

# DESENVOLVIMENTO DE APLICATIVO MÓVEL PARA AUXILIAR NO PROCESSO DE VISTORIA DE EDIFICAÇÕES REALIZADA PELO CORPO DE BOMBEIROS MILITAR DO TOCANTINS

João Paulo S. Paiva<sup>1</sup>

Bruno Santos Moraes<sup>2</sup>

Arlenés Buzatto Delabary Spada<sup>3</sup>

## RESUMO

O Corpo de Bombeiros Militar do Tocantins, além de atender a população em situações de emergência, também presta o serviço de proteção e segurança contra incêndios em edificações e eventos temporários, serviço esse que visa certificar se os locais estão de acordo com as normas técnicas. Contudo, para identificar se as edificações estão de acordo, é necessário fazer vistorias minuciosas. Sendo assim, visando um melhor serviço e a agilidade do sistema de vistorias, foi desenvolvido um aplicativo com base no sistema operacional *Android*, consumindo uma API REST, que visa diminuir as falhas humanas e proporcionar um melhor tempo de resposta para a corporação e a população tocaninense. Os resultados obtidos com este aplicativo demonstraram maior agilidade na vistoria e automatização de algumas tarefas que outrora eram repetitivas.

**Palavras-Chave:** Android. Bombeiros. Aplicativo. Vistoria. Habitação.

## ABSTRACT

The Tocantins Military Fire Brigade, further to serving the population in emergency situations, also provides the fire protection and security service in buildings and temporary events, a service that aims to certify that the locations are in accordance with state technical standards. However, to determine if the buildings are in agreement, it is necessary to carry out thorough surveys. In order to improve the service and agility of the survey system, an application was developed for the Android operating system that consumes a REST API that used the REST, concepts the way aims to reduce human faults and provide a better response time for the corporation and population of Tocantins. The results obtained with this application prove more agility in the inspection in the survey and automation of the same tasks that were previously repetitive.

**Keywords:** Android. Firefighters. App. Survey. Habitation.

## 1. INTRODUÇÃO

O processo de vistoria de edificações, estabelecido por meio de legislação vigente, Lei n. 1.787 de 15 de maio de 2007, é atribuído aos corpos de bombeiros. Até

<sup>1</sup> Pós-Graduando do Curso de Programação para Dispositivos Móveis – Faculdade Católica do Tocantins (FACTO) – e-mail: [bypaiva@gmail.com](mailto:bypaiva@gmail.com)

<sup>2</sup> Pós-Graduando do Curso de Programação para Dispositivos Móveis - Faculdade Católica do Tocantins (FACTO) - e-mail: [brunomoraisti@gmail.com](mailto:brunomoraisti@gmail.com)

<sup>3</sup> Professora da Faculdade Católica do Tocantins (FACTO) – e-mail: [arlenes.delabary@catolica-to.edu.br](mailto:arlenes.delabary@catolica-to.edu.br)

a efetivação da vistoria, a maioria dos projetos passa por análise para que sejam verificadas a validade e viabilidade, a fim de que a edificação fique liberada para ser construída e, posteriormente, habitada.

Uma edificação só poderá ser liberada para a ocupação de pessoas quando atender aos requisitos de segurança. Estes requisitos são previstos por meio de Normas Técnicas Regulamentadoras (NR's) disponíveis para consulta através do *sítio* do ministério trabalho, e são estas normas, as adotadas pelos corpos de bombeiros.

Uma vez que a edificação finalize sua construção, é solicitada a vistoria. Neste momento, o técnico do corpo de bombeiros deverá verificar todos os itens constantes no projeto da edificação e seguir meticulosamente, não deixando passar nenhum requisito de segurança, analisando se tudo está em conformidade com as normas regulamentadoras e com o projeto que foi aprovado. Caso algum item passe despercebido pelo vistoriador, poderá colocar em risco todos os indivíduos que irão ocupar a edificação.

Geralmente, o vistoriador realiza a vistoria munido de toda papelada do processo da edificação, devendo orientar-se por meio da planta baixa impressa e da legislação que, muitas vezes, pode não estar facilmente acessível. Entretanto, há nesse contexto uma série de situações que facilmente poderiam ser resolvidas se todos esses arquivos estivessem centralizados em um único dispositivo tecnológico.

As Tecnologias da Informação e Comunicação (TICs) têm como objetivo, além de expandir os conhecimentos e realizar o gerenciamento da informação, sanar e garantir que se possa diminuir a falha humana, criando mecanismos que orientem seus usuários a como proceder e cumprir as tarefas que lhes forem propostas.

Com o crescente uso dos *smartphones* nasceu um novo universo de possibilidades, empresas como *Google* e *Apple*, dona das duas maiores plataformas de desenvolvimento *mobile* (*Android* e *iOS*, respectivamente), disponibilizam um ambiente em que qualquer pessoa interessada possa vender ou distribuir gratuitamente seus *apps* (aplicativos, sistemas também conhecidos como aplicativos móveis) nas lojas virtuais.

Diante desse cenário, este trabalho propõe-se a abordar o desenvolvimento de um aplicativo para dispositivos móveis baseado na plataforma *Android*, sistema presente em cerca de 93,2% dos *smartphones* utilizados no Brasil, de acordo com a empresa de análise de dados Kantar (IMASTER, 2017).



O aplicativo “Vistoria 198” visará garantir a agilidade do trabalho dos vistoriadores, com funcionalidades que facilitem a comunicação orientando a forma como este profissional deve proceder a vistoria, possibilitando que ela seja supervisionada, reduzindo falhas na verificação dos requisitos de segurança, trazendo um *feedback* durante o uso do *app*.

## 2. DISPOSITIVOS MÓVEIS

O termo dispositivo móvel, embora abrangente, refere-se a um equipamento dotado de sistema operacional, recursos de entrada e saída, podendo possuir acesso à internet, sistema multimídia, sistema de posicionamento geográfico (GPS), dentre outros recursos.

Nos últimos anos, os avanços da tecnologia e maior investimento na mobilidade fomentaram o surgimento de diversos equipamentos e mudaram a forma como as pessoas vivem suas vidas. Os *Smartphones*, *Tablets*, *Notebooks*, PDA's, Aparelhos de GPS (Sistema de Posicionamento Global), estão cada vez mais conectados à informação.

O vertiginoso crescimento dos dispositivos móveis, além do fator “mobilidade”, está ligado a diversos fatores: à capacidade de serem multitarefas, à convergência de tecnologias, aos inúmeros aplicativos, à facilidade de comunicação e ao entretenimento.

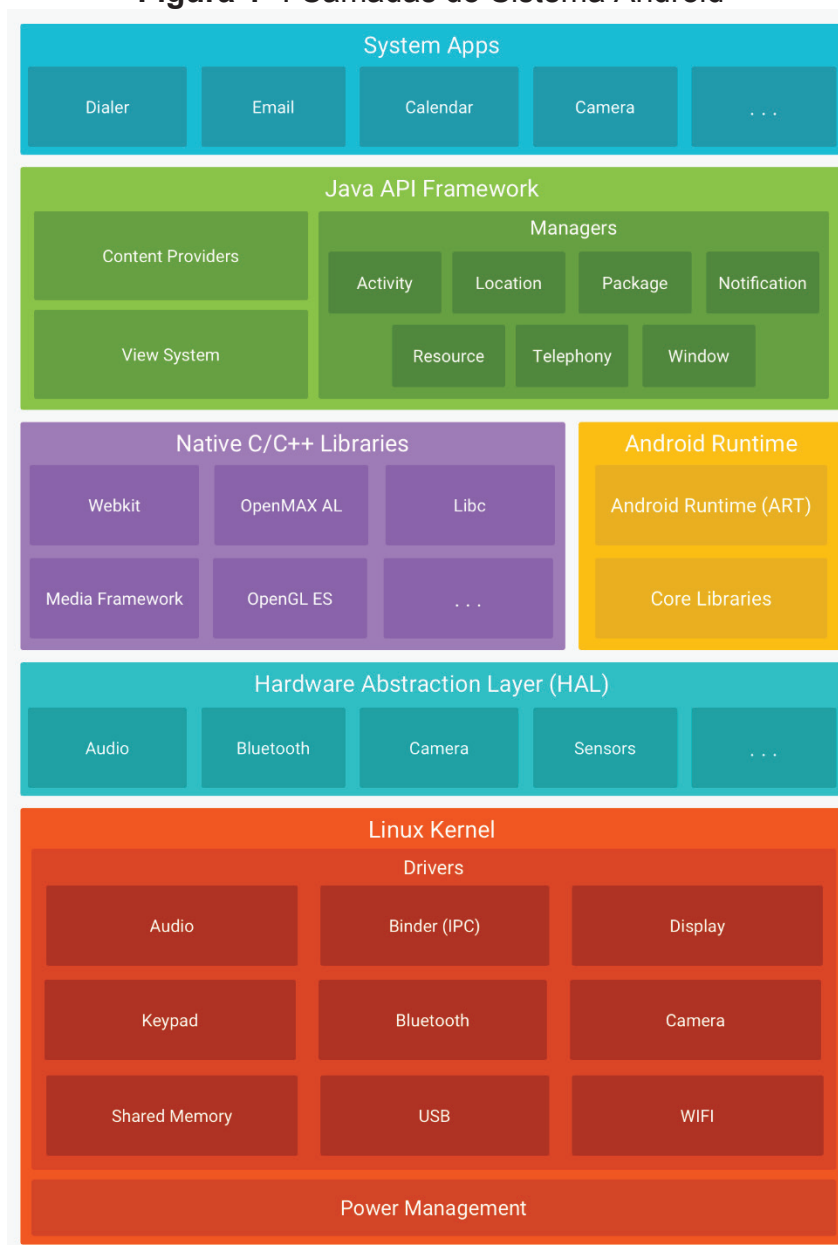
Aos poucos, os Computadores Pessoais (PC's) estão sendo substituídos, pois, o que antes estava acessível somente a um computador de mesa, para uso pessoal ou profissional, agora pode ser facilmente acessado pelo dispositivo móvel, graças à diversidade de aplicativos que estão surgindo nos *Smartphones* e *Tablets*.

Uma pesquisa realizada pelo Centro Regional de Estudos para o Desenvolvimento da Sociedade da Informação (Cetic.br), revela que, aproximadamente, 47% dos brasileiros com 10 anos ou mais usaram a internet por meio de um dispositivo móvel, em 2014. Isto representa cerca de 81,5 milhões de pessoas só no Brasil (CGI, 2017).

### 3. PLATAFORMA ANDROID

Segundo (DEV MEDIA, 2017, s/p), “o Android é uma plataforma para tecnologia móvel completa, envolvendo um pacote com programas para celulares, já com um sistema operacional, middleware, aplicativos e interface do usuário”.

**Figura 1** -. Camadas do Sistema Android



**Fonte:** (DEVELOPER, 2017)

A base do Android é um *kernel* Linux, ou seja, o núcleo do sistema operacional responsável pelo gerenciamento do hardware e periféricos, controle de usuários, sistema de arquivos e segurança. Em uma camada mais acima, encontram-se as

bibliotecas padrão, *OpenGL*, *SQLite*, *Media Framework* e o *Android Runtime*, que compreende as *Core Libraries* e o ART que substituiu a antiga *DVM (Dalvik Virtual Machine)*.

A partir da versão 5.0 (Lollipop) do Android foi adotado o uso de um mecanismo mais moderno para a execução das aplicações chamado ART (Android Runtime). A principal diferença entre ART e a *DVM* é a forma como ocorre a compilação dos dados, a Dalvik realizava essa compilação em tempo de execução (*Just-in-Time*), já no ART a compilação é realizada durante a instalação do app (*Ahead-in-Time*), tornando a execução bem mais rápida (TECMUNDO, 2018).

Na camada *Application Framework* estão as principais API's (*Application Programming Interface*) utilizadas pelos aplicativos que são executados no sistema como os componentes visuais (botões, caixas de texto), gerenciadores de localização e notificações, gerenciadores de atividade e ciclo de vida dos aplicativos.

Todos os aplicativos instalados, sejam eles nativos do sistema, tais como o aplicativo de gerenciamento de chamada, a agenda de contatos, o correio eletrônico, SMS, *Play Store* e também aqueles que podem ser adquiridos posteriormente, encontram-se na camada *Application* (DEVELOPER, 2017).

O aplicativo desenvolvido foi baseado na versão 4.1 do Android (Jelly Bean) e é compatível com versões superiores. As principais API's utilizadas pelo app restringem-se às de gerenciamento de GPS, que por meio desta é possível registrar as coordenadas das edificações nas quais as vistorias são realizadas, gerenciamento de conexão sem fio, rede móvel ou *WI-FI*, utiliza também API de notificação para informar o usuário sobre vistorias que encontram-se em aberto.

#### 4. INTEROPERABILIDADE DE SISTEMAS

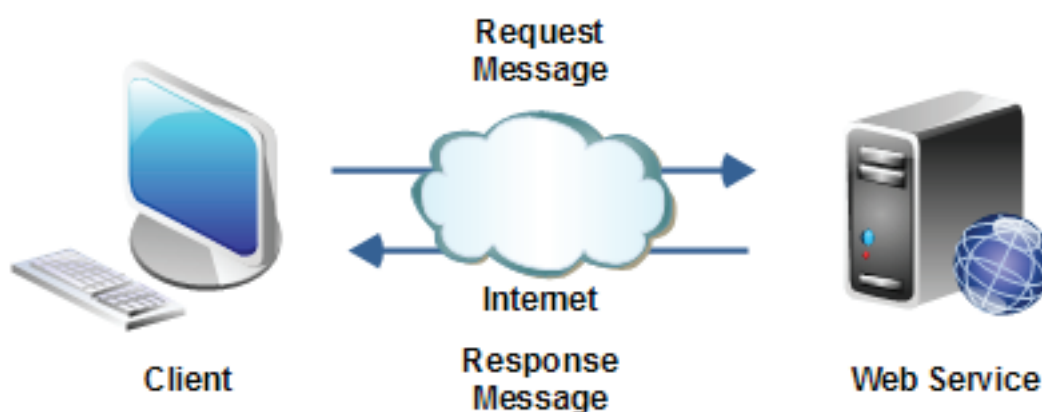
Como interoperabilidade entende-se a capacidade de um sistema informático de interagir ou se comunicar com outro sistema. Em Oliveira (2015) são definidas algumas das facetas da interoperabilidade descritas como técnica, semântica, política, intercomunitária, legal e internacional.

A interoperabilidade técnica trabalha no sentido de promover uma unificação e padronização dos recursos com o armazenamento e tratamento das informações que são disponibilizadas. Pode ser tratado como um trabalho cooperativo que instituições promovem para criar um meio de convergência para favorecer a própria comunidade.

O *Web Service* é um mecanismo que tem evoluído muito e teve sua ascensão graças às facilidades de implementação, cujo uso visa garantir a interoperabilidade entre os vários dispositivos de diferentes plataformas.

De acordo com Abinader (2006), para garantir que sistemas se tornem interoperáveis, um *Web Service* deve permitir que qualquer computador, programa ou plataforma de hardware ou software, sob os padrões estabelecidos na internet como REST, SOAP, JSON e XML possam trocar informações.

**Figura 2** - Comunicação Cliente e Provedor de Serviço



Fonte: (JENKOV, 2017)

Uma arquitetura de *Web Services* muito utilizada atualmente pelo fato de simplificar a sua implementação é a arquitetura REST (*Representation State Transfer*). Diferentemente da arquitetura SOAP (*Simple Object Access Protocol*), que permite trabalhar com vários protocolos de comunicação, o REST é baseado integralmente no protocolo HTTP (*Hypertext Transport Protocol*).

Para garantir a interoperabilidade, dois modelos de representação de dados são mais comuns entre os *Web Services*, o XML (*Extensible Markup Language*) e o JSON (*JavaScript Object Notation*). Essa padronização permite que mensagens de texto simples não encontrem dificuldades para chegar ao destinatário, pois trafegam por meio de portas padrão de comunicação.

Graças à popularização dos *Web Services*, sistemas móveis encontraram espaço para desenvolver-se. Estabeleceu-se um ambiente propício à evolução de novas tecnologias, de forma a sintetizar o conceito de interoperabilidade, garantindo

que sistemas móveis pudessem trocar informações, tornando viável um conceito chamado Computação em Nuvem, na qual o armazenamento de dados e serviços diversos está muito além do equipamento local, mas em qualquer parte do globo.

## 5. WEBSERVICE REST

A abordagem utilizada nesta pesquisa teve como ambiente de desenvolvimento o Netbeans 8.1 com pacote para desenvolvimento em PHP, cuja disponibilização é gratuita no site *netbeans.org*. A IDE está regida sob duas licenças, a CDDL (Common Development and Distribution Licence) versão 1.0 e a GPL (General Public Licence) versão 2, o que possibilita que os usuários possam usar livremente o sistema e seu código é aberto, dentre outras atribuições (NETBEANS, 2017).

Um dos entraves que muitas soluções encontram para a obtenção de êxito no mercado é o custo de colocar uma ideia em prática. Todavia, existem várias ferramentas e *Frameworks*, conjuntos de ferramentas, disponibilizadas gratuitamente e que oferecem funcionalidades interessantes para serem aplicadas dentro do escopo do problema que será solucionado.

Nesse contexto, tanto no quesito de redução de custos como também na facilidade de implementação, o Web Service foi desenvolvido baseado no *Slim Framework*, um *framework* em PHP que apresenta todos os recursos necessários para executar um provedor de serviços eficiente, utilizando a arquitetura REST.

O conceito da arquitetura REST é bastante ampla, mas pode ser resumida como sendo uma arquitetura que sabe como utilizar os métodos padrão do protocolo HTTP (*Hypertext Transport Protocol*) – *GET*, *POST*, *PUT* e *DELETE*. É muito comum que os desenvolvedores de aplicações não levem em consideração e não respeitem a documentação que está por trás da utilização desses métodos, utilizando por exemplo o método *GET* para praticamente tudo, subutilizando completamente o poder que o protocolo oferece.

Antes de criar uma instância do Slim no servidor, inicialmente, é necessário declarar quais métodos de envio que o servidor possuirá contendo nele a URL (*Uniform Resource Locator*) que o cliente deverá buscar e o nome da função que será executada. Além disso, é necessário apenas definir um cabeçalho especificando o tipo do conteúdo da página e chamar o método *registerAutoLoader* do Slim.



Todo o sistema, amplamente utilizado para as soluções desenvolvidas dentro do Corpo de Bombeiros Militar do Tocantins, está desenvolvido utilizando a linguagem PHP e banco de dados MySQL, o que tornou ainda mais concisa a adoção do *Slim*. Foram implementadas diversas funções tais como o controle de login, busca de vistorias em aberto, acesso aos arquivos de imagens, plantas baixas e demais documentos referentes à legislação vigente que pode ser consultada durante a vistoria.

## 6. MATERIAIS E MÉTODOS

A Diretoria de Serviços Técnicos (DISTEC) do CBM-TO possui um sistema web que começou o seu desenvolvimento no primeiro semestre de 2010. Hoje, ele conta com recursos como cadastramento de projetos e processos, análise de projetos, fiscalização, emissão de certificados, controle de vistorias, além de outras opções. No entanto, um dos serviços principais e que demanda grande efetivo é o serviço de vistoria, a qual está presente em todo o estado do Tocantins. É um serviço minucioso e que exige expertise, pois envolve a segurança de pessoas.

Analisando o serviço de vistoria de edificações do Corpo de Bombeiros Militar do Tocantins, foram observados alguns critérios, tais como:

- Mobilidade: a equipe de vistoria necessita visitar a edificação com toda a documentação, processo e planta baixa da edificação;
- Acesso à informação: o usuário só terá acesso ao certificado, após o vistoriador retornar à unidade para lançar o relatório de vistoria;
- Possibilidade de falhas: a vistoria de edificações é composta por itens e o vistoriador precisa identificar quais destes são necessários para aprovação do respectivo projeto.

Diante disso, após a elaboração de um projeto com os requisitos necessários para suprir as necessidades e agilizar o serviço de vistoria, foi desenvolvido um aplicativo para sistema operacional *Android* que funciona *online* e *offline*, uma vez que o vistoriador trabalha, muitas vezes, em locais sem acesso à internet.

No aplicativo com o número do projeto e ano, o vistoriador tem acesso às informações do projeto, itens a serem vistoriados, planta baixa e fotos da edificação. O vistoriador pode imprimir o relatório de vistoria, enviar para o banco de dados, e compartilhar o endereço geo-localizado.





Para o desenvolvimento do aplicativo foi necessária a utilização de softwares: *Android Studio*, *Netbeans* e *Slim Framework*.

## 7. APP VISTORIAS

O aplicativo de vistorias foi desenvolvido no Android Studio, Ambiente de Desenvolvimento Integrado (IDE) nativa para o desenvolvimento de *apps* do sistema Android, que encontra-se na versão 3.0. A IDE utiliza nativamente a linguagem Java, adotada para esse aplicativo, embora dê suporte para o desenvolvimento baseado em C++ e Kotlin, sendo esta última a mais nova linguagem adotada pelo Google.

O desenvolvimento no *Android* é realizado basicamente por meio da integração entre a interface gráfica, representada por um arquivo XML que descreve todos os componentes visuais que serão utilizados e o arquivo Java, responsável por vincular os componentes, criar a lógica, a execução das funções e eventos gerados a partir da interface.

Como pode ser observado na Figura 3, o aplicativo conta com uma tela de *login*, na qual o usuário irá inserir os dados necessários para ter acesso às funcionalidades que por sua vez possibilitará efetuar as vistorias. Após a inserção dos dados de acesso, duas etapas de autenticação são necessárias, sendo a primeira para o aparelho, através do IMEI (*International Mobile Equipment Identity*), onde este deverá estar devidamente autorizado para uso e o segundo por meio das credenciais do usuário.

**Figura 3 - Tela de Login.**



Fonte: acervo pessoal

Em situação normal que haja conexão com a internet, o aplicativo está em constante comunicação com o Webservice, seja para realizar *login*, solicitar dados de vistoria ou atualização de registros. Todavia, é comum que em algumas regiões onde há a necessidade de um trabalho de vistoria não seja possível acessar a internet.

Para que fosse possível realizar as vistorias em modo off-line, foi necessário fazer uso do SQLite, um Sistema Gerenciador de Banco de Dados (SGBD) nativo da plataforma Android que oferece todos os recursos necessários para armazenar registros utilizando a linguagem SQL (*Simple Query Language*). Todos os registros básicos para efetuar as vistorias são baixados para o aparelho e cada vistoria atendida pelo vistoriador é baixada à medida que houver a requisição.

Na Figura 4 observa-se que para realizar o download das vistorias, são necessários o número do projeto e o ano que este foi protocolado. Caso haja uma solicitação de vistoria em aberto, os dados necessários para a realização da vistoria serão baixados para o aparelho do vistoriador, ou seja, nome do proprietário, endereço da edificação, CNPJ, telefone de contato, arquivos de plantas baixas e relação dos itens que deverão orientar o vistoriador sobre o que deve ser analisado na edificação.

**Figura 4 - Busca projetos na base de dados web.**

A imagem mostra um formulário de busca web. No topo, há dois campos de entrada: o primeiro é rotulado 'Número do Projeto' e o segundo é rotulado 'Ano'. Abaixo desses campos, há um botão retangular de cor vermelha com o texto 'BUSCAR' em letras brancas maiúsculas.

Fonte: acervo pessoal

Como acesso aos recursos multimídia, o *app* também permite que o usuário possa registrar imagens durante a vistoria de uma edificação. Estas imagens poderão ser usadas no caso de contestação. Esse recurso permite que as imagens sejam salvas no aparelho e, posteriormente, enviadas para os servidores do Corpo de Bombeiros, quando haver conexão com a internet através da rede *Wifi*.

Um dos pontos mais relevantes e amplamente discutido durante o processo de elaboração do aplicativo de vistorias é o *checklist* de itens que deverão ser analisados durante a vistoria. Essa relação de itens passa por análise antes de ser incluída na vistoria, devendo ser exclusiva, compreendendo tudo que deve ser avaliado pelo vistoriador.

Uma vez que algum item da vistoria não esteja condizente com as normas de segurança, o vistoriador deverá criar uma anotação dentro do *app* para aquele item informando os detalhes da irregularidade. Todas essas anotações e a relação dos itens vistoriados ao final estarão contidos no relatório de vistoria.

Após a finalização da vistoria, o usuário poderá clicar sobre o ícone simbolizado pelo cadeado (Figura 5) para efetivar o fechamento da vistoria e assim ter acesso aos demais recursos disponibilizados pelo aplicativo: imprimir o relatório de vistoria, submeter o envio da vistoria para a base de dados do Corpo de Bombeiros e compartilhá-la no formato PDF por e-mail, *WhatsApp*, entre outros.

Uma vez que o usuário tenha finalizado o relatório, este não poderá mais excluir a vistoria até que tenha submetido à base de dados do Corpo de Bombeiros, haja vista que tal procedimento impede que o usuário use de má fé e disponibilize dados que não correspondam aos que foram enviados pelo sistema.

**Figura 5.** Funcionalidades do Aplicativo.



Fonte: acervo pessoal

Outro recurso disponível no aplicativo é o acesso ao GPS, por meio deste é possível traçar uma rota até o local da edificação, desde que tenha sido previamente registrada na base de dados. Durante a vistoria, o aplicativo permite também realizar a atualização das coordenadas, caso esses dados ainda não tenham sido registrados.

Para garantir que as vistorias estejam devidamente em conformidade com as normas, muitas vezes, faz-se necessário que o vistoriador tenha em mãos a norma propriamente dita para que possa consultá-la em caso de dúvidas sobre determinado requisito de segurança. Nesse contexto, todos os arquivos referentes às normas do CBMTO são baixadas para o aparelho e ficam disponíveis no aplicativo para serem consultadas durante a vistoria.

Ao término da vistoria, caso não haja internet para fazer a sincronização, os dados gerados ficam armazenados no aparelho até que uma conexão seja estabelecida e os dados sincronizados. O envio das imagens somente poderá ser realizado por meio de conexão Wi-fi, evitando dessa forma o consumo de dados móveis.

## 8. CONCLUSÃO

O trabalho de vistoriar edificações é realizado por todos os Corpos de Bombeiros do país e compreende uma tarefa que deve ser realizada com muita atenção, baseando-se sempre em legislação vigente.

Um dos maiores problemas relatados pelos vistoriadores e pessoal responsável pelo setor de vistorias do CBMTO e que motivou o desenvolvimento desta solução é o fato de alguns erros recorrentes estarem acontecendo. Além disso, alguns requisitos de segurança passam despercebidos em algumas vistorias, gerando inúmeros problemas tanto para a administração do CBMTO, como também para os proprietários e a própria comunidade e profissionais que convivem nesses ambientes.

A lista de itens que, anteriormente, estava disposta em uma ficha não proporcionava um meio eficiente, dispunha de uma quantidade exorbitante de itens a serem analisados ou descartados, uma vez que não os categorizava. Por meio do aplicativo, foi possível segmentar e personalizar as vistorias, tornando-as muito mais dinâmicas e confiáveis, a seleção dos itens é feita em duas etapas sob o crivo de especialistas e profissionais que irão filtrá-los e disponibilizá-los para os vistoriadores.

O objetivo do aplicativo é trazer, dentro do escopo de atuação, diversos mecanismos que reduzam os riscos de o vistoriador “deixar algo passar despercebido” durante a vistoria, bem como disponibilizar todos os dados, legislações e orientações de como ele deve proceder.

Esses procedimentos visam tornar a vistoria mais transparente para o solicitante por meio da impressão do relatórios e sua imediata notificação.

Os testes realizados com o aplicativo foram bastante satisfatórios no quesito usabilidade. Houve, por parte dos vistoriadores, grande aceitação e adaptação aos novos mecanismos dado a sua simplicidade de operação. Entretanto, algumas alterações são necessárias no que diz respeito ao processo de cadastro dos itens, uma vez que ele é feito pela intranet do CBMTO.

O trabalho aqui desenvolvido abre margem para que outras soluções semelhantes surjam, bem como incitem que gestores enxerguem alguns dos resultados obtidos mesmo com orçamento limitado, cenário comum dentro da administração pública.

## REFERÊNCIAS

ABINADER, JA. LINS, RD. **Web Services em Java**. Rio de Janeiro: Brasport, 2006.

CGI. **Uso da internet pelo celular cresce entre os brasileiros**. Disponível em: <<https://www.cgi.br/noticia/releases/uso-da-internet-pelo-celular-cresce-entre-os-brasileiros-revela-cetic-br/>> Acessado em: 15 de out. 2017.

DEVELOPERS. **Arquitetura da plataforma**. Disponível em: <<https://developer.android.com/guide/platform/index.html?hl=pt-br>> Acessado em 15 de out. 2017.

DEVMEDIA. **Adapters em diferentes views no Android**. Disponível em: <<http://www.devmedia.com.br/adapters-em-diferentes-views-no-android/28246>> Acessado em: 15 de out. 2017.

TECMUNDO. **Android: entenda as diferenças entre o ART e o Dalvik**. Disponível em: <<https://www.tecmundo.com.br/android/54387-android-entenda-diferencas-entre-art-o-dalvik.htm>> Acessado em: 13 de abr. 2018.

IMASTER. **Uso do android cresce no Brasil**. Disponível em: <<https://imasters.com.br/noticia/uso-do-android-cresce-no-brasil/>> Acessado em: 20 de out. 2017.

JENKOV. **Web Services Over View**. Disponível em: <<http://tutorials.jenkov.com/web-services/overview.html>> Acessado em: 15 de out. 2017.

NETBEANS. **Legal Materials**. Disponível em: <<https://netbeans.org/about/legal/index.html>> Acessado em: 31 de out. 2017.

OLIVEIRA, HPC, VIDOTTI, SABG, BENTES, V. **Arquitetura da informação persuasiva** [online]. São Paulo: Editora UNESP; São Paulo: Cultura Acadêmica, 2015.

PEREIRA, LCO and SILVA, ML. **Android para desenvolvedores** [online]. Rio de Janeiro: Brasport, 2009.

