

Aplicação Móvel na Infraestrutura Municipal

Fabricio Venturini Gonçalves¹, Fernando Sarturi Prass¹

¹Centro Universitário Franciscano
Caixa Postal 97010-032 – Santa Maria – RS – Brasil

venturini.fabricio@hotmail.com, fernando.prass@unifra.br

Abstract. *This paper presents the proposal of an application developed to manage requests for lightning and street repairs, and also city cleaning. The main objective is to provide society the possibility of making requests and complaints through an application that bridges the gap between citizens and public agencies. The methodology adopted for this study was Feature Driven Development (FDD). The designed application was developed for Android platform and data loading occurs online.*

Resumo. *Este artigo apresenta a proposta de um aplicativo desenvolvido para gerenciar a solicitação de pedidos de conserto de iluminação, ruas e também de limpeza das cidades. Tem como objetivo principal proporcionar à sociedade a possibilidade de realizar pedidos e reclamações, por meio da aplicação, de maneira a criar uma ponte entre os cidadãos e os órgãos públicos. A metodologia adotada para a elaboração do trabalho foi a Feature Driven Development (FDD). O aplicativo desenvolvido tem a plataforma de desenvolvimento Android e o carregamento de informações ocorre online.*

1. Introdução

De acordo com [Teixeira e Júnior 2015], a internet tem possibilitado grandes evoluções, entre as quais grande parte envolve o cidadão. Com o desenvolvimento da tecnologia móvel, foi aberto um espaço para aplicações que vêm avançando em um ritmo acelerado, apresentando ferramentas que facilitam o cotidiano das pessoas nos ambientes de trabalho e de lazer localizados no meio urbano.

Diante de alguns imprevistos que, por vezes, ocorrem nas cidades – como o descontrole da sua infraestrutura, estradas deterioradas, iluminação precária e falta de remoção de entulho –, faz-se cada vez mais necessário um aplicativo que possibilite controlar o processo de degradação dos centros urbanos, de forma que o cidadão possa contribuir para a fiscalização da gestão pública. Os meios tecnológicos devem funcionar como uma ferramenta da administração pública e como um meio flexível de a sociedade interagir com a gestão dos municípios.

O problema identificado, no decorrer desta pesquisa, diz respeito ao fato de a gestão pública investir dinheiro e tempo em fiscalização privada, a qual, por vezes, não consegue ser eficiente, deixando a desejar na opinião da população. Sendo assim, a aplicação proposta traz a possibilidade de agilizar os procedimentos e a fiscalização nas ruas, modernizando a administração, economizando verba pública e aproximando mais a comunidade da administração municipal, uma vez que, com a sua utilização, o cidadão passará a fiscalizar sua própria cidade.

1.1. Objetivo geral

O objetivo deste trabalho é desenvolver um aplicativo que auxilie nos processos de fiscalização da infraestrutura de obras e gestão pública, facilitando o acesso da comunidade aos serviços públicos e criando um canal de comunicação entre prefeitura e cidadão, de modo a permitir a este a solicitação de serviços de limpeza, consertos de ruas e iluminação pública.

1.2. Objetivos específicos

Os objetivos específicos que guiaram a elaboração deste projeto foram:

- Estudar trabalhos correlatos e pesquisar artigos sobre a gestão pública com a finalidade de aprofundar o conhecimento necessário para a preparação do projeto proposto;
- Seguir uma metodologia *Feature Driven Development* (FDD) no projeto da aplicação, concluindo-a com objetivo positivo;
- Implementar um aplicativo usando a API do Google Maps, *PUSH* de notificação, recursos do GPS e marcação de pinos no mapa.
- Estudar as funcionalidades, seguindo os requisitos elaborados e aplicar os testes de validação no aplicativo final.

2. Referencial teórico

Esta seção compreende o referencial teórico pesquisado para fundamentar a proposta a partir de pesquisas e doutrinas relacionadas.

2.1. A ineficácia do poder público e sua exacerbada burocracia

Segundo [Aragão 1997], a burocracia aparece como um dos principais problemas atuais na gestão pública. Juntamente com o déficit de servidores, ela é responsável pelas paralisações de procedimentos e pela impaciência do cidadão, que se torna a cada dia mais exigente.

A burocracia, apesar de ter por objetivo regularizar todos os atos de fiscalização de concertos e compras, acaba, muitas vezes, sendo mais rígida do que necessário. Assim, a sociedade acaba por encontrar carência e dificuldade nos atendimentos de suas demandas, o que caracteriza a ineficácia do poder público.

Nesse contexto, torna-se de grande valia o uso da tecnologia da informação, das aplicações móveis e de diversos outros elementos que possam facilitar o andamento de processos, principalmente os referentes ao setor licitatório das prefeituras.

Segundo o Censo [IBGE 2010], a cidade de Formigueiro/RS possui 7.014

habitantes. A Prefeitura Municipal realizou uma pesquisa, por meio da Secretaria de Obras, na qual foram coletados dados de pedidos de conserto de iluminação pública. Com base nesses dados, foi possível observar o número de pedidos no mês, sua localização (meio urbano ou rural) e se foi efetuado o conserto ou não. Na **Figura 1** a seguir, mostra-se o relatório do ano de 2016, que abrange desde o início de janeiro até o início de abril.

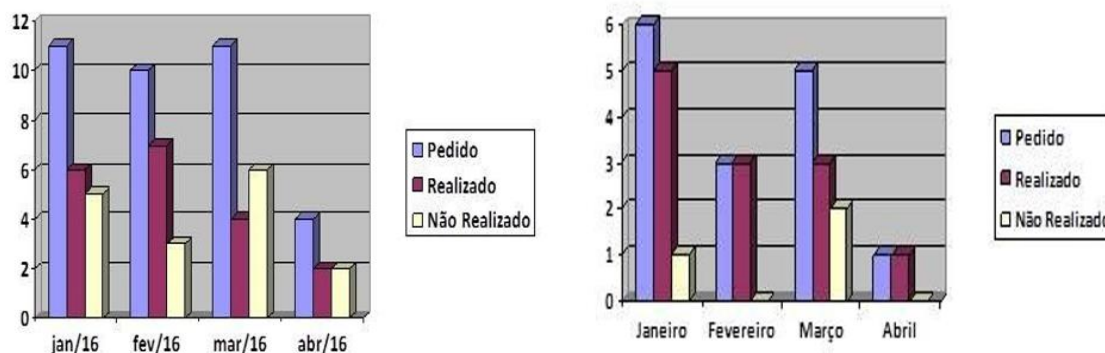


Figura 1. Dados da pesquisa realizada em Formigueiro 2016

Fonte: Prefeitura Municipal de Formigueiro, 2016

Os gráficos apresentam pedidos que foram realizados pelos cidadãos que moram no interior do município e pelos que residem no centro da cidade. Tal pesquisa teve como intuito mostrar, de forma genérica, como a população solicita pedidos de conserto, estes muitas vezes esquecidos ou não encaminhados para o servidor responsável pela manutenção.

2.2. Aplicação móvel

A aplicação móvel é um *software* desenvolvido para ser instalado em um dispositivo móvel, como *smartphones* e *tablets*. Devido ao aumento da procura por esses produtos pelo consumidor, esta tecnologia está em constante atualização e tem como propósito facilitar o dia a dia do usuário, fornecendo-lhe as mais diversas funcionalidades.

A inserção das redes na rotina da população leva a aplicação móvel ao topo dos itens mais buscados por órgãos públicos e privados com o objetivo de obter um *feedback* dos serviços prestados.

A computação móvel pode ser considerada uma evolução computacional, pois permite que o usuário, estando em qualquer lugar, tenha acesso à informação em tempo real [Maurício e Figueiredo 2016]. Além disso, foi desenvolvida uma grande variedade de *apps* (abreviatura de aplicativos) com a finalidade de facilitar a vida de todos que, de alguma forma, utilizam-nos em seu cotidiano.

Os sistemas de gestão interna dos governos enfrentam desafios como o déficit de servidores, o acréscimo de reclamações por parte do cidadão, o regime governamental, as modificações decorrentes das leis e a necessidade de transmitir informação à população. Nesse contexto, a aplicação móvel mostra-se ideal para a implantação de recursos novos, a serem utilizados em ambientes externos, sanando a lacuna enfrentada pelos servidores.

2.3 Google Maps

O *Google Maps* é uma tecnologia utilizada para serviços de pesquisa e visualização de mapas e de imagens de satélite da Terra. Ele possui diversas funcionalidades, como navegação, marcação de determinado ponto no mapa e a conversão dos endereços em coordenadas geográficas.

De acordo com [Reis et al. 2006], o *Google Maps* tem recursos que podem ser alterados pelo usuário – balões que contêm informações e que são marcados no mapa, trazendo, além de endereço e coordenadas geográficas, até mesmo fotos do local exato.

A integração do *Google Maps* com aplicativos móveis, de acordo com [Ayres 2009], é realizada por uma conexão *Google Static Maps API*. Ela ocorre a partir da solicitação, via HTTP, de uma imagem estática do mapa que represente determinada localidade.

Dessa forma, é possível criar uma aplicação capaz de usar o *Google Maps*, mostrando mapas de todas as cidades e permitindo navegar de forma interativa pela tela do *smartphone*.

2.4. GPS

O *Global Position System* (GPS) é um sistema de posicionamento global que funciona via satélite, permitindo que o receptor móvel indique uma posição. Desse modo, qualquer pessoa ao redor do mundo pode identificar sua localização com precisão exata. Conforme [Galo e Camargo 1994], um dos grandes benefícios do GPS é poder ser utilizado como ponto de referência de coordenadas exatas. Para que o sinal aumente e o sistema funcione, é necessário estar na visão de três ou mais satélites.

2.5. Web Services

Web Services é um padrão de comunicação utilizado para múltiplos fins, que integra diferentes sistemas e plataformas, promovendo serviços por meio da internet e do tráfego de dados.

De acordo com [Freitas 2002], a arquitetura do *Web Service* é capaz de trocar mensagens com diferentes plataformas, e sua implantação é baseada em um conjunto de protocolos e linguagens web.

2.6. Android

Android é o sistema operacional *open source* desenvolvido pela Google. Além disso, é uma plataforma complexa de desenvolvimento de aplicativos baseada no núcleo do Linux [Lecheta 2010]. Sendo um sistema operacional de fácil manejo, ele pode ser personalizado e utilizado em milhões de dispositivos.

As empresas desenvolvedoras de aplicações buscam cada vez mais uma plataforma para desenvolver produtos modernos para seus usuários finais, que buscam novidades em seus aplicativos e sistemas operacionais.

Android Studio é um Ambiente de Desenvolvimento Integrado (IDE – *Integrated development environment*) criado para facilitar a elaboração de novas aplicações na plataforma Android. Ele é baseado no *IntelliJCommunity Version* (utilizado para desenvolver em Java) e usa a Gradle para compilar as aplicações desenvolvidas. Além

disso, ele fornece vasta opção para empacotamento.

A ferramenta utilizada neste trabalho é a versão 2.1, última versão disponível no site *Android Studio SDK* (<http://developer.android.com/sdk/index.html>).

3. Trabalhos relacionados

Apesar de a pesquisa realizada sobre trabalhos relacionados não ter encontrado artigos que fossem fortemente similares ao projeto aqui elaborado, foram encontrados dois artigos sobre o desenvolvimento de aplicativos que abordam um contexto semelhante ao utilizado na presente proposta.

3.1. Um sistema de identificação e sugestão de rotas de ônibus baseado em informações de redes sociais

O trabalho realizado por [Gomes et al. 2012] teve como objetivo elaborar um sistema que permite ao usuário de transporte urbano informar-se sobre a rota dos ônibus que circulam pelas ruas das grandes cidades. Esse consumidor utiliza as redes sociais para informar-se também sobre o tráfego e sobre situações que tenham ocorrido no perímetro urbano. Os dados de rota mostrados no aplicativo podem ser visualizados, comentados e postados no *Twitter*, funcionando, assim, como provedor de informações transmitidas pela população. Essa ferramenta, desenvolvida na plataforma Android, foi denominada *UbibusRoute*. Ela utiliza o API do *Google Maps* como fonte de geolocalização e captura automaticamente informações dos perfis dos usuários.

O *UbibusRoute* identifica a localização do usuário, que pode marcá-la no mapa ou realizar sugestão de rotas diferentes com menos fluxo de carros. Esse conceito é semelhante ao do projeto realizado, pois nele são apresentadas funcionalidades como a marcação no mapa e a identificação da localização do usuário. Além disso, ao utilizá-lo, o cidadão poderá acessar informações no mapa, interagindo mais com as informações postadas.

3.2 EASYT – Sistema Integrado De Mobilidade Urbana

Segundo [Nunes e Ruaro 2012], o EASYT é uma ferramenta que transforma o cotidiano da população. Ele funciona de maneira que o usuário selecione no mapa o táxi mais próximo do seu ponto de localização e que o taxista receba o pedido de solicitação se estiver disponível. O aplicativo identifica também a localização do ônibus selecionado pelo usuário no mapa, permitindo, assim, o acompanhamento do veículo mais próximo em seu deslocamento por todas as paradas.

O EASYT foi desenvolvido sobre a plataforma Android, utilizando GPS e *Web Service* para gerenciar o controle de usuários, taxistas e empresas de ônibus.

O EASYT auxiliou no processo de elaboração do aplicativo de comunicação entre cidadão e órgãos públicos, pois trouxe um contexto em que o usuário tem seu *login* e pode usufruir de funcionalidades como solicitar uma corrida de táxi e visualizar no mapa determinados pontos de taxistas e o ônibus. A versão da aplicação web voltada para as empresas nela cadastradas tem uma função que permite aceitar ou não os pedidos dos usuários, recurso aproveitado nesta pesquisa.

4. Metodologia

A metodologia utilizada neste trabalho é a *Feature Driven Development* (FDD). As

funções e os processos que a FDD traz são necessários para o desenvolvimento do trabalho.

4.1 Feature Driven Development (FDD)

Segundo [Figueredo 2007], a FDD é uma metodologia ágil, utilizada para a elaboração de processos de engenharia. Seu objetivo é que os processos sejam concluídos e efetuados de maneira que o cliente receba o produto funcionando, na versão final, e tenha uma noção completa do desenvolvimento deste.

Conforme [Hepatagon 2015], a FDD tem algumas características específicas, como soluções e resultados a cada duas semanas ou menos, em pequenos blocos de aplicabilidade prezados pelo cliente – o chamado *feature*. Ademais, o planejamento detalhado e o guia para medição com rastreabilidade, além da incrível precisão, fazem com que os monitoramentos sejam detalhados com resumos de alto nível para que o cliente e o gerente do projeto, mesmo dentro dos primeiros 10% da obra, já tenham uma forma de saber se o plano e a estimativa são sólidos. O autor mostra que a metodologia é concreta e dispõe de duas funções: (1) concepção e planejamento: pensar antes de tomar atitudes e (2) construção: fazer de forma comunicativa. [Hepatagon 2015] afirma ainda que, na fase de concepção e de planejamento, encontram-se três processos:

1). Desenvolver um modelo abrangente: é realizada uma análise orientada a objetos envolvendo desenvolvimento de requisito. Seguindo-se esse passo, será obtida uma aplicação de alto nível;

2). Construir uma lista de funcionalidades: assim, são identificadas as funcionalidades requisitadas para realizar o projeto;

3). Planejar por funcionalidade: é o plano de desenvolvimento que planeja a produção e que será executado uma única vez durante o projeto.

Dos processos descritos, foram aproveitados neste trabalho: a pesquisa sobre a estrutura do artigo, assunto principal deste trabalho (disponível na seção 2); a realização de estudo da documentação para o funcionamento do *software* e seu desenvolvimento (disponível na seção 5.2); e um modelo de execução do projeto, que foi desenvolvido a partir de um diagrama de atividades para desenvolver modelo (disponível na seção 5.2.3).

A utilização de uma metodologia ágil fornece resultados à medida que os requisitos são listados. Para o entendimento do sistema proposto, seguiram-se os passos do FDD.

Na fase de construção, existem dois processos:

1). Detalhar por funcionalidade: a equipe detalha os requisitos e artefatos, tendo cada funcionário uma função;

2). Construir por funcionalidades: o processo é executado uma vez para cada funcionalidade.

A utilização da metodologia FDD permite oferecer o planejamento e o modelo do projeto, contribuindo para a obtenção de resultados positivos, como o programado.

4.2 Proposta

A partir da pesquisa realizada sobre trabalhos de desenvolvimento de aplicações para a população visando facilitar sua locomoção e criar rotas para fugir do tráfego pesado de

grandes cidades, percebe-se que, em geral, os usuários estão interagindo mais com aplicações que forneçam resultados imediatos. Nessa lógica, o usuário, de alguma maneira, contribui para mudar o ambiente onde vive.

A elaboração do presente trabalho terá como plataforma os *smartphones* que têm sistema operacional Android. Com o aplicativo, objetiva-se permitir, em tempo real, a solicitação ou a reclamação a partir de algum ponto crítico na cidade. Para tanto, o órgão público terá um *login* de acesso com dados da prefeitura, recebendo as informações dos usuários, cujas solicitações podem ou não serem aceitas. Dessa forma, será gerada uma nova conexão entre prefeitura e cidadão. A aplicação irá transformar o modo como a gestão pública trabalha nos dias de hoje, uma vez que os cidadãos irão informar, por meio do aplicativo, os erros e os pontos críticos de uma cidade.

A interação entre prefeitura e população não ocorrerá em um só município, mas no mundo inteiro. Assim, o cidadão ganha, com a aplicação proposta, poder de fiscalização, e os órgãos públicos recebem as informações marcadas no mapa da cidade, podendo agilizar os processos de infraestrutura e de limpeza em geral.

Conforme a **Figura 2** apresenta os passos da proposta do aplicativo.

1. O usuário se cadastra no serviço;
2. O usuário escolhe a opção (limpeza, buracos ou iluminação pública) e efetua o pedido, podendo tirar uma foto do problema para melhor identificação;
3. É realizada a postagem, que será enviada para a prefeitura;
4. A postagem realizada será marcada no mapa com ícone referente ao pedido (a marcação estará disponível para todos que possuírem o aplicativo);
5. Os dados são enviados para o órgão público responsável;
6. Os dados de solicitação são visualizados por um responsável da prefeitura, que irá analisar o pedido de conserto, podendo aceita-lo ou não;
7. Os dados serão armazenados no gerenciador de banco de dados remoto.



Figura 2. Visão do funcionamento da aplicação

4.2.1 Desenvolvimento

Esta seção apresenta o desenvolvimento do trabalho, detalhando-o até sua resolução.

4.2.2 Processo 1 da aplicação

De forma sintética, serão mostradas as funcionalidades do aplicativo desenvolvido.

1) Funcionalidades do usuário:

- Funcionalidade 1: realizar o cadastro do usuário;
- Funcionalidade 2: exibir a localização;
- Funcionalidade 3: selecionar o pedido ou a reclamação;
- Funcionalidade 4: tirar foto ou relatar a solicitação;
- Funcionalidade 5: visualizar no mapa a solicitação postada.

2) Funcionalidades do órgão público:

4.2.2.1 Funcionalidade 1: realizar cadastro do órgão público;

4.2.2.2 Funcionalidade 2: visualizar solicitações no mapa;

4.2.2.3 Funcionalidade 3: aceitar ou não as solicitações e reclamações;

4.2.2.4 Funcionalidade 4: marcar, no mapa, se foi o pedido foi atendido ou se está em andamento.

4.2.3 Requisitos funcionais e não funcionais

De acordo com [Cysneiros 2001], os requisitos funcionais e não funcionais do projeto são fundamentais para a elaboração do processo de *software*. No projeto, foram seguidas as fases do FDD: construir uma lista de funcionalidades e listar os requisitos funcionais e não funcionais, que trazem detalhes de como o sistema vai ser organizado para implementação.

4.2.4 Requisitos funcionais

Os requisitos funcionais descrevem as funcionalidades que o sistema deve fornecer:

- 1) Gerenciar cadastro de usuário: o sistema deve permitir o cadastro novos usuários (nome, e-mail, telefone e senha);
- 2) Gerenciar cadastro da prefeitura: o sistema deve permitir o cadastro de prefeitura (razão social, CNPJ, endereço completo, e-mail e senha);
- 3) Gerenciar solicitação usuário: o sistema deve fornecer para o usuário o cadastro de uma nova solicitação, sendo obrigatória a descrição do pedido (limpeza, buraco e iluminação) e opcional o envio de uma foto que mostre a situação;
- 4) Gerenciar mapa usuário: o sistema deve permitir que o usuário e todos que usufruem da ferramenta possam visualizar as solicitações no mapa;
- 5) Gerenciar mapa prefeitura: o sistema deve permitir que o usuário do órgão público realize a fiscalização dos pedidos solicitados tanto no mapa como na caixa de entrada de solicitações;
- 6) Gerenciar solicitação prefeitura: o sistema deve permitir que o usuário da prefeitura possa alterar o *status* do pedido (não atendido, rejeitado, em andamento ou atendido);

e o sistema pedirá confirmação. Se o usuário optar por “sim”, a solicitação será realizada e armazenada no sistema e enviada para o gerenciador de banco de dados; caso contrário, ele será redirecionado de volta para a tela de cadastro de solicitação (menu).

O órgão público, ao acessar o aplicativo, visualizará os dados recebidos e executará uma fiscalização para aceitar ou rejeitar a solicitação. Se for aceita, a realização do conserto será efetuada, o sistema receberá uma atualização de que o conserto está em andamento e, ao final dela, será enviado para o usuário um aviso de que o conserto foi concluído. Se for rejeitada, o pedido será arquivado no sistema e o usuário receberá um aviso de que a solicitação foi negada pelos seguintes motivos: os pedidos não foram localizados ou os pedidos não estão aos serviços da prefeitura e serão excluídos. Por fim, o aplicativo encerra.

4.2.7 Descritivos de caso de uso

No trabalho foi elaborado o diagrama de caso de uso, e seus descritivos de caso de uso e os protótipos de telas com as funcionalidades do aplicativo. Esse documento teve como propósito detalhar as regras de negócios e suas funcionalidades, tornando possível fazer um esboço do aplicativo e extraíndo as informações dos requisitos. Dessa maneira, visualizou-se de forma mais técnica os protótipos de telas, possibilitando melhor entendimento do aplicativo. Os descritivos e o caso de uso estão em anexo.

4.2.8. Modelo de dados

O modelo de dados, ou Entidade e Relacionamento (ER), trata-se de uma descrição das informações que estão armazenadas em banco de dados. A modelagem da ER do aplicativo é apresentada na **Figura 4**, que apresentam as seguintes tabelas nas quais os dados são armazenados: prefeitura, solicitação, usuário, status solicitação e tipo de solicitação.

A Tabela Prefeitura é responsável por armazenar os dados de cadastro inicial. Nela também são armazenadas as notificações (razão social, CNPJ, e-mail, endereço completo, senha e descrição *PUSH* notificação).

A Tabela Usuário é responsável por manter os dados cadastrados e receber as notificações (código do usuário, e-mail, usuário, telefone, senha e descrição *PUSH* notificação).

A Tabela Solicitações é a principal, na qual são armazenadas as localizações que, no aplicativo, são visualizadas por meio do *Google Maps*. Além disso, no momento da solicitação, é procurada, na mesma tabela, toda descrição dos usuários e prefeitura, (número da sequência de solicitações, código usuário, código prefeitura, data, hora e dia, tipo da solicitação, latitude e longitude, endereço, cidade, bairro, observações, *status* da solicitação, imagem).

A Tabela Status Solicitação, por sua vez, é responsável por mudar o *status* toda vez que prefeitura atualizar pedido de solicitação do usuário (antedida, em andamento, não atendida e rejeitada) e o tipo solicitação (limpeza, buraco e iluminação).

A Tabela Tipo de Solicitação é encarregada de permitir que, toda vez que o usuário efetuar uma solicitação, ele possa escolher qual tipo de solicitação pretende realizar.



Figura 4. Modelo de dados ER

4.2.9 Desenvolvimento do aplicativo

O desenvolvimento do aplicativo integra *API Google Maps*, *GPS* e *Web Service* para o funcionamento do aplicativo. A **Figura 5** apresenta duas classes que são responsáveis por identificação de ID e por mensagem “*PUSH*” de notificação. A classe “*MyFirebaseIIDService*” é responsável pelo serviço de registro de ID no *Smartphone*, caso mude esse ID o serviço é automaticamente executado e atualizado no banco. A classe “*MyFirebaseMessagingService*” tem trabalho de receber o *PUSH* do sistema operacional android, pelo motivo que mesmo que o aplicativo esteja fechado ele seja capaz de receber a mensagem solicitada.

```

public class MyFirebaseInstanceIDService extends FirebaseInstanceIdService {

    private static final String TAG = "MyFirebaseIIDService";

    @Override
    public void onTokenRefresh() {

        String refreshedToken = FirebaseInstanceId.getInstance().getToken();

public class MyFirebaseMessagingService extends FirebaseMessagingService {

    private static final String TAG = "MyFirebaseMsgService";

    @Override
    public void onMessageReceived(RemoteMessage remoteMessage) {
        Log.d(TAG, "From: " + remoteMessage.getFrom());
        Log.d(TAG, "Notification Message Body: " + remoteMessage.getNotification().getBody());

        sendNotification(remoteMessage.getNotification());
    }
}

```

Figura 5. Firebase cloud messaging

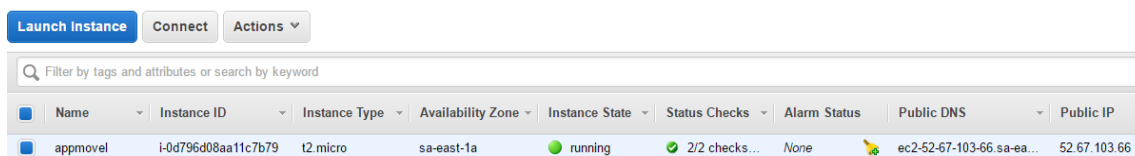
5 Resultado

Foi criada uma conta *Amazon Web Service*, que fornece um conjunto de ferramentas na nuvem que abrangem serviços nos quais operam redes, bancos de dados, análise e serviço de gerenciamento de aplicação móvel. No projeto, foram utilizados o banco de dados e o gerenciamento de aplicação móvel, de maneira que o aplicativo pudesse obter informações e enviá-las para o banco de dados quando solicitado. No *Web Service*, foram construídas duas instâncias:

1) EC2 (*Amazon Elastic Compute Cloud*) funciona como uma máquina virtual, nela roda o sistema operacional *Linux Ubuntu*.

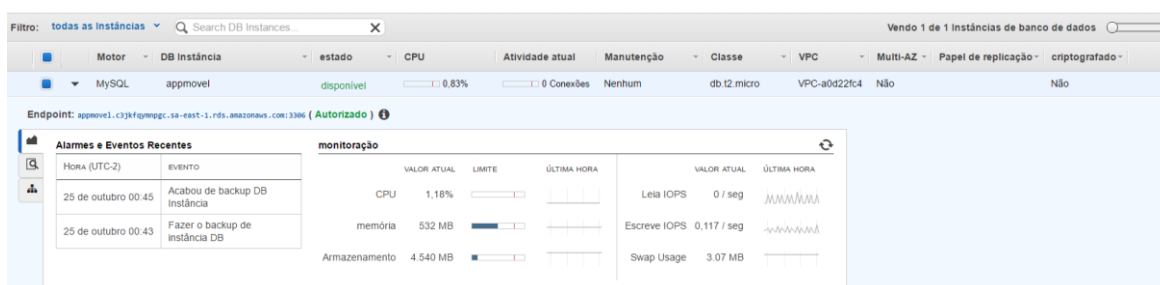
2) a RDS (*Amazon Relational Database Service*), é usada para configuração do banco de dados na nuvem, que fornece vários tipos de instâncias de banco de dados, como: *Amazon auroa, PostreSQL, Mysql, MariaDB, Oracle, Microsoft SQL Server*. Para instalações, no projeto foi usado o instalador *Mysql*.

Nas **Figura 6** e **Figura 7**, são ilustradas as duas instâncias do *WebService*.



Name	Instance ID	Instance Type	Availability Zone	Instance State	Status Checks	Alarm Status	Public DNS	Public IP
apmovel	i-0d796d08aa11c7b79	t2.micro	sa-east-1a	running	2/2 checks...	None	ec2-52-67-103-66.sa-ea...	52.67.103.66

Figura 6. EC2 Máquina virtual do Web Service



Motor	DB Instância	estado	CPU	Atividade atual	Manutenção	Classe	VPC	Multi-AZ	Papel de replicação	criptografado
MySQL	apmovel	disponível	0.83%	0 Conexões	Nenhum	db.t2.micro	VPC-a0d22f14	Não	Não	Não

Alarques e Eventos Recentes	monitoração																										
<table border="1"><thead><tr><th>Hora (UTC-2)</th><th>EVENTO</th></tr></thead><tbody><tr><td>25 de outubro 00:45</td><td>Acabou de backup DB Instância</td></tr><tr><td>25 de outubro 00:43</td><td>Fazer o backup de instância DB</td></tr></tbody></table>	Hora (UTC-2)	EVENTO	25 de outubro 00:45	Acabou de backup DB Instância	25 de outubro 00:43	Fazer o backup de instância DB	<table border="1"><thead><tr><th>VALOR ATUAL</th><th>LIMITE</th><th>ÚLTIMA HORA</th><th>VALOR ATUAL</th><th>ÚLTIMA HORA</th></tr></thead><tbody><tr><td>CPU</td><td>1.18%</td><td></td><td>Leia IOPS</td><td>0 / seg</td></tr><tr><td>memória</td><td>532 MB</td><td></td><td>Escreve IOPS</td><td>0.117 / seg</td></tr><tr><td>Armazenamento</td><td>4.540 MB</td><td></td><td>Swap Usage</td><td>3.07 MB</td></tr></tbody></table>	VALOR ATUAL	LIMITE	ÚLTIMA HORA	VALOR ATUAL	ÚLTIMA HORA	CPU	1.18%		Leia IOPS	0 / seg	memória	532 MB		Escreve IOPS	0.117 / seg	Armazenamento	4.540 MB		Swap Usage	3.07 MB
Hora (UTC-2)	EVENTO																										
25 de outubro 00:45	Acabou de backup DB Instância																										
25 de outubro 00:43	Fazer o backup de instância DB																										
VALOR ATUAL	LIMITE	ÚLTIMA HORA	VALOR ATUAL	ÚLTIMA HORA																							
CPU	1.18%		Leia IOPS	0 / seg																							
memória	532 MB		Escreve IOPS	0.117 / seg																							
Armazenamento	4.540 MB		Swap Usage	3.07 MB																							

Figura 7. Instância remota Mysql no Web Service

No ambiente desenvolvido para o usuário, realiza-se o cadastro no aplicativo, em que o cidadão informa os dados necessários, como (nome, e-mail, telefone e senha). Se as informações forem recebidas com sucesso, o usuário já poderá logar no aplicativo. Após efetuado o *login*, ficará disponível a tela do mapa em seu dispositivo móvel, mostrando sua exata localização. Prosseguido com a solicitação, o usuário selecionará o botão “+”, chamado “tela de solicitações”, em que automaticamente o campo “endereço” será preenchido com os dados de sua localização. O usuário selecionará a solicitação desejada e escreverá uma descrição e/ou realizará o envio de uma foto. A representação de “*login*”, “cadastro usuário” e “mapa” consta na **Figura 8**.

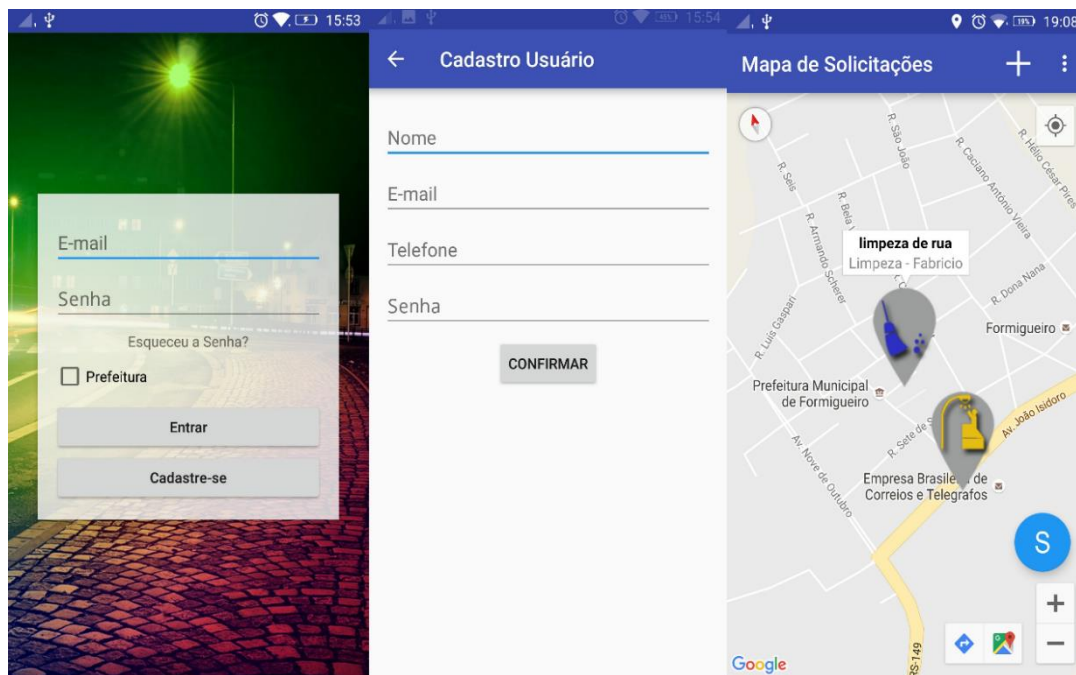


Figura 8. Telas do aplicativo

No ambiente desenvolvido para prefeitura, o órgão público realiza o seu cadastro com seus dados (razão social, CNPJ, e-mail, senha, CEP e endereço completo). A informações são geradas e enviadas para o gestor do aplicativo, que confirma se realmente são os dados da prefeitura que está realizando a solicitação.

Confirmados os dados, o órgão público já poderá efetuar o *login* no aplicativo. Após realizado o *login*, é mostrado o mapa de solicitações com pinos e descrições dos serviços solicitados pelos usuários. O aplicativo disponibiliza para a prefeitura um botão flutuante “S”, que redireciona para as solicitações recebidas, na qual a aplicação lista todas as solicitações enviadas pelos usuários, de forma que o responsável possa fiscalizar os pedidos.

O aplicativo tem função de notificação *PUSH*: quando é solicitado pedido ou reclamação pelo usuário, a aplicação gera uma mensagem notificando a prefeitura. Não é preciso estar com aplicativo aberto para receber a notificação. A **Figura 9** apresenta a tela do gestor do aplicativo, que mostra a autorização de prefeituras em que o gestor verifica se realmente é a prefeitura que está solicitando o aplicativo, de maneira a evitar usuários mau interseccionados. A **Figura 10** representa a tela da versão para o órgão público, na qual o responsável fiscaliza as solicitações.

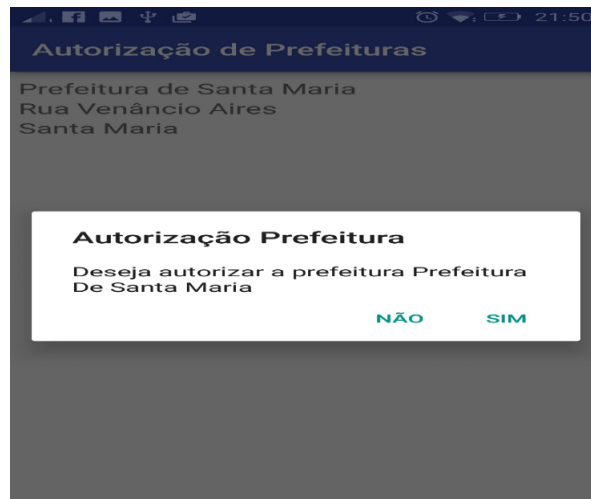


Figura 9. Tela do gestor

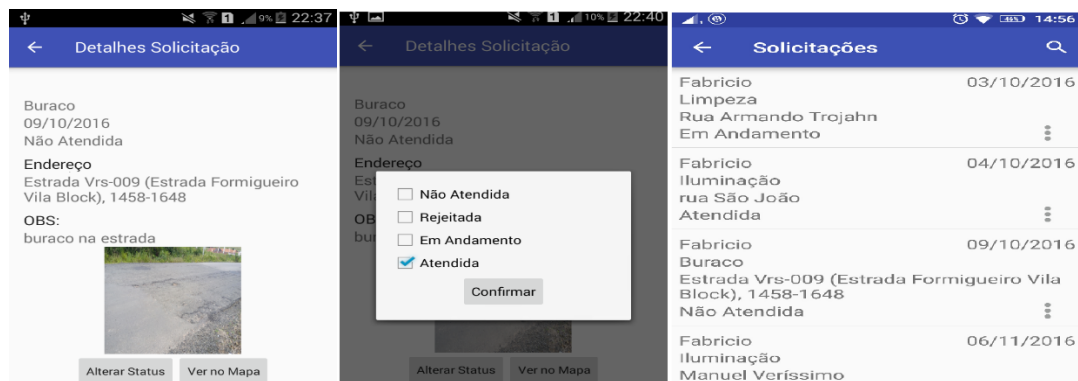


Figura 10. Tela do aplicativo da prefeitura

6. Conclusão

Neste trabalho, foi realizada uma pesquisa e uma análise de como funciona um ambiente de solicitações de limpeza, de consertos de buracos e de iluminação pública. Foram estudados artigos relacionados ao tema para elaborar uma proposta de ferramenta que visa agilizar e facilitar a interação do cidadão com o órgão público no que tange à eficácia dos serviços prestados.

Verificou-se que a maioria das reclamações dos cidadãos encontra-se em atraso, mesmo em pedidos básicos, como uma simples troca de lâmpada. A esse setor de gestão do governo, faz-se necessária uma aplicação em que o usuário possa solicitar e reclamar, permitindo a visualização por todos que possuem o aplicativo.

Foi concreta a dificuldade em localizar documentos correlatos, que trouxessem mais exemplos para orientar a criação desta aplicação e que tivessem a mesma finalidade à qual este trabalho se propõe. No decorrer do desenvolvimento do aplicativo, foram surgindo várias dúvidas e dificuldades. Uma delas foi quanto ao uso do *Web Service*, pois o pesquisador não tinha base nem conhecimento sobre como trabalhar com a ferramenta.

O aplicativo estará em constante atualização. O próximo *update* será a implantação do *login* pelas redes sociais, com a função de compartilhamento das solicitações, outra forma é parte de duplicação de solicitação que não foi tratada, de maneira que outro o usuário solicite a mesma, tornando duplicada.

Expandir com mais recursos, em que possa ser usado em vários cenários, como

por exemplo, poda de árvores, bueiros entupidos, sinaleiras ou talvez até adaptado para usos de urgência como, acidentes.

Desse modo, a pesquisa não se encerra na iniciativa do aplicativo projetado, pois novos recursos serão buscados, visando um aprimoramento do produto.

Referências

Formigueiro. (2016) “Solicitação de conserto de iluminação pública”, Prefeitura Municipal de Formigueiro, Secretaria de Obras.

IBGE (2010) “População residente, total no município de Formigueiro”. Disponível: <http://www.censo2010.ibge.gov.br/sinopse/index.php?dados=21&uf=43>

Aragão, C. V. De (1997). Burocracia, eficiência e modelos de gestão pública: um ensaio. *Revista do Serviço Público*, Artigo, v. 48, n. 3, p. 104–132.

Cysneiros, L. M. (2001). Requisitos Não Funcionais: Da Elicitação ao Modelo Conceitual. *Tese apresentada ao Departamento de Informática da PUC/RJ*, v. Doutor, p. 224 f.

Figueredo, A. M. (2007). FDD em uma casca de banana. *Artigo-PDF*, v. 1, n. Um guia de rápido aprendizado para a Feature-Driven Development.

Freitas, L. C. De (2002). Web Services. *Faculdade de Tecnologia de Americana - Artigo - tcc*, p. 575–668.

Galo, M. and Camargo, P. (1994). UTILIZAÇÃO DO GPS NO CONTROLE DE QUALIDADE DE CARTAS. *COBRAC-1994, 1 Congresso Brasileiro de Cadastro Técnico Multifinalitário*, n. Janeiro 1994, p. 41–48.

Gomes, V., Magalhães, F. de M. R., Tito, A. de oliveira, et al. (2012). UbibusRoute: Um Sistema de Identificação e Sugestão de Rotas de Ônibus Baseado em Informações de Redes Sociais. *Simposio Brasileiro de Sistemas de informação - Artigo*, n. Sbsi, p. 516–527.

Gudwin, R. R. (2013). Diagramas de Atividade e Diagramas de Estado. *Artigo - DCA-FEEC-UNICAMP*,

Lecheta, R. R. (2010). Aprenda a criar aplicações para dispositivos móveis em o Andriod SDK. *Livro*, v. 4a Edição, p. 8–9.

Maurício, C. and Figueiredo, S. (2016). Computação Móvel : Novas Oportunidades e Novos Desafios COMPUTAÇÃO MÓVEL : NOVAS OPORTUNIDADES E NOVOS DESAFIOS. *Artigo*, n. SEPTEMBER 2015.

Nunes, L. M. and Ruaro, F. L. de M. (2012). Revista da Graduação. *Artigo-Revista de Graduação*, v. 5, n. 2.

Pressman, R. S. (2002). Engenharia de Software. *Artigo - DevMedia*, p. 843.

Reis, T. D. A., Bouffleur, M. P., Vizzotto, J. K., Velho, H. F. de C. and Charão, A. S. (2006). Automatização da criação de mapas com o Google Maps. *Artigo-Acadêmico do Curso de Ciência da Computação, UFSM.*, p. 127–134.

Teixeira, M. M. and Júnior, J. A. de L. (2015). Cidadania digital: uma proposta de dispositivo móvel para o monitoramento das cidades. *Artigo - Revista Temática*, v. 1, p. 1–22.