

**UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ
COORDENAÇÃO DE LICENCIATURA EM INFORMÁTICA
DESENVOLVIMENTO DE SISTEMAS PARA INTERNET E DISPOSITIVOS
MÓVEIS**

EVERTO FÁBIO DA SILVA MACHADO

**DESENVOLVIMENTO DE SISTEMAS DE GEOLOCALIZAÇÃO E
RASTREAMENTO PARA A PLATAFORMA ANDROID - COMPASS**

MONOGRAFIA DE ESPECIALIZAÇÃO

FRANCISCO BELTRÃO

2015

EVERTO FABIO DA SILVA MACHADO

**SISTEMA DE GEOLOCALIZAÇÃO E RASTREAMENTO PARA A
PLATAFORMA ANDROID - COMPASS**

Monografia de especialização apresentada a Coordenação de Licenciatura em Informática da Universidade Tecnológica Federal do Paraná como requisito parcial para obtenção do título de “Especialista em Desenvolvimento de Sistemas para Internet e Dispositivos Móveis”.

Orientador: Prof. MSc. Robison Cris Brito

FRANCISCO BELTRÃO

2015



Ministério da Educação
Universidade Tecnológica Federal do Paraná
Campus Francisco Beltrão
Diretoria de Francisco Beltrão
Coordenação de Licenciatura em Informática
Desenvolvimento de Sistemas para Informática e Dispositivos
Móveis



TERMO DE APROVAÇÃO

DESENVOLVIMENTO DE SISTEMA PARA GEOLOCALIZAÇÃO E RASTREAMENTO

por

EVERTO FÁBIO DA SILVA MACHADO

Esta Monografia de Especialização foi apresentada em 24 de fevereiro de 2015 como requisito parcial para a obtenção do título de Especialista em Desenvolvimento de Sistemas de Internet e Dispositivos Móveis. O(a) candidato(a) foi arguido pela Banca Examinadora composta pelos professores abaixo assinados. Após deliberação, a Banca Examinadora considerou o trabalho aprovado.

Robison Cris Brito
Prof. Orientador

Danilo Giacobbo
Membro convidado

Venécio Pegorini
Membro convidado

Dedico meu TCC a minha esposa, que me impulsiona a buscar vida nova a cada dia, agradeço por ter se privado de minha presença em função dos estudos, permitindo a mim esta oportunidade.

AGRADECIMENTOS

Agradeço aos meus pais por todo o apoio, incentivos que sempre me fortalece e me fez capaz de mais uma importante conquista.

Agradeço ao professor Robison, que mais do que um grande mestre, foi um grande motivador e um paciente orientador, talvez sem seu apoio este trabalho não seria possível e ao qual serei sempre grato.

À Jaqueline, que é uma grande companheira com quem amo compartilhar a vida. Obrigado pelo apoio e compreensão.

A todos aqueles que, de alguma forma, fizeram parte desta realização.

Eu só vim até aqui e apenas o amanhã
guia o meu caminho (Dave Mathews
Band)

RESUMO

A tecnologia embutida nos *smartphones* com sistema operacional Android é extremamente evoluída, o que inclui sistema de posicionamento global que permite a localização geográfica do aparelho a nível global. O presente projeto, por meio de uma pesquisa bibliográfica e um estudo de caso, pretende utilizar tais recursos para implementar o mapeamento de todos os clientes de uma empresa de logística, com objetivo de construir um banco de dados capaz de fornecer meios mais eficazes para realizar o controle das visitas dos representantes comerciais.

Palavras-chave: Android. Geolocalização. Mapeamento.

ABSTRACT

The embedded technology in smart phones with Android operating system is extremely evolved, which includes global positioning system that allows the geographical location of the device. This project, through a literature review and a case study, you want to use these resources to implement the mapping of all customers of a logistics company, in order to build a database able to provide smarter ways to perform control of visits by sales representatives.

Keywords: Android. Geolocation. Mapping.

LISTA DE ILUSTRAÇÕES

| | |
|---|----|
| Figura 1 – Latitude em graus a partir do Equador | 19 |
| Figura 2 - Longitude em graus a partir do Equador | 20 |
| Figura 3- Disposição dos satélites de GPS | 21 |
| Figura 4 - Modelo de triangulação utilizado pelo sistema de GPS | 22 |
| Figura 5 - Comunicação e Ferramentas | 25 |
| Figura 6 - IDEs mais utilizadas por desenvolvedores Java | 26 |
| Figura 7 - Tela de Desenvolvimento Eclipse e <i>plug-in</i> ADT | 27 |
| Figura 8 - Modelo Sequencial Linear | 31 |
| Figura 9 - Diagrama de Casos de Uso | 33 |
| Figura 10 - Diagrama de Classes | 34 |
| Figura 11 - DER Projeto Android | 35 |
| Figura 12 – Comunicação entre Servidor e Dispositivo | 36 |
| Figura 13 - Menu Principal | 37 |
| Figura 14 - Localização No Mapa | 38 |
| Figura 15 - Consulta de Rota | 40 |
| Figura 16 - Caminho Percorrido | 41 |

LISTAGEM DE CÓDIGO FONTE

| | | |
|--------------|--|----|
| Listagem 1- | Leiaute XML para construção de Mapa do GoogleMaps..... | 42 |
| Listagem 2- | Activity com método onCreate para criação do mapa..... | 42 |
| Listagem 3- | Código com instancia de LocationManagem, para localização por GPS..... | 43 |
| Listagem 4- | Código que grava a localização no bando de dados..... | 44 |
| Listagem 5- | Exibição da rota no mapa com marcadores..... | 44 |
| Listagem 6- | Exibição no mapa do trajeto percorrido..... | 45 |
| Listagem 7- | Consulta da geolocalização pelo endereço..... | 46 |
| Listagem 8- | Formato das mensagens no padrão JSON..... | 46 |
| Listagem 9- | Modelo aplicação servidor em Socket..... | 47 |
| Listagem 10- | Consulta cliente por meio de Socket..... | 49 |

LISTA DE SIGLAS

| | |
|------|---|
| ADT | Android Developer Tools |
| API | Application Programming Interface |
| DDL | Data Definition Language |
| DML | Data Manipulation Language |
| ERP | Enterprise Resource Planning |
| GPS | Global Positioning System |
| IDE | Integrated Development Environment |
| JSON | JavaScript Notation Object |
| LBS | Location-Based Services |
| SDK | Software Development Kit |
| SGBD | Sistema Gerenciador de Banco de Dados |
| UI | User Interface |
| UML | Unified Modeling Language |
| USB | Universal Serial Bus |
| XML | Extensible Markup Language |
| IBGE | Instituto Brasileiro de Geografia e Estatística |

SUMÁRIO

| | |
|--|-----------|
| 1 INTRODUÇÃO | 14 |
| 1.1 PROBLEMA DE PESQUISA | 14 |
| 1.3 JUSTIFICATIVA | 15 |
| 1.4 OBJETIVOS | 16 |
| 1.4.1 Objetivo Geral | 16 |
| 1.4.2 Objetivos Específicos | 16 |
| 1.5 METODOLOGIA..... | 17 |
| 1.6 ESTRUTURA DO TABALHO | 17 |
| 2 FUNDAMENTAÇÃO TEÓRICA | 18 |
| 2.1 LOCALIZAÇÃO GEOGRÁFICA | 18 |
| 2.1.1 Coordenadas Geográficas | 19 |
| 2.1.2 Latitude | 19 |
| 2.1.3 Longitude | 20 |
| 2.2 FUNCIONAMENTO DO GPS | 20 |
| 2.2.1 Conversão dos Geocódigos..... | 22 |
| 2.3 GEOLOCALIZAÇÃO EM APARELHOS ANDROID | 23 |
| 3 MATERIAIS E MÉTODOS | 24 |
| 3.1 MATERIAIS | 24 |
| 3.1.1 Eclipse..... | 25 |
| 3.1.2 Android SDK..... | 27 |
| 3.1.4 PostgreSQL..... | 28 |
| 3.1.5 JSON e GSON | 29 |
| 3.1.6 Google Maps | 29 |
| 3.2 MÉTODOS | 30 |
| 4 RESULTADOS | 33 |
| 4.1 ANÁLISE DO PROJETO | 33 |
| 4.2 APRESENTAÇÃO DO SISTEMA COMPASS | 36 |
| 4.2.1 Localização e Armazenamento do percurso | 37 |
| 4.2.2 Lista de Clientes e Lista de Rotas | 39 |
| 4.2.3 Consulta do percurso percorrido no dia..... | 40 |
| 4.2.4 Atualizar localização e sincronizar dados..... | 41 |
| 4.3 DESENVOLVIMENTO DO APLICATIVO | 42 |
| 4.3.1 Leitura da Localização através do GPS | 43 |
| 4.3.2 Marcadores no Mapa..... | 45 |
| 4.3.3 Atualizar Localização pelo Endereço | 47 |
| 4.3.4 Comunicação <i>Socket</i> e GSON | 47 |
| 5 CONSIDERAÇÕES FINAIS | 50 |

| | |
|-------------------------|-----------|
| REFERÊNCIAS..... | 52 |
|-------------------------|-----------|

1 INTRODUÇÃO

A expansão das tecnologias de comunicação móvel é evidente, como novas redes para acesso a dados e *smartphones* com novos recursos, assim como os inúmeros recursos embarcados nestes novos aparelhos. Estas tecnologias permitem a criação de sistemas avançados para controle logístico para empresas de representação comercial e transporte, já que estes aparelhos disponibilizam acesso a grande rede mundial de computadores – Internet – na maioria dos locais, assim como possuem recursos de auto-localização, como o GPS (*Global Position System*).

Conforme Globo, o Android está entre os sistemas operacionais mais utilizados em *smartphones* no mundo, e inclui importantes recursos, dentre elas destacam-se o *touchscreen* (Tela sensível ao toque), sensores, GPS, acesso a rede Internet por meio das operadoras, entre outros recursos que os tornam muito mais do que aparelhos que realizam e recebem ligações, tornando-os verdadeiros computadores portáteis.

Desta forma, estes dispositivos estão sendo usados em projetos cada vez mais complexos, como o proposto pelo presente trabalho, que visa fornecer informações sobre a localização de funcionários de empresa de vendas e representação comercial, visando um avanço significativo em rotinas de controle de visitas aos clientes e logística de transportes. Pretende também manter a localização geográfica dos clientes, desta forma será possível controlar as visitas realizadas pelos seus vendedores, identificando a data e hora de cada visita, conseqüentemente o cumprimento das rotas dos vendedores.

1.1 PROBLEMA DE PESQUISA

Hoje em dia, para registrar a distância percorrida pelos representantes comerciais de uma empresa se faz uso de tabelas com distância, sendo que este valor não é exato, pois depende muito da localização onde está o estabelecimento comercial que se deseja visitar. Em cidades grandes, a variação da distância real com a distância estimada é bastante grande, o que compromete a geração de dados para tomada de decisões.

Outro problema encontrado é que os representantes comerciais não podem ser monitorados, para saber se realmente visitou todos os estabelecimentos comerciais previstos, quanto tempo ficou em cada estabelecimento (medir a qualidade da visita), assim como se houve desvios de rota.

Todos os representantes comerciais possuem aparelhos celulares para comunicação com a empresa, sendo que muitos são *smartphones* com recursos de acesso a rede e localização via GPS, problemas como os citados podem ser abordados com o desenvolvimento de um software específico para os dispositivos móveis que controlam a localização de cada representante comercial.

1.3 JUSTIFICATIVA

Nos dias de hoje, o foco de todas as empresas, em especial, as que dependem de representantes comerciais é aumentar suas vendas, reduzindo sempre que possível os seus custos. Com o acesso a informações mais confiáveis e seu correto processamento, abre-se um leque de possibilidades que podem levar a esta evolução.

Com o controle eficiente sobre a equipe de vendas, a empresa é capaz de melhorar a qualidade de atendimento, sendo esta uma qualidade essencial para a competitividade e crescimento, além de criar estratégias para otimizar as rotas, os tempos de visita, atendendo com isso cada vez mais clientes e sabendo quais são os locais e/ou representantes comerciais que geram mais lucro para a empresa.

Desta forma, o presente trabalho visa desenvolver um sistema de localização, integrado com servidor *socket*, ERP (*Enterprise Resource Planning*) e banco de dados, que irão fornecer dados dos clientes e rotas a serem seguidas. As informações geradas no aparelho serão enviadas para o servidor *socket*, onde serão gravadas no banco de dados do ERP.

Com tais informações torna-se possível um melhor atendimento ao cliente, cálculo mais justo do valor de frete da mercadoria, planejamento de rotas mais inteligentes e conseqüentemente redução de custos no serviço de transporte.

1.4 OBJETIVOS

Nesta seção serão apresentados os objetivos geral e específico do trabalho.

1.4.1 Objetivo Geral

Desenvolver um protótipo de aplicativo Android que permita realizar o registro da posição dos representantes comerciais da empresa usando dispositivos móveis.

1.4.2 Objetivos Específicos

Como objetivo específico este trabalho pretende-se:

- Com a utilização de um servidor *socket*, integrar um cadastro de clientes com o aplicativo a ser desenvolvido.
- Com a utilização de um servidor *socket*, integrar as rotas a serem a cumpridas pelo representante comercial.
- Um dos principais objetivos do aplicativo é realizar o georreferenciamento de cada cliente, onde o responsável pelo registro, localizado no endereço de domicílio do cliente captura esta informação com a utilização de um aparelho que utilize plataforma Android, desta forma construir uma massa de dados que possibilitará a construção de rotinas mais complexas e eficientes.
- Através da aplicação Android apresentar os pontos (rota) a serem visitados em um mapa.
- Registrar o cumprimento da rota, gravando a localização do representante no momento em que o mesmo visitou o cliente.
- Implementar uma rotina que atualize o georreferenciamento do cliente, enviando através de um servidor *socket* esses dados atualizados.

1.5 METODOLOGIA

Conforme Gil (2007, p. 64) “a pesquisa bibliográfica é desenvolvida com base em material elaborado, constituído principalmente de livros e artigos científicos”.

Segundo Marconi e Lakatos (1996, p. 66) a pesquisa bibliográfica “abrange toda a bibliografia já tornada pública em relação ao tempo de estudo”.

Portanto, no ponto de vista técnico a pesquisa bibliográfica será necessária para a compreensão de tecnologias e recursos tecnológicos já existentes.

O projeto não limita-se ao levantamento teórico, portanto outras técnicas de pesquisa serão necessárias. No trabalho com levantamento e análise de dados é necessária uma abordagem que qualifique e quantifique a proposta.

De acordo com Kauark, Manhães e Medeiros (2010, p. 26) em uma pesquisa qualitativa “há uma relação dinâmica entre o mundo real e o sujeito, isto é, um vínculo indissociável entre o mundo objetivo e a subjetividade do sujeito que não pode ser traduzido em números”, já a quantitativa segundo Kauark, Manhães e Medeiros (2010, p. 26) “considera o que pode ser quantificável, o que significa traduzir em números opiniões e informações para classificá-las e analisá-las”.

Sendo assim, para o desenvolvimento do presente trabalho será um estudo de caso e adotará a abordagem qualitativa aliada à pesquisa bibliográfica.

1.6 ESTRUTURA DO TABALHO

No capítulo dois é apresentado a fundamentação teórica, onde é descrito o funcionamento do GPS e os recursos de localização disponível em dispositivos Android. O capítulo três trata dos materiais utilizados, as ferramentas de desenvolvimento, como o Eclipse, bibliotecas e demais tecnologias necessárias para o desenvolvimento deste trabalho. No capítulo quatro são apresentados os resultados, com a análise do projeto, com diagramas da UML (*Unified modeling language*), trechos de código fonte e a explicação técnica de seu funcionamento, também as telas e opções disponíveis no sistema.

2 FUNDAMENTAÇÃO TEÓRICA

Aqui serão detalhados alguns conceitos importantes para a apresentação deste trabalho. A devida compreensão de alguns conceitos é essencial para o desenvolvimento e conclusão deste projeto de software.

2.1 LOCALIZAÇÃO GEOGRÁFICA

Uma das preocupações mais antigas da humanidade é saber “onde se está” ou então onde alguém está a nível regional ou até mesmo global. Uma das formas mais tradicionais de identificar a própria localização é com a utilização da bússola, esse equipamento composta basicamente por uma agulha magnetizada sobre uma superfície plana indica sempre a direção do Norte-Sul.

No período conhecido como era do descobrimento, momento em que os europeus descobriram novas terras, a bússola possibilitou que os viajantes, seja de navio ou a cavalo, se localizassem de forma eficiente. Esse instrumento possibilitou que este mesmo percurso pudesse ser feito posteriormente.

Uma técnica contemporânea para o geolocalização é a utilização do GPS. Segundo Dilião (2014), atualmente o rastreamento por GPS é realizado por um sistema que foi desenvolvido pelos Estados Unidos, e é composto por 24 satélites que determinam a posição de um receptor na superfície terrestre, através dos satélites dispostos sobre a região.

Este recurso largamente difundido é encontrado em rastreadores de automóveis, aparelhos de orientação de direção e rotas e mais atualmente alguns aparelhos celulares modernos conhecidos como *smartphones*. Com a popularização desta tecnologia surgiu a possibilidade do desenvolvimento de aplicativos para *smartphone* que utilizam recurso de GPS, desta forma recursos de geolocalização.

Conforme Ogliari e Brito (2014) a utilização de aplicativos LBS (Location-based services) não é nenhuma novidade no mundo da tecnologia há um bom tempo. Mesmo antes do surgimento e do crescimento incrível do *smartphone*, os telefones celulares já forneciam estes *features* em diversos aplicativos.

Pode-se construir aplicações muito ricas utilizando recursos de geolocalização. Utilizando dados de altitude, longitude e latitude a plataforma Android é capaz de indicar em um mapa uma determinada localização, podendo, inclusive construir rotas a serem seguidas.

2.1.1 Coordenadas Geográficas

Para localizar um ponto na superfície terrestre de forma exata foi criado um sistema de linhas imaginárias chamada sistema de coordenadas geográficas. Esta medição é realizada levando em consideração a linha do equador e o paralelo de Greenwich. Estas medições se traduzem em geocódigos formados pela latitude e longitude.

Logo, para encontrar um ponto exato sobre a superfície terrestre precisa-se saber quais são as coordenadas de longitude e latitude. A seguir será descrito de forma mais detalhada de que se tratam estas coordenadas.

2.1.2 Latitude

Conforme Dilião (2014) a latitude é a distância ao equador medida ao longo do meridiano de Greenwich. Esta distância mede-se em graus, podendo variar entre 0° e 90° para o Norte ou para o Sul. Além dos graus, estas podem ser formadas por frações, e neste caso são utilizados os minutos e segundos (que variam entre 0 e 59). Por exemplo, a cidade de Francisco Beltrão, no estado do Paraná, está localizada na latitude $26^\circ 04' 51''$ S (26 Graus, 04 Minutos e 51 segundos ao SUL).

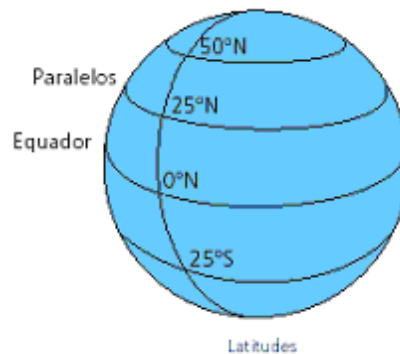


Figura 1 – Latitude em graus a partir do Equador
Fonte: Adaptado de Dilião (2014)

2.1.3 Longitude

Conforme Dilião (2014) a longitude é a distância ao meridiano de Greenwich medida ao longo do Equador. Esta distância mede-se em graus, podendo variar entre 0° e 180° para o Leste ou para o Oeste. Além dos graus, estas podem ser formadas por frações, e neste caso são utilizados os minutos e segundos (que variam entre 0 e 59). Por exemplo, a cidade de Francisco Beltrão, no estado do Paraná, está localizada na longitude $53^\circ 03' 18''$ O (53 Graus 03 Minutos e 18 Segundos Oeste).



Figura 2 - Longitude em graus a partir do Equador
 Fonte: Adaptado de Dilião (2014)

2.2 FUNCIONAMENTO DO GPS

Conforme IBGE (Instituto Brasileiro de Geografia e Estatística) (2014) em 1978 entraram em funcionamento os primeiros satélites NAVSTAR – GPS (*Navigation Positioning System with Time And Ranging – Global Positioning System*), que compõem um sistema de navegação desenvolvido pelo departamento de defesa dos Estados Unidos da América, para uso restrito dos militares americanos.

Em meados dos anos 80 este sistema foi aberto para uso civil e para outros países, isto popularizou e difundiu esta tecnologia para diversos projetos, aparelhos e aplicações. Tornou-se tão popular que passou a ser usado para localização de

diversos objetos na superfície terrestre, no auxílio à navegação de carros, aviões e navios, entre outras inúmeras aplicações.

Existem no espaço um conjunto de 27 satélites, sendo 3 deles desativados, que servem como “reservas”, desta forma caso algum dos 24 ativos falhe ou pare de funcionar, estes três reservas irão ocupar o lugar deste satélite problemático. Os 24 satélites em funcionamento estão dispostos de forma que, em qualquer ponto da terra, pelo menos 4 satélites estejam visíveis para o sistema de GPS. Assim, sempre que alguém utilizar o serviço de geolocalização terá a certeza de que estará coberto pelo serviço. A **Erro! Fonte de referência não encontrada.** mostra a disposição dos satélites em relação ao globo terrestre.



Figura 3 - Disposição dos satélites de GPS

Fonte: Adaptado de Dilião (2014)

Com o uso dos receptores GPS é possível fazer a localização de onde algo está no planeta. Os satélites emitem sinais de ondas de rádio para os receptores, através de complexos cálculos, com base na hora do envio do sinal, o aparelho calcula a distância que o receptor está do satélite, combinando o cálculo realizado por alguns satélites, através de triangulação o sistema identifica um ponto comum entre as distâncias e os satélites utilizados.

A localização encontrada é colocada em coordenadas geográficas, a partir desta informação é possível marcar a posição em um mapa. A Figura 4 exemplifica graficamente e de maneira simplificada como que é calculado o posicionamento de um receptor através do sinal de GPS.



Figura 4 - Modelo de triangulação utilizado pelo sistema de GPS

Fonte: Tecmundo (2014)

2.2.1 Conversão dos Geocódigos

Os valores de latitude e longitude são apresentados em graus, minutos, segundos e em casos com maior precisão até mesmo milissegundos. Mas para a utilização destes valores por sistemas computacionais é necessária à conversão deste geocódigo para formato decimal, desta forma fica mais fácil de trabalhar em nível de desenvolvimento de software e para cálculos realizados pelos sistemas computacionais.

Como exemplo será utilizada a localização da cidade de Francisco Beltrão, no estado do Paraná, que está localizada na latitude 26° 04' 51" S e na longitude 53° 03' 18" O. Para omitirmos o ponto cardeal a qual se refere a coordenada, coloca-se como latitude negativa as localizadas no hemisfério sul e longitude negativa o que está localizado na parte ocidental do planeta, ficando da seguinte forma: latitude -26° 04' 51" e longitude -53° 03' 18". As coordenadas do hemisfério norte e da parte oriental do planeta ficam com valores positivos.

Para realizar a conversão de graus para um valor de ponto flutuante deve-se considerar que:

- Um grau equivale a 60 minutos.
- Um minuto equivale a 60 segundos.
- Um grau equivale a 3600 segundos.

Sendo o cálculo da seguinte forma:

$$\text{Graus em decimais} = -\left(\text{graus} + \frac{\text{minutos}'}{60} + \frac{\text{segndos}'}{3600}\right)$$

Seguindo o exemplo de Francisco Beltrão:

Latitude: -26° 04' 51''

$$\text{Latitude} = -\left(26 + \frac{04'}{60} + \frac{51'}{3600}\right)$$

$$\text{Latitude} = -26.080833$$

Longitude: -53° 03' 18''

$$\text{Longitude} = -\left(53 + \frac{03'}{60} + \frac{18'}{3600}\right)$$

$$\text{Longitude} = -53.0550$$

2.3 Geolocalização em Aparelhos Android

A tecnologia GPS tomou aspectos populares, estando disponível em diversos dispositivos eletrônicos de fácil aquisição, entre eles os *tablets* e *smartphones*. Os recursos existentes nestes aparelhos, da forma como descrito nos capítulos anteriores possibilitam a localização do aparelho, conseqüentemente do portador deste.

A API (*Application Programming Interface*) de desenvolvimento Android permite a leitura dos valores apresentados pelo GPS, retornando os valores em geocódigos, o que facilita o armazenamento destas informações no SGBD (Sistema gerenciador de banco de dados). Facilita também a realização de cálculos matemáticos necessários para esta aplicação.

Os aparelhos celulares com sistema operacional Android permitem identificar sua geolocalização através do recurso de GPS, provavelmente o modo mais eficiente, pois independe de recursos de rede, funcionando em estradas e lugares distantes dos centros urbanos, pois o sinal de GPS cobre toda a superfície terrestre, desde que em área aberta. Outras formas são triangulação por meio da rede de telefonia e localização por meio da rede wi-fi.

3 MATERIAIS E MÉTODOS

Neste capítulo serão apresentados os materiais e ferramentas de softwares utilizados para o desenvolvimento do projeto, também serão detalhados os métodos de desenvolvimento, destacando os principais pontos abordados.

3.1 MATERIAIS

As ferramentas utilizadas para este projeto foram o Eclipse Juno como IDE (*Integrated development environment*). Também foi instalado o Android SDK (*Software Development Kit*) para que fosse possível desenvolver aplicativos para a plataforma Android versão 4.1 utilizando um computador desktop. Foram utilizados dois bancos de dados no projeto: o SQLite para persistência dos dados no dispositivo Android e o PostgreSQL 9.3 como banco de dados no servidor (repositório central de dados). A IDE Eclipse também será utilizada no desenvolvimento de um aplicativo desktop servidor, o qual receberá dados do dispositivo Android via *socket*, para uma posterior persistência em um banco de dados, o trânsito das informações será realizado por meio de objetos, para isso foi utilizada a biblioteca GSON versão 2.3.1. Para a construção de modelos da UML foi utilizado o Astah Community 2.0.

Neste trabalho, com exceção ao sistema operacional, foi utilizado para o desenvolvimento somente software livre, estes instalados no sistema operacional Windows 8 64 bits. Esta escolha foi feita devido a facilidade de aquisição destas ferramentas, por não haver custo para a utilização, visto que é um dos princípios do software livre a distribuição gratuita.

As aplicações para Android que possuem mapas não funcionam corretamente no emulador, pois o Google Play Services não é compatível com o emulador, por isso foi necessário o uso de um aparelho real para a realização de testes. Assim, foi utilizado o celular Samsung Galaxy SIII mini, com Android versão 4.1. A Figura 5 apresenta a integração das ferramentas utilizadas no desenvolvimento deste projeto.



Figura 5 - Comunicação e Ferramentas

Fonte: Autoria Própria (2014)

3.1.1 Eclipse

Conforme Palmeira (2014), o Eclipse é uma IDE que compreende a vários tipos de linguagens e aceita a instalação de *plug-ins* para o desenvolvimento em diferentes plataformas.

É desenvolvido em Java, sendo iniciado pela IBM em novembro de 2000, que posteriormente disponibilizou como software livre. Atualmente é uma das ferramentas líder de mercado para desenvolvimento em Java.

Neste trabalho está sendo utilizado o Eclipse Juno Service Release 2 (2013), Conforme IDE Eclipse (2014).

Uma pesquisa realizada em 2012 pela Rabel Labs, levantou quais são as IDEs mais utilizadas por desenvolvedores Java, conforme apresentado na Figura 6.

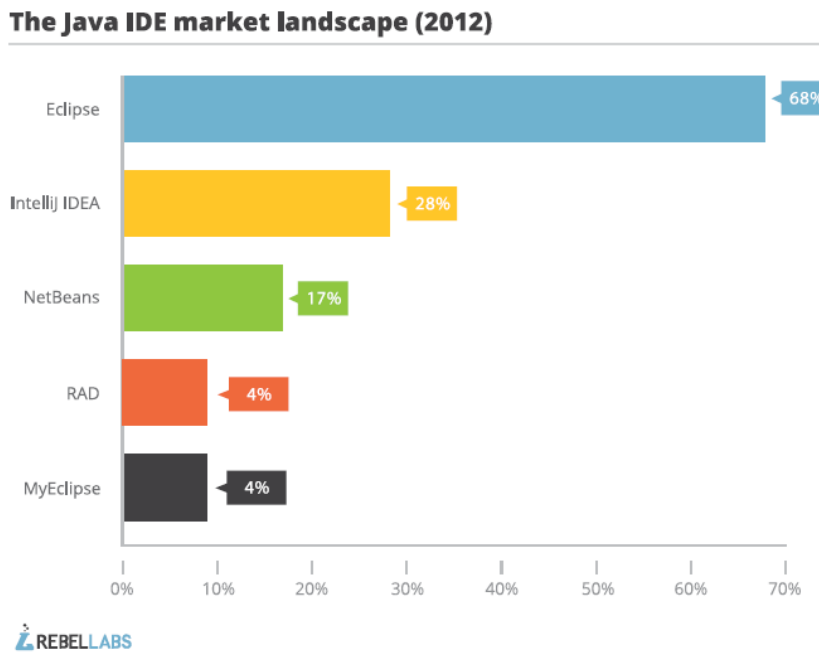


Figura 6 - IDEs mais utilizadas por desenvolvedores Java

Fonte: RebelLabs 2012

Conforme a pesquisa da Rebel Labs, o ambiente de desenvolvimento Eclipse figura como o mais utilizado por desenvolvedores Java, com 68% do mercado, mais do que o dobro de utilizadores do IntelliJ IDEA, segundo colocado na pesquisa, sendo utilizado por 28% dos desenvolvedores. É importante ressaltar também que a pesquisa perguntou quais IDEs eram utilizadas, podendo ser mais de uma por pessoa pesquisada, por este motivo a soma de todos os percentuais ultrapassam cem por cento.

Para o desenvolvimento de aplicativos para a plataforma Android, deve-se utilizar o *plug-in* ADT (Android Developer Tools). Este é um *plug-in* para a IDE Eclipse que disponibiliza um ambiente integrado para o desenvolvimento de aplicativos para a plataforma Android. O ADT permite criar uma aplicação UI (*User interface*), adicionar API e Frameworks, depurar o código fonte, exportar o aplicativo para distribuição e instalação, entre outros recursos que visam facilitar o desenvolvimento de aplicativos Android.

Na Figura 7 é apresentada a tela de desenvolvimento no Eclipse utilizando o *plug-in* ADT.

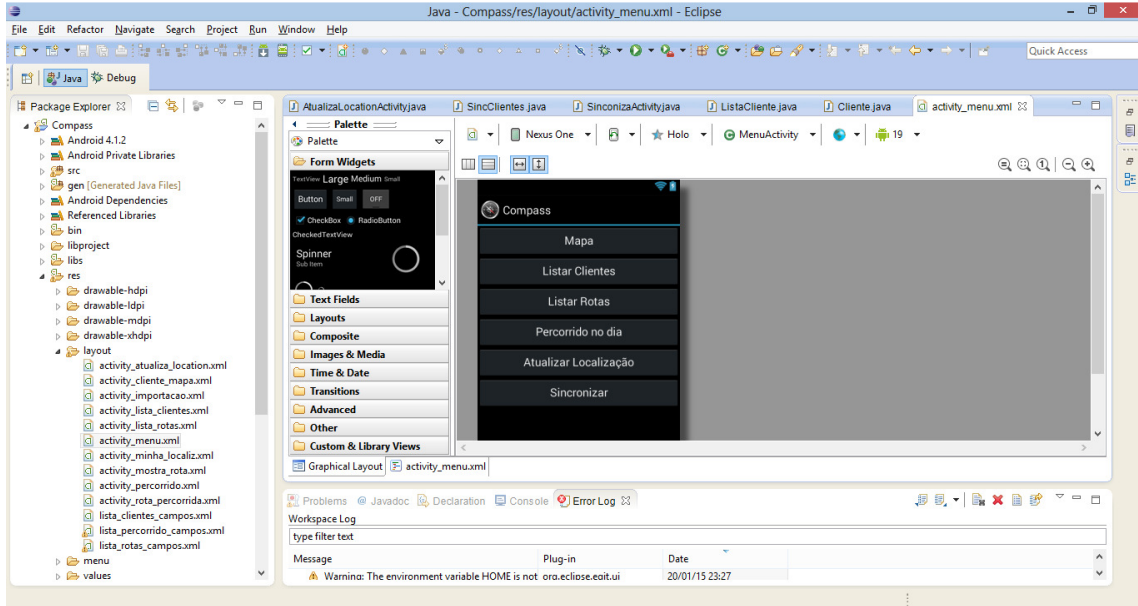


Figura 7 - Tela de Desenvolvimento Eclipse e *plug-in* ADT

Fonte: Autoria Própria (2015)

3.1.2 Android SDK

O Android SDK fornece um conjunto de bibliotecas para linguagem Java e C, assim como as ferramentas necessárias para que os desenvolvedores possam construir, testar e depurar aplicativos para a plataforma Android.

Este SDK pode ser integrado com o Eclipse, a partir do *plug-in* ADT, unindo assim, a IDE às bibliotecas necessárias para o desenvolvimento de aplicativos para aparelhos Android, facilitando a execução dos aplicativos, bem como a depuração.

O Android SDK também dispõe de um emulador para simular dispositivos com sistema operacional Android, e este pode ser executado como um aplicativo comum, existe um *plug-in* dentro do Eclipse que visa justamente integrar o ambiente com o emulador.

É possível também executar e depurar os aplicativos diretamente no dispositivo real (por exemplo um *smartphone* ou *tablet*), através da porta USB (*Universal Serial Bus*) do computador, isto sem dúvidas facilita muito o desenvolvimento da aplicação, por permitir a depuração diretamente no aparelho, além da execução ser mais rápida e a apresentação da interface gráfica é mais eficiente.

3.1.3 SQLite

Uma das formas mais comuns de persistir informações no Android é utilizando o banco de dados relacional SQLite. Uma característica interessante é que este banco já vem instalando na plataforma Android, o que facilita seu uso e o tornou o banco de dados padrão para a plataforma Android.

A API de desenvolvimento Android traz algumas bibliotecas para a conexão e manutenção do banco, permitindo a utilização de comandos SQL, assim como controles transacionais e a utilização de frameworks objeto-relacionais.

Alguns bons exemplos de classes que facilitam o uso deste banco de dados relacional são *SQLiteOpenHelper*, *SQLite* e *SQLiteDatabase*. Estas são classes utilizadas para criar, manter o banco, permitindo executar comandos DDL (*Data definition language*) e DML (*Data manipulation language*) no banco de dados criado.

3.1.4 PostgreSQL

Segundo Postgres (2014), o PostgreSQL é um sistema gerenciador de banco de dados objeto-relacional de código aberto. Foi criado na Universidade de Berkley, na Califórnia, há algumas décadas e é atualmente o banco de dados mais avançado de código aberto.

Fornecer suporte a grande parte do padrão SQL e oferece muitas funcionalidades modernas como integridade transacional, gatilhos, visões, chaves estrangeiras, sub consultas entre outros. Os recursos deste banco ainda podem ser ampliados pelo usuário, pois podem ser criados novos tipos de dados, novas funções, métodos e índices e linguagens procedurais.

Uma característica deste banco é a robustez, podendo ter tabelas com até 32TB e campos com tamanho de 1.6TB de tamanho em disco, colocando-o entre as melhores opções de SGBD gratuitos e open source do mercado.

3.1.5 JSON e GSON

O JSON (*JavaScript Notation Object*) é uma forma simples de transportar dados computacionais. É uma notação de objeto de JavaScript e se tornou uma alternativa mais leve para a utilização de arquivos XML (*Extensible markup language*).

Foi concebido por um trabalho de Douglas Crockford em 1999. Este formato de texto é independente de linguagem ou plataforma de programação, o que tem ajudado na facilidade de aplicação. Também porque tem surgido muitas bibliotecas, muitas vezes de terceiros, que permitem a tradução de objetos em JSON e JSON em objetos. Esses componentes facilitam o trabalho de montagem e desmontagem dos dados para os programadores e analistas.

Sendo o JSON um padrão de comunicação, algo semelhante ao XML, plugins e frameworks foram desenvolvidos para dar suporte ao seu uso, dentre estes, destaca-se o GSON, esta é uma biblioteca desenvolvida pela Google, tendo sua primeira versão lançada em 2008. A versão mais atual é a 2.3 lançada em 2014.

Esta biblioteca permite converter objetos Java em Strings JSON. Também pode ser utilizada para converter um texto no formato JSON para um objeto Java equivalente. O GSON pode trabalhar com muitos objetos Java incluindo objetos pré-existentes no qual o desenvolvedor não tem acesso ao código fonte.

Existem muitos projetos *open source* que convertem objetos Java em JSON, mas para isso é necessário adicionar *annotations* dentro das classes, o que deixa o processo mais trabalhoso e sujeito a erros. Isso também faz com que não seja possível utilizar objetos já existentes no qual o desenvolvedor não tem acesso ao código fonte, pois não teria como adicionar as *annotations*.

3.1.6 Google Maps

O Google Maps é uma ferramenta para a visualização de mapas disponível para o Android. Conforme Google (2014) para facilitar a adição de recursos de mapeamento a seu aplicativo, o complemento das APIs do Google inclui uma biblioteca externa do Google Maps, *com.google.android.maps*.

Esta biblioteca possui inúmeros recursos de controle, dentre estes recursos é possível incluir marcadores em pontos específicos no mapa, ou então uma série de

pontos, permite também personalizar tais marcadores, construir trajetos, rotas, integrar com dispositivos de localização do aparelho, dentre outros.

A ferramenta permite diferentes visões do mapa, com a possibilidade de controlar a rotação, zoom e propriedades de mudar a perspectiva, apresentando com aspecto tradicional, com marcações de cidades, ruas e pontos importantes, como também a visão real da superfície terrestre através de fotos aéreas.

Todo acesso ao Google Maps necessita de uma chave única de acesso, para evitar acessos desnecessários. A chave é utilizada também para limitar o número de acessos por um determinado sistema ou aplicativo. Caso o número de acessos seja expansivo o serviço passa de gratuito a pago.

3.2 MÉTODOS

A abordagem para o desenvolvimento do Aplicativo de Controle de Rotas e Rastreamento – Compass, foi dividido em várias etapas conforme o modelo sequencial linear descrito por Pressman (2006). Este modelo de desenvolvimento de software começa pela especificação dos requisitos pelo cliente e progride ao longo do planejamento, modelagem, construção até a implantação do sistema, com sugere a Figura 8.

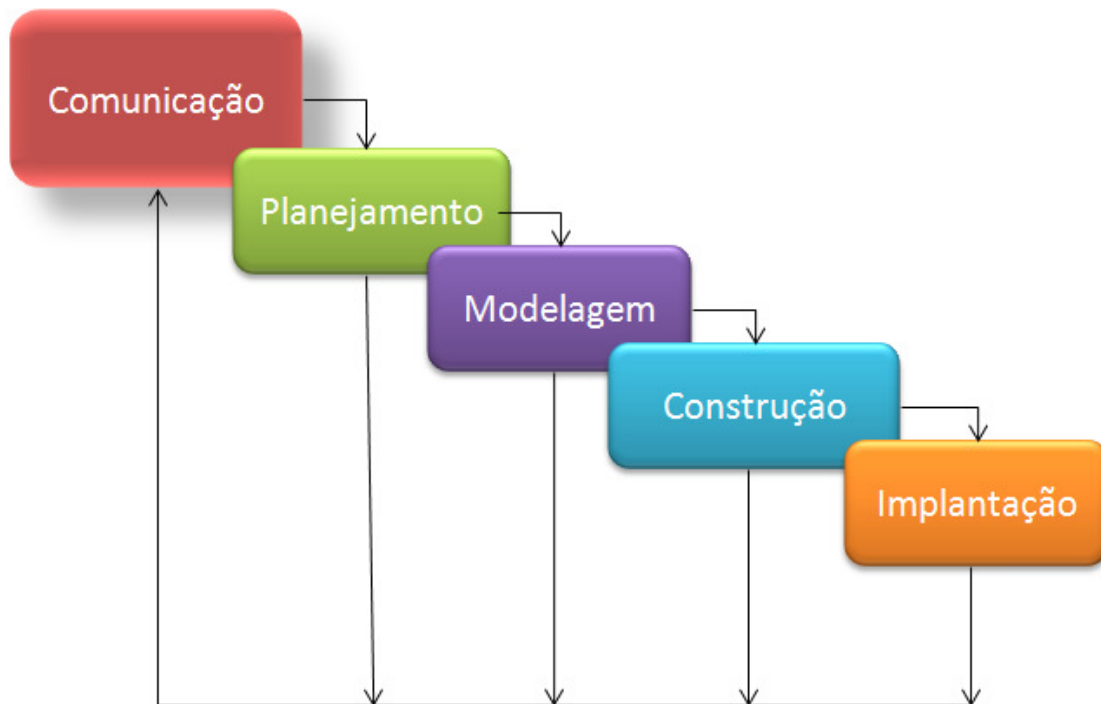


Figura 8 - Modelo Sequencial Linear

Fonte: Adaptado de Pressman (2006)

Este modelo foi adaptado de acordo com a necessidade do projeto, incluindo e retirando etapas, o resultado é apresentado na sequência:

A) **Requisitos:** Foi feito um levantamento inicial dos requisitos para o desenvolvimento das principais funcionalidades. Novos requisitos são criados conforme as necessidades e serão implementados nas próximas versões.

B) **Análise:** Será criado um diagrama de casos de uso para descrever um cenário com as principais funcionalidades do aplicativo. Para a construção do diagrama será utilizado a ferramenta AstahCommunity.

C) **Modelagem:** Para o projeto será desenvolvido um diagrama de classes, mostrando a estrutura principal das classes do aplicativo Android.

Também será criado um diagrama de Entidade e Relacionamentos que irá mostrar a estrutura do banco de dados da parte do Servidor *Socket* e também do Aplicativo. Para o modelo de Classes também será utilizado a ferramenta Astah Community.

D) **Construção:** No desenvolvimento serão utilizadas as ferramentas e tecnologias listadas no item 3.1. Para testes será utilizado um aparelho Android real, contendo o Android 4.1.

E) **Implantação:** A implantação será feita em um representante comercial na cidade de Francisco Beltrão – PR, para aprovar os recursos disponibilizados e elicitar novos requisitos para as versões posteriores.

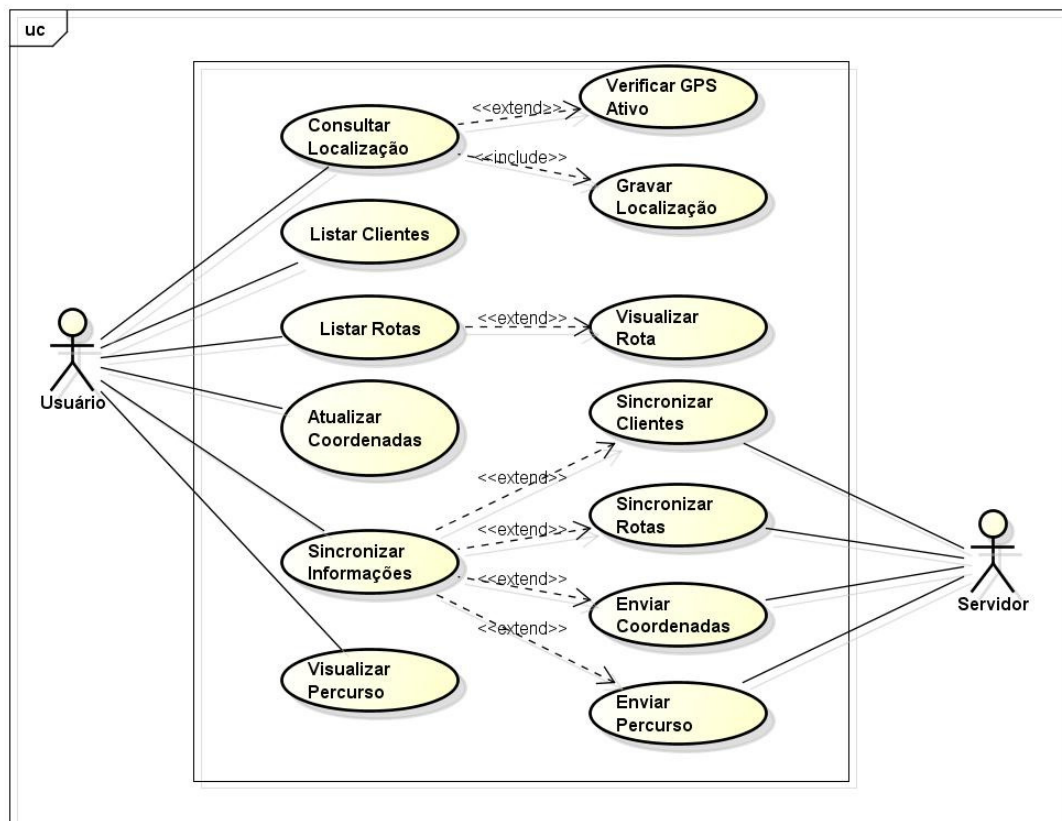
4 RESULTADOS

Neste capítulo será apresentado inicialmente a análise do projeto, com diagrama de casos de uso, de classes conforme os padrões definidos pela UML, também o diagrama de entidade e relacionamento do banco de dados do aplicativo Android.

Na sequência será apresentada as principais telas do sistema, com seus resultados e trechos dos códigos fontes.

4.1 ANÁLISE DO PROJETO

Os principais requisitos funcionais do sistema serão descritos através do diagrama de casos de uso conforme a Figura 9, mostrando as atividades possíveis entre o utilizador do aplicativo móvel, representado pelo ator Usuário e as opções disponíveis.



powered by Astah

Figura 9 - Diagrama de Casos de Uso

Fonte: Autoria Própria (2014)

O ator Servidor apresentado no diagrama representa o servidor *socket* responsável pelo envio e recebimento de informações entre o aplicativo Compass e o banco de dados onde as informações ficam centralizadas.

Com base no diagrama de casos de uso, foi criada a estrutura de classes para implantação dos sistemas, que é apresentada na Figura 10.

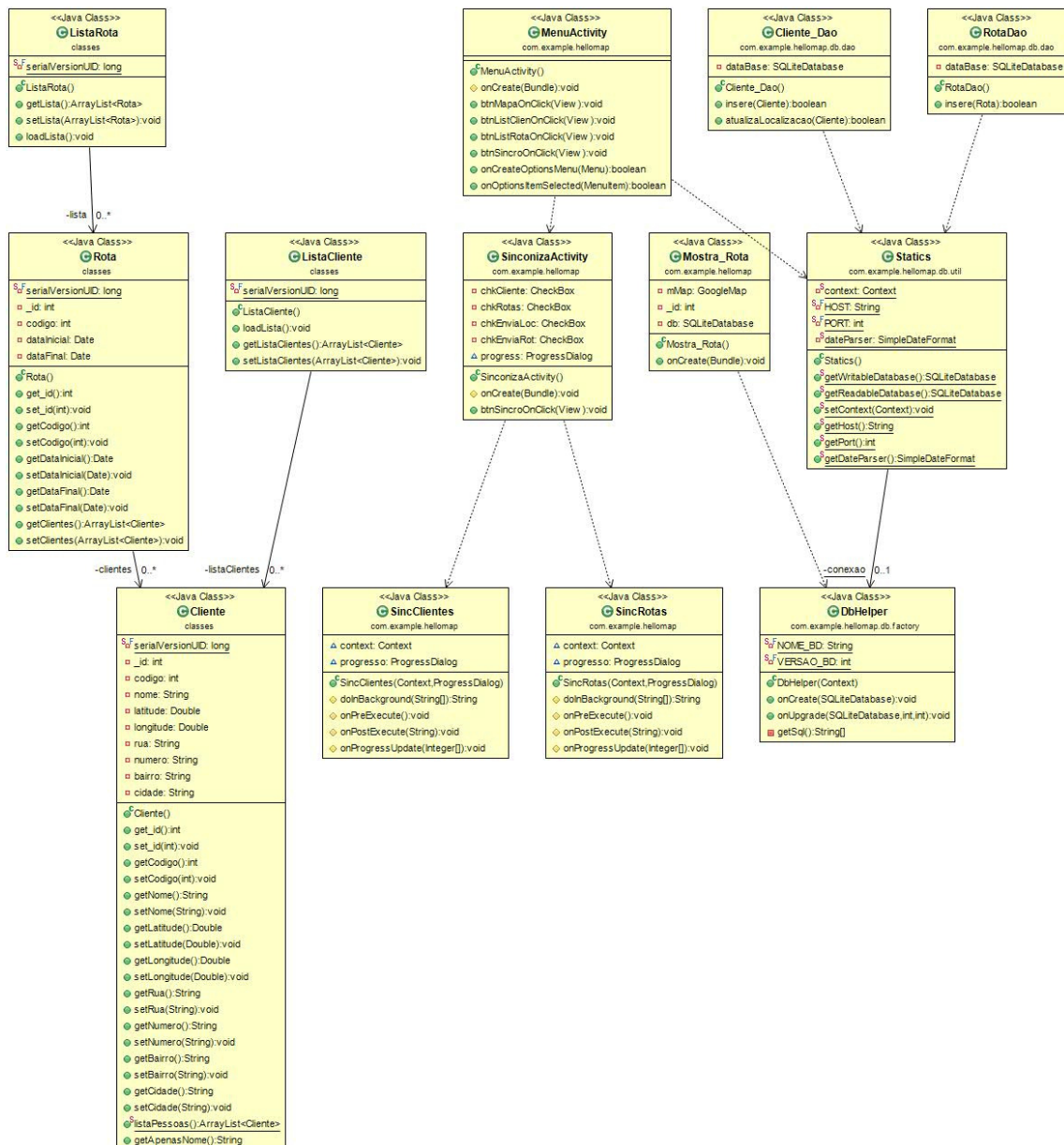


Figura 10 - Diagrama de Classes

Fonte: Autoria Própria (2014)

O modelo de entidade e relacionamento descreve conceitualmente os dados que serão utilizados pelos sistemas. A entidade Cliente representa os dados relevantes para a aplicação Compass, como os atributos latitude e longitude, fundamentais para a localização do mesmo. Outra importante entidade é a Rota, que representa as rotas a serem cumpridas.

O diagrama apresentado na Figura 11 mostra a estrutura de tabelas, campos e associações do banco de dados da aplicação Android.

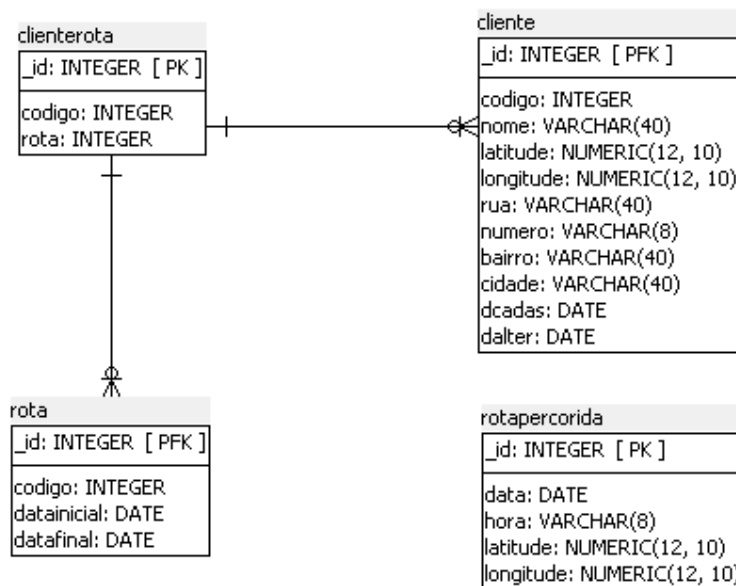


Figura 11 - DER Projeto Android

Fonte: Autoria Própria (2015)

O aplicativo possui quatro entidades no banco de dados. A entidade *rotapercorrida* armazena a data, hora e localização geográfica sempre que uma leitura nova da localização é realizada, conforme descrito no capítulo 4.2.1.

A tabela *cliente* mantém os dados dos clientes, como endereço completo e localização geográfica (latitude e longitude). A tabela *rota* armazena o período inicial e final em que determinada rota deve ser realizada, a tabela *clienterota* é responsável pela informação de quais clientes devem ser visitados em cada rota, havendo apenas duas chaves estrangeiras na tabela, uma com a tabela de clientes e outra com a tabela de rotas.

4.2 APRESENTAÇÃO DO SISTEMA COMPASS

O sistema para dispositivos móveis desenvolvido se comunica através de um servidor *socket*, o qual realiza conexão com um servidor de banco de dados PostgreSQL. O servidor *socket* é utilizado para que compartilhar estas informações com todos os dispositivos móveis integrados no sistema.

As informações trafegadas consistem em dados dos clientes, como dados pessoais, endereço e coordenadas geográficas. Também são enviadas informações das rotas a serem realizadas e o dispositivo móvel envia atualizações dos clientes e dados do percurso realizado.

Outras funcionalidades importantes do aplicativo é a apresentação dos pontos a serem visitados diretamente em um mapa, utilizando o recurso do Google Maps e a identificação dos pontos já visitados.

A Figura 12 demonstra a como se dá a troca de mensagens entre o servidor e o dispositivo móvel.

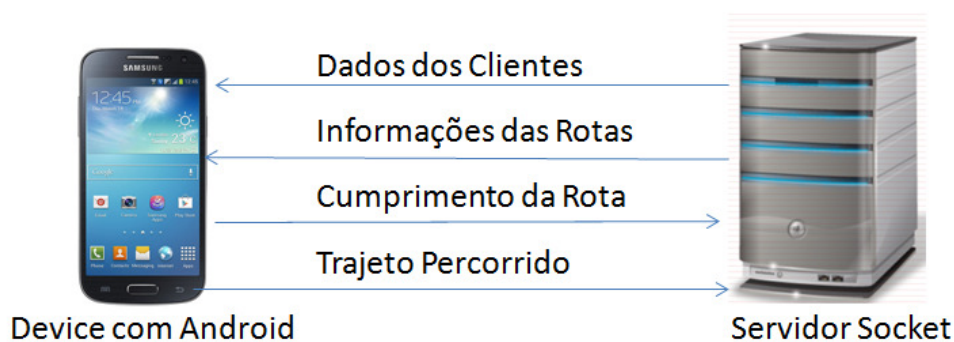


Figura 12 – Comunicação entre Servidor e Dispositivo

Fonte: Autoria Própria (2015)

As cargas de dados, com informações relevantes dos clientes e quais serão as rotas a serem cumpridas são enviadas para o dispositivo Android. Após o cumprimento do percurso, os dados são atualizados no servidor com os novos dados do aparelho.

O sistema possui inúmeras funcionalidades que permitem o controle de cumprimento de rota, armazenamento de percurso realizado, dentre outras funcionalidades. A tela principal, conforme mostrado na Figura 13, fornece acesso às rotinas do sistema.



Figura 13 - Menu Principal

Fonte: Autoria Própria (2015)

A função da tela principal é fornecer acesso as demais funcionalidades do sistema. Cada botão permite acesso cada uma das funcionalidades principais descritas do diagrama de casos de uso.

4.2.1 Localização e Armazenamento do percurso

Ao acessar a opção Mapa, através dos recursos de GPS, é exibida no mapa do Google Maps a localização do dispositivo, conforme apresentado na Figura 14. A

localização é indicada no mapa através de um ponto na cor azul, e atualizada sempre que a localização alterar em 20 metros ou mais.



Figura 14 - Localização No Mapa

Fonte: Autoria Própria (2015)

Sempre que a localização do aparelho é atualizada, esta é gravada no banco de dados, para que posteriormente seja possível construir o percurso realizado, e para demais rotinas que necessitam desta informação.

A localização pode ser realizada através de redes sem fio ou através de recurso de GPS, ambas indicam a localização a nível da rua. Nos dispositivos com sistema operacional Android é possível desabilitar estes dois recursos, fazendo com que os sistemas não funcionem corretamente. Antes de exibir o mapa com a localização, caso o recurso de localização por GPS não estiver habilitado a aplicação abre a tela de configuração de serviço de localização.

4.2.2 Lista de Clientes e Lista de Rotas

A Lista de Clientes apresenta os principais dados de cada cliente, estes dados são enviados pelo servidor, uma vez que não é possível cadastrar manualmente informações pelo sistema Compass. Clicando sobre o registro, é exibida no mapa a localização do cliente, indicando também o seu endereço, esta consulta é informativa. Dentro da consulta da localização do cliente no mapa é possível realizar a atualização do endereço geográfico do mesmo. Esta rotina é importante por permite a atualização da localização de forma individual.

Na consulta de rotas, são apresentadas as datas de início e fim da rota. Estas datas indicam qual é o período que a rota deve ser cumprida. Selecionando um item da lista, a rota é exibida no mapa. Esta exibição indica quais são os pontos que devem ser visitados no período, mas não sugere um 'melhor' caminho.

Cada marcador apresentado no mapa indica um cliente a ser visitado, se clicado apresenta o nome do cliente e o seu endereço. O marcador pode ser de duas diferentes cores, o vermelho indica que o cliente ainda não foi visitado no período previsto, o marcador verde indica aqueles pelos quais o vendedor já passou como na Figura 15.



Figura 15 - Consulta de Rota

Fonte: Autoria Própria (2015)

4.2.3 Consulta do percurso percorrido no dia.

O resultado de todo o rastreamento via GPS realizado pela aplicação pode ser consultado através da tela de consulta do percurso percorrido no dia. Em primeiro plano são exibidos os dias em que houve monitoramento da rota. Ao selecionar um determinado dia, o aplicativo exibe o caminho que realizado e gravado pelo aplicativo, como apresentado na Figura 16.

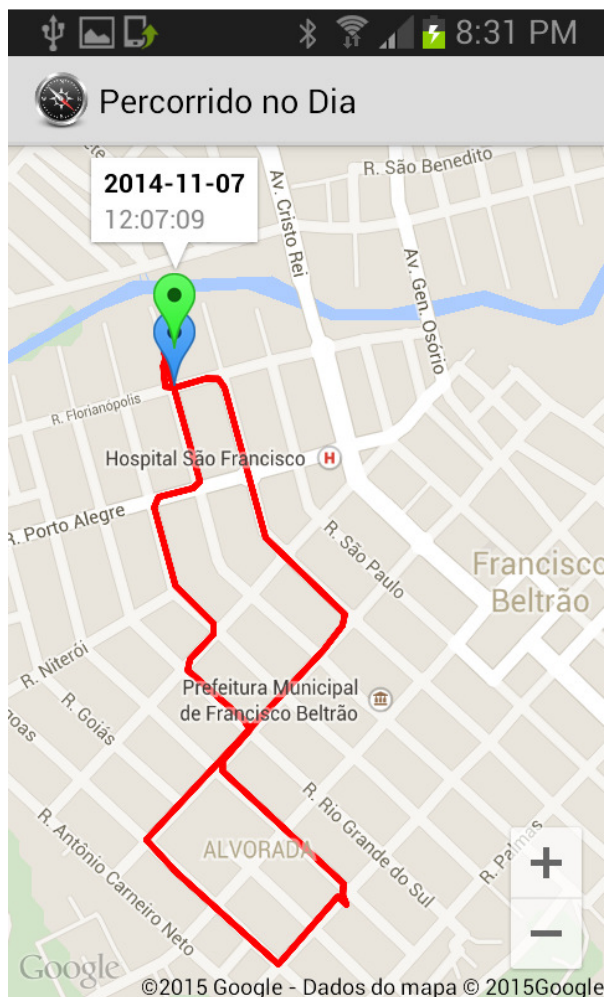


Figura 16 - Caminho Percorrido

Fonte: Autoria Própria (2015)

No mapa é apresentado o caminho, marcando com uma linha contínua as localizações geográficas registradas, na ordem em que foram registradas, reconstruindo assim o caminho realizado. Os marcadores indicam a data e hora de início e fim de percurso, o azul indica o início e o verde o fim.

4.2.4 Atualizar localização e sincronizar dados

Tradicionalmente sabe-se a localização de uma pessoa pelo endereço, cidade, bairro, número, etc. A nível geográfico é algo mais complexo, sendo que mesmo no banco de dados do sistema não haverá esta informação. Através da rotina de atualizar a localização, o aplicativo executa um procedimento que percorre a lista de clientes existentes e tenta localizar a latitude e longitude por meio do

endereço cadastrado para cada cliente. É possível que o endereço seja ambíguo, ou seja, exista mais de um resultado possível para um endereço, por este motivo é gravado no banco de dados apenas o resultado mais relevante encontrado. Caso este já possua a latitude e longitude gravadas, estes dados serão sobre-escritos.

Os dados atualizados dos clientes devem ser transmitidos do aparelho para uma aplicação servidora que irá manter estas informações de forma centralizada, por isso foi adicionada a rotina de sincronizar. Como no aparelho não existem cadastros, os dados de clientes e rotas programadas são importados por meio das opções “Receber Cliente” e “Receber Rotas”.

A opção “Enviar Localizações” irá mandar para o servidor todas as localizações de latitude e longitude que foram atualizadas para o cliente, na opção “Enviar Dados Rotas” são sincronizadas todas as informações de latitude, longitude e data dos pontos registrados pelo GPS e gravados no banco de dados. Desta forma a aplicação para dispositivos móveis fica integrada com um banco de dados relacional mais robusto, o que permite que outras aplicações, em outros sistemas operacionais possam compartilhar dos mesmos dados.

4.3 DESENVOLVIMENTO DO APLICATIVO

Este trabalho trata do desenvolvimento de um aplicativo para o sistema operacional Android. Também foi necessário um servidor *socket* para trafegar os dados necessários entre o aplicativo e o banco de dados que tinham informações necessárias, além do envio de novas informações.

A aplicação Android tem como principal recurso de desenvolvimento o pacote de serviços disponibilizado pelo Google Play Services. Este pacote disponibiliza acesso fácil a diversos serviços do Google, sendo totalmente integrado com o sistema operacional Android.

A API do Google inclui uma biblioteca do Google Maps - *com.google.android.maps*. As classes da biblioteca do Google Maps incorporam o download, a renderização e o armazenamento em *cache* de blocos do Google Maps, bem como uma grande variedade de opções de controle. Desta forma, sempre que acessada uma localização no mapa, esta é baixada da internet, carregada e renderizada na aplicação. Mesmo que o sinal de internet caia, depois de carregado o

mapa, este continuará sendo exibido, pois as partes já baixadas permanecerão em *cache*.

Para adicionar um mapa na aplicação é necessário adicionar a classe *com.google.android.gms.maps.SupportMapFragment* no arquivo de leiaute XML, esta classe está inclusa na biblioteca Google Play Services. O *Fragment*, da forma como colocado no trecho de código da Listagem 1 é a maneira mais simples de colocar um mapa em um aplicativo.

```
<fragment
  class="com.google.android.gms.maps.SupportMapFragment"
  android:id="@+id/map"
  android:layout_width="match_parent"
  android:layout_height="match_parent"/>
```

LISTAGEM - 1 Leiaute XML para construção de Mapa do GoogleMaps

Construído o *fragment* com os recursos necessários para o mapa, pode ser adicionado a qualquer *activity*, isso porque o *fragment* permite gerenciar leiautes flexíveis e independentes dentro de uma mesma *activity*. No código exibido na Listagem 2, o fragmento irá ocupar os espaços da tela, para isso é necessário estender a classe *FragmentActivity* e chamar o leiaute da Listagem 1.

```
publicclass Mapa_Activity extends FragmentActivity {
  @Override
  public void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_minha_localiz);
  }
}
```

LISTAGEM - 2 Activity com método OnCreate para criação do mapa

4.3.1 Leitura da Localização através do GPS

Para a leitura da localização é necessário adicionar a interface *LocationListener* para que seja possível a interação entre a tela de interface do sistema e o serviço de localização através do GPS. Esta interface permite também a localização através de recursos da rede de telefonia, sendo que esta costuma trazer a localização em menor tempo, também funcionar melhor em ambientes fechados

ou com cobertura (como prédios). O GPS por sua vez tem precisão melhor, além de não depender da cobertura por redes de internet ou telefonia.

No método *onCreate* da *activity* deve ser chamada uma instância da classe *LocationManager*, sobre esta instância serão baseadas as leituras da localização do aparelho, conforme a Listagem 3.

```
public class Mapa_Activity extends FragmentActivity implements
LocationListener {
    LocationManager lm = (LocationManager) this.getSystemService(
Context.LOCATION_SERVICE );
    lm.requestLocationUpdates(LocationManager.GPS_PROVIDER, 10000, 20,
(LocationListener) this );
}
}
```

LISTAGEM - 3 Código com instancia de *LocationManager*, para localização por GPS

O método *requestLocationUpdates* deve ser definido para registrar as atualizações da localização do aparelho. O primeiro parâmetro define o nome do fornecedor com o qual a localização será atualizada, o segundo é o intervalo mínimo entre uma atualização e outra, medido em milissegundos, o terceiro define a distância mínima entre a última localização e a próxima atualização, medido em metros. O último parâmetro é o método que deve ser chamado a cada atualização.

Sempre que a localização é atualizada, o método *onLocationChanged* é chamado, trazendo um objeto *location* com os dados da nova localização.

A instância de *location* possui três importantes métodos, *getLatitude()* e *getLongitude()* e *getAccuracy()*. Os dois primeiros retornam a latitude e longitude em graus. A eficiência da localização pode sofrer interferência de vários fatores, o dispositivo pode estar em local fechado, cercado de prédios ou arvores, entre outros, isso pode fazer com que a localização não seja precisa. O método *getAccuracy* retorna a precisão aproximada desta localização, em metros, caso a precisão seja menor que vinte metros, esta será persistida no banco de dados. Sem esta verificação é possível que o caminho percorrido persistido no banco de dados ficasse desorganizado. A Listagem 4 apresenta código fonte desta rotina.

```
@Override
public void onLocationChanged(Location location) {
    if (location.getAccuracy() < 20) {
```

```

        ContentValues registro = new ContentValues();
        registro.put("latitude", location.getLatitude());
        registro.put("longitudo", location.getLongitude());
        //Grava a localização no banco de dados
        dataBase.insert("rotapercorrida", null, registro);
    }
}

```

LISTAGEM - 4 Código que grava a localização no banco de dados

4.3.2 Marcadores no Mapa

Os pontos de interesse, destacados em telas como de consulta de rotas e cliente podem ser adicionados em um mapa através de objetos da classe *MarkerOptions*, esta classe adiciona um marcador no mapa. No caso da rota, deve ser adicionado uma lista de marcadores, conforme o número de clientes definidos na rota.

Para criar um *MarkerOption* basta criar um objeto *LatLng*, que armazena as coordenadas geográficas, passando a latitude e longitude em graus, conforme a Listagem 5.

```

@Override
public void onCreate(Bundle savedInstanceState) {
    MarkerOptions marker;
    LatLng posicao;
    String endereco;

    super.onCreate(savedInstanceState);

    setContentView(R.layout.activity_mostra_rota);

    mMap = ((SupportMapFragment)
    getSupportFragmentManager().findFragmentById(R.id.mapRota)).getMap();

    Intent intent = getIntent();
    _id = intent.getIntExtra("_id", 0);

    db = new DBHelper(this).getReadableDatabase();
    Cursor rota = db.rawQuery("select * from clienterota JOIN cliente
    on cliente.codigo=clienterota.codigo where clienterota.rota="+_id, null);

    while (rota.moveToNext()) {
        posicao = new
        LatLng(rota.getDouble(rota.getColumnIndex("latitude")),
        rota.getDouble(rota.getColumnIndex("longitudo")));
        endereco =
        rota.getString(rota.getColumnIndex("rua"))+","+rota.getString(rota.getColu
        mnIndex("numero"));
    }
}

```

```

        marker = new MarkerOptions();
        marker.title("Nome: "+rota.getString(rota.getColumnIndex("nome")));
        marker.snippet("Endereço: "+endereco);
        marker.position(posicao);
        mMap.addMarker(marker);
    }
}

```

LISTAGEM - 5 Exibição da rota no mapa com marcadores

A marcação do percurso percorrido, da forma como exibido na Figura 16 é realizado utilizando a classe *PolylineOptions*, onde é construída uma linha contínua, ligando todos os pontos gravados pelo rastreamento realizado pelo GPS.

Para definir a cor da linha, na propriedade *color* deve ser informada um valor estático predefinido pelo Android, neste caso *Color.RED*, ficando a linha em vermelho. A definição da largura é feita pela propriedade *width*, onde é informada a largura em pixels.

O ponto mais importante é a construção do trajeto, para isso, devem ser adicionadas as coordenadas, em graus, por meio de uma instancia de *LatLng*. Tanto pode ser adicionado uma instancia de *LatLng* por vez, quando uma lista destes objetos, como na Listagem 6.

```

PolylineOptions polylineOptions = new PolylineOptions();
ArrayList<LatLng> points = new ArrayList<LatLng>();
LatLng posicao;

polylineOptions.color(Color.RED);
polylineOptions.width(5);

while (rota.moveToNext()) {
    posicao = new
    LatLng(rota.getDouble(rota.getColumnIndex("latitude")),
    rota.getDouble(rota.getColumnIndex("longitude")));
    points.add(posicao);
}
polylineOptions.addAll(points);

```

LISTAGEM - 6 Exibição no mapa do trajeto percorrido

4.3.3 Atualizar Localização pelo Endereço

Uma forma ágil e eficiente de localizar as coordenadas geográficas de um determinado cliente é através do endereço utilizando a classe *geocoder*. Esta classe processa um determinado endereço ou outra descrição de um local em coordenadas (latitude e longitude), conforme a Listagem 7.

A consulta não é feita diretamente na plataforma Android, depende de conexão com a internet para o consumo dos serviços da Google. Como resultado da consulta é retornado uma lista de instancias da classe *Address*, ordenados pela relevância. Caso nenhum registro relevante for encontrado, a lista retorna vazia.

```
endereco = "Rua Helena J Zonta, 26, Marrecas, Francisco Beltrão";
List<Address> geoResults = geocoder.getFromLocationName(endereco, 1);
```

LISTAGEM - 7 Consulta da geolocalização pelo endereço

O método *getFromLocationName* recebe como parâmetro primeiramente uma string com o endereço ou localização, o segundo parâmetro representa o número de registros relevantes que deve ser retornado pelo método.

4.3.4 Comunicação *Socket* e GSON

A biblioteca GSON, obtida através da classe *com.google.gson.Gson*, permite converter objetos Java em representação JSON, também permite converter Strings JSON em objetos Java. Esta biblioteca agiliza a comunicação entre cliente Android e o Servidor *Socket*, pois realiza a atividade de transformar o texto enviado na comunicação em instancias programáveis.

Os dados são enviados por meio de listas, desta forma em apenas um pacote são transportados vários registros, como a string JSON apresentada na Listagem 8.

```
{"listaClientes":[
  {"_id":0,"codigo":257,"nome":"FABIENNE CHARVET
INCKOT","latitude":0.0,"longitude":0.0,"rua":"CAMPINA DAS
PEDRAS","numero":"S/N  ", "bairro":"ZONARURAL","cidade":"ARAUCARIA"},
```

```

        {"_id":0,"codigo":455,"nome":"GERALDO
SCHLOGL","latitude":0.0,"longitude":0.0,"rua":"VILA
BITUVINHA","numero":"S/N","bairro":"BITUVINHAMFA","cidade":"SAO BENTO DO
SUL"}
    ]
}

```

LISTAGEM - 8 Formato das mensagens no padrão JSON

O modelo de comunicação funciona através de serviço no servidor, que escuta as mensagens enviadas para a uma porta configurada. Ao criar um objeto *ServerSocket*, a porta que estava fechada passa a ficar aberta e ocupada pelo programa.

O programa fica parado no método *accept* da classe *ServerSocket*, esperando alguém realizar a conexão. Desta forma o método *accept* é bloqueante, segurando o processo até que alguma conexão libere. Ao receber o fluxo de dados, é possível ler as requisições e montar o fluxo de dados de resposta para o cliente. A seguir é apresentado o modelo do código implementado para o servidor.

```

ServerSocket serverSocket = new ServerSocket(4444);
Gson g = new Gson();
try {
    socket = serverSocket.accept();
    dataInputStream = new DataInputStream(socket.getInputStream());
    dataOutputStream = new DataOutputStream(socket.getOutputStream());
    ListaCliente lista = new ListaCliente();
    String json = g.toJson(lista);
    dataOutputStream.writeUTF(json);
} catch (Exception e) {
    e.printStackTrace();
}

```

LISTAGEM - 9 Modelo aplicação servidor em Socket

Na Listagem 10 é apresentado o modelo do código implementado para o cliente Android.

```

try {
    Socket servidor = new Socket(Statics.getHost(), Statics.getPort());
    DataInputStream in = new DataInputStream(servidor.getInputStream());
    DataOutputStream out = new
DataOutputStream(servidor.getOutputStream());

```



```
out.writeUTF("GETCLIEN");

out.flush();
final String resposta = in.readUTF(); //Estalinha é bloqueante

in.close();
out.close();
servidor.close();
} catch (Exception e) {
    return e.getMessage();
}
```

LISTAGEM - 10 Consulta cliente por meio de *Socket*

5 CONSIDERAÇÕES FINAIS

Neste trabalho foi desenvolvido um aplicativo para plataforma Android, com integração com um banco de dados relacional localizada em servidor remoto. Esta integração permite a utilização de recursos e mobilidade possíveis através da utilização de dispositivos móveis, com tecnologia de baixo custo de aquisição e desenvolvimento.

Utilizando o recurso de comunicação por meio da rede, com *sockets*, foi desenvolvida uma aplicação servidora responsável por enviar dados de clientes e rotas a serem realizadas para o portador do aparelho contendo o aplicativo, objeto deste trabalho. As requisições são recebidas pelo servidor, e como resposta são enviados os dados requisitados no formato JSON.

Entre as informações recebidas pelo aparelho estão dados do endereço de cada cliente, esta informação é importante para a rotina de atualização de rota, onde o aplicativo percorre a lista de clientes, buscando a localização geográfica dos mesmos, persistindo estas informações no banco de dados da aplicação.

Com a localização dos clientes e considerando as rotas cadastradas e recebidas do servidor, o aplicativo marca no mapa, fornecido pela API do Google Maps, a exata localização de cada cliente a ser visitando em cada rota.

Após aberto o aplicativo, o sistema passa a registrar os pontos geográficos percorridos. A leitura da posição é feita a cada poucos segundos, sempre que a localização mudar em uma distância de vinte metros, um novo registro é gravado no banco de dados, com a latitude, longitude, data e hora. Desta forma é possível reconstruir o caminho percorrido, considerando data e horário. Em consequência é possível identificar se o representante cumpriu de forma devida a rota previamente cadastrada.

Os dados gerados do aplicativo podem ser enviados ao servidor, através de uma rotina de sincronizar dados, da mesma forma como realizado com os clientes e rotas, mas desta vez em sentido contrário, do aparelho para o servidor, as atualizações da localização geográfica e os dados do percurso realizado são enviados. Os dados seguem no padrão utilizado neste trabalho - JSON.

As maiores dificuldades enfrentadas neste projeto, está o fato de o mapa só funcionar em um dispositivo real, desta forma não foi possível à utilização de emuladores para a execução do programa e realização de testes.

O desenvolvimento deste projeto já prevê a realização de trabalhos futuros, o envio dos dados das rotas e dados dos clientes gerados pelo aplicativo ao servidor será utilizado para os desenvolvimentos módulos, com interface Web, que permitam um controle detalhado das rotas de forma centralizada, pois até aqui, isto é possível apenas no aparelho. Outro objetivo futuro é a construção de uma rota ideal, onde seja sugerido o percurso mais rápido para o cumprimento da rota. Também estão previstos a melhoria no leiaute das telas, facilidade de uso, opções de configuração e deixar visualmente mais agradável.

REFERÊNCIAS

ANDROID. **API Android do Google Maps – Biblioteca Externa**. Disponível em: <https://developers.google.com/maps/documentation/android>. Acessado em 24/07/2014.

DILIÃO, Rui. **GPS: Global Position System**. Disponível em: <http://www.cienciaviva.com/latlong/anterior/gps.asp>. Acessado em 28/07/2014.

ANDROID. **Getting Started**. Disponível em: <https://developers.google.com/maps/documentation/android/start>. Acessado em 28/07/2014.

GIL, Antônio Carlos. **Métodos e técnicas de pesquisa**. São Paulo: Editora Atlas, 1997.

GLOBO. **No mundo, 4 a cada 5 smartphones vendidos rodam sistema Android**. Disponível em: <http://g1.globo.com/tecnologia/noticia/2013/10/no-mundo-4-cada-5-smartphones-vendidos-rodam-o-sistema-android.html>. Acessado em 01/01/2015.

IBGE. **Como funciona o GPS**. Disponível em: <http://www.ibge.gov.br/ibgeteen/atlasescolar/apresentacoes/tecnicas.sw>. Acessado em 28/07/2014.

IBGE. **Sistema de Posicionamento Global – GPS**. Disponível em: <http://atlasescolar.ibge.gov.br/conceitos-gerais/conceitos-e-tecnicas>. Acessado em 23/07/2014. Acessado em 06/09/2014.

ECLIPSE. **IDE Eclipse**. Disponível em: <https://eclipse.org/juno/>. Acessado em 20/10/2013.

LACHETA, Ricardo R. **Google Android: aprenda a criar aplicações para dispositivos móveis com o Android SDK**. São Paulo: Novatec Editora, 2010.

KAUARK, Fabiana; MANHÃES, F. C.; MEDEIROS, C. H. **Metodologia de pesquisa:** Um guia prático. Itabuna: Editora Via Litterarum, 2010.

CIENCIAVIVA. **Latitude e Longitude: Instrumentos e medição.** Disponível em: <http://www.cienciaviva.pt/latlong/anterior/gps.asp>. Acessado em: 22/07/2014.

MARCONI, M. A.; LAKATOS, E. M. **Técnicas de pesquisa:** Planejamento e execução de pesquisas, amostragens e técnicas de pesquisa, elaboração, análise e interpretação de dados. São Paulo: Editora Atlas, 1996.

PALMEIRA, Thiago V. V. **Conhecendo o Eclipse** – Uma apresentação detalhada da IDE. Disponível em: <http://www.devmedia.com.br/conhecendo-o-eclipse-uma-apresentacao-detalhada-da-ide/25589>. Acessado em 06/09/2014.

POSTGRES. **PostgreSQL.** Disponível em: <http://www.postgresql.org/about/>. Acessado em 06/09/2014.

PRESSMAN, Roger S. **Engenharia de Software.** Nova Iorque: Editora McGraw-Hill, 2006.

TECMUNDO. **Como funciona o GPS?** Disponível em: <http://www.tecmundo.com.br/gps/2562-como-funciona-o-gps-.htm>. Acessado em 06/09/2014.

OGLIARI, Ricardo da Silva.; BRITO, Robison Cris. **Android - do básico ao avançado.** Rio de Janeiro: Editora Ciência Moderna, 2014.



Presidência da República
Casa Civil
Subchefia para Assuntos Jurídicos

LEI Nº 9.610, DE 19 DE FEVEREIRO DE 1998.

Mensagem de veto

Altera, atualiza e consolida a legislação sobre direitos autorais e dá outras providências.

O PRESIDENTE DA REPÚBLICA Faço saber que o Congresso Nacional decreta e eu sanciono a seguinte Lei:

Título I

Disposições Preliminares

Art. 1º Esta Lei regula os direitos autorais, entendendo-se sob esta denominação os direitos de autor e os que lhes são conexos.

Art. 2º Os estrangeiros domiciliados no exterior gozarão da proteção assegurada nos acordos, convenções e tratados em vigor no Brasil.

Parágrafo único. Aplica-se o disposto nesta Lei aos nacionais ou pessoas domiciliadas em país que assegure aos brasileiros ou pessoas domiciliadas no Brasil a reciprocidade na proteção aos direitos autorais ou equivalentes.

Art. 3º Os direitos autorais reputam-se, para os efeitos legais, bens móveis.

Art. 4º Interpretam-se restritivamente os negócios jurídicos sobre os direitos autorais.

Art. 5º Para os efeitos desta Lei, considera-se:

I - publicação - o oferecimento de obra literária, artística ou científica ao conhecimento do público, com o consentimento do autor, ou de qualquer outro titular de direito de autor, por qualquer forma ou processo;

II - transmissão ou emissão - a difusão de sons ou de sons e imagens, por meio de ondas radioelétricas; sinais de satélite; fio, cabo ou outro condutor; meios óticos ou qualquer outro processo eletromagnético;

III - retransmissão - a emissão simultânea da transmissão de uma empresa por outra;

IV - distribuição - a colocação à disposição do público do original ou cópia de obras literárias, artísticas ou científicas, interpretações ou execuções fixadas e fonogramas, mediante a venda, locação ou qualquer outra forma de transferência de propriedade ou posse;

V - comunicação ao público - ato mediante o qual a obra é colocada ao alcance do público, por qualquer meio ou procedimento e que não consista na distribuição de exemplares;

VI - reprodução - a cópia de um ou vários exemplares de uma obra literária, artística ou científica ou de um fonograma, de qualquer forma tangível, incluindo qualquer armazenamento permanente ou temporário por meios eletrônicos ou qualquer outro meio de fixação que venha a ser desenvolvido;

VII - contrafação - a reprodução não autorizada;

VIII - obra:

a) em co-autoria - quando é criada em comum, por dois ou mais autores;

b) anônima - quando não se indica o nome do autor, por sua vontade ou por ser desconhecido;

c) pseudônima - quando o autor se oculta sob nome suposto;

d) inédita - a que não haja sido objeto de publicação;

e) póstuma - a que se publique após a morte do autor;

f) originária - a criação primígena;

g) derivada - a que, constituindo criação intelectual nova, resulta da transformação de obra originária;

h) coletiva - a criada por iniciativa, organização e responsabilidade de uma pessoa física ou jurídica, que a publica sob seu nome ou marca e que é constituída pela participação de diferentes autores, cujas contribuições se fundem numa criação autônoma;

i) audiovisual - a que resulta da fixação de imagens com ou sem som, que tenha a finalidade de criar, por meio de sua reprodução, a impressão de movimento, independentemente dos processos de sua captação, do suporte usado inicial ou posteriormente para fixá-lo, bem como dos meios utilizados para sua veiculação;

IX - fonograma - toda fixação de sons de uma execução ou interpretação ou de outros sons, ou de uma representação de sons que não seja uma fixação incluída em uma obra audiovisual;

X - editor - a pessoa física ou jurídica à qual se atribui o direito exclusivo de reprodução da obra e o dever de divulgá-la, nos limites previstos no contrato de edição;

XI - produtor - a pessoa física ou jurídica que toma a iniciativa e tem a responsabilidade econômica da primeira fixação do fonograma ou da obra audiovisual, qualquer que seja a natureza do suporte utilizado;

XII - radiodifusão - a transmissão sem fio, inclusive por satélites, de sons ou imagens e sons ou das representações desses, para recepção ao público e a transmissão de sinais codificados, quando os meios de decodificação sejam oferecidos ao público pelo organismo de radiodifusão ou com seu consentimento;

XIII - artistas intérpretes ou executantes - todos os atores, cantores, músicos, bailarinos ou outras pessoas que representem um papel, cantem, recitem, declamem, interpretem ou executem em qualquer forma obras literárias ou artísticas ou expressões do folclore.

Art. 6º Não serão de domínio da União, dos Estados, do Distrito Federal ou dos Municípios as obras por eles simplesmente subvencionadas.