa uma busca ou inserção no banco de dados e devolve uma resposta, a resposta pode ser: um dos possíveis estados de respostas padrão do protocolo HTTP, dados do banco em formato JSON.

4.4. Configuração da Single-Page Application

A fatia da aplicação *web* que opera no navegador de internet funciona estruturada em forma de *Single-Page Application*. Para essa tarefa, o *framework* Javascript, Vue.js, foi utilizado devido a sua simplicidade para criação de interfaces gráficas e *single-page applications*, também utiliza a arquitetura *Model-view-viewmodel*.

O recurso *two-way data binding* provido pelo Vue.js permite que os dados sejam automaticamente atualizados dentro das representações de objeto a medida que o usuário interage com a interface, dessa forma a usabilidade se assemelha com um aplicativo nativo e trás uma sensação de resposta imediata para o usuário.

Como o aspecto mais importante em uma SPA se trata da comunicação de dados via AJAX, a biblioteca Axios foi utilizada. Axios é uma biblioteca de comunicação assíncrona para Javascript, funciona tanto no cliente como no servidor, as requisições são manipuladas através de *Promises*. Para isso, o modelo de troca de dados, JSON, foi utilizado devido a flexibilidade de uso, pois o ambiente como um todo utiliza Javascript e o

JSON é uma funcionalidade nativa da linguagem.

Como o nome sugere, uma SPA se trata de um único arquivo ou página. Na prática, se trata de um arquivo HTML que carrega os *scripts* do *framework* em Javascript e da aplicação desenvolvida, esses arquivos são denominados *chunks*. A aplicação desenvolvida aparenta ter várias telas pois o *framework* manipula os *scripts* da aplicação em tempo real, os injetando na página e efetuando diversas trocas de contexto e atualizando os dados quando necessário, essas trocas podem ser configuradas para serem imperceptíveis ou comportar-se semelhante ao acesso de diferentes páginas *web*. Para a tarefa de roteamento, foi utilizado o *plugin* Vue Router, responsável por efetuar as trocas de contexto entre os seguintes componentes: website público, formulário de *login*, página inicial da administração, busca de ocorrências, listagem de ocorrências e exibição de ocorrências.

Como os *scripts* são injetados na página principal, todas as dependências e recursos necessitam de alguma forma de referenciamento para sua utilização ser possível. O Vue.JS por padrão já utiliza a ferramenta webpack para gerenciamento de dependências e *recursos* de forma estática, isso significa que na aplicação desenvolvida, todos os *plugins*, componentes, imagens, *scripts* são importados e disponibilizados dentro dos *chunks* da aplicação.

4.5. Website público: Mapeamento de dados e Geração de Gráficos

O *layout* de todo o *website* foi construído em HTML5 e CSS3, utilizando o pré-processador SASS para aumento da produtividade na escrita de folhas de estilo. O *framework* Bulma atua como *framework* HTML/CSS, minimizando a escrita e teste de componentes. Utiliza uma linguagem de usabilidade já validada, em sua maioria, com componentes padrões na internet. Esse *framework* permite que *websites* responsivos sejam construídos através de uso de *Grid Columns* e *Flexbox*, sem necessitar a escrita de *Media Queries* e poucas adaptações necessitam ser feitas.

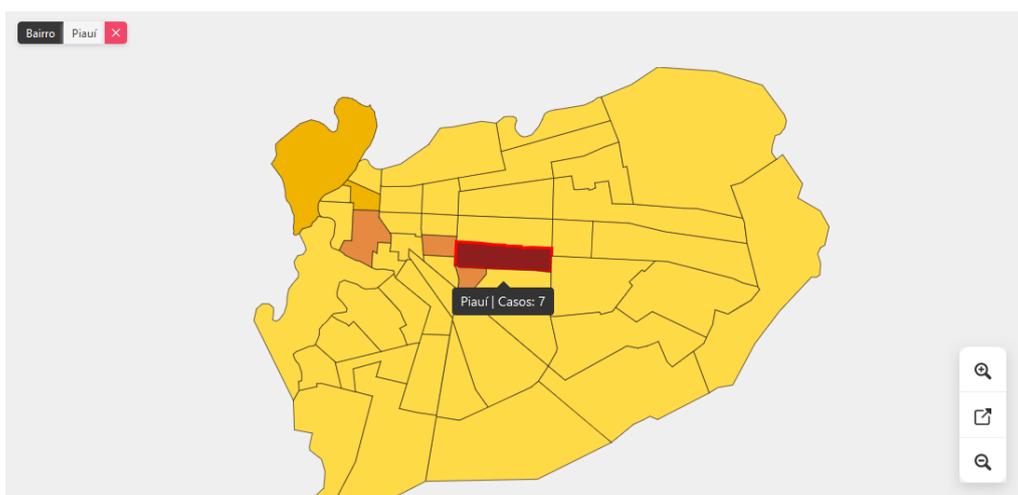


Figura 3. Mapa da cidade com indicação de cores e dados do bairro

Como mostra na figura 3, o mapeamento dos dados foi desenvolvido com base na relação atualizada dos bairros da cidade, relação disponibilizada pela prefeitura da

cidade. O mapeamento foi desenvolvido em formato de gráfico, através do uso do *framework* Raphael.js, um arquivo em formato SVG contendo todos os vetores referentes aos bairros da cidade é renderizado na página *web*, e por meio do uso de bibliotecas como o PinchZoom.js e Hammer.js foi possível o suporte a zoom em formato de pinça e a telas sensíveis ao toque. Dessa forma foi possível desenvolver um mapa da cidade com usabilidade semelhante ao *Google Maps* e *Bing Maps*, fazendo proveito da familiaridade já presente no usuário. Adicionalmente a biblioteca Tippy.js foi utilizada para a visualização da quantidade de ocorrências, ao sobrepor com o mouse ou utilizar o toque sobre um bairro, um balão é criado no ponto do clique com um texto dos dados.

Essa solução foi adotada ao invés de uma API externa de mapas, como o *Google Maps API*, pois implicaria em: custos monetários por acesso, custos de tráfego de rede, maior consumo de recursos do navegador e excesso de informações sendo renderizados. Apesar dos dados de georreferenciamento estarem armazenados no servidor, o acesso é restrito aos policiais, não estando presentes na página pública.

GRÁFICOS

Obtenha dados de ocorrências do mapa.

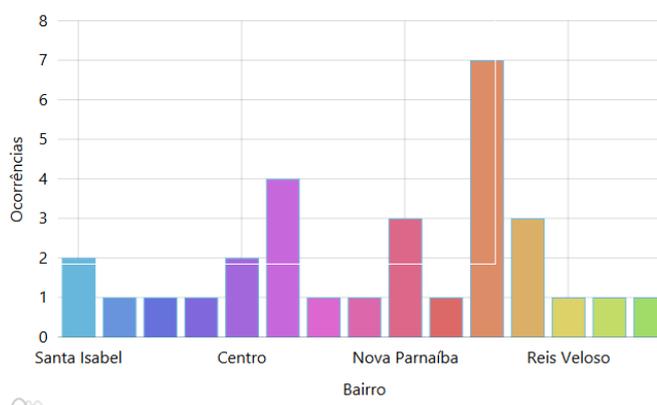


Figura 4. Gráfico gerado através dos dados do mapa

Como mostra na Figura 4, os gráficos foram providos através da ferramenta *amCharts*, biblioteca capaz de criar diversos tipos de gráficos através dos dados fornecidos em formato JSON. De início, foi definido o tipo de gráfico de barras na ferramenta, levando em consideração o alto número de bairros a serem renderizados na tela. Desde a fase de requisitos, os filtros definidos foram dados por: todos os bairros, bairro específico, intervalo de datas, quantidade de ocorrências totais, quantidade de ocorrências abertas, quantidade de ocorrências em andamento, quantidade de ocorrências solucionadas.

Ao carregamento da página, duas requisições são feitas ao servidor, uma para o recurso de dados do mapa que retorna todos os dados de ocorrências em números totais. Após o carregamento dos dados, o total de ocorrências de cada bairro é avaliado, o maior e o menor valor registrado são definidos como uma escala de 0 a 100, o valor da ocorrência é normalizado dentro da escala e então localizado em um intervalo de cinco cores (formando um degradê de amarelo para vermelho), a cor encontrada é aplicada ao respectivo bairro no mapa SVG.

A segunda requisição é feita para o recurso de gráficos (informando o tipo de filtro desejado), os dados recebidos são formatados e enviados para o componente amCharts que se encarrega do agrupamento e renderização.

4.6. Plataforma para os policiais

A plataforma para os policiais foi indexada através de um link para página de login, localizado na barra de navegação do website público, apenas policiais tem o tipo de usuário *admin* que os permitem entrar na plataforma. O foi adicionado ao sistema de roteamento do SPA, os seguintes componentes: tela de login, página principal da administração, página de busca por CPF, página de últimas ocorrências e página de visualização de ocorrência.

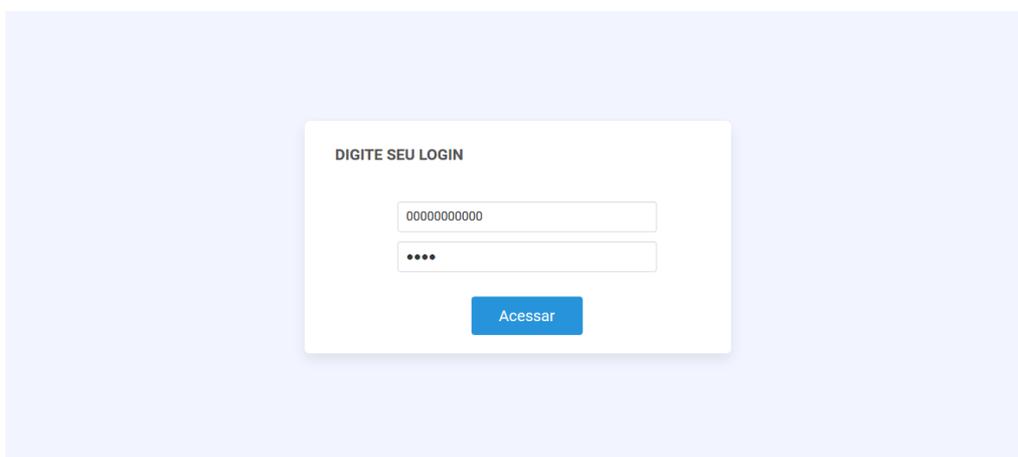


Figura 5. Página de Login da área restrita

Na figura 5, a página de login utiliza o recurso de login da API, que avalia se as credenciais enviadas são válidas para aquele tipo de usuário, caso afirmativo, um token alfanumérico gerado pelo servidor é retornado e armazenado no browser através do recurso *sessionStorage* e o usuário redirecionado para a página de administração, recurso presente na API do Javascript para armazenamento de dados no formato chave-valor, com tempo de vida de uma sessão do usuário. Caso o token não se encontre presente ao acessar uma página de acesso restrito, o sistema de roteamento automaticamente redireciona o usuário para a página de login.

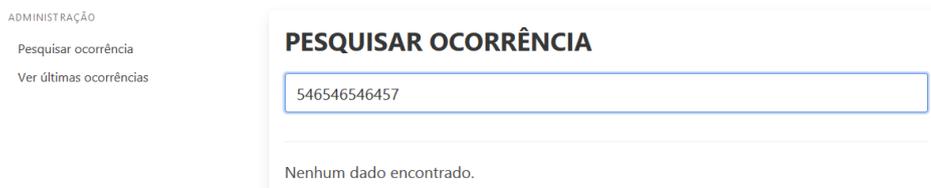


Figura 6. Página da administração e busca.



Figura 7. Página da visualização de últimas ocorrências.

Na figura 6 e 7, os componentes de busca são indexados na página de administração, são eles: busca por CPF, listagem de últimas ocorrências. Ambos componentes são semelhantes, diferindo apenas na necessidade de uma caixa de texto para o campo CPF ser informado na busca. A listagem de ocorrências que ambos apresentam dados como descrição, tipo de roubo e data para auxílio visual na seleção, cada ocorrência tem um botão "Visualizar ocorrência" que redireciona a página para o componente de visualização de ocorrência.



Figura 8. Página de visualização da ocorrência.

Na figura 8 e 9, o componente de visualização de ocorrência apresenta uma página com dados do usuário que a informou e todos os campos do documento de ocorrência registrado no banco de dados. Os dados de georreferenciamento foram utilizados para incorporar um mapa provido pela *Google Maps API*, mostrando a localização informada pelo GPS do aparelho (ou informada pelo usuário). Os dados são oriundos do aplicativo DelitosPHB.

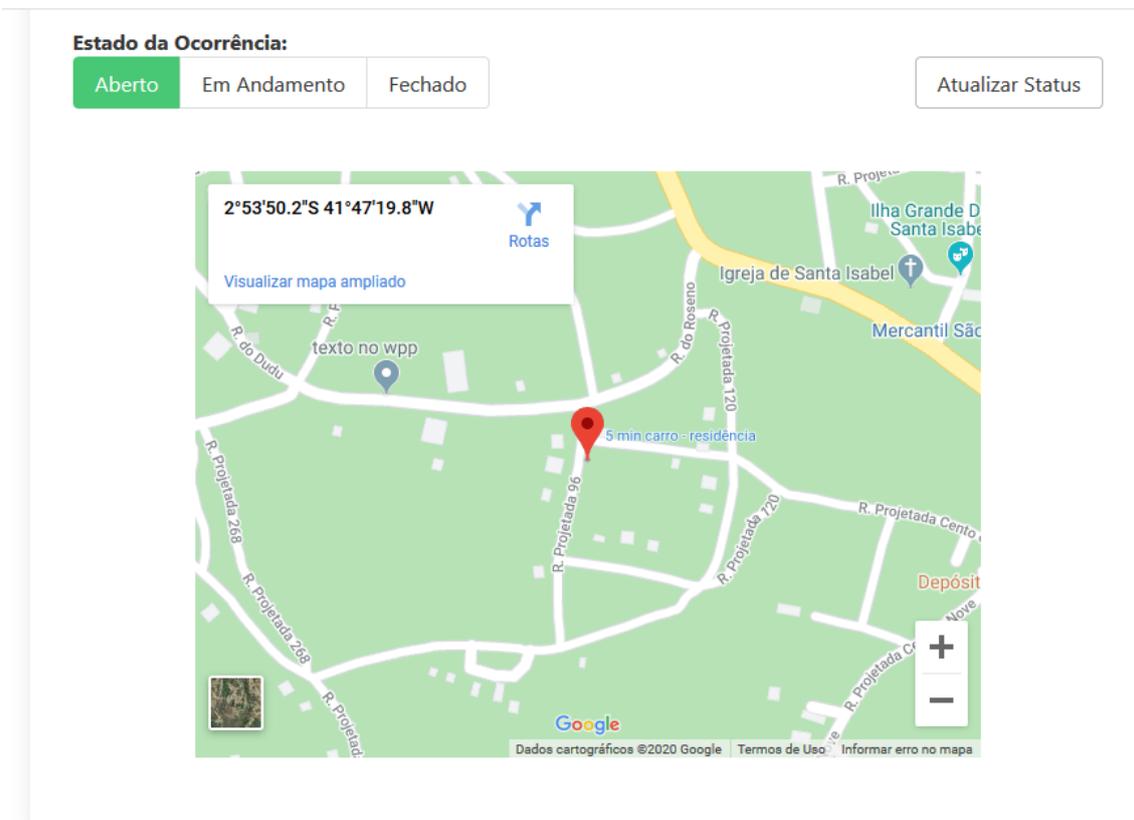


Figura 9. Página de visualização da ocorrência.

Adicionalmente, através da biblioteca *qrcode* do Node.js, um link para o *website* do Google Maps, indicando o ponto informado no GPS, foi incorporado a página para ser escaneado pelo *smartphone*, assim, sendo capturado pelo aplicativo do Google Maps, permitindo outros serviços de busca e localização por GPS.

4.7. Implantação

A plataforma utilizou a ferramenta Git em um repositório privado no site Github para controle de versão e agilidade do processo de implantação, necessitando apenas um *fork* do projeto no servidor. O próximo passo foi realizar o processo de *build* da aplicação. No processo de *build*, a aplicação sai do modo desenvolvedor e entra no modo produção, onde mensagens de erro, alertas e detalhes da implementação param de ser exibidos. Para minimizar o consumo de largura de banda ao transferir os arquivos da aplicação, os *scripts*, folhas de estilos e páginas HTML passam por um processo de minificação, onde comentários, que são desnecessários ao funcionamento da aplicação, são removidos e o nome de variáveis são substituídos por letras do alfabeto para consumirem menos espaço. A aplicação antes do processo de *build* consumia 150 kilobytes, após o processo, foi observado que passou a consumir 8 kilobytes, indicando uma redução de aproximadamente 95% do tamanho original, tornando a aplicação apta para implantação.

O sistema foi disponibilizado *online* através de hospedagem própria utilizando um *netbook* Intel Atom x86-64 de 1.70Ghz com 2 gigabytes de memória RAM e uma conexão com a internet ADSL2+ com *downstream* de 15Mbps e *upstream* de 950kbps. O

servidor utilizou de um serviço externo de DNS dinâmico, capaz de converter o endereço IP da máquina em um nome. Para isso, foi utilizado um serviço gratuito, DDNS, o nome adquirido foi definido como <http://delitosphb.ddns.net:8081/>.

Quando se desenvolve uma aplicação para a *web*, o tempo de carregamento e consumo de recursos da aplicação deve ser sempre levado em consideração, quanto menor, melhor. Através do inspetor do navegador Mozilla Firefox, foi obtido o tempo médio de carregamento do site de 1,66 segundos, com um total transferido de 2,90 megabytes. Os dados indicam que a plataforma desenvolvida é leve o suficiente para consumir poucos recursos computacionais pois operou em hardware legado e com conexão lenta.

5. Avaliação de usabilidade

A avaliação da plataforma consistiu inicialmente em uma aplicação prática da plataforma *web*, agora nomeada como e-Delitos, operando em conjunto com o aplicativo externo, DelitosPHB. Ambas aplicações foram disponibilizadas online através do endereço público <http://delitosphb.ddns.net:8081/> para que os usuários fizessem o acesso e uso da aplicação. Este trabalho trata de uma ferramenta que engloba dois perfis de usuários: uma parte de acesso público para o perfil o cidadão, uma região de acesso restrito para o perfil dos policiais.

Na avaliação prática foram utilizados dois questionários atendendo os requisitos da escala de usabilidade de sistemas, SUS. Segundo [Brooke 1996], O *System Usability Scale* (SUS) consiste em uma escala de dez itens simples, provendo uma visão global a respeito de avaliações de usabilidade que apresentam um cunho subjetivo. Como o SUS é uma escala *Likert*, assume que a pessoa a qual responde se depara com questões de escolhas forçadas, onde uma afirmação é feita e necessita que o respondente indique o grau de aceitação ou discordância através de uma escala de pontos, geralmente entre cinco e sete pontos. "Enquanto escalas *Likert* são construídas dessa forma, as afirmativas que o respondente indica aceitação ou discordância tem que ser escolhidas cuidadosamente"[BROOKE, 1996].

Os formulários desenvolvidos foram dotados de dez afirmativas, cada formulário, a respeito de diversos fatores de usabilidade em relação as aplicações. Cada item é intercalado entre um concordando e outro discordando com a afirmativa, cada afirmativa disponibilizou cinco opções de resposta com determinadas pontuações, são elas: Discordo Plenamente (1 ponto), Discordo Parcialmente (2 pontos), Neutro (3 pontos), Concordo Parcialmente (4 pontos) e Concordo Plenamente (5 pontos).

Os formulários foram disponibilizados *online* através da ferramenta Google Forms. O formulário referente as aplicações DelitosPHB e a parte pública da plataforma aqui desenvolvida, e-Delitos, foi disponibilizada de forma pública. O público foi apto a obter o acesso, efetuar os testes das aplicações e por livre arbítrio preencher o formulário de avaliação, não havendo distinção de indivíduos por parte do pesquisador ou mesmo elemento que permitisse identificação de alguma característica própria do respondente dentro do formulário. Em relação a parte restrita do sistema, o formulário referente foi disponibilizado aos policiais encarregados, oriundos da disponibilização por parte da polícia desde a reunião para definição de requisitos.

O aspecto motivacional do usuário foi levado em consideração, dessa forma, algumas afirmativas foram preparadas com atenção nesse quesito, entretanto, sem modificar

as características do formulário SUS.

6. Resultados e discussões

O método de avaliação apresentado pelo SUS utiliza a pontuação obtida pela resposta de cada item. A pontuação de cada resposta é utilizada no cálculo das notas com base na ordem da afirmativa.

- **Questões ímpares:** pontuação da resposta - 1
- **Questões pares:** 5 - pontuação da resposta

Em ordem, o resultado final é obtido através da soma das respostas das dez afirmativas, multiplicado pela constante 2,5. O intervalo do resultado se encontra entre 0 e 100 pontos. O resultado final é classificado da seguinte forma:

- **0 a 60 pontos:** - Inaceitável
- **60 a 70 pontos:** - Razoável (Necessita de muitas melhorias)
- **70 a 80 pontos:** - Bom (Necessita de pequenos ajustes)
- **80 a 90 pontos:** - Excelente
- **90 a 100 pontos:** - Nível Máximo

Os itens do formulário desenvolvido para ambos os perfis de usuário é descrito a seguir.

1. A aplicação "e-Delitos" é fácil de usar;
2. O layout da tela do e-Delitos é ruim e incompatível com a resolução gráfica do meu dispositivo móvel (celular, tablet, etc.);
3. As funcionalidades presentes no e-Delitos são fáceis de serem encontradas e acessadas;
4. Os textos e informações exibidas no e-Delitos são difíceis de entender e não tem uma linguagem clara e compreensiva, a primeira vista;
5. Acho possível que os futuros usuários possam aprender a utilizar o e-Delitos sozinhos, sem o apoio de materiais de ajuda (manual, tutorial, help, etc.);
6. Não achei o ambiente do e-Delitos agradável e esteticamente atraente;
7. Pretendo utilizar o e-Delitos novamente, com maior frequência;
8. Não acredito que o e-Delitos possa ser útil para a sociedade parnaibana e não contribui para melhorar a questão da segurança pública da minha cidade
9. Me sinto confiante para recomendar o e-Delitos para outras pessoas;
10. O e-Delitos aborda um tema irrelevante para os habitantes de Parnaíba.

Item	Discordo Plenamente	Discordo Parcialmente	Neutro	Concordo Parcialmente	Concordo Plenamente	Resultado
1				2	7	62,5
2	7	1		1		82,5
3				1	8	87,5
4	9					85
5	2		1	4	2	90
6	6	4				85
7	6	3				95
8		2			7	92,5
9	4			2	3	100
10	9					
TOTAL						87

Tabela 1. Pontuação do SUS de usabilidade da parte restrita aos policiais.

O resultado final do questionário aplicado com a população para a parte pública do website é descrito na Tabela 1.

Item	Discordo Plenamente	Discordo Parcialmente	Neutro	Concordo Parcialmente	Concordo Plenamente	Resultado
1					2	97,5
2	2					80
3					2	
4	2					
5		1			1	
6		2				
7					2	
8	2					
9					2	
10	1				1	
TOTAL						88,75

Tabela 2. Pontuação do SUS de usabilidade da parte restrita aos policiais.

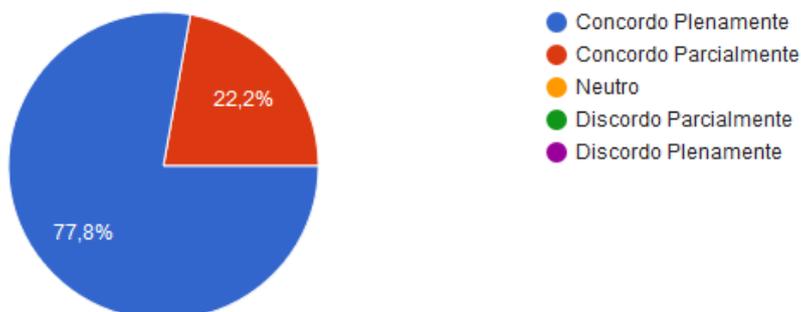
O resultado final do questionário aplicado com os policiais para a parte restrita do website é descrito na Tabela 2.

Observando os valores da área de acesso público da aplicação descritos na Figura 10, mostra que a pontuação final encontra-se entre 80 e 90 pontos, considerado na escala SUS como "Excelente". Quanto na Figura 11, a pontuação da área de acesso restrito aos policiais, o cálculo chegou a um valor entre 80 e 90 pontos, também sendo considerado "Excelente" na escala SUS.

Logo abaixo, serão comentados o resultado, em porcentagem, de cada afirmativa respondida.

1-A aplicação "e-Delitos" é fácil de usar

9 respostas



1- A aplicação e-Delitos é fácil de usar

2 respostas

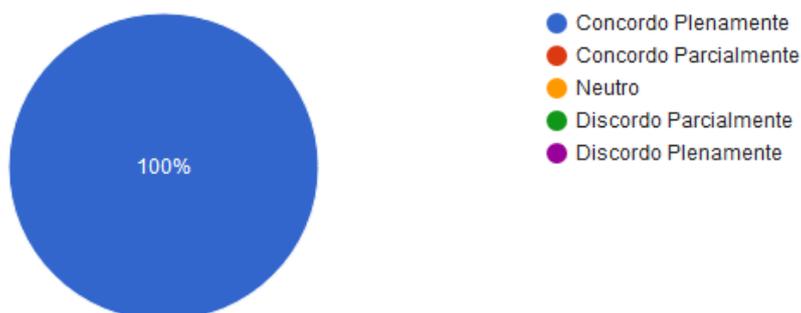
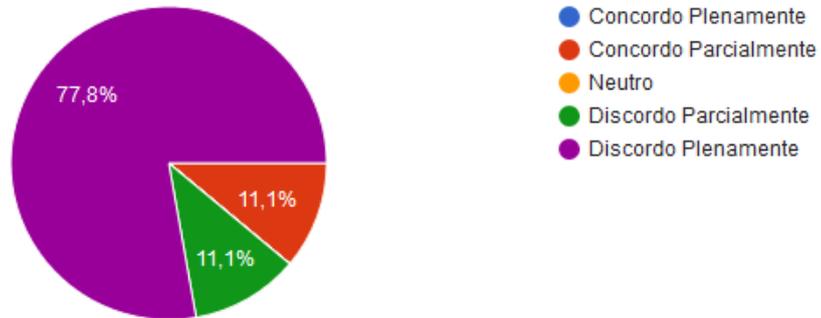


Figura 10. Respostas da população acima. Respostas dos policiais abaixo.

Na figura 10, o item "1 - A aplicação e-Delitos é fácil de usar", 77,8% dos usuários concordaram plenamente que a ferramenta pública de mapeamento é fácil de utilizar e 22,2% concordaram parcialmente a respeito da área pública. Na área restrita, 100% dos usuários concordaram plenamente com a afirmativa.

2-O layout da tela do e-Delitos é ruim e incompatível com a resolução gráfica do meu dispositivo móvel (celular, tablet, etc.)

9 respostas



2-O layout da tela do e-Delitos é ruim e incompatível com a resolução gráfica do meu dispositivo móvel (celular, tablet, etc.)

2 respostas

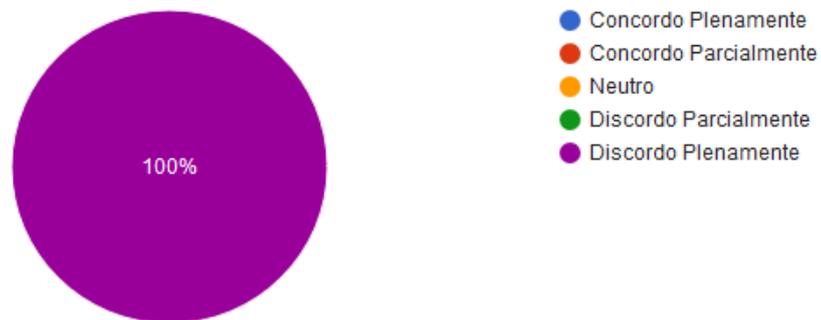
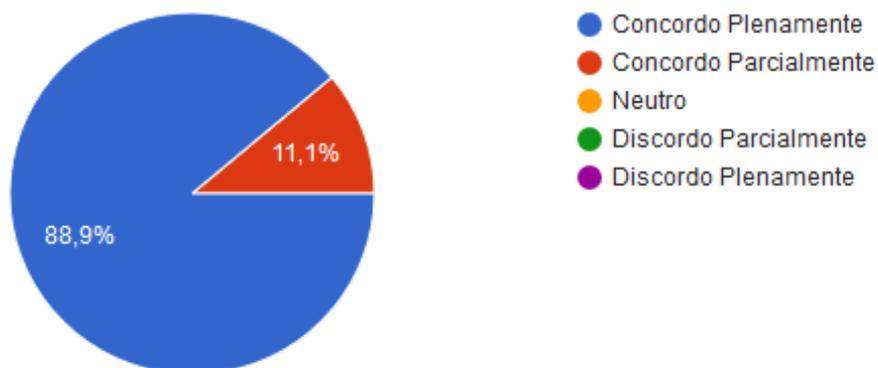


Figura 11. Respostas da população à esquerda. Respostas dos policiais à direita.

Na figura 11, o item "2 - O layout da tela do e-Delitos é ruim e incompatível com a resolução gráfica do meu dispositivo móvel (celular, tablet, etc.)", Os usuários da área pública: 77,8% discordaram plenamente, 11,1% discordaram parcialmente e apenas 11,1% concordaram parcialmente. Na área restrita, 100% dos usuários discordaram plenamente.

3-As funcionalidades presentes no e-Delitos são fáceis de serem encontradas e acessadas

9 respostas



3-As funcionalidades presentes no e-Delitos são fáceis de serem encontradas e acessadas

2 respostas

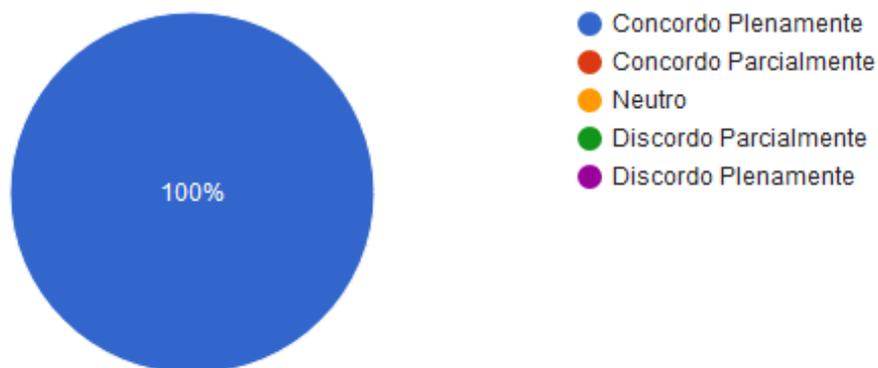
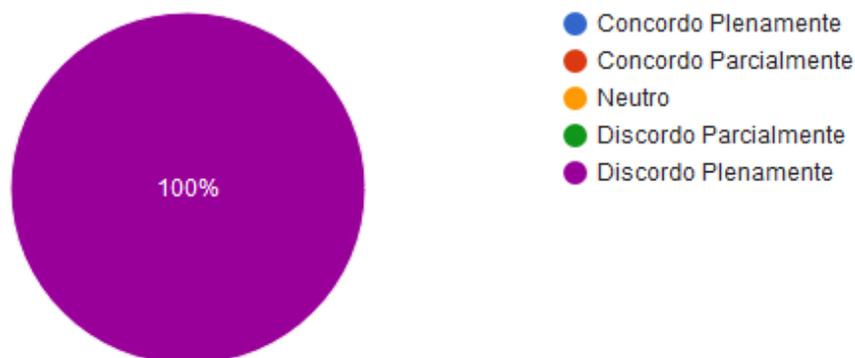


Figura 12. Respostas da população à esquerda. Respostas dos policiais à direita.

Na figura 12, o item "3 - As funcionalidades presentes no e-Delitos são fáceis de serem encontradas e acessadas", 88,9% dos usuários da área pública concordaram plenamente e 11,1% dos usuários concordaram parcialmente. Na área restrita, 100% dos usuários concordaram plenamente.

4-Os textos e informações exibidas no e-Delitos são difíceis de entender e não tem uma linguagem clara e compreensiva, a primeira vista

4 respostas



4-Os textos e informações exibidas no e-Delitos são difíceis de entender e não tem uma linguagem clara e compreensiva, a primeira vista

2 respostas

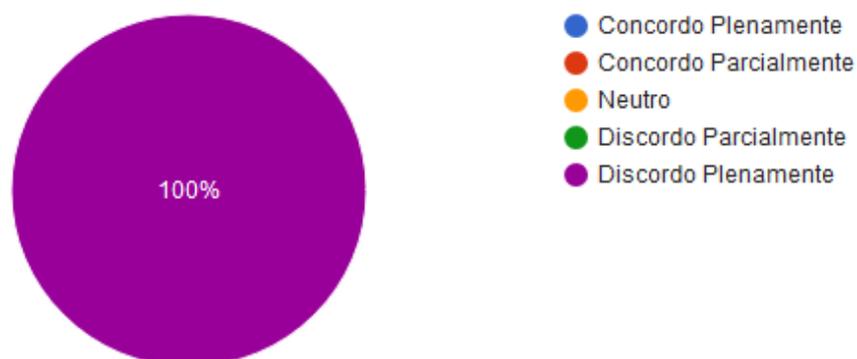
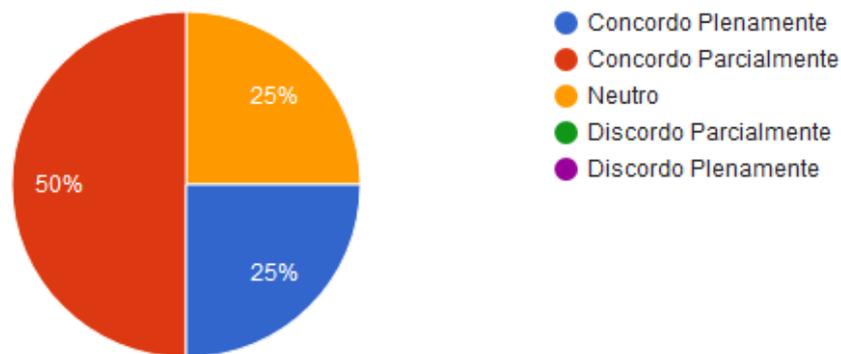


Figura 13. Respostas da população à esquerda. Respostas dos policiais à direita.

Na figura 13, o item "4 - Os textos e informações exibidas no e-Delitos são difíceis de entender e não tem uma linguagem clara e compreensiva, a primeira vista.", ambos usuários de todas as áreas discordaram plenamente da afirmativa.

5-Acho possível que os futuros usuários possam aprender a utilizar o e-Delitos sozinhos, sem o apoio de materiais de ajuda (manual, tutorial, help, etc.)

4 respostas



5-Acho possível que os futuros usuários possam aprender a utilizar o e-Delitos sozinhos, sem o apoio de materiais de ajuda (manual, tutorial, help, etc.)

2 respostas

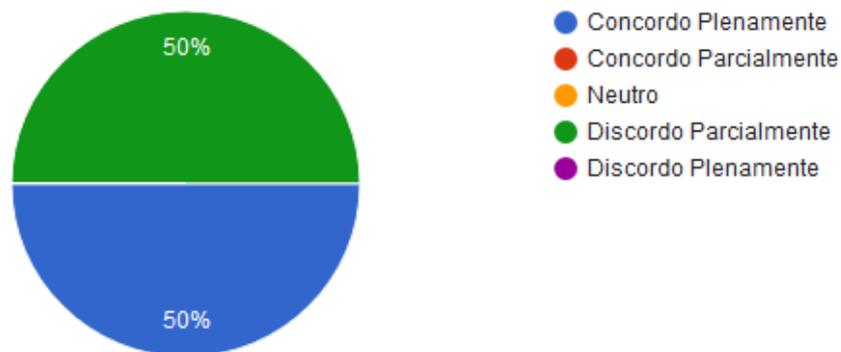
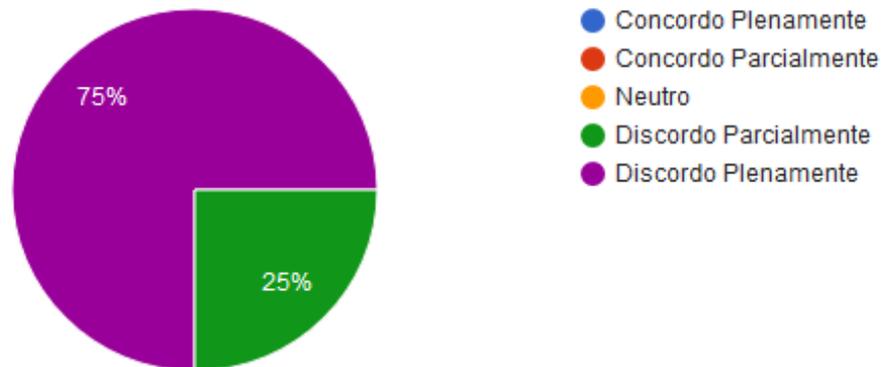


Figura 14. Respostas da população à esquerda. Respostas dos policiais à direita.

Na figura 14, o item "5 - Acho possível que os futuros usuários possam aprender a utilizar o e-Delitos sozinhos, sem o apoio de materiais de ajuda (manual, tutorial, help, etc.)", na parte pública, 50% dos usuários concordaram parcialmente, 25% concordaram plenamente e 25% tem opinião neutra a respeito. Na parte restrita aos policiais, 50% concordaram plenamente e 50% discordaram parcialmente.

6-Não achei o ambiente do e-Delitos agradável e esteticamente atraente

4 respostas



6-Não achei o ambiente do e-Delitos agradável e esteticamente atraente

2 respostas

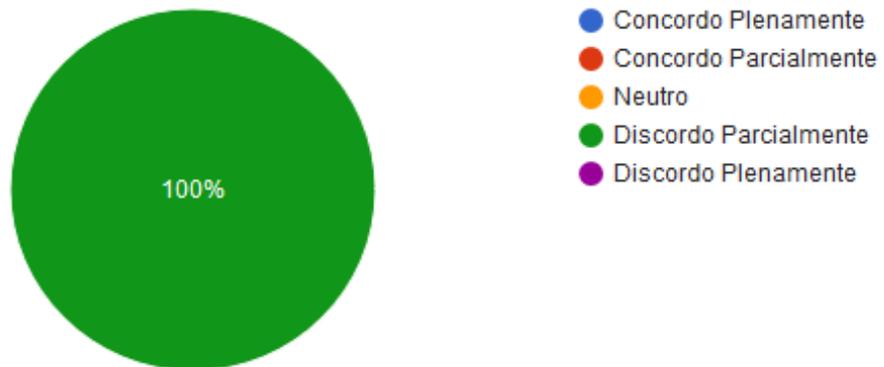
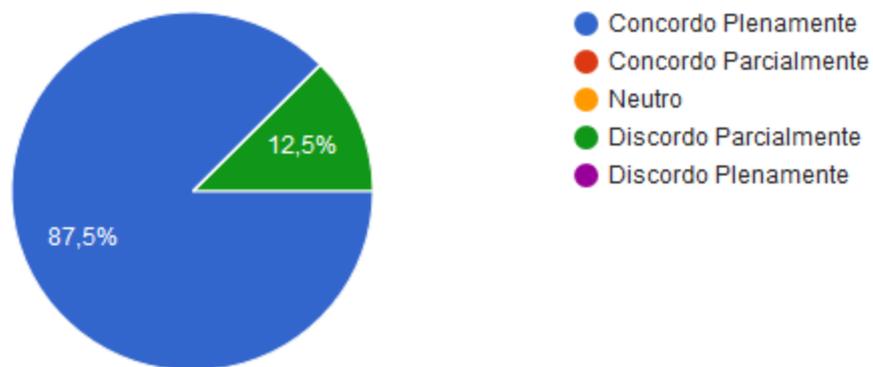


Figura 15. Respostas da população à esquerda. Respostas dos policiais à direita.

Na figura 15, o item "6 - Não achei o ambiente do e-Delitos agradável e esteticamente atraente", na parte pública 75% discordaram plenamente e 25% discordaram parcialmente. Na parte registra aos policiais 100% discordaram parcialmente.

7-Pretendo utilizar o e-Delitos novamente, com maior frequência

8 respostas



7-Pretendo utilizar o e-Delitos novamente, com maior frequência

2 respostas

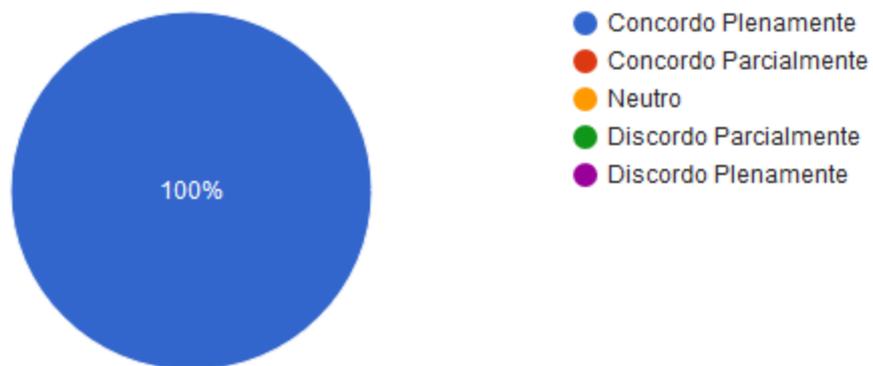
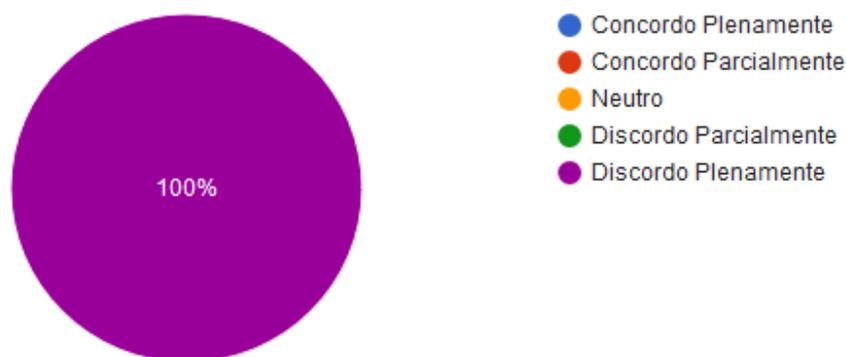


Figura 16. Respostas da população à esquerda. Respostas dos policiais à direita.

Na figura 16, o item "7 - Pretendo utilizar o e-Delitos novamente, com maior frequência", na parte pública, 87,5% dos usuários concordaram plenamente e 12,5% discordaram parcialmente. Na parte restrita aos policiais, 100% concordaram plenamente.

8-Não acredito que o e-Delitos possa ser útil para a sociedade parnaibana e não contribui para melhorar a questão da segurança pública da minha cidade

4 respostas



8-Não acredito que o e-Delitos possa ser útil para a sociedade parnaibana e não contribui para melhorar a questão da segurança pública da minha cidade

2 respostas

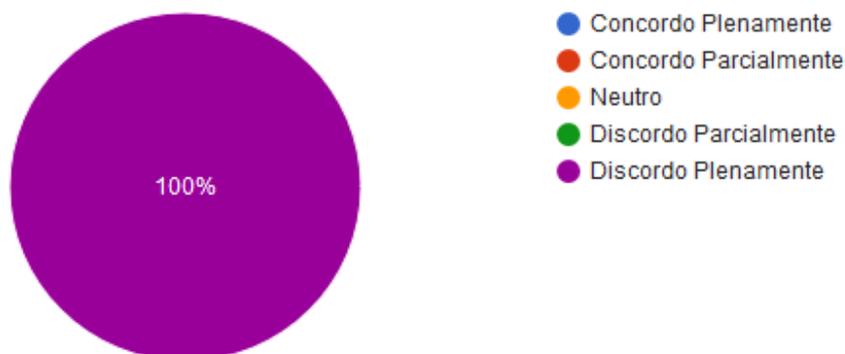
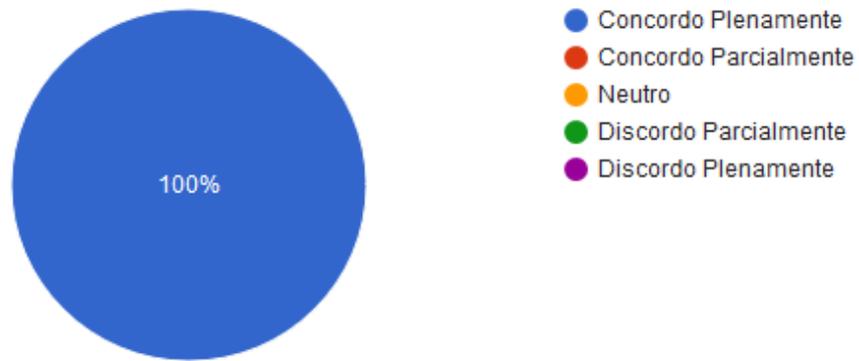


Figura 17. Respostas da população à esquerda. Respostas dos policiais à direita.

Na figura 17, o item "8 - Não acredito que o e-Delitos possa ser útil para a sociedade parnaibana e não contribui para melhorar a questão da segurança pública da minha cidade", 100% dos usuários discordaram plenamente.

9-Me sinto confiante para recomendar o e-Delitos para outras pessoas

8 respostas



9-Me sinto confiante para recomendar o e-Delitos para outros policiais, por que ele disponibiliza informações importantes para o planejamento de ações de segurança preventiva

2 respostas

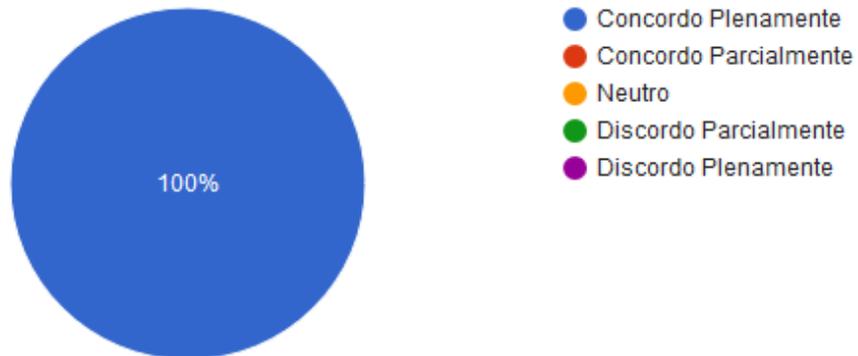
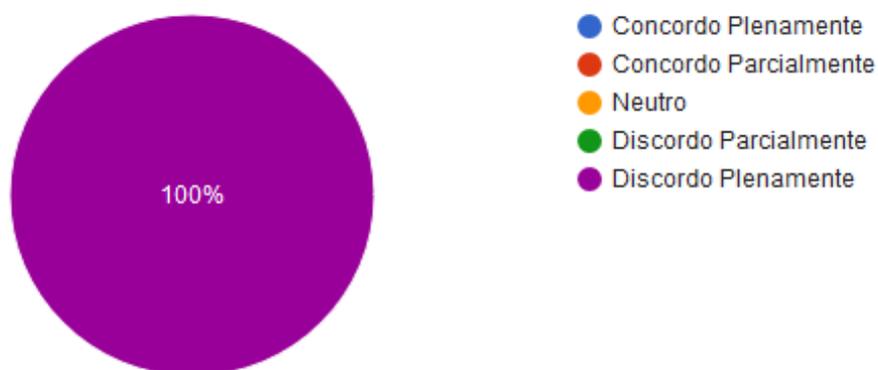


Figura 18. Respostas da população à esquerda. Respostas dos policiais à direita.

Na figura 18, o item "9 - Me sinto confiante para recomendar o e-Delitos para outras pessoas.", 100% dos usuários concordaram plenamente.

10-O e-Delitos aborda um tema irrelevante para os habitantes de Parnaíba

4 respostas



10-O e-Delitos aborda um tema irrelevante para os habitantes de Parnaíba

2 respostas

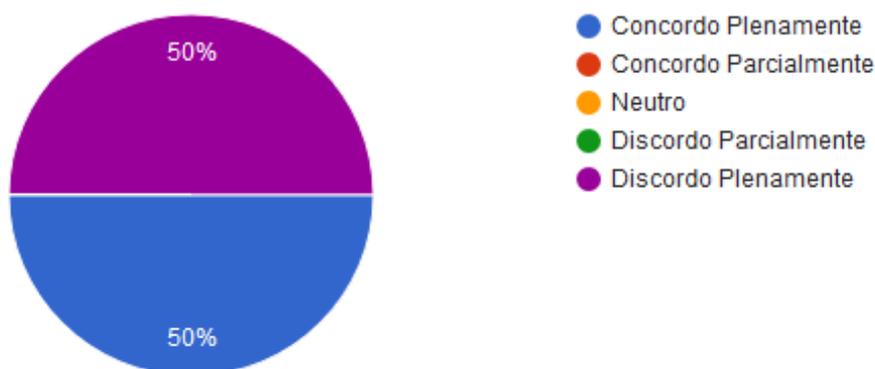


Figura 19. Respostas da população à esquerda. Respostas dos policiais à direita.

Na figura 19, o item "10 - O e-Delitos aborda um tema irrelevante para os habitantes de Parnaíba.", na parte pública 100% dos usuários discordaram plenamente, e na parte restrita aos policiais, 50% discordaram plenamente e 50% concordaram plenamente.

7. Considerações finais

Observando os resultados apresentados pela pesquisa, onde a aplicação atendendo dois perfis diferentes de usuários, obteve pontuação entre 80 e 90 pontos, sendo considerado excelente, pode-se afirmar que a ferramenta atingiu o objetivo ao ser validada pelo estudo e teve sua usabilidade confirmada.

Através dos trabalhos relacionados foi possível observar que apesar de todo o contexto social, uma ferramenta como a desenvolvida neste trabalho se faz necessária até em contexto internacional. Dessa forma, reforçando a motivação em investigar mais maneiras de desenvolver ferramentas públicas que auxiliem no combate à violência urbana.

A ferramenta, apesar de conseguir ter atendido a maioria dos itens pontuados pelos formulários, pode ser melhorada no quesito responsividade para dispositivos móveis, documentação, aumento de qualidade da informação através de textos e subtítulos, e possível implementação de um *tour* de usuário, servindo de guia na primeira utilização do sistema.

8. Trabalhos Futuros

Como proposta de trabalho futuro, através da reunião executada na fase de definição de requisitos, foi captada a necessidade para que a ferramenta seja capaz de lidar o anonimato do cidadão ao informar uma ocorrência, o desafio implica em detectar e filtrar a veracidade das ocorrências.

Por parte dos policiais, foi indicada uma funcionalidade de obtenção de imagens da rua onde a ocorrência foi registrada, caso possível, registrando a imagem de frente do local.

Por último, a expansão dos tipos de ocorrência, reestruturando a aplicação para atender outros tipos de violência, com possível capacidade de integração com outros agentes de combate a violência e/ou acidentes.

Referências

- [Algaba 2018] Algaba, M. F. S. (2018). Laguna patroller: Mobile application for public awareness about violence with global positioning system and image processing. *IEEE*.
- [Blom 2012] Blom, M. (2012). Is scrum and xp suitable for cse development? *International Conference on Computational Science*.
- [Brooke 1996] Brooke, J. (1996). Sus - a quick and dirty usability scale.
- [Corbellini 2016] Corbellini, A. (2016). Persisting big-data: The nosql landscape. *Elsevier*.
- [Fielding 2000] Fielding, R. T. (2000). *Architectural Styles and the Design of Network-based Software Architectures*. PhD thesis, UNIVERSITY OF CALIFORNIA IRVINE.
- [Liu 2017] Liu, K. (2017). Location-aware smart campus security application. *IEEE*.
- [Secretaria de Segurança do Piauí 2020] Secretaria de Segurança do Piauí (2020). Relatório de criminalidade 2019.
- [Shahzad 2017] Shahzad, F. (2017). Modern and responsive mobile-enabled web applications. *Elsevier*.
- [Tello 2020] Tello, M. B. (2020). Design and implementation of a panicalert system to notify theft events. In editor, T., editor, *Advances in Emerging Trends and Technologies*, volume 2, pages 303–312. Springer, The address of the publisher, 2 edition.