

# CityGuardian – Uma Ferramenta para Gerenciamento de Problemas em Cidades

Luana Silva Lima  
Instituto Federal de Educação, Ciência e  
Tecnologia da Bahia  
Rua Emídio dos Santos, S/N, Barbalho  
Salvador, Bahia  
luanalima@ifba.edu.br

Renato Lima Novais  
Instituto Federal de Educação, Ciência e  
Tecnologia da Bahia  
Rua Emídio dos Santos, S/N, Barbalho  
Salvador, Bahia  
renato@ifba.edu.br

## RESUMO

A atividade contínua e crescente nos ambientes urbanos gera diversas ocorrências diariamente. Como exemplos tem-se semáforos com defeito e vazamentos de encanações nas vias públicas. Dada a aleatoriedade com que esses eventos podem acontecer, não é trivial para os órgãos públicos, por exemplo, saberem em tempo aceitável cada uma das ocorrências. Existem aplicações móveis disponíveis para dar suporte a essa tarefa, entretanto, não se tem conhecimento de como os dados fornecidos são gerenciados. Assim, o presente trabalho visa apresentar o *CityGuardian*, um sistema para gerenciamento de problemas em cidades. O sistema funcionará baseado em dois módulos: *mobile* e *web*. Através de um aplicativo *Android*, o indivíduo pode consultar e informar ocorrências na cidade, enviando dados da localização, textos ou imagens. Foi proposta uma forma alternativa de reportar ocorrências, através de um marcador que pode ser arrastado pela tela. Além disso, em casos emergenciais, o usuário poderá ver a sua rota mais recente, facilitando a localização do ponto exato onde houve a ocorrência. Estas informações serão tratadas pela aplicação *web*, que utiliza técnicas de visualização da informação para enriquecer a experiência do usuário em lidar com os dados registrados, facilitando a gestão dos problemas ocorridos. Dessa forma, as entidades competentes, como prefeituras e secretarias, podem tomar decisões eficazes a fim de solucionar os conflitos. Para avaliar a ferramenta, realizaram-se testes com seis usuários, que reportaram ocorrências em seus trajetos durante uma semana. Os resultados coletados foram visualizados através dos recursos da ferramenta *web*.

## Palavras-chave

Dispositivos móveis, problemas em cidades, visualização, informação

## 1. INTRODUÇÃO

Segundo dados de 2014 do Instituto Brasileiro de Geografia e Estatística (IBGE), a cidade de Salvador (BA) possui uma

população de 2.902.927 habitantes, caracterizando-a como a terceira maior e mais populosa cidade do Brasil [13]. Assim, é natural que as mais diversas situações de conflito ocorram, e nem sempre o cidadão conhece o melhor caminho para direcionar as suas requisições. Algumas soluções de resposta não imediata são aplicadas pela Prefeitura de Salvador, como o Sistema de Relacionamento ao Cidadão Fala Salvador [14]. Esse sistema integra ouvidorias, centrais de atendimento por telefone e presenciais para centralizar as demandas. Outro projeto, implantado em janeiro de 2015, é o Ouvindo Nosso Bairro, que promove encontros nos bairros para que os seus residentes expressem necessidades, opiniões e sugestões de melhoria para a comunidade [15]. Apesar de serem soluções interessantes, elas são limitadas, por não permitir uma comunicação rápida e difusa, a fim de que várias pessoas tomassem conhecimento de que a ocorrência foi reportada, e de seu direcionamento em relação às entidades competentes.

Outras alternativas podem ser agregadas às soluções já existentes. Um exemplo é a popularização e o acesso aos diversos recursos dos dispositivos móveis, que crescem a cada dia. Com isso, uma prática que se tornou comum é o compartilhamento de informações na *internet*, seja transmitindo a sua localização ou imagens do cotidiano.

Neste contexto, o presente trabalho visa apresentar o *CityGuardian*, um sistema para gerenciamento de problemas em cidades. O sistema funcionará baseado em dois módulos: *mobile* e *web*. Para isso faz uso de recursos trazidos pelos dispositivos móveis para auxiliar o cidadão a reportar os problemas na sua cidade. Isso possibilita manter um canal centralizado, onde as informações enviadas serão vistas por quem pode solucioná-las, além de promover um meio rápido e eficaz para a população direcionar eventuais ocorrências. As propostas apresentadas por este trabalho são: i) mostrar a rota realizada pelo usuário na última hora, a fim de facilitar a localização do problema para reportá-lo em momento posterior; ii) ao invés de usar um botão para reportar ocorrências em outros locais, utilizar um ícone marcador que pode ser deslizado pela tela.

A aplicação proposta é baseada em dois módulos: *mobile* e *web*. O *CityGuardian mobile* permite ao usuário reportar ocorrências na sua cidade através de um dispositivo que utilize o sistema operacional *Android* [25]. Atualmente, existem aplicações móveis disponíveis para reportar ocorrências em cidades. Este trabalho propõe uma maneira alternativa

de utilização do mapa para realizar este objetivo. As ocorrências podem ser reportadas por meio de um marcador, que é arrastado pela tela para selecionar o ponto desejado no mapa. Além disso, a rota mais recente realizada pelo indivíduo pode ser visualizada. Este recurso permite que uma ocorrência seja reportada em um momento posterior, facilitando a localização em casos emergenciais. Caso o usuário precise sair rapidamente do local exato da ocorrência, ao chegar em local seguro, pode-se encontrar mais facilmente no mapa o ponto onde o problema aconteceu.

O *CityGuardian web* é responsável por gerenciar as informações obtidas, de forma a comunicar à entidade competente o problema ocorrido. Foi feito uso de técnicas de visualização da informação [19] para permitir a melhor compreensão dos dados obtidos através do aplicativo *mobile*. O conhecimento sobre a dinâmica das ocorrências na cidade facilita a tomada de decisões por parte dos órgãos públicos, bem como o planejamento de estratégias que tragam benefícios à população.

Para avaliar este trabalho, foi realizado um questionário com oito perguntas acerca da versão móvel do sistema. Usuários tanto de *smartphones* quanto de *tablets* com versões distintas do Android opinaram sobre os resultados do uso da aplicação. Do ponto de vista do módulo *web*, foi realizado os testes, sob a perspectiva dos autores deste trabalho, das visualizações produzidas considerando os dados reais enviados pelos usuários da aplicação *mobile*.

Além desta introdução, este trabalho está organizado da seguinte maneira: a Seção 2 trata da fundamentação teórica, abordando os principais conceitos estudados durante a elaboração do trabalho. A Seção 3 mostra os trabalhos relacionados ao tema, fazendo a análise de aplicações já existentes. A Seção 4 descreve a visão geral e as etapas de análise e projeto do sistema. A Seção 5 descreve o que foi desenvolvido, listando os requisitos funcionais das duas aplicações. A Seção 6 mostra os procedimentos realizados para avaliação do sistema, e os resultados obtidos. Por fim, a Seção 7 relata as conclusões do trabalho e os trabalhos futuros.

## 2. FUNDAMENTAÇÃO TEÓRICA

Durante o desenvolvimento deste trabalho, foram consultadas diversas fontes da literatura para o embasamento teórico deste trabalho. Nesta seção serão apresentados os principais temas pesquisados para o desenvolvimento da solução aqui proposta.

### 2.1 Computação Móvel e *Android*

O paradigma da Computação Móvel tem como base permitir que usuários de um ambiente tenham acesso a serviços independente de onde estão localizados e das mudanças em sua localização [26]. O usuário não precisa estar conectado a infraestruturas fixas de rede, dessa forma, a computação móvel está intimamente ligada com as comunicações sem fio. Para acessar os serviços de qualquer lugar, basta possuir um computador portátil, um PDA (*Personal Digital Assistants*) ou um celular.

Na evolução dos dispositivos móveis, surgiram os *smartphones*. Desde o lançamento do *iPhone* pela *Apple Inc.* em 2007, a telefonia móvel sofreu grandes modificações. A novi-

dade trazida pelas telas multitoque deu início a uma grande competição nesse segmento. Como resultado, as tecnologias foram evoluindo rapidamente, e nesse contexto, em 2008 surgiu a primeira geração de *smartphones Android* [8].

O início do desenvolvimento do *Android* aconteceu na empresa *Android, Inc* em 2005, e posteriormente o produto foi adquirido pela *Google*. A visão desta companhia em adentrar ao mercado dos dispositivos móveis moveu a mesma a firmar diversas parcerias. Assim surgiu a *Open Handset Alliance*, um grupo formado atualmente por 84 empresas, dentre elas as operadoras móveis, fabricantes de aparelhos, empresas de semicondutores, *software* e comercialização. O foco foi (e é) desenvolver tecnologias móveis em um padrão aberto e comum, diminuindo os custos e melhorando a experiência para o usuário final [4]. Esse intuito pôde ser alcançado pelo fato do *Android* possuir código aberto, permitindo a qualquer pessoa contribuir com seu desenvolvimento. A *Google* é responsável por centralizar e gerenciar as versões que são disponibilizadas de forma a manter a compatibilidade. Apesar disso, tanto as empresas quanto usuários e desenvolvedores podem reportar erros e sugestões que otimizem a cada dia o sistema. O código aberto também permite a sua inovação e personalização [5].

Devido a todas as facilidades para desenvolvimento já citadas, a plataforma *Android* foi a escolhida para o desenvolvimento da aplicação móvel. Também foi considerado o fato dos *softwares* usados no desenvolvimento serem gratuitos, e o extenso uso do sistema no mercado, permitindo que as ocorrências sejam reportadas facilmente, onde quer que o usuário esteja.

### 2.2 *Web Services e Arquitetura REST*

As arquiteturas orientadas a serviço (SOA - *Service-Oriented Architecture*) estabelecem uma abstração da lógica de negócios e tecnologia, resultando em um baixo acoplamento entre esses domínios. O seu uso promove aos sistemas características como interoperabilidade, extensibilidade e reuso potencial de serviços [11].

Um serviço é uma representação computacional de algumas funcionalidades de negócio. Ele aceita uma ou mais requisições e retorna uma ou mais respostas por meio de uma interface bem definida, permitindo que o seu comportamento e a semântica sejam conhecidos pela aplicação que o irá consumir [12].

A principal forma de utilizar a arquitetura SOA é por meio de *web services*: componentes de *software* que seguem um conjunto de normas visando a promover a interoperabilidade entre sistemas. Essas normas definem tanto os protocolos que são usados para a comunicação quanto o formato das interfaces que são usadas para especificar os serviços. Uma das normas fundamentais é a *eXtensible Markup Language* (XML) [28], padronização usada para descrever formatos e modelos de dados e objetos; outro é o HTTP (*HyperText Transfer Protocol*) [29], utilizado para transferências de dados na rede. Ambos são independentes de plataforma.

A simplicidade e facilidade trazida pelo uso do REST (*Representational State Transfer*) [17] têm feito com que esta arquitetura seja a escolhida para a implementação dos *web ser-*

vices [17]. O REST foi apresentado em 2000 na tese de Roy Fieldman [30]. Sua arquitetura é baseada no estilo cliente-servidor, que é a base da *web*, onde clientes (geralmente um navegador de páginas) enviam as requisições, e o servidor processa e envia de volta a resposta solicitada [16].

As características mais interessantes do REST são a escalabilidade, a facilidade de implementação e o uso de formatos de dados mais leves. O REST é baseado em *resources*, uma forma de manipular os dados através dos métodos já conhecidos do protocolo HTTP (como por exemplo GET, POST, PUT, DELETE). Os *resources* são acessíveis através de uma URI (*Uniform Resource Identifier*). O uso das URIs torna fácil a identificação das operações e dispensa o uso de protocolos comuns nos *web services*, como o WSDL (*Web Service Description Language*) [17]. Outra característica é o uso de *hyperlinks* nos *resources*. O estado atual não é armazenado no servidor, mas controlado pelo cliente como um estado do aplicativo e criado pelo caminho que o cliente leva através da *web*. O servidor orienta o caminho do cliente para o *resource* seguinte por meio de hiperlinks: *links* e formulários dentro de representações de hipertexto [18]. O REST também suporta o *caching* já utilizado normalmente pelas páginas *web*, por meio de *proxies* (próximos ao cliente) e *gateways* (próximos ao servidor), diminuindo o acesso direto ao servidor e melhorando a escalabilidade da aplicação que o usa. Por fim, o REST permite o uso de outros formatos de dados além do XML, como o JSON (*JavaScript Object Notation*) [24]. O uso do JSON é cada vez mais comum, por ser mais leve de usar, além de possuir uma notação mais simples para representar objetos.

Neste trabalho foi utilizado um *web service* REST para centralizar todo o acesso ao banco de dados, além de facilitar a comunicação e integração entre as duas aplicações desenvolvidas. A simplicidade e leveza do JSON também é preponderante para os dispositivos móveis, onde os recursos de processamento e memória são mais limitados. Além disso, o usuário frequentemente contrata pacotes de dados fornecidos pelas operadoras telefônicas, e portanto tem seu consumo reduzido.

### 2.3 Visualização de Informação

A expressão “visualização de informação” foi cunhada pelos pesquisadores da Xerox PARC (*Palo Alto Research Center*) no fim dos anos 1980, para distinguir uma nova disciplina preocupada com a criação de artefatos visuais que visam ampliar a cognição [19]. Desde essa época até os dias atuais, a quantidade de informação a que o indivíduo é submetido diariamente cresceu muito. No intuito de auxiliar a transformação de dados em informações providas de significado e permitir melhor compreensão das mesmas, a visualização de informação se baseia no fato de a percepção humana para elementos visuais facilitar o processo cognitivo, quando comparada a outras formas, como os elementos textuais [19].

As Figuras 1 e 2 exemplificam os elementos e propriedades gráficas que podem ser usados na visualização. Outro fator a ser considerado é a interação do usuário. Informações podem ser filtradas com o uso de técnicas como rotação, translação, *zoom*, barras de rolagem, hierarquia em árvore, dentre outras. Contudo, o que uma visualização deve garantir é a integridade, de forma a minimizar ambiguidades na

interpretação do que é exibido.

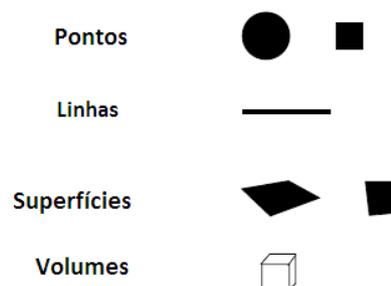


Figura 1: Exemplos de elementos gráficos [19]

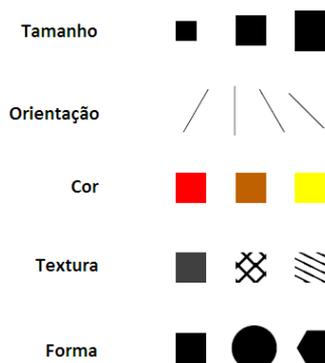


Figura 2: Exemplos de propriedades gráficas [19]

Os conceitos apresentados sobre a visualização da informação foram aplicados para exibição dos dados reportados pela aplicação móvel e *web*, enriquecendo os mapas. No caso do módulo *web*, foram utilizadas metáforas visuais específicas para a apresentação e consequente auxílio à interpretação dos dados obtidos.

## 3. TRABALHOS RELACIONADOS

Foram selecionadas quatro ferramentas que abordam o tema central deste trabalho. Analisou-se o funcionamento das versões disponíveis para o usuário reportar as ocorrências. As principais características de cada uma delas serão descritas nesta Seção.

### 3.1 Ferramentas de Gerenciamento de Problemas

- Cidadera

A Cidadera é uma ferramenta focada em denúncias e reclamações acerca dos problemas urbanos. Está disponível para a *web* e *mobile*. Ela permite que pessoas apresentem inconformidades na cidade por meio de descrições e fotos, nas categorias por ele listadas (é possível também criar uma nova categoria). O usuário pode optar entre o cadastro por *email* ou pela rede social *Facebook*, bem como fazer a denúncia como cidadão anônimo. Outras pessoas também podem demonstrar apoio às denúncias feitas [2].

- Colab

O Colab é uma rede social colaborativa que permite ao usuário disponibilizar informações sobre sua cidade. Através da ferramenta, o usuário tem a possibilidade de fiscalizar, propor e avaliar ações para a cidade. Na opção de fiscalização, o usuário envia, através de fotos e geolocalização, os diversos problemas da cidade que podem ser denunciados. Na opção de propor, os usuários podem apresentar projetos e ideias para melhoria da cidade. Por fim, na opção de avaliação, há uma seção que permite a indicação de notas através de cinco estrelas para um local, considerando os quesitos: geral, equipamentos, limpeza e acessibilidade. A ferramenta está disponível nas versões *web* e *mobile*, e está integrada ao *Facebook*. O ponto positivo disso é o fato de utilizar a principal rede social da atualidade. Por outro lado, obriga o usuário a ter um registro nessa rede [1].

- *PublicStuff*

A *PublicStuff* é uma ferramenta utilizada apenas nos Estados Unidos, que tem como objetivo prover integração entre o governo e a população. Os usuários podem fazer relatos de problemas, pagamentos de contas e solicitar informações sobre os serviços governamentais. Os órgãos competentes ficam responsáveis por dar encaminhamento as solicitações. A comunicação ocorre através da *web*, SMS (*Short Message Service*), aplicativo *mobile* ou ligações. A implantação do sistema tem um custo que varia conforme a população ou investimento necessário para integração com os sistemas já existentes [3].

- Vigilante

O aplicativo *Vigilante* permite usuários reportarem problemas em oito categorias básicas, e em diversas subcategorias, totalizando mais de 250 possíveis indicações. Ela está disponível na versão *mobile*, e possui uma versão *web* em desenvolvimento. O usuário pode realizar o *login* por *Facebook* ou por *e-mail*. A ferramenta promove ainda a interação de outros usuários por meio de opções que permitem demonstrar apoio, informar que viu a ocorrência, e se a ocorrência se trata de um registro falso. Também permite postagens anônimas. Por ser uma ferramenta mais recente em relação às já citadas, a empresa responsável está em busca de parcerias com entidades públicas para utilizar as informações fornecidas pela aplicação *mobile* [10].

### 3.2 Comparativo entre ferramentas

A Tabela 1 lista um comparativo entre as ferramentas estudadas.

Conforme pode ser visto na Tabela 1, existem diversas soluções disponíveis para reportar ocorrências em cidades. Este trabalho propõe uma maneira alternativa de utilização do mapa para realizar este objetivo. O usuário pode optar entre utilizar um botão disponível para reportar a ocorrência na sua localização atual, ou utilizar um marcador, que pode ser deslizado pela tela a fim de facilitar a seleção do ponto desejado no mapa. Outro recurso proposto por esta ferramenta é a visualização da rota mais recente realizada pelo indivíduo. Assim, a localização de uma ocorrência pode ser mais

facilmente encontrada e reportada em um momento posterior. Um exemplo de aplicação desse recurso é em situações de emergência. Caso o usuário precise sair rapidamente do local exato da ocorrência, ao chegar em local seguro, pode-se encontrar mais facilmente no mapa o ponto onde o problema aconteceu.

Um outro ponto a ser avaliado é o fato de não se conhecer as ferramentas utilizadas para gerenciar as ocorrências reportadas. Este trabalho propõe uma ferramenta *web* para gerenciar e facilitar a interpretação das informações pelos órgãos responsáveis, através do uso de visualização da informação. Para melhor análise dos dados, são disponibilizadas três visualizações: *bullet chart*, *bubble chart* e *treemap*.

## 4. O SISTEMA CITYGUARDIAN

O *CityGuardian* é uma ferramenta para suporte ao gerenciamento de problemas em cidades. Para se construir esse sistema foi necessário seguir etapas básicas do desenvolvimento de sistemas (e.g., análise, projeto, implementação) [9]. As subseções a seguir apresentam detalhes de projeto da solução proposta.

### 4.1 Requisitos do Sistema

#### 4.1.1 Requisitos Funcionais

Os requisitos funcionais do sistema são apresentados nas seções 5.2 e 5.3. Essas duas seções discutem em detalhes as funcionalidades dos módulos *mobile* e *web*, respectivamente.

#### 4.1.2 Requisitos Não-Funcionais

Um requisito ou propriedade não-funcional define uma restrição que impacta na maneira como o *software* implementa e cumpre sua funcionalidade [16]. Para o sistema apresentado, foram definidos os seguintes requisitos:

- Usabilidade: ambas as aplicações devem permitir que o usuário possa usufruir de suas funcionalidades com o menor nível de dificuldade e esforço possível. Para atingir esse objetivo, as interfaces gráficas foram desenvolvidas de forma simples, buscando fornecer ao usuário informações diretas que o auxiliem na navegação por meio de *tooltips* e *popups*;
- Compatibilidade: a aplicação *mobile* deverá ser compatível com o máximo possível de dispositivos móveis *Android*. Os dispositivos testados, seus modelos e versões do sistema operacional, bem como os resultados dos testes serão descritos em detalhes na Seção 6.

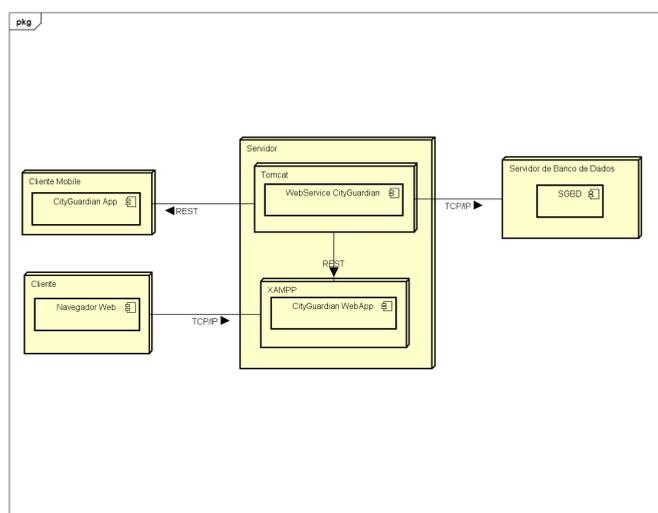
### 4.2 Projeto do Sistema

O sistema utilizou o estilo arquitetural Cliente-Servidor, que é um estilo baseado em camadas. Nesse estilo, o cliente inicia a comunicação enviando uma requisição ao servidor, e este recebe e responde com as informações necessárias, não existindo comunicação entre clientes [16]. Um sistema multicamada permite manter a organização, a separação de conceitos e viabiliza a integração das duas aplicações desenvolvidas.

Na Figura 3 pode-se observar o diagrama de implantação do sistema, mostrando a operação conjunta dos principais módulos da solução proposta. A aplicação *CityGuardian mobile*

**Tabela 1: Comparação das características das diversas ferramentas**

	Colab	Cidadera	PublicStuff	Vigilante	CityGuardian
Login por <i>email</i>	X	X	X	X	X
Login por <i>Facebook</i>	X	X	X	X	
Versão <i>Web</i>	X	X	X	X	X
Registro de Problema Anônimo	X	X		X	X
Interações com redes sociais	X	X	X	X	
Uso do marcador para reportar ocorrências					X
Visualização da rota realizada na última hora					X



**Figura 3: Diagrama de implantação do Sistema**

é o meio de envio e visualização de dados sobre as ocorrências. Ela exhibe, além dos dados reportados pelos usuários, o status da ocorrência. Esta etapa é realizada através da aplicação *web*, onde os gestores visualizam o problema reportado e informam se já foi encaminhado ao órgão responsável ou se já está resolvido. Todas as requisições e respostas são tratadas pelo módulo que implementa o *web service*. Este *web service* atua intermediando o acesso à base de dados comum aos dois módulos, promovendo a integração de ambos.

## 5. DESENVOLVIMENTO DO SISTEMA CITY-GUARDIAN

Esta seção discute detalhes de implementação destacando as tecnologias utilizadas para o desenvolvimento da solução proposta, bem como as interfaces visuais dos módulos *mobile* e *web*.

### 5.1 Tecnologias utilizadas

Esse sistema foi todo desenvolvido com tecnologias sem custo. As principais estão listadas abaixo:

- Servidor para o *web service*: Apache Tomcat 7.0.
- SGBD (Sistema de Gerenciamento de Banco de Dados): MySQL 5.5.
- Servidor da aplicação *web*: Apache XAMPP com o *framework CodeIgniter* [22].

- Biblioteca para as visualizações usadas na aplicação *web*: D3.js [23].

### 5.2 Aplicação Móvel

As principais funcionalidades da aplicação móvel são descritas nas subseções seguintes:

#### 5.2.1 Registro

Inicialmente, o usuário precisará realizar seu cadastro para ter acesso ao sistema. Na tela inicial, o usuário informa o nome, que caso deseje, será visível ao reportar ocorrências, o *email* e a senha. O sistema verifica se o *email* digitado é uma sequência válida (exemplo: usuario@provedor.com ou usuario@provedor.com.br). Em caso afirmativo, o sistema exhibe uma mensagem de confirmação e direciona o usuário para a tela de *login*. Em caso negativo, o sistema exhibe uma mensagem informando que o *email* é inválido. A tela pode ser vista na Figura 4.

#### 5.2.2 Login

Para ter acesso ao sistema, o usuário precisa se autenticar. Na Figura 5 observa-se a tela de *login*, onde o usuário informa seu *email* e sua senha. O sistema verifica se o *email* digitado é uma sequência válida, além de verificar se os dados informados são corretos. Em caso afirmativo, o sistema exhibe uma mensagem de confirmação e direciona o usuário para a tela do mapa onde os problemas são reportados. Em caso negativo, o sistema exhibe uma mensagem informando

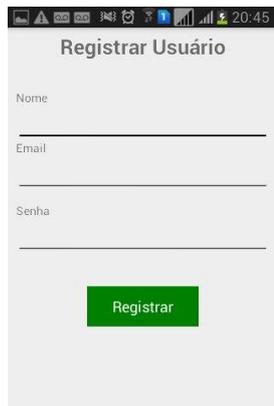


Figura 4: Tela de Registro de Usuário

que o *email* é inválido ou que os dados digitados são incorretos.



Figura 5: Tela de Login da Aplicação

### 5.2.3 Mapa para visualização das ocorrências

Após a realização do login, o usuário é redirecionado para o mapa da sua cidade, centralizado na sua localização corrente. Conforme pode ser visto na Figura 6, um círculo marca a localização atual do usuário, e vários marcadores demonstram as ocorrências já reportadas. O marcador preto em destaque pode ser clicado e arrastado pela tela, a fim de se direcionar um ponto no mapa onde uma nova ocorrência será registrada. Ao clicar em um marcador, uma tela com os dados detalhados da ocorrência será exibida.

### 5.2.4 Reportar Problema

O usuário possui duas maneiras de reportar um problema: ao clicar na opção do menu “Reportar problema aqui”, a tela é exibida e a ocorrência será registrada na localização corrente do usuário. No entanto, se a localização do problema é selecionada no mapa por meio do marcador, o usuário também é redirecionado para esta tela, conforme exemplificado na Figura 7. Nesse momento, o usuário deve selecionar uma das dez categorias disponíveis (Água e Esgoto, Defeito em Via Pública, Eletricidade, Lixo em Local Inadequado, Poluição Sonora, Poluição Visual, Roubo/Furto, Sinalização de Trânsito, Telefone/Internet, Outros), a fim de classificar o



Figura 6: Mapa para visualização e registro de problemas

tipo de ocorrência. Uma descrição textual pode ser fornecida no campo “Descrição”. É possível também anexar uma imagem ao conteúdo reportado. O usuário tem a opção de ocultar seu nome no momento da visualização do registro, ao selecionar o item “Registrar como Anônimo”. Essa possibilidade de reportar um problema em uma região diferente é interessante para os casos de o usuário da aplicação não poder fazer isso no momento específico da ocorrência. Isso pode acontecer por exemplo em casos de emergências, onde o usuário, por questões de segurança, precisa primeiramente se afastar do local onde o incidente está acontecendo.

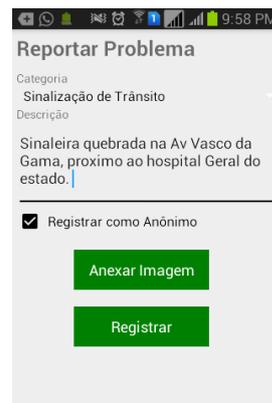


Figura 7: Tela de registro de problemas

### 5.2.5 Visualizar Problema

Ao clicar no marcador da ocorrência exibido no mapa, a tela com os seus principais dados é exibida, conforme pode ser visto na Figura 8.

O usuário poderá optar entre realizar outras ações: visualizar a imagem anexada, caso a mesma esteja disponível; Demonstrar apoio à ocorrência ao clicar no botão Apoiar; visualizar outros detalhes da ocorrência;

### 5.2.6 Visualizar Mais Detalhes

Nesta opção, os detalhes da ocorrência reportados pelo módulo *web* podem ser visualizados, como o *status* do problema: encaminhado ao órgão responsável ou resolvido. Tam-



Figura 8: Tela de visualização dos detalhes principais

bém pode ser vista a quantidade de apoios que ele possui. O resultado pode ser visto na Figura 9. Apesar da implementação utilizar *checkboxes*, será realizada uma implementação com *labels* do *status*.

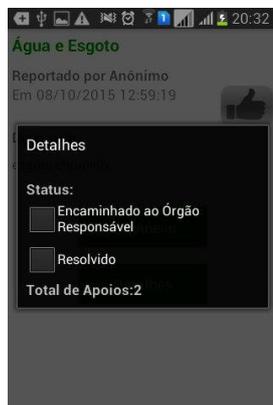


Figura 9: Tela de visualização de mais detalhes do problema

### 5.2.7 Visualizar Rota Recente

Neste item, o usuário pode visualizar o trajeto realizado durante a última hora. As coordenadas são registradas a cada minuto, e salvas em um banco de dados (SQLite) no próprio dispositivo. O usuário pode selecionar se o trajeto foi realizado a pé ou de carro por meio do menu. Através da *Google Maps Directions API* [21], essas informações são utilizadas para calcular e exibir o trajeto aproximado no mapa, conforme a Figura 10. Algumas limitações podem ser observadas. A API depende de conexão à *internet* para seu funcionamento. Caso as coordenadas não estejam sendo registradas por meio de GPS, apenas os pontos de início e fim da rota são registrados, a rota traçada será a considerada mais próxima para se realizar, que pode não corresponder a rota real realizada.

A API permite que apenas oito pontos do trajeto sejam sugeridos, além dos pontos de início e fim da rota. Para selecionar estes pontos, são obtidas todas as coordenadas no intervalo da última hora. A quantidade de pontos obtidos



Figura 10: Tela de visualização da rota realizada na última hora

dividida por 8 resulta no intervalo necessário na escolha dos pontos para auxiliar a composição da rota, se o total de pontos é maior que 10. Abaixo segue o trecho do código do algoritmo usado.

```
String destinations = "&waypoints=";
int size = listaCoordenadas.size();

BigDecimal bd = new BigDecimal(size);
bd = bd.divide(new BigDecimal(8));
bd = bd.setScale(0, BigDecimal.ROUND_HALF_UP);
int intervalo = bd.intValue();

for (int i = 1; i < size - 1; i+=intervalo)
{
    destinations += String.valueOf(
        listaCoordenadas.get(i).getLatitude()
    )
    + "," + String.valueOf(listaCoordenadas
        .get(i).getLongitude());
    if (i != size - 1) {
        destinations += "|";
    }
}
```

## 5.3 Aplicação web

A aplicação *web* é responsável por concentrar os dados dos relatos de problemas enviados pelos diferentes usuários da aplicação *mobile*. Ela pode ser utilizada por exemplo por gestores de cidades no intuito de gerenciar os problemas. A primeira tela da aplicação mostra filtros possíveis de serem utilizados nas ocorrências a serem visualizadas. É possível selecionar a categoria a qual um problema pertence, um intervalo de datas em que o problema foi registrado, além de marcar se o problema está ou não resolvido. Esta tela pode ser vista na Figura 11. O resultado da seleção é exibido em um mapa, onde cada item correspondente aos critérios determinados pode ser clicado para observação de maiores detalhes acerca da ocorrência, conforme a Figura 12. Neste momento o usuário é redirecionado para uma tela onde ob-

serva os detalhes completos da ocorrência, além de poder modificar o seu *status*, indicando se foi encaminhada ao órgão responsável ou se foi resolvida.

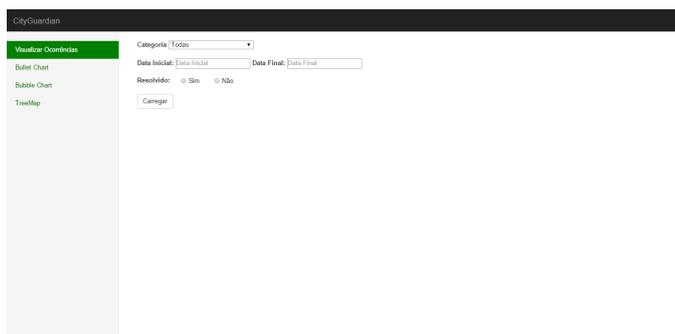


Figura 11: Tela de exibição dos filtros

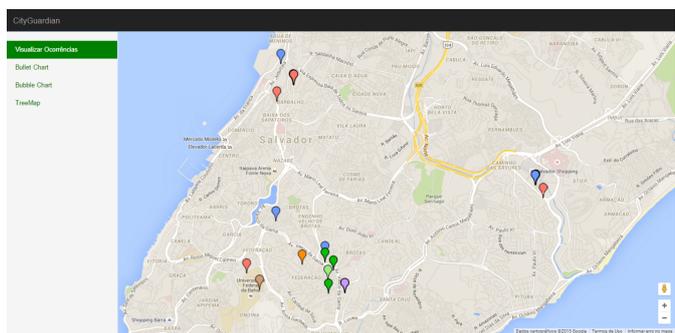


Figura 12: Tela de exibição do mapa

Para facilitar a interpretação dos dados, foram utilizados recursos de visualização de informação. Três visualizações diferentes foram utilizadas para mostrar os dados em perspectivas diferentes. Elas são apresentadas nas subseções seguintes.

### 5.3.1 Bullet Chart

O objetivo desta visualização é permitir uma comparação rápida entre a quantidade de problemas resolvidos e o total de problemas reportados entre dois períodos. O usuário seleciona o mês e ano desejados, e os valores são apresentados para o respectivo período e o mês anterior. Esta visualização utiliza-se de barras para representar os dados. Tem como elementos gráficos a cor e o tamanho das barras. Essas características permitem observar a evolução da quantidade de problemas solucionados entre os períodos, e por cada categoria. A cor cinza representa o mês atual e a cor azul representa o mês anterior. Os tons mais escuros representam a quantidade de problemas solucionados, e os tons mais claros o total de problemas reportados. O resultado é exibido na Figura 13.

### 5.3.2 Bubble Chart

O objetivo desta visualização é permitir uma rápida distinção da quantidade de problemas resolvidos e não resolvidos em cada categoria, tendo como período o mês atual. Esta visualização utiliza-se de círculos para representar os dados como elemento gráfico. A cor e o tamanho dos círculos são

as propriedades gráficas utilizadas. Possui uma precisão menor, mas pode comportar muitos dados em pouco espaço, dispondo os círculos em espiral. Para este sistema, a cor dos círculos representa a categoria das ocorrências. O tamanho refere-se à quantidade de problemas resolvidos e não resolvidos em um determinado mês. O fato de possuir dois círculos para cada categoria, um representando o total de ocorrências solucionadas e outro o total não solucionado, permite a percepção para o usuário da evolução da solução dos problemas no período. O uso de recursos visuais tamanho e da cor facilita a percepção de quais categorias são mais frequentemente reportadas. O resultado é exibido na Figura 14.

### 5.3.3 Treemap

O objetivo desta visualização é permitir uma rápida distinção da quantidade de problemas apoiados que ainda não foram resolvidos em cada categoria, ao longo do mês. As propriedades gráficas representadas por esta visualização são as cores e os tamanhos dos retângulos. A hierarquia entre categorias e seus respectivos problemas representa a divisão dos retângulos, enquanto as cores representam as diferentes categorias, e no segundo nível da hierarquia, representam os diferentes problemas. Uma outra vantagem desta visualização é a possibilidade de interação do usuário, que pode utilizar o *zoom* para visualizar mais detalhes, e os *tooltips* para visualizar o texto da descrição do problema nos retângulos menores. O resultado é exibido nas Figuras 15 e 16.

## 6. AVALIAÇÃO DA SOLUÇÃO

Esta seção apresenta o estudo conduzido que teve como objetivo: i) realizar testes em ambiente real com usuários reais; ii) coletar *feedback* desses usuários em relação a aplicação *mobile*; e iii) avaliar, do ponto de vista do pesquisador, a solução *web* considerando os dados reais produzidos.

### 6.1 Projeto do estudo

Os participantes foram convidados a participar do estudo por conveniência, sendo que 6 efetivamente utilizaram a ferramenta. A idade varia entre 18 e 38 anos. Destes, 3 são estudantes/atuentes da área de Computação e 3 são estudantes/trabalhadores de outras áreas. Todos os participantes são do sexo masculino.

O período para utilizar a ferramenta foi de 07/10/2015 a 13/10/2015. Os participantes testaram suas funcionalidades e reportaram ocorrências reais em seus respectivos trajetos. Após o período estipulado, foi disponibilizado um questionário para que os participantes avaliassem a ferramenta *mobile*, conforme pode ser visto no apêndice C.

### 6.2 Dispositivos Utilizados

A Tabela 2 lista todos os dispositivos utilizados durante os testes, bem como a versão do Sistema Operacional presente nos mesmos. Todos são de propriedade dos respectivos participantes. Dois dos participantes testaram a aplicação com *tablet* e *smartphone*.

### 6.3 Análise dos dados reais

Foram reportadas 14 ocorrências em diversos locais de Salvador, a exemplo do Imbuí, Boca do Rio, Comércio, Pituba,

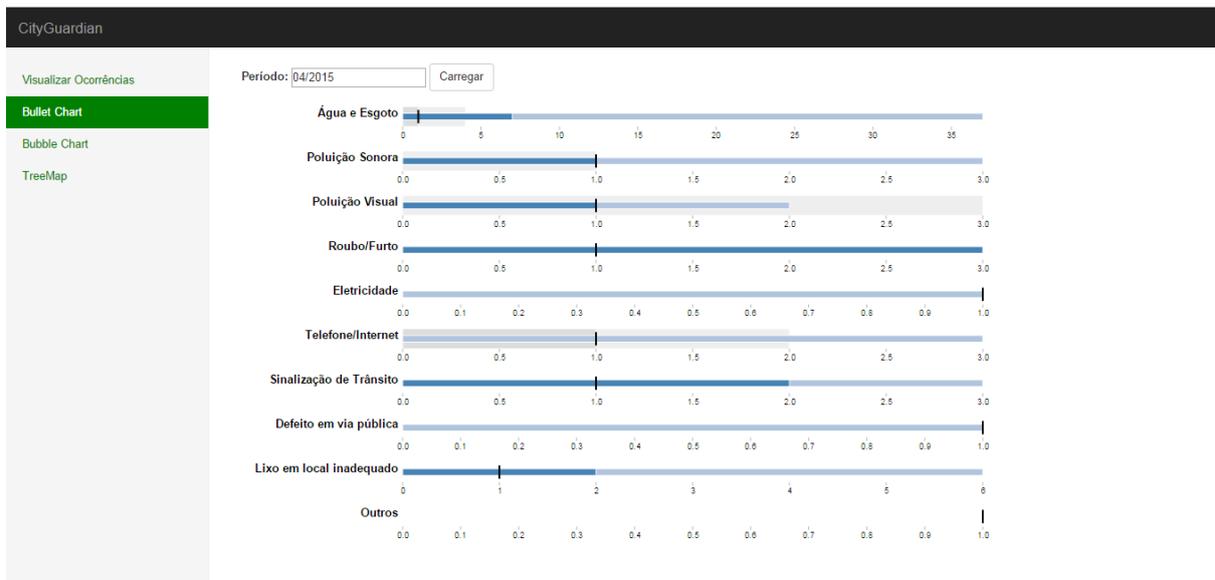


Figura 13: Visualização Bullet Chart

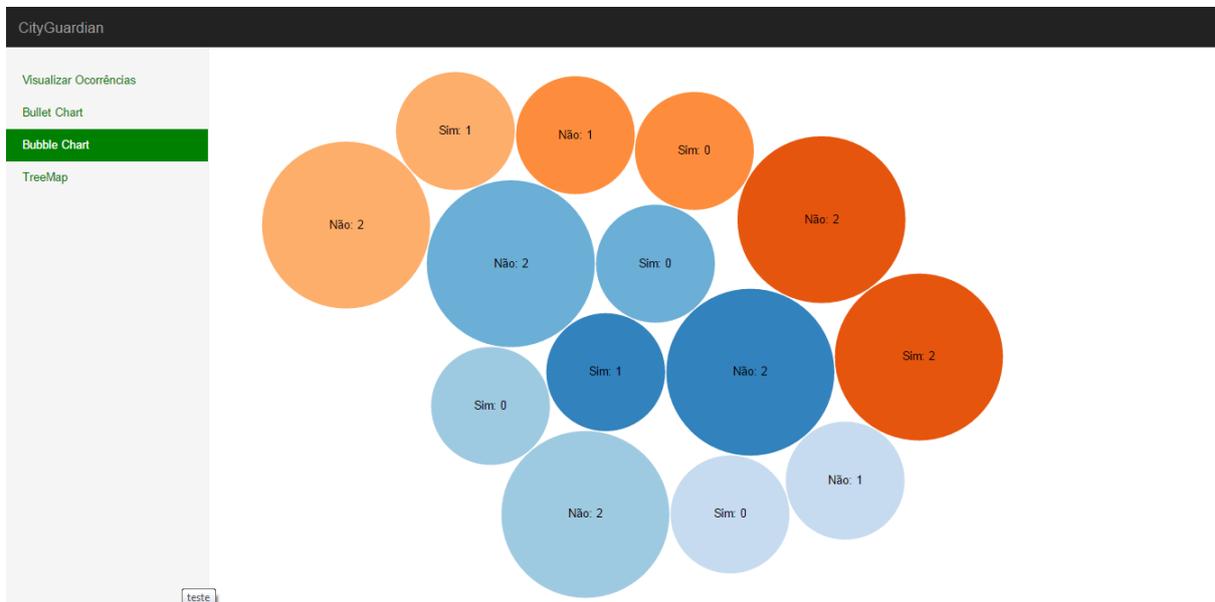


Figura 14: Visualização Bubble Chart

Tabela 2: Modelos e Versões do *Android* dos dispositivos utilizados.

Dispositivo	Tipo	Versão do <i>Android</i>
LG L Prime	<i>Smartphone</i>	5.0.2 ( <i>Lollipop</i> )
Motorola Moto G	<i>Smartphone</i>	5.1 ( <i>Lollipop</i> )
Motorola Moto X	<i>Smartphone</i>	5.1 ( <i>Lollipop</i> )
Samsung Galaxy Grand Neo DUOS	<i>Smartphone</i>	4.2.2 ( <i>Jelly Bean</i> )
Zenfone 5	<i>Smartphone</i>	4.4.2 ( <i>KitKat</i> )
Samsung Galaxy Note	<i>Tablet</i>	4.1.2 ( <i>Jelly Bean</i> )
Samsung Galaxy Tab S	<i>Tablet</i>	5.0.2 ( <i>Lollipop</i> )

Caminho das Árvores, Vasco da Gama e Federação. As categorias selecionadas foram: Água e Esgoto, Defeito em via

pública, Lixo em local inadequado, Poluição Sonora, Outros;



Figura 15: Visualização Treemap

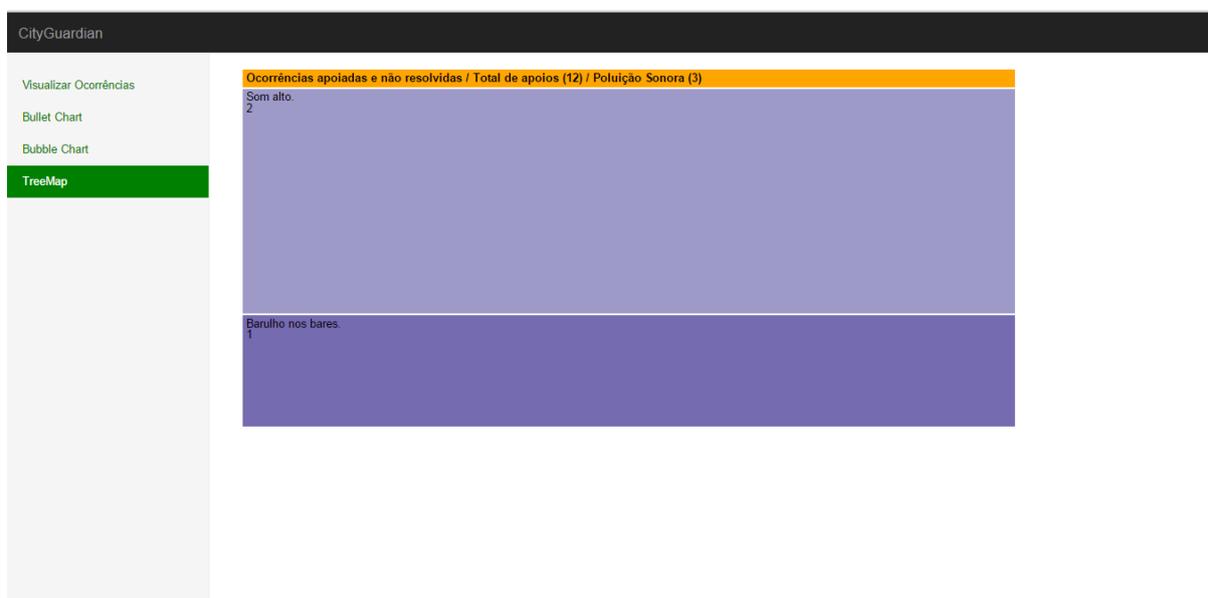


Figura 16: Visualização Treemap após zoom na categoria Poluição Sonora

#### 6.4 Análise do feedback dos usuários

O questionário respondido pelos usuários produziu os resultados seguintes:

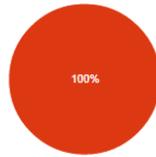
Nenhum dos usuários havia usado anteriormente algum aplicativo semelhante, e tiveram facilidade de utilizar a aplicação testada, conforme as Figuras 17 e 18. A análise dos dados permite observar que a visualização das ocorrências foi a funcionalidade que melhor cumpriu o objetivo, obtendo as notas mais positivas. Em seguida, pode-se observar que os problemas foram reportados com sucesso pela maioria dos usuários, conforme pode ser visto nas Figuras 20 e 21. Os pontos negativos foram observados nos seguintes aspectos: A Figura 19 mostra que a interface visual da aplicação teve

avaliações distintas, onde alguns pontos de melhoria foram sugeridos. Dois dos seis participantes relataram haver problemas com a função de anexar imagens através da galeria do dispositivo. Houveram relatos de algumas discrepâncias na visualização da rota recente. Esses itens podem ser observados nas Figuras 22 e 23. Por fim, foram feitas sugestões de melhoria na interface gráfica, de forma a melhorar o entendimento e a experiência na utilização da aplicação (Figura 24).

#### 6.5 Análise da solução web

A solução *web* foi analisada através da realização do fluxo dos casos de uso com os dados reportados pelos participantes. Na Figura 25, pode-se observar o mapa com as ocor-

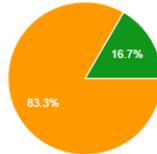
Você já utilizou anteriormente algum aplicativo semelhante?



Sim	0	0%
Não	6	100%

Figura 17: Questão 1

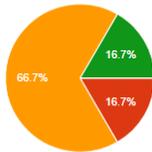
Qual o grau de facilidade para utilizar a aplicação?



Péssimo	0	0%
Ruim	0	0%
Bom	5	83.3%
Ótimo	1	16.7%

Figura 18: Questão 2

Qual o grau de avaliação da interface visual da aplicação?



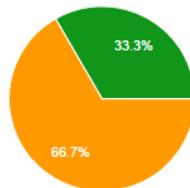
Péssimo	0	0%
Ruim	1	16.7%
Bom	4	66.7%
Ótimo	1	16.7%

Comentários:

O menu, que fica no canto esquerdo superior, as vezes pode passar despercebido. Acredito que seria melhor haver uma barra na parte superior da tela, com o nome da tela, o menu e o botão de centralização do mapa, de preferência com a barra em uma cor que não confunda com o mapa. Botão de "Apoio" pouco intuitivo.

Figura 19: Questão 3

De que forma a ferramenta cumpriu a função de reportar ocorrências na cidade?



Péssima	0	0%
Ruim	0	0%
Boa	4	66.7%
Ótima	2	33.3%

Figura 20: Questão 4

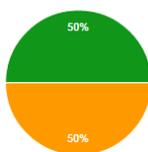
rências reportadas, filtradas por *status*, exibindo todos os problemas ainda não resolvidos.

A Figura 26 mostra o resultado da tela de detalhes para uma ocorrência. Em seguida, foram consultadas e marcadas

como resolvidas quatro ocorrências, sendo uma da categoria Água e Esgoto, duas da Categoria Defeito em via pública, e uma da categoria Outros.

Por fim, foram observados os dados gerados pelas visualiza-

Como você avalia a visualização das ocorrências reportadas?



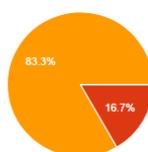
Péssima	0	0%
Ruim	0	0%
Boa	3	50%
Ótima	3	50%

Comentários:

Cumpre sua função, exibindo o descritivo associado ao report, gênero da ocorrência e possibilita a adição de imagens para comprovar o problema.

Figura 21: Questão 5

Como você avalia a visualização da rota recente?



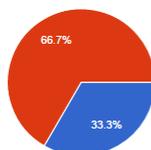
Péssima	0	0%
Ruim	1	16.7%
Boa	5	83.3%
Ótima	0	0%

Comentários:

No geral a rota descreve o caminho percorrido, porém, em alguns casos, em alguns poucos trechos aparecem trajetos que não foram percorridos, na rota à pé.

Figura 22: Questão 6

Você teve problemas para usar a ferramenta?



Sim	2	33.3%
Não	4	66.7%

Em caso afirmativo, relate quais foram:

Problemas ao identificar a rota, e dificuldades ao carregar imagens como anexo.

Não foi possível anexar uma imagem da galeria do meu aparelho ao meu report.

Figura 23: Questão 7

ções. A visualização *Bubble Chart* exibiu os dados conforme esperado, com as ocorrências reportadas no mês, e permitindo apurar em cada uma delas quantos problemas estão ou não resolvidos. A comparação entre os momentos antes e após as ocorrências serem marcadas como resolvidas pode ser vista na Figura 27.

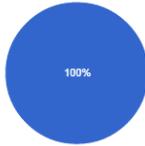
A Figura 28 mostra o resultado da visualização *Bullet Chart*. Os dados são exibidos conforme as ocorrências reportadas, mostrando acima as ocorrências por categoria antes de serem marcadas como resolvidas, e abaixo, após serem marcadas.

No entanto, não foi possível observar a comparação entre períodos, visto que todas as ocorrências passíveis de serem visualizadas compreendem o período do mês de outubro de 2015.

Os resultados da visualização *TreeMap* podem ser vistos na Figura 29. Acima, pode-se observar o *TreeMap* antes de se realizar o fluxo dos casos de uso, e abaixo, o resultado após as quatro ocorrências terem sido marcadas como resolvidas.

Assim, apesar da pouca quantidade de dados, pode-se di-

Na sua opinião, a ferramenta foi útil?



Sim	6	100%
Não	0	0%

Sugestões de melhorias e/ou críticas:

Adicionar report ao pressionar em um ponto do mapa. Criar uma barra superior para que o menu ganhe mais destaque. Utilizar um sidebar menu. Login sem ser pelo endereço de e-mail. Salvar a senha e o login na tela inicial. Criar um botão de informações para a tela/função que o usuário está acessando.

Figura 24: Questão 8

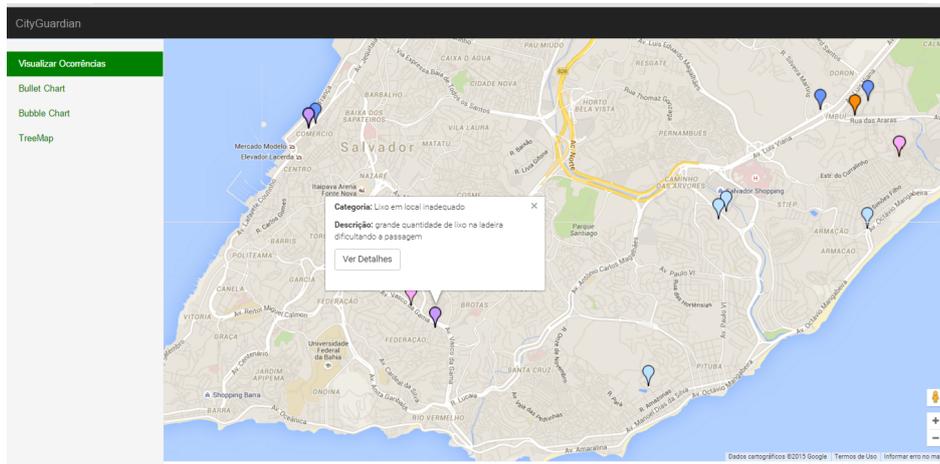


Figura 25: Problemas reportados pelos participantes do teste.

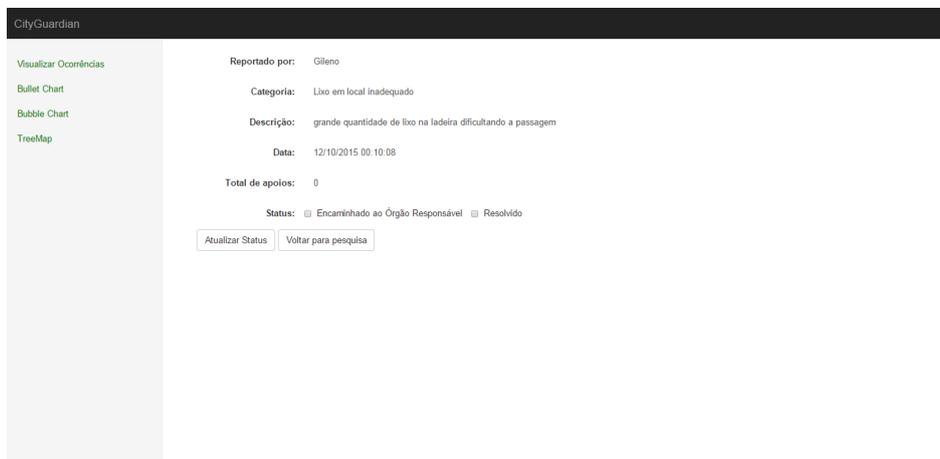


Figura 26: Detalhes da ocorrência selecionada.

zer que a ferramenta *web* cumpre seu objetivo. Entretanto, para melhorar a sua avaliação, poderia-se estender o tempo de uso da ferramenta em no mínimo dois meses, e com mais usuários, a fim de se obter mais dados através da aplicação móvel. Isso poderia trazer melhores resultados para as vi-

sualizações, em especial com o *Bullet Chart*. Além disso, outro método mais apurado para avaliação poderia ser utilizado além do teste de usabilidade.

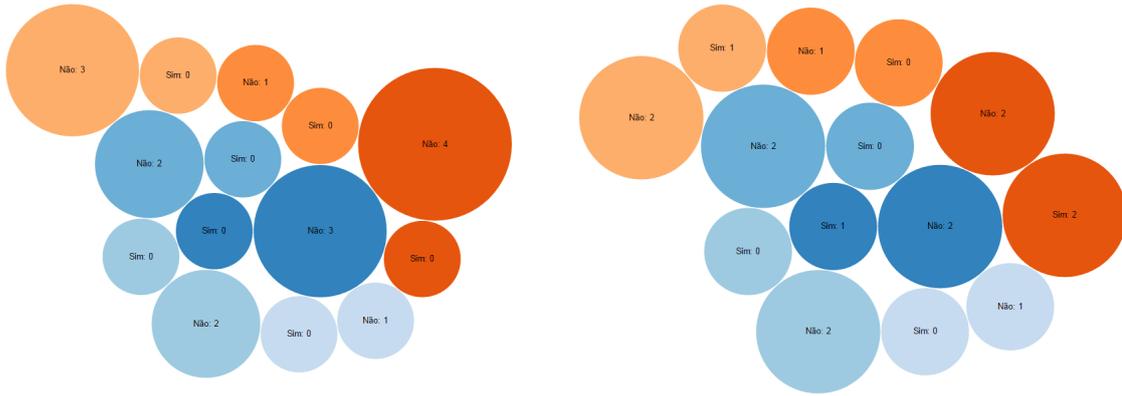


Figura 27: Comparação dos resultados do Bubble Chart.

## 7. CONCLUSÃO

O presente trabalho apresentou o sistema *CityGuardian*, para gerenciamento de problemas em cidades. A partir dos conceitos de desenvolvimento para dispositivos móveis e uso de *web services* com arquitetura REST, observando as principais etapas na análise e projeto do sistema. Também foram descritas as funcionalidades desenvolvidas para cada um dos módulos. A avaliação realizada com a ferramenta móvel permite constatar que a mesma cumpriu o seu objetivo, sendo apontada como útil e tendo boa avaliação nas suas duas funcionalidades mais importantes: a de reportar ocorrências e visualizar os dados reportados. Entretanto, observa-se a necessidade de investigar e avaliar algumas falhas apontadas no processo de envio de imagens da galeria e na visualização da rota. Também é preciso avaliar a pertinência das sugestões de melhoria da interface realizadas. A ferramenta *web* também cumpriu seu objetivo, exibindo corretamente os dados reportados pelos usuários da aplicação *mobile*. Houveram limitações em uma das visualizações por conta da quantidade de dados referente ao período de testes. Uma metodologia mais apurada deve ser utilizada.

Como trabalhos futuros, pode-se avaliar a possibilidade de realizar a portabilidade do sistema para outra plataforma móvel. Isto aumentaria a possibilidade de agregar mais usuários ao sistema, bem como mais registros de problemas podem ser realizados e sanados. Além disso, algumas melhorias podem ser realizadas. Um exemplo seria a avaliação da segurança dos dados, com a implementação de perfis de acesso à cada funcionalidade da ferramenta *web*. Além de um meio de comunicação escrita entre as aplicações. Outro ponto a ser analisado é a escalabilidade da aplicação. Além da arquitetura já apresentada, outros quesitos podem ser estudados de forma mais aprofundada, como balanceamento de carga em servidores e análise das conexões com o banco de dados.

## 8. REFERÊNCIAS

- [1] Colab. Disponível em: <http://colab.re/> Acesso em: 14/08/2014
- [2] Cidadera - Reclame sua cidade. Disponível em: <http://cidadera.com/?locale=pt-BR> Acesso em: 14/08/2014
- [3] Approximately 200 municipal governments are using the Web-based communications system to automate interactions between citizens and city agencies. Disponível em: <http://www.smartplanet.com/blog/the-report/talk-to-your-city-with-publicstuff-it-will-talk-back/> Acesso em: 20/08/2014
- [4] Open Handset Alliance. Disponível em: [www.openhandsetalliance.com/](http://www.openhandsetalliance.com/) Acesso em: 08/01/2014
- [5] The Android Source Code. Disponível em: <https://source.android.com/source/index.html> Acesso em: 30/10/2014
- [6] Lecheta, Ricardo R. 2010. Google Android: aprenda a criar aplicações para dispositivos móveis com o Android SDK. São Paulo: Novatec Editora.
- [7] ART and Dalvik. Disponível em: <http://source.android.com/devices/tech/dalvik/> Acesso em: 15/01/2015
- [8] Deitel, Paul et al. 2013. Android para programadores: uma abordagem baseada em aplicativos. Porto Alegre: Bookman.
- [9] Bezerra, Eduardo. 2007. Princípios de análise e projeto de sistemas com UML. Rio de Janeiro: Campus.
- [10] Vigilante. Disponível em: <http://vigilanteapp.com.br/inicio/> Acesso em: 31/10/2014
- [11] Erl, Thomas. 2005. Service-Oriented Architecture: Concepts, Technology, and Design. Upper Saddle River, NJ: Prentice Hall.
- [12] Josuttis, Nicolai M. 2007. SOA in Practice: The Art of Distributed System Design. O'Reilly Media, Inc.
- [13] IBGE - Cidades@. Disponível em: <http://www.cidades.ibge.gov.br/xtras/perfil.php?lang=&codmun=292740> Acesso em: 25/01/2015

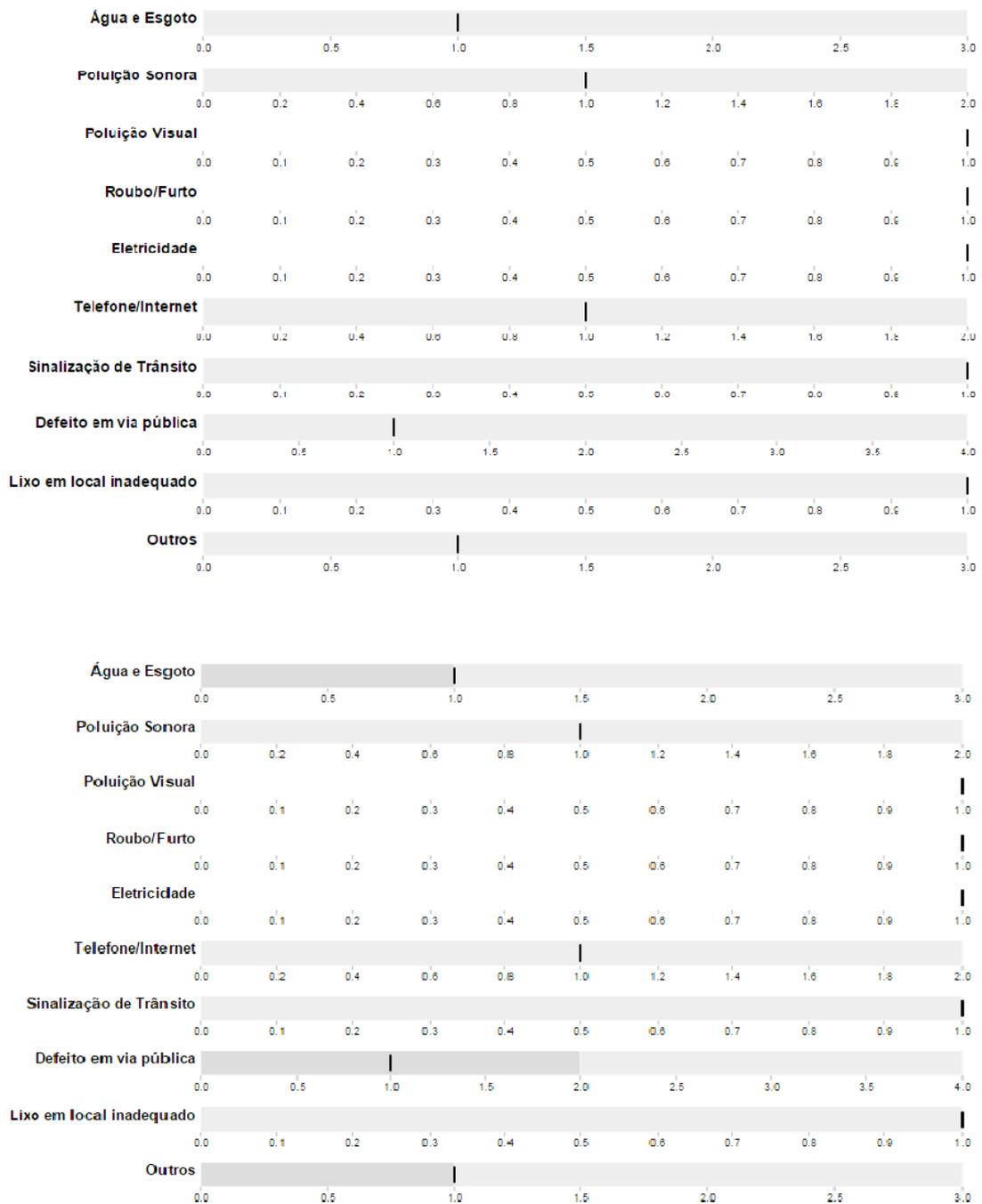


Figura 28: Comparação dos resultados do Bullet Chart.

[14] Fala Salvador. Disponível em:  
<http://www.ouvidoria.salvador.ba.gov.br/index.php/atendimentos/fala-salvador> Acesso em: 21/01/2015

[15] Ouvindo Nosso Bairro. Disponível em:  
<http://www.ouvindonossobairro.salvador.ba.gov.br/>  
 Acesso em: 21/01/2015

[16] Taylor, R. N., Medvidovic, N. and Dashofy, E. M. 2009. Software Architecture: Foundations, Theory, and

Practice. Wiley Publishing.

[17] Bohara, M.H. ; DA-IICT, Gandhinagar, India ; Mishra, M. ; Chaudhary, S. RESTful Web Service Integration Using Android Platform.

[18] Richardson, Leonard and Ruby, Sam. 2007. RESTful Web Services. O'Reilly.

[19] Mazza, Riccardo. 2009. Introduction to Information Visualization (1 ed.). Springer Publishing Company,



Figura 29: Comparação dos resultados do Treemap.

Incorporated.

[20] Jersey. Disponível em: <https://jersey.java.net/> Acesso em: 01/06/2015

[21] Google Maps Directions API. Disponível em: <https://developers.google.com/maps/documentation/directions/> Acesso em: 01/03/2015

[22] CodeIgniter. Disponível em: <http://www.codeigniter.com/> Acesso em: 23/07/2015

[23] D3js. Disponível em: <http://www.d3js.org/> Acesso em: 23/07/2015

[24] JSON. Disponível em: <http://www.json.org/> Acesso em: 30/09/2015

[25] Android. Disponível em: <https://www.android.com/> Acesso em: 01/10/2015

[26] Mateus, Geraldo R.; Loureiro, Antonio A. F. Introdução à Computação Móvel. 1998. 2013.

[27] Introdução e Motivação: Arquitetura em n camadas. Disponível em: <http://www.dsc.ufcg.edu.br/jacques/-cursos/j2ee/html/intro/intro.htm> Acesso em:

03/10/2015

[28] Extensible Markup Language (XML). Disponível em: <http://www.w3.org/XML/>. Acesso em: 10/11/2015

[29] HTTP - Hypertext Transfer Protocol. Disponível em: <http://www.w3.org/Protocols/>. Acesso em: 10/11/2015

[30] Architectural Styles and the Design of Network-based Software Architectures. Disponível em: <https://www.ics.uci.edu/fielding/pubs/dissertation/-top.htm>. Acesso em: 10/11/2015

## APÊNDICE

### A. CASOS DE USO

Um caso de uso deve definir o uso de uma parte da funcionalidade de um sistema, sem revelar a estrutura e o comportamento internos. É a especificação de uma sequência de interações entre o sistema e os agentes externos que o utilizam, chamados de atores [9]. As Figuras 30 e 31 mostram, respectivamente, os diagramas dos principais casos de uso

dos módulos *mobile* e servidor.

## A.1 Módulo Móvel

A seguir, a descrição enumerada referente ao fluxo básico dos casos de uso do módulo móvel:

### Realiza Cadastro

- 1- O ator informa o seu nome, email e sua senha.
- 2- O ator submete as informações ao sistema.
- 3- O sistema valida os dados e confirma se o cadastro foi ou não realizado com sucesso.

### Realiza Login

- 1- O ator informa o seu email e sua senha.
- 2- O ator submete as informações ao sistema.
- 3- O sistema valida os dados e confirma se o login foi ou não realizado com sucesso.

### Reporta Problema

- 1- O ator seleciona no mapa a região onde o problema será reportado.
- 2- O ator seleciona a categoria a qual o problema pertence.
- 3- O ator insere uma descrição sobre o problema.
- 4- O ator pode ou não inserir anexos que auxiliem na descrição.
- 5- O ator submete as informações ao sistema.

### Consulta problemas reportados

- 1- O ator consulta os problemas reportados no mapa.
- 2- O ator seleciona o problema do seu interesse.
- 3- O sistema exibe as informações detalhadas acerca do problema.

### Apoia problema

- 1- O ator busca os problemas reportados no mapa.
- 2- O ator seleciona o problema do seu interesse.
- 3- O sistema exibe as informações detalhadas acerca do problema.
- 4- O ator confirma a veracidade da informação e mostra apoio a ela.

### Visualiza Rota Recente

- 1- O ator informa se a rota foi realizada a pé ou de carro.
- 2- o sistema cacula a rota realizada na última hora e exibe na tela.

## A.2 Módulo Web

A seguir, a descrição enumerada referente ao fluxo básico dos casos de uso do módulo *web*:

### Verifica Problema

- 1- O ator, utilizando ou não os filtros de informação disponíveis, consulta os problemas reportados no mapa.
- 2- O ator seleciona o problema do seu interesse.
- 3- O sistema exibe as informações detalhadas acerca do problema.

### Marca Problema como resolvido

- 1- O ator busca os problemas reportados no mapa.
- 2- O ator seleciona o problema do seu interesse.
- 3- O sistema exibe as informações detalhadas acerca do problema
- 4- O ator confirma a resolução do problema e o marca como resolvido.

## B. MODELO DE DADOS

A tabela do SQLite utilizada para salvar as coordenadas no dispositivo do usuário está exibida na Figura 32. A Figura 33 mostra a representação do modelo lógico do banco de dados usado para o sistema, com suas entidades e relacionamentos.

## C. QUESTIONÁRIO DE AVALIAÇÃO DA APLICAÇÃO MÓVEL

Na sequência, o formulário do questionário disponibilizado aos participantes e utilizado para avaliar os resultados dos testes.

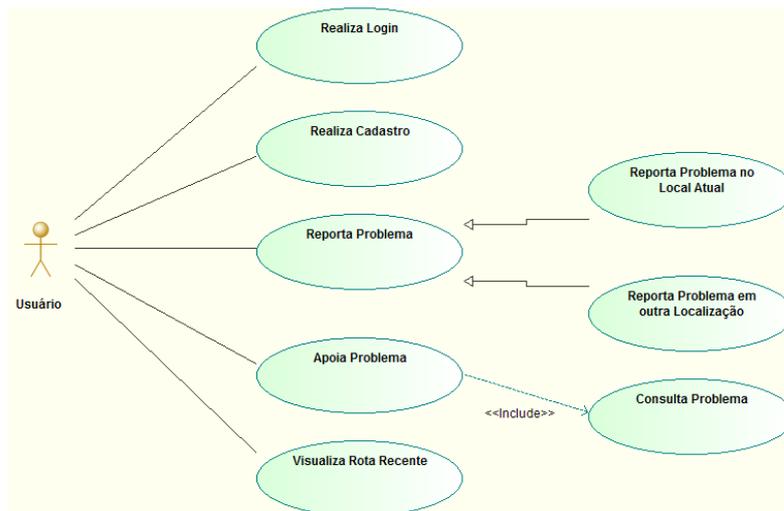


Figura 30: Diagrama de Casos de Uso do Módulo *Mobile*

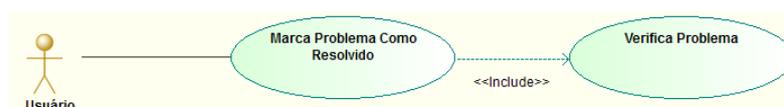


Figura 31: Diagrama de Casos de Uso do Módulo *Servidor*

Usuario_Coordenadas	
id:	INTEGER
id_usuario:	INTEGER
latitude:	REAL
longitude:	REAL

Figura 32: Modelo de dados do SQLite

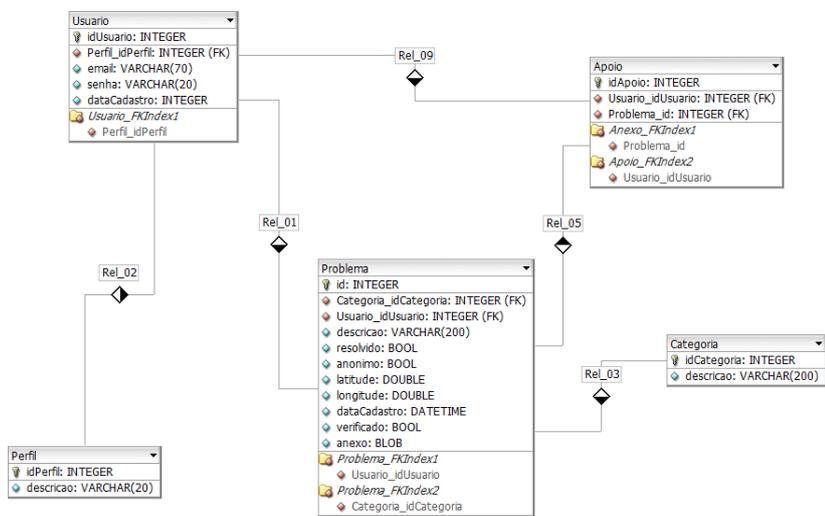


Figura 33: Modelo de dados do sistema