

Análise sobre o desenvolvimento de um sistema para rastreamento de dispositivos móveis com o sistema operacional Android

Thiago Silva Prates¹

Resumo:

A integração entre computadores e dispositivos móveis vem revolucionando o mundo. Novos desafios e oportunidades vem surgindo a todo momento, hoje principalmente referentes aos conceitos de mobilidade. A evolução e dependência destes aparelhos é tamanha que acabaram por se tornarem itens indispensáveis ao cotidiano da maioria das pessoas. Neste cenário, essas mudanças acabaram por também influenciar o desenvolvimento de software. Em decorrência a isso, o seguinte trabalho propõe análise e estudo de um protótipo de um sistema de rastreamento móvel, focado na plataforma *Android*, capaz de utilizar os recursos e conceitos da geolocalização ao suporte à tomada de decisões e gerenciamento operacional, em tempo real, por parte da empresa e entidades competentes interessadas.

Palavras-Chave: *Android*. Geolocalização. Internet. Dispositivos móveis. *Webservices*.

1 Introdução

O rastreamento de dispositivos móveis vem se destacando ao oferecer à *smartphones* e *tablets* a possibilidade de obtenção das coordenadas geográficas em tempo real para construção de aplicativos móveis mais inteligentes e sofisticados.

Não é difícil encontrar diversas aplicações móveis que fazem uso de recursos de geolocalização. Exemplos incluem desde a obtenção da localização atual do aparelho até complexos cálculos de trânsito, em tempo real, de melhores rotas e percursos, bem como a proximidade de pontos turísticos. Tudo isso *out-of-the-box*, sem a necessidade de dispositivos externos complementares (MORIMOTO, 2009).

Um fator importante neste novo cenário se deve a queda de preços de aparelhos GPS, não impactando decisivamente no custo final do aparelho. Para Morimoto (2009, p.351), "com a miniaturização dos componentes, passou a fazer sentido incluir receptores de GPS em *smartphones*, aproveitando a tela, o processador, memória e os demais componentes".

O sistema operacional móvel *Android* oferece suporte e recursos de geolocalização que aliados a um interessante modelo de desenvolvimento que permite para fins didáticos a solução ideal à realização do trabalho a seguir.

¹ Graduado em Engenharia da Computação pela Faculdade de Ciência e Tecnologia de Montes Claros - FACIT
Pós-Graduado em Engenharia de Sistemas pela Escola Aberta Superior do Brasil - ESAB

A plataforma Android desfruta hoje de um papel de destaque no mercado, tanto pela quantidade significativa de dispositivos produzidos como também por oferecer uma API rica, disponibilizando fácil acesso a vários recursos de hardware, tais como Wi-Fi e GPS, além de boas ferramentas para o desenvolvedor (MONTEIRO, 2012, p. 1).

Conforme resume Lecheta (2013, p.21), "o mercado de celulares está crescendo cada vez mais. Estudos mostram que hoje em dia mais de 3 bilhões de pessoas possuem um aparelho celular, e isso corresponde a mais ou menos metade da população mundial". Um imenso contingente de usuários à espera de aplicações móveis. Ao contextualizar para a realidade brasileira percebe-se ser um campo de oportunidades acerca do tema proposto.

Num país de dimensão continental como o Brasil, com uma grande carência de informações adequadas para a tomada de decisões sobre os problemas urbanos, rurais e ambientais, o Geoprocessamento apresenta um enorme potencial, principalmente se baseado em tecnologias de custo relativamente baixo, em que o conhecimento seja adquirido localmente. (CÂMARA & DAVIS, p. 1, s. d.)

2 Sistema de Informações Geográficas (SIG)

Os *Sistemas de Informações* destinados ao processamento de dados espaciais são denominados de *Sistema de Informações Geográficas* (SIG), e podem envolver diferentes áreas: desde diferentes tipos de *softwares* até sofisticados modelos de *hardware*. De acordo Brito & Rosa (1996), são instrumentos ou recursos destinados a aquisição, armazenamento, manipulação, análise e apresentação de dados ou informações espacialmente referidos.

O termo *Sistemas de Informação Geográfica* (SIG) é aplicado para sistemas que realizam o tratamento computacional de dados geográficos e recuperam informações não apenas com base em suas características alfanuméricas, mas também através de sua localização espacial; oferecem ao administrador (urbanista, planejador, engenheiro) uma visão inédita de seu ambiente de trabalho, em que todas as informações disponíveis sobre um determinado assunto estão ao seu alcance, interrelacionadas com base no que lhes é fundamentalmente comum -- a localização geográfica (CÂMARA & QUEIROS, p. 1, s. d.).

Ao analisarmos o contexto dos dispositivos móveis atuais, verdadeiros “computadores” capazes de integrar a diversos serviços oferecidos pela *Web*, é possível perceber as várias opções que estes podem trazer na construção de Sistemas de Informações mais consistentes e eficazes. Pois conforme diz Câmara & Monteiro (s. d.), trabalhar com “geoinformação” significa justamente utilizar computadores à representação de dados espacialmente referenciados.

Através da utilização de dispositivos móveis busca-se através deste trabalho oferecer um estudo sobre temas e ferramentas capazes a auxiliar à solução problemas que envolvam localização e rastreamento.

3 GPS: *Global Positioning System*

O GPS (*Sistema de Posicionamento Global*) é um serviço de localização desenvolvido pelo *Departamento de Defesa dos Estados Unidos* (DoD), que a partir do ano 2000 acabou se tornando gratuito e de uso geral, eliminando a necessidade de degradação do sinal em tempos de guerra ou em períodos de conflito (TECHGEO, 2014).

O GPS é composto por 24 satélites artificiais que orbitam na atmosfera terrestre a 20.2 Km de altitude em uma órbita geoestacionária. As órbitas dos satélites foram planejadas de maneira que pelo menos 4 satélites estejam visíveis a partir de qualquer ponto do globo terrestre (MORIMOTO, 2009).

Na figura 1 é apresentado um esquema dos satélites que compõem o GPS.

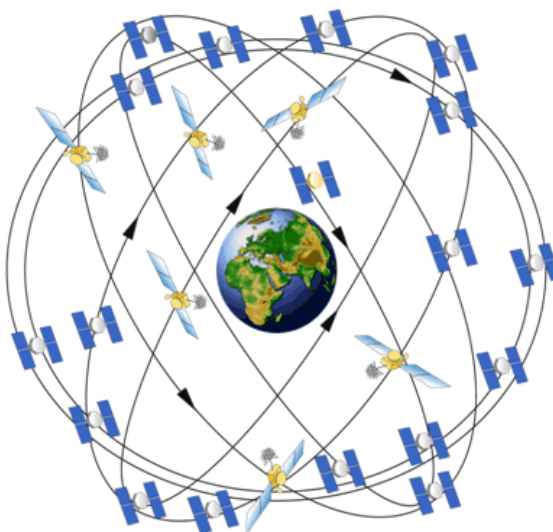


Figura 1: Satélites de Sistema de Posicionamento Global (GPS).

Fonte: http://www.fgi.fi/fgi/sites/default/files/topics_and_themes/Paikannussatelliittijarjestelmat-teemakuva.gif

Cada satélite apresenta um relógio atômico a bordo responsável pelo cálculo da localização do usuário através de um processo denominado de “*triangulação*”. Nesse processo, três satélites são posicionados de forma a calcular diante de qualquer posição a atual localização do usuário.

Os satélites transmitem um sinal de alta frequência, contendo pacotes de informação com indicações precisas da hora em que cada um foi transmitido. Os receptores em terra captam o sinal e usam um sistema de trilateração para calcular a posição, comparando a diferença de tempo entre a transmissão e a recepção de cada pacote,

calculando assim a distância em relação aos satélites muda, gerando uma pequena diferença no tempo do percurso, que usado para atualizar a localização. (MORIMOTO, 2009, p. 349)

Para utilização do GPS há a necessidade de um “receptor” um aparelho, especialmente projetado, capaz de entender o sinal enviado pelo satélite informando as coordenadas (latitude e longitude) do usuário (HUERTA et al, 2005).

4 Google Maps

O *Google Maps* é um famoso serviço de mapas que permite uma navegação de forma rápida, simples e intuitiva. É oferecido gratuitamente pela Google e pode ser encontrado em praticamente qualquer dispositivo móvel, com versões nativas para o *Symbian*, *Android*, *Windows Phone* e *iPhone* (MORIMOTO, 2009).

Segundo Marisco (2004), os serviços de mapas podem disponibilizar aos usuários comuns a possibilidade de acesso às informações geográficas de modo interativo e dinâmico, com recursos como símbolos gráficos em um produto cartográfico interativo através de objetos ativos, clicáveis e recuperáveis através de um servidor de banco de dados.

Apesar de existirem diversos outros serviços de mapas, tais como: o *Nokia Maps*, *Bing Maps*, *Garmin Mobile XT*, etc. que podem ser facilmente integrados a suas aplicações, o *Google Maps* se destaca ao permitir a visualização de dados e informações geográficas de maneira eficiente e intuitiva através de sua *Application Programming Interface* (API). Atualmente, tal é o seu sucesso, que provavelmente seja a API mais utilizada na Internet (SVENNERBERG, 2010).

5 Sistema operacional Android

O sistema operacional *Android* é um sistema operacional destinado a dispositivos móveis baseado no *kernel* do Linux. Inicialmente desenvolvido pela empresa de sistemas embarcados *Android Inc.* que posteriormente foi adquirida pelo Google no ano de 2005 (MONTEIRO, 2012).

Desde seu lançamento oficial o Google procurou unir forças com outras gigantes do mercado de tecnologia a fim de auxiliar e promover o desenvolvimento da plataforma. Dessa

união surgiu o órgão *Open Handset Alliance* (OHA). Uma cooperação sem fins lucrativos composto por empresas tais como: *HTC, LG, Motorola, Samsung, Sony Ericsson, Intel, Toshiba, Dell, ASUS*, dentre outras, além do próprio Google.

Assim, o objetivo do grupo é definir uma plataforma única e aberta para celulares para deixar os consumidores mais satisfeitos com o produto final. Outro objetivo principal dessa aliança é criar uma plataforma moderna e flexível para o desenvolvimento de aplicações corporativas. O resultado dessa união, como você já deve saber, foi o nascimento do Android (LECHETA, 2013, p.23).

Dentro da estratégia de negócios do Google, a plataforma *Android* destaca-se como peça fundamental, uma vez que permitiria ampliar ainda mais seu público consumidor, principalmente ao facilitar o acesso aos seus serviços ou produtos como o *Gmail, Google Docs, Google Maps*, além de seu mais famoso produto, seu motor de busca (MORIMOTO, 2009).

O Android é a resposta do Google para ocupar esse espaço. Consiste em uma nova plataforma de desenvolvimento para aplicativos móveis, baseada em um sistema operacional Linux, com diversas aplicações já instaladas e, ainda, um ambiente de desenvolvimento bastante poderoso, ousado e flexível (LECHETA, 2013, p.22).

Ao contrário do seu principal concorrente, o *iOS* do *iPhone* (*smartphone* desenvolvido pela *Apple*), o Google procurou adotar uma política de desenvolvimento totalmente diferente, com uma licença menos restritiva e mais flexível que atraindo a atenção de diversos desenvolvedores. Para Morimoto (2009) ao seguir um caminho oposto ao da *Apple*, o Google criou uma plataforma mais convidativa a diversas empresas e novos desenvolvedores.

A figura 2 ilustra a participação nos dias de hoje da plataforma dentro do mercado frente a suas concorrentes.

Segundo relatório do International Data Corporation (IDC) publicado em maio de 2012, o Android possui 59% do mercado de smartphones e soma a quantia de 89,9 milhões de aparelhos distribuídos apenas no primeiro trimestre deste ano, em todo o mundo. Em segundo aparece o *iOS* que o sistema operacional da *Apple iPhone* (MONTEIRO, 2012, p.2).

Participação no mercado

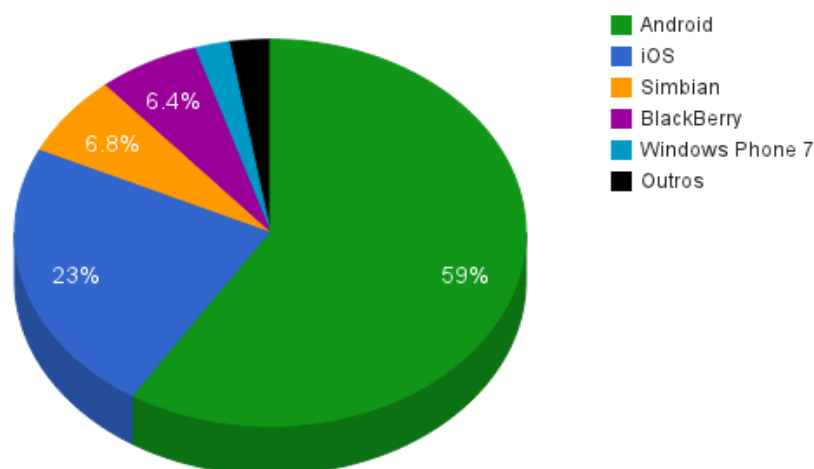


Figura 2: Participação de sistemas operacionais móveis.
Fonte: Monteiro (2012).

Outro fator interessante é o grande número de aplicativos distribuídos pela *Google Play*, que revela o sucesso do Android. Aproximadamente existem 2,5 milhões de aplicativos diferentes (entre pagos a gratuitos) em seu catálogo, dos quais já foram contabilizados mais de 48 bilhões de *downloads* desde sua fundação (DERMARTINI, 2013). Uma plataforma consistente e poderosa que dispõe de um grande mercado aliado a uma política mais interessante para o tema deste trabalho de desenvolvimento se comparados a suas outras concorrentes.

6 Java

O Java é a linguagem de desenvolvimento oficial para *Android*. Nasceu em 1991 pelo projeto *Green Project*, sediado nos laboratórios da *Sun Microsystems*, inicialmente destinado a sistemas embarcados. Entretanto, seu lançamento público só veio a ocorrer anos depois, no ano de 1995, quando foi oficialmente batizada com o nome Java.

Os microprocessadores têm um impacto profundo em dispositivos inteligentes eletrônicos voltados para o consumidor. Reconhecendo isso, a *Sun Microsystems*, em 1991, financiou um projeto de pesquisa corporativo interna com o codinome *Green*, que resultou no desenvolvimento de uma linguagem baseada em C++ que seu criador, James Gosling, chamou de *Oak*. Quando uma equipe da *Sun* visitou uma cafeteria local, o nome Java (cidade de origem de um tipo café importado) foi sugerido; e o nome pegou. (DEITEL & DEITEL, 2005, p.6).

Foi através da explosão da Internet que o Java começou a desfrutar de certa posição de destaque, sendo considerado uma das tecnologias mais importantes do mercado. É facilmente encontrada em milhões de celulares e diversos outros aparelhos tais como *palms*, *set-top boxes*, *blu-rays* etc (KUNG et al, 2012).

O Google *Android* apesar de usar a linguagem Java, é um caso a parte, pois não utiliza a JVM, máquina virtual oficial do Java, e sim a Dalvik, uma implementação do próprio Google, otimizada para trabalhar com dispositivo móveis. Segundo apresenta Monteiro (2009), a simplicidade e popularidade da linguagem Java contribuíram decisivamente para disseminação da plataforma.

7 Android vs. Java ME

O Google preferiu investir em sua própria máquina virtual, denominada de *Dalvik*. Uma implementação cujo principal objetivo é trabalhar de forma otimizada em dispositivo de poucos recursos de *hardware* e memória (LECHETA, 2013).

Mas, então por que não escolher a implementação oficial JME (*Java Micro Edition*), destinada justamente a oferecer suporte a dispositivos móveis? Segundo Maker (2011), a JME apresenta uma série de desvantagens em relação à capacidade e funcionalidades atuais requeridas pelos dispositivos móveis. A falta de gráficos *3D*, *Bluetooth* e *Session Initiation Protocol* (SIP) se torna um empecilho para a utilização da tecnologia.

Há muitas críticas, contudo, ao JCP e ao processo de especificação do Java no geral. Fala-se da demora das novas especificações e que o JCP adota apenas aspectos básicos das tecnologias, para agradar a todos, travando assim a inovação. Mas, afinal, especificações ajudam ou atrapalham? (KUNG et al, 2012, p.14).

Por ser uma plataforma aberta, é permitido que cada empresa possua a liberdade de escrever sua própria implementação da máquina virtual Java. No mercado são encontradas diversas implementações, como: *J9* da IBM, *JRockit* da BEA/Oracle, a *Harmony* da Apache, etc. além da própria *Dalvik* que difere das demais por não seguir o padrão de especificação adotado pelo Java.

8 PHP & MySQL

O *HyperText PreProcessor* (PHP) é uma linguagem de programação livre e interpretada, que tem como principal objetivo a criação de páginas de conteúdo dinâmico. Há diversos projetos construídos a partir da linguagem: desde sistemas de *Content Management System* (CMS), como exemplos *WordPress*, *Drupal*, *Joomla*, até complexos sistemas como de *e-commerce: Magento*. Segundo pesquisa da Netcraft (2013), o PHP seria encontrado em aproximadamente 244 milhões de *websites*, o que corresponde a 39% de *sites* da *Web*, até o ano de 2013.

A PHP é uma linguagem de scripting de uso geral com fonte aberto. O mecanismo interpretador da PHP é escrito na linguagem de programação C, de modo que pode ser utilizado em quase todos os tipos de computadores e sistemas operacionais (ELMASRI & NAVATHE, 2011, p.325).

Para o armazenamento do conjunto de dados, ou seja, os registro das coordenadas obtidas pelo aparelho móvel, é utilizada o *Sistema de Gerenciamento de Banco de dados* (SGDB), MySQL: um banco de dados relacional, comum em aplicações e sistemas PHP. De acordo Câmara & Queiroz (s. d.), o uso de SGBD permite realizar, com maior facilidade, a interligação do banco de dados com o sistema responsável pelo gerenciamento do geoprocessamento.

9 Abordagem REST

As aplicações móveis tem utilizado cada vez mais os serviços remotos comumente chamados *webservices*. O *Android* apresenta boas ferramentas e interessantes recursos para tal finalidade, o que vem atraindo atenção de desenvolvedores, ao facilitar a consistência das informações em um único ponto.

Algo que tem se tornado bastante comum é que as aplicações, tanto web como *mobile* e até mesmo aplicações *desktop*, utilizem serviços remotos disponibilizados por outras aplicações. Os usuários esperam cada vez mais centralização e integração de dados, com o intuito de trazer acesso rápido e comodidade, obtendo as informações literalmente com a ponta dos dedos (MONTEIRO, 2012, p.174).

Grandes empresas tais como: *Twitter*, *Facebook*, *Google*, *Yahoo!*, etc. tem investindo bastante na área, disponibilizando diversas APIs para acesso a seus produtos e serviços, muitas vezes gratuitamente, pela *Web*.

O valor de uma aplicação de mobilidade é medido justamente por sua capacidade de fazer diversos serviços interagirem entre si. Para atender a essa necessidade, é preciso desenvolver aplicações que se comuniquem com outras já existentes pela internet (LECHETA, 2013, p.560).

Neste contexto, o protocolo *HyperText Transfer Protocol* (HTTP) tem desempenhado um importante papel em conjunto com a abordagem *Representational State Transfer* (REST).

O HTTP é o protocolo padrão para troca de informações na *Web*, ao permitir que sistemas possivelmente com linguagens e pilhas de tecnologias diferentes possam se comunicar (KUNG et al, 2009).

Já o REST ao trabalhar fundamentalmente sobre o protocolo HTTP se torna o caminho mais fácil e cômodo encontrado por muitas empresas ao facilitar o acesso a seus dados e informações. Hoje, tem se tornando o padrão comum quando se trata de serviços oferecidos pela *Web* (MONTEIRO, 2012).

Segundo Tilkov (2008) ao aderir a princípios REST em sua aplicação, automaticamente poderá explorar a arquitetura da *Web* em seu próprio favor, favorecendo-se das vantagens e benefícios inerentes arquitetura da Internet.

10 O Sistema

O sistema tem como finalidade a construção de um módulo cliente, mais especificamente, um aplicativo Android, que faz uso de um sistema *Web* (módulo servidor) para compartilhamento de informações referentes à localização do aparelho móvel.

A *Arquitetura Cliente-Servidor* é aplicada na estrutura geral do projeto ao oferecer bons *trade-offs*. Conforme diz Deitel & Deitel (2005), nesta arquitetura de sistema o cliente solicita uma ação ao servidor que após realizada informa ao cliente sobre os resultados processados, encaixando-se bem ao funcionamento do protótipo proposto.

Aplicado a esse esquema ao projeto, é possível observar a existência de dois módulos distintos:

- **Módulo Cliente:** constitui-se da aplicação móvel propriamente dita. Compõem-se de um aplicativo capaz de utilizar recursos de geolocalização presentes no sistema operacional *Android*;
- **Módulo Servidor:** é um sistema de rastreamento Web que servirá como repositório de registro de localização dos dispositivos Android. Sendo cuja principal função é permitir a um administrador acompanhar, através da visualização de mapas, a localização geográfica, em tempo real, de respectivos aparelhos móveis.

A figura 2 apresenta a seguir um esquema em relação ao funcionamento do projeto:

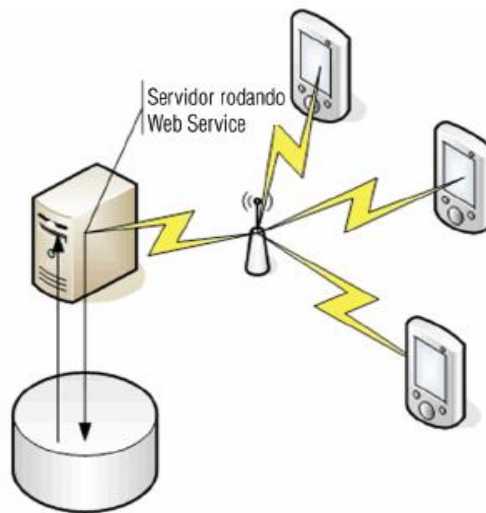


Figura 2: Arquitetura do sistema.
Fonte: JÚNIOR (2013).
Nota: Adaptado pelo autor.

11 O cliente

O módulo cliente é destinado aos usuários *Android*. É responsável por alimentar o banco de dados do servidor remoto com a localização atual do aparelho. Foi construído a partir dos recursos da linguagem Java para plataforma *Android* através da API de geolocalização sob a abordagem REST.

Durante testes e simulações foram utilizados dois métodos distintos, ou seja, através do uso de um aparelho físico e através de uma simulação do emulador em um ambiente virtual sobre a plataforma de desenvolvimento para *Android*.

O aparelho físico utilizado foi um modelo *Samsung Galaxy Ace S5830* com a versão *Android 2.3.3 (Gingerbread)*. A figura 3 apresenta o aparelho utilizado. Durante o processo de realização de testes de funcionamento, o sistema se mostrou satisfatório com resultados bastante positivos.

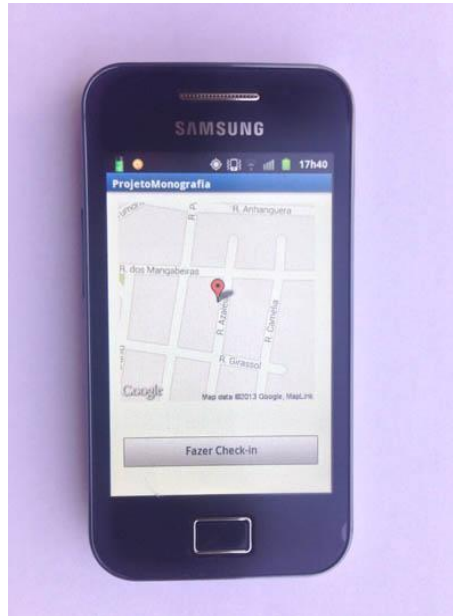


Figura 3: Teste em dispositivo físico.
Fonte: Elaboração própria (2013).

Vale ressaltar que o dispositivo em questão, apresenta o padrão de telefonia móvel GSM, através da operadora TIM com acesso a Internet. O *Global System for Mobile* (GSM) é um padrão de telefonia unificado surgido em países da Europa que tem contribuído como um dos principais fatores da massificação de celulares e *smartphones* (Morimoto, 2009). Segundo PCMagazine Brasil (2012), todas operadoras atualmente no Brasil utilizam o GSM.

Na figura 4 é apresentado o mesmo processo, porém desta vez utilizando o emulador para receber as coordenadas do georeferenciamento obtido.

O emulador do *Android* é capaz de simular o comportamento bem próximo de um aparelho móvel real, contando com algumas restrições, que para os fins deste experimento, não impactam decisivamente nos resultados dos testes obtidos.

Para a facilitação da visualização do mapa na figura 4 é utilizado um *WebView*, componente responsável por renderizar elementos e páginas *Web* no *Android*.

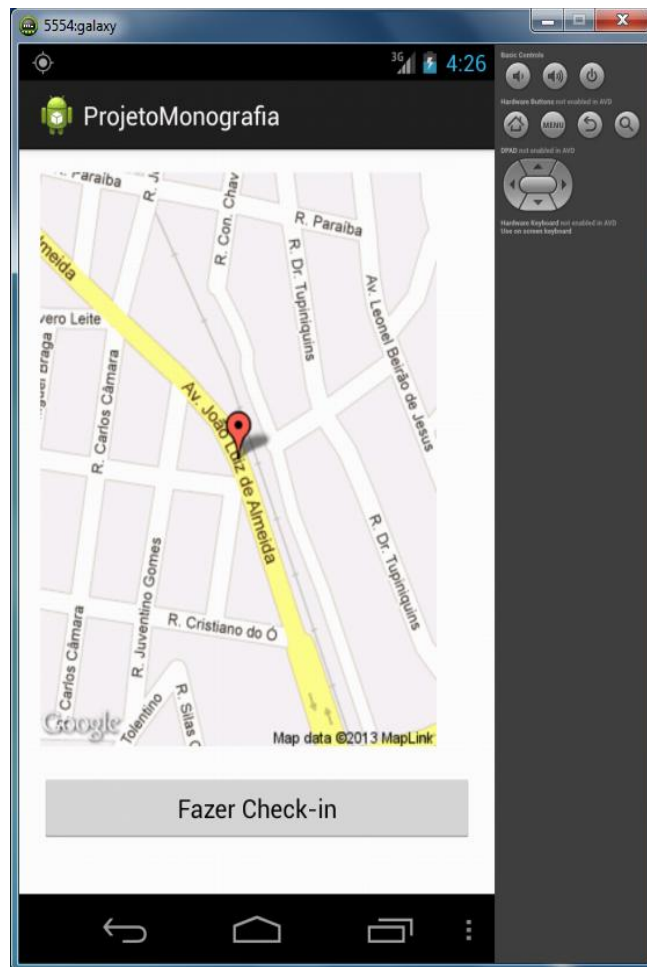


Figura 4: Emulador Android com localização encontrada.
 Fonte: Elaboração própria (2013).

O Google oferece o emulador a partir de kit de desenvolvimento: *Android SDK*. O kit é oferecido gratuitamente pela Google. O SDK é composto pelo emulador, além de várias outras ferramentas utilitárias ao desenvolvimento de aplicações para *Android*.

A liberação, por parte dos fabricantes, de um kit de desenvolvimento de software (SDK) para suas plataformas e a criação de lojas para a distribuição de aplicativos viabilizou a abertura deste mercado para praticamente qualquer empresa ou desenvolvedor, criando assim novas oportunidades de negócio (MONTEIRO, 2012, p.1).

O *Java Development Kit* (JDK) utilizado foi o de versão 7 com o *plugin* do *Android Development Tools* (ADT) sobre a *Integrated Development Environment* (IDE) *Eclipse*. Uma IDE de uso geral construída sobre a linguagem Java. A extensão ADT é capaz de preparar e executar todas as ferramentas e rotinas necessárias ao desenvolvimento *Android*, provendo todas as facilidades de criação e codificação de um projeto Java (MONTEIRO, 2012).

Para utilização das funcionalidades de geolocalização e comunicação via HTTP no *Android* foram necessárias as seguintes permissões: *android.permission.INTERNET* e

android.permission.ACCESS_FINE_LOCATION dentro do *AndroidManifest.xml* (um tipo de arquivo de configurações do *Android*).

Ao declarar a permissão "*ACCESS_FINE_LOCATION*", além de autorizar a utilização do GPS, também permite-se o utilização do *Network Location Provider* (NFC), uma forma de busca pela localização através de sinais *Wi-Fi* e da rede de celular.

O NFS pode apresentar certas vantagens em relação ao GPS, o que o torna também uma ferramenta bastante interessante, com um menor consumo de bateria e mais rápido no processo de obtenção dos dados geográficos. Entretanto, apesar destas vantagens o GPS ainda prevalece como principal serviço de georeferenciamento, devido principalmente a seu alto nível precisão em comparação ao NFC.

Outra forma de obter a localização é através do *Network Location Provider* que utiliza os sinais da rede de celular e WI-FI para determinar a localização do usuário. Apesar de ser menos precisa, esta forma consome menos bateria e obtém resultados de localização com mais rapidez além de funcionar tanto em ambientes abertos como em recintos fechados (MONTEIRO, 2012, p.244).

Na figura 5 é apresentado o trecho de código referente à utilização dos recursos de geolocalização do Android. Nele é possível observar a API de localização. Quando habilitado é permitindo ao aparelho atualizar suas próprias coordenadas automaticamente.

```
// Tempo mínimo para atualizações de localização.
private static final long TEMPO_ATUALIZACAO =1000*60*10;// 10 minutos

// Distância mínima para haver atualizações.
private static final float MIN_DISTANCIA =10;// 10 metros

private LocationManager locationManager;

/**
 * Quando houver alguma mudança de localização, obter coordenadas e atualizar o
 * mapa.
 *
 * @param location
 * @return void
 */
@Override
public void onLocationChanged(Location location) {
    double lat = location.getLatitude();
    double lng = location.getLongitude();
    atualizaMapa(lat, lng);
}

@Override
public void onProviderDisabled(String provider) {
}

@Override
public void onProviderEnabled(String provider) {
}
```

```

}

@Override
public void onStatusChanged(String provider,int status,Bundle extras) {
}

/**
 * Carrega funções de localização doAndroid.
 *
 * @return void
 */
public void liga() {
    // Habilita GPS
    if(locationManager.isProviderEnabled(LocationManager.GPS_PROVIDER)) {
        locationManager.requestLocationUpdates(
            LocationManager.GPS_PROVIDER, TEMPO_ATUALIZACAO,
            MIN_DISTANCIA,this);
    }
    // Habilita 'NetWork Location Provider'
    if(locationManager.isProviderEnabled(LocationManager.NETWORK_PROVIDER)) {
        locationManager.requestLocationUpdates(
            LocationManager.NETWORK_PROVIDER, TEMPO_ATUALIZACAO,
            MIN_DISTANCIA,this);
    }

    Log.w("Rastreamento","Liga rastreamento.");
}

```

Figura 5: Trecho de código referente ao recurso de geolocalização do Android.
Fonte: Elaboração própria (2014).

A classe *android.location.LocationManager* é o principal elemento neste cenário, uma vez que oferece as ferramentas e recursos necessários para obtenção da localização do aparelho. A partir de um “*limite de tempo*” e “*raio de localização*”, como se observado na figura 5, pode-se encontrar as coordenadas do aparelho.

Optou-se por utilizar uma atualização automática de 10 minutos com uma distância mínima de 10 metros, através da atualização da posição em *android.location.LocationListener*, injetada como último parâmetro do método *requestLocationUpdates*.

No trecho de código apresentado na figura 5, é possível também observar a utilização do NFC, que neste caso é mais utilizado mais como um *fallback*, caso haja alguma instabilidade por parte dos recursos do GPS no momento em que for requisitado o serviço de localização.

A comunicação entre o aplicativo e o servidor, já com as coordenadas obtidas, se faz por meio da API de conexão HTTP pela API da linguagem Java. A figura 6 apresenta o trecho do código responsável pela comunicação HTTP e constitui-se como parte integrante

fundamental para o sistema de rastreamento, ao fornecer o meio de comunicação entre diferentes plataformas de tecnologia: o aplicativo *Android* e o sistema *Web*.

```
private static URL baseUrl =null;

private static HttpURLConnection http =null;

private static StringBuilder resultado =null;

/**
 * Método que envia as coordenadas ao servidor.
 *
 * @param url
 * @param parametros
 * @return
 */
public static String em(String url,Map<String,String> parametros) {
    resultado =new StringBuilder();

    try {
        baseUrl =new URL(url);
        http =(HttpURLConnection) baseUrl.openConnection();

        // http query builder
        String params="";
        Set<String> paramsKey = parametros.keySet();

        for(String key : paramsKey) {
            params += ("&" + key +"="+URLEncoder.encode(
                parametros.get(key),"UTF-8"));
        }

        if(params.length()>0) {
            params=params.substring(1);
        }

        http.setDoInput(true);
        http.setDoOutput(true);

        // http headers
        http.setRequestMethod("POST");
        http.setRequestProperty("Content-Type",
            "application/x-www-form-urlencoded");
        http.setRequestProperty("Content-Length",
            Integer.toString(params.getBytes().length));

        // enviando dados
        OutputStreamout= http.getOutputStream();
        out.write(params.getBytes());
        out.flush();
        out.close();
    }
}
```

Figura 6: Trecho de código responsável por enviar as coordenadas ao servidor.
Fonte: Elaboração própria (2014).

Por fim, na figura 7 é ilustrado um esquemático do projeto aqui apresentado forma de um Diagrama de Classes, em que é possível observar a relação das classes que compõem o módulo do cliente e suas interações.

Figura 7: Diagrama de Classes.
Fonte: Elaboração própria (2014).

Com as coordenadas de localização, o mapa é habilitado com a possibilidade de confirmação através do botão de envio do aplicativo, contendo a posição geográfica atual do dispositivo. Enviados os dados da localização é recebida uma mensagem do servidor a respeito do processo de acesso ao banco de dados do mesmo. A partir deste momento, o administrador, representado pelo módulo servidor, já contará com a localização dos usuários do módulo cliente em seu banco de dados. O código do módulo cliente encontra-se hospedado em: <<https://github.com/tsprates/projeto-cliente>>.

12 O servidor

O servidor é construído sobre a plataforma PHP e representa o *webservice* proposto. Constitui-se uma parte importante da aplicação capaz de gravar as coordenadas fornecidas pelo módulo cliente através da arquitetura REST. Há a troca de informações entre o cliente, o

aplicativo Android e o servidor. O endereço de teste se encontra no endereço: <http://tracking.comoj.com/postdata.php>.

A figura 8 mostra o esquema de um diagrama da tabela do banco de dados responsável pelo registro e consulta das coordenadas enviadas pelo dispositivo móvel ao servidor.



Figura 8: Modelo da tabela de coordenadas do banco de dados do *webservice*.
Fonte: Autoria Própria (2014).

Já na figura 9 é apresentado uma lista de registros de testes e simulações. É apresentada a posição dos dispositivos móveis através dos dados latitude, longitude e data de *check-in*. Estas coordenadas são registros enviados pelos clientes (aparelho) ao sistema.

| Data | Tel. | Lat. | Lng. | Opções |
|---------------------|-----------------|---------------------|--------------------|----------|
| 17:47:55 24/05/2013 | 352346052552396 | -16.7387264 | -43.8738019 | Ver mapa |
| 12:32:22 24/05/2013 | 000000000000000 | -16.736865 | -43.8652466666666 | Ver mapa |
| 07:25:26 24/05/2013 | 357957040388662 | -16.71874305233593 | -43.87951707467437 | Ver mapa |
| 15:59:04 23/05/2013 | 352346052552396 | -16.73873204153588 | -43.87387563666433 | Ver mapa |
| 15:58:31 23/05/2013 | 352346052552396 | -16.738757549698004 | -43.87378541174369 | Ver mapa |
| 15:51:46 23/05/2013 | 357957040388662 | -16.7386755 | -43.8737104 | Ver mapa |
| 15:51:10 23/05/2013 | 357957040388662 | -16.7386823 | -43.8737135 | Ver mapa |
| 15:44:03 23/05/2013 | 000000000000000 | -16.738728 | -43.8737888 | Ver mapa |
| 15:42:34 23/05/2013 | 000000000000000 | -16.73871004488319 | -43.87376650236547 | Ver mapa |
| 15:35:04 23/05/2013 | 000000000000000 | -16.738651604606214 | -43.87374404238045 | Ver mapa |

Figura 9: Listagem de todos os pontos de localização cadastrados.
Fonte: Elaboração própria (2013).

Vale ressaltar a utilização, para fins didáticos neste trabalho, do *International Mobile Equipment Identity* (IMEI) como identificador para registro do dispositivo no banco de dados.

Este é um identificador composto por uma sequência de 15 dígitos que é exclusiva para cada aparelho. Para o emulador esse valor é composto de “000000000000000”.

A figura 10 representa a visualização de uma das localizações já previamente cadastradas no sistema. A visualização de usuários é feita através de recursos disponibilizados pela API do *Google Maps*. Encontra-se hospedado no endereço de teste <<http://tracking.comoj.com>>.

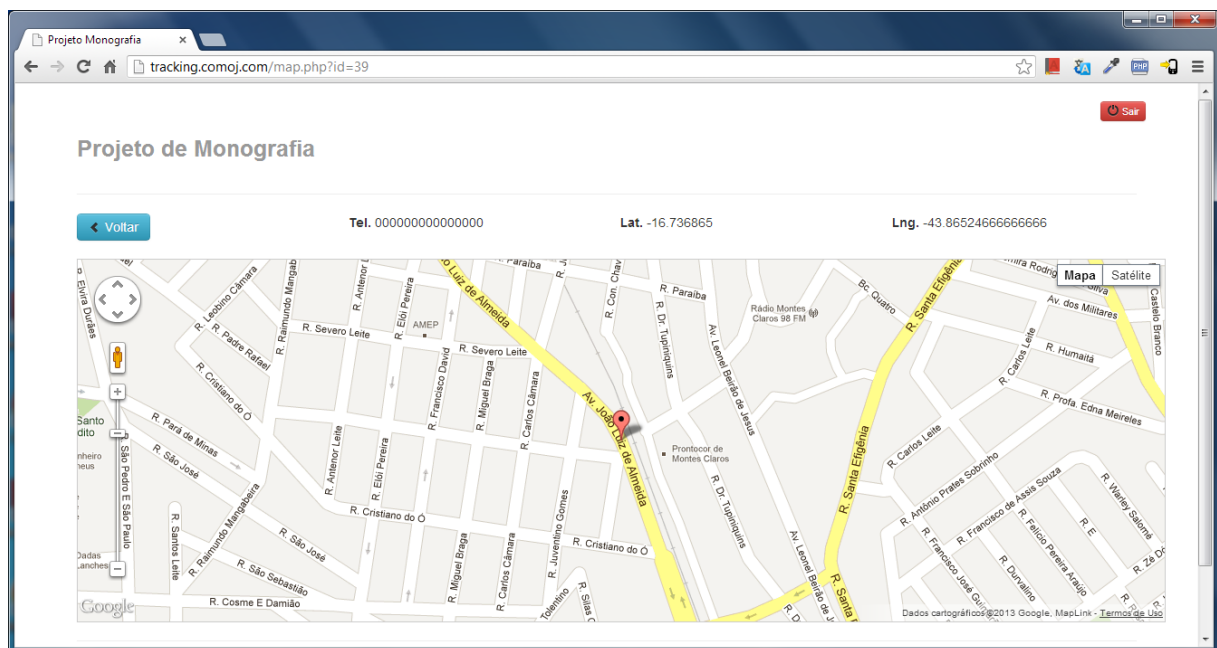


Figura 10: Visualização de um ponto de localização cadastrada.
Fonte: Elaboração Própria (2013).

A figura 11 apresenta o panorama de uso do sistema na formato de Diagrama de Casos de Uso. Sendo possível identificar seu funcionamento através do ponto de vista do usuário ao desenvolvedor. Conforme revela Neto & Sampaio, o Diagrama de Casos de Uso descreve um cenário capaz de mostrar todas as funcionalidades propostas pelo sistema sobre o ponto de vista do usuário, com o objetivo de auxiliar a comunicação entre os analistas e o seu cliente.

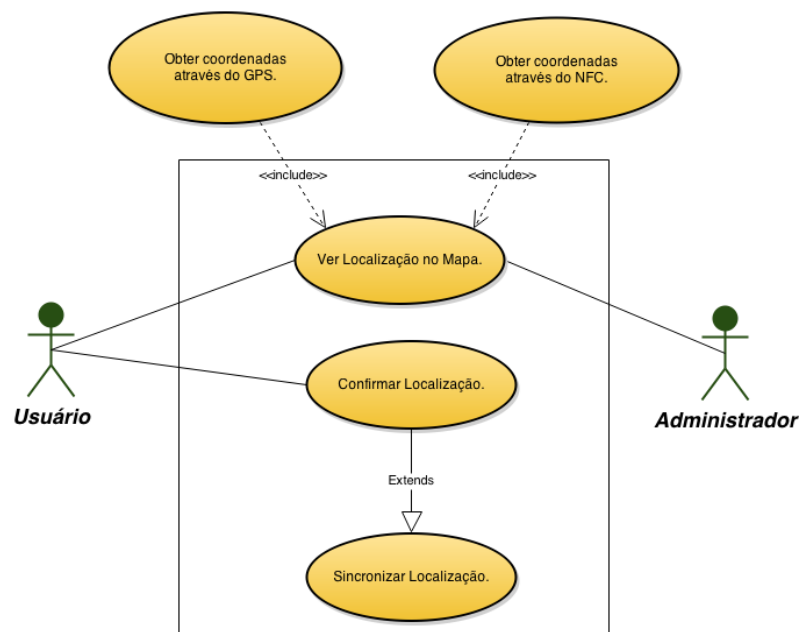


Figura 11: Diagram de Casos de Uso.
 Fonte: Elaboração Própria (2013).

Por fim, o que é possível inferir é que os resultados obtidos através do experimento aqui apresentado foram, de forma geral, bastante esclarecedores. A ferramenta apresentou uma boa cobertura dos requisitos aqui propostos e atendeu o objetivo inicial de análise. Contudo, há ainda a necessidade de estudo e verificação de outros aspectos fundamentalmente importantes para que a aplicação possa se concretizar em um produto acabado, com alguns pontos que ainda requerem maior cuidado e atenção. O código do módulo servidor encontra-se hospedado em: <<https://github.com/tsprates/projeto-servidor>>.

13 Considerações finais

O presente trabalho buscou apresentar, de forma clara e objetiva, um estudo relativo as motivações e implicações acerca do desenvolvimento de um protótipo de um sistema de rastreamento *Web* para dispositivos móveis, focado na plataforma *Android*.

Durante o respectivo estudo, foi possível observar que a plataforma móvel *Android* se destaca em comparação a outras por oferecer uma política interessante de desenvolvimento de aplicações destinadas ao geoprocessamento, que em conjunto com as tecnologia para PHP e MySQL, permitiram a construção de um Sistemas de Informação com suporte geográfico na resolução de problemas que envolvam rastreamento e localização.

Observou-se também que esta é uma área ainda pouco explorada, com amplas oportunidades, carecendo de mais trabalhos a respeito do tema, uma vez que o geoprocessamento, pautado dentro da metodologia *open-source* e tecnologias móveis, poderia contribuir nas questões referentes de geolocalização e rastreamento.

De maneira geral os resultados obtidos atenderam aos objetivos iniciais desta pesquisa. Contudo, alguns pontos ainda requerem maior atenção, principalmente a partir de análises mais aprofundadas acerca do tema e do respectivo processo de desenvolvimento.

Para o futuro, espera-se que a partir dos resultados obtidos, o trabalho possa evoluir em um produto final, capaz de auxiliar usuários no gerenciamento de dispositivos móveis à otimização de processos e gerência da empresa para a tomada de decisões.

Referências

BRITO, Jorge L. S.; ROSA, Roberto. **Introdução ao Geoprocessamento: Sistema de Informação Geográfica**. Uberlândia: 1996.

CÂMARA, Gilberto; DAVIS, Clodoveu. **Introdução**. Disponível em: <<http://www.dpi.inpe.br/gilberto/livro/introd/cap1-introducao.pdf>>. Acesso em: 11 de Agosto de 2013.

CÂMARA, Gilberto; MONTEIRO, Antônio Miguel Vieira. **Conceitos Básicos em Ciência da Geoinformação**. Disponível em: <<http://www.dpi.inpe.br/gilberto/livro/introd/cap2-conceitos.pdf>>. Acesso em: 11 de Agosto de 2013.

CÂMARA, Gilberto; QUEIROZ, Gilberto Ribeiro de. **Arquitetura de Sistemas de Informação Geográfica**. Disponível em: <<http://www.dpi.inpe.br/gilberto/livro/introd/cap3-arquitetura.pdf>>. Acesso em: 11 de Agosto de 2013.

DEITEL, Harvey M.; DEITEL, Paul J. **Java: como programar**. São Paulo: Prentice-Hall, 2005.

DERMATINI, Felipe. **48 bilhões de aplicativos já foram baixados na Google Play**. Disponível em: <<http://www.tecmundo.com.br/google-i-o-2013/39755-48-bilhoes-de-aplicativos-ja-foram-baixados-na-google-play.htm>>. Acesso em: 16 de Maio de 2013.

ELMASRI, Ramez; NAVATHE, Shamkant B. **Sistemas de banco de dados**. 6 ed. São Paulo: Pearson Addison Wesley, 2011.

KUNG, Fábio *et al.* **Introdução á Arquitetura e Design de Software: Uma visão sobre a plataforma Java**. Rio de Janeiro: Elsevier, 2012.

HUERTA, Eduardo; MANGIATERRA, Aldo; NOGUERA, Gustavo. **GPS: posicionamiento satelital**. Rosario: UNR Editora, 2005.

JÚNIOR, Edward E. Martins. **Artigo publicado na Webmobile 8-Desenvolvendo uma aplicação móvel com .NET**. Disponível em: <<http://http://www.devmedia.com.br/artigo-publicado-na-webmobile-8-desenvolvendo-uma-aplicacao-movel-com-net/3086>>. Acesso em: 2 de Setembro de 2013.

LECHETA, Ricardo. **Google Android: Aprenda a criar aplicações para dispositivos móveis com Android SDK**. 3. ed. São Paulo: Novatec Editora, 2013.

MAKER, Frank. **Android Dalvik Virtual Machine**. Disponível em: <<http://jaxenter.com/android-dalvik-virtual-machine-35498.html>>. Acesso em: 10 de Maio de 2013.

MONTEIRO, João Bosco. **Google Android: Crie Aplicações para celulares e tablets**. São Paulo: Casa do Código. 2012

MORIMOTO, Carlos E. **Smartphones: Guia Prático**. Porto Alegre: Sul Editores, 2009.

NETCRAFT. **PHP just grows & grows**. Disponível em: <<http://news.netcraft.com/archives/2013/01/31/php-just-grows-grows.html>>. Acesso em: 05 de setembro de 2014

NETO, Eloi Rocha; SAMPAIO, Marcus Costa. **Material sobre UML**. Disponível em: <<http://www.dsc.ufcg.edu.br/~jacques/cursos/map/html/uml/>>. Acesso em: 1 de Julho de 2014.

PCMAGAZINE BRASIL. **CDMA vs GSM: Qual a diferença?** Disponível em: <<http://www.pcmag.com.br/noticias/cdma-vs-gsm-qual-a-diferenca/>>. Acesso em: 1 de Outubro de 2014.

SVENNERBERG, Gabriel. **Beginning Google Maps API 3**. New York: Apress, 2010.

TECHGEO. **Você sabe o que foi disponibilidade seletiva (SA)?** Disponível em: <<http://site.techgeo.com.br/2013/08/nova-noticia/>>. Acesso em: 1 de Julho de 2014.

TILKOV, Stefan. **Uma rápida introdução ao REST**. Disponível em: <<http://www.infoq.com/br/articles/rest-introduction>>. Acesso em: 1 de Setembro de 2014.