

DESENVOLVIMENTO DE APLICATIVO ANDROID PARA GERENCIAMENTO DE OCORRÊNCIAS DE INCÊNDIO FLORESTAL

João Paulo S. Paiva¹
Silvano Maneck Malfatti²

RESUMO

A Defesa Civil Estadual é um órgão que trabalha na prevenção, socorro, assistência e recuperação de desastres, cujo trabalho exige a coleta de dados inerentes aos eventos que oferecem risco às comunidades, de forma que tais dados gerem informações para tentar prevenir ou minimizar os impactos gerados pelos desastres. Nesse sentido, o presente trabalho tem como objetivo demonstrar os aspectos do desenvolvimento e resultados obtidos com o aplicativo Brigada 199, que por sua vez foi desenvolvido para agilizar o processo de coleta dos dados e tomada de decisão, garantindo sua disponibilidade e exibindo com mais rapidez e precisão a real situação do cenário dos desastres que atingem os municípios do Estado do Tocantins.

Palavras-Chave: Android. Desastres Ambientais. Brigada de Incêndio.

ABSTRACT

The State Civil Defense is an organ that works on the prevention, help, assistance and disaster recovery, whose work requires the data collection inherent to events that threaten communities, so that such data manage informations to try prevent or minimize the impacts generated by disasters. In this sense, the present work aims to demonstrate the aspects of development and results obtained with app Brigada 199, aims to speed up to process of data collection and dacion-making, ensuring that data is readily available, displaying more quickly and accurately the real situation of the disaster scenario affecting the cities of the State of Tocantins.

Keywords: Android. Environment Disasters. Fire Brigade.

1. INTRODUÇÃO

O Estado do Tocantins há muito tempo sofre com desastres ambientais geralmente ocasionados devido às características climáticas e ambientais presentes nessa região do Brasil. Embora a vegetação tocantinense seja bastante variada, há maior predomínio do cerrado que, em épocas de baixa umidade no ar, torna-se o cenário ideal para as queimadas, afetando demasiadamente a qualidade do ar e aumentando a temperatura.

¹ Pós-Graduando do curso de Programação para Dispositivos Móveis - Faculdade Católica do Tocantins (FACTO) - email: bypaiva@gmail.com

² Professor Orientador do curso de Programação para Dispositivos Móveis - Faculdade Católica do Tocantins (FACTO) - email: silvanomalfatti@gmail.com

De acordo com o Instituto Nacional de Pesquisas Espaciais (INPE), no ano de 2017 foram registrados em torno de 4.070 mil focos de incêndio no Estado do Tocantins, deixando-o em segundo lugar no ranking de queimadas no Brasil, perdendo apenas para o Estado do Mato Grosso que registrou 7.312 mil focos de incêndio (CONEXAOTO, 2017).

Outros fatores que contribuem para aumentar a probabilidade dos desastres no Tocantins são a falta de informação, investimento financeiro, condições culturais e de infraestrutura. Até então não existe um mecanismo efetivo para apresentar números reais e precisos desses desastres. Muitos são baseados em estimativas que tentam aproximar-se dos reais fatores que os desencadearam.

A gestão de riscos e desastres é tema de muitas discussões, mas para que seja possível alcançar os objetivos esperados, faz-se necessário, primeiramente, aperfeiçoar os mecanismos que até então vêm sendo utilizados para a obtenção dos dados referente aos desastres. Outrossim, o estudo da técnica é papel das Tecnologias da Informação e Comunicação (TIC) que vêm investindo ao longo do tempo para tentar gerir os dados da melhor forma possível, garantindo agilidade, confiabilidade e integridade nos processos de obtenção, tratamento e representação.

Devido à popularização dos *Smartphones*, aparelhos telefônicos dotados de sistema operacional com acesso à internet, Sistema de Geolocalização (GPS), controle multimídia, controle de acelerômetro e giroscópio tornou-se viável a implementação de diversos tipos de soluções acessíveis para uma grande quantidade de pessoas. Além disso, a facilidade de implementação, em especial no que se refere ao suporte disponibilizado a interessados em desenvolver esses aplicativos (apps) para os principais sistemas operacionais do mercado, *Android* e *IOS*, tem estimulado o desenvolvimento de apps nas mais diferentes áreas.

Diante disso, o presente trabalho tem como objetivo apresentar uma solução, baseada no desenvolvimento de um aplicativo, para atender principalmente a Defesa Civil Estadual no trabalho de registro e gestão das ocorrências relacionadas aos desastres ambientais que têm grande influência no Estado do Tocantins, os quais estão relacionados aos incêndios florestais.

2. RECURSOS DO SISTEMA

As ferramentas utilizadas no desenvolvimento do aplicativo foram baseadas em licença *open-source*, para reduzir os custos de implementação. O sistema operacional



escolhido foi o *Android*, comum entre a maioria dos *smartphones* utilizados e que oferece todos os recursos necessários para viabilidade do sistema desenvolvido. O sistema web é baseado na linguagem PHP e o banco de dados é o MySQL.

2.1 SISTEMA ANDROID

O sistema foi elaborado tomando como base o *Kernel Linux*, núcleo do sistema operacional, responsável por fazer a integração entre hardware e software. Embora haja algo em comum com o sistema Linux, até pouco tempo não havia muita compatibilidade entre os sistemas. Entretanto, a partir da versão 3.3 do núcleo do Linux, o sistema passou a incorporar o código-fonte do *Kernel Android*, possibilitando que haja um trabalho colaborativo ainda maior em ambos os sistemas. (TECNOBLOG, 2017).

O *Android* surgiu em meados de 2007 como o sistema operacional para celulares inteligentes (*smartphones*) da empresa Google aliada a outras empresas num consórcio chamado OHA (*Open Handset Alliance*). Junto com o sistema, o Google lançou a sua loja virtual (*Play Store*), na qual usuários podem baixar *apps* (*Applications*) e desenvolvedores credenciados podem disponibilizar seus trabalhos (TECMUNDO, 2017).

Assim como outros sistemas que surgiram no mercado, o *Android* criou um ambiente bastante propício para diversos tipos de soluções dentro do mercado *mobile*. É bastante acessível tornar-se um desenvolvedor *Android*. Empresas e profissionais liberais têm investido e o resultado disso é um número crescente de aplicações sendo disponibilizadas na loja. Softwares dos mais variados segmentos estão tomando o lugar dos que outrora estavam dependentes do PC (*Personal Computer*).

A maioria dos *Smartphones* existentes atualmente é dotado de GPS, Câmera de alta definição, acesso a rede sem fio (Wifi, 3G ou 4G), controle de toques (*Touch Screen*) e alguns mais modernos possuem inclusive sistema biométrico para leitura de digital. O sistema deve criar uma interface para realizar o gerenciamento desses recursos e disponibiliza *API's* (*Application Programing Interface*) para que os desenvolvedores possam interagir com esses recursos para criar suas soluções de software.

De acordo com Hubsch (2012), na arquitetura do *Android*, como pode ser observado na Figura 1, entre o *Kernel* e os aplicativos podemos encontrar as bibliotecas (*libraries*), o *Android Runtime* e o conjunto de *frameworks*.

Semelhantemente ao que ocorre com a linguagem Java, todas as aplicações no *Android* são interpretadas por uma máquina virtual, a *Dalvik Virtual Machine*, por meio de um arquivo *Dalvik Executable* (dex) que, ao final, são compactados gerando um *Android Package File* (apk).

Figura 1 - Arquitetura da Plataforma *Android*



Fonte: (VIVAOLINUX, 2018)

Os *frameworks* disponíveis no *Android* abrangem os principais recursos do sistema para o desenvolvimento das aplicações, como o conjunto de ferramentas para o provedor de conteúdo, gerenciamento de notificações, sistema de interface gráfica (*Views*), controle do ciclo de vida das atividades (*Activities*) e do próprio aparelho, como controle de estado e de chamadas telefônicas.

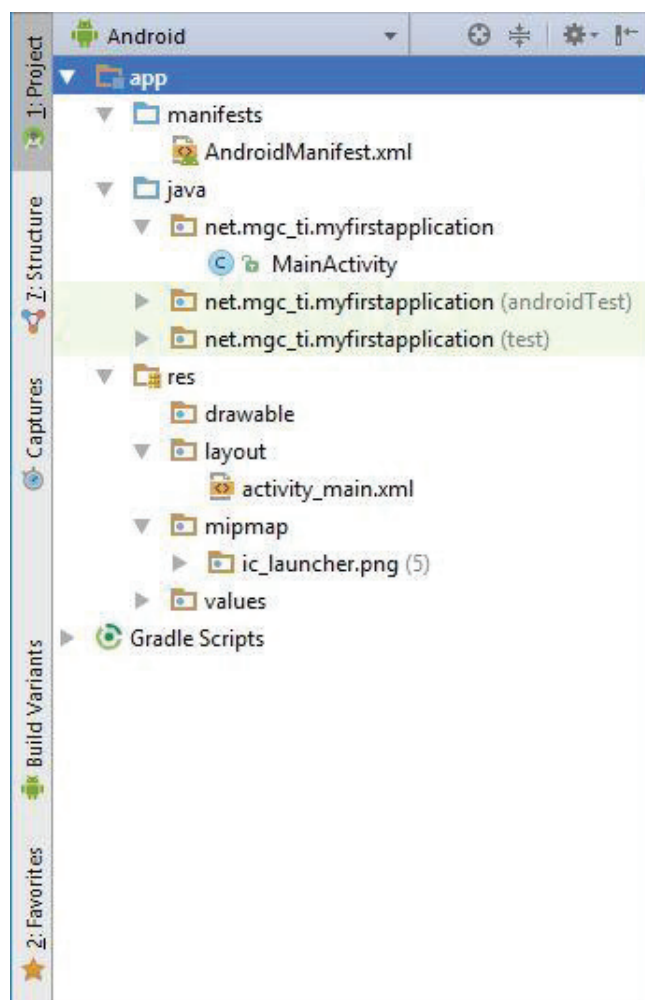
Neste trabalho, foram utilizados o GPS e a Câmera do aparelho, bem como armazenamento e bibliotecas internas e externas. Dentre as principais bibliotecas externas utilizadas estão a de mapas do google, notificações *push* com o *firebase message* e componentes *cardview* e *recycleviews*.

2.2. AMBIENTE DE DESENVOLVIMENTO

A *Integrated Development Environment* (IDE) do *Android* é o *Android Studio*, atualmente, encontra-se na versão 3.0.1 é desenvolvido em cooperação com a JetBrains, baseado na plataforma IntelliJ IDEA e disponibilizado gratuitamente pelo Google. A linguagem de programação nativa é o Java. Entretanto, após o lançamento da versão 3 a IDE inclui suporte nativo à linguagem kotlin (DEVELOPER, 2017).

O gerenciamento de compilação no *Android Studio* é feito por meio do *Gradle*, um *kit* de ferramentas avançado que permite configurações de compilação personalizada e controle de dependências. Quando iniciamos um projeto, um processo *build* é chamado executando-se o *Gradle* para construir a estrutura de módulos, diretórios e arquivos necessários, como pode ser visto na Figura 2 (ANDROIDPRO, 2017).

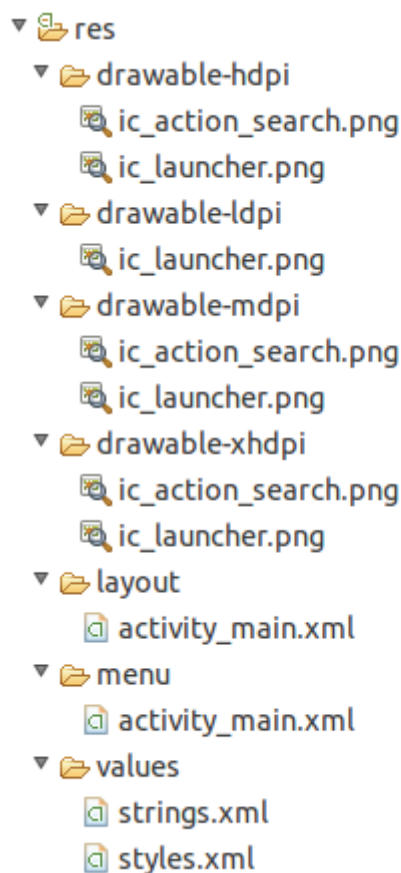
Figura 2. Estrutura de um Projeto *Android*



Fonte: (MGC, 2017)

Durante compilação de um projeto *Android*, o *Gradle* criará um processo de *build* para mesclar todos os recursos que estão sendo utilizados e que estão localizados nas pastas *res* (*Resources*) e *asset*, dando origem a um arquivo de controle chamado *R.java*. Como pode ser observado na Figura 3, os recursos descritos na pasta *res* estão representados na forma de imagens e arquivos XML (*Extensible Markup Language*).

Figura 3. Pasta de Recursos



Fonte: GUIA DO DESENVOLVEDOR (2017)

De acordo com a Figura 3, os arquivos XML que estão distribuídos nas pastas *layout*, *menu* são representações de elementos visuais que através da classe *R* do Java são mapeados e referenciados por meio de identificadores ID. Na pasta *values*, o arquivo *strings.xml* fornece ao desenvolvedor a possibilidade de configurar diferentes tipos de idioma para o aplicativo e por sua vez o arquivo *styles.xml* cria estilos de formatação diferentes para serem configurados nos arquivos de *layout*.

O *AndroidManifest*, que também está descrito no formato XML, carrega consigo informações essenciais para o *build* do aplicativo. Nele são configurados por exemplo:

o nome do pacote Java; as permissões que o aplicativo deve ter para acessar partes protegidas da API e os componentes do aplicativo; os processos que hospedam os componentes do *app*; as bibliotecas às quais o aplicativo deve se vincular; o nível mínimo da *Android* API que o aplicativo exige, entre outras configurações são especificados (DEVELOPER, 2017).

Muitas dependências que por ventura possam ser utilizadas nem sempre necessitam ser baixadas e copiadas para o projeto, pois o *Gradle* suporta dependências externas a partir de repositórios *Maven* e *Ivy*, bastando apenas serem referenciadas especificando o nome do pacote no arquivo *Gradle*. Nos repositórios públicos do *Maven* estão as bibliotecas e ferramentas mais populares utilizadas pelos desenvolvedores (ANDROID PRO, 2017).

No projeto, devido à necessidade de acesso à internet tanto por meio da *wifi* como também da rede móvel, foi necessário descrever dentro do *Android Manifest* as permissões de acesso para esses recursos. Permissões para verificar o estado da conexão (*Wifi State* e *Network State*) devem ser explicitamente declaradas, bem como as permissões para armazenamento de arquivos no aparelho (*Write External Storage*), GPS (*Fine Location* e *Coarse Location*) e estado do aparelho (*Read Phone State*).

Duas bibliotecas externas foram implementadas no projeto e que necessitam de permissão adicional no *Manifest* para serem manipulados dentro do aplicativo, as bibliotecas de mapa e notificações *push*. Em versões mais atualizadas do sistema *Android* é exigido que o programador adicione no código que implementa o recurso, uma solicitação de permissão para que o usuário possa liberar ou bloquear o acesso do *app* a tal recurso do aparelho, como por exemplo a do GPS.

2.3 BANCO DE DADOS SQLITE

O SQLite é um banco de dados relacional cujo pacote está disponível nativamente no SDK (*Software Development Kit*) do *Android*, além de ser *Open-Source*. O que torna o SQLite um banco de dados bastante versátil para uso em dispositivos móveis é o fato de toda a estrutura do banco ficar armazenada em um arquivo binário, contendo as *tables*, *views*, *procedures* e *triggers*.

Os dados persistidos podem ser manipulados por meio de comandos SQL (*Simple Query Language*) e suportam dados do tipo *NULL*, *INTEGER*, *REAL*, *TEXT* e *BLOB*. O SQLite é um Sistema Gerenciador de Banco de Dados (SGBD) dinâmico, no

qual um campo de uma tabela pode receber qualquer tipo de dado, exceto para valores inteiros. Dessa forma, o motor simplesmente ignora o tipo informado na criação das tabelas.

Os tipos de dados suportados pelo SQLite são, na verdade, classes de armazenamento. Na maioria dos bancos de dados convencionais, os dados estão associados a contêineres de tipos de valores, já o SQLite trata-os por meio de seus valores propriamente ditos e, durante a execução de um comando SQL, pode convertê-los entre classes numéricas (*INTEGER* e *REAL*) ou texto (*TEXT*).

Dado o número limitado das classes já mencionadas que o SQLite possui, outros tipos de dados que comumente são utilizados são tratados por meio de funções que os convertem, como é caso dos valores booleanos, que podem ser interpretados como zero (*false*) e um (*true*). Valores do tipo data e hora podem ser tratadas como *TEXT*, *REAL* ou *INTEGER* (SQLITE, 2018).

Na API de acesso ao SQLite no sistema *Android*, duas classes são mais utilizadas, *SQLiteDatabase* e *SQLiteOpenHelper*. A primeira contém métodos para manipulação dos dados no banco e a segunda é responsável pela criação do banco e pelo versionamento. As classes que estendem de *SQLiteOpenHelper* devem implementar dois métodos, *onCreate* e *onUpgrade* para, respectivamente, criação do banco e atualização (DEVMEDIA, 2018).

Para realizar a manipulação dos dados, deve-se chamar o método *getWritableDatabase* de *SQLiteOpenHelper* retornando uma instância de *SQLiteDatabase* que possui os métodos *query*, *insert*, *update* e *delete*. Nos métodos de inserção e atualização espera-se que seja utilizada a classe *ContentValues*, que faz associação chave e valor dos registros que serão inseridos no banco.

Embora existam outros mecanismos para persistência de dados nos dispositivos móveis, o banco de dados SQLite é um dos mais utilizados pelo fato de já estar nativamente disponível no sistema *Android* e por ser de fácil manipulação, uma vez que utiliza comandos SQL que a maior parte dos desenvolvedores já está familiarizada.

Neste projeto, foi criada uma classe intitulada de *DatabaseHelper* que estende de *SQLiteOpenHelper* e classes DAO (*Data Access Object*) para acesso e manipulação dos dados nas tabelas criadas no banco de dados. Toda classe DAO possui como atributo um objeto da classe *SQLiteDatabase* e implementa os métodos



de uma *interface* para abrir e fechar o acesso ao banco e os métodos de inserir, remover, atualizar e buscar os registros cadastrados.

3. SISTEMAS LEGADOS

O Brigada 199 foi desenvolvido baseado em um sistema que já estava sendo utilizado pela Defesa Civil. Esse sistema estava incorporado à intranet do CBMTO (Corpo de Bombeiros Militar do Tocantins) e sua finalidade era registrar os dados coletados em campo pelos agentes de Defesa Civil que o faziam de forma manual por meio de formulários de papel.

Após o registro dessas ocorrências em campo, que se valiam de aparelhos GPS para identificar as coordenadas do local da ocorrência, esses formulários eram recolhidos e encaminhados para a unidade onde eram transcritos para o sistema WEB localizado no QCG-BM (Quartel do Comando Geral do Corpo de Bombeiros Militar).

Em detrimento à logística empregada até o efetivo registro das ocorrências no sistema, muitos dos dados eram perdidos antes de tornarem-se estatística, o que deteriorava a capacidade de tomada de decisões por parte da administração do CBMTO e deixavam uma lacuna no número real de ocorrências atendidas.

Outro fato relevante é que não havia o devido tratamento desses dados. Uma vez registrados, eles eram exibidos apenas de forma quantitativa e não havia no sistema legado um mecanismo que pudesse qualificar tal dado a ponto de ser considerado informação de cunho valioso para administração que busca informações precisas para designar os devidos recursos para, por exemplo, reduzir os danos causados pelos desastres naturais, como os incêndios florestais.

4. SOLUÇÃO

Duas linhas de desenvolvimento foram abordadas para a solução do problema. A primeira refere-se à coleta dos dados, que a princípio era realizada de forma manual, lenta e precária. Agora, por meio de um aplicativo para dispositivo móvel, busca-se automatizar o máximo possível, deixando a cargo do usuário o mínimo de informações possíveis.

A segunda abordagem é no tratamento adequado dos dados coletados, realizando o cruzamento de dados e geração de gráficos e estatística de atendimentos realizados por meio de um aprimoramento do sistema WEB já existente.

4.1 BRIGADA 199

O Aplicativo foi desenvolvido com o foco no registro de ocorrências de incêndio florestal, embora permita que sejam registradas ocorrências de diversas naturezas com base na Classificação e Codificação Brasileira de Desastres (COBRADE), cuja classificação diz respeito às ocorrências que comumente a Defesa Civil atende.

O *app* foi disponibilizado na loja do Google devido às constantes atualizações submetidas durante o desenvolvimento e a praticidade para que os usuários possam baixar e atualizar o aplicativo sempre que houver a necessidade. Todavia é um aplicativo restrito aos profissionais ligados à Defesa Civil do Estado do Tocantins.

Para ter acesso ao aplicativo, o usuário deve estar devidamente cadastrado no sistema WEB. Após a autenticação por meio de usuário e senha, Figura 4, o sistema exige que o aparelho do usuário esteja liberado, essa verificação é feita através do IMEI (*International Mobile Equipment Identify*).

Figura 4 - Tela de Login do Aplicativo

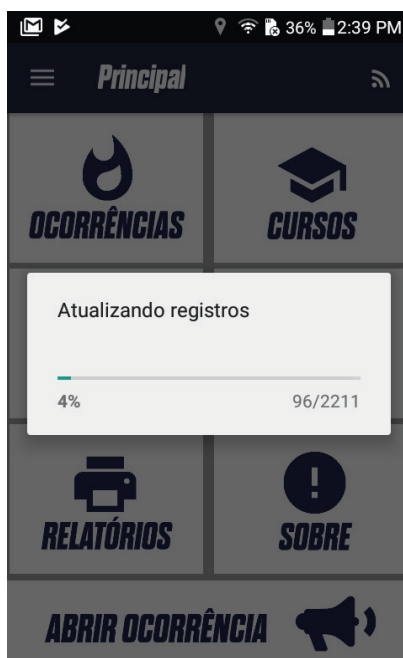


Fonte: acervo pessoal

Como pode ser observado na Figura 5, o aplicativo inicialmente solicita ao servidor do CBMTO todos os dados necessários para o registro de ocorrências. Todos os dados devem ser baixados e armazenados no banco de dados local do aparelho.

Essa sincronização inicial é obrigatória, pois sem ela não é possível operar o aplicativo.

Figura 5 - Tela de Atualização com Servidor Web



Fonte: acervo pessoal

As principais funcionalidades disponíveis no *app* são: gestão de ocorrências, gestão de cursos, gestão de brigadas e gestão de brigadistas (Figura 6). O Brigada 199 conta também com recurso de envio e recebimento de mensagens (*Push Notifications*), esse recurso foi desenvolvido baseado no FCM (*Firebase Cloud Message*) disponibilizado pelo Google gratuitamente.

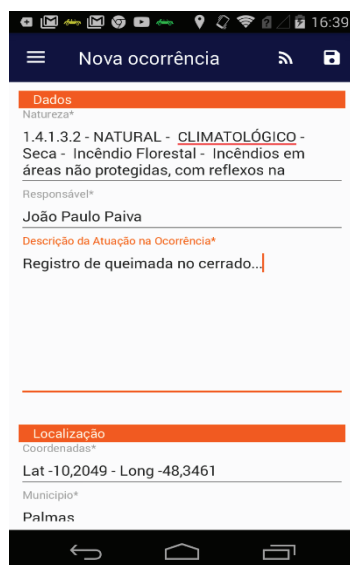
Figura 6 - Tela Principal do Aplicativo



Fonte: acervo pessoal

Para o registro de ocorrências o usuário deve selecionar a natureza da ocorrência (Figura 7), uma breve descrição do cenário onde ocorreu e registrar imagens utilizando a câmera do aparelho. Demais dados, como nome do responsável, coordenadas, município, data e hora, são obtidos automaticamente.

Figura 7 - Tela de Cadastro de Ocorrências



Fonte: acervo pessoal

As ocorrências, uma vez registradas, podem ser visualizadas pelo usuário que as registrou e também profissionais lotados no mesmo município ou unidade. De

acordo com a Figura 8, os registros podem ser visualizados no *app* por meio de uma lista, na qual as ocorrências estão ordenadas cronologicamente e também por meio do mapa, exibindo-as através das coordenadas que foram registradas.

Figura 8 - Lista e Mapa com Ocorrências Registradas



Fonte: acervo pessoal

As funcionalidades de gestão de cursos, brigadas e brigadistas, embora estejam disponíveis no *app*, ainda não foram efetivamente exploradas no campo de atuação em que o aplicativo foi submetido, entretanto os resultados obtidos com o registro de ocorrências foram bastante satisfatórios.

4.2 RESULTADOS

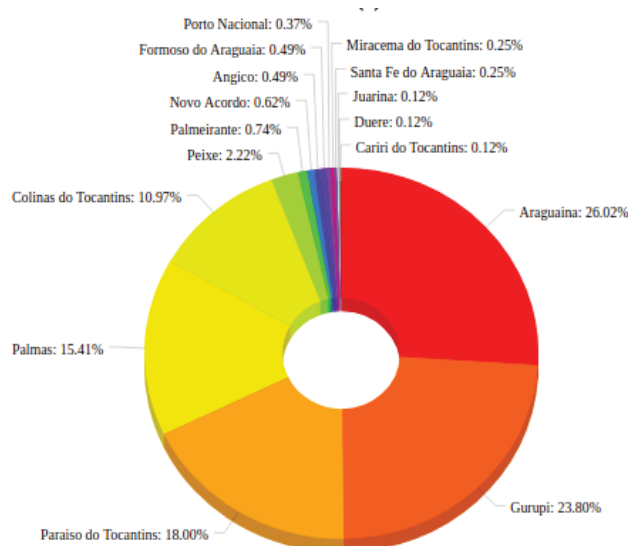
Os testes com o aplicativo foram realizados entre de julho e novembro de 2017, período em que a umidade sofre uma queda drástica e o cerrado fica propício a desencadear incêndios devido às próprias características da vegetação, associadas às más práticas dos moradores da região, com a queima ilegal como meio de roçagem.

Os resultados foram determinados através da atuação de bombeiros e brigadistas que realizaram o combate e registro das ocorrências atendidas nos municípios de Araguaína, Gurupi, Paraíso, Porto Nacional, Palmas, Colinas do Tocantins, entre outros municípios com menor número de ocorrências.

De acordo com a análise realizada, verificou-se que dentre os municípios citados, Araguaína atendeu 26,02% do total de ocorrências, o maior número de

ocorrências registradas pelo aplicativo, seguido de Gurupi com 23,8%, Paraíso do Tocantins com 18% e Palmas com 15,41%, como pode ser observado na Figura 9.

Figura 9 - Gráfico de Ocorrências Registradas no ano de 2017



Fonte: acervo pessoal

Baseando-se nessas informações, já é possível direcionar maiores recursos para as localidades que mais foram afetadas pelos incêndios florestais. Observou-se, também, que a eficiência e praticidade na coleta dos dados torna possível uma análise em tempo real para então buscar a melhor solução para o problema.

Durante o período em que houve a coleta dos dados e registro das ocorrências pelo aplicativo, gestores de vários municípios estavam continuamente acompanhando por meio da plataforma *web* os resultados que seus municípios vinham gerando, o que motivou ainda mais a utilização do *app* ao perceber que os registros estavam facilmente acessíveis.

5. CONCLUSÕES

O aplicativo foi desenvolvido com foco no atendimento de ocorrências de incêndios florestais devido ser um problema que assola o Estado do Tocantins, bem como outros Estados do Brasil que carecem de um mecanismo que traga uma resposta mais rápida e eficiente para a tomada de decisão.

Em pouco tempo, o aplicativo já demonstrou ser muito mais eficiente que os métodos de coleta e tratamento de dados anteriormente utilizados. Explorando a

praticidade e facilidade de operação, rapidamente e com poucos toques na tela, o usuário pode registrar e, automaticamente, a ocorrência é enviada para a base de dados tornando-se informação útil.

Como proposta futura, deve-se buscar explorar um mecanismo que possa, além de registrar um fato ocorrido, desenvolver um ambiente no qual a própria comunidade possa participar do processo e envolver-se cada vez mais como agente responsável e defensor de seus interesses e da comunidade e possa realizar denúncias ou alertar as autoridades quanto a possíveis riscos.

Em consonância às condições financeiras que o país enfrenta e diante de inúmeras limitações que o setor público oferece, o sistema foi todo desenvolvido com softwares livres e o custo empregado no projeto foi extremamente baixo. A adaptação desse sistema a outros Estados que tenham interesse em adotá-lo é possível com poucas alterações.

REFERÊNCIAS

ANDROID PRO. **Gradle para Android.** Disponível em: <<http://www.androidpro.com.br/gradle/>> Acessado em 29 de nov. de 2017.

CONEXAOTO. **Tocantins é o segundo no ranking de queimadas em 2017.** Disponível em: <<http://conexaoto.com.br/2017/07/25/tocantins-e-o-segundo-no-ranking-de-queimadas-em-2017- apenas-neste-mes-ja-foram-registrados-1-759-focos>> Acessado em 28 de fev. 2018.

DEVELOPER. **Introdução ao Android.** Disponível em: <<https://developer.android.com/studio/intro/index.html?hl=pt-br>> Acessado em 29 de nov. de 2017.

DEVMEDIA. Criando um CRUD com Android Studio e SQLite. Disponível em: <<https://www.devmedia.com.br/criando-um-crud-com-android-studio-e-sqlite/32815>> Acessado em: 03 de jan. de 2018.

GUIA DO DESENVOLVEDOR. Estrutura básica de um projeto Android. Disponível em: <<https://guiadodesenvolvedor.wordpress.com/2012/11/27/estrutura-basica-de-um-projeto-android/>> Acessado em: 29 de nov. de 2017.

HUBSCH, EDUARDO. **Uma Abordagem Comparativa do desenvolvimento de aplicações para dispositivos móveis.** São Paulo: Faculdade de Tecnologia de São Paulo. 2012.

MGC. Estrutura de um Projeto Android. Disponível em: <<http://www.mgc-ti.net/estrutura-de-um-projeto-android/>> Acessado em: 29 de nov. de 2017.

SQLITE. *Datatypes is SQLite version 3*. Disponível em: <<http://www.sqlite.org/datatype3.html>> Acessado em: 03 de jan. de 2018.

TECMUNDO. **Android completa 7 anos**. Disponível em: <<https://www.tecmundo.com.br/android/65508-android-completa-7-anos-confira-fatos-marcantes-trajetoria.htm>> Acessado em: Acessado em 28 de nov. 2017.

TECNOBLOG. **Kernel 3.3 do Linux incorpora núcleo do Android**. Disponível em: <<https://tecnoblog.net/95250/kernel-linux-android/>> Acessado em 28 de nov. 2017.

VIVAOLINUX. **Instalando o Android SDK na Plataforma Linux**. Disponível em: <<https://www.vivaolinux.com.br/artigo/Instalando-o-Android-SDK-na-plataforma-Linux>> Acessado em 28 de fev. de 2018.

