

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
INSTITUTO DE INFORMÁTICA
CURSO DE CIÊNCIA DA COMPUTAÇÃO

HUMBERTO BERNARDES FELIZZOLA

**Desenvolvimento de Aplicativo Mobile para
Pesquisa de Informações sobre Transportes
Públicos**

Trabalho de Graduação

Prof. Dr. Cláudio Fernando Resin Geyer
Orientador

Porto Alegre, Julho de 2012.

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL

Reitor: Prof. Carlos Alexandre Netto

Vice-Reitor: Prof. Rui Vicente Oppermann

Pró-Reitora de Graduação: Profa. Valquiria Link Bassani

Diretor do Instituto de Informática: Prof. Luís da Cunha Lamb

Coordenador do CIC: Prof. Raul Fernando Weber

Bibliotecária-Chefe do Instituto de Informática: Beatriz Regina Bastos Haro

AGRADECIMENTOS

Agradeço aos meus pais Roberto e Deborah pelo apoio, pelo carinho e pela educação. Ao meu irmão Henrique, por ser meu grande amigo e companheiro e por compreender minha ausência nos últimos tempos.

À minha namorada Anelise Frozza pelo incentivo, companheirismo, carinho, paciência e apoio incondicional durante toda a minha faculdade, e pela grande ajuda extra para revisar a monografia.

Aos meus amigos Marcelo Benites, Renata Nonnenmacher e Rodrigo Azevedo pela parceria tanto nos momentos de felicidade como nos difíceis.

Aos meus colegas de faculdade, que compartilharam das dificuldades e dos desafios para concluir essa trajetória, principalmente aos meus companheiros de FC COSMOS, que me ajudaram a perceber o quão boa é essa fase da vida.

A todos os professores que tive durante a faculdade, que fizeram jus ao ensino de excelência pelo qual o nosso curso é reconhecido.

Ao professor Geyer pela orientação no desenvolvimento deste trabalho.

Aos meus colegas da MECOMO do Brasil, onde descobri minha verdadeira paixão pelo mundo mobile.

E por fim, a todos aqueles que contribuíram de alguma forma para o desenvolvimento deste trabalho.

SUMÁRIO

LISTA DE ABREVIATURAS E SIGLAS	7
LISTA DE FIGURAS	9
LISTA DE TABELAS	11
RESUMO	12
ABSTRACT	13
1 INTRODUÇÃO	14
1.1 Motivação.....	14
1.2 Objetivo.....	14
1.3 Organização do Texto	15
2 CONCEITOS RELACIONADOS	17
2.1 LBS	17
2.1.1 Áreas de Aplicação	19
2.1.2 Exemplos Práticos.....	20
2.2 Mobilidade e Plataformas Móveis.....	24
2.2.1 Mobilidade	24
2.2.2 Plataformas	25
2.2.3 Comparação de Plataformas para Dispositivos Móveis	29
2.2.4 Plataforma Android.....	34

3	MODELO DA APLICAÇÃO	40
3.1	Identificação do Problema	40
3.2	Modelagem da Solução	41
3.2.1	Requisitos do sistema.....	41
3.2.2	Características do sistema.....	51
3.3	Modelagem da Arquitetura	54
4	IMPLEMENTAÇÃO DA APLICAÇÃO.....	58
4.1	Justificativas da Escolha de Plataforma para Implementação.....	58
4.2	Modelo de Desenvolvimento da Aplicação	59
4.2.1	Modelagem por Prototipação.....	59
4.2.2	Prototipação Evolucionária.....	60
4.2.3	Prototipação Descartável	60
4.2.4	Combinação e Utilização	61
4.3	Obtenção dos dados sobre a localização do aparelho.....	61
4.4	Obtenção dos dados para consulta de linhas e paradas (modo online).....	63
4.5	Obtenção dos dados para consulta de linhas e paradas (modo off-line).....	68
4.6	Visualização dos resultados das consultas no mapa	69
4.7	Funcionamento da aplicação	70
4.7.1	Principais Telas de Interação	70
4.7.2	Principais Funcionalidades	74
4.8	Validação da Implementação	80
4.8.1	Análise dos Requisitos.....	80
4.9	Comparação com Trabalhos Semelhantes	81
5	CONCLUSÃO	83
5.1	Análise Geral do Trabalho Desenvolvido.....	83
5.2	Possibilidades Futuras	84

6	REFERÊNCIAS.....	86
6.1	Livros.....	86
6.2	Artigos	87
6.3	Sites.....	87

LISTA DE ABREVIATURAS E SIGLAS

AAC	Advanced Audio Coding
AMR	Adaptive Multi-Rate
API	Application Programming Interface
EDGE	Enhanced Data rates for Global Evolution
EPTC	Empresa Pública de Transporte e Circulação
FIFA	Fédération Internationale de Football Association
GIF	Graphics Interchange Format
GPS	Global Positioning System
GSM	Global System for Mobile Communications
HTTP	Hypertext Transfer Protocol
IDE	Integrated Development Environment
JPG	Joint Photographic Experts Group
LBS	Location-based Service
MP3	MPEG-2 Audio Layer III
MVC	Model View Controller
MVP	Model View Presenter
PNG	Portable Network Graphics
PROCENPA	Companhia de Processamento de Dados de Porto Alegre
SDK	Software Development Kit
SGL	Scene Graph Library

SMS Short Message Service

SSL Secure Socket Layer

VM Virtual Machine

LISTA DE FIGURAS

<i>Figura 2-1: Painel de controle do sistema Prey.</i>	21
<i>Figura 2-2: Tela de sugestões do sistema Foursquare.</i>	22
<i>Figura 2-3: Tela de visualização de informações sobre o tráfego do sistema Waze.</i>	23
<i>Figura 2-4: Tela de monitoramento de frotas do sistema Fleet Web.</i>	23
<i>Figura 2-5: Gráfico construído a partir de dados de Fling (2009).</i>	24
<i>Figura 2-6: Os computadores de propósito geral Hashimi, Komatineni e MacLean (2010).</i>	25
<i>Figura 2-7: Comparação de Plataformas Mobile (Vision Mobile 2011).</i>	33
<i>Figura 2-8: T-Mobile G1, o primeiro aparelho comercializado rodando a plataforma Android (Scientific American 2008).</i>	35
<i>Figura 2-9: Camadas da Arquitetura da Plataforma Android (GOOGLE 2012).</i>	38
<i>Figura 3-1: Padrão de Arquitetura MVP (Pottel 1996).</i>	55
<i>Figura 3-2: Wireframe da Tela Principal.</i>	56
<i>Figura 4-1: Processo de Prototipação Evolucionária</i>	60
<i>Figura 4-2: Location Listeners.</i>	62
<i>Figura 4-3: Desligamento dos Serviços de Localização.</i>	63
<i>Figura 4-4: Área Utilizada na Pesquisa.</i>	64
<i>Figura 4-5: Retorno da Chamada por Linhas de Ônibus.</i>	65
<i>Figura 4-6: Retorno da Chamada para Pontos de Táxi.</i>	66
<i>Figura 4-7: Retorno de Chamada para Itinerário.</i>	66
<i>Figura 4-8: Trecho do código referente ao Parser JSON.</i>	67
<i>Figura 4-9: HTTP Requests for Data</i>	68
<i>Figura 4-10: Creating SQLite Database Structure.</i>	69
<i>Figura 4-11: Exemplo de Consulta no Banco de Dados Local.</i>	69
<i>Figura 4-12: Criando Overlays no Mapa.</i>	70
<i>Figura 4-13: Visualizando informações da parada de ônibus.</i>	71
<i>Figura 4-14: Tela de informações sobre a parada.</i>	72
<i>Figura 4-15: Tela de visualização de linha de ônibus.</i>	73
<i>Figura 4-16: Tela principal mostrando localização atual.</i>	75
<i>Figura 4-17: Busca com vários resultados.</i>	76
<i>Figura 4-18: Visualização do resultado.</i>	77

<i>Figura 4-19: Exemplo de localização imprecisa.....</i>	<i>78</i>
<i>Figura 4-20: Próximos a Mim (Ônibus e Táxi, respectivamente).....</i>	<i>78</i>
<i>Figura 4-21: Exemplo usando a função Trajeto.</i>	<i>80</i>

LISTA DE TABELAS

<i>Tabela 2-1: Comparação entre Aplicações Web e Aplicações Nativas.....</i>	<i>31</i>
<i>Tabela 2-2: Tabela de Versões da plataforma Android.</i>	<i>36</i>
<i>Tabela 3-1: Requisitos Funcionais.....</i>	<i>42</i>
<i>Tabela 3-2: Requisitos Não Funcionais</i>	<i>49</i>

RESUMO

O tema principal deste trabalho é o desenvolvimento de uma aplicação para dispositivos móveis que permita ao usuário realizar pesquisas sobre transportes públicos com base na localidade do usuário ou em um endereço fornecido. O trabalho foi desenvolvido em três partes. Na primeira, foi realizado um estudo das tecnologias envolvidas na criação de uma aplicação desse tipo. Isso permitiu a escolha da plataforma, da arquitetura da aplicação e dos módulos a serem utilizados na implementação. Na segunda parte, foi apresentada a modelagem proposta para um aplicativo mobile para pesquisa de transportes públicos. Na terceira e última, foram abordados os tópicos relativos à implementação da aplicação proposta. A aplicação, intitulada POAMóvel, foi desenvolvida com base nos dados referentes a transportes públicos da cidade de Porto Alegre, Rio Grande do Sul, Brasil – fornecidos pela Companhia de Processamento de Dados de Porto Alegre (Procempa).

Palavras-Chave: aplicativo mobile, LBS, transportes públicos, mobilidade, Porto Alegre, Android.

Development of a Mobile Application for Information Search about Public Transports

ABSTRACT

The main theme of this work is the development of a mobile application that allows users to search information about public transports based on the user location or in a location provided by text. The work was developed in three phases. On the first phase, it was made a research based on the technologies involved on the development of an application of this kind. This has allowed the choices about which mobile platform, which application architecture and which modules to choose for the application development. On the second phase, it was presented a model for the mobile application for information search about public transports. On the third and last phase, implementation related topics were presented. The final application, called POAMóvel, was developed with data related to the public transports available on the city of Porto Alegre, Rio Grande do Sul, Brasil – this data was provided by Companhia de Processamento de Dados de Porto Alegre (Procempa).

Keywords: mobile application, LBS, public transports, mobile, Porto Alegre, Android.

1 INTRODUÇÃO

1.1 Motivação

A importância das tecnologias móveis nos dias atuais é inquestionável. Através do constante aperfeiçoamento e de diversas novidades lançadas no mercado, pessoas usam e compram cada vez mais essas tecnologias.

Com a crescente popularização dos smartphones e dos planos de dados que possibilitam acesso dos aparelhos à internet, vemos aumentar as funções adicionadas aos aparelhos que antes carregávamos somente para efetuarmos ligações. Entre todas essas funções podemos destacar, como uma das mais importantes acrescentadas, o acesso a serviços baseados na localização (em inglês, LBS).

A partir destes serviços de localização é possível que os usuários de aparelhos celulares descubram, acessem e aproveitem de uma forma melhor o ambiente que os cerca. Um exemplo de melhor aproveitamento do meio em que vivemos é a utilização efetiva dos transportes públicos oferecidos por uma cidade, bairro ou localidade.

Com a utilização efetiva de aplicativos móveis que melhorem a experiência do uso de transportes públicos, é possível incentivar a comunidade a usar menos transportes privados, melhorando uma gama de aspectos relacionados ao trânsito e ao meio ambiente, como a poluição, a perda de tempo em engarrafamentos, o combustível consumido pelos veículos no trânsito e as nossas necessidades de locomoção em geral.

1.2 Objetivo

Este trabalho tem como objetivo o desenvolvimento de um aplicativo mobile para a pesquisa de informações sobre transportes públicos. O desenvolvimento do aplicativo

compreende o estudo de tecnologias envolvidas, a definição do problema a ser resolvido, a modelagem de uma solução para esse problema e a implementação de um aplicativo com base nessa modelagem.

A modelagem deve ser capaz de resolver o seguinte problema: disponibilizar para um usuário, portador de um dispositivo móvel, informações sobre diversos tipos de meios de transportes públicos baseados na sua localização ou em uma localização fornecida, de maneira prática, usual, personalizada e útil.

A implementação da aplicação se dará em uma das plataformas mais populares disponíveis para os aparelhos celulares da atualidade que é a plataforma do Google chamada Android.

Outro aspecto da implementação é que ela utilizará dados reais dos meios de transporte públicos da cidade de Porto Alegre, Rio Grande do Sul, Brasil, fato que caracterizará o seu nome: POAMóvel. Esses dados serão obtidos através do sistema para consulta de transportes públicos de Porto Alegre desenvolvido pela Procempa, chamado PoaTransporte.

O aplicativo POAMóvel possibilitará aos usuários da plataforma Android a consulta de dados sobre os mais comuns meios de transportes públicos coletivos e individuais disponibilizados na cidade de Porto Alegre a partir da sua localização atual ou de uma localização fornecida por meio de texto. Os usuários, através da utilização do aplicativo, poderão obter informações, como as paradas de uma linha de ônibus, os itinerários de uma linha de lotação ou o número de telefone para contato de um ponto de táxi.

1.3 Organização do Texto

Este trabalho está estruturado em cinco capítulos. No segundo capítulo serão apresentados os conceitos relacionados com o trabalho. Nele serão abordadas questões referentes ao embasamento teórico utilizado para a realização dos capítulos seguintes.

O terceiro capítulo apresentará o modelo da aplicação. Ele será constituído de seções com a identificação do problema e com a modelagem da solução da arquitetura.

A implementação da aplicação proposta por este trabalho será descrita no quarto capítulo, juntamente com maiores detalhamentos sobre o seu funcionamento, resultados de testes e maiores exemplificações da utilização do aplicativo.

Por fim, será apresentada a conclusão do trabalho, na qual constarão os objetivos alcançados e sugestões para possíveis melhoramentos e complementos da aplicação resultante.

2 CONCEITOS RELACIONADOS

Neste capítulo serão expostos os conceitos fundamentais relacionados com o trabalho: serviços baseados em localização, e mobilidade e plataformas móveis.

Na seção sobre LBS serão abordados aspectos gerais sobre serviços baseados na localização (LBS). Primeiro teremos uma breve descrição de LBS na qual constarão os conceitos básicos dessa tecnologia, logo após será apresentada uma relação de áreas onde essa tecnologia é aplicada e, por fim, serão mostrados alguns exemplos de utilização dessa tecnologia.

Na seção que trata de mobilidade e plataformas móveis, constarão informações sobre a evolução da mobilidade na computação, relações e comparações entre essas plataformas e, por fim, uma descrição da plataforma Android, escolhida para o desenvolvimento do sistema proposto por este trabalho.

2.1 LBS

Segundo Schiller e Voisard (2004), o termo serviços baseados em localização (LBS) é um conceito que visa representar as aplicações que integram localização geográfica (coordenadas espaciais) com uma noção geral de serviços. Como exemplos de aplicações desse tipo podem ser listados: serviços de emergência, sistemas de navegação para veículos móveis, ferramentas para planejamento de passeios turísticos ou combinações de lista de classificados com mapas.

Porém, ao se consultar outro autor, temos a informação de que o conceito de LBS ainda não se encontra definido de forma comum pela comunidade envolvida. De acordo com Küpper (2005), não são claras nem a nomenclatura nem a definição de LBS. Por ser um assunto abordado de forma multidisciplinar, principalmente pelas áreas de

computação ubíqua e telecomunicações, termos semelhantes acabam se referindo ao mesmo assunto, como serviços conscientes da localização, serviços relacionados à localização e serviços de localização.

Uma das definições mais aceitas e a que vai ser usada neste trabalho é a feita pela Associação GSM, que é um consórcio composto por centenas de operadoras de rede GSM. A associação define LBS como serviços que utilizam a localização do alvo para adicionar valor ao serviço, onde o alvo é a entidade a ser localizada (não necessariamente o usuário do dispositivo) (Küpper 2005).

Para se compreender melhor essa definição, é também explicado o conceito de adicionar valor ao serviço, que vai de encontro à primeira definição apresentada na seção.

Como exemplo de adicionar valor ao serviço, podemos citar uma ação comum na sociedade atual, na qual um usuário que possui um celular com acesso à internet e módulo GPS - capaz de lhe proporcionar informações sobre sua localização atual -, ao acessar um determinado programa, consegue visualizar que tipos de estabelecimentos existem a sua volta, como restaurantes, mercados, lojas e outros.

Outro exemplo cada vez mais presente na nossa sociedade é a utilização de aparelhos de localização em carros, permitindo que os usuários que possuam esses dispositivos consigam se valer da posição indicada pelo aparelho e das informações sobre as estradas e ruas para encontrar o destino ao qual desejam chegar, mesmo sem ter o conhecimento das ruas e estradas da localidade na qual estão trafegando.

Como último caso de exemplificação da adição de valor a um LBS, é possível citar a utilização da mudança da localização do aparelho a fim de disparar eventos em um dispositivo móvel. A aplicação de lembretes por localização lançada pela Apple, presente na última versão de sua plataforma móvel, utiliza essa tecnologia. Ela permite cadastrar lembretes para serem mostrados quando o usuário modificar a sua posição em relação a locais predeterminados.

É importante deixar claro que, diferentemente dos LBS, um serviço de localização não implica no processamento da informação, nem no tratamento da mesma de forma a ser utilizada pelo usuário. Sua responsabilidade é somente a de prover ao usuário

informações sobre sua localização. Os responsáveis por utilizar essas informações para executar ações de alto nível como as citadas anteriormente são os LBS (Küpper 2005).

É possível que o LBS funcione sem a utilização de um serviço de localização, como é o caso de algumas funções desenvolvidas na aplicação deste trabalho. Essa situação ocorre quando o usuário deseja obter informações sobre outras localizações que não são a que ele se encontra no momento. Para isso é necessário que o usuário informe total ou parcialmente as informações necessárias sobre a localidade a qual deseja, de forma que o LBS possa utilizar esse dado de entrada como se tivesse vindo de um serviço de localização.

2.1.1 Áreas de Aplicação

Nesta subseção serão apresentadas algumas áreas onde os LBS podem ser aplicados.

2.1.1.1 Serviços de Informação

Este tipo de serviço, apesar de simples, pode ser classificado como um dos mais populares dentre as áreas de aplicação de LBS. Seu funcionamento se dá com base na entrega de informações de pontos de interesse próximos ao usuário com base na sua localização.

O fornecimento de informações pode se dar de formas variadas, como listar contatos pessoais com base na proximidade permitindo a troca de mensagens instantâneas, procurar por um cinema próximo e descobrir a melhor maneira de chegar até ele, além de serviços de propaganda sensíveis à localização.

2.1.1.2 Gerenciamento de Tráfego

O gerenciamento de tráfego visa disponibilizar suporte aos utilizadores de carros, caminhões e afins. Inclui não somente o auxílio na navegação, como também configurações automáticas do veículo, diagnósticos de possíveis falhas e a disseminação de mensagens de aviso.

A aplicação mais comum nesses casos são os módulos de GPS integrados aos carros. Geralmente esses aparelhos são utilizados de tal forma que auxiliem o piloto do veículo

a chegar a um destino predeterminado por meio de instruções passadas por voz e visualização de mapas associados.

Sistemas desse tipo também podem apresentar, em tempo real, mensagens de aviso com base nas informações sobre o trajeto. Esse tipo de serviço permite o conhecimento, por parte do usuário, do tráfego atual, de acidentes e de bloqueios nas estradas, assim facilitando a execução do percurso.

2.1.1.3 Gerenciamento de Frotas e Logísticas

Enquanto a área de aplicação anterior tem por objetivo o gerenciamento de veículos isoladamente, o gerenciamento de frotas visa à aplicação de todos os serviços definidos no item anterior, porém a um conjunto de veículos e por um ponto central de controle de ações. Esse tipo de tecnologia torna possível o aprimoramento da velocidade dos transportes de carga, a utilização efetiva de diferentes veículos e o desenvolvimento de maneiras eficazes de comportamento em caso de falhas ou imprevistos.

2.1.2 Exemplos Práticos

2.1.2.1 Prey

Prey é uma ferramenta destinada à recuperação de aparelhos perdidos ou roubados que pode ser instalada tanto em smartphones quanto em laptops.

Esse sistema é um exemplo de LBS de informação, no qual uma vez informado pelo usuário a perda ou o roubo do seu dispositivo a partir de uma página na web ou de uma mensagem SMS enviada ao aparelho, é possível o monitoramento constante do deslocamento do dispositivo em questão e a realização de algumas operações de configuração do dispositivo, como travamento do aparelho, envio de mensagens e utilização da câmera do dispositivo.

O sistema utiliza tanto o posicionamento fornecido pelo GPS do dispositivo, quanto a triangulação de pontos de acesso de internet sem fio próximos para oferecer a localização da forma mais apurada possível com base nas limitações do dispositivo. É um sistema com licença de código aberto e bastante leve, sendo executado de fato nos aparelhos somente quando feita a notificação de perda.



Figura 2-1: Painel de controle do sistema Prey.

2.1.2.2 Foursquare

FourSquare é uma rede social voltada ao compartilhamento de informações sobre localidades.

Disponível tanto em plataformas móveis quanto para desktops e notebooks este sistema é um exemplo de LBS reativo de informação. A partir dele é possível compartilhar com uma rede de contatos experiências de visitaç o de variados lugares. Outra funcionalidade interessante desse sistema   a de descobrir que tipos de estabelecimentos existem por perto, obtendo mais informa oes sobre estes, como opini es, classifica o, endere o e telefone.

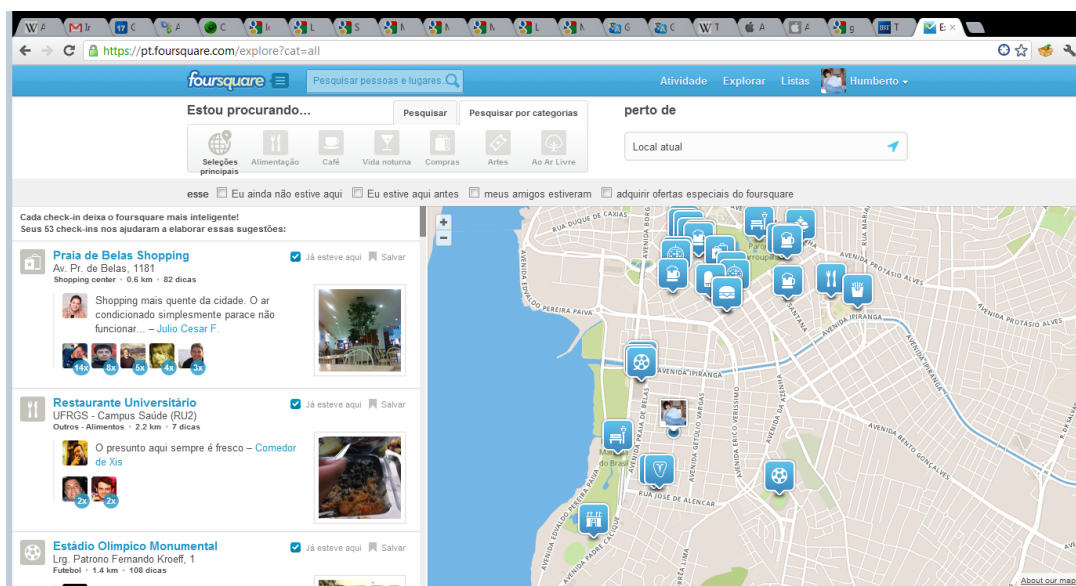


Figura 2-2: Tela de sugestões do sistema Foursquare.

2.1.2.3 Waze

Waze é um aplicativo móvel de gerenciamento de tráfego e navegação que funciona de forma colaborativa. Monitora constantemente a posição dos seus usuários, possibilitando a busca de informações sobre o tráfego dos seus veículos de maneira colaborativa.

O sistema funciona a partir da interação dos seus usuários, fornecendo a sua posição atual e algumas outras questões sobre o tráfego ou acidentes nas vias. Essas informações são compartilhadas com todos os usuários do aplicativo conectados e, somadas, permitem inferir melhores trajetos a se percorrer e rotas a se evitar por motivos de acidentes ou bloqueios, configurando o ambiente de colaboração.

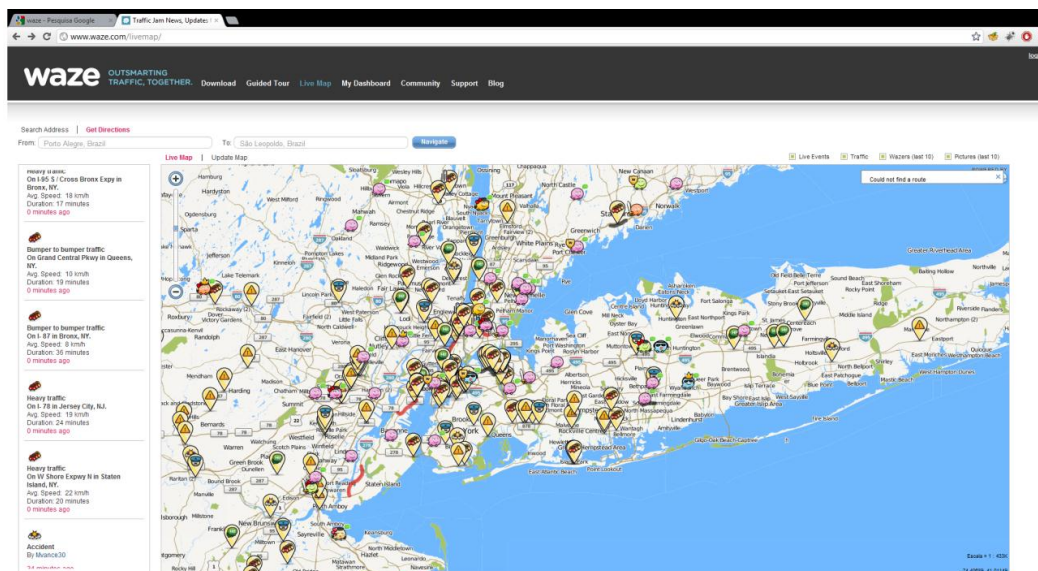


Figura 2-3: Tela de visualização de informações sobre o tráfego do sistema Waze.

2.1.2.4 Mecom Fleet Web

Fleet Web é um sistema de monitoramento de frotas de veículos desenvolvido pela empresa Mecom do Brasil. A partir dele é possível realizar uma vasta seleção de operações relacionadas tanto a veículos únicos quanto a frotas inteiras, bem como analisar as informações provenientes dos movimentos de veículos componentes dessas frotas.

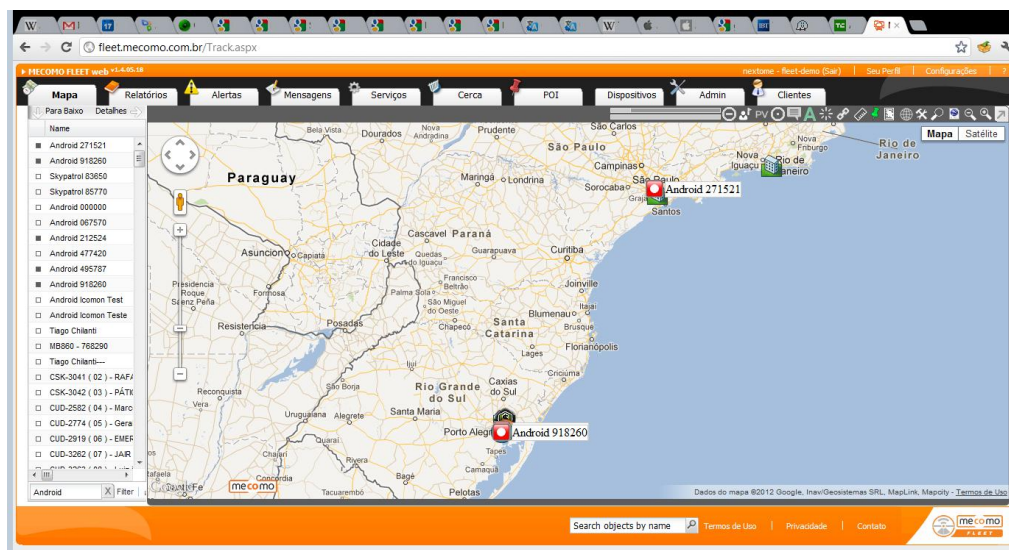


Figura 2-4: Tela de monitoramento de frotas do sistema Fleet Web.

2.2 Mobilidade e Plataformas Móveis

2.2.1 Mobilidade

Para termos uma ideia melhor do significado do crescimento e da popularização dos dispositivos móveis, é interessante observar o seguinte depoimento sobre o impacto dos aparelhos móveis na humanidade.

O alcance dos telefones móveis na humanidade já é mais do que o dos carros (800 milhões), da TV (1,5 bilhão), ou da internet (1,1 bilhão). Os dispositivos móveis irão trazer maiores mudanças na próxima década do que qualquer dos aparelhos citados anteriormente já trouxeram. Os celulares acrescentam a combinação do telefone fixo, da internet, do computador, do cartão de crédito e da TV. Os celulares vão impactar nossas vidas de mais maneiras do que podemos imaginar, por causa dos seus aspectos de multifuncionalidade e de alcance. (Ahonen apud Fling 2009, tradução nossa)

Hoje, mais pessoas acessam a internet por meio de um dispositivo móvel do que por meio de um computador, e a disparidade entre esses dois números crescerá mais acentuadamente nos próximos anos, como mostra o gráfico a seguir com base nos dados de Fling (2009).

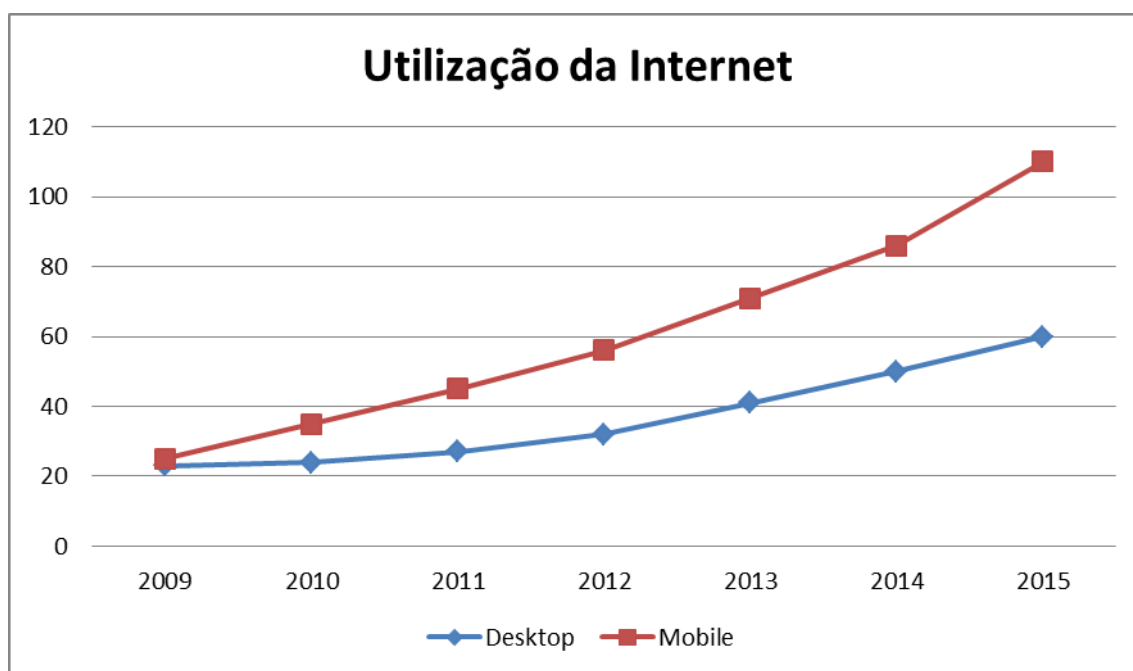


Figura 2-5: Gráfico construído a partir de dados de Fling (2009).

Como resultado do acréscimo dessa série de novas funcionalidades aos aparelhos e da crescente popularização destes dispositivos é possível imaginar que eles se tornem computadores tão capazes e de uso tão generalizado, que em um futuro próximo terão o papel de ocupar o espaço onde hoje se encontram nossos computadores pessoais (PCs). Essa visão é compartilhada por Hashimi, Komatineni e MacLean (2010) e ilustrada pela figura exposta a seguir.

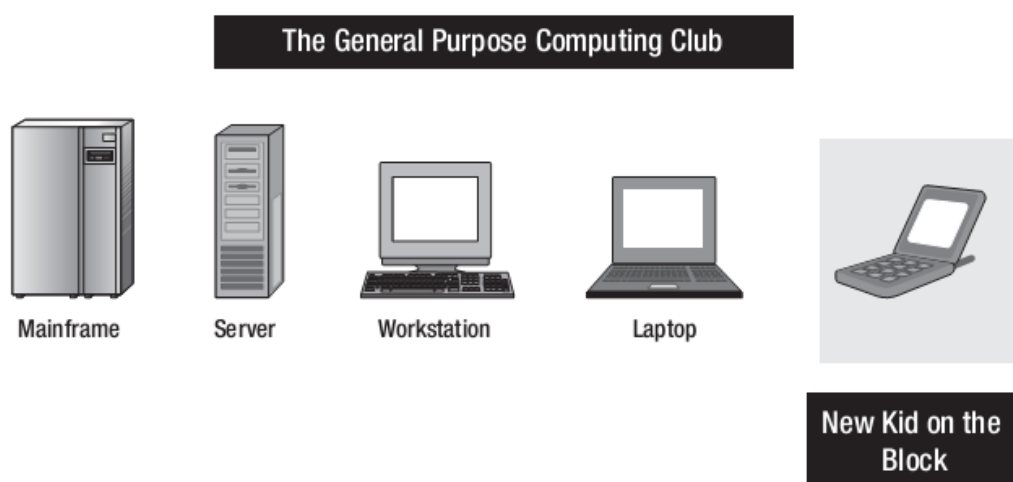


Figura 2-6: Os computadores de propósito geral Hashimi, Komatineni e MacLean (2010).

Tendo em vista os fatos citados anteriormente, é possível afirmar que empresas tradicionais do mercado de computadores pessoais vão entrar na competição junto com outras que já atuam nesse mercado das tecnologias móveis, constituindo assim, segundo Hashimi, Komatineni e MacLean (2010), uma nova batalha entre sistemas operacionais, plataformas, linguagens de programação e ambientes de desenvolvimento.

2.2.2 Plataformas

Uma plataforma mobile tem como principal dever proporcionar ao usuário o acesso aos recursos de seu dispositivo móvel. Todos os aplicativos e serviços que estarão rodando nesse aparelho necessitam de uma plataforma Fling (2009).

A partir da evolução dos dispositivos móveis, várias empresas começaram a investir no desenvolvimento de soluções para ingressar no mercado de plataformas mobile, como veremos a seguir através de exemplos de produtos desse tipo, acompanhados de suas características.

2.2.2.1 *Android*

Android é uma plataforma de código aberto desenvolvida para dispositivos móveis, suportada pela Google e de propriedade da Open Handset Alliance, um grupo sem fins lucrativos comprometido com a abertura e a inovação da experiência para o usuário móvel (Gargenta 2011). Este grupo é formado por empresas de vários segmentos relacionados às tecnologias móveis, como por exemplo, as grandes operadoras de telefonia celular e as fabricantes de aparelhos móveis.

A plataforma Android surgiu do trabalho deste grupo - com grande destaque para o Google que é o membro principal - em torno do objetivo de acelerar a inovação das tecnologias móveis e oferecer aos consumidores uma experiência móvel melhorada, mais valiosa e com menos custo.

Segundo a opinião de Gargenta (2011), o Android está revolucionando o espaço das tecnologias móveis. Pela primeira vez, temos uma plataforma verdadeiramente aberta que separa o hardware do software que está rodando no aparelho. Isso permite que uma quantidade muito maior de aparelhos executem as mesmas aplicações, criando um ecossistema muito mais rico para desenvolvedores e consumidores.

Como a implementação do modelo proposto por esse trabalho foi desenvolvida para a plataforma Android, posteriormente teremos uma seção descrevendo mais detalhadamente essa plataforma.

2.2.2.2 *BlackBerry*

BlackBerry OS é um sistema operacional dedicado a plataformas móveis desenvolvido pela empresa RIM (Research In Motion) para uso na sua linha homônima de smartphones. Como destaque dessa plataforma, e grande motivo de sua popularidade, está o suporte nativo para a troca de e-mails corporativos com segurança avançada. Por esse motivo, os smartphones dessa companhia têm uma larga presença no mercado corporativo.

2.2.2.3 *iOS*

iOS é a plataforma desenvolvida e distribuída pela Apple para seus dispositivos móveis, dentre os mais famosos, iPhone, iPad e iPod. Originalmente foi lançada em

2007 com o nome de iPhone OS para os produtos iPhone e iPod Touch. Posteriormente, foi estendida para suportar outros aparelhos da companhia, como o iPad e a Apple TV, trocando o seu nome para iOS. Diferentemente de algumas outras companhias anteriormente citadas, a Apple não licencia o iOS para utilização em aparelhos móveis que não sejam produzidos por ela.

Baseada na plataforma da empresa para desktops e laptops OS X, esta plataforma teve como pontos principais no seu lançamento a inovação no quesito de interação - utilizando-se o conceito de manipulação direta com gestos de múltiplos toques na tela -, e a utilização de acelerômetros internos - que capacitam o dispositivo a se adaptar à maneira como o seu conteúdo é mostrado ao usuário a partir da orientação atual em três dimensões.

Com atualizações anuais liberadas para quase todos os dispositivos suportados, uma quantidade crescente de aplicativos e uma base de usuários muito satisfeita, a plataforma para dispositivos móveis da Apple é uma das mais presentes no mercado atual, como poderá ser verificado, posteriormente, na seção de comparação das plataformas.

2.2.2.4 Java Micro Edition (Java ME)

Anteriormente conhecida como J2ME, Java ME é constituída por um subconjunto da plataforma Java e fornece uma coleção de APIs do Java para o desenvolvimento de software destinados a aparelhos móveis (Fling 2009). Dentro de suas principais características podemos destacar a interface com o usuário flexível, a segurança robusta, os protocolos de redes incorporados e o suporte a aplicações online e off-line que podem ser baixadas dinamicamente.

Desenvolvida especificamente para aparelhos móveis, a plataforma atende desde celulares de baixo custo até os mais modernos smartphones. Atualmente está sendo usada em cerca de três bilhões de telefones pelo mundo inteiro (Oracle 2012).

2.2.2.5 Mobile Web

Aplicações Mobile Web são sites na internet confeccionados com tecnologias Web para uso específico a partir de smartphones. Esse tipo de sistema se assemelha muito às

aplicações nativas, porém, permite a visualização e a interação, na maioria das vezes, de maneira semelhante entre plataformas diferentes.

As aplicações Mobile Web não precisam ser instaladas nem compiladas no aparelho em que irão rodar. Usando XHTML, CSS e Java Script, elas são capazes de prover uma experiência semelhante a das aplicações nativas. Porém, na realidade, estão rodando a partir da utilização de um navegador da web (Fling 2009).

Esse tipo de aplicação será descrito com maiores detalhes posteriormente na seção de comparação entre modelos nativos ou web.

2.2.2.6 Windows Phone

Windows Phone é o sistema operacional destinado a aparelhos móveis desenvolvido e utilizado pela Microsoft. Embora seja o sucessor da antiga plataforma Windows Mobile, não apresenta retrocompatibilidade com a mesma. Diferentemente da sua antecessora, não compartilha o foco no mercado corporativo. Lançado em 2010, busca o mercado de consumidores. “O lançamento dessa plataforma simboliza a tentativa da Microsoft de recomeçar no sentido de desenvolvimento de software para o mercado móvel” (Petzold 2010).

Apresentando inovações nos paradigmas organizacionais e suportando as plataformas de programação Silverlight e XNA, o sistema é bastante distinto dos outros sistemas disponíveis no mercado quanto ao quesito interação, com destaque para seu design chamado Metro, que pode ser definido como “... uma linguagem de design única definida pelo reducionismo, tipografia e formatos e cores sem adornos” (Ulanoff 2012).

Atualmente está sendo adotado por diversas fabricantes, com destaque para Nokia e para HTC.

2.2.2.7 Symbian OS

Symbian OS é um sistema operacional para dispositivos móveis utilizado nos smartphones da Nokia. Ele roda exclusivamente em processadores ARM e foi criado a partir de um sistema operacional rudimentar desenvolvido nos anos 90 destinado a agendas eletrônicas (Conjecture Corporation 2012).

2.2.3 Comparação de Plataformas para Dispositivos Móveis

Nesta subsecção constarão duas importantes comparações a serem observadas no âmbito do desenvolvimento para dispositivos móveis.

A primeira é uma comparação entre os dois caminhos disponíveis quando se vai começar o desenvolvimento de um sistema móvel, o caminho de desenvolver uma aplicação nativa e o de desenvolver uma aplicação web.

Na primeira delas são utilizadas ferramentas para a implementação de um aplicativo voltado ao uso específico em uma plataforma, e na segunda, utilizando-se de um conjunto de ferramentas e linguagens web, é possível disponibilizar uma aplicação acessada pela internet em várias plataformas diferentes.

2.2.3.1 *Aplicação Nativa ou Aplicação Web*

2.2.3.1.1. Aplicação Nativa

Segundo trecho do livro Fling (2009), para expressar o verdadeiro significado da maneira como é desenvolvida a aplicação nativa e das diferenças desse tipo de aplicação, deveríamos chamar de aplicações de plataforma, remetendo ao real processo de desenvolvimento que foi utilizado para cada plataforma.

Criar uma aplicação do tipo plataforma ou nativa envolve uma gama de questões a serem observadas, como:

- Quais os dispositivos que atendem a essa plataforma e poderão rodar o aplicativo.
- Quais as ferramentas de desenvolvimento disponíveis para essa plataforma e qual a complexidade de se desenvolver um programa a partir dessas ferramentas.
- Qual a maneira de distribuição de aplicativos que essa plataforma suporta e que poderemos usar tanto para disponibilizar os aplicativos aos seus usuários finais quanto para recolher o retorno financeiro pelo trabalho desenvolvido.

Por serem aplicações criadas utilizando-se de recursos de hardware disponibilizado pela plataforma, esse tipo de aplicação pode aproveitar de forma completa os diversos recursos que um dispositivo móvel pode oferecer, dentre eles a câmera fotográfica, o

sistema de arquivos, os serviços de localização, os sensores de orientação, entre outros. Deve-se destacar, também, que esse tipo de aplicativo permite que a utilização por parte do usuário ocorra mesmo com a ausência de uma conexão com a internet e, geralmente, não dependa da existência de outros aplicativos relacionados para seu funcionamento.

Por terem acesso aos recursos do aparelho, normalmente é requisitado que esse tipo de aplicativo passe por uma aprovação por parte da empresa desenvolvedora da plataforma. Isso garante que, salvo algum possível erro por parte da equipe de verificação, nenhum aplicativo malicioso seja aprovado e liberado para instalação nos aparelhos.

2.2.3.1.2. Aplicação Web

A grande motivação para o uso de uma aplicação móvel web é a fragmentação de dispositivos.

Desde a popularização da web 2.0, a partir da utilização de aplicativos como Facebook e Flickr, surgiu a discussão sobre como realizar a portabilidade dessas experiências para o mundo dos dispositivos móveis. Porém, os navegadores desses aparelhos estavam relativamente atrasados em quesitos de tecnologia se comparados com os disponíveis para desktops, tornando necessária uma adaptação do conteúdo para que este pudesse apresentar funções e comportamento semelhantes.

Com a introdução do iPhone, em 2007, esse panorama mudou bastante. A utilização do WebKit como navegador e a forma de interação inovadora permitiram ao aparelho renderizar o conteúdo não otimizado para dispositivos móveis de forma utilizável. A partir daí, mais dispositivos seguiram o caminho de prover uma melhor experiência de navegação na web portátil, aumentando, assim, o conteúdo na rede voltado a esse mercado em ascensão.

Com destaque para a ausência da necessidade de grandes modificações - para se atender um variado espectro de plataformas móveis - e para a baixa complexidade do sistema final - na maioria dos casos -, esse modelo de aplicação é utilizado quando se busca rapidez e baixo custo no desenvolvimento e alto alcance de plataformas.

É importante destacar que, embora não seja possível garantir a consistência na visualização das aplicações web mobile por todos os dispositivos disponíveis, é muito provável que a maioria dos dispositivos atuais seja capaz de executar corretamente as aplicações desenvolvidas dessa forma.

2.2.3.1.3. Comparação entre aplicação Web e aplicação Nativa

Se formos analisar a complexidade dos aplicativos atuais, veremos que grande parte deles poderia ser implementado utilizando-se de um misto de tecnologias web como XHTML, CSS, JavaScript e Ajax, sem que isso comprometesse a experiência do usuário. As aplicações desenvolvidas dessa maneira têm algumas vantagens: funcionarão em mais tipos de aparelho sem necessitar de uma grande adaptação, necessitarão menos testes e poderão ser atualizadas sem a burocracia de aprovação pelo desenvolvedor presente nas aplicações nativas. Porém, as aplicações web terão mais dificuldade, e em alguns casos a impossibilidade, de acessar recursos do aparelho e necessitarão de acesso constante à internet.

No caso das aplicações nativas, teremos muito mais facilidade e possibilidades quanto à utilização de recursos do aparelho, uma vez que aprovada a aplicação pelo desenvolvedor da plataforma teremos a garantia de funcionamento do aplicativo. Por fim, é muito provável que tenhamos um modo de distribuição e de venda dos aplicativos com o acesso bastante facilitado no que diz respeito ao usuário final.

Na tabela a seguir podemos encontrar um resumo das diferenças entre os tipos de aplicativos móveis web e nativos que foram tratados nesta subseção.

Tabela 2-1: Comparação entre Aplicações Web e Aplicações Nativas

Aspectos / Tipo	Web	Nativo
Desenvolvimento	Menor complexidade ao criar aplicações que não utilizem recursos do aparelho, utilizando-se conceitos de HTML básico, CSS e JavaScript.	Utilização completa dos recursos do dispositivo (câmera, localização,...) e possibilidade de um design diferenciado.

Disponibilidade	Dependem da presença de uma conexão com a internet e de um navegador móvel que atenda seus requisitos mínimos.	Dependem da plataforma e da versão da plataforma instalada no aparelho.
Multi-plataforma	Podem ser utilizados em qualquer plataforma que possua um navegador de internet que atenda seus requisitos.	Necessita do desenvolvimento específico de uma versão para cada plataforma em que será utilizada.
Testes	Pode ser facilmente testada a partir dos navegadores dos dispositivos.	Dependendo do caso pode ser utilizado um emulador, mas geralmente é necessário possuir dispositivos para realizar testes mais apurados.
Experiência de Usuário	Varia significativamente com relação ao aparelho utilizado. A experiência pretendida pode acabar sendo atingida em apenas alguns aparelhos.	Experiência de usuário similar é praticamente garantida nas plataformas que a aplicação atender.
Distribuição	Acesso pelos navegadores não necessita instalação, pode ser feita utilizando-se um aplicativo nativo simples que só direciona para o navegador do aparelho com o endereço carregado.	Feita geralmente a partir de lojas de aplicativos, fornecem um ponto de distribuição centralizado e de segurança garantida, permitindo variados tipos de arrecadação por aplicativo.
Custo	Relativamente baixo para a abrangência de dispositivos que podem atingir e pela dificuldade no	Pode tornar-se alto quando se deseja atender várias plataformas, pois será

	desenvolvimento.	necessário desenvolvimento específico para cada uma.
--	------------------	------------------------------------------------------------

2.2.3.2 Comparação das Plataformas

Nesta subseção será mostrada uma figura com gráficos comparativos entre as plataformas mobile citadas anteriormente, as plataformas serão comparadas de acordo com critérios relevantes aos desenvolvedores.

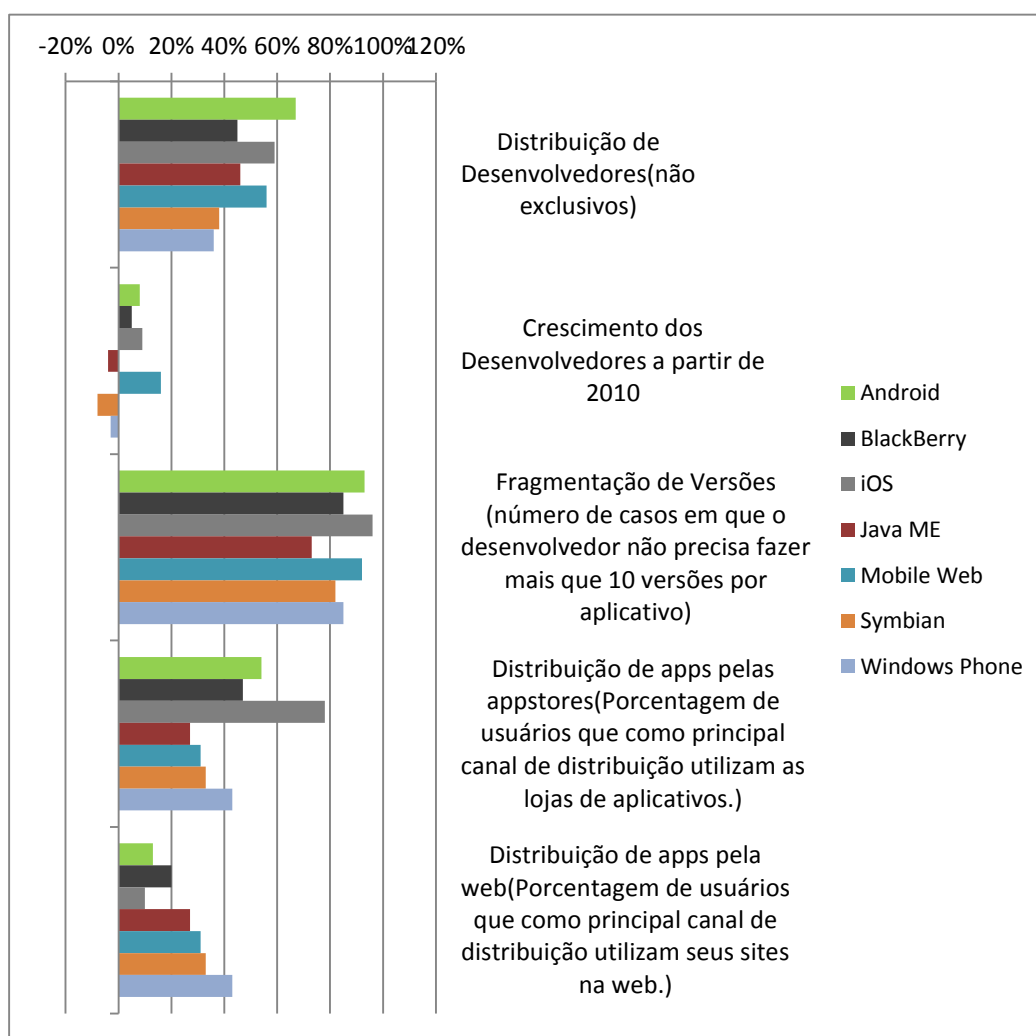


Figura 2-7: Comparação de Plataformas Mobile (Vision Mobile 2011).

2.2.4 Plataforma Android

Esta seção acrescentará ao trabalho uma série de dados sobre a plataforma Android, escolhida para a implementação do trabalho.

Primeiro veremos alguns dados relativos ao surgimento, crescimento e atual estado dessa plataforma no mundo. Em seguida teremos uma descrição resumida da arquitetura da plataforma, como as principais características, as questões envolvendo desempenho, a integração com diferentes bibliotecas e o modo de funcionamento da arquitetura. Por fim, teremos uma descrição mais detalhada dos módulos que o sistema desenvolvido nesse trabalho utiliza.

2.2.4.1 Histórico e Atualidade¹

A história do Android começa com a aquisição, pelo Google, de uma empresa startup homônima no ano de 2005. Como justificativa dessa aquisição estava o intuito de desenvolver uma nova plataforma para dispositivos móveis.

Mais tarde, em 2007, um grupo de empresas líderes na indústria de dispositivos móveis se reuniu com os objetivos de inovar rapidamente e responder melhor as necessidades dos consumidores e desenvolvedores na área das plataformas mobile. Esse grupo existe até hoje e é chamado de Open Handset Alliance. Dentre as empresas envolvidas inicialmente nesse grupo, podemos citar Sprint Nextel, T-Mobile, Motorola, Samsung, Sony Ericsson, Toshiba, Vodafone, Google, Intel e Texas Instruments.

Um fato interessante é que todos os membros do grupo se comprometeram a produzir essa solução de plataforma móvel a partir da licença de código aberto Apache License versão 2.0, o que significa que, para a utilização da plataforma em quaisquer dispositivos, não seria feita nenhuma cobrança.

O kit de desenvolvimento de software do Android ou Android SDK foi liberado para consulta pela primeira vez em novembro de 2007, em um ato descrito em tradução livre como “primeira olhada” do inglês “early look”.

¹ Esta seção foi elaborada com base nos dados presentes em Hashimi, Komatineni e MacLean (2010), Burnette (2010) e Gargenta (2011).

Em setembro de 2008 foi anunciado pela companhia telefônica T-Mobile, em parceria com a fabricante de aparelhos móveis HTC, que o aparelho T-Mobile G1 seria o primeiro aparelho disponibilizado ao público em geral a possuir a plataforma Android.



Figura 2-8: T-Mobile G1, o primeiro aparelho comercializado rodando a plataforma Android (Scientific American 2008).

Em outubro de 2008, como acordado pela Open Handset Alliance, o Google tornou o código fonte do SDK do Android disponível sob a licença Apache, mencionada anteriormente, caracterizando-o oficialmente como um programa de código aberto.

Também neste ano, a fim de incentivar os programadores a fazerem aplicativos para a plataforma e disseminar a utilização da mesma em dispositivos móveis, o Google disponibilizou um aparelho chamado Android Dev Phone, capaz de rodar aplicações de Android e de prover a experiência completa da plataforma sem necessitar de nenhum contrato com operadoras de telefonia.

Daí em diante o Google foi lançando atualizações ao sistema que acrescentavam cada vez mais recursos e possibilidades à plataforma. Esses fatos serão abordados a seguir, em forma de uma tabela mostrando o lançamento de cada nova versão da plataforma.

Ao acompanhar essa tabela é interessante verificar que o número da versão em si não é a informação mais significativa a ser considerada, pois independentemente da

modificação realizada na plataforma, desde uma pequena correção até uma grande modificação, esses números são modificados. O mais importante a se verificar, então, são os níveis da API.

Particularmente para os desenvolvedores, é interessante atentar qual o nível mais baixo da API que a aplicação a ser desenvolvida pode rodar e qual o nível de API mais utilizado pelos aparelhos na atualidade, garantindo, assim, o funcionamento do aplicativo a ser desenvolvido no maior número possível de aparelhos.

Tabela 2-2: Tabela de Versões da plataforma Android.

Version	Codename	API Level	Distribution
1.5	Cupcake	3	0.3%
1.6	Donut	4	0.6%
2.1	Eclair	7	5.2%
2.2	Froyo	8	19.1%
2.3 - 2.3.2	Gingerbread	9	0.4%
2.3.3 - 2.3.7		10	64.6%
3.1	Honeycomb	12	0.7%
3.2		13	2%
4.0 - 4.0.2	Ice Cream Sandwich	14	0.4%
4.0.3 - 4.0.4		15	6.7%

Fonte: (GOOGLE 2012)

2.2.4.2 Características Gerais²

Android é uma pilha de softwares para dispositivos móveis que inclui um sistema operacional, middleware e aplicações principais. O Android SDK fornece as

² Essa subseção foi desenvolvida com base nos dados presentes em Google (2012).

ferramentas necessárias para começar a desenvolver aplicações para essa plataforma utilizando a linguagem de programação Java.

Tem como principais características:

- Framework de aplicações: Fornece módulos de alto nível que serão usados para a criação das aplicações.
- Máquina virtual Dalvik: responsável por executar o código da aplicação no dispositivo.
- Navegador integrado: navegador baseado em WebKit.
- Gráficos otimizados: possui biblioteca customizada de gráficos de duas dimensões e biblioteca de gráficos de três dimensões baseada na especificação OpenGL ES 1.0.
- SQLite: suporte nativo para utilização de SQLite para armazenamento de dados estruturados.
- Suporte a variados tipos de mídia: áudio, vídeo e formatos de imagem, entre eles: MPEG4, H.264, MP3, AAC, AMR, JPG, PNG, GIF.
- Telefonia utilizando tecnologia GSM.
- Bluetooth, EDGE, 3G, e WiFi.
- Câmera, GPS, bússola e acelerômetro.
- Rico ambiente de desenvolvimento: oferece emulador de aparelho, ferramentas para depuração de código e análises de desempenho e memória e plugin específico para a IDE Eclipse.

2.2.4.3 Resumo da Arquitetura

A arquitetura deste sistema está dividida em quatro grandes componentes, que podem ser visualizados na figura 2-9.



Figura 2-9: Camadas da Arquitetura da Plataforma Android (GOOGLE 2012).

Na base desta plataforma encontra-se o kernel Linux versão 2.6. Esse componente atua como uma camada de abstração entre o hardware e o resto dos componentes da arquitetura. É principalmente utilizado pelos seus serviços de segurança, gerenciamento de memória, gerenciamento de processos, pilha de redes e modelo de drivers (GOOGLE 2012).

A segunda camada de arquitetura é composta pelas bibliotecas e pelos componentes de execução do Android. Podemos destacar aqui a variedade das bibliotecas disponíveis e a Dalvik VM.

As bibliotecas nativas são bibliotecas C/C++, geralmente retiradas da comunidade de código aberto com a finalidade de fornecer os serviços necessários para a camada de aplicação. Dentre as mais relevantes estão: WebKit, SQLite, SSL, OpenGL, SGL e FreeType.

Desenvolvida e escrita por Dan Bornstein, funcionário do Google, é a Dalvik VM que executa o código compilado no dispositivo. Essencialmente é uma máquina virtual Java otimizada para dispositivos móveis. Permite que múltiplas instâncias rodem ao

mesmo tempo e aproveita características do sistema operacional Linux nos termos de segurança e isolamento de processos (BURNETTE 2010).

Quanto à definição da camada de framework de aplicações, segundo Gargenta:

O framework de aplicações é um rico ambiente que fornece uma série de serviços para ajudar o desenvolvedor. Essa camada da plataforma é a que habilita os desenvolvedores a utilizarem sua criatividade para criar aplicações fantásticas (2011, nossa tradução).

Essa camada provê acesso total ao mesmo framework de APIs utilizado pelas aplicações principais. A arquitetura foi desenvolvida visando simplificar o reuso de componentes de forma que qualquer aplicação possa publicar suas capacidades para utilização por outras aplicações.

Dentre os componentes que constituem essa camada de framework de aplicações estão:

- Gerenciador de Atividades: responsável por gerenciar o ciclo de vida das aplicações e fornecer uma pilha de navegação.
- Gerenciador de Recursos: disponibiliza acesso a recursos que não são código, como gráficos, textos e arquivos de layout.
- Gerenciador de Conteúdo: possibilita que aplicações acessem dados de outras aplicações.
- Gerenciador de Localidade: possibilita o acesso aos serviços de localidade disponíveis pelo aparelho.
- Gerenciador de Notificações: responsável por gerenciar a visualização e a interação com as notificações mostradas ao usuário na barra de status.

Na última camada da arquitetura desse sistema, localizada no topo da pilha mostrada anteriormente, estão as aplicações. Nela ficam tanto as aplicações principais do sistema, que já vêm instaladas por padrão, como as aplicações instaladas posteriormente.

Esse sistema já vem com uma série de aplicações principais instaladas, entre elas um cliente de e-mail, um programa de SMS, calendário, mapas, navegador, contatos, entre outras.

3 MODELO DA APLICAÇÃO

Neste capítulo será apresentada a modelagem proposta para um aplicativo mobile para pesquisa de transportes públicos, que tem por finalidade disponibilizar informações sobre variados tipos de transportes públicos, como ônibus, lotação e táxi, com base na localidade do usuário ou em um endereço fornecido. Como poderá ser visto a seguir, os conceitos tratados anteriormente são a base fundamental da aplicação.

Primeiramente será descrito de forma mais detalhada o problema a ser resolvido pela aplicação e quais os objetivos que ela deverá atender ao final do seu desenvolvimento. Em seguida será apresentada a modelagem da aplicação em si, na qual poderão ser observados tanto aspectos gerais do funcionamento da aplicação, quanto aspectos específicos dos seus modos de funcionamento.

3.1 Identificação do Problema

Neste trabalho foi feito um estudo sobre LBS e dispositivos móveis. Tais conceitos serão o cerne da aplicação. Durante a definição do problema foi pensado que não se poderia simplesmente definir um objetivo singular, pois um dos desafios deste trabalho é incentivar o usuário, através da praticidade e da usabilidade da aplicação, a utilizar transportes públicos. Portanto, foi definido o seguinte problema: disponibilizar para um usuário portador de um dispositivo móvel, informações sobre diversos tipos de meios de transporte público baseadas na sua localização ou em uma localização fornecida, de maneira prática, usual, personalizada e útil.

3.2 Modelagem da Solução

Esta seção mostrará como foi efetuada a modelagem da aplicação, descrevendo, primeiramente, a importância dos requisitos do sistema e como eles foram definidos, bem como as definições gerais do sistema.

3.2.1 Requisitos do sistema

Na base da modelagem dos projetos estão os seus requisitos, é fundamental que ao se desenvolver um sistema saibamos quais as necessidades das partes interessadas – usuários, clientes, fornecedores, desenvolvedores, empresas – e o que o sistema deverá fazer para satisfazer essas necessidades.

Sem uma base de requisitos relativamente estável, um projeto em desenvolvimento só pode fracassar. É como sair para uma viagem marítima sem ideia do destino e sem mapa de navegação. Requisitos fornecem tanto o mapa de navegação quanto os meios de caminhar na direção correta (Hull, Jackson e Dick 2005).

É importante ressaltar que os requisitos da aplicação além de disponibilizarem a base para o planejamento de como o sistema será desenvolvido, também servirão como testes para aceitação da solução após o seu desenvolvimento.

3.2.1.1 Estrutura

Os requisitos da aplicação proposta neste trabalho seguirão a estrutura de requisitos funcionais e requisitos não funcionais, descrita a seguir:

- **Requisitos funcionais:** Esses requisitos são as definições dos serviços que o sistema deve fornecer; como o sistema deve reagir a partir de determinadas entradas e como o sistema deve se comportar em situações específicas. Em alguns casos os requisitos funcionais podem também explicitar o que o sistema não deve fazer (KOTONYA e SOMMERVILLE 1998).

- **Requisitos não funcionais:** Os requisitos denominados não funcionais descrevem os atributos qualitativos que um sistema deve apresentar em relação as suas funcionalidades (BEZERRA 2007).

Além de seguir essa estrutura, os tipos de requisitos serão classificados quanto ao seu caráter de importância e contribuição ao objetivo final da aplicação, como segue:

- **Obrigatório:** requisito que deve ser atendido para que o sistema tenha suas funcionalidades básicas ao fim do desenvolvimento, possui a maior prioridade.
- **Importante:** requisito que, se atendido, acrescentará grande valor à aplicação ajudando a cumprir com totalidade seus objetivos, mas que se não atendido não comprometeria as funcionalidades básicas da aplicação.
- **Desejável:** requisito que, se atendido, acrescentaria funcionalidades pouco expressivas ou que não teria relação direta com o objetivo final da aplicação.

3.2.1.2 Requisitos Funcionais

Na tabela 3-1 serão relacionados os requisitos funcionais.

Tabela 3-1: Requisitos Funcionais

Identificação	Nome	Descrição	Categoria
RF01	Disponibilizar a visualização e o uso da localização atual do usuário.	Com base no hardware disponível e habilitado no aparelho do usuário, disponibilizar a sua localização de forma que o usuário possa visualizar, nos mapas presentes na aplicação, onde ele está posicionado, bem como utilizar os dados da localização (latitude, longitude) para finalidades do sistema.	Obrigatório
RF02	Disponibilizar o local e as linhas de paradas de ônibus a partir de coordenadas.	Com base nas coordenadas fornecidas pelo sistema (que podem ser provenientes tanto da posição atual do usuário quanto de um endereço desejado), disponibilizar uma lista de paradas de ônibus	Obrigatório

		próximas a essas coordenadas, bem como as linhas que passam nessa parada.	
RF03	Disponibilizar o código, o nome e o itinerário de linhas de ônibus ou lotação.	A partir de uma identificação própria do sistema, recuperar o código, o nome e o itinerário de uma linha de ônibus ou lotação.	Obrigatório
RF05	Disponibilizar o local e, quando disponível, o telefone de pontos de táxi a partir de coordenadas	Com base nas coordenadas fornecidas pelo sistema (que podem ser provenientes tanto da posição atual do usuário quanto de um endereço desejado), disponibilizar uma lista de pontos de táxi próximos a essas coordenadas, bem como o telefone desses pontos de táxi quando disponível.	Obrigatório
RF06	Permitir ao usuário fazer uma busca por texto de um local sobre o qual deseja informações.	Com base no texto fornecido pelo usuário, verificar se existem pontos geográficos nesse endereço de forma que seja possível revelar informações (paradas de ônibus, por exemplo) com base nesse ponto.	Obrigatório
RF07	Permitir ao usuário fazer uma busca por texto de uma linha de	Com base no texto fornecido pelo usuário, verificar a existência de nomes ou códigos	Obrigatório

	ônibus ou lotação.	de linhas de ônibus ou lotação que se relacionem com esse texto.	
RF08	Permitir ao usuário fazer a busca de uma rota utilizando linhas de ônibus.	Com base nas informações de origem e destino fornecidas pelo usuário, fazer uma busca por linhas de ônibus que possuam paradas próximas a esses endereços e retornar as possibilidades de rotas que o usuário poderia utilizar.	Obrigatório
RF09	Permitir ao usuário personalizar aplicação.	Tornar possível a modificação de configurações iniciais da aplicação pelo usuário, de forma que essas modificações sejam armazenadas e sempre carregadas e utilizadas quando o usuário for fazer uso da aplicação, possibilitando uma experiência customizada ao gosto do usuário.	Obrigatório
RF10	Permitir a visualização e interação no mapa das informações disponibilizadas pelo programa.	Disponibilizar visualização e interação de informações que o usuário deseja visualizar em um mapa, respeitando a posição geográfica dessas informações e distinguindo elas de maneira intuitiva.	Obrigatório
RF11	Possibilitar a utilização de serviços online para	Disponibilizar para o sistema a possibilidade de armazenar	Obrigatório

	a obtenção dos dados do programa.	somente a quantidade necessária de dados para a visualização das informações desejadas, fazendo a busca do restante das informações a partir de um servidor na internet.	
RF12	Possibilitar a busca de informações em um banco de dados local.	Disponibilizar para o sistema a possibilidade de recuperar localmente todos os dados necessários para suas funcionalidades, dispensando a utilização de uma conexão com a internet para esse tipo de serviço.	Obrigatório
RF13	Disponibilizar a busca de linhas de lotação pela proximidade com uma coordenada.	Buscar linhas de lotação pela eventual proximidade de algum dos pontos de seu itinerário a uma coordenada fornecida.	Importante
RF14	Disponibilizar ao usuário a customização do raio de busca por ponto geográfico.	Permitir ao usuário customizar dentro de uma margem de valores pré-estabelecidos pelo sistema, um raio padrão a ser utilizado nas buscas de informações por ponto geográfico.	Importante
RF15	Disponibilizar variados modos de exibição dos mapas.	Permitir ao usuário escolher entre variados modos de visualização e componentes do mapa, de forma a moldar a	Importante

		aplicação conforme suas necessidades.	
RF16	Disponibilizar ao usuário maiores informações sobre as linhas de ônibus.	Possibilitar que o usuário tenha maiores informações sobre as linhas de ônibus, tais como: tabela de horários, consórcio e disponibilidade de veículos especiais.	Importante
RF17	Permitir ao usuário fazer uma ligação para o número de um ponto de táxi diretamente do aplicativo.	Permitir ao usuário que, ao visualizar a informação sobre um ponto de táxi que possui número telefônico, ao selecionar esse número, seja encaminhado para a tela de ligação telefônica com o esse número carregado.	Importante
RF18	Permitir ao usuário cadastrar pontos de interesse.	Permitir ao usuário cadastrar pontos de interesse de sua escolha, que ficariam sempre marcados no mapa com ícones distintos, facilitando sua identificação, e teriam suas coordenadas salvas para realizar as pesquisas a partir deles mais facilmente.	Importante
RF19	Acrescentar ao resultado da pesquisa por rotas o caminho que o usuário deverá percorrer a pé até as	Acrescentar informações sobre a parte da rota não contemplada pela linha que o usuário terá que percorrer a pé, de forma que o mesmo possa	Desejável

	paradas.	visualizar no mapa qual o trajeto sugerido para se percorrer até as paradas.	
RF20	Acrescentar ao resultado de pesquisa por rotas o tempo e o custo estimado para o percurso do trajeto sugerido.	Acrescentar ao resultado da busca por rotas uma estimativa de quanto tempo o usuário demoraria no mínimo, em média e no máximo para percorrer o trajeto sugerido, juntamente com o custo do meio de transporte que está sendo sugerido.	Desejável
RF21	Sugestão de passeios turísticos	Permitir ao usuário acessar uma lista de passeios turísticos, bem como quais pontos serão vistos no passeio, quais linhas e meios de transporte o usuário irá utilizar e o custo total do passeio.	Desejável
RF22	Inclusão de alarmes por proximidade	Permitir ao usuário que cadastre alarmes que o alertem quanto à proximidade de um ponto de interesse, parada de ônibus, linha de lotação ou ponto de táxi.	Desejável
RF23	Função de acompanhamento do trajeto	Permitir ao usuário que uma vez visualizado uma rota a se percorrer, selecione aquela rota para acompanhamento e seja guiado de forma a facilitar a	Desejável

		compreensão dos passos que deve tomar para seguir o trajeto corretamente.	
RF24	Função de taxímetro	Permitir ao usuário que seja possível estimar o valor total de um trajeto percorrido por um táxi com base na origem, destino, dia, hora e quantidade de passageiros.	Desejável
RF25	Permitir ao sistema realizar buscas utilizando mais de uma linha ou tipo de meio de transporte.	Permitir ao sistema fazer buscas mais complexas utilizando cruzamento entre linhas e tipos de meios de transporte diferentes para que seja possível retornar rotas mais precisas e com maior valor de utilidade para o usuário.	Desejável
RF26	Permitir a utilização da aplicação por outros aplicativos.	Permitir a outros aplicativos chamar a aplicação, passando alguns parâmetros que permitam à aplicação a descoberta de informações interessantes ao usuário com base nesses parâmetros.	Desejável

3.2.1.3 Requisitos não funcionais

Na tabela 3-2 serão relacionados os requisitos não funcionais.

Tabela 3-2: Requisitos Não Funcionais

Identificação	Nome	Descrição	Categoria
RNF01	Usabilidade	O sistema deve prover suas funcionalidades ao usuário da maneira mais usual possível, de modo que em poucas interações e de forma intuitiva o usuário consiga encontrar as informações que deseja facilmente.	Obrigatório
RNF02	Customização	Deve ser possível customizar o sistema, tanto quanto possível, de forma a ele se moldar, visto as necessidades dos mais variados tipos de usuários, sem comprometer as suas funcionalidades.	Obrigatório
RNF03	Compatibilidade	Deve ser possível executar a aplicação plenamente em todas as plataformas suportadas, oferecendo uma experiência semelhante de interação quaisquer que sejam essas plataformas e suas versões.	Obrigatório
RNF04	Desempenho quanto à utilização de recursos	O sistema deve usar os recursos disponibilizados pelos aparelhos móveis de forma exata as suas necessidades, não desperdiçando a utilização de nenhum destes recursos, evitando o desperdício de informações e aumentando a	Obrigatório

		autonomia da bateria do aparelho.	
RFN05	Desempenho quanto ao tempo de resposta	O sistema deve ter um limite na obtenção das respostas a serem fornecidas ao usuário dependendo da operação requisitada, de forma que o usuário não fique esperando infinitamente por uma opção e seja corretamente informado quando alguma operação está demorando mais do que o previsto ou estourou seu tempo de limite.	Obrigatório
RFN06	Tratamento de erros	A aplicação deve tratar erros inerentes ao sistema em que está rodando, como problemas na conexão, eventual fechamento da aplicação por falta de memória, falta de espaço local para armazenamento de dados, entre outros, de forma que seu funcionamento não seja prejudicado ou, em última instância, que o usuário seja avisado do porquê do não funcionamento da aplicação.	Obrigatório
RFN07	Desempenho quanto à utilização da conexão de dados	A aplicação deve utilizar somente o necessário da conexão de dados provida pelo aparelho móvel, de forma a não ocupar desnecessariamente recursos compartilhados com outras aplicações.	Obrigatório
RFN08	Retorno do tempo	O usuário deve ter retorno visual	Importante

	das operações ao usuário	claro da demora de suas operações, de modo que saiba estimar o quanto tempo cada uma delas demorará a atender as suas necessidades.	
--	--------------------------	-------------------------------------------------------------------------------------------------------------------------------------	--

3.2.2 Características do sistema

Tendo em vista o fato de que existem algumas limitações que necessitam de observação pelo desenvolvedor quando se está implementando um sistema destinado a rodar em uma plataforma móvel, foi observado um conjunto dessas limitações que poderiam impactar negativamente o funcionamento do sistema proposto. A seguir estão listadas estas limitações e as justificativas de terem sido escolhidas como principais empecilhos para o pleno funcionamento do aplicativo, bem como as decisões tomadas para suprimir possíveis problemas causados por essas limitações e outras definições quanto ao funcionamento geral da aplicação.

3.2.2.1 Limitações consideradas

- Espaço de armazenamento interno: como a utilização da aplicação não está sendo limitada a um tipo de hardware específico, não é possível conhecer em tempo de desenvolvimento o espaço de armazenamento interno disponível para a aplicação ser executada e armazenar os seus dados.
- Disponibilidade da conexão com a internet: a possibilidade do dispositivo se conectar a internet não nos garante a velocidade nem a qualidade dessa conexão e, principalmente, não nos dá a garantia da disponibilidade dessa conexão. Por exemplo, é possível que o aparelho não tenha um plano de dados que o permita acessar a internet afora momentos em que utiliza uma rede sem fio local para acessar tal recurso.
- Telas e modos de interação variados: ressaltando novamente que não se está limitando a aplicação a um tipo de hardware específico, é de fundamental importância que se leve em conta os diferentes tipos de tamanhos de telas e modos de interação que podem estar disponíveis dependendo de onde está sendo utilizada a aplicação. Por exemplo, a aplicação poderá estar sendo executada tanto em um aparelho que possui

uma tela com alta resolução e interação por toques na tela disponível, quanto em um dispositivo onde a resolução da tela é baixa, não se tem a possibilidade de captar os toques na tela e a interação se dará por botões disponíveis no aparelho.

3.2.2.2 *Definições*

3.2.2.2.1. Modo online

Para lidar com a limitação de espaço disponível no aparelho, está previsto que a aplicação possua um modo de funcionamento online. Nesse modo, toda consulta realizada para obtenção de informações referentes aos meios de transporte que o usuário possa fazer será feita utilizando-se uma arquitetura do tipo cliente servidor. Ou seja, todos os dados de consulta estarão presentes em um servidor, enquanto no cliente que roda a aplicação só estará sendo ocupado o espaço de armazenamento referente aos resultados das consultas, pelo tempo em que estes são de relevância para o usuário.

Este modo de funcionamento visa possibilitar ao usuário que possui um dispositivo com espaço de memória de armazenamento pequeno ou inexistente executar a aplicação sem problemas, desde que tenha a sua disposição uma conexão com a internet e seja capaz de requisitar as informações que deseja do servidor de dados.

3.2.2.2.2. Modo off-line

Quanto aos quesitos disponibilidade e qualidade de conexão com a internet, é previsto que a aplicação possua um modo de funcionamento “off-line”. Neste modo a aplicação dispensará a utilização de uma conexão com a internet para a obtenção de informações referentes às consultas por parte do usuário. Para isso, está previsto que a aplicação seja capaz de fazer consultas e manter um banco de dados local que ficará localizado na memória de armazenamento do aparelho, dispensando, assim, a utilização de um servidor localizado na rede para a aquisição das informações.

Este modo de funcionamento visa possibilitar ao usuário que não possui um plano de dados - que permita ao seu dispositivo móvel conectar-se à internet independente da sua localização -, ou um usuário que não tenha internet disponível para o seu aparelho em um determinado momento, utilizar o sistema proposto neste trabalho de forma que

consiga ter acesso às mesmas informações que o servidor de dados da aplicação ofereceria para um usuário do modo online. É importante observar que este modo de funcionamento ocupa consideravelmente a memória de armazenamento do aparelho em uso, e é necessário que o aparelho tenha espaço disponível para a utilização correta da aplicação.

3.2.2.2.3. Disposição das funções na tela

Em relação à última limitação mencionada no início desta seção, que relata a variedade da resolução das telas e da forma de interação dos aparelhos, é previsto que o design e a forma de interação que será utilizada na aplicação sejam desenvolvidos de forma que possam se adaptar às variadas especificações de aparelhos. Isso deve ser feito de maneira a garantir que em todos os aparelhos que forem capazes de rodar a aplicação (os quais tiverem as especificações mínimas de software) o usuário tenha uma experiência fluída e natural, por mais limitado que sejam os recursos oferecidos pelo aparelho.

3.2.2.2.4. Função de Pesquisa

Esta decisão não tem relação específica com as limitações citadas anteriormente, mas parte da observação de como atingir com excelência o primeiro requisito não funcional do objeto deste trabalho que é a usabilidade. Enquanto o usuário estiver executando a aplicação, estará disponível uma gama de informações sobre os meios de transporte que o mesmo pode desejar obter maiores detalhes. Dentro das possibilidades disponibilizadas pelo aplicativo para consultar essas informações está a pesquisa, que deve se dar da forma mais simples e útil possível para o usuário. Para isso foi destinada uma atenção especial.

Primeiramente se pensou que seria necessário fazer essa pesquisa em uma determinada quantidade de eventos, esses eventos questionariam o usuário em relação ao conteúdo ao qual ele quer pesquisar e depois possibilitariam ao usuário finalmente fazer a pesquisa. Depois de se investir um bom tempo para o desenvolvimento desta característica da aplicação foi observado que essa maneira de realizar a uma pesquisa não é a mais ideal.

Basta observar uma das empresas mais competentes e populares nesse aspecto (Google) para termos uma melhor ideia de como realizar de forma eficaz essa tarefa e disponibilizar uma boa experiência de usabilidade ao usuário. Quando vamos fazer uma pesquisa utilizando-se da ferramenta de busca do Google, não é necessário especificar o tipo de dado que estamos procurando, porém desde que bem dispostas as respostas e bem feita a busca pelo sistema, teremos um resultado se, não exatamente igual ao que esperávamos, muito próximo.

Tendo esse ótimo e muito popular sistema como espelho, foi pensado que a melhor maneira de realizar essa busca é aproximar ao máximo do sistema do Google o modo de funcionamento da função de busca do aplicativo. Para isso é definido que o usuário fará a pesquisa somente digitando as palavras pelas quais quer pesquisar, sejam elas referentes a um endereço, a uma linha de ônibus, a uma linha de lotação, a um bairro da cidade ou a um ponto de interesse cadastrado pelo usuário. Posteriormente os resultados dessa busca serão disponibilizados para o usuário e o mesmo poderá seguir a utilização da aplicação.

3.3 Modelagem da Arquitetura

Nesta seção será descrito o padrão de arquitetura utilizado para a modelagem que, juntamente com as seções anteriores, constitui a modelagem da solução.

O padrão de arquitetura utilizado para modelar esta aplicação foi o MVP (Model View Presenter).

Segundo a definição de MVP por Pottel (1996), seis perguntas divididas em duas classes devem ser respondidas para se chegar à conclusão de que o usuário está criando uma aplicação baseada em MVP, são elas:

Quanto ao gerenciamento de dados:

1. Quais são os dados? (Model)
2. Como especificar os dados? (Selections)
3. Como modificar os dados? (Commands)

Quanto à interface com o usuário:

4. Como mostrar os dados? (View)
5. Como os eventos podem modificar os dados? (Interactor)
6. Como ligar todas essas estruturas? (Presenter)

Estes conceitos podem ser mais facilmente analisados e relacionados com a arquitetura MVC pela visualização da figura a seguir.

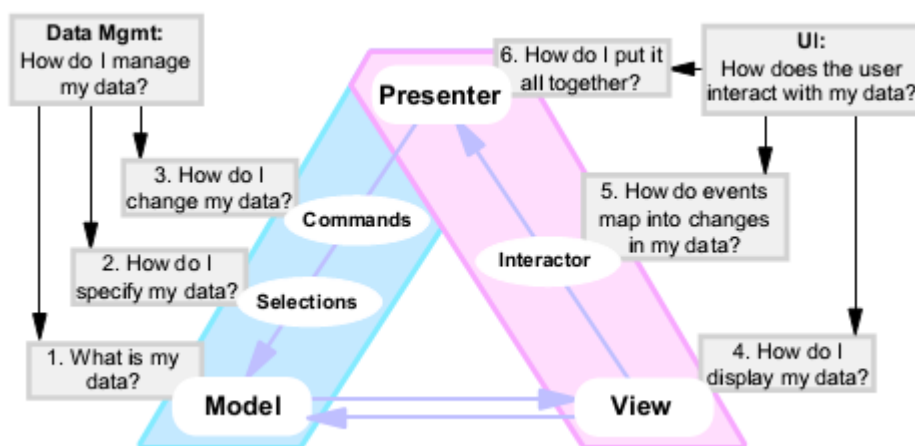


Figura 3-1: Padrão de Arquitetura MVP (Pottel 1996).

A partir desse modelo foram criados módulos que representam cada funcionalidade descrita acima. Estes módulos foram distribuídos como é descrito a seguir.

No módulo relacionado às Views do programa, ficarão as definições principais de cada tela de interação com o usuário, que na plataforma Android são feitas utilizando-se da linguagem de marcação de texto XML.

O módulo relativo aos agentes de interação (Interactors) do programa definirá a interação com os dados apresentados pelas Views. Na plataforma Android estes agentes serão representados pelas Activities.

A modelagem dos dois componentes citados até agora foi feita utilizando um software para confecção de wireframes (estrutura simbólica da interface de usuário). Nesses wireframes foram destacadas as maneiras de interação do usuário e o comportamento da aplicação a partir dessas entradas, constituindo, assim, uma definição geral das telas e da interface da aplicação. A figura a seguir representa o wireframe da tela principal do aplicativo.

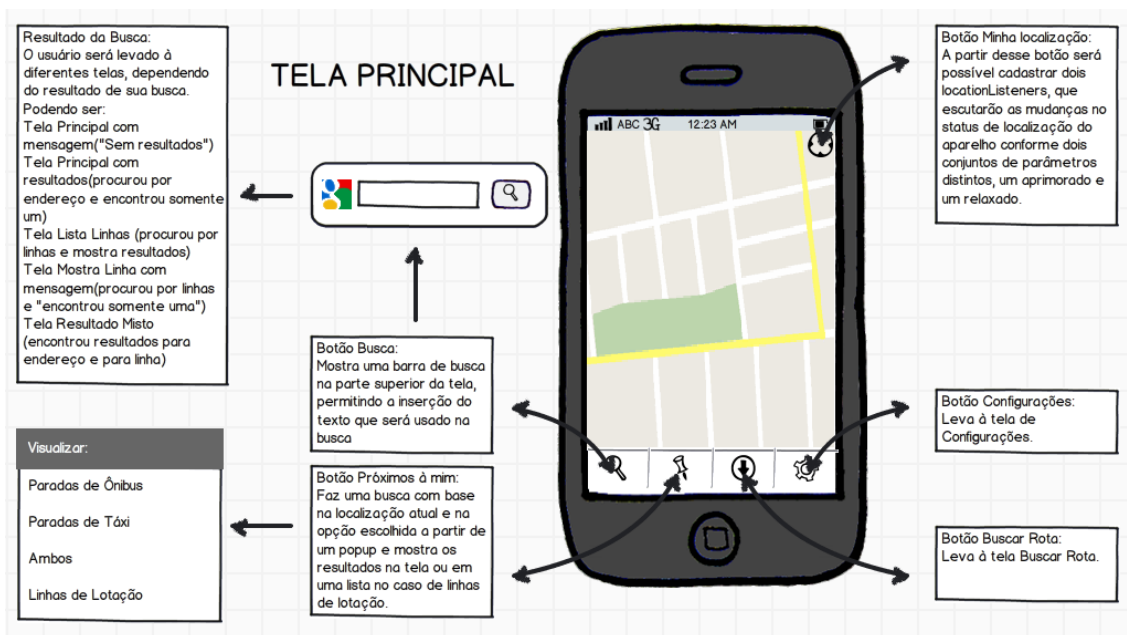


Figura 3-2: Wireframe da Tela Principal.

No apresentador (Presenter) será feito o tratamento e a preparação dos dados de forma que possam ser utilizados corretamente para a visualização e interação nas Views, as classes responsáveis por lidar com o resultado de buscas realizadas pelo usuário, a adaptação dos dados para visualização nos mapas e o cálculo de coordenadas para utilização nas buscas estarão presentes nesse componente.

Os comandos (Commands) definirão como será feita a aquisição e modificação dos dados utilizados pelo programa. As classes responsáveis pela aquisição de dados na internet ou no banco de dados local, pela manutenção do banco de dados local e pelo armazenamento dos dados de preferência do usuário, representarão este componente.

No item relativo às seleções (Selections) estará a classe responsável por prover os dados necessários ao apresentador a partir da necessidade verificada e dos dados disponíveis.

Por fim, no componente Model estarão os modelos dos dados a serem utilizados pela aplicação. Estes modelos incluem as definições da estrutura do banco de dados e das classes que irão armazenar os dados para uso temporário pelo programa (por exemplo, a classe Stop que será utilizada para armazenar os dados de uma parada).

No caso em que a aplicação utiliza o servidor de dados da Procempa, a parte de modelagem, seleção e comandos dos dados não acontece somente na aplicação, caracterizando uma arquitetura cliente/servidor. Assim sendo, foi adicionada uma terceira camada à figura mostrada anteriormente, na qual o apresentador faz requisições ao servidor da Procempa e depois armazena os dados localmente para a utilização.

4 IMPLEMENTAÇÃO DA APLICAÇÃO

Neste capítulo serão abordados os tópicos relativos à implementação da aplicação proposta neste trabalho, chamada POAMóvel.

Primeiramente serão descritas as decisões tomadas quanto à maneira de implementar o modelo proposto pelo trabalho, juntamente com as principais partes da aplicação e como estas foram implementadas. Após, serão descritos os modos de funcionamento da aplicação. Em seguida, teremos a validação do sistema; nela será realizada uma análise dos objetivos alcançados pela aplicação. Por fim, será mostrada uma comparação da aplicação desenvolvida com soluções semelhantes existentes no mercado.

4.1 Justificativas da Escolha de Plataforma para Implementação

Dada essa série de informações sobre as plataformas móveis e as necessidades da aplicação, podemos destacar com maior clareza os principais motivos da escolha dessa plataforma de desenvolvimento Android.

- Arquitetura com suporte nativo para uma gama de recursos de hardware: é evidente que existem variados aparelhos e plataformas de desenvolvimento com recursos semelhantes, porém, a maneira com que foi pensada a arquitetura deste sistema permite aos desenvolvedores que forem utilizá-lo aproveitar os recursos de forma eficaz. Isso garante, além de uma experiência semelhante para usuários com os mais diversos aparelhos móveis, o aproveitamento dos recursos do aparelho pela garantia de acesso total do desenvolvedor às mesmas APIs utilizadas pelas aplicações principais.
- Vasta e valiosa documentação e comunidade de desenvolvedores: o suporte disponível para quem deseja tanto ingressar como aprimorar seu conhecimento no desenvolvimento de programas para Android é extenso. Podemos citar como exemplos

deste suporte fóruns, blogs, sites, livros, entre outros. Porém, o que mais valoriza essa grande quantidade de conteúdo sobre como se deve proceder nos mais variados casos possíveis durante o desenvolvimento de uma aplicação para Android é a vivacidade, o crescimento e a qualidade percebidas nos meios de suporte citados anteriormente. Da mesma forma como é possível ressaltar o rápido lançamento de novas versões da plataforma pelo fato de sete grandes atualizações terem sido lançadas em três anos (vide capítulo sobre a plataforma Android), podemos observar como a comunidade de suporte acompanhou e incentivou tal crescimento, possibilitando aos desenvolvedores uma ótima forma de tentarem cada vez mais explorar novas características e criar novas ideias e possibilidades de aplicativos.

- Distribuição dos aparelhos e crescimento da plataforma no país: segundo matéria retirada do site Olhar Digital:

Em 2011 foram vendidos cerca de 10 milhões de smartphones no Brasil, e, de acordo com um estudo do IBOPE, os dispositivos inteligentes representam 10% do mercado nacional de celulares. Apesar disso, o Google está confiante. "O número de aparelhos com Android ativados no Brasil aumentou 400% no último ano e estamos vendo os preços caírem", afirmou Hugo Barra, gerente de produtos móveis da empresa. "No Brasil, especificamente, além da queda nos preços de smartphones, o Google acredita que o interesse do consumidor no Android é alto. As pessoas já sabem o que é o sistema e vão às lojas perguntando quais são os celulares com Android. Alguns chegam a perguntar qual é a versão específica usada em cada aparelho", disse Barra (2011).

- Conhecimento do desenvolvedor: o conhecimento sobre a plataforma Android foi adquirido através de experiências profissionais e trabalhos para a faculdade, o que motivou o autor a aprofundar-se no assunto e a tomar a decisão final pela escolha dessa plataforma.

4.2 Modelo de Desenvolvimento da Aplicação

O modelo utilizado para o desenvolvimento deste trabalho foi o de modelagem por prototipação.

4.2.1 Modelagem por Prototipação

Apesar de praticamente todos os requisitos da aplicação estarem definidos antes do começo do seu desenvolvimento, não se sabia com precisão como modelar a aplicação

de forma a atender a todos eles, por esse motivo foi escolhida a modelagem por prototipação.

Para essa modelagem foi utilizada uma combinação de dois tipos de prototipação, a evolucionária e a descartável, descritas a seguir.

4.2.2 Prototipação Evolucionária

Segundo McConnell (1996), prototipação evolucionária é um modelo de ciclo de vida de software em que o sistema é desenvolvido em incrementos, permitindo que seja prontamente modificado com base nas respostas do usuário final e do consumidor.

Nesse tipo de abordagem primeiro são desenvolvidas pequenas partes do sistema e depois vai se construindo o sistema em sua totalidade a partir dessas partes. A figura 4-1 ilustra como se dá esse processo:

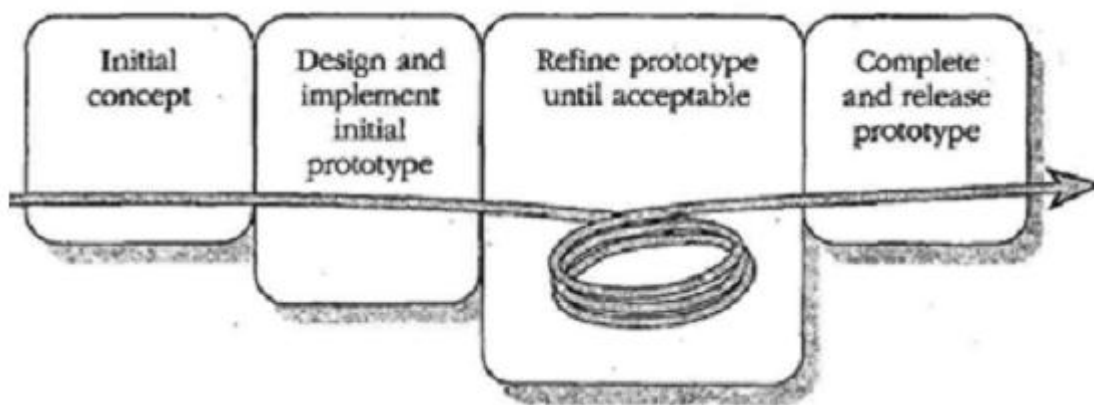


Figura 4-1: Processo de Prototipação Evolucionária

4.2.3 Prototipação Descartável

Pela definição de prototipação descartável presente em McConnell (1996), ela consiste no desenvolvimento de código para a exploração de fatores críticos para o sucesso do sistema e depois no descarte desse código. A implementação desse tipo de prototipação utiliza linguagens de programação ou práticas de desenvolvimento que são mais rápidas que a linguagem de programação ou as práticas de desenvolvimento finais. É possível utilizar a prototipação descartável em qualquer fase do projeto a fim de esclarecer os requisitos, definir uma decisão de arquitetura, comparar opções de design ou desenvolvimento ou testar o desempenho do produto.

4.2.4 Combinação e Utilização

Em princípio foram definidas cinco áreas de risco para a aplicação, nas quais seriam desenvolvidos protótipos descartáveis. Estas áreas foram definidas principalmente com base na ausência parcial e em alguns casos total do conhecimento necessário sobre como implementar e solucionar o problema proposto. São elas: aquisição da posição do usuário, obtenção dos dados de consulta utilizando o servidor da Procempa, armazenamento e recuperação de dados utilizando um banco de dados local SQLITE.

A partir dessas definições foram feitos protótipos simplificados para cada uma dessas funcionalidades. Por exemplo: no caso da obtenção de dados a partir do servidor da Procempa foi desenvolvido um protótipo que fazia uma simples requisição ao servidor sobre os dados de uma linha de ônibus e traduzia os dados da resposta em JSON do servidor para instanciar um objeto com as informações.

Todos os protótipos foram desenvolvidos e comprovaram a possibilidade de implementação de todas as áreas de risco verificadas acima. Como foram utilizados somente para prova das funcionalidades e foram desenvolvidos seguindo poucas regras de codificação, os protótipos, como parte da prototipação descartável, foram descartados. Deles ficaram boas noções dos desafios que seriam enfrentados a seguir e de como esses desafios poderiam ser superados.

Após a devida identificação das áreas de risco da aplicação sabia-se como estas seriam tratadas. A partir daí foram estabelecidos os módulos a serem utilizados na prototipação incremental.

Nessa etapa foi definido que todos os módulos da aplicação seriam desenvolvidos de forma incremental e que a cada verificação de possível melhoria ou correção, estes módulos seriam modificados até se obter uma versão final.

4.3 Obtenção dos dados sobre a localização do aparelho

Nesta seção será descrito como a tarefa relacionada pelo item RF01 da tabela de requisitos - no qual é descrita a necessidade da solução de prover a localidade atual do aparelho - foi implementada.

Para a obtenção da localidade do aparelho, foi utilizada a classe do Android chamada `LocationManager`. Essa classe disponibiliza o acesso aos serviços de localização do aparelho. Estes serviços possibilitam que aplicações obtenham atualizações periódicas sobre a localização geográfica do aparelho ou que o sistema execute uma determinada ação quando o aparelho se aproxima de uma localização geográfica pré-estabelecida.

Ao utilizar esses serviços de localização é necessário que se atente aos seguintes aspectos:

- Existem diversas maneiras de localizar um aparelho móvel, GPS, triangulação do sinal do telefone e Wi-Fi. A determinação da fonte dos dados de localidade é fundamental nos aspectos de precisão, velocidade do retorno de informação e eficiência de bateria.
- A movimentação do usuário deve ser levada em conta, deve se detectar quando esse movimento ocorreu e calcular a posição atualizada do aparelho.
- Um dos maiores problemas em se lidar com a determinação da localização de um aparelho móvel é a variação de precisão das posições informadas. Nesse quesito é necessário garantir que não se substitua incorretamente uma posição com maior precisão por uma pior e que por pior que seja a aproximação da localização do usuário seja possível utilizá-la com uma aproximação razoável.

Para lidar com estes problemas foram tomadas as seguintes definições:

```

24
25@  /**
26  * Method responsible for defining criteria for the listeners, getting a first approximate location and registering the location
27  * listeners
28  */
29@  public static void registerLocationListeners() {
30      LocationManager = (LocationManager) MainActivity.context.getSystemService(Context.LOCATION_SERVICE);
31
32      // Initialize criteria for location providers
33      Criteria fine = new Criteria();
34      fine.setAccuracy(Criteria.ACCURACY_FINE);
35      Criteria coarse = new Criteria();
36      coarse.setAccuracy(Criteria.ACCURACY_COARSE);
37
38      // Get at least something from the device,
39      // could be very inaccurate though
40      currentLocation = LocationManager.getLastKnownLocation(LocationManager.getBestProvider(fine, true));
41
42      if (listenerFine == null || listenerCoarse == null)
43          createLocationListeners();
44
45      // Keeps updating about every 500 ms until
46      // accuracy is about 1000 meters to get quick fix.
47      LocationManager.requestLocationUpdates(LocationManager.getBestProvider(coarse, true), 500, 1000, listenerCoarse);
48      // Will keep updating about every 500 ms until
49      // accuracy is about 50 meters to get accurate fix.
50      LocationManager.requestLocationUpdates(LocationManager.getBestProvider(fine, true), 500, 50, listenerFine);
51  }

```

Figura 4-2: Location Listeners.

Como, geralmente, o tempo de espera para a primeira resposta do sistema de localização pode ser bastante longo, enquanto novos dados de localização não são obtidos será utilizada a última localização disponível, isso pode ser visto na linha 40 da figura 4-2.

Quanto aos quesitos utilização da bateria e precisão na informação da localidade, foram utilizados dois serviços diferentes para obtenção da localização. Um serviço utiliza critérios mais relaxados para aquisição de posições e o outro utiliza critérios mais aprimorados. A definição dos critérios pode ser vista nas linhas 32 a 36 e o registro dos serviços de localização nas linhas 45 a 50 (figura 4-2).

Ambos os serviços são desligados quando se obtêm uma posição satisfatória, como pode ser visto na 4-3.

```

76 public void onLocationChanged(Location location) {
77     currentLocation = location;
78     locationAvailable = true;
79     if (location.getAccuracy() < 1000 && location.hasAccuracy())
80         locationManager.removeUpdates(this);
81 }
82

```

Figura 4-3: Desligamento dos Serviços de Localização.

4.4 Obtenção dos dados para consulta de linhas e paradas (modo online)

Para obtenção dos dados sobre as linhas e paradas a serem usadas na aplicação foi utilizado o sistema disponibilizado pela Procempa, chamado PoaTransporte. Ele consiste em um site na internet onde é possível realizar requisições utilizando protocolo HTTP para obter uma série de dados fornecidos utilizando o protocolo JSON.

Foram feitos contatos com a empresa com o intuito de manifestar o interesse na utilização do sistema, tais contatos foram bem recebidos e a empresa se prontificou a responder quaisquer dúvidas que surgissem durante o desenvolvimento da aplicação móvel, ressaltando a possibilidade de utilização do sistema deles, que é aberto e disponibilizado a qualquer pessoa.

Essas operações de requisição de dados ao site da Procempa foram implementadas a partir das consultas disponíveis pelo sistema, detectadas utilizando-se de uma análise do

código de interação do sistema com o seu o servidor de dados. A seguir consta uma lista das funções detectadas:

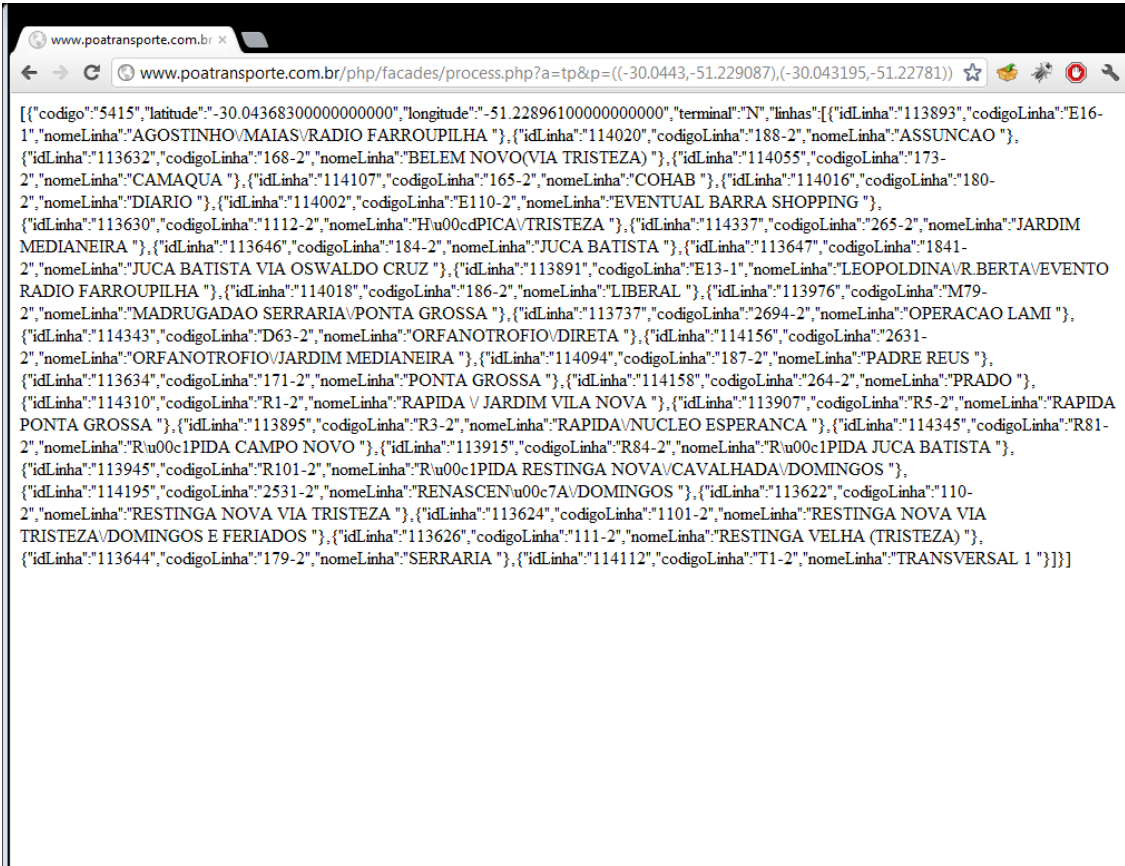
- Obtenção de paradas de ônibus conforme limites de coordenadas geográficas: a partir dessa consulta é possível receber uma lista de paradas de ônibus compostas por um código de referência, uma coordenada geográfica (latitude e longitude), indicação que mostra se a parada é referente a um terminal de ônibus ou não e uma lista com as linhas que atendem a essa parada. Esta consulta toma como parâmetros dois pontos geográficos que devem ser passados em forma de latitude e longitude em graus. Estes pontos geográficos são utilizados como limites para a busca, onde o primeiro será o limite inferior e o segundo o limite superior, como pode se verificar na figura 4-4.



Figura 4-4: Área Utilizada na Pesquisa.

A lista de linhas retornadas referente à parada de ônibus segue a seguinte estrutura: identificação da linha, código da linha e nome da linha. Na figura 4-5 pode ser

verificado um exemplo de uma requisição de dados feito pelo navegador de um desktop a fim de melhor compreendermos como se dá essa comunicação.



```

[{"codigo":"5415","latitude":"-30.0436830000000000","longitude":"-51.2289610000000000","terminal":"N","linhas":[{"idLinha":"113893","codigoLinha":"E16-1","nomeLinha":"AGOSTINHO/MAIAS/RADIO FARROUPILHA "},{idLinha":"114020","codigoLinha":"188-2","nomeLinha":"ASSUNCAO "},{idLinha":"113632","codigoLinha":"168-2","nomeLinha":"BELEM NOVO(VIA TRISTEZA) "},{idLinha":"114055","codigoLinha":"173-2","nomeLinha":"CAMAQUA "},{idLinha":"114107","codigoLinha":"165-2","nomeLinha":"COHAB "},{idLinha":"114016","codigoLinha":"180-2","nomeLinha":"DIARIO "},{idLinha":"114002","codigoLinha":"E110-2","nomeLinha":"EVENTUAL BARRA SHOPPING "},{idLinha":"113630","codigoLinha":"1112-2","nomeLinha":"Hu00cDPICA/TRISTEZA "},{idLinha":"114337","codigoLinha":"265-2","nomeLinha":"JARDIM MEDIANEIRA "},{idLinha":"113646","codigoLinha":"184-2","nomeLinha":"JUCA BATISTA "},{idLinha":"113647","codigoLinha":"1841-2","nomeLinha":"JUCA BATISTA VIA OSWALDO CRUZ "},{idLinha":"113891","codigoLinha":"E13-1","nomeLinha":"LEOPOLDINA/R. BERTA/EVENTO RADIO FARROUPILHA "},{idLinha":"114018","codigoLinha":"186-2","nomeLinha":"LIBERAL "},{idLinha":"113976","codigoLinha":"M79-2","nomeLinha":"MADRUGADAO SERRARIA/PONTA GROSSA "},{idLinha":"113737","codigoLinha":"2694-2","nomeLinha":"OPERACAO LAMI "},{idLinha":"114343","codigoLinha":"D63-2","nomeLinha":"ORFANOTROFIO/DIRETA "},{idLinha":"114156","codigoLinha":"2631-2","nomeLinha":"ORFANOTROFIO/JARDIM MEDIANEIRA "},{idLinha":"114094","codigoLinha":"187-2","nomeLinha":"PADRE REUS "},{idLinha":"113634","codigoLinha":"171-2","nomeLinha":"PONTA GROSSA "},{idLinha":"114158","codigoLinha":"264-2","nomeLinha":"PRADO "},{idLinha":"114310","codigoLinha":"R1-2","nomeLinha":"RAPIDA V JARDIM VILA NOVA "},{idLinha":"113907","codigoLinha":"R5-2","nomeLinha":"RAPIDA PONTA GROSSA "},{idLinha":"113895","codigoLinha":"R3-2","nomeLinha":"RAPIDA/NUCLEO ESPERANCA "},{idLinha":"114345","codigoLinha":"R81-2","nomeLinha":"Ru00c1PIDA CAMPO NOVO "},{idLinha":"113915","codigoLinha":"R84-2","nomeLinha":"Ru00c1PIDA JUCA BATISTA "},{idLinha":"113945","codigoLinha":"R101-2","nomeLinha":"Ru00c1PIDA RESTINGA NOVA/CAVALHADA/DOMINGOS "},{idLinha":"114195","codigoLinha":"2531-2","nomeLinha":"RENASCENu00c7A/DOMINGOS "},{idLinha":"113622","codigoLinha":"110-2","nomeLinha":"RESTINGA NOVA VIA TRISTEZA "},{idLinha":"113624","codigoLinha":"1101-2","nomeLinha":"RESTINGA NOVA VIA TRISTEZA/DOMINGOS E FERIADOS "},{idLinha":"113626","codigoLinha":"111-2","nomeLinha":"RESTINGA VELHA (TRISTEZA) "},{idLinha":"113644","codigoLinha":"179-2","nomeLinha":"SERRARIA "},{idLinha":"114112","codigoLinha":"T1-2","nomeLinha":"TRANSVERSAL 1 "}]}}]

```

Figura 4-5: Retorno da Chamada por Linhas de Ônibus.

- Obtenção de todas as linhas de ônibus disponíveis no sistema: esta função retorna como resultado todas as linhas presentes no sistema para consulta, seguindo a mesma estrutura de linhas citada anteriormente, identificação, código e nome.
- Obtenção de todas as linhas de lotação disponíveis no sistema: funciona de forma semelhante à consulta anterior, porém, destinada a resultados referentes a linhas de lotação.
- Obtenção de pontos de táxi conforme limites de coordenadas geográficas: semelhante à primeira consulta citada, esta função retorna uma lista de pontos de táxi. A estrutura da lista de pontos de táxi é como segue: endereço do ponto, telefone do ponto (quando disponível) e a coordenada geográfica do ponto.



Figura 4-6: Retorno da Chamada para Pontos de Táxi.

- Obtenção do itinerário de uma linha específica: essa consulta pode ser utilizada tanto para linhas referentes a ônibus quanto a linhas referentes a lotações. Toma como parâmetro a identificação da linha no sistema e retorna como resposta uma lista de pontos geográficos representando o itinerário da linha referenciada.



Figura 4-7: Retorno de Chamada para Itinerário.

Após a requisição em HTTP e o recebimento desses dados no formato JSON, o aplicativo utiliza um interpretador para transformar em objetos utilizáveis pelo programa os dados recebidos nas consultas. Para a implementação deste interpretador de dados no formato JSON foi utilizada a biblioteca voltada especificamente para essa finalidade “org.json”. Essa biblioteca já está disponível na plataforma Android, portanto não foi necessário fazer a sua importação.

No trecho de código a seguir é possível detectar a parte onde estão sendo tratados os dados em formato JSON provenientes da consulta referente às paradas de ônibus.

```

143
144- /**
145  * Parses a list of stops with lines from the JSONArray format to a list of Stop objects.
146  * @param stopsJSONArray
147  * @return a list of stops or null if none was found
148  */
149- public static List<Stop> getBusStopsWithLines(JSONArray stopsJSONArray) {
150     try {
151         List<Stop> stops = new ArrayList<Stop>();
152         if (stopsJSONArray != null) {
153             for (int i = 0; i < stopsJSONArray.length(); i++) {
154                 JSONObject JSONStop = stopsJSONArray.getJSONObject(i);
155                 String code = JSONStop.getString(TAG_COD);
156                 JSONArray linesJSONArray = JSONStop.getJSONArray(TAG_LINES);
157                 List<Line> lines = GetLines(linesJSONArray);
158                 GeoPoint geoPoint = CoordinateCalculation.CoordinatesToGeoPoint(
159                     JSONStop.getDouble(TAG_LAT), JSONStop.getDouble(TAG_LON));
160                 String isTerminal = JSONStop.getString(TAG_TERMINAL);
161                 Drawable icon = MainActivity.context.getResources().getDrawable(
162                     R.drawable.bus_stop_full_marker);
163                 Type type = Type.Bus;
164                 Stop stop = new Stop(code, geoPoint, isTerminal, icon, type, lines);
165                 stops.add(stop);
166             }
167             return stops;
168         }
169         else {
170             return null;
171         }
172     } catch (Exception e) {
173         e.printStackTrace();
174         return null;
175     }
176 }
177

```

Figura 4-8: Trecho do código referente ao Parser JSON.

Como pode ser visto na linha 149 (figura 4-8) esse método recebe como parâmetro de entrada um JSONArray, que como o nome indica é um array de objetos do tipo JSON. É interessante destacar que para o armazenamento e posterior visualização da parada em um mapa, os dados referentes à posição geográfica da parada (latitude e longitude) fornecidos pelo serviço são transformados em um tipo de dado mais adequado para essas operações chamado GeoPoint. Isso é feito pelo método de uma classe auxiliar responsável pelo tratamento das coordenadas do sistema chamada CoordinateCalculation, como pode ser visto na linha 158 (figura 4-8). O array de objetos JSON mencionado anteriormente vem do método definido como pode ser visto no trecho de código a seguir:

```

50
51  /**
52   * Gets Bus stops by the limits provided on coordinates
53   * @param coordinates
54   * @return JSON array with the stops or null if none was found
55   */
56  public static JSONArray getBusStopsFromCoordinate(String coordinates) {
57      try {
58          String action = "?a=tp&p=" + coordinates;
59          HttpClient httpClient = new DefaultHttpClient();
60          HttpPost httpPost = new HttpPost(URL + action);
61          HttpResponse response = httpClient.execute(httpPost);
62          HttpEntity entity = response.getEntity();
63          return new JSONArray(EntityUtils.toString(entity));
64      } catch (Exception e) {
65          e.printStackTrace();
66          return null;
67      }
68  }
69

```

Figura 4-9: HTTP Requests for Data

4.5 Obtenção dos dados para consulta de linhas e paradas (modo offline)

Para a utilização dos dados referentes a linhas e paradas sem que haja a necessidade de uma conexão com a internet, foi criado um banco de dados local em SQLITE, que é o sistema de estrutura de dados suportado nativamente pela plataforma Android. Esse banco de dados foi estruturado de uma forma que seria possível, além de prover todos os serviços já disponibilizados pelo servidor da Procempa, acrescentar mais funcionalidades à aplicação.

A definição da estrutura do banco de dados foi feita a partir da utilização das bibliotecas nativas de SQLITE disponíveis no Android. Como pode ser visto na figura 4-10.

```

70-  /**
71   * TABLES CREATION
72   */
73-  private static final String LINE_CREATE = "create table " + LINE_TABLE + " ("
74      + LINE_ID + " integer primary key autoincrement, "
75      + LINE_SYSTEM_ID + " integer, "
76      + LINE_CODE + " text, "
77      + LINE_NAME + " text, "
78      + LINE_ITINERARY + " text, "
79      + LINE_TYPE + " text, "
80      + LINE_COMPANY + " text, "
81      + LINE_DIRECTION + " text);";
82
83-  private static final String STOP_CREATE = "create table " + STOP_TABLE + " ("
84      + STOP_ID + " integer primary key autoincrement, "
85      + STOP_CODE + " text, "
86      + STOP_NAME + " text, "
87      + STOP_LATITUDE + " integer, "
88      + STOP_LONGITUDE + " integer, "
89      + STOP_TYPE + " text, "
90      + STOP_IS_TERMINAL + " integer, "
91      + STOP_PHONE_NUMBER + " text, "
92      + STOP_ADDRESS + " text);";
--

```

Figura 4-10: Creating SQLite Database Structure.

As consultas a esse banco de dados são realizadas da seguinte forma:

```

192
193  // QUERIES
194
195-  /**
196   * Tries to find a line where the line's name contains the name passed as parameter
197   *
198   * @param lineName
199   * @return
200   */
201-  public List<Line> getLineByName(String lineName) {
202      open();
203      Cursor cursor = database.query(
204          // table
205          DataBaseHelper.LINE_TABLE,
206          // columns
207          new String[] { DataBaseHelper.LINE_CODE, DataBaseHelper.LINE_SYSTEM_ID, DataBaseHelper.LINE_NAME, DataBaseHelper.LINE_TYPE },
208          // where
209          DataBaseHelper.LINE_NAME + " LIKE ?",
210          // arguments to where
211          new String[] { "%" + lineName.trim().toLowerCase() + "%" },
212          // groupBy, having, order
213          null, null, DataBaseHelper.LINE_NAME);
214      List<Line> lines = new ArrayList<Line>();
215      if (cursor.moveToFirst()) {
216          do {
217              // creates line
218              Line line = new Line(cursor.getString(0), cursor.getInt(1), cursor.getString(2), Line.Type.valueOf(cursor.getString(3)));
219              lines.add(line);
220          } while (cursor.moveToNext());
221      }
222      if (cursor != null && !cursor.isClosed()) {
223          cursor.close();
224      }
225      close();
226      return lines;
227  }

```

Figura 4-11: Exemplo de Consulta no Banco de Dados Local.

4.6 Visualização dos resultados das consultas no mapa

A visualização de mapas na aplicação foi implementada utilizando-se a API do Google Maps para Android. Essa API foi escolhida, pois, além de ser desenvolvida em conjunto com a plataforma e apresentar suporte nativo, apresenta fácil implementação

das funcionalidades relacionadas com a aplicação de camadas sobre o mapa e o cadastro de eventos de interação.

Na figura 4-12 temos um trecho de código referente à criação de uma camada de overlay no mapa, relacionado à visualização de uma posição ou endereço desejado.

```

105
106
107  /**
108   * Receiving a geopoint as parameter, add this geopoint to the overlay items;|
109   * @param geopoint
110   */
111  public Overlay(GeoPoint geopoint){
112      super(null);
113      Drawable icon = MainActivity.context.getResources().getDrawable(R.drawable.map_pin);
114      Stop location = new Stop(geopoint, icon, Stop.Type.Location, "location", "location");
115      items.add(location);
116      populate();
117      MapController mapController = MainActivity.mainMap.getController();
118      mapController.animateTo(geopoint);
119  }

```

Figura 4-12: Criando Overlays no Mapa.

4.7 Funcionamento da aplicação

Nesta seção serão detalhadas e exemplificadas as quatro principais telas de interação e as três principais funcionalidades da aplicação.

4.7.1 Principais Telas de Interação

Nas telas principais é que se dá a interação do usuário com os objetos fundamentais da aplicação, as paradas e as linhas de transportes públicos. Teremos essas telas ao final da maioria das funcionalidades a serem demonstradas posteriormente.

4.7.1.1 Interação com Paradas de Ônibus

O aplicativo proporciona ao usuário duas maneiras de interagir e buscar mais informações sobre as paradas de ônibus. A primeira é pelo simples toque na tela do ícone referente à parada. Após esse toque, a parada será selecionada, a visualização será centralizada na parada e na parte inferior da tela será apresentada uma janela de informações.

Na janela de informações teremos as informações resumidas da parada, um botão com a possibilidade de obter mais informações sobre ela, um botão para fechar a janela

de informações e uma lista de botões na horizontal, na qual cada botão corresponde a uma linha que atende essa parada identificada pelo código da linha.

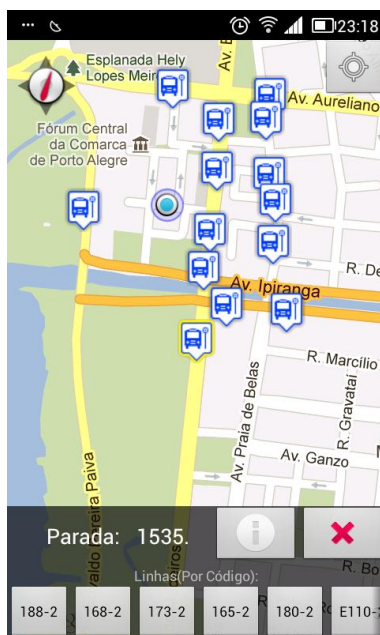


Figura 4-13: Visualizando informações da parada de ônibus.

Se o usuário desejar mais informações sobre a parada e clicar no botão “i”, à esquerda do botão fechar, ele será direcionado para a tela de informações sobre a parada.

Nessa tela será possível visualizar o endereço da parada e as linhas que atendem a ela, agora em uma lista vertical onde constam também os nomes, como pode ser visto na figura 4-14.



Figura 4-14: Tela de informações sobre a parada.

Quando o usuário clicar em um botão relacionado com uma linha de ônibus, tanto na janela de informações quanto na tela de informações da parada, ele será encaminhado para a tela a ser descrita a seguir.

4.7.1.2 Interação com Linha de Ônibus

Nessa tela serão mostradas algumas das informações mais relevantes com relação à linha de ônibus selecionada. Entre elas estão o código da linha, o nome da linha e o itinerário da linha desenhado no mapa. Como pode se ver na figura 4-15.

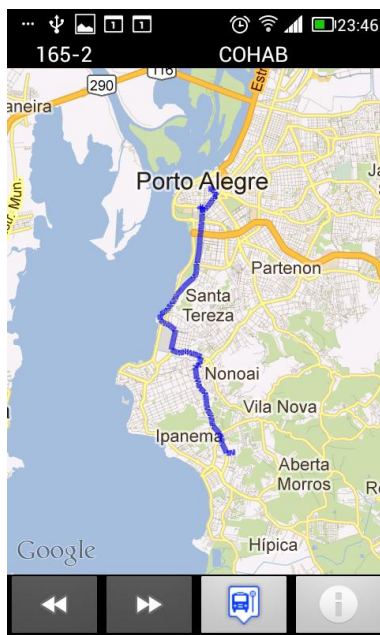


Figura 4-15: Tela de visualização de linha de ônibus.

É possível observar que na base da tela temos uma série de ícones, os quais estão relacionados a outras opções de interação com as informações sobre linhas de ônibus.

Os três primeiros ícones são referentes à visualização e à navegação entre as paradas desse ônibus. Um clique no terceiro ícone, da esquerda para a direita, que utiliza o mesmo ícone das paradas de ônibus, carrega as paradas referentes à linha que está sendo visualizada e as mostra no mesmo mapa. Essa ação também torna disponíveis os outros dois primeiros ícones, que ao serem clicados, permitem a navegação entre as paradas da linha.

Por fim, o último ícone nos leva à página de mais informações sobre a linha, onde constarão o código da linha, o nome da linha, o nome do consórcio ao qual a linha está ligada, a tabela de horários, a disponibilidade de veículos especiais e os tempos de atraso mínimo, médio e máximo.

4.7.1.3 Interação com Pontos de Táxi

Esta tela de interação é semelhante à tela referente às paradas de ônibus, porém aqui não temos uma tela especial para mais informações, isso ocorre devido ao fato de as informações sobre os pontos de táxi não serem tão extensas quanto às relacionadas às paradas de ônibus.

Uma vez que um ponto de táxi seja selecionado no mapa, uma janela de informações aparece na base da tela. A partir dessa janela podemos visualizar o endereço onde se localiza o ponto de táxi, o número telefônico desse ponto de táxi (quando disponível) e o botão para cancelamento da janela.

É interessante destacar que o usuário pode ser encaminhado para a tela de efetuar chamadas do seu celular com o número do ponto de táxi pré-carregado a partir de um clique no número de telefone.

4.7.1.4 Interação com Linhas de Lotação

A tela de interação com as linhas de lotação é muito semelhante com a tela de linhas de ônibus. As únicas diferenças entre elas são a ausência de um botão com mais informações sobre a linha e, como as lotações não possuem tantas paradas quanto os ônibus, os botões de navegação entre as paradas são simplesmente trocados por botões simbolizando os dois terminais, o de origem e o de destino.

4.7.2 Principais Funcionalidades

A aplicação possui três principais funcionalidades: Busca, Próximos a Mim e Trajeto.

O acesso a estas funcionalidades está disponível na parte inferior da tela principal, onde também se encontra no canto direito o botão que leva à tela de configurações do aplicativo, como pode ser visualizado na figura 4-16.

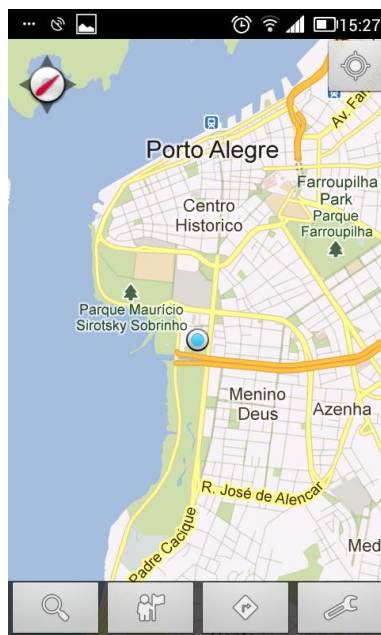


Figura 4-16: Tela principal mostrando localização atual.

4.7.2.1 Busca

A aplicação implementada possui uma ferramenta de busca integrada que possibilita a consulta de dados a partir da pesquisa por palavras. Essa ferramenta de busca foi desenvolvida de modo que o usuário não necessite especificar o contexto da informação que está procurando.

No exemplo mostrado na figura 4-17 o usuário realiza uma busca com a entrada “Prado”. A partir dessa entrada o sistema fará uma verificação visando determinar qual o contexto da entrada. Esse pode ser: uma linha de ônibus, uma linha de lotação, uma rua, um bairro, um ponto de referência da cidade, entre outros.

Aí então são apresentados os resultados da busca, separadas por tipos de resultado. Se não forem detectados múltiplos contextos na fase de reconhecimento da busca o usuário será levado diretamente ao seu resultado. Essas interações podem ser vistas nas sequências de telas a seguir.

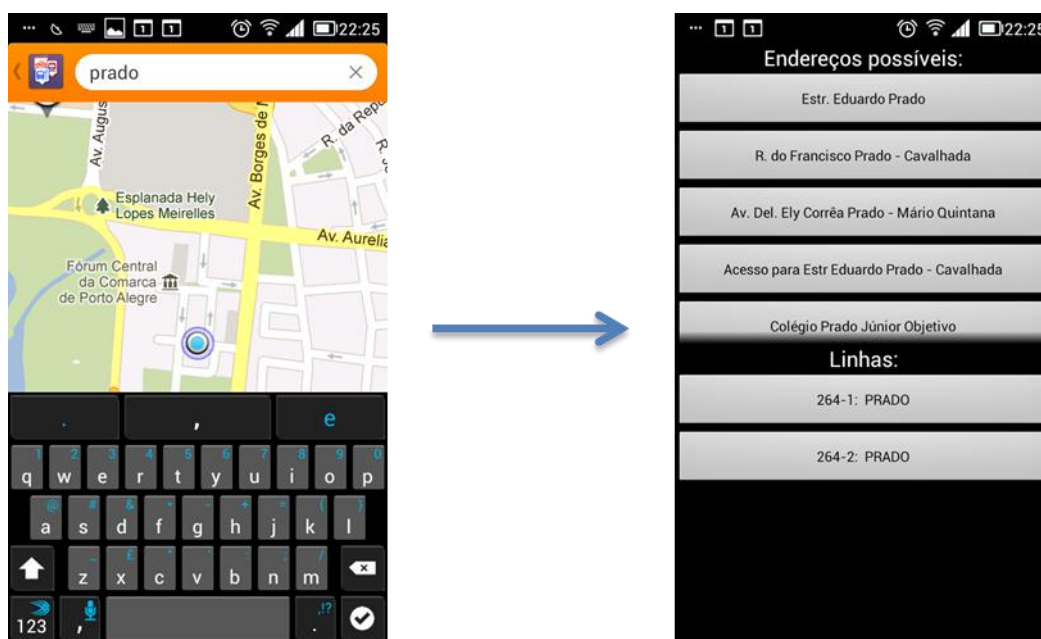


Figura 4-17: Busca com vários resultados.

Na segunda tela é possível visualizar dois tipos de resultados diferentes. A metade superior da tela fica reservada para resultados relacionados a endereços ou locais e a metade inferior para resultados relacionados a linhas de ônibus ou lotação, distintas pela cor do botão (natural para ônibus e vermelha para lotação).

Ao clicar em uma linha, o usuário será direcionado à tela de informação da linha em questão, já descrita anteriormente.

Ao clicar nos endereços, o usuário é levado novamente à tela principal do programa, porém agora a visualização do mapa estará centralizada no ponto relacionado com o item clicado e o mapa irá conter informações sobre meios de transporte públicos próximos a esse ponto, como na figura 4-18.

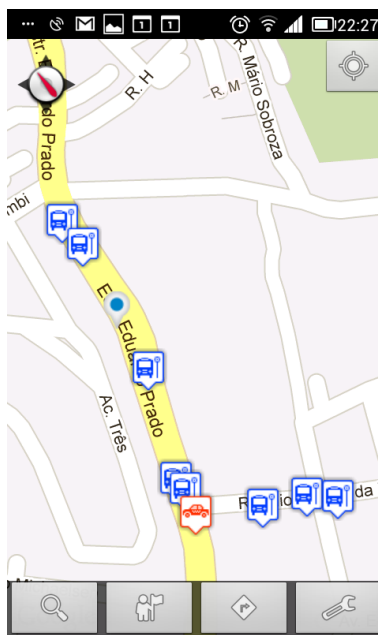


Figura 4-18: Visualização do resultado.

Caso o resultado da busca fosse único, o usuário seria diretamente encaminhado para a tela de resposta.

4.7.2.2 *Próximos a mim*

Essa funcionalidade do aplicativo visa disponibilizar as informações sobre os meios de transporte público com base na localização atual do usuário. Para isso é necessário que o sistema tenha conhecimento dessa posição. Portanto, uma vez que o sistema tenha iniciado e os serviços de localização estejam rodando, dentro de alguns segundos o sistema retorna à posição do usuário.

Essa tarefa também pode ser iniciada manualmente pelo clique no botão localizado no canto superior direito da tela, visto que a aplicação para de monitorar a posição atual do usuário depois de um tempo determinado para economizar bateria.

No caso do sistema não conseguir definir a posição, será mostrada uma mensagem pedindo a ativação dos serviços de localização ou um maior tempo de espera para o usuário para que o sistema seja capaz de detectar essa posição. Mesmo que a precisão dessa posição não seja a melhor, figura 4-19, após a visualização do ponto no mapa já é possível fazer consultas com base na localização.

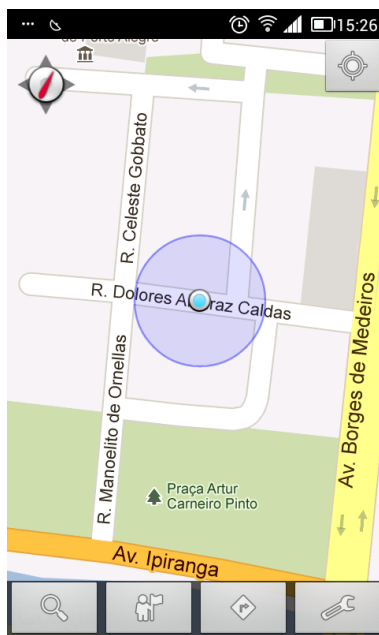


Figura 4-19: Exemplo de localização imprecisa.

Ao pressionar o segundo botão da esquerda para a direita na parte inferior da tela, o usuário terá ao seu dispor três formas de consulta: Paradas de Ônibus, Pontos de Táxi e Linhas de Lotação.

As duas primeiras opções retornarão telas semelhantes, tendo o tipo do meio de transporte como a única característica distinta (figura 4-20).

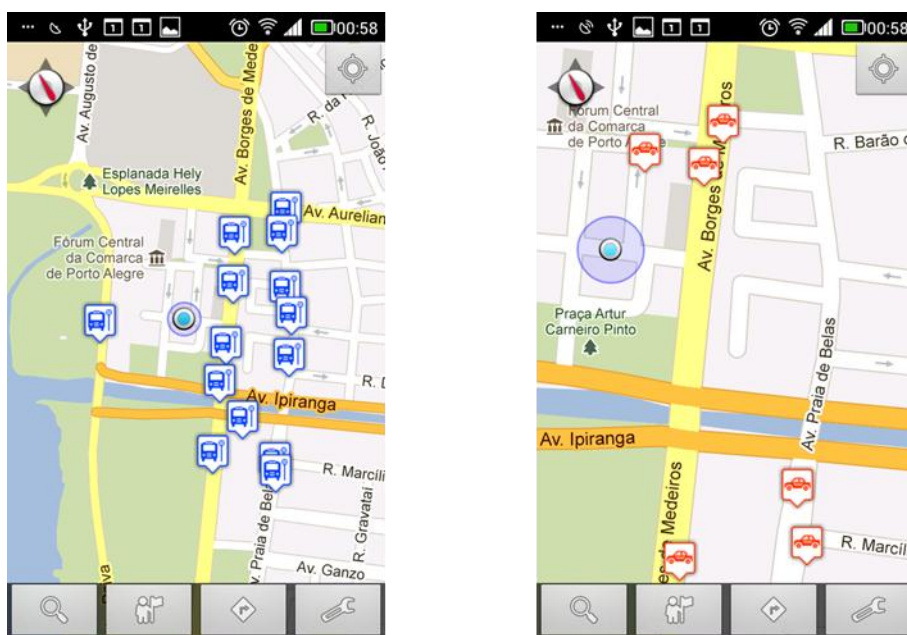


Figura 4-20: Próximos a Mim (Ônibus e Táxi, respectivamente).

Como as linhas de lotação só possuem paradas de terminais, a busca nesse caso é realizada a partir do itinerário da linha. Linhas que passam perto da localidade serão adicionadas como resultados. A partir desses resultados o usuário será encaminhado a uma tela com as linhas que passam perto da sua localização listadas. Ao clicar em uma das linhas, ele será encaminhado para a tela de informações da linha clicada.

4.7.2.3 Trajeto

Essa funcionalidade da aplicação provê ao usuário a possibilidade de fazer uma pesquisa por meios de transporte que atendam a dois pontos simultâneos, origem e destino.

Pelo fornecimento de dois endereços e algumas outras especificações sobre a maneira como o usuário deseja que a busca seja realizada, o sistema cruza suas informações para encontrar um trajeto que ligue esses dois pontos e, por fim, sugere uma lista de opções de linhas que atendem esses pontos e que o usuário pode utilizar para se locomover. Segue abaixo, um exemplo da utilização dessa funcionalidade.

Primeiramente o usuário preenche os campos de origem e destino, com destaque para o botão de localização que fornece o endereço da localização atual do usuário (se disponível). Depois é necessário fazer a escolha dos meios de transporte que o usuário deseja utilizar no cálculo do trajeto, ônibus ou lotação. Por fim, ele escolhe dentre uma lista de valores previamente estabelecidos pelo sistema quais serão as distâncias máximas de locomoção aceitas para o cálculo, distância do ponto de origem e distância do ponto de destino (figura 4-21, primeira tela).

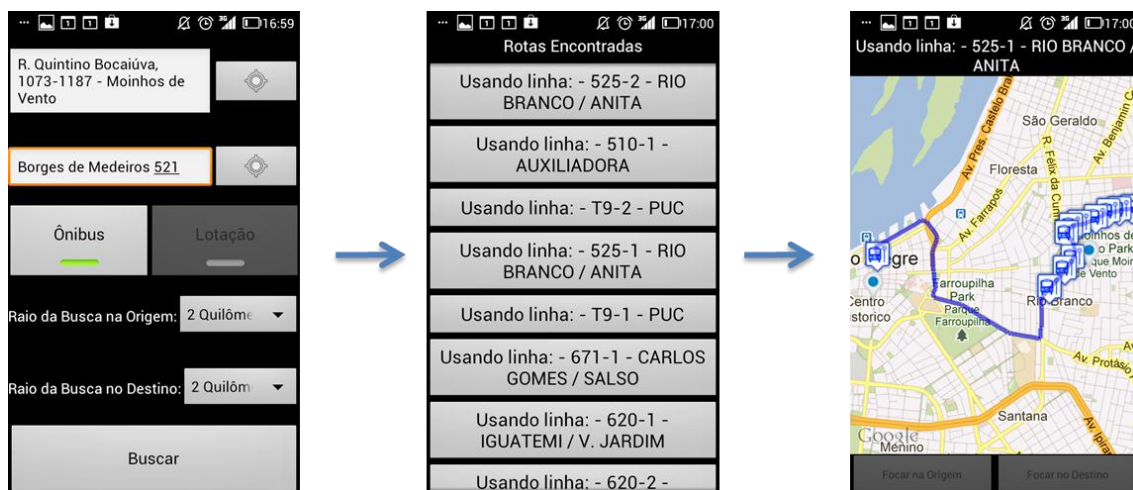


Figura 4-21: Exemplo usando a função Trajeto.

A partir desses dados fornecidos o sistema calcula os trajetos que atendem as especificações e lista os resultados. (figura 4-21, segunda tela). Clicando em uma das rotas encontradas é possível visualizar as paradas que poderiam ser utilizadas e o trajeto a ser percorrido (figura 4-21, terceira tela).

4.8 Validação da Implementação

4.8.1 Análise dos Requisitos

Para a validação da implementação foram feitos testes no sistema desenvolvido, de acordo com os requisitos estabelecidos no capítulo do Modelo da Aplicação. Estes testes foram realizados tanto em dispositivos reais quanto em máquinas virtuais disponibilizadas pelo SDK do Android. Os requisitos foram confrontados com as funcionalidades disponibilizadas pelo aplicativo ao final do seu desenvolvimento, assim verificando seus atendimentos ou não. Entre os dispositivos utilizados para teste foram:

- Samsung Galaxy S2 rodando Android versão IceCreamSandwich (4.0.3)
- Samsung Galaxy S rodando Android versão GingerBread (2.3.3)
- Motorola Defy rodando Android versão Froyo (2.2.2)
- Máquina Virtual rodando Android versão IceCreamSandwich (4.0.3)
- Máquina Virtual rodando Android versão GingerBread (2.3.3)

- Máquina Virtual rodando Android versão Froyo (2.2.2)

Resultados:

- 100% dos requisitos funcionais de caráter obrigatório foram atendidos.
- 80% dos requisitos funcionais de caráter importante foram atendidos.
- 20 % dos requisitos funcionais de caráter desejável foram atendidos.
- 100% dos requisitos não funcionais foram atendidos.

Com base no volume de requisitos que foram efetuados com sucesso é possível concluir que a aplicação, apesar de não possuir todas as funcionalidades previstas na sua modelagem, cumpre as tarefas básicas relacionadas com a solução do problema em questão e, portanto, pode ser considerada como uma solução viável para o problema.

4.9 Comparação com Trabalhos Semelhantes

Pela visualização de soluções semelhantes ao sistema desenvolvido como objeto desse trabalho, foi realizada uma comparação quanto a alguns aspectos relevantes quando se trata de uma aplicação mobile voltada a pesquisa de transportes públicos. A seguir teremos alguns casos dessas soluções, acompanhados da comparação com o trabalho realizado.

4.9.1.1 POABUS (JURKOVSKI 2012)

Sistema web para consulta de linhas de ônibus da cidade de Porto Alegre. Este sistema possibilita realizar a busca de informações sobre linhas de ônibus por endereços e por marcadores no mapa. Seus destaques são a utilização de um banco de dados colaborativo, onde os usuários podem cadastrar linhas e paradas e a possibilidade de se realizar a busca por pontos de origem e destino.

4.9.1.2 POATRANSPORTE (PROCEMPA 2012)

Outro sistema web para consulta de linhas com interface bastante semelhante ao anterior, porém este possui mais tipos informações. Foi desenvolvido com incentivo da prefeitura municipal de Porto Alegre em parceria com a Procempa e a EPTC. Este sistema disponibiliza dados sobre ônibus, lotação e táxi. Seus serviços de requisição de

dados são abertos pra uso público, permitindo que qualquer pessoa possa criar uma aplicação com base neles.

4.9.1.3 PortoBus - Porto Alegre (SCHMIDT 2012)

Sistema mobile para a plataforma Android que trata sobre pesquisas de linhas de ônibus. Tem várias funcionalidades interessantes como, visualização de paradas no mapa, visualização de tabela de horários e utilização do posicionamento do aparelho para consulta de informações.

4.9.1.4 Comparação com o Sistema Desenvolvido

A aplicação POAMóvel teve como inspiração algumas das ideias citadas anteriormente, porém, seus principais aspectos não estão relacionados a aplicações já existentes. No âmbito dos aplicativos móveis disponíveis para a cidade de Porto Alegre, é a primeira a apresentar uma gama de novas funcionalidades: consultas sem utilizar conexão com a internet, pesquisa por linhas de lotação, pesquisa por pontos de origem e destino, customização do raio das buscas, comparação entre rotas sugeridas, sistema de busca de informação aprimorado, entre outras.

5 CONCLUSÃO

5.1 Análise Geral do Trabalho Desenvolvido

Como objetivo desse trabalho foi traçado o desenvolvimento de uma aplicação móvel que resolvesse o seguinte problema: disponibilizar para um usuário, portador de um dispositivo móvel, informações sobre diversos tipos de meios de transporte público baseados na sua localização ou em uma localização fornecida, de maneira prática, usual, personalizada e útil.

Ao longo da realização do trabalho, a partir do estudo realizado e das ferramentas e tecnologias disponíveis para o uso, definiu-se um escopo para a aplicação do modelo estabelecido. Esse escopo permitiu que o modelo de aplicação proposto fosse testado e avaliado quanto a sua capacidade de resolver o problema definido como objetivo do trabalho.

As escolhas referentes a quais tecnologias deveria se utilizar foram vitais no desenvolvimento deste trabalho, com destaque para a definição da utilização da plataforma mobile Android, da utilização das APIs do Google Maps para mapas, da utilização da IDE Eclipse e do aproveitamento dos serviços de fornecimento de dados da Procempa.

Escolhas de como modelar o sistema também foram fundamentais, com destaque para a definição dos modos online e off-line da aplicação - ideia que, se concebida após o início do desenvolvimento, certamente traria empecilhos por causa do tempo necessário para adaptação e retrabalho em cima de módulos que já tivessem sido desenvolvidos sem essa mentalidade.

O resultado da implementação do modelo foi a aplicação POAMóvel. A partir da análise do cumprimento dos requisitos, realizada no capítulo Implementação da Aplicação, foi possível observar que o desempenho do sistema foi satisfatório. Alguns requisitos importantes e desejados acabaram não sendo desenvolvidos, pois não houve tempo hábil de serem analisados, planejados e executados da mesma maneira e com o mesmo empenho e dedicação que os outros requisitos foram. Entretanto, estes foram considerados como boas ideias para melhorar a solução. Essa aplicação está muito próxima de poder ser distribuída para todos os usuários da plataforma Android, acrescentando um leque de novas possibilidades de locomoção dentro da cidade de Porto Alegre.

Acredito que o objetivo do trabalho foi alcançado e que, através de esforço e dedicação, foi possível desenvolver um modelo de aplicação que resolve o problema proposto e que por sua generalização não está preso a nenhuma plataforma mobile específica nem a alguma localidade, permitindo sua implementação de outras formas além da que foi utilizada nesse trabalho.

5.2 Possibilidades Futuras

Como possibilidades futuras para a continuação deste trabalho ficam todos os requisitos não atingidos e funcionalidades pensadas, porém não desenvolvidas, com destaque para alguns que estão listados a seguir.

- Cadastro de pontos de interesse: o acréscimo dessa funcionalidade tornaria a aplicação muito mais pessoal e útil, permitindo a utilização um sistema de favoritos para carregar mais facilmente os pontos de interesse cadastrados pelo usuário;
- Utilização de mapas off-line: essa funcionalidade agregaria considerável valor à aplicação. Uma experiência completamente off-line tornaria a utilização do aplicativo praticamente independente de condições de funcionamento, aumentando drasticamente sua disponibilidade e utilidade;
- Sugestão de passeios turísticos: com a proximidade de grandes eventos esportivos no Brasil, como a Copa do Mundo FIFA de 2014 e as Olimpíadas de 2016, os quais deverão atrair muitos turistas, seria interessante permitir o cadastro de rotas

turísticas pré-definidas. Somando essa capacidade à de mostrar o custo dessa rota e um modo de guia para o trajeto, teríamos uma ótima aplicação para turistas que desejam se aventurar pela cidade e não gastar muito no caminho;

- Combinação de tipos de meios de transporte para o cálculo das rotas: um cálculo misto de meios de transporte para as sugestões de trajetos utilizadas no programa seria um importante adendo às funcionalidades já implementadas, pois, além de aumentar consideravelmente a possibilidade de se encontrar uma rota pesquisada, as opções de rotas também seriam mais numerosas e mais flexíveis;

- Implementação em outras cidades e em outras plataformas: como os serviços e tecnologias específicos à cidade de Porto Alegre e à plataforma Android só foram utilizados na parte de implementação do trabalho, toda sua modelagem pode ser portada para outras plataformas de desenvolvimento móvel e para outras cidades. No primeiro caso, basta que as plataformas em questão sejam capazes de implementar os conceitos básicos do trabalho, como localização, utilização de mapas, conexão com a internet e armazenamento de dados para que a aplicação possa ser desenvolvida nos mesmos moldes. No segundo caso, seria necessário que a cidade ou local em questão tivesse ao seu dispor as informações sobre os seus serviços de transporte público.

6 REFERÊNCIAS

6.1 Livros

BEZERRA, Eduardo. Princípios de Análise e Projeto de Sistemas com UML. 2ª Edição. Rio de Janeiro: Campus, 2007.

BURNETTE, Ed. Hello, Android. Dallas: Pragmatic Programmers, 2010.

FLING, Brian. Mobile Design and Development. Sebastopol: O'Reilly, 2009.

GARGENTA, Marko. Learning Android. Sebastopol: O'Reilly, 2011.

HASHIMI, Sayed Y., Satya KOMATINENI, e Dave MACLEAN. Pro Android 2. New York: Apress, 2010.

HULL, Elizabeth, KEN Jackson, e JEREMY Dick. Requirements Engineering. London: Springer, 2005.

KOTONYA, Gerald, e Ian SOMMERVILLE. Requirements engineering: processes and techniques. Wiley, 1998.

KÜPPER, Alex. Location-Based Services - Fundamentals and Operation. Chichester: John Wiley & Sons, Ltd, 2005.

MCCONNELL, Steve. Rapid Development - Taming Wild Software Schedules. Redmond: Microsoft Press, 1996.

PETZOLD, Charles. Programming Windows Phone 7. Washington: Microsoft Corporation, 2010.

SCHILLER, Jochen, e Agnès VOISARD. Location Based Services. San Francisco: Elsevier, 2004.

6.2 Artigos

POTTEL, Mike. "MVP: Model-View-Presenter The Taligent Programming Model for C++ and Java." 1996: 7-8.

6.3 Sites

CONJECTURE CORPORATION. What is Symbian OS. Disponível em: <http://www.wisegeek.com/what-is-symbian-os.htm>. Acesso em: Maio de 2012.

GOOGLE. Android Developers. Disponível em: <https://developer.android.com/about/dashboards/index.html>. Acesso em: Maio de 2012.

JURKOVSKI, Bruno. POABUS. Disponível em: <http://www.poabus.com.br/> Acesso em: Junho de 2012.

ORACLE. About Java for Mobile Devices. Disponível em: <http://www.oracle.com/technetwork/java/javame/javamobile/overview/about/index.html>. Acesso em: Maio de 2012.

PROCEMPA. PoaTransporte. Disponível em: <http://www.poatransporte.com.br/>. Acesso em Junho de 2012.

SCHMIDT, Emerson. Porto Bus. Disponível em: <http://www.portobus.com.br/>. Acesso em Junho de 2012.

SCIENTIFIC AMERICAN. Disponível em: <http://www.scientificamerican.com/article.cfm?id=google-t-mobile-launch-g1>. Acesso em Abril de 2012.

ULANOFF, Lance. Mashable Tech. Disponível em: <http://mashable.com/2012/03/29/microsoft-metro-is-a-philosophy/>. Acesso em Abril de 2012.

UOL. Olhar Digital. Disponível em: <http://olhardigital.uol.com.br/produtos/mobilidade/noticias/android-cresceu-400-no-brasil-em-2011>. Acesso em Junho de 2012.

Vision Mobile. [Infographic] The Mobile Platform Race – How do mobile platforms stack up? Disponível em: <http://www.visionmobile.com/blog/2011/07/infographic-developer-economics-2011/>. Acesso em Junho de 2012.