

# Captura e análise por similaridade de ciclorotas a partir de dispositivos móveis

Vanessa Malvar Santana<sup>\*</sup>  
Instituto Federal de Educação, Ciência e  
Tecnologia da Bahia  
vanessamalvar@ifba.edu.br

Pablo Vieira Florentino<sup>†</sup>  
Instituto Federal de Educação, Ciência e  
Tecnologia da Bahia  
pablovf@ifba.edu.br

## Resumo

*Identificar as principais rotas utilizadas por ciclistas dentro da cidade de Salvador faz com que investimentos em infraestrutura de transporte para ciclistas sejam aplicados na cidade atendendo as demandas e evitando prejuízos. Este projeto tem como objetivo criar uma base de dados que apresente informações de rotas utilizadas por ciclistas dentro da cidade de Salvador, bem como entender a dinâmica dos ciclistas. Um aplicativo móvel com tecnologia Android foi desenvolvido para a captura de rotas e identificação de encontros e sobreposições utilizando consultas por similaridade.*

## Termos gerais

Mobilidade, Tecnologia móvel, Computação Ubíqua, Consultas por Similaridade

## Palavras-Chave

Localização, GPS, Android

## 1. INTRODUÇÃO

O atual cenário dos grandes centros urbanos de desenvolvimento acelerado e desordenado, que associados à falta de mobilidade urbana, gera efeitos prejudiciais à ordem social. Embora esses problemas urbanos sejam reconhecidos perante o governo e sociedade, ainda é grande o desafio de criar mecanismos de mobilidade que acompanhem o ritmo de crescimento populacional e consequentemente de transporte nas grandes cidades.

De acordo com censo de 2010, realizado pelo IBGE (Instituto Brasileiro de Geografia e Estatística), a população urbana do Brasil passou de 81% (dados do censo de 2000) para 84% [2], e dados da DENATRAN (Departamento Nacional

<sup>\*</sup>Estudante orientando do curso de Análise e Desenvolvimento de Sistemas.

<sup>†</sup>Professor Orientador do curso de Análise e Desenvolvimento de Sistemas.

de Trânsito) de 2010 afirmam que o Brasil fechou o ano de 2010 com quase 65 milhões de veículos registrados. Em dez anos, o aumento acumulado é de 119% [3]. Diante desses desafios verificamos que cada vez mais políticas de mobilidade urbana devem ser repensadas para atender a demanda atual e das próximas décadas.

Nas regiões metropolitanas acontece um fenômeno denominado de espraiamento urbano [17], no qual núcleos são formados em volta do centro urbano e consequentemente as distâncias entre eles se tornam cada vez maiores. Nas regiões metropolitanas é grande o número de deslocamentos realizados diariamente, devido à dimensão populacional bem como a distância enfrentada pelos cidadãos de suas residências até o local de trabalho ou ambientes de lazer, escolas, hospitais, universidades e faculdades.

Nessas condições a população possui três opções mais significativas para se locomover, levando em consideração o fator econômico: a utilização do transporte coletivo público, a realização do percurso a pé ou o uso da bicicleta. Promover infraestrutura urbana de qualidade, de forma a atender a todas essas locomoções com equidade, é um desafio social diante de um cenário de falta de planos de mobilidade, conjugado com o aumento populacional das grandes cidades. A segunda opção nem sempre é viável pelo fator distância e diante dessas situações, um transporte alternativo aos motorizados, para pequenas e médias distâncias, é a bicicleta. Muitos cidadãos preferem economizar o valor que se destinaria às passagens/bilhetes dos transportes coletivos, e ao consumo de gasolina e derivados, necessários na utilização e manutenção dos veículos motorizados individuais, e optam por um meio de transporte mais barato ou sem custo.

A Constituição Brasileira estabelece princípios e ferramentas para que os municípios possam planejar um sistema de transporte que atenda aos diferentes públicos da população. Como parte de uma política de mobilidade, acessibilidade e inclusão social, viabilizar e dar suporte ao transporte – a pé ou por meio de bicicletas – é algo a ser considerado, dentre tantos métodos, como uma alternativa.

Na atual situação da cidade de Salvador, isso não é diferente. Além das questões citadas anteriormente, a ausência de infraestrutura para os usuários de bicicletas dentro da cidade impede que o cidadão usuário de bicicleta tenha o seu espaço definido no meio público, acarretando em um âmbito maior, um desfavorecimento na hora de se locomover.

ver com segurança dentro da cidade. A acessibilidade e a mobilidade estão previstas em lei no Brasil, como se pode observar na lei de Política Nacional de Mobilidade Urbana (número 12.587/2012), que foi sancionada em 3 de Janeiro de 2012:

*A Política Nacional de Mobilidade Urbana é instrumento da política de desenvolvimento urbano de que tratam o inciso XX do artigo 21 e o artigo 182 da Constituição Federal, que tem por objetivo a integração entre os diferentes modos de transporte e a melhoria da acessibilidade e mobilidade das pessoas e cargas no território do município [4].*

Esta lei tem como uma de suas diretrizes a priorização de transportes não-motorizados (por exemplo, caminhada e bicicleta) ou coletivos. Além disso, é fundamental que as mudanças provenientes das ações de política de mobilidade urbana levem em consideração as necessidades e especificidades de cada município. E para que isso ocorra é necessário que existam informações pertinentes às demandas de cada cidade. Assim, este projeto tem como finalidade criar uma base de dados que possibilite este estudo dentro da cidade de Salvador tendo como alvo os usuários de bicicleta da região, bem como entender a dinâmica desses utilizando técnicas capazes de identificar rotas, encontros e sobreposições.

Compreender a dinâmica dos usuários de bicicleta dentro da cidade de Salvador consiste em um estudo a cerca das principais rotas em que os mesmos percorrem, em quais rotas acontece fluxo de ciclistas, quais rotas se sobrepõem, quais os pontos de encontro entre as rotas utilizadas. Após obter resultados de tal estudo é possível que investimentos sejam norteados de forma a atender a demanda da cidade e da população, de maneira específica e eficaz.

Este projeto divide-se em seções. A seção 2 apresenta os trabalhos relacionados ao objeto deste projeto. A seção 3 aborda o referencial teórico. A seção 4 refere-se às informações do aplicativo desenvolvido, bem como a sua modelagem. A seção 5 refere-se à metodologia dos testes realizados. Na seção 6 é feita uma proposta de análise dos dados por consultas de similaridade. Nas seções seguintes 7, 8 e 9 são apresentados a conclusão, as referências e por último os anexos.

## 2. TRABALHOS RELACIONADOS

Nesta seção serão apresentadas duas pesquisas que serviram de apoio e nortearam o desenvolvimento deste projeto. A primeira pesquisa se trata de uma tabela comparativa realizada com alguns aplicativos móveis já disponibilizados para o uso, e a segunda pesquisa apresenta o projeto realizado pela CONDER (Companhia de Desenvolvimento Urbano do Estado da Bahia).

### 2.1 Grade de aplicativos observados

Uma pesquisa comparativa foi realizada com o intuito de conhecer melhor as funcionalidades e as tecnologias presentes nas aplicações de localização, nacionais e internacionais, bem como identificar as preferências e necessidades dos usuários de bicicleta. Informações que compõem a grade comparativa foram retiradas observando os próprios sites e repositório das aplicações. Outra tentativa de obtenção de informações se deu através da aplicação de um questionário web

submetido aos desenvolvedores e/ou distribuidores que disponibilizaram contatos em seus respectivos sites. Contudo, não houve resposta por parte dos casos observados.

O total de 13 aplicativos fizeram parte desta pesquisa (Tabela 1). A maioria dos aplicativos escolhidos possui funcionalidades em comum, como por exemplo, localização em um mapa, seja para localizar um veículo, um pedestre ou um usuário de bicicleta. Existem diversos aplicativos móveis para usuários de bicicleta disponíveis no mercado, gratuitos ou não, e ao observar a semelhança entre eles percebe-se que as funcionalidades mais presentes são: criação de rota, sugestão de rota, cálculo de velocidade média, distância, tempo e altimetria. A diferença se faz nas funções secundárias e na tecnologia utilizada.

Dentre os aplicativos citados na tabela, algumas funcionalidades secundárias se destacam, como por exemplo no aplicativo *Waze*, em que é possível verificar a situação do trânsito de determinada rua ou região em tempo real através de sinalizações feitas por outros usuários. Outro exemplo são as aplicações de apoio aos ciclistas, que identificam pontos de aluguel, bicicletários e oficinas mais próximos do usuário, como é observado no aplicativo *Barclays Bikes*. Aplicativos como o *RunKeeper* e *CycleStreets* permitem que o usuários publiquem fotos durante o percurso. O app *Cibi-me* escolhe rotas que apresentam percursos mais planos para os usuários.

A análise das aplicações pesquisadas foram de fundamental importância para o desenvolvimento do aplicativo desenvolvido neste projeto, pois esta pesquisa pôde orientar as decisões de projeto e a modelagem do sistema para melhor atender as demandas e aos objetivos.

### 2.2 Pesquisa CONDER

O Programa Cidade-Bicicleta é um programa de mobilidade não motorizada para o Estado da Bahia, promovido pela CONDER no ano de 2009 e motivada pelas realidades urbanísticas da metrópole de Salvador e Lauro de Freitas. O projeto teve como objetivos desenvolver e implementar soluções de mobilidade não motorizada, envolvendo um conjunto de ações e projetos cicloviários que proporcionem democratização do espaço urbano, garantindo inclusão social e territorial, economicidade, sustentabilidade ambiental e cultural.

Foram aplicados 1367 questionários aos ciclistas que trafegam em vias de influência dos pólos geradores de demanda, em 18 pontos de pesquisa diferentes. E aos usuários de transporte coletivos que utilizam as estações e terminais para alcançar seus destinos, foram 2506 questionários aplicados. As perguntas contidas nestes questionários permitiram analisar o perfil socioeconômico, as características dos percursos e as motivações de uso da bicicleta.

De acordo com o gráfico representado na Figura 1, 80% dos entrevistados apontam o tráfego perigoso como a principal característica do percurso ciclistico. Outras características também foram citadas, dentre elas a pavimentação inadequada, o excesso de ladeiras, a sinalização precária e a ausência de estacionamentos.

A pesquisa do perfil socioeconômico aponta que 94% dos

Tabela 1: Grade de aplicativos observados

Nome	Pontos de Localização	Rotas	Tempo	Distância	Velocidade	Altimetria	Outras funcionalidades
Waze	Sim	Sim	Sim	Sim	Sim	Sim	Tráfego em tempo real.
Mapa Cicloviário do Rio de Janeiro	Sim	Não	Não	Não	Não	Não	Localização de ciclovias, bicicletários, pontos de aluguel e oficinas.
Cibi.me	Sim	Sim	Sim	Sim	Não	Sim	Rota com percurso mais plano.
Geovelo	Sim	Sim	Sim	Sim	Sim	Sim	Percurso com o nome das ruas
RunKeeper	Sim	Sim	Sim	Sim	Sim	Não	Postar fotos e adicionar música.
BUS-U	Sim	Não	Sim	Não	Não	Não	Identificar horário e local que determinada linha de ônibus passa em ponto de ônibus específico.
BikeMap	Sim	Sim	Sim	Sim	Sim	Sim	Registrar pontos interessantes do percurso.
BikeCityGuide	Sim	Sim	*	*	*	*	Ruas movimentadas são evitadas.
Brandenburg App	Sim	Sim	Sim	Sim	Não	Não	Agendar passeios e pesquisar atrações de Berlim.
CycleStreets	Sim	Sim	Sim	Sim	Sim	Sim	Postar fotos e evitar subidas.
Bike Hub Cycle Journey planner	Sim	Sim	Sim	Sim	Sim	Não	Rota recalculada automaticamente
Barclays Bikes	Sim	Sim	Sim	Sim	Não	Não	Encontrar pontos mais próximos para "ancorar" a bike.
ViewRanger Outdoors GPS	Sim	Sim	Sim	Sim	Sim	Sim	Mapa de esqui e de transporte.

item[\*] Informação não encontrada.

entrevistados ciclistas são homens, 47% estão na faixa de 18 a 35 anos e 48% possui uma renda de até 1 salário mínimo. Diante dessa realidade o perfil do ciclista mudou e continua mudando ao longo dos anos, cada vez mais os fatores que influenciam a decisão de usar bicicleta deixam de ser apenas por causa do preço elevado das passagens ou do transporte público coletivo insuficiente, e passam a ser por uma forma alternativa, não-motorizada, de baixo impacto para a cidade e que não é afetado por congestionamentos.

### 3. REFERENCIAL TEÓRICO

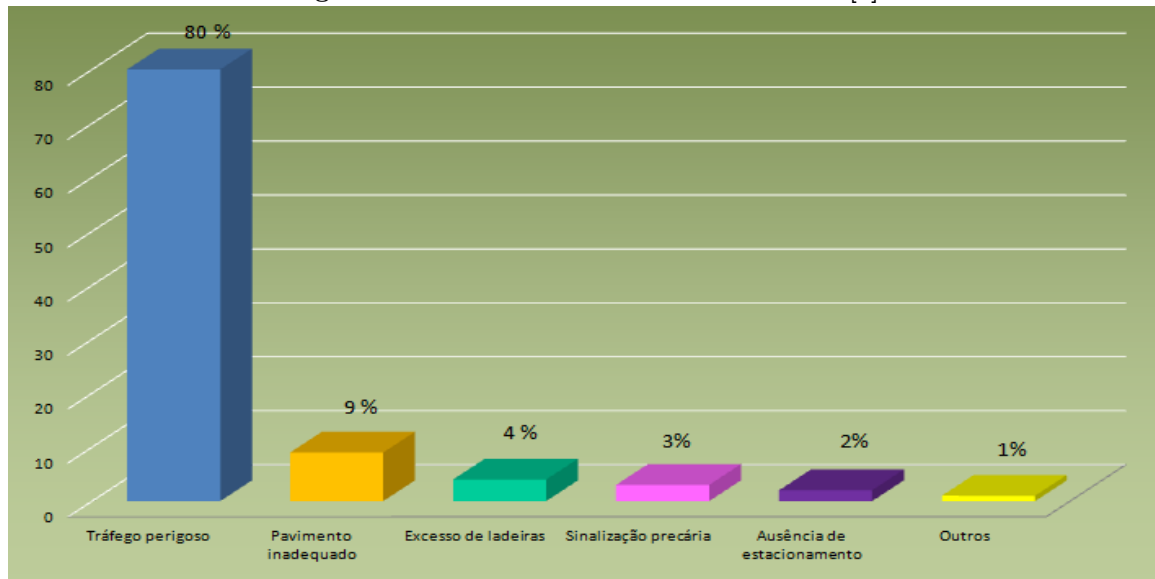
Neste tópico, estão apresentadas as informações e os conceitos inerentes ao trabalho e as análises que nortearam o seu desenvolvimento. De início, a seção 3.1 apresenta características da computação na atualidade e da Computação Ubíqua. Na seção 3.2 é abordado o tema Tecnologia de Localização e como subtema está definido GPS e A-GPS e suas vantagens. E por último, a seção 3.3 apresenta benefícios de

uma Interface amigável ao usuário.

#### 3.1 Computação Ubíqua

O termo Computação Ubíqua foi definido pela primeira vez no artigo *The Computer for the 21st Century* (O Computador do Século 21), em 1988 pelo cientista americano Mark Weiser, que hoje é considerado o pai da computação ubíqua [18]. A partir deste artigo um novo panorama surgiu na computação, no qual computadores seriam inteligentes o suficiente para identificar os mais diversos sinais emitidos por usuários e/ou computadores (sinais sonoros, sinais gestuais, movimentação e calor), entretanto, a visibilidade dos processos por detrás da computação é a menor possível na perspectiva do usuário. Weiser afirmou em seu artigo que as tecnologias mais profundas são aquelas que desaparecem, que se dissipam nas coisas do dia-a-dia até que se tornem indistinguíveis.

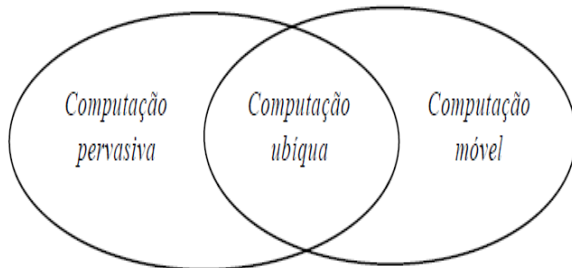
Figura 1: Gráfico *Característica do Percurso* [9]



Weiser pensou a computação ubíqua como uma oportunidade de melhorar a maneira como os usuários se relacionavam com a computação. Na era *Mainframe*, majoritariamente, a relação era de um computador com muitos usuários. Na era PC a relação era unilateral, um computador para um usuário, e na terceira e atual era é a computação ubíqua, na qual uma pessoa se relaciona com vários computadores em diferentes momentos, lugares, dispositivos e situações. Dessa forma a computação ubíqua aponta um presente e um futuro em que computadores estarão embarcados nos lugares mais triviais do cotidiano e a interação será cada vez mais natural para o usuário [12].

A computação ubíqua apresenta características de duas outras áreas da computação, a computação móvel e a computação pervasiva (Figura 2). A computação móvel permite que a computação e os serviços computacionais acompanhem o usuário independente de sua localização, e a computação pervasiva sustenta a ideia de que tudo a sua volta pode ser um dispositivo embarcado capaz de se comunicar com outros dispositivos e interagir com as informações do ambiente. Portanto, a alta mobilidade da computação móvel juntamente com o alto grau de dispositivos embarcados propostos pela computação pervasiva, contribuíram para o desenvolvimento da computação ubíqua.

Figura 2: Computação Ubíqua



### 3.1.1 Fator limitante

A interação ubíqua acontece entre dispositivos com diferentes sistemas operacionais, interfaces e comunicação em rede. Mesmo com todo o avanço tecnológico ainda existem limitações que a computação ubíqua procura estudar e amenizar os seus efeitos. Entre estas limitações, o consumo de energia juntamente com a capacidade de obter essa energia através da bateria. A tecnologia da bateria é limitante, pois o seu desenvolvimento não acompanha o acelerado desenvolvimento dos dispositivos móveis, das aplicações e serviços computacionais.

Um dos ramos da computação ubíqua pesquisa sistemas e protocolos de rede capazes de limitar o consumo dessa energia utilizada na comunicação e na transmissão de dados, pois a maior perda de energia acontece justamente durante a comunicação sem fio.

### 3.2 Tecnologia de Localização

A Tecnologia de localização tem sido uma parte importante da computação ubíqua nas últimas décadas [12]. A capacidade de determinar a localização de um alvo permite que uma variedade de aplicações ubíquas utilizem desse mecanismo para agregar valor aos serviços que eles oferecem. As aplicações chamadas de *location-aware* têm crescido em sofisticação, utilidade e popularidade, muito desse crescimento se deu com o avanço e confiabilidade dos usuários quanto ao *Global Positioning System* (GPS), um sistema de rastreamento de localização ao ar livre mais popular do mundo.

Visando a importância da informação de localização, não existe uma única tecnologia precisa, de baixo custo e de fácil implantação, ao contrário, existe uma coleção de tecnologias, nas quais cada uma deve ser utilizada de forma que atenda os requisitos de precisão da aplicação em específico. Uma abordagem mais realista é combinar várias tecnologias existentes para um sistema integrado de localização.

### 3.2.1 GPS e A-GPS

Inicialmente o GPS era utilizado para fins militares, mas atualmente muitas outras aplicações civis e de consumo utilizam os seus serviços. As aplicações GPS civis possuem uma precisão média de 10 metros ao ar livre, mas em situações de áreas montanhosas, altos edifícios, ambientes fechados e subsolos, essa precisão diminui, degradando a qualidade do sinal.

O GPS busca a melhor posição de acordo com os 24 satélites disponíveis na órbita da Terra. Estes satélites ficam constantemente enviando sinais de rádio que contém informações de sua própria localização e o tempo de emissão do sinal, para qualquer receptor de GPS na superfície da Terra. Dessa forma, o aparelho se comunica com o satélite, que por sua vez, faz a comunicação com outros satélites e realiza o sistema de trilateração para obter o resultado da posição do receptor. O cálculo é feito através do cruzamento das informações dos satélites, devido a diferença de distância entre eles. Pelo menos 4 satélites são necessários para esta verificação.

Os serviços de localização permitem que aplicativos móveis dependentes de localização usem informações das torres de celulares, Wi-fi e do GPS para determinar a localização aproximada. A Rede WI-FI utiliza a posição das torres das operadoras de celular para encontrar a localização. Já o GPS, como foi explanado anteriormente, busca a melhor posição de acordo com os satélites disponíveis na órbita da Terra.

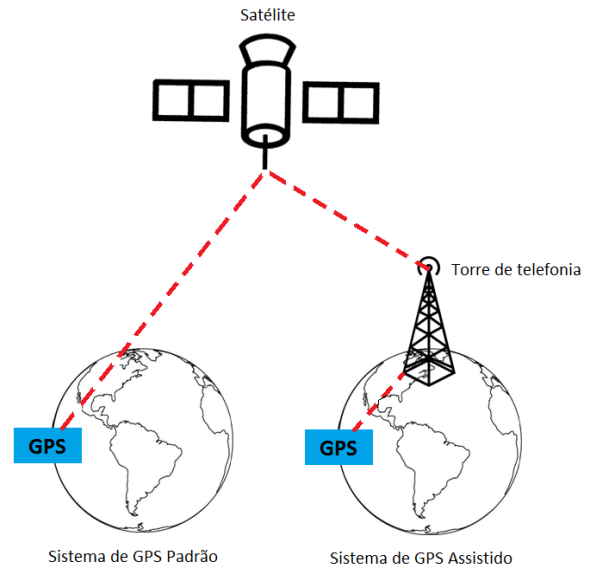
O processo do GPS costuma ser lento devido a conexão GPS-satélite. Em locais abertos dificilmente o GPS apresenta erros ou desvios muito elevados, porém este não é um cenário frequentes nas cidades. Uma maneira eficiente de resolver este tipo de problema é a utilização da rede 2G/3G, que agregado ao GPS possibilita uma diminuição no erro de precisão do GPS. Com o auxílio da rede de dados, o cálculo para encontrar a posição real do usuário é feito utilizando a antena telefônica na qual o usuário estar conectado. Este tipo de processo chamamos de GPS assistido ou A-GPS (*Assisted Global Positioning System*), é um auxiliar do GPS, pois acelera a comunicação do celular com o satélite através das torres de celulares que possuem receptores de GPS, tornando a funcionalidade mais rápida, evitando a espera pela conexão dos satélites por mais tempo, além de diminuir o grau de desvio e consequentemente apontando uma melhor precisão na localização.

### 3.3 Interface Amigável

A interface usuário-computador, também denominada interface com o usuário, interface homem-computador ou simplesmente interface, é um fator determinante para a aceitação de produtos interativos, uma vez que proporciona o elo de comunicação entre os usuários e um dado sistema computacional, exercendo influência no modo como o usuário visualiza e entende a funcionalidade de um sistema [7]. Dessa forma é importante que as interfaces se comportem de uma maneira confortável para o usuário [16].

No âmbito da qualidade de software, a usabilidade dos produtos é um fator competitivo no mercado. Segundo Nielsen [15], a usabilidade nada mais é do que a associação de questão como a facilidade de aprendizado, eficiência na pro-

Figura 3: Comunicação GPS-AGPS



atividade, facilidade de memorização, tolerância a erros e satisfação do usuário.

Com a evolução dos dispositivos móveis e a variedade de ferramentas e bibliotecas disponíveis atualmente, os desenvolvedores de aplicativos móveis cada vez mais se preocupam com a aparência dos seus aplicativos, bem como com a eficiência e a facilidade de interação do usuário com a interface. Maneiras de aplicar os conceitos de usabilidade em um aplicativo móvel e facilitar a interação com usuário são, dentre outros fatores, reduzir o número de cliques do usuário para executar as funções, estudar um layout na qual as principais informações sejam facilmente identificadas e organizar as funcionalidades.

## 4. O APLICATIVO DESENVOLVIDO

Nesta seção será apresentado o aplicativo móvel desenvolvido, que tem como finalidades disponibilizar informações baseadas nas rotas realizadas por ciclistas da cidade de Salvador, podendo auxiliar nas futuras políticas de investimentos em infraestrutura para transportes urbanos, além de colaborar com as pesquisas de mobilidade urbana já existentes na cidade.

O ambiente utilizado no desenvolvimento deste aplicativo foi o pacote de ferramentas de desenvolvimento Android SDK (*Software Development Kit*) que permite criar e testar aplicações Android. Os principais componentes deste pacote são o Eclipse e os plugins ADT (Android Development Tools) [13]. A linguagem de desenvolvimento do Android é o Java, e o Eclipse é a IDE (*Integrated Development Environment*) que permite criar e depurar o código Java da aplicação, além de gerenciar os recursos utilizados na implementação. O ADT é integrado ao Eclipse para facilitar a comunicação e o acesso as ferramentas do SDK através de ícones e menus. O SDK possui ferramentas necessárias para simular, depurar, visualizar os aplicativos, sejam eles em dispositivos físicos ou virtuais. Para este projeto foi utilizada a versão v22.0.1 do Android SDK.

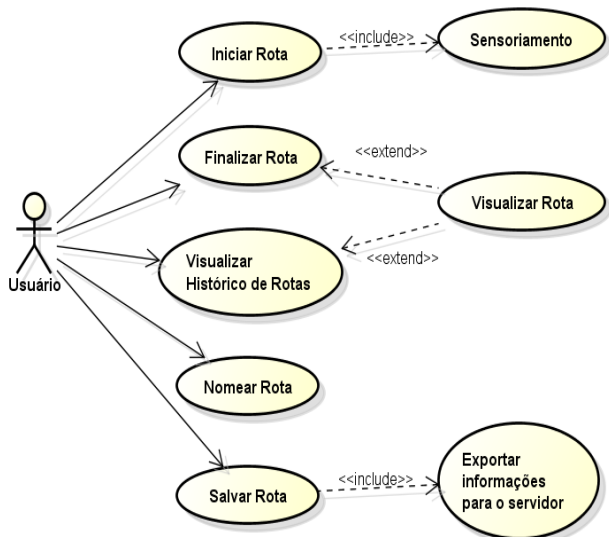
Nesta seção serão apresentados a modelagem do sistema, apresentando os artefatos elaborados de acordo com o padrão UML (*Unified Modeling Language*), além dos serviços utilizados pela aplicação e o seu funcionamento.

### 4.1 Especificação dos Casos de Uso

Esta seção descreve as funcionalidades que o sistema deve realizar através de casos de uso. Um caso de uso é a especificação de uma sequência completa de interações entre um sistema e um ou mais agentes externos a esse sistema [8]. A UML fornece a notação de diagramas de casos de uso para ilustrar os nomes dos casos de uso e dos atores, bem como os relacionamentos entre eles.

O Diagrama Geral dos Casos de Uso deste projeto é representado pela Figura 4, e a Tabela 2 apresenta, em forma de narrativa particionada, as descrições desses casos de uso. Esta forma de estruturação da narrativa tem como objetivo separar as ações do ator e as reações do sistema, já que o aplicativo responde requisições a partir da interação com o usuário.

Figura 4: Diagrama Geral dos Casos de Uso



### 4.2 Arquitetura

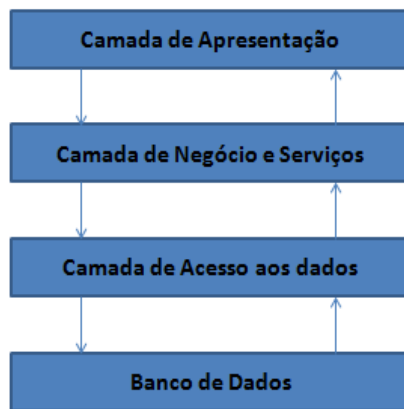
A camada do topo da arquitetura Android é o conjunto de todos os aplicativos nativos (navegador de internet, correio eletrônico, calendário, contatos, entre outros) e os aplicativos adquiridos pelo usuário através de download (Ver arquitetura Android completa na seção 9.2).

Diante dos fundamentos da Arquitetura de Software, a arquitetura prescritiva do aplicativo desenvolvido consiste em uma combinação do estilo arquitetural em camadas Cliente-Servidor e do padrão arquitetural MVC (*Model View Controller*). O estilo Cliente-Servidor separa a aplicação em camadas ordenadas, no qual um programa de uma camada pode solicitar serviços de uma camada inferior, e por outro lado o padrão MVC promove a separação e a independência dos processos de desenvolvimento, da informação manipulada entre o programa e o usuário (Figura 5).

Tabela 2: Narrativa Particionada dos Casos de Uso

Usuário	Aplicativo
Inicia Rota	Apresenta na tela do usuário os pontos de latitude e longitude, além da distância e velocidade máxima e mínima durante o percurso.
Visualiza Rotas	Apresenta uma lista com todas as rotas já realizadas pelo usuário.
Finaliza Rota	Apresenta o percurso desenhado no mapa, marcando o início e o fim da rota. E permite que o usuário nomeie e salve a rota visualizada.
Nomea Rota	Armazena o nome sugerido pelo usuário no banco de dados local.
Salva Rota	Envia todas as informações da Rota e dos Pontos capturados para o Servidor.

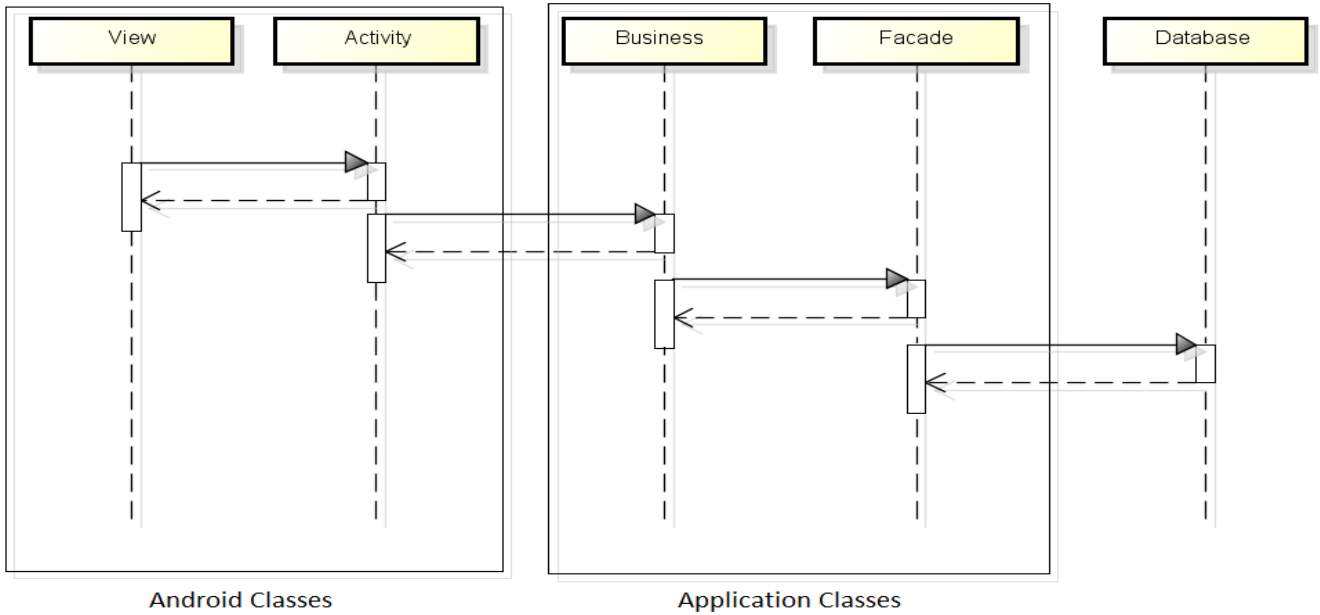
Figura 5: Comunicação entre as camadas



Semelhante a abordagem utilizada na arquitetura de aplicação Java para Web, a arquitetura para esta aplicação Android também foi organizada em camadas, levando em consideração as necessidades do Android. Dessa forma, as telas em XML (*View*) acionam as *Activities*, que implementam métodos para responder as demandas das telas da aplicação e enviam mensagens para os objetos de negócio (*Business*), os quais tem a função de executar as regras de negócio e se comunicar com o *Facade*, que seria a interface entre a aplicação e o Banco de Dados (Figura 6).

A camada *View* é a responsável pela comunicação com o usuário através de *layouts* de arquivos .xml. Em Android a classe *R.java* faz a interação entre o código java das *Activities* e a código XML. A camada *Business* serve para separar o negócio da aplicação dos componentes de tela (*Activities*). E por último o *Facade* - assim chamado por ter como função simplificar a utilização de um subsistema - apresentando uma interface mais razoável, a qual é responsável pela comunicação com o Banco de Dados.

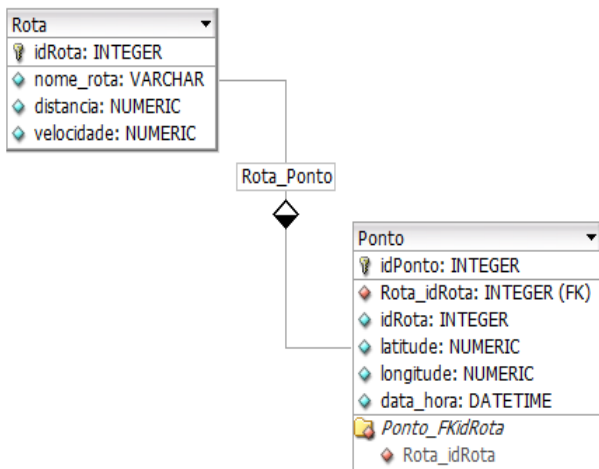
Figura 6: Arquitetura Conceitual



### 4.3 Persistências dos Dados

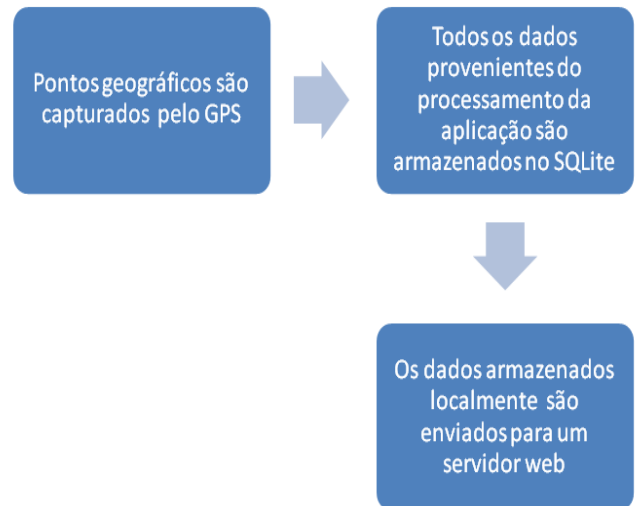
O Android oferece suporte nativo ao banco de dados SQLite e para atender as funcionalidades deste aplicativo apenas duas tabelas são necessárias: a tabela Rota e a tabela Ponto. A relação de dependência entre essas tabelas é de 1-n, ou seja, uma rota pode estar associada a vários pontos e um ponto referencia apenas uma rota. A tabela **Rota** possui 4 atributos, o identificador como chave primária, o nome da rota que é escolhido pelo usuário, a distância percorrida e a velocidade média do percurso. Já a tabela **Ponto** tem um identificador como chave primária, e uma referência para o identificador da tabela Rota como chave estrangeira, além desses possui os atributos latitude, longitude e datahora. A Figura 7 apresenta o modelo do banco de dados relacional aplicado neste projeto.

Figura 7: Modelo de Banco de Dados



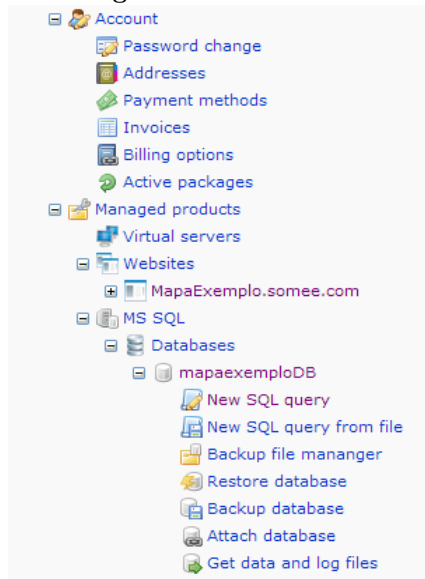
As rotas e pontos são armazenados no banco de dados local (SQLite) e depois enviado para um *WebService*, que é uma tecnologia capaz de integrar várias plataformas diferentes utilizando uma única base de dados centralizada na internet. Com as informações do usuário no serviço de armazenamento web é possível aplicar consultas em SQL e analisar informações sobre as rotas criadas pelos usuários (Figura 9). O fluxo dos dados na aplicação é representado pelo diagrama na Figura 8.

Figura 8: Fluxo de dados



Não faz parte do escopo deste projeto apresentar informações pertinentes ao usuário ou alguma informação socioeconômica. Na seção 7.1 serão apresentadas algumas das

Figura 9: Webservice



funcionalidades planejadas para trabalhos futuros.

#### 4.4 Módulos e Diagrama de Classes

Nesta seção será apresentada a organização e interação entre os módulos e as classes do aplicativo desenvolvido neste projeto. A aplicação foi organizada em 6 módulos, como podemos ver na Figura 10.

De acordo com a figura, a cor verde representa os módulos desenvolvidos e a cor vermelha representa os módulos proprietários, que foram desenvolvidos de forma privada por terceiros. Neste projeto apenas o módulo **app.arquitetura** é considerado proprietário e foi adicionado a aplicação com base no princípio de reutilização da Engenharia de Software, que tem como um dos objetivos garantir a redução de esforços e custos no processo [5]. Segue a descrição e as peculiaridades de cada módulo:

- **como.example.app.arquitetura**: Este módulo funciona como um framework de persistência, que é formado por um conjunto de interfaces e classes que colaboram para fornecer serviços de um subsistema lógico, neste caso o serviço *WebService*. Este módulo está subdividido em 4 partes: arquitetura.base, arquitetura.exception, arquitetura.framework e arquitetura.listener. O arquitetura.base possui 4 classes que funcionam como interface para que as classes dos módulos **BLL** (*Business Logic Layer*), **DALL** (*Data Access Layer*) e Facade possam se comunicar com as outras partes do módulo arquitetura.

A arquitetura.exception faz o tratamento das exceções do serviço. A arquitetura.framework possui a classe *ServiceAccess.java* que representa o acesso ao *WebService* passando os parâmetros necessários a comunicação. E a parte arquitetura.facade é representado pela classe *FacadeListener.java* que faz a comunicação com a o módulo Facade da aplicação.

- **com.example.app.bll**: O módulo BLL possui a classe de negócio *RotaBLL.java* que define as funções do objeto Rota: inserir, atualizar, selecionar e listar.
- **com.example.app.dal**: O módulo DAL é responsável pela comunicação com a base de dados local (SQLite) e com o acesso ao *WebService*. A classe *MapaDALL.java* faz a comunicação com o banco de dados local e a classe *RotaDAL.java* abre uma conexão com o serviço e grava as informações da Rota criada pelo usuário (Figura 11).

Figura 11: Parte do código da classe RotaDAL.java

```
public RotaDAL(Context contexto) {
    super(contexto);
}

public void inserir(Rota rota) throws ServiceException, NoConnectionException{
    ServiceAccess serviceAccess = new ServiceAccess(getContexto(), "AplicativoService");

    serviceAccess.Execute("GravarRota", serializeRota(rota));
}
```

- **com.example.app.dto**: O módulo **DTO** (*Data Transfer Object*) possui 2 classes *beans* que representam os objetos Ponto e Rota apenas com os métodos getters e setters, possibilitando o acesso às suas propriedades.
- **com.example.app.facade**: O Facade tem a função de fazer a comunicação entre o módulo .dal e o módulo .arquitetura.

Na figura 12, através do gráfico de dependência podemos visualizar as interações e dependências entre os módulos da aplicação. De acordo com o plugin Metrics do Eclipse, a cor da linha amarela indica o acoplamento entre os principais módulos a partir de um módulo central, neste caso o módulo .mapaexemplo. A cor vermelha sinaliza a dependência entre os módulos e aponta para as relações fortes. Os pacotes que não participam do ciclo estão na cor azul.

A camada *View* é formada pelas Activities MainActivity, RotaActivity, MapActivity e pelos arquivos de layout (.xml) activity\_main, activity\_map e list\_rotas. Para cada tela do aplicativo existe um arquivo .xml e um arquivo .java integrados. O arquivo java cria métodos que correspondem as funcionalidades oferecidas pelas telas. Para armazenar o estado de uma ação do usuário se cria uma *intent* que guarda o estado para ser utilizado de uma Activity para outra.

O aplicativo possui 3 telas, a tela inicial é formada pelos arquivos activity\_main.xml e MainActivity.java, a segunda tela é formada pelos arquivos activity\_map.xml e MapActivity.java e a última tela é formada pelos arquivos list\_rotas.xml e RotaActivity.java. As telas do aplicativo desenvolvido podem ser vistas na seção 4.6.

##### 4.4.1 Diagrama de Classes

O diagrama de classes do aplicativo é representado pela Figura 13. Considerando o diagrama, é possível visualizar os acoplamentos entre módulos e classes.



Figura 10: Organização dos módulos e principais classes do aplicativo

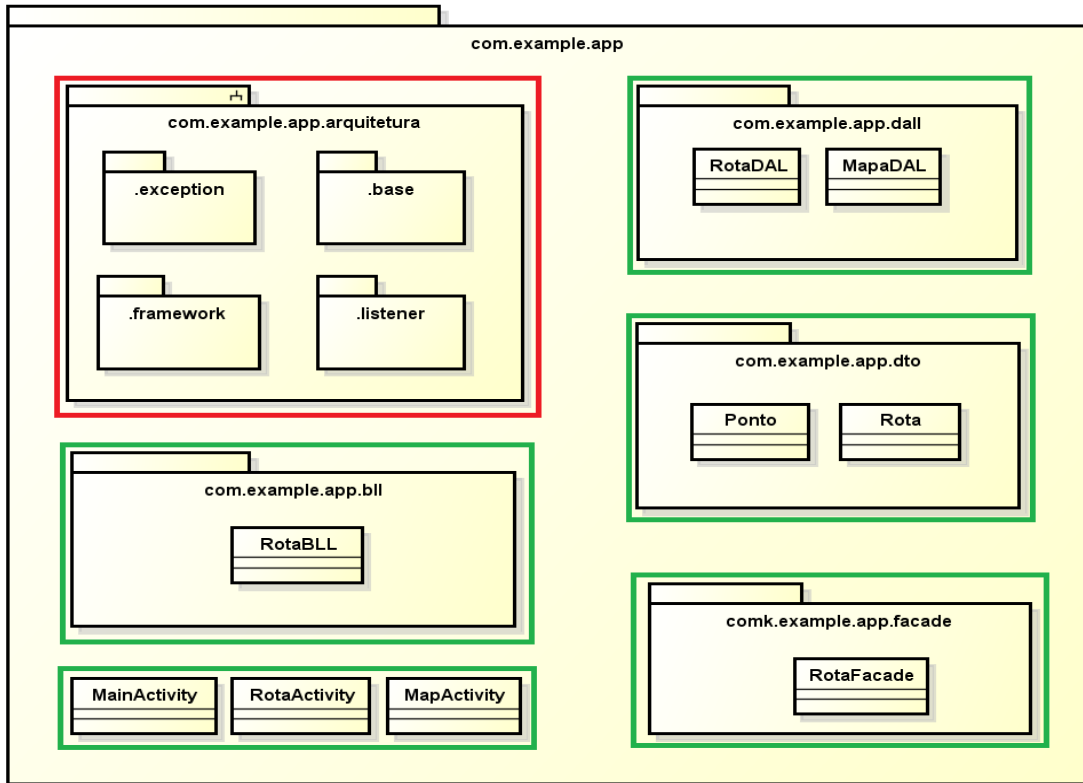
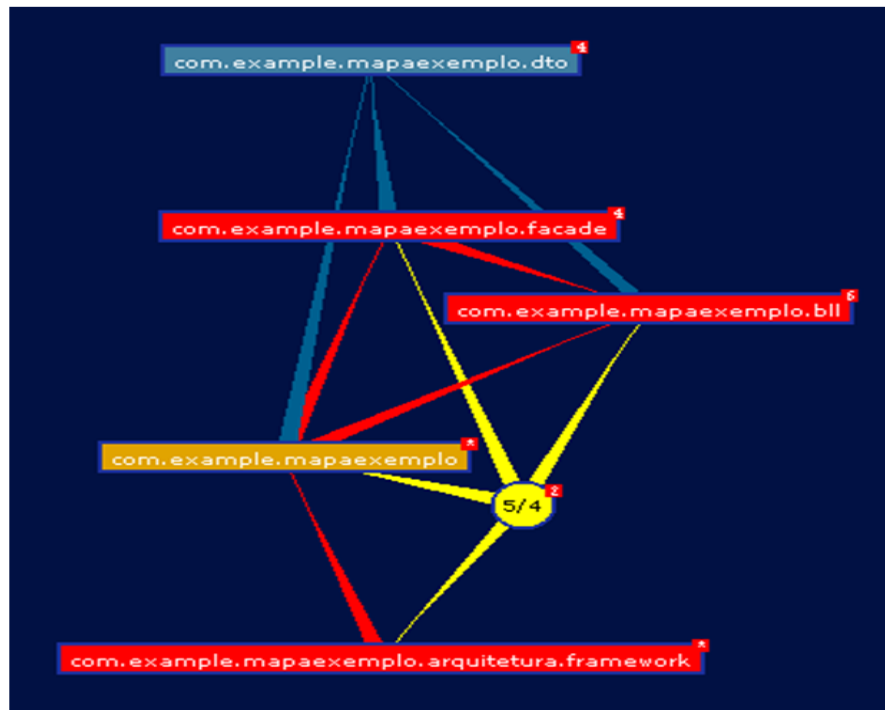


Figura 12: Dependência entre os Módulos a partir módulo .mapaexemplo





## 4.5 Serviços e Bibliotecas

O modelo de segurança do Android não permite que um aplicativo utilize os recursos dos dispositivos móveis sem a autorização do usuário. Para isso, é necessário adicionar as permissões no arquivo `AndroidManifest.xml` (Figura 14) e no momento da instalação, o aplicativo informa ao usuário quais permissões serão necessárias na execução das funcionalidades, estas que podem ter acesso às informações provenientes do GPS, das torres de celular, do WI-FI, da internet, gravar no cartão de memória externo, utilizar o microfone e a câmera. As permissões presentes no aplicativo desenvolvido podem ser visualizadas na Figura 14.

Figura 14: Permissões da Aplicação

```
<permission
    android:name="com.example.mapaexemplo.permission.MAPS_RECEIVE"
    android:protectionLevel="signature" />

<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />
<uses-permission android:name="com.example.mapaexemplo.permission.MAPS_RECEIVE" />
<uses-permission android:name="android.permission.INTERNET" />
<uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
<uses-permission android:name="com.google.android.providers.gsf.permission.READ_GSERVICES" />
<uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION" />
<uses-permission android:name="android.permission.ACCESS_FINE_LOCATION" />
```

O Google Maps é uma API (*Application Programming Interface*) de mapas presente na biblioteca Google Play Service SDK. Para associar o Google Maps ao projeto é necessário baixar esta biblioteca através do SDK Manager e habilitar o serviço no console de administração do Google APIs que disponibiliza uma chave de autenticação [13].

Com o Google Maps API v2 associado ao projeto é possível criar mapas com diferentes características e com locais definidos, controlar de zoom e gerar de rotas.

## 4.6 Funcionamento do Aplicativo

O Aplicativo possui três telas, nas quais apresentam todas as informações necessárias para que o usuário crie uma rota com um número reduzido de interações.

A tela inicial mostra um botão **Iniciar** que permite ao usuário marcar o início do percurso e dar continuidade ao sensoriamento da rota marcando os pontos geográficos por onde passar. Durante o sensoriamento, informações de velocidade e distância são apresentadas na tela, além das informações de latitudes e longitudes dos pontos armazenados. Quando o usuário desejar finalizar a rota, o mesmo deve selecionar o botão **Parar** e aguardar a visualização do percurso realizado no mapa que será apresentado em outra tela. A segunda tela além de apresentar graficamente o percurso, permite que o usuário nomeie e salve todas as informações que se referem a rota.

A terceira e última tela corresponde ao histórico de todas as rotas realizadas pelo usuário. Para acessar o histórico o usuário deve selecionar o botão **Rotas** posicionado ao lado do botão **Iniciar** na tela inicial. Nesta tela é possível que o usuário visualize novamente o seu percurso desenhado no mapa, bem como as informações de distância percorrida e

velocidade média de determinada rota. As telas e suas variações de estado estão representadas na Figura 15.

### 4.6.1 Distância entre Pontos

Para calcular a distância entre 2 pontos ao longo da superfície da Terra é considerada a fórmula abaixo [1], onde:

- $(\phi) = \textit{latitude}$ ;
- $(\lambda) = \textit{longitude}$ ;
- $R = \textit{raio médio da Terra (raio médio = 6371km)}$ .

Os ângulos precisam estar em radiano para validar a função.

$$a = \sin^2(\Delta\phi/2) + \cos(\phi_1) \cdot \cos(\phi_2) \cdot \sin^2(\Delta\lambda/2)$$

$$c = 2 \cdot \text{atan2}(\sqrt{a}, \sqrt{1-a})$$

$$d = R \cdot c$$

Aplicando esta fórmula no cálculo de distância entre os pontos capturados pelo GPS na aplicação, teremos a seguinte estrutura:

Figura 16: Cálculo da Distância entre pontos

```
private double calcularDistancia(Ponto p1, Ponto p2){

    double lat1 = p1.getLatitude();
    double lon1 = p1.getLongitude();
    double lat2 = p2.getLatitude();
    double lon2 = p2.getLongitude();

    double R = 6371; // km
    double dLat = toRad(lat2-lat1);
    double dLon = toRad(lon2-lon1);
    lat1 = toRad(lat1);
    lat2 = toRad(lat2);

    double a = Math.sin(dLat/2) * Math.sin(dLat/2) +
        Math.sin(dLon/2) * Math.sin(dLon/2) * Math.cos(lat1) * Math.cos(lat2);
    double c = 2 * Math.atan2(Math.sqrt(a), Math.sqrt(1-a));
    return R * c;
}

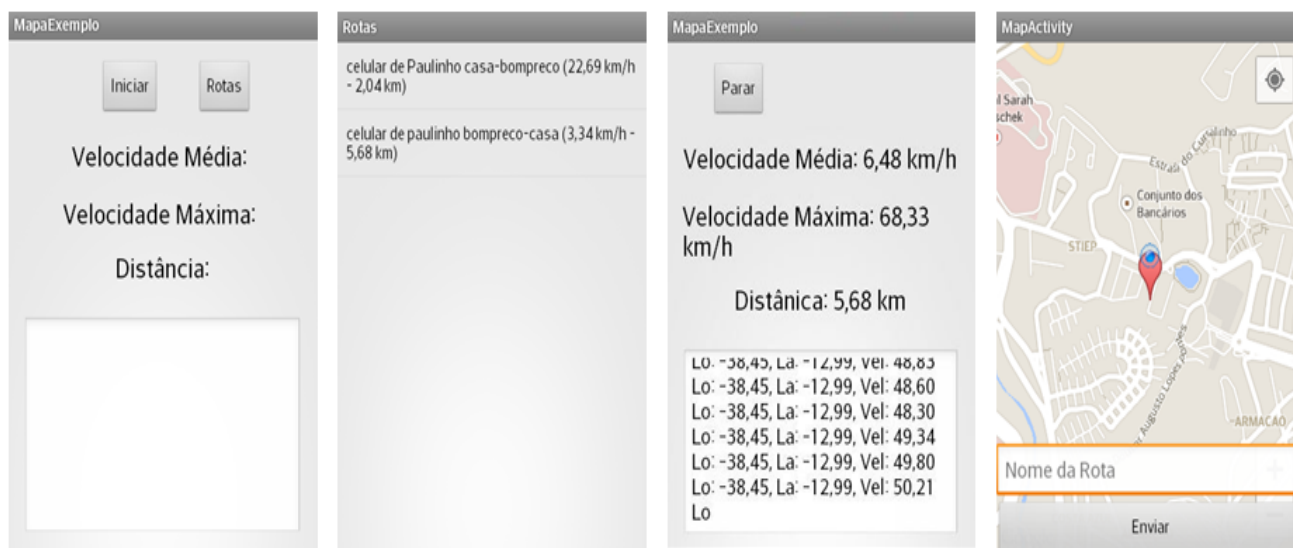
private double toRad(double x){
    return x * Math.PI / 180;
}
```

No código da aplicação, foi setado que o GPS captura um ponto a cada meio metro (0.0005 Km).

Figura 17: Intervalo entre os pontos

```
double dist = calcularDistancia(p2, p);
if(dist > 0.0005){ //meio metro
    rota.addPonto(p);
    rota.setDistancia(rota.getDistancia() + dist);
    logLocation(p);
}
}
```

Figura 15: Apresentação das Telas



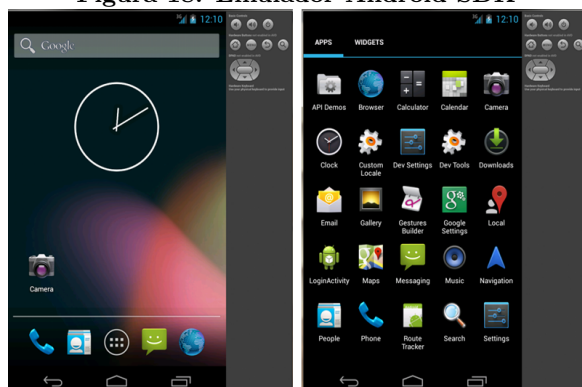
## 5. TESTES

A qualidade de um software é avaliada conforme os requisitos, os padrões e as características esperadas. Para garantir esta qualidade existe uma série de atividades, coletivamente chamadas de **VV&T** (Validação, Verificação e Teste), com a finalidade de garantir que a maneira que o software estar sendo desenvolvido, bem como o produto final estejam em conformidade com o especificado [10]. Nesta seção serão descritos e avaliados todos os testes realizados sobre o aplicativo e a apresentação dos ambientes utilizados.

### 5.1 Ambientes de teste

O Android SDK inclui em seu pacote um emulador de dispositivos móveis que permite criar protótipos e testar aplicativos sem o uso de um dispositivo móvel físico. Inicialmente os testes eram realizados em um Emulador (Figura 18), e através dele foi possível testar a organização das telas e suas interações, pois o emulador reproduz a tela inicial do sistema Android.

Figura 18: Emulador Android SDK



Nesta fase do projeto ainda não existia a estrutura necessária para carregar um mapa pelo Googlemap. No entanto,

a medida que a aplicação foi ganhando novas funcionalidades essa maneira de testar a aplicação se tornou inviável, pois era preciso esperar alguns minutos para que o emulador carregasse completamente as alterações, o que também era influenciado pelas configurações do computador que estava sendo utilizado no processo, assim o desempenho do emulador não reflete o desempenho do aplicativo em uma situação real.

Ao substituir o emulador por um dispositivo móvel foi possível testificar a usabilidade do aplicativo em um ambiente real e constatar que o desempenho é condizente com a realidade do projeto. Para que o aplicativo funcione adequadamente é fundamental que o GPS e a acesso a internet, através da conexão a dados, estejam habilitados.

A última fase de ambiente de teste foi verificar a portabilidade do aplicativo em outros dispositivos Android. A versão mínima do sistema Android exigida pelo aplicativo apresenta a API Level igual a 5. Na etapa anterior o teste foi realizado com dispositivo móvel Motorola RAZR D3 Android 4.1.2 (API Level 16), Nexus 4 Android 4.4.2 (API Level 16) e posteriormente testado com um dispositivo LG L4II Android 4.1.2. O aplicativo também foi testado em um Galaxy Note 10.1 Android versão 4.3 (API Level 18). Em todos esses casos, o aplicativo funcionou da maneira esperada.

Imagens dos dispositivos citados nesta seção estão apresentadas na seção 9.

## 6. ANÁLISE DOS DADOS

Esta seção tem por finalidade analisar as informações de todas as rotas e pontos armazenados no banco de dados deste sistema, aplicando conceitos de consultas por similaridade e dados espaciais, de modo a identificar encontros, fluxos e sobreposições de rotas.

Os dados espaciais representam informações sobre o local físico e a forma de objetos geométricos. Um tipo de dado

espacial é o *Geometry*, que armazena dados baseados em pontos nas dimensões (X,Y), as quais podem ser conhecidas através de um plano cartesiano. Considerando as coordenadas disponibilizadas pelo GPS, o encontro entre a latitude e a longitude consiste em um ponto geométrico, e consequentemente em um dado espacial simples.

Os mecanismos tradicionais de pesquisa em banco de dados utilizam operadores de igualdade relacionais para manipular os dados tradicionais, porém os resultados por meio de consultas exatas nem sempre atendem aos interesses quando se trata de dados espaciais [6]. Considerando latitudes e longitudes, informações imprecisas ou aproximadas refletem melhor o grau de aproximação entre os pontos, já que os valores dos dados armazenados são normalmente números quebrados e com grande variação de casas decimais. Essas consultas imprecisas ou aproximadas são consideradas consultas por similaridade.

## 6.1 Consultas por similaridade

As consultas por similaridade comparam todos os elementos do conjunto a um elemento de consulta e selecionam apenas aqueles elementos que atendem a um certo critério de similaridade [11]. A análise deste projeto consiste em identificar, dentre as rotas armazenadas, aquelas que possuem um determinado grau de similaridade, considerando a distância ou a aproximação entre as medidas de latitudes e longitudes dos pontos encontrados entre rotas distintas.

Para saber quais pontos passam próximos a uma distância Z de um determinado ponto (X,Y) utilizamos como base a consulta seguinte:

```
SELECT * FROM [Tabela_Pontos_Geograficos]
WHERE ABS(latitude - '[Valor_Latitude_Referência]') < '[Valor_Distancia_Base]'
AND ABS(longitude - '[Valor_Longitude_Referência]') < '[Valor_Distancia_Base]'
ORDER BY id_rota
```

Os dados de latitude e longitude armazenados no banco de dados são do tipo *NUMERIC(p, s)*, onde a precisão *p* é igual a 9 e a escala *s* igual a 4 (Exemplo: Latitude=-12.914700).

ABS é uma função matemática que retorna o valor absoluto (positivo) de expressão da categoria de tipo de dados numéricos exatos ou aproximados, por exemplo: para uma consulta **SELECT ABS(-1.0), ABS(0.0), ABS(1.0)**, o resultado obtido será **1.0, 0, 1.0**.

Heuristicamente, foi considerado o valor de 25m como raio médio de aproximação entre rotas (considera-se aqui a largura média de uma avenida com 3 pistas gira em torno de 9m no total). Considerando uma avenida com 3 pistas no sentido A e 3 pistas no sentido oposto B, temos 18m de largura total, que aliados a um canteiro central de 7 metros, resulta em 25m. Este valor corresponde ao grau de similaridade entre os pontos, um ponto é considerado similar a outro se estiver dentro da distância definida pelo raio, caso o ponto esteja situado fora dos limites do raio o grau de si-

milaridade desses pontos é considerado baixo. Assim, 25m corresponde a 0.025Km.

Como exemplo, para simular quais rotas possuem pontos que são similares a um ponto A(-12.978900, -38.458900), temos a seguinte situação:

- [Valor\_Latitude\_Referência] igual a -12.978900;
- [Valor\_Longitude\_Referência] igual a -38.458900;
- [Valor\_Distancia\_Base] igual a 0.025.

```
SELECT * FROM Ponto
WHERE ABS(latitude - '-12.978900') < '0.025'
AND ABS(longitude - '-38.458900') < '0.025'
ORDER BY id_rota
```

O resultado desta consulta são todos os pontos que possuem similaridade com o ponto A, isto é, todos os pontos que estão dentro do raio de 25m com relação ao ponto A. Com isso, uma rotina dentro do banco de dados foi desenvolvida para buscar por todos os pontos de diferentes rotas que se encontram dentro do raio de 25m com relação ao ponto A. Um corte do resultado da consulta é apresentado na Figura 19.

**Figura 19: Pontos similaridade a um determinado ponto**

id_ponto	id_rota	latitude	longitude	datahora
8098	84	-12.983300	-38.465600	3/11/2014 10:35:45 AM
8099	84	-12.981300	-38.441900	3/11/2014 10:35:58 AM
8100	84	-12.981400	-38.441700	3/11/2014 10:36:00 AM
8101	84	-12.981400	-38.441600	3/11/2014 10:36:01 AM
8102	84	-12.981500	-38.441600	3/11/2014 10:36:02 AM
8103	84	-12.981600	-38.441500	3/11/2014 10:36:03 AM
8104	84	-12.981700	-38.441400	3/11/2014 10:36:04 AM
8105	84	-12.981700	-38.441300	3/11/2014 10:36:05 AM
8106	84	-12.981800	-38.441200	3/11/2014 10:36:06 AM
8107	84	-12.981800	-38.441100	3/11/2014 10:36:07 AM
8108	84	-12.981900	-38.441100	3/11/2014 10:36:08 AM
8109	84	-12.982000	-38.441100	3/11/2014 10:36:09 AM
8110	84	-12.982000	-38.441100	3/11/2014 10:36:10 AM
8111	84	-12.982000	-38.441100	3/11/2014 10:36:11 AM
8112	84	-12.982100	-38.441100	3/11/2014 10:36:12 AM
8113	84	-12.982100	-38.441100	3/11/2014 10:36:13 AM
8114	84	-12.982200	-38.441200	3/11/2014 10:36:14 AM
8115	84	-12.982300	-38.441200	3/11/2014 10:36:15 AM
8116	84	-12.982300	-38.441200	3/11/2014 10:36:16 AM
8117	84	-12.982400	-38.441300	3/11/2014 10:36:17 AM

Diante desta informação podemos saber a quantidade de viagens de bicicleta realizadas por ciclistas em determinadas regiões, assim como identificar o horário de maior fluxo em determinado ponto.

Para saber quantos pontos, por rota, são similares ao ponto A, basta aplicar a próxima consulta por similaridade e o resultado desta consulta pode ser visto através na figura 20.

```
WITH similaridade AS (
SELECT * FROM Ponto
WHERE ABS(latitude - '-12.978900') < '0.025'
AND ABS(longitude - '-38.458900') < '0.025'
ORDER BY id_rota )
SELECT COUNT(id_rota) AS Qnt_Pontos, id_rota AS RotaID
FROM similaridade
GROUP BY id_rota
```

**Figura 20: Quantidade de pontos similaridade a determinado ponto agrupados por rota**

Qnt_Pontos	RotaID
155	84
163	85
558	87
259	88
200	90
93	91
298	92
297	93
603	96
520	99
334	102
95	104
469	105
1	108
200	110
373	113

## 6.2 Similaridade entre rotas

Compreender a similaridade entre pontos é a base para o entendimento da similaridade entre rotas. A similaridade entre rotas acontece quando existe uma grande quantidade de pontos similares entre os pontos de diversas rotas, dessa forma a rota que tiver mais pontos similares de outras rotas vizinhas é considerada uma rota forte, pois existe uma concentração de rotas próximas, com percursos similares.

A consulta para se chegar a este resultado deve aplicar a função de similaridade entre pontos para todos os pontos cadastrados no banco de dados e o modelo computacional se responsabilizará de fazer todas as combinações entre pontos para descobrir todas as relações de similaridade. A consulta a seguir mostra a quantidade de pontos similares que uma

rota possui (Figura 21). Esta consulta permite analisar qual é a rota mais forte, ou seja, qual é a rota que possui mais pontos similares.

```
WITH similaridade AS(
SELECT A.id_ponto AS PontoA, A.id_rota AS RotaA,
B.id_ponto AS PontoB, B.id_rota AS RotaB
FROM Ponto A, Ponto B
WHERE ABS(A.latitude - B.latitude) < '0.025'
AND
ABS(A.longitude - B.longitude) < '0.025'
AND A.id_rota <> B.id_rota)
SELECT COUNT(RotaB) AS Qnt_Pontos_Similares, RotaA
FROM similaridade
GROUP BY RotaA
```

**Figura 21: Quantidade de pontos similares por Rota**

Qnt_Pontos_RotaB	RotaA
667614	84
775447	85
1685390	87
1124919	88
993728	90
446954	91
1413188	92
1191487	93
1044	94
2276584	96
2275111	99
1548231	102
488282	104
1755618	105
384129	106
145476	108
920640	110
1269	112
1657204	113
268	114
726090	115
464999	116

Os dados gerados pela consulta acima permite que as análises sejam realizadas baseando-se nas rotas, a próxima consulta permite fazer a mesma análise só que baseada nos pontos. A consulta mostrará quantos pontos são similares a cada ponto registrado, além de apresentar a rota que cada

ponto pertence (Figura 22).

```
SELECT distinct A.id_ponto AS Ponto, A.id_rota AS Rota,  
COUNT(*) AS QTD_PONTOS_SIMILARES  
FROM Ponto A, Ponto B  
WHERE ABS(A.latitude - B.latitude) < '0.025'  
AND  
ABS(A.longitude - B.longitude) < '0.025'  
AND A.id_rota<>B.id_rota  
GROUP BY A.id_ponto, A.id_rota  
ORDER BY A.id_rota
```

**Figura 22: Quantidade de pontos similares por Ponto**

Ponto	Rota	QTD_PONTOS_SIMILARES
8240	84	4480
8241	84	4486
8242	84	4486
8243	84	4486
8244	84	4486
8245	84	4493
8246	84	4493
8247	84	4493
8248	84	4493
8249	84	4497
8250	84	4497
8251	84	4497
8252	84	4497
8253	85	4513
8254	85	4513
8255	85	4548
8256	85	4548
8257	85	4549
8258	85	4551
8259	85	4549
8260	85	4549
8261	85	4549
8262	85	4549

Para efeito de análise é importante saber em quais pontos da cidade ocorre um fluxo maior de ciclistas. Este resultado permite ao planejador urbano, por exemplo, a identificar pontos de maior fluxo e aplicar melhorias de infraestrutura nesses pontos, colocar mais placas de sinalização, ou até

mesmo implantar pontos de oficinas e aluguel para bicicletas que melhor atendam ao usuário. A consulta abaixo mostra o *ranking* de acordo com a quantidade de pontos similares que cada ponto possui (Figura 23).

```
WITH similaridade AS(  
SELECT DISTINCT A.id_ponto AS Ponto, A.id_rota AS Rota,  
COUNT(*) AS QTD_PONTOS_SIMILARES  
FROM Ponto A, Ponto B  
WHERE ABS(A.latitude - B.latitude) < '0.025'  
AND  
ABS(A.longitude - B.longitude) < '0.025'  
AND A.id_rota<>B.id_rota  
GROUP BY A.id_ponto, A.id_rota)  
SELECT DISTINCT Ponto, Rota, QTD_PONTOS_SIMILARES,  
ROW_NUMBER() OVER  
(ORDER BY QTD_PONTOS_SIMILARES DESC) AS Ranking  
FROM similaridade
```

A consulta seguinte permite selecionar os 10 primeiros pontos deste *ranking* (Figura 24). O resultado desta consulta seleciona os 10 primeiros pontos com maior que possui maior quantidade de pontos similares, com isso conclui-se que nestes pontos apresentados ocorre os pontos de fluxos mais críticos, que merecem uma maior atenção por parte do poder público de infraestrutura e transporte.

```
WITH similaridade AS(  
SELECT DISTINCT A.id_ponto AS Ponto, A.id_rota AS Rota,  
COUNT(*) AS QTD_PONTOS_SIMILARES  
FROM Ponto A, Ponto B  
WHERE ABS(A.latitude - B.latitude) < '0.025'  
AND  
ABS(A.longitude - B.longitude) < '0.025'  
AND A.id_rota<>B.id_rota  
GROUP BY A.id_ponto, A.id_rota)  
SELECT TOP 10 * FROM similaridade  
ORDER BY QTD_PONTOS_SIMILARES DESC
```

Figura 23: Ranking da quantidade de pontos similares entre pontos

Ponto	Rota	QTD_PONTOS_SIMILARES	Ranking
19727	118	5313	1
19728	118	5313	2
19729	118	5313	3
19730	118	5311	4
15289	104	5307	5
15290	104	5307	6
15291	104	5305	7
15292	104	5305	8
19731	118	5304	9
19732	118	5304	10
15293	104	5303	11
15294	104	5303	12
15295	104	5302	13
15296	104	5299	14
15297	104	5298	15
15299	104	5298	16
15298	104	5297	17
19733	118	5297	18
19734	118	5296	19
19723	118	5296	20
19724	118	5296	21
19725	118	5295	22
19726	118	5295	23
15300	104	5295	24
15301	104	5294	25

### 6.3 Limitação

A técnica de similaridade entre pontos proposta por este projeto permite que apenas as informações de latitude, longitude, velocidade, distância e tempo possam ser manipuladas pelas consultas. Identificando o fluxo de ciclistas em uma determinada avenida é possível identificar inclusive em qual horário este fluxo é maior. Porém, não é possível considerar o sentido utilizado pelos ciclistas ao passar por essa avenida, caracterizando-se uma limitação neste projeto em específico. Para exemplificar, considere a seguinte situação, de acordo com a Figura 25:

O ciclista **A** faz o seu percurso em um sentido da avenida, enquanto o ciclista **B** passa pela mesma avenida, só que no sentido oposto. O ciclista **B** está posicionado dentro do raio de 25m com relação ao ciclista **A**, isto é, o ponto de posicionamento do ciclista **B** é considerado um ponto similar ao ponto de posicionamento do ciclista **A**. Porém, se dezenas de ciclistas atravessarem esta mesma avenida em um determinado horário, não é possível identificar em qual sentido da avenida o fluxo de ciclistas é maior.

Figura 24: TOP 10 dos pontos com mais pontos similares

Ponto	Rota	QTD_PONTOS_SIMILARES
19727	118	5313
19728	118	5313
19729	118	5313
19730	118	5311
15289	104	5307
15290	104	5307
15291	104	5305
15292	104	5305
19731	118	5304
19732	118	5304

## 7. CONCLUSÃO

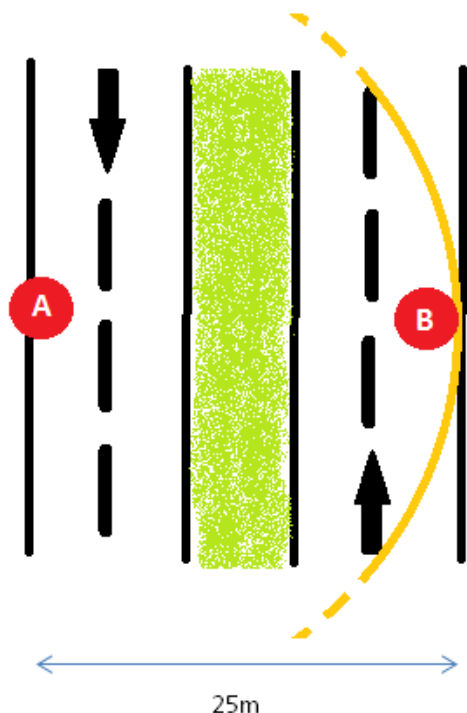
Diante dos meios de transportes privados e públicos utilizados no dia-a-dia das grandes cidades, a bicicleta cada vez mais ganha visibilidade no âmbito social, e muitos cidadãos preferem economizar o valor que se destinaria as passagens/bilhetes dos transportes coletivos, e ao consumo de gasolina e derivados, necessários na utilização e manutenção dos veículos motorizados individuais, e optam por um meio de transporte mais barato ou sem custo. Porém a ausência de infraestrutura para os usuários de bicicletas dentro da cidade impede que o cidadão usuário de bicicleta tenha o seu espaço definido e respeitado no meio público, gerando um desfavorecimento na hora de se locomover com segurança.

No entanto, é fundamental que as mudanças provenientes das ações de política de mobilidade urbana levem em consideração as necessidades de cada município. Este projeto teve como objetivo criar uma base de dados contendo informações das rotas utilizadas por ciclistas, através de um aplicativo móvel. As informações armazenadas podem ser analisadas e os resultados dessas análises podem contribuir com informações pertinentes às demandas da cidade de Salvador, tendo como alvo os usuários de bicicleta da região. Bem como, a aplicação permite entender a dinâmica dos ciclistas utilizando técnicas capazes de identificar a similaridade entre pontos de rotas distintas, considerando um raio médio de aproximação entre rotas como critério de similaridade, além do fluxo de rotas em uma determinada localidade.

A validação da ferramenta foi realizada através de testes com dispositivos móveis de diferentes marcas e em variações de velocidade e distância. Em todos esses casos, o aplicativo funcionou da maneira esperada. Foi observado que nas situações em que o usuário percorre uma grande distância, considerando uma rota média de 15Km, a aplicação demora alguns segundos a mais para carregar o percurso realizado graficamente sobre o mapa, além de salvar e enviar todas as informações da rota para o *WebService* criado. Os resultados obtidos permitem constatar que o aplicativo desenvolvido apresenta boa usabilidade e bom desempenho.



Figura 25: Rotas com sentidos oposto



Em virtude do que foi mencionado, conclui-se que a aplicação consegue organizar dados que podem subsidiar as tomadas de decisões futuras das autoridades públicas quanto as intervenções urbanas de transporte e mobilidade, com o apoio do aplicativo móvel Android desenvolvido. Aplicativo este que se destaca por disponibilizar informações corretas e seguras quanto as rotas criadas por ciclistas dentro da cidade de Salvador, pois os demais aplicativos móveis acessíveis e disponíveis encontrados na internet não permitem que os dados das rotas criadas por seus usuários sejam disponibilizados para fins de pesquisa, o que faz do aplicativo desenvolvido neste projeto um diferencial.

Apesar do principal objetivo deste projeto ter sido alcançado, os trabalhos correlatos juntamente com a análise de similaridade entre os pontos, demonstram que ainda existem projetos futuros a serem realizados para melhor atender as próximas demandas. As sugestões para trabalhos futuros estão sinalizados na seção seguinte.

### 7.1 Trabalhos Futuros

O aplicativo apresentado neste trabalho é restrito às informações pertinentes a rota, o perfil do usuário não é levado em consideração. Criar um cadastro de usuários permitirá que outras análises sejam aplicadas sobre o banco de dados. Para isso será necessário adicionar uma nova tela inicial, na qual o usuário fará um cadastro informando um *login* e uma senha.

Como trabalho futuro, pretende-se integrar o aplicativo as principais redes sociais da atualidade (Facebook e Twitter), ampliando a possibilidade de interações entre os usuários,

uma vez que permita ao usuário publicar as suas rotas e as suas metas alcançadas, além de compara-las com as rotas e metas de outros usuários cadastrados.

Além disso, pretende-se futuramente disponibilizar o aplicativo para um número maior de usuários ciclistas, a fim de aumentar o número de rotas inseridas na base de dados, de forma que se avalie o fluxo de ciclistas em diversas localidades da cidade de Salvador.

## 8. REFERÊNCIAS

- [1] Calculate distance, bearing and more between latitude/longitude points. Movable Type Ltd. <http://www.movable-type.co.uk/scripts/latlong.html> [Acesso em Janeiro de 2014].
- [2] População do Brasil. Censo IBGE, Novembro 2010. <http://censo2010.ibge.gov.br/noticias-censo?busca=1&id=3&idnoticia=1766&view=noticia> [Acesso em Janeiro de 2014].
- [3] Crescimento da frota de veículos. G1, Fevereiro 2011. <http://g1.globo.com/carros/noticia/2011/02/frota-de-veiculos-cresce-119-em-dez-anos-no-brasil-apos.html> [Acesso em Janeiro de 2014].
- [4] Política nacional de mobilidade urbana. Janeiro 2012. [http://www.planalto.gov.br/ccivil\\_03/\\_ato2011-2014/2012/lei/112587.htm](http://www.planalto.gov.br/ccivil_03/_ato2011-2014/2012/lei/112587.htm) [Acesso em Janeiro de 2014].
- [5] E. Almeida. *Towards a robust framework for software reuse*. IEEE, Las Vegas, USA.
- [6] A. Arantes, M. Vieira, C. Jr., and A. Traina. *Operadores de Seleção por Similaridade para Sistemas de Gerenciamento de Bases de Dados Relacionais*. Universidade de São Paulo, São Carlos, RS, Brasil.
- [7] A. Barbosa. *Abordagem híbrida para a avaliação de interfaces multimodais*. Outubro 2009.
- [8] E. Bezerra. *Princípios de Análise e Projeto de Sistemas com UML*, volume 2. Elsevier Editora Ltda.
- [9] CONDER. *Resultado da Pesquisa de Opinião com Ciclistas. Cidade Bicicleta - Mobilidade para todos*. 2009.
- [10] M. Delamaro, J. Maldonado, and M. Jino. *Introdução ao Teste de Software*. Elsevier, 2007.
- [11] M. Ferreira. *Suporte a consultas por similaridade unárias em SQL*. São Paulo, Janeiro 2008.
- [12] J. Krumm. *Ubiquitous Computing Fundamentals*. CRC Press, Washington, USA., 2010.
- [13] R. Lecheta. *GOOGLE ANDROID. Aprenda a criar aplicações para dispositivos móveis com Android SDK*, volume 3. Novatec.
- [14] W. Lee. *Beginning Android Application Development*. Wiley Publishing, 2011.
- [15] J. Nielsen. *Usability Engineering*. Elsevier, 1994.
- [16] A. Pereira, G. Pappa, M. Winckler, and R. Gomes. *Tópicos em Sistemas Colaborativos, Interativos, Multimídia, Web e Banco de Dados*. Sociedade Brasileira de Computação.
- [17] N. Reis. *Notas sobre urbanização dispersa e novas formas de tecido urbano*. Via das Artes, São Paulo, 2006.
- [18] M. Weiser. *The computer for the 21st century*. 1988.

## 9. ANEXO

### 9.1 Dispositivos de teste

As figuras 26, 27 e 28 exibidas nesta seção correspondem aos dispositivos móveis utilizados durante a validação dos testes do aplicativo desenvolvido.

Figura 26: LG L4II



Figura 27: Motorola Razr D3



Figura 28: Galaxy Note 10.1



### 9.2 Arquitetura Android

A Figura 29 mostra as 4 camadas que compõem o sistema operacional Android: Applications, Applications framework, Bibliotecas + Android Runtime e Linux Kernel.

### 9.3 Gráfico geral de dependência das classes

O gráfico representado pela Figura 30 foi gerado a partir de um plugin do Eclipse chamado Metrics e demonstra a dependência entre as principais partes da aplicação, dentre elas módulos e arquivos .xml.

Figura 29: Modelo Arquitetura Android [14]

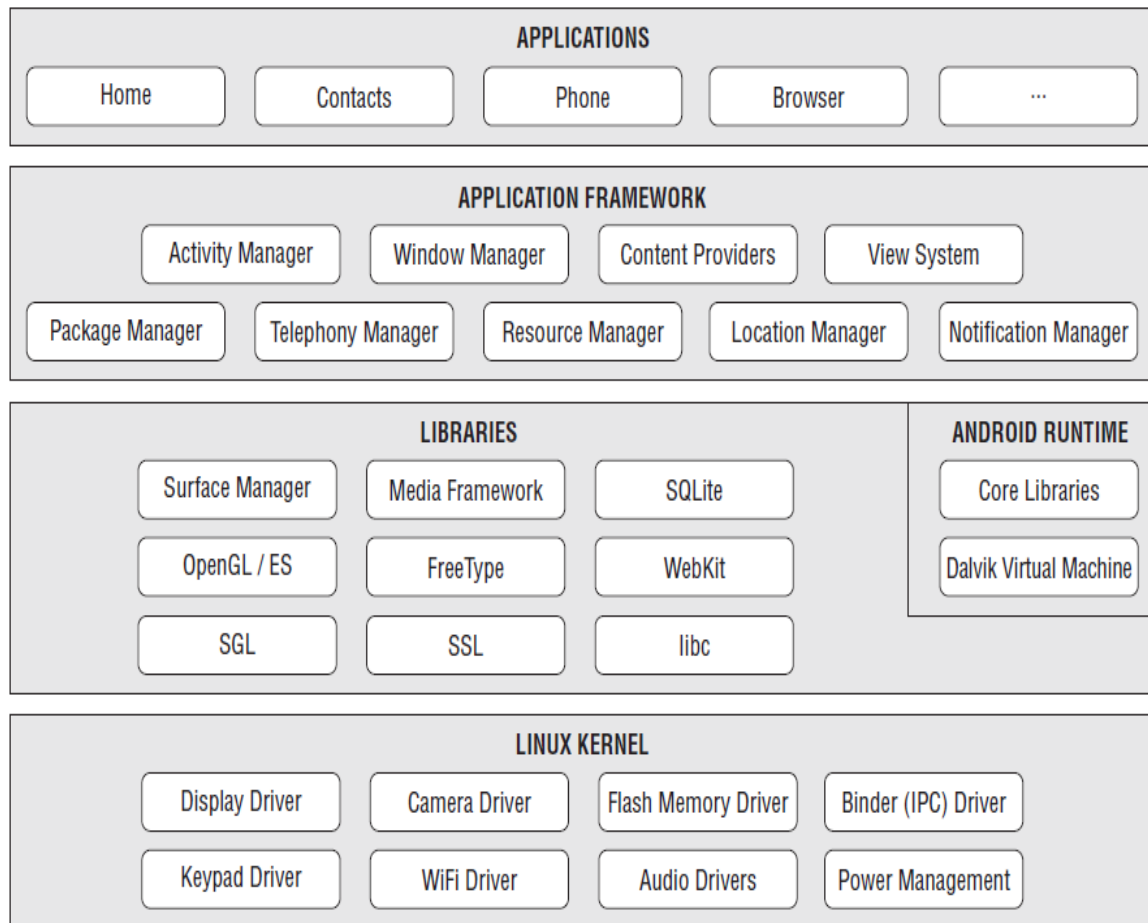


Figura 30: Gráfico de Dependência entre classes e módulos

