

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
INSTITUTO DE INFORMÁTICA
CURSO DE ENGENHARIA DE COMPUTAÇÃO

LUCIANO GOULARTE SIQUEIRA

**Aplicação Bus Tracker – Oferecendo uma melhor
experiência aos usuários do transporte urbano, a partir da
utilização de informações de rastreamento veicular**

Trabalho de Graduação.

Prof. Dr. Cláudio Fernando Resin Geyer
Orientador

Porto Alegre, junho 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 ECP: Prof. Sérgio Cechin

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

AGRADECIMENTOS

Agradeço aos meus pais José e Nora, por tudo que me ensinaram e por todo o apoio que me deram para enfrentar este grande desafio. A minha amada esposa Joice pelo amor, dedicação, paciência e compreensão nas horas mais difíceis. A minha querida filha Piaf, pela companhia inseparável durante as longas horas em que me dediquei a este trabalho. A toda minha família, por todo o apoio.

Agradeço ao professor Cláudio F. R. Geyer, pela inspiração e pela orientação durante o desenvolvimento deste trabalho.

Agradeço aos meus queridos amigos Alexandre Azevedo e Georgina Reátegui por todas as contribuições que deram, em especial as agradáveis horas de almoço.

Agradeço a todos que direta ou indiretamente contribuíram para esta realização.

SUMÁRIO

LISTA DE ABREVIATURAS E SIGLAS	6
LISTA DE FIGURAS	8
LISTA DE TABELAS	9
RESUMO	10
ABSTRACT	11
1 INTRODUÇÃO	12
1.1 MOTIVAÇÃO	12
1.2 OBJETIVO.....	12
1.3 ORGANIZAÇÃO DO TEXTO.....	13
2 SERVIÇOS COM BASE EM LOCALIZAÇÃO E APLICAÇÕES MÓVEIS	14
2.1 SERVIÇOS COM BASE EM LOCALIZAÇÃO	14
2.2 LBS E SERVIÇOS CONSCIENTES DO CONTEXTO.....	14
2.3 FUNDAMENTOS DE LOCALIZAÇÃO	15
2.3.1 <i>Localização espacial</i>	16
2.3.2 <i>Projeções em mapas</i>	16
2.3.3 <i>Bases de dados espaciais e Sistemas de Informação Geográfica (GIS)</i>	16
2.4 POSICIONAMENTO.....	16
2.5 TECNOLOGIAS UTILIZADAS NO DESENVOLVIMENTO DE APLICAÇÕES MÓVEIS	17
2.5.1 <i>Plataforma Android</i>	17
2.5.1.1 JVM.....	17
2.5.1.2 Arquitetura.....	17
2.5.1.3 Fundamentos das aplicações	19
2.5.1.4 Localização e Mapas	19
2.5.2 <i>Plataforma iOS</i>	19
2.5.2.1 Plataforma.....	20
2.5.2.2 Localização e Mapas	21
2.5.3 <i>Servidor</i>	22
3 RASTREAMENTO VEICULAR APLICADO AO TRANSPORTE URBANO	23
3.1 HELSINQUE – FINLÂNDIA	24
3.2 AUCKLAND – NOVA ZELÂNDIA	25
3.3 DENVER, COLORADO - USA	26
3.4 SYDNEY – AUSTRÁLIA	26
3.5 FORTALEZA – BRASIL.....	27
3.6 SÃO PAULO – BRASIL	27
3.7 UBERLÂNDIA – BRASIL.....	27
3.8 PORTO ALEGRE – BRASIL	28
3.9 CONSIDERAÇÕES FINAIS	29
4 PROJETO	31

4.1	ENGENHARIA DE SOFTWARE.....	31
4.2	PROCESSO UNIFICADO.....	31
4.3	ANÁLISE DE REQUISITOS.....	32
4.4	CONCEPÇÃO – 1ª FASE DO PU	33
4.4.1	<i>Visão</i>	33
4.4.1.1	Módulo Cliente	33
4.4.1.2	Módulo Servidor.....	34
4.4.1.3	A Base de Dados	34
4.4.2	<i>Modelo de Casos de Uso</i>	34
4.5	ELABORAÇÃO – 2ª FASE DO PU	38
4.5.1	<i>Modelo de Domínio</i>	38
4.5.2	<i>Diagrama de Sequência do Sistema</i>	39
4.5.3	<i>Contrato de Operação</i>	42
4.5.4	<i>Projeto do Banco de Dados</i>	43
4.5.5	<i>Modelo Conceitual do Banco de Dados</i>	44
4.5.6	<i>Modelo Lógico do Banco de Dados</i>	45
4.5.6.1	Escolhendo um Sistema de Gerenciamento de Banco de Dados	45
4.5.6.2	Projeto Lógico.....	46
4.6	CONSIDERAÇÕES FINAIS	47
5	PROTOTIPAÇÃO	48
5.1	ESCOLHENDO UMA LINHA DE ÔNIBUS.....	49
5.2	ACOMPANHANDO VEÍCULOS EM TEMPO REAL	54
5.3	RECEBENDO ALERTAS DE APROXIMAÇÃO DE VEÍCULO	56
5.4	VALIDAÇÃO DA IMPLEMENTAÇÃO.....	58
6	CONCLUSÃO	64
6.1	ANÁLISE GERAL DO TRABALHO	64
6.2	POSSIBILIDADES FUTURAS	64
	ANEXO <MONOGRAFIA DO TRABALHO DE GRADUAÇÃO I>	69

LISTA DE ABREVIATURAS E SIGLAS

API	Application Programming Interface
ARTA	Auckland Regional Transport Authority
BSD	Berkeley Software Distribution
CAS	Context-aware Services
CCO	Centro de Controle e Operação
CITFOR	Controle Integrado de Transportes de Fortaleza
CO	Contrato de Operação
DSS	Diagrama de Sequência do Sistema
DVM	Dalvik Virtual Machine
EPTC	Empresa Pública de Transporte e Circulação
ER	Entidade-Relacionamento
LBS	Location Based Services
LS	Location Service
GIS	Geographic Information Systems
GPS	Global Positioning System
GPRS	General Package Radio Service
GSM	Global System for Mobile Communications
IDE	Integrated Development Environment
JVM	Java Virtual Machine
PDA	Personal Digital Assistant
POSIX	Portable Operating System Interface
PU	Processo Unificado
RFID	Radio Frequency Identification
RTD	Regional Transportation District
SaaS	Software as a Service
SDK	Software Development Kit
SGBD	Sistema de Gerenciamento de Banco de Dados

SIT	Sistema Integrado de Transporte
SOMA	Sistema de Ônibus Monitorado Automaticamente
UML	Unified Modeling Language
UTCC	Urban Traffic Control Center
TCRP	Transit Cooperative Research Program

LISTA DE FIGURAS

Figura 2.1: Principais componentes do Sistema Operacional Android	19
Figura 2.2: Camadas do Sistema Operacional iOS	21
Figura 3.1: Configuração no sistema no projeto HelMi de Helsinque	25
Figura 3.2: Módulo de monitoramento veicular Maxtrack	28
Figura 4.1: Modelo de Domínio	39
Figura 4.2: DSS para o caso de uso Criar Identificador de Usuário	40
Figura 4.3: DSS para o cenário principal do caso de uso Selecionar Linha	40
Figura 4.4: DSS para um cenário alternativo do caso de uso Selecionar Linha	40
Figura 4.5: DSS para um cenário alternativo do caso de uso Selecionar Linha	41
Figura 4.6: DSS para o cenário principal do caso de uso Rastrear Veículo	41
Figura 4.7: DSS para o cenário principal do caso de uso Cadastrar Evento Espacial ...	42
Figura 4.8: Diagrama Entidade-Relacionamento da aplicação Bus Tracker	45
Figura 4.9: Modelo de Dados Relacional da aplicação Bus Tracker	47
Figura 5.1: Módulo Cliente – Tela inicial	50
Figura 5.2: Módulo Cliente – Pontos de parada próximos ao usuário	52
Figura 5.3: Módulo Cliente – Lista de linhas de ônibus que utilizam o ponto de parada selecionado pelo usuário	53
Figura 5.4: Módulo Cliente – Itinerário da linha “5201-1 TRIÂNGULO/24 DE OUTUBOR/PUC”	54
Figura 5.5: Módulo Cliente – Rastreando veículos da linha (Posição 1)	55
Figura 5.6: Módulo Cliente – Rastreando veículos da linha (Posição 2)	56
Figura 5.6: Módulo Cliente – Rastreando veículos da linha (Posição 3)	57
Figura 5.7: Módulo Cliente – Alertando ao usuário a aproximação de um ônibus	57
Figura 5.8: Módulo Cliente – Sequência de telas obtidas durante avaliação com módulo rastreador	62

LISTA DE TABELAS

Tabela 4.1: Caso de Uso: Criar Identificador de Usuário	34
Tabela 4.2: Caso de Uso: Selecionar Linha	35
Tabela 4.3: Caso de Uso: Rastrear Veículo	36
Tabela 4.4: Caso de Uso: Cadastrar Evento Espacial	37
Tabela 4.5: Caso de Uso: Monitorar Evento	37
Tabela 4.6: Caso de Uso: Alertar Ocorrência de Evento	38
Tabela 4.7: Contrato de Operação: criarIdentificadorDeUsuario	42
Tabela 4.8: Contrato de Operação: rastrearLinha	42
Tabela 4.9: Contrato de Operação: cadastrarEvento	43

RESUMO

O constante avanço tecnológico dos dispositivos móveis (smartphones, tablets), em conjunto com a queda nos preços e a atual abrangência e disponibilidade de redes wireless, tornou-os amplamente populares, gerando um crescimento explosivo do mercado de aplicativos móveis, que continua a prover novas oportunidades a todo momento. Como consequência deste cenário os aplicativos móveis tornaram-se de fundamental importância e os desenvolvedores têm a oportunidade, e o desafio, de criar aplicativos que explorem e beneficiem-se das constantes novas funcionalidades dos dispositivos, tornando a informação disponível a qualquer momento, em qualquer lugar, facilitando tarefas rotineiras das pessoas, permitindo melhor uso de seu tempo pessoal. Dentre os diversos aplicativos oferecidos existem aqueles baseados em geolocalização, ou seja, na identificação global da posição de um objeto, o que pode ser utilizado das mais diversas formas para fornecer um melhor serviço e experiência aos usuários. O objetivo deste trabalho é projetar e desenvolver o protótipo de uma aplicação que permitirá aos usuários de transporte público (ônibus, trem, lotação) acompanharem a posição e o deslocamento dos veículos em tempo real, assim como configurar eventos pelos quais gostariam de ser notificados, como por exemplo uma mensagem que os avise que certo ônibus atingiu determinado ponto de ônibus.

Palavras-Chave: mobilidade, localização, rastreamento veicular, transporte urbano.

Bus Tracker App - providing a better experience to public transport users through vehicle tracking usage

ABSTRACT

The continuous technological advancement of mobile devices (smartphones, tablets), combined with its declining cost and broad coverage of wireless networks, have made them widely used, generating an explosive growth of the mobile application market, which continues to provide new opportunities each day. Due to that mobile applications became prominent and the developers face the possibility, and the challenge, to create applications to fully exploit mobiles' constant new capabilities, making information available at any time, in any place, and making easier people daily task routines, permitting a better use of personal time. Within various popular applications are those based on geolocation, or the identification of real world geographic location of an object, which can be used in many different ways to provide better services and experience to the users. The objective of this project is to design and develop a prototype for an application that will allow public transport users to follow vehicles' (buses, trains, vans) position and displacement in real time, as well as to set notification events, like a message to indicate that a bus line have reached a specific bus stop.

Keywords: mobile, geolocation, public transport, vehicle tracking.

1 INTRODUÇÃO

1.1 Motivação

Com o surgimento dos primeiros computadores pessoais surgiu também um novo mercado com infinitas possibilidades. Com o tempo os computadores evoluíram, se tornaram portáteis, foram integrados a outros dispositivos, como no caso dos telefones celulares, criando novas possibilidades de evolução para os sistemas de informação. Toda essa explosão tecnológica deu origem a novos mercados, novas possibilidades de serviços que até então não eram possíveis devido a não existência de tecnologias necessárias. Um destes novos mercados diz respeito aos dispositivos móveis (como telefones celulares, PADs, smartphones, e mais recentemente os tablets) e ao novo universo de possibilidades que esses dispositivos apresentam tanto para os desenvolvedores de aplicações quanto para os usuários.

À medida que as tecnologias relacionadas à mobilidade evoluem, também evoluem as aplicações relacionadas a ela. Aplicações conscientes do contexto passam a ser comuns (Dey e Abowd, 1999); o usuário passa a ser reconhecido como indivíduo; os serviços passam a ter consciência do contexto relacionado a este indivíduo e a fazer uso deste contexto para agregar valor ao que é ofertado ao usuário. Os serviços com base em localização (LBS – Location Based Services) fazem parte deste paradigma, onde o referido contexto é aqui representado pela localização (Kupper, 2005). Os LBSs tem a capacidade de conhecer a localização, ou do usuário ou de algum alvo específico, e a partir desta informação propiciar uma melhor experiência para o usuário.

1.2 Objetivo

É no contexto de dispositivos móveis e tecnologias de posicionamento que este trabalho fundamenta o projeto e a prototipagem de um LBS que ofereça, aos usuários do transporte público das grandes cidades, serviços que permitem o monitoramento em tempo real dos veículos de transporte e que façam uso destas informações de rastreamento para gerar avisos e mensagem de alerta ao usuário à medida que eventos pré configurados por eles (como no caso de determinado veículo aproximar-se do ponto de embarque do usuário) sejam identificados.

Os LBSs começaram a ganhar força com o popularização e barateamento das tecnologias de posicionamento, como o Sistema Global de Posicionamento (GPS - Global Positioning System) e dos serviços de comunicação prestados pelas redes de telefonia celular GSM, como é o caso do Serviço de Rádio de Pacote Geral (GPRS – General Package Radio Service) (3GPP).

Levando em consideração que o número de pessoas que tem algum tipo de dispositivo móvel com tais tecnologias é cada vez maior, e que há uma necessidade de melhoria constante por parte dos serviços de transporte público, pode-se imaginar que serviços propostos neste trabalho tragam um benefício considerável aos usuários do transporte.

Não é objetivo deste trabalho definir uma infraestrutura de rastreamento veicular para o transporte público. Ele apenas apresenta um panorama das tecnologias utilizadas para tal fim, e define um modelo de base de dados para armazenar as informações de posicionamento dos veículos geradas, em tempo real, por uma possível infraestrutura de monitoramento, assim como informações referentes a itinerários, pontos de parada e identificação dos veículos. Tendo como ponto de partida uma base de dados com as informações acima referidas, é então definida uma aplicação (aqui denominada de Bus Tracker), no modelo cliente-servidor, que permite ter acesso, via dispositivos móveis, às informações de posicionamento dos veículos em tempo real.

1.3 Organização do texto

Este trabalho está organizado em uma sequência lógica, partindo da motivação e do objetivo do trabalho, passando pelas tecnologias relacionadas a este, até apresentar a análise de requisitos, o projeto e a prototipação da aplicação Bus Tracker.

O capítulo 2, Serviços com base em localização e aplicações móveis, traz uma definição do que são Serviços Conscientes do Contexto e Serviços com Base em Localização, e qual a relação entre eles. Apresenta ainda, conceitos relacionados com as principais tecnologias que dão suporte aos LBSs, e as aplicações móveis.

O capítulo 3, Rastreamento veicular aplicado ao transporte urbano, traz algumas experiências, com rastreamento veicular, voltadas ao transporte urbano, além de demonstrar como a aplicação destas tecnologias melhora o controle de qualidade do transporte bem como a experiência dos usuários em relação ao serviço de transporte. Além disso, são apresentadas algumas empresas que vendem serviços ou até mesmo soluções completas para o rastreamento e monitoramento do transporte urbano.

O capítulo 4, Projeto, apresenta inicialmente uma breve definição do que é a Engenharia de Software. O restante do capítulo traz a análise de requisitos e o projeto da aplicação Bus Tracker, sendo que este processo segue as diretrizes do Processo Unificado - metodologia de Análise de Projeto Orientado a Objetos, a qual também é apresentada neste capítulo à medida que o projeto evolui.

O capítulo 5, Prototipação, inicia com um breve comentário sobre as tecnologias que serão utilizadas, apresentando, logo em seguida, a implementação de uma protótipo que visa validar as principais funcionalidades da aplicação Bus Tracker.

O Capítulo 6, Conclusão, apresenta, de uma perspectiva geral, os resultados deste primeiro trabalho, assim como algumas possibilidades futuras, além de alternativas para cidades onde o rastreamento do transporte urbano não é uma realidade.

2 SERVIÇOS COM BASE EM LOCALIZAÇÃO E APLICAÇÕES MÓVEIS

Este capítulo apresenta de maneira sucinta uma definição para Serviços com Base em Localização (contexto da aplicação Bus Tracker) e conceitos relacionados. Além disso, algumas das principais tecnologias relacionadas ao desenvolvimento de aplicações móveis, como as plataformas Android e iOS, também serão abordadas aqui, visando uma contextualização nas tecnologias que servirão de base para o protótipo a ser desenvolvido.

2.1 Serviços com base em localização

Esta seção aborda os serviços com base na localização e as principais tecnologias que auxiliam no desenvolvimento destes serviços. É apresentada uma definição para serviços conscientes do contexto, e como os serviços com base em localização (LBS – Location Based Services) pertencem ao conjunto destes serviços. Logo em seguida são apresentadas algumas das tecnologias que auxiliam no desenvolvimento de um LBS.

2.2 LBS e serviços conscientes do contexto

Serviços Conscientes do Contexto (CAS - Context-aware Services) são definidos como serviços que automaticamente adaptam seu comportamento de acordo com um ou mais parâmetros que refletem o contexto de um determinado alvo (Dey e Abowd, 1999). Estes parâmetros são chamados de informações de contexto, e podem pertencer a diferentes categorias de informações, como por exemplo, contexto pessoal, contexto espacial, contexto social, contexto físico, etc. Os LBSs são considerados um subconjunto dos serviços conscientes do contexto, onde o contexto é a localização espacial.

A forma como informações de contexto são utilizadas em um serviço com base em localização fica clara nesta definição apresentada em (Kupper, 2005): LBSs são serviços que fornecem informações que foram criadas, compiladas, selecionadas, ou filtradas levando em consideração a localização atual dos usuários, de outras pessoas ou de objetos móveis.

Um LBS não é responsável pela geração das informações de localização. Ele apenas faz uso destas informações, por ele obtidas junto a um subsistema chamado de Serviço de Localização (LS - Location Service). Este sim é responsável pela geração e distribuição das informações de localização. Essencialmente, os Serviços de Localização contribuem para o funcionamento dos LBSs, sendo aqueles um importante subsistema destes.

Os LBSs podem ser classificados como reativos ou proativos. Um LBS reativo é sempre ativado pelo usuário enquanto um LBS proativo responde automaticamente a algum evento de localização predefinido. Ao contrário de LBSs reativos onde, normalmente, o usuário precisa ter sua localização identificada apenas uma vez, os LBSs proativos precisam verificar constantemente a localização do usuário para detectar algum evento de localização.

A tecnologia chave dos LBSs é chamada de Posicionamento (Positioning). É ela que torna tão interessante os LBSs, pois permite obter a informações de posicionamento sem que os usuários necessitem entrar com estas informações manualmente. Existem inúmeras técnicas para implementação desta tecnologia, sendo que elas diferem na qualidade dos parâmetros de posicionamento resultantes.

Um exemplo simples de um LBS é aquele que provê ao usuário pontos de interesse (como restaurantes, farmácias, etc.) próximos a sua localização. Esse tipo de sistema, quando solicitado, automaticamente identifica a localização do usuário e a utiliza em sua pesquisa como uma forma de filtro para as informações a serem apresentadas. Normalmente, a pesquisa é encaminhada a um servidor que devolve uma lista de pontos que foram compilados utilizando as informações de localização do usuário e uma distância limite entre o usuário e o ponto em questão. Este tipo de serviço é uma espécie de páginas amarelas que utilizam o contexto de localização para refinar os resultados da busca.

2.3 Fundamentos de localização

No contexto dos LBSs, a palavra localização denota o lugar onde determinado objeto se encontra no mundo real. Segundo este conceito, as localizações podem ser classificadas de três formas:

- **Localização Descritiva** – Representa a geografia natural, como montanhas, rios, ou ainda objetos geográficos feitos pelo homem, como fronteiras, rodovias, prédios, etc. Estas localizações são de fundamental importância, pois construímos nossos sistemas de localização e referência baseados nelas.
- **Localização Espacial (Posição)** – Representa, simplesmente, um ponto no espaço Euclidiano. Normalmente é representada por coordenadas em duas ou três dimensões. Este tipo de localização serve de base para o levantamento e mapeamento de localizações descritivas.
- **Localização na Rede** – Refere-se a forma como redes de telecomunicações identificam seus nodos ou nodos em outras redes.

Uma importante função de um LBS é realizar o mapeamento entre as diferentes categorias de localização, a fim de cumprir com sua tarefa. Por exemplo, uma localização descritiva precisa ser convertida em uma localização espacial para que sua distância em relação a uma outra localização possa ser calculada. Ou ainda, uma localização espacial pode ser convertida em uma localização de rede para que algum tipo de roteamento baseado no contexto seja realizado corretamente. Todos esses mapeamentos podem ser atingidos com a utilização de bases de dados espaciais e Sistemas de Informação Geográfica (GIS - Geographic Information Systems) (Kupper 2005).

2.3.1 Localização espacial

A localização espacial faz uso de sistemas de referência que dividem uma área geográfica, normalmente a terra inteira, em unidades de formato e tamanhos fáceis de serem tratados. Um sistema de referência espacial é composto por:

- um sistema de coordenadas, por exemplo, Sistema de coordenadas cartesianas;
- um ponto de partida, que refere-se a uma definição da forma e do tamanho da terra;
- e uma forma de projeção em mapa (OGC, 2001).

2.3.2 Projeções em mapas

Projeções em mapas são usadas para representar a superfície tridimensional da terra em uma superfície bidimensional, o que é mais conveniente para ser apresentado em monitores e telas de dispositivos móveis.

2.3.3 Bases de dados espaciais e Sistemas de Informação Geográfica (GIS)

A maioria dos métodos de posicionamento usados pelos LBSs resulta na informação de uma localização espacial. O problema é que nem sempre esse tipo de informação é adequado para todos os propósitos do LBS. Por exemplo, no caso de um LBS que apresente em um mapa os possíveis caminhos entre dois pontos de uma cidade utilizando o transporte público, ele deve, de alguma maneira, ser capaz de fazer conversões entre localizações espaciais e localizações descritivas.

As bases de dados espaciais e os GISs representam a tecnologia base para realização desta tarefa. Uma base de dados espacial é uma base de dados otimizada para as necessidades de armazenamento e consultas relacionada ao domínio da localização espacial. Um GIS pode ser caracterizado como um sistema que permite a captura, gerência, integração, análise, manipulação e apresentação de dados espaciais relacionados a Terra. Desta forma, pode-se dizer que uma base de dados espacial é parte integrante de um GIS, sendo responsável pelo armazenamento de dados espaciais (Rigaux, Scholl e Voisard, 2002).

2.4 Posicionamento

O posicionamento é o processo de obter a posição espacial de algum objeto. Existem vários métodos para realizar esta tarefa, sendo que eles diferem em questões como qualidade do resultado, quantidade de processamento realizado para obter o resultado, dentre outros. Posicionamento é, normalmente, determinado pelos seguintes elementos:

- um ou mais parâmetros a serem observados pelos métodos de medição;
- um método de posicionamento para calcular a posição a partir dos parâmetros medidos;
- um sistema de referência descritivo ou espacial;
- uma infraestrutura (rede telefônica, rede de satélites);
- protocolos para coordenar o processo de Posicionamento.

O ponto-chave do posicionamento é a medição dos parâmetros observados. Estes parâmetros refletem uma relação espacial entre o ponto a ser medido e um ou mais pontos fixos próximos a este ponto, onde estes pontos fixos são pontos conhecidos (Hightower e Borriello, 2001a).

2.5 Tecnologias utilizadas no desenvolvimento de aplicações móveis

Neste seção serão apresentadas as principais tecnologias relacionadas com o desenvolvimento de aplicação móveis, quais recursos para obtenção e processamento de informações geoespaciais elas apresentam, e quais as vantagens de utilizá-las no desenvolvimento de LBSs.

No que diz respeito ao desenvolvimento de aplicações móveis, o mercado é dominado por duas grandes plataformas: a Plataforma Android do Google e a plataforma iOS da Apple. Estas duas plataformas juntas representam uma fatia de aproximadamente 75% do mercado de smartphones e tablets. A seguir, temos uma breve introdução destas plataformas e dos recursos disponibilizados por elas para o desenvolvimento de aplicações com base em localização.

2.5.1 Plataforma Android

O Android é uma plataforma de software para dispositivos móveis que inclui sistema operacional, middleware e aplicativos importantes. O SDK (Software Development Kit) do Android fornece as ferramentas e APIs necessárias para desenvolver aplicativos que executam em dispositivos com Android. Algumas de suas principais características são:

- framework de aplicação que permite a reutilização e a troca de componentes;
- Máquina Virtual Java (Dalvik Virtual Machine) otimizada para dispositivos móveis;
- browser integrado;
- interfaces gráficas otimizadas;
- SQLite para armazenamento de dados;
- suporte para uma variedade de formatos de áudio, vídeo e imagens;
- Bluetooth, EDGE, 3G, Wifi, Câmera, GPS, Bússola, etc (dependente de hardware);
- ambiente de desenvolvimento completo, com emulador de dispositivos, ferramentas para debug, gerenciamento de consumo de memória e testes de performance.

2.5.1.1 JVM

A Dalvik Virtual Machine (DVM) é uma máquina virtual Java baseada em registradores. Ela é otimizada para requerer pouca memória, e é projetada para permitir que múltiplas instâncias sejam executadas ao mesmo tempo, deixando para o sistema operacional o isolamento de processos, o gerenciamento de memória e o suporte a threading.

2.5.1.2 Arquitetura

O figura abaixo apresenta os principais componentes da pilha de softwares da plataforma Android.

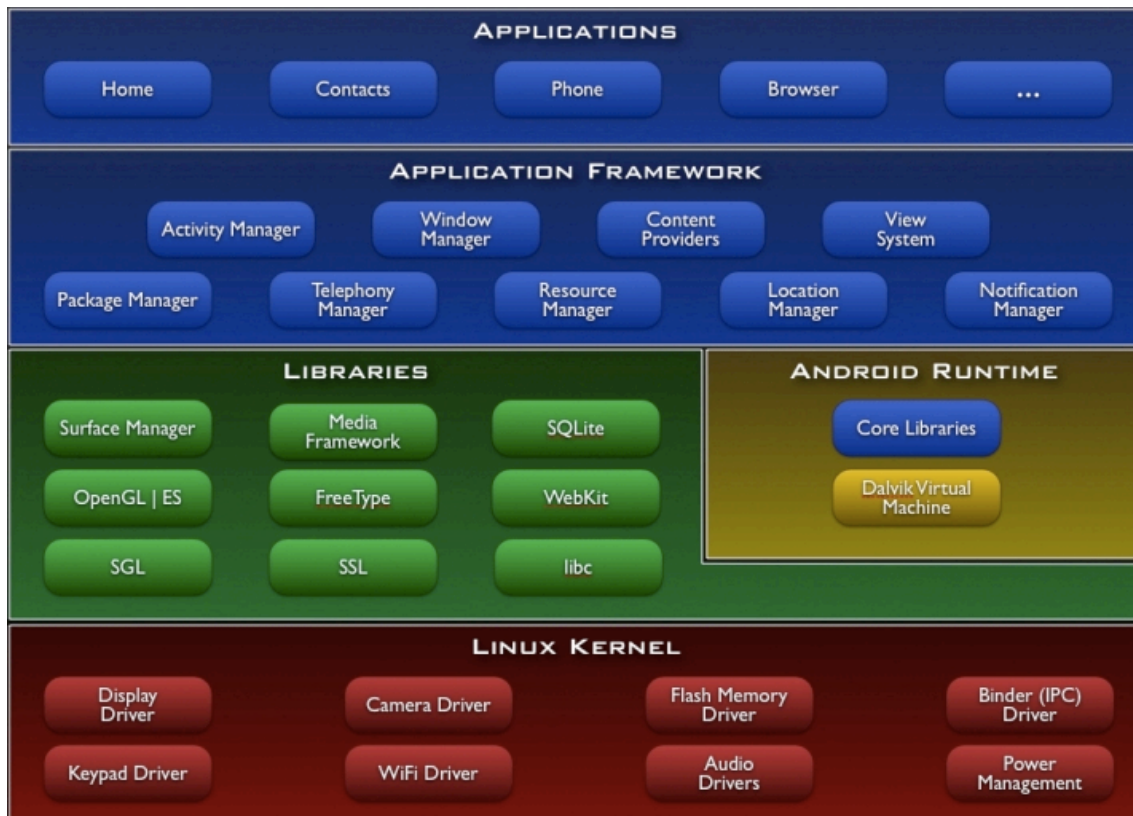


Figura 2.1: Principais componentes do Sistema Operacional Android (Android Developer).

A plataforma Android traz no topo da sua pilha de software um conjunto de aplicações básicas que inclui cliente de email, calendário, mapas (google maps), navegador, lista de contatos entre outros. Todas as aplicações são escritas em Java.

Provendo uma plataforma de desenvolvimento aberta, o Android permite aos desenvolvedores fazer uso de recursos avançados de hardware, como recursos de localização, execução de tarefas em segundo plano, adicionar notificações na barra de status, etc.

Os desenvolvedores tem acesso total as mesmas APIs do framework usadas pelas aplicações básicas providas juntamente com o Android. A arquitetura do Android é projetada visando facilitar a reutilização de componentes. Qualquer aplicação pode publicar uma interface de serviços que ela provê, os quais podem ser acessados por qualquer outra aplicação. Esta mesma arquitetura permite o intercâmbio de componentes pelo próprio usuário.

O Android inclui um conjunto de bibliotecas C/C++, as quais são usadas por vários componentes do sistema. Além disso, um conjunto de bibliotecas básicas provêm a maioria das funcionalidades existentes nas bibliotecas básicas do Java. Cada aplicação Android é executada em um processo separado, sobre sua própria instância da DVM.

O Android é construído sobre o kernel do Linux versão 2.6, o qual fornece serviços de segurança, gerenciamento de memória e processos (threads), além de disponibilizar uma pilha de protocolos de rede. O kernel é a primeira camada de abstração entre o hardware do dispositivo e o restante da pilha de softwares que compõem o Android.

2.5.1.3 Fundamentos das aplicações

Aplicações desenvolvidas para Android são escritas em Java. O código compilado e possíveis arquivos necessários para a execução da aplicação são empacotados pela ferramenta aapt em um pacote Android (extensão .apk). Este tipo de pacote é a forma como aplicações Android são distribuídas e instaladas em dispositivos móveis. Todo o código dentro de um pacote .apk é considerado como sendo uma aplicação.

Por padrão, cada aplicação é executada no seu próprio processo Linux. O Android inicia um processo quando uma aplicação precisa ser executada, e termina o processo quando a aplicação não é mais usada ou os recursos para ela reservados são requeridos por outra aplicação. Cada processo tem sua própria máquina virtual. Desta forma o código de cada aplicação é executado isoladamente do código de outras aplicações. Por padrão, cada aplicação é designada para um único usuário Linux. Permissões são configuradas de forma que os arquivos da aplicação são visíveis apenas pela própria aplicação, embora existam formas de exportá-los para outras aplicações. Também é possível configurar a execução de duas aplicações pelo mesmo usuário Linux. Neste caso elas irão compartilhar seus arquivos. Outra opção é configurar aplicações com o mesmo usuário para que executem o mesmo processo Linux, compartilhando a mesma máquina virtual, o que permite economizar recursos do sistema.

2.5.1.4 Localização e Mapas

O Android provê uma API de localização que pode ser usada para determinar a localização geográfica do dispositivo. Além disso a biblioteca externa do Google Maps pode ser facilmente agregada ao desenvolvimento de aplicações Android, permitindo que aplicações tirem proveito da sua integração com a API de localização.

O Android permite que suas aplicações acessem os serviços de localização providos pelo dispositivo por meio do pacote de classes android.location. O componente central deste pacote é o serviço de sistema chamado LocationManager, o qual provê uma API de localização, caso o dispositivo em questão suporte tais serviços.

O Google disponibiliza uma biblioteca externa que inclui o pacote com.google.android.maps. As classes desse pacote oferecem uma série de serviços para manipulação de mapas, além de uma variedade de opções de visualização e controle. A principal classe deste pacote é com.google.android.maps.MapView, a qual apresenta um mapa com base em dados obtidos a partir dos serviços Google Maps. Quando o MapView tem o foco, ele captura teclas pressionadas e toques de tela para automaticamente manipular o mapa (ex. zoom). O MapView também provê todos os elementos necessários para que o usuário controle o mapa. A biblioteca Google Maps faz parte dos módulos de extensão Google API para Android SDK (Android Developer).

2.5.2 Plataforma iOS

O sistema operacional iOS é o sistema utilizado pela Apple em seus dispositivos móveis equipados com telas sensíveis ao toque. É baseado no sistema operacional Mac OS X incluso em computadores Macintosh que, por sua vez, é baseado no sistema UNIX BSD. Anteriormente ao lançamento da versão 4.0, era conhecido como iPhone OS, mas teve seu nome modificado para melhor refletir o fato de que ele também serve como base para outros produtos da empresa. O iOS, atualmente, serve de plataforma para três linhas de produtos: iPhone, iPod Touch e iPad.

O desenvolvimento de aplicativos para iOS é feito com o uso do iPhone SDK, um conjunto de ferramentas de desenvolvimento acompanhadas de documentação. O kit é disponibilizado no site da empresa para desenvolvedores cadastrados e sua instalação requer um computador equipado com a versão mais recente do Mac OS X. O cadastro no site e o download do SDK podem ser feitos gratuitamente, o que permite usar as ferramentas e testar aplicações em um simulador. Para realizar testes em dispositivos reais e submeter aplicativos para publicação na Apple Store (loja de aplicativos da Apple), é preciso pagar uma taxa anual de aproximadamente US\$99,00. O pagamento dessa taxa também traz outros benefícios como suporte técnico e acesso a versões beta do iOS e do SDK.

As principais ferramentas inclusas no SDK são:

- Xcode – Ambiente integrado de desenvolvimento (IDE), usado para gerência, edição, compilação e depuração de código-fonte e demais recursos associados a um projeto.
- Interface Builder – Usado para criação de interfaces gráficas de forma visual.
- Storyboard – Permite a definição em detalhes do fluxo da aplicação.
- Instruments – Ferramenta de depuração e análise de desempenho usada para identificar problemas como vazamentos de memória e gargalos de desempenho. Permite a visualização em tempo real de diversos aspectos de uma aplicação enquanto ela é executada no simulador ou em um dispositivo real conectado à máquina de desenvolvimento.
- Simulator – Simula iPhones e iPads, executando aplicações em um ambiente semelhante ao encontrado em um dispositivo real.

Aplicativos iOS são escritos primariamente na linguagem Objective-C, mas podem incluir código escrito em C e/ou C++. A linguagem Objective-C adiciona à linguagem C mecanismos de reflexão e orientação a objeto inspirados na linguagem Smalltalk. Por se tratar de um super conjunto de C, qualquer código escrito de acordo com o padrão ANSI C também é válido como Objective-C.

2.5.2.1 Plataforma

As tecnologias disponibilizadas aos desenvolvedores são organizadas em um conjunto de camadas, onde as camadas mais baixas são compostas de serviços fundamentais e as mais altas contém serviços mais sofisticados.

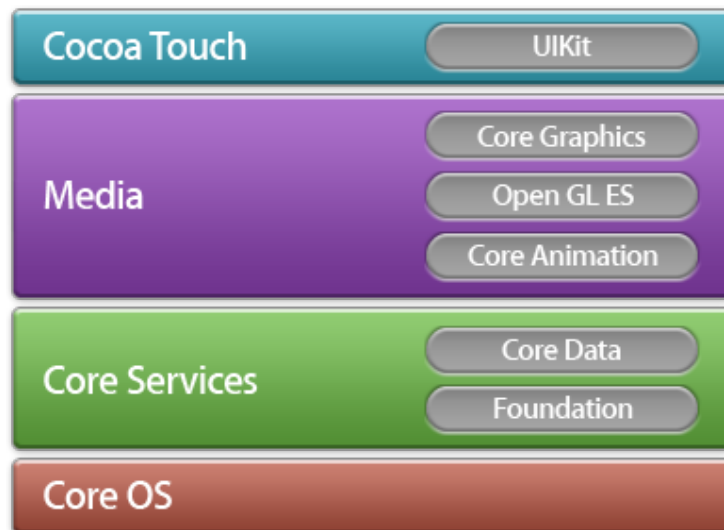


Figura 2.2: Camadas do Sistema Operacional iOS (Apple Developer).

A documentação de introdução à plataforma recomenda o uso dos níveis mais altos de abstração sempre que possível, pois eles facilitam o desenvolvimento através do encapsulamento de funcionalidades mais complexas. No entanto, o uso de frameworks de nível mais baixo é disponibilizado para desenvolvedores que preferem utilizá-los ou que precisam de acesso a aspectos não expostos pelos níveis mais altos. A seguir uma breve descrição do que pode ser encontrado em cada camada.

- Core OS – A camada mais próxima do kernel. Inclui acesso direto a funcionalidades de baixo nível, como threads POSIX, sockets BSD e acesso ao sistema de arquivos, entre outros.
- Core Services – Provê os serviços fundamentais usados por todas as aplicações. Inclui, entre outros, frameworks de localização, persistência de dados, telefonia, acesso à agenda de contatos, acesso ao calendário e visualização de arquivos com formatos suportados pelo sistema.
- Media – Inclui frameworks de reprodução e gravação de áudio e vídeo e bibliotecas gráficas aceleradas por hardware para animação e renderização 2D e 3D. A renderização de gráficos 3D é feita através da API OpenGL ES, uma versão simplificada do padrão OpenGL projetada para sistemas embarcados.
- Cocoa Touch – Contém a infraestrutura para implementação da interface visual das aplicações, incluindo eventos gerados pelo usuário, reconhecimento de gestos, funcionalidades multitarefa (introduzidas na versão 4.0 do iOS), notificações locais e remotas e acesso aos sensores (acelerômetro, bússola, etc.) e à câmera embutida.

2.5.2.2 Localização e Mapas

Na plataforma iOS, os serviços e recursos de localização são fornecidos pelo Core Location framework, o qual permite a obtenção da localização atual do dispositivo assim como o monitoramento de alterações nesta localização. Assim como no Android, o principal componente do framework de localização é o gerenciador de localização, representado no iOS pela classe CLLocationManager. Além de permitir o acesso e o monitoramento a informações de localização, o framework Core Location também permite, através de sua API, a conversão entre localizações espaciais e descritivas.

No iOS, a renderização e manipulação de mapas fica a cargo do framework Map Kit, onde a classe MKMapView contém toda a informação necessária para apresentação de mapas e tratamento de eventos relacionados e estes.

2.5.3 Servidor

Praticamente todas as aplicações móveis fazem uso de serviços externos (normalmente Web Services) para realizar suas tarefas e apresentar aos usuários seus resultados. Com a aplicação Bus Tracker não é diferente. Nela, o módulo Servidor é a parte da aplicação responsável por processar e responder às requisições dos clientes (dispositivos móveis). O processamento de uma requisição passa pelo acesso a informações armazenadas na base de dados e posterior processamento destas informações, para então entregar ao cliente aquilo que foi solicitado em sua requisição (por exemplo, o resultado da busca por uma linha).

Hoje, o número de tecnologias utilizadas no desenvolvimento de Web Services é grande. Cada uma com suas características, funções, vantagens e desvantagens. O objetivo não é avaliar qual a tecnologia mais apropriada ao desenvolvimento de serviços que servem à aplicações móveis, uma vez que, para a grande maioria das aplicações esta escolha é muito mais de gosto pessoal do que uma escolha que leve em considerações características técnicas. Algumas destas tecnologias são amplamente usadas em aplicações comerciais, como por exemplo, a plataforma Java EE (Java Enterprise Edition) que tem seu uso difundido como uma das plataformas mais seguras, estáveis e escaláveis. As grandes desvantagens do Java EE são a questão de performance, a qual para ser atingida exige um grande consumo de hardware, e o custo alto de desenvolvimento para se ter uma aplicação funcionando. As linguagens de scripts (como PHP, Ruby, Python) e seus frameworks têm a preferência de desenvolvedores que usam metodologias de desenvolvimento ágeis. Isso porque, frameworks como Rails escrito em Ruby, Django escrito em Python e tantos outros escritos em PHP, trazem pronta grande parte das funcionalidades necessárias para o desenvolvimento de aplicações Web, permitindo que os desenvolvedores foquem especificamente na lógica de negócio relativa ao domínio da aplicação em questão. Além disso, estas tecnologias são reconhecidas pela performance, em especial Python, que é utilizada por grandes empresas (como Google, NASA) em seus servidores, com resultados surpreendentes (PYTHON).

3 RASTREAMENTO VEICULAR APLICADO AO TRANSPORTE URBANO

Antes mesmo da popularização de tecnologias como o GPS, algumas cidades já experimentavam o rastreamento veicular como ferramenta de controle e melhoria dos serviços de transporte público. Em meados da década de 90, Londres (Inglaterra) deu início à implantação do seu sistema de rastreamento. O primeiro sistema implantado na cidade, denominado de Countdown, utilizava transponders, permitindo assim a detecção dos veículos ao passarem por pontos de leitura. O sistema começou a ser testado em 1992 nos ônibus da rota 18. Os testes realizados com os passageiros sobre o projeto foram muito positivos. Entre 1993 e 1994 o projeto foi expandido e amplamente testado em outros corredores de ônibus. A partir de 1996 teve início a instalação de sistemas automáticos de localização de veículos como tecnologia de rastreamento. Em março de 2002, 1473 painéis já haviam sido instalados e estavam operacionais. O plano era ter um total de 2400 painéis instalados em março de 2003 e 4000 no ano de 2005. Esse número de 4000 painéis instalados chegaria a cobrir 25% de todos os pontos de parada de Londres e beneficiaria 60% de todas as viagens de passageiros (Magalhães, 2008).

Além de Londres, diversas cidades europeias têm investido em tecnologias visando a criação de sistemas para monitoramento do transporte público em tempo real. As informações obtidas por estes sistemas, e que são de interesse dos usuários (como localização dos veículos), são disponibilizadas por meio de painéis de mensagem em pontos de parada e terminais de embarque e desembarque, assim como por meio de computadores ou celulares que tenham acesso a Internet.

Nos Estados Unidos, estudos do Transit Cooperative Research Program (TCRP) demonstram que agências de trânsito têm realizado investimentos em tecnologias que proporcionam mais segurança, eficiência e qualidade em seus serviços. Assim como na Europa, muitas cidades norte-americanas têm investido no rastreamento de veículos para uma melhor gestão e eficiência dos meios de transporte.

No Brasil algumas cidades de médio e grande porte já utilizam o sistema de rastreamento por GPS para gestão de parte ou de toda a frota do transporte público. Mas a utilização e disponibilização das informações obtidas por estes sistemas para uso em aplicações que beneficiem o usuário final ainda é pequena.

O número de cidades que tem algum tipo de infraestrutura de rastreamento aplicada ao monitoramento do transporte urbano é cada vez maior. Alguns bons exemplos são Auckland (Nova Zelândia), Denver (Colorado – USA), Sydney (Austrália), Uberlândia, que foi a primeira cidade brasileira a ter toda a frota de transporte urbano monitorada (GeoSIT), e Helsinque (Finlândia), que integrou o sistema de rastreamento com

semáforos permitindo que veículos que estejam atrasados tenham prioridade de passagem.

Tanto o poder público quando as empresas de transporte estão cada vez mais conscientes da necessidade de melhoria constante deste serviço tão essencial que é o transporte urbano. Na iniciativa privada não é diferente, pois já é grande o número de empresas que de alguma forma exploram este mercado tão promissor. A empresa Maxtrack, por exemplo, oferece um grande número de equipamentos e soluções de rastreamento, tanto para o transporte urbano de passageiros, como para outras áreas de transporte. Algumas empresas como a G3VB Sistemas e Tecnologia (G3VB), Cittati e Wplex oferecem soluções completas (hardware, software, treinamento, etc.) voltadas para o transporte público, com serviços próprios para este tipo aplicação. A Wplex, por exemplo, desenvolveu o sistema WplexCO, o qual permite o monitoramento da frota de ônibus em tempo real, assim como a comunicação direta com os motoristas em trânsito (Wplex). A instalação do sistema demanda pouco investimento em infraestrutura, pois ele foi desenvolvido segundo o conceito de Software como um Serviço (SaaS). Ou seja, todas as informações de gerenciamento e todo o monitoramento é feito via Web. A comunicação de dados entre veículos e empresa ocorre no seguinte sentido: junto ao veículo é instalado um rastreador GPS/GPRS embarcado e um terminal com teclado; o rastreador envia informações de posicionamento, assim como outras informações de controle para o centro de dados da empresa, onde as informações são processadas e armazenadas em uma base de dados. A empresa de transporte tem acesso, via Web, a serviços que consultam tal base de dados, permitindo assim, o rastreamento e controle da frota por parte da empresa de transporte. O WplexINFO é parte complementar do WplexCO, disponibilizando aos usuários informações e serviços que facilitam a utilização do transporte público, como itinerários, pontos de parada, previsão de chegada, dentre outras.

Um panorama geral de como tais tecnologias vêm sendo utilizadas em diferentes cidades, com o objetivo de melhorar constantemente o transporte urbano, é apresentado nas próximas seções.

3.1 Helsinque – Finlândia

O projeto HelMi (HelMi), desenvolvido em Helsinque pelo Urban Traffic Control Center [UTCC], utiliza recursos da telemática para prover um sistema de monitoramento do transporte público cujos principais objetivos são:

- prioridade nas interseções semaforicas;
- informação ao passageiro;
- controle da frota.

O processo de obtenção das informações de localização dos veículos é dividido em três etapas:

- o sistema de GPS fornece continuamente a posição do veículo antes e depois da sua passagem pelos pontos de parada;
- a precisa localização da direção de deslocamento do ônibus na rota é dada pelo uso de tecnologias auxiliares;
- hodômetros são usados para calcular com precisão a distância entre um ponto e outro da rota, permitindo a obtenção precisa da localização do ônibus na rota.

A priorização do sinal de tráfego consiste em manter um determinado semáforo aberto além do seu tempo normal, uma vez que algum ônibus em atraso esteja se aproximando dele. Esse sistema funciona a partir de um processo de requisição, onde o ônibus emite um sinal ao se aproximar das interseções onde os semáforos possuem detectores instalados. Como já foi dito, a priorização somente é concedida a veículos que não estejam com sua viagem adiantada. Um segundo sinal é emitido imediatamente depois da passagem do veículo pelo semáforo e o processo de ampliação do sinal verde é encerrado.

No sistema de rastreamento por GPS usado em Helsinque, as informações são pré-programadas nos computadores ficando ao encargo do motorista somente informar o número correto do veículo e o horário de início da primeira viagem. O sistema controla automaticamente as demais informações, avisando passageiros sobre os próximos pontos de parada, enviando sinais de requisição de prioridade aos semáforos, e emitindo informações em tempo real para a central de controle. Informações sobre o tempo de chegada dos veículos nos pontos de parada são mostradas aos passageiros em painéis de mensagens. Na figura a seguir é apresentado um esquema ilustrando o funcionamento do sistema implantado em Helsinque.

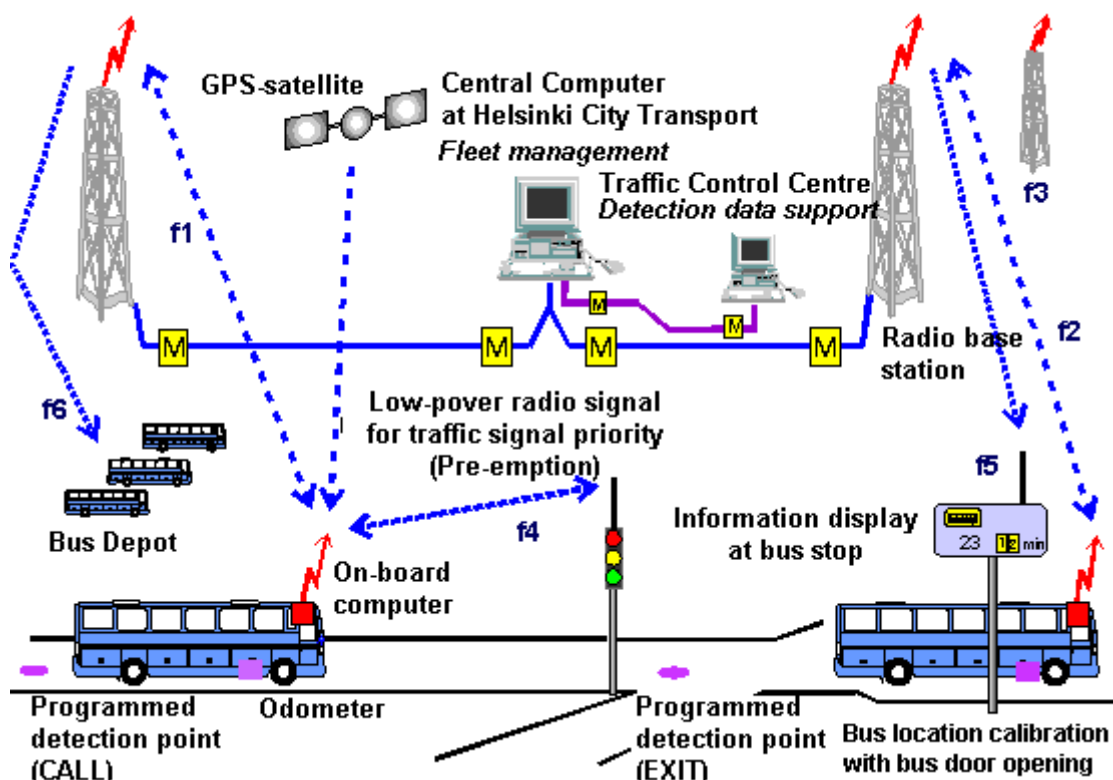


Figura 3.1: Configuração no sistema no projeto HelMi de Helsinque (UTCC).

3.2 Auckland – Nova Zelândia

A autoridade regional do transporte de Auckland (ARTA - Auckland Regional Transport Authority) e os operadores do sistema de ônibus trabalharam juntos na instalação do sistema de monitoramento do transporte urbano no intuito de melhorar a qualidade dos serviços prestados e incentivar o uso do transporte coletivo por parte de outros segmentos da população.

A partir de informações de posicionamento obtidas e enviadas por meio de um módulo GPS/GPRS instalado em cada veículo, informações de previsão de chegada dos veículos são disponibilizadas em painéis de mensagem instalados em pontos de parada. No início de cada viagem os motoristas inserem dados relacionados ao número da rota e horário de partida. Os ônibus também possuem equipamentos audiovisuais em seu interior, os quais são usados para informar qual será a próxima parada, assim como mensagens institucionais e informações sobre pontos turísticos. Além disso, o sistema tem sido usado para monitorar o transporte dos passageiros por parte das operadoras e pelo ARTA.

A implantação do sistema faz parte do projeto de transporte sustentável adotado pela cidade com o objetivo de atrair mais usuários para o transporte público municipal. Desde a implantação do sistema em 2003 a cidade já possui equipamentos de GPS instalados em mais de 730 veículos, 180 painéis de informações em pontos de parada, além de equipar 174 interseções semaforizadas com o sistema de detecção dos veículos de transporte (Auckland Transport).

3.3 Denver, Colorado - USA

Estudos feitos pela TCRP levantaram dados sobre o uso da tecnologia de rastreamento de veículos por GPS na cidade de Denver, sendo relatado que o Regional Transportation District (RTD) foi uma das primeiras agências de transporte público a implantar um sistema de localização automática de veículos baseado na tecnologia de GPS (TCRP).

A decisão de compra dos módulos de rastreamento foi feita em 1992, e o término da implantação destes foi em 1996. A partir de 1999 a agência começou a utilizar os dados gerados pelo sistema para fornecer informações, em tempo real, para os passageiros do transporte público. Em Dezembro de 2001 foram introduzidos novos serviços que disponibilizam informações da chegada dos ônibus nos pontos de parada através do Talk-n-Ride (serviço via telefone) e do Bus Locator através do site <http://www.rtd-denver.com>. Além dos próprios serviços disponíveis no site, existem outras ferramentas, tanto para smartphones (<http://www.rtd-denver.com/MobileTools.shtml>) quanto para outros dispositivos, que fazem uso das informações disponibilizadas pelo sistema de rastreamento para fornecer outros serviços aos usuários (RTD).

Ao permitir que sistemas desenvolvidos por terceiros tenham acesso as informações do sistema de rastreamento, o RTD possibilitou a criação de serviços especializados para determinado grupo ou tipo de usuário. Criando assim um nicho de mercado, onde existem milhares de prováveis clientes.

3.4 Sydney – Austrália

A cidade de Sydney, assim como muitos outros grandes centros urbanos, tem disponibilizado informações para os passageiros do transporte público em tempo real permitindo o acesso através do uso de smartphones, celulares que se conectam a Internet, telefone convencionais, dentre outros meios. Por meio do site <http://www.131500.com.au>, o usuário pode planejar sua viagem, sendo disponibilizadas as informações sobre todos os procedimentos necessários para se chegar ao destino pretendido, como a localização dos pontos de parada, o meio de transporte a ser usado

(ônibus, trem ou bonde), o tempo e distância a ser percorrida, o tempo de chegada de cada veículos no ponto de parada dentre outras (NSW – TransportInfo).

3.5 Fortaleza – Brasil

Com a implantação do projeto Controle Integrado de Transportes de Fortaleza (CITFOR), os usuários passaram a usufruir de informações e operação otimizada não disponível até então. O sistema funciona a partir de um computador de bordo instalado nos ônibus. Este computador recebe informações de geoposicionamento (GPS), que informa a sua exata localização, e envia estas informações à central utilizando serviços da rede de celular (GPRS). O servidor CITFOR recebe as informações de todos os veículos da frota, analisa os dados e repassa as informações para os painéis de mensagem, bem como para o sistema de semáforos (CITFOR).

3.6 São Paulo – Brasil

Através de um projeto de reestruturação do sistema de transporte municipal, a Prefeitura de São Paulo criou o projeto de Transporte Integrado, que conta com o uso de diversas ferramentas inteligentes nos sistemas de transporte para garantir a operação do transporte coletivo. Antes de tais investimentos, não se tinha controle sobre a localização ou o cumprimento dos itinerários e horários programados para os veículos, ou seja, não era possível a obtenção dos dados pertinentes à operação da frota do transporte público.

Em São Paulo, uma cidade onde as linhas são longas e o trânsito sofre com congestionamentos, é fundamental que seja possível, a partir de um Centro de Controle, tomar decisões que garantam a circulação normal dos veículos. Para tanto foi elaborado um modelo de comunicação, localização dos veículos e informação aos usuários totalmente integrados.

As metas a serem atingidas com o rastreamento dos veículos são:

- dispor de infraestrutura de equipamentos e de um sistemas que permita supervisionar e controlar a operação da frota de veículos em circulação;
- identificar e tratar ocorrências específicas como: quebras mecânicas, assaltos, ocorrências médicas, congestionamentos, etc;
- estabelecer comunicação direta com o motorista de ônibus, notificando-o sobre eventuais problemas operacionais ou sobre o cumprimento do horário da viagem.

O projeto de monitoramento tem sido implantado na cidade de São Paulo por iniciativa da SPTrans, órgão responsável pelo gerenciamento do sistema de transporte público do município. Esse sistema de monitoramento, baseado na utilização do sistema de posicionamento global (GPS), visa melhorar a eficiência do controle dos serviços prestados pelas operadoras e também gerar uma base de dados estruturada para o planejamento das linhas (SPTrans).

3.7 Uberlândia – Brasil

A cidade de Uberlândia conta com o Sistema Integrado de Transporte (SIT), o qual foi projetado para racionalizar o transporte coletivo e para que este opere com maior

produtividade e qualidade, agilizando o serviço prestado e acabando com o pagamento de nova passagem a cada transbordo realizado. A implantação deste sistema teve início em 1997 (SIT, 2005).

Em 2003, através de processo de licitação, a cidade de Uberlândia firmou um contrato com o Consórcio VERMAX para realizar o monitoramento da frota de veículos do SIT. Este novo projeto foi denominado GeoSIT (GEOSIT).

A implantação do GeoSIT teve início com a instalação de módulos GPS/GPRS nos veículos. Tais módulos foram adquiridos junto à empresa Maxtrack, uma das líderes no mercado nacional de monitoramento veicular. A figura a seguir apresenta um dos módulos instalados em veículos do SIT.



Figura 3.2: Módulo de monitoramento veicular Maxtrack (GEOSIT).

A transmissão dos dados, a partir dos módulos, é feita utilizando a tecnologia GPRS, a qual consiste em uma conexão contínua, sem fio, com redes de dados GSM. Nesse sistema, a transmissão é realizada por pacotes, possibilitando a transferência de dados em alta velocidade, com a vantagem da cobrança da transmissão ser realizada por bytes recebidos e enviados e não mais por tempo de conexão, promovendo um barateamento dos custos com transmissão de dados.

O módulo possui um receptor de GPS que envia uma vez por segundo a data, hora, latitude, longitude, direção e velocidade do veículo. As informações são processadas gerando novas informações, que são transmitidas para a central de monitoramento. O modo de controle on-line permite que sejam feitas, em tempo real, consultas à posição atual do veículo, alterações na configuração do módulo, além de permitir a abertura de um canal de voz para comunicação completa com o veículo.

O software de gerenciamento da frota foi desenvolvido visando permitir o rastreamento e gerenciamento de rotas utilizando as informações transmitidas para um servidor pelo módulo instalado nos veículos. Esse software é composto por um aplicativo servidor e um cliente que é instalado nos computadores onde ficará a central de monitoramento, possibilitando a comunicação local ou remota com os veículos do SIT.

3.8 Porto Alegre – Brasil

Porto Alegre, capital do estado do Rio Grande do Sul, tem aproximadamente 1.500.000 habitantes e conta com 1570 ônibus que transportam 25.000 passageiros todo dia. O sistema de transporte é formado por quatro empresas de ônibus (Carris, Conorte, STS, Unibus) que disponibilizam diferentes linhas. Em 1998 foi criada a Empresa

Pública de Transporte e Circulação (EPTC), para regular e fiscalizar as atividades relacionadas com o trânsito e os transportes no município, atendendo a uma tendência internacional de municipalização da mobilidade urbana.

Com a intenção de melhorar a qualidade dos serviços prestados pelas empresas de ônibus, em 1998, a EPTC adquiriu e implantou o Sistema de Ônibus Monitorado Automaticamente (SOMA). O sistema é constituído de 52 estações de monitoração distribuídas pela cidade, que registram a passagem dos ônibus e transmitem as informações via rádio para o centro de controle e operação. O sistema tem como principal objetivo fiscalizar o cumprimento dos horários e realização das viagens previstas. Com o sistema, a frota de ônibus é monitorada 24 horas por dia, todos os dias da semana. E a cobertura das viagens realizadas é de 97%.

O SOMA é baseado em etiquetas eletrônicas (transponders RFID - Radio Frequency Identification) instaladas em todos os ônibus. Estações fixas, distribuídas em vários pontos da cidade, identificam os veículos e o horário da passagem transmitindo os dados através de um sistema de rádio-modem até a central.

Os elementos do sistema são agrupados em quatro subsistemas:

- Sistema de identificação – Este é composto por transponders instalados em cada ônibus e pelas estações de leitura, que contêm leitores e uma antena de solo.
- Unidade de Controle Local – Localizada a cada estação de leitura, executa a comunicação com o leitor de transponder, recebendo a identificação de cada ônibus, a qual é armazenado juntamente com o horário de passagem para posterior transmissão à central.
- Sistema de Comunicação – Composto pelo rádio da central, pela retransmissora localizada em torre própria no morro São Caetano, além dos rádios de cada ponto de leitura.
- Sistema de Monitoramento – Consiste basicamente de um software que executa a função de leitura dos dados. O software se comunica com as estações, utilizando um protocolo proprietário, para receber e armazenar os dados de monitoramento.

As estações de leitura são compostas de um quadro elétrico contendo rádio-modem, uma fonte de alimentação, uma unidade de controle e unidades de leitura de transponder. Os quadros são instalados em postes montados próximos aos laços indutivos (antenas) enterrados na via. Cada laço possui uma caixa de sintonia para ajustes da frequência de ressonância e sensibilidade.

O Centro de Controle e Operação (CCO) é composto de um computador que executa o software de controle e se comunica com as estações via rádio-modem. O CCO coleta os dados das estações e gera os registros em arquivos para que um sistema gerencial interprete os dados e gere as informações que serão formatadas na forma de relatórios (Alfacomp).

3.9 Considerações finais

Como se pode ver até aqui, a tendência tecnológica no rastreamento de veículos aplicado ao transporte urbano é a utilização de módulos rastreadores, com tecnologias GPS e GPRS, instalados nos veículos. Estes módulos enviam suas informações para um centro de dados que as processa e as armazena em uma base de dados. Além disso, a

utilização de sistemas web (software como um serviço), para disponibilizar serviços tanto para a empresa de transporte quanto para os usuários, é comum em quase todos os sistemas.

4 PROJETO

Este capítulo apresenta a análise e o projeto da aplicação Bus Tracker, a qual deve possibilitar que seus usuários acompanhem o deslocamento de veículos do transporte urbano por meio de dispositivos móveis, além de notificar estes usuários sobre a ocorrência de eventos por eles configurados, como por exemplo, a aproximação de um veículo do ponto de embarque do usuário.

4.1 Engenharia de Software

A Engenharia de software é uma área do conhecimento voltada para a especificação, desenvolvimento e manutenção de sistemas de software, aplicando tecnologias e práticas que englobam linguagens de programação, banco de dados, ferramentas, plataformas, bibliotecas, padrões e processos dentre outras. Os fundamentos científicos da engenharia de software envolvem o uso de modelos abstratos que permitem especificar, projetar, implementar e manter sistemas de software, avaliando e garantindo suas qualidades. Além disso, a engenharia de software oferece mecanismos para planejar e gerenciar o processo de desenvolvimento (Magela, 2006).

Assim como toda engenharia, a engenharia de software tem seus processos. Um processo de desenvolvimento de software é uma sequência coerente de práticas que objetiva o desenvolvimento ou evolução de sistemas de software. Estas práticas englobam as atividades de especificação, projeto, implementação e testes; caracterizam-se pela interação de ferramentas, pessoas e métodos.

À medida que as tecnologias relacionadas ao desenvolvimento de software evoluem, as metodologias, ou seja, os processos de engenharia de software, também evoluem. Um bom exemplo foi a mudança de paradigma proposta pelo Processo Unificado em relação ao Modelo em Cascata ou Sequencial, como também é chamado. Em um Modelo em Cascata há uma tentativa de definir em detalhes todos ou a maioria dos requisitos, ou ainda, criar um projeto abrangente antes da programação. Há também a tentativa de definir um plano ou cronograma confiável logo no início do processo. A suposição de que as especificações são previsíveis e estáveis, e podem ser corretamente definidas no início, com baixas taxas de modificação, faz com que o modelo em cascata não seja realmente adequado ao desenvolvimento de software, levando assim a uma taxa grande de falhas, quando da sua utilização.

4.2 Processo unificado

Neste trabalho será utilizada como metodologia de desenvolvimento uma abordagem ágil do Processo Unificado (PU) (Larman, 2004). A ideia central nesta abordagem é a

de usar iterações curtas, com duração fixa, em um processo de desenvolvimento iterativo, evolutivo e adaptativo. Algumas das melhores práticas desta abordagem incluem:

- enfrentar os problemas que envolvem altos riscos e alto valor já nas iterações iniciais;
- construir uma arquitetura central coesa nas primeiras iterações;
- verificar continuamente a qualidade; fazer testes logo de início, com frequência e em situações o mais realísticas possíveis;
- aplicar casos de uso quando adequado;
- modelar visualmente o software (UML);
- gerenciar requisitos cuidadosamente.

Um projeto que utilize o PU deve organizar o trabalho e as iterações de acordo com as seguintes fases:

- **Concepção** – A concepção é um passo inicial curto, para estabelecer uma visão comum e o escopo básico do projeto. Inclui a análise dos requisitos funcionais críticos e a preparação do ambiente de desenvolvimento para que a programação possa começar na fase seguinte;
- **Elaboração** – A elaboração é a fase inicial de iterações durante a qual a arquitetura central e de alto risco do software é programada e testada, a maioria dos requisitos é descoberta e estabilizada e os principais riscos são mitigados e retirados;
- **Construção** – A construção representa a segunda fase de iterações onde são implementados os elementos restantes de menor risco e mais fáceis, e onde é feita a preparação para a implantação;
- **Transição** – A transição é a fase de testes beta e implantação.

Cada uma das fases descritas acima produz seus próprios artefatos (artefato é o termo usado para qualquer produto de trabalho, como código, gráficos, esquemas de banco de dados, documentos de texto, modelos, dentre outros). Entretanto, um aspecto importante do PU é que quase todas as atividades e os artefatos são opcionais, ficando a critério da equipe do projeto definir quais práticas adotar e quais artefatos gerar.

4.3 Análise de Requisitos

Requisitos são capacidades e condições às quais o sistema deve atender. O PU promove um conjunto de melhores práticas, dentre as quais gerenciar requisitos tem destaque, pois ela estimula o uso de uma abordagem sistemática para encontrar, documentar, organizar e rastrear as mudanças de requisitos de um sistema (Kruchten, 2000). Para tanto, o PU oferece vários artefatos de requisitos. Os principais são:

- **Modelo de Casos de Uso** – Conjunto de cenários típicos de uso do sistema;
- **Especificação Suplementar** – Tudo o que não está nos casos de uso. Este artefato se aplica principalmente a requisitos como desempenho, autenticação, etc;
- **Glossário** – Define termos importantes do domínio da aplicação;
- **Visão** – Resume requisitos de alto nível que são detalhados no modelo de casos de uso e especificação suplementar. É um documento curto contendo uma visão geral para entendimento rápido das grandes ideias do projeto;

4.4 Concepção – 1ª Fase do PU

A concepção é a fase do PU em que o escopo do produto é concebido. Esta fase tem como alguns de seus artefatos: a Visão, o Modelo de Casos de Uso, Especificações Suplementares, Lista de Riscos, Protótipos e Provas de Conceito. As subseções a seguir apresentam os artefatos escolhidos para este trabalho (Visão e Modelo de Casos de Uso) na fase de concepção.

4.4.1 Visão

Como foi mencionado no capítulo introdutório, o propósito deste trabalho é o projeto e prototipagem de uma aplicação para dispositivos móveis que permita aos usuários do transporte público, não apenas acompanhar em tempo real o deslocamento dos veículos de transporte, como também obter serviços de alerta (alarmes, mensagens) à medida que determinados eventos (como a aproximação de determinado veículo de um ponto específico) são detectados pela aplicação.

Em um nível mais elevado de abstração, a aplicação pode ser descrita como a interação entre os seguintes componentes:

- uma Base de Dados que armazena informações referentes ao domínio da aplicação (que neste caso é o setor de transporte coletivo de uma cidade) e informações de tempo real pertinentes ao rastreamento dos veículos de transporte;
- um módulo Servidor que interage com a base de dados para obter as informações necessárias ao processamento das requisições recebidas de clientes;
- um módulo Cliente, a ser instalado em dispositivos móveis, que permite que aos usuários do transporte obtenham benefícios das informações de rastreamento por meio de requisições ao módulo Servidor.

Nas seções a seguir é dada uma definição mais detalhada dos três componentes e de suas interações, tendo como ponto de partida a interface com o usuário final, ou seja, o módulo cliente.

4.4.1.1 Módulo Cliente

Sendo uma das três partes fundamentais da aplicação, o módulo Cliente representa a interface com os usuários. É através do módulo Cliente, o qual é uma aplicação para dispositivos móveis, que o usuário poderá acompanhar, em tempo real, os veículos do transporte público.

A aplicação Cliente deve permitir ao usuário, como serviços básicos, localizar determinada linha de ônibus através de um sistema de busca e permitir que ele visualize, em um mapa, a posição dos veículos desta linha em tempo real. A busca por uma linha pode utilizar os seguintes critérios:

- o usuário simplesmente busca uma linha pelo nome ou parte do nome que identifica a linha;
- o usuário pode marcar um determinado ponto no mapa e buscar por linhas que passem próximas a ele;
- caso o dispositivo móvel utilizado tenha recursos tecnológicos que permitam obter a posição do usuário (como o GPS), ele pode refinar os métodos de busca

levando em consideração sua posição atual, como no caso de buscar por linhas próximas a ele.

A partir de uma linha selecionada, a aplicação deve permitir a visualização do itinerário, dos pontos de parada, assim como a posição atual dos veículos (ônibus, lotação, etc.) que pertencem a esta linha, sendo esta posição atualizada a uma taxa constante, a qual poderá ser configurada pelo usuário.

Até aqui foram definidos diferentes meios pelos quais o usuário pode localizar uma determinada linha e visualizar a posição de seus veículos em tempo real. Este serviço, por si só, já se apresenta como algo bastante útil. Mas o propósito desta aplicação é ir um pouco além, permitindo que o usuário possa cadastrar eventos espaciais pelos quais deseja ser avisado. No contexto de eventos espaciais relacionados ao domínio da aplicação, a aproximação de um veículo de um ponto específico dentro do seu itinerário é sem dúvida um excelente ponto de partida para definir quais eventos devem ser reconhecidos pela aplicação. Sendo assim, a aplicação deve permitir que os usuários configurem eventos desta natureza, pelos quais desejam ser avisados, e a forma de aviso que desejam receber (alerta sonoro, mensagem visual, etc.). A próxima seção apresenta a definição do módulo Servidor, o qual recebe as requisições dos clientes.

4.4.1.2 Módulo Servidor

O módulo Servidor é a parte da aplicação responsável por processar e responder as requisições dos clientes. O processamento de uma requisição passa pelo acesso a informações armazenadas na base de dados e posterior processamento destas informações, para então entregar ao cliente aquilo que foi solicitado em sua requisição (por exemplo, o resultado da busca por uma linha). O módulo Servidor deve ser capaz de responder a todas as requisições necessárias para que todos os requisitos definidos na seção Módulo Cliente sejam atendidos.

4.4.1.3A Base de Dados

Não sendo finalidade deste trabalho definir como os dados de rastreamento são obtidos, é então, assumido como verdadeira a existência de uma infraestrutura de rastreamento veicular para o transporte público e que, de alguma forma, as informações obtidas por esta infraestrutura são armazenadas na base de dados da aplicação em tempo real. Partindo dessa premissa, a base de dados da aplicação deve ser capaz de armazenar as informações referentes ao domínio da aplicação (linhas, veículos, itinerários, etc.), assim como armazenar as informações de rastreamento que chegam em tempo real. Além disso, a base de dados deve ser estruturada de maneira a responder de forma eficiente as consultas, principalmente as consultas referentes ao monitoramento de veículos, permitindo que o usuário tenha a impressão de precisão por parte do sistema.

4.4.2 Modelo de Casos de Uso

O Modelo de Casos de Uso são documentos de texto que descrevem os requisitos funcionais (comportamentais) através de um conjunto de cenários típicos do uso de um sistema (Cockburn 2001). As tabelas a seguir apresentam o Modelo de Casos de Uso da aplicação Bus Tracker.

Tabela 4.1: Caso de Uso: Criar Identificador de Usuário.

Escopo: aplicação Bus Tracker

Nível: objetivo do Sistema
Ator Principal: Sistema
Interessados e Interesses: - Sistema: identificar de maneira única o usuário. - Usuário: ser identificado pelo sistema.
Precondições: o Usuário instalou o módulo Cliente da aplicação em seu dispositivo móvel, mas ainda não o executou.
Garantia de Sucesso: o Sistema criou um identificador único para o usuário. O usuário recebeu seu identificador.
Cenário de Sucesso Principal: 1. Usuário inicia o módulo Cliente (Sistema) instalado em seu dispositivo móvel. 2. Sistema cria um identificador único para o usuário. 3. Sistema informa ao Usuário que ele foi identificado e o apresenta o identificador.
Extensões: 2a. O Sistema falha ao criar o identificador do Usuário 1. Sistema informa a ocorrência de uma falha. 2. Usuário finaliza a operação.

Tabela 4.2: Caso de Uso: Selecionar Linha.

Escopo: aplicação Bus Tracker
Nível: objetivo do Usuário
Ator Principal: Usuário
Interessados e Interesses: - Usuário: selecionar uma linha de transporte.
Precondições: o Sistema tem conhecimento das linhas de transporte, com seus itinerários e pontos de parada.
Garantia de Sucesso: o Usuário encontra e seleciona uma linha.
Cenário de Sucesso Principal: 1. Usuário informa ao Sistema que deseja buscar uma linha. 2. Sistema solicita ao Usuário que indique os critérios de busca. 3. Usuário insere o nome ou parte do nome da linha desejada. 4. Sistema busca por linhas que se enquadram no critério de busca do Usuário (nome ou parte do nome da linha). 5. Sistema apresenta o resultado. 6. Usuário repete os passos 2-4 até encontrar a linha desejada. 7. Usuário seleciona uma linha.
Extensões: *a. A qualquer momento, Usuário cancela a operação. 2a. Usuário seleciona um ponto no mapa para ser usado como critério na busca por linha. 1. Usuário informa ao Sistema que deseja selecionar um ponto no mapa da cidade. 2. Sistema apresenta o mapa da cidade. 3. Usuário seleciona um ponto no mapa. 4. Sistema questiona a distância máxima em relação ao ponto selecionado, para ser considerada na busca. 5. Usuário define o raio de distância máximo do ponto selecionado. 6. Usuário continua a busca (podendo inserir parte do nome da linha).

<p>2b. Usuário utiliza a própria posição como ponto de referência para a busca.</p> <ol style="list-style-type: none"> 1. Usuário informa ao Sistema que deseja usar sua posição como critério na busca. 2. Sistema tenta obter a posição do usuário, solicitando ao Sistema de Posicionamento. <ol style="list-style-type: none"> 2a. Sistema obtém a posição do Usuário. <ol style="list-style-type: none"> 1. Sistema recebe informação de posicionamento. 2. Sistema informa ao Usuário que obteve a posição dele, e questiona a distância máxima em relação a esta posição, para ser considerada na busca. 3. Usuário define o raio de distância máximo da sua posição. 4. Usuário continua a busca (podendo inserir parte do nome da linha). 2b. Sistema NÃO obtém a posição do Usuário. <ol style="list-style-type: none"> 1. Sistema é avisado, pelo Sistema de Posicionamento, que houve falha na obtenção da posição do Usuário. 2. Sistema informa ao Usuário o problema. 3. Usuário finaliza a operação. <p>4a. Sistema falha na busca por linhas.</p> <ol style="list-style-type: none"> 1. Sistema informa ao Usuário a falha na busca por linhas. <ol style="list-style-type: none"> 1a. Sistema tenta novamente a busca. <ol style="list-style-type: none"> 1. Usuário solicita ao Sistema que ele tente novamente. 2. Sistema tenta executar novamente a busca. 2a. Usuário interrompe a busca. <ol style="list-style-type: none"> 1. Usuário finaliza a operação. <p>6a. Usuário não encontra uma linha que satisfaça seus critérios.</p> <ol style="list-style-type: none"> 1. Usuário finaliza a operação.
--

Tabela 4.3: Caso de Uso: Rastrear Veículo.

Escopo: aplicação Bus Tracker
Nível: objetivo do usuário
Ator Principal: Usuário
Interessados e Interesses: - Usuário: visualizar em tempo real a posição de um veículo de uma determinada linha.
Precondições: o Sistema tem conhecimento das informações de posicionamento dos veículos de transporte da linha selecionada.
Garantia de Sucesso: o Usuário visualiza em um mapa a posição, em tempo real, do veículo de uma linha por ele escolhida.
Cenário de Sucesso Principal: <ol style="list-style-type: none"> 1. Usuário invoca <u>Selecionar Linha</u> a fim de encontrar a linha desejada. 2. Sistema e Usuário interagem até uma linha ser selecionada. 3. Usuário informa ao Sistema que deseja rastrear os veículos desta linha. 4. Sistema solicita que o Usuário informe algumas configurações necessárias ao rastreamento como: o período de atualização da posição dos veículos no mapa, se deve ser apresentado o itinerário e/ou os pontos de parada da linha. 5. Sistema busca informações de rastreamento. 6. Sistema apresenta um mapa com a posição real do ou dos veículos daquela linha, conforme configurações escolhidas pelo Usuário. 7. Sistema repete os passos 5 e 6, atualizando a posição dos veículos conforme a taxa de atualização, até que o Usuário cancele a operação.
Extensões:

<p>*a. A qualquer momento, Usuário cancela a operação.</p> <p>2a. Usuário não encontra a linha desejada.</p> <ol style="list-style-type: none"> 1. Usuário finaliza a operação. <p>5a. Sistema falha na busca por informações de rastreamento.</p> <ol style="list-style-type: none"> 1. Sistema tenta novamente buscar informações de rastreamento. 2. Sistema repete o passo 1 em caso de falhas. <p>2a. Sistema obtém sucesso na busca por informações de rastreamento.</p> <ol style="list-style-type: none"> 1. Sistema prossegue o rastreamento buscando e apresentando as informações ao Usuário. <p>2b. Sistema falha em 3 tentativas consecutivas.</p> <ol style="list-style-type: none"> 1. Sistema informa ao Usuário falha na busca de informações de rastreamento. 2. Usuário finaliza operação.

Tabela 4.4: Caso de Uso: Cadastrar Evento Espacial.

Escopo: aplicação Bus Tracker
Nível: objetivo do usuário
Ator Principal: Usuário
<p>Interessados e Interesses:</p> <ul style="list-style-type: none"> - Sistema: armazenar os eventos cadastrados pelo Usuário de forma que eles possam ser monitorados e facilmente identificados quando da sua ocorrência. - Usuário: cadastrar um evento espacial pelo qual deseja ser avisado pelo Sistema.
<p>Garantia de Sucesso: o Usuário cadastra um evento espacial pelo qual deseja ser avisado quando da sua ocorrência. O Sistema armazena o evento cadastrado pelo Usuário e monitora a ocorrência deste evento.</p>
<p>Cenário de Sucesso Principal:</p> <ol style="list-style-type: none"> 1. Usuário informa ao Sistema que deseja cadastrar um evento. 2. Sistema apresenta ao Usuário os possíveis eventos espaciais a serem cadastrados. 3. Usuário seleciona o tipo de evento que deseja cadastrar. 4. Sistema apresenta ao Usuário opções de configuração conforme o tipo de evento selecionado. 5. Usuário entra com as configurações necessárias para o cadastro do evento. 6. Sistema armazena as informações de configuração do evento. 7. Sistema invoca <u>Monitorar Evento</u>. 8. Sistema informa ao Usuário que o evento foi cadastrado. 9. Usuário finaliza a operação.
<p>Extensões:</p> <p>*a. A qualquer momento, Usuário cancela a operação.</p> <p>6a. Sistema falha ao armazenar as informações de configuração do evento.</p> <ol style="list-style-type: none"> 1. Sistema informa ao Usuário que houve uma falha e que o evento não pode ser cadastrado. 2. Usuário finaliza a operação.

Tabela 4.5: Caso de Uso: Monitorar Evento.

Escopo: aplicação Bus Tracker
Nível: objetivo do Sistema
Ator Principal: Sistema
Interessados e Interesses:

- Sistema: monitorar e identificar a ocorrência de eventos espaciais cadastrados.
Precondições: existe pelo menos um evento cadastrado pelo Usuário.
Garantia de Sucesso: o Sistema identifica a ocorrência de um evento espacial cadastrado pelo Usuário e o avisa da ocorrência.
Cenário de Sucesso Principal: <ol style="list-style-type: none"> 1. Sistema busca informações de eventos cadastrados pelo usuário. 2. Sistema verifica se os requisitos que definem a ocorrência de um evento foram atingidos. 3. Sistema repete os passos 1 e 2 até que a ocorrência de um evento seja identificada. 4. Sistema identifica a ocorrência de um evento. 5. Sistema invoca <u>Alertar Ocorrência de Evento</u>. 6. Sistema repete passos 2-5 até que não existam mais eventos a serem monitorados.
Extensões: <ol style="list-style-type: none"> 1a. A busca por informações de eventos cadastrados falha. <ol style="list-style-type: none"> 1. Sistema informa a ocorrência de uma falha. 2. Sistema finaliza monitoração de eventos.

Tabela 4.6: Caso de Uso: Alertar Ocorrência de Evento.

Escopo: aplicação Bus Tracker
Nível: objetivo do Sistema
Ator Principal: Sistema
Interessados e Interesses: <ul style="list-style-type: none"> - Sistema: alertar ao Usuário a ocorrência de um evento espacial por ele cadastrado. - Usuário: ser alertado pelo Sistema sobre a ocorrência de um evento esperado.
Precondições: o Sistema identificou a ocorrência de um evento espacial cadastrado. O Usuário pode, de alguma forma, ser alertado pelo Sistema.
Garantia de Sucesso: o Sistema informa ao Usuário a ocorrência de um evento espacial pré cadastrado.
Cenário de Sucesso Principal: <ol style="list-style-type: none"> 1. Sistema busca informações de como o Usuário deseja ser informado sobre a ocorrência do evento. 2. Sistema informa o Usuário da ocorrência do evento. 3. Usuário finaliza a operação.

4.5 Elaboração – 2ª Fase do PU

A elaboração é a fase inicial de iterações onde, em um projeto, a arquitetura central e de alto risco do software é definida e implementada. A maioria dos requisitos são descobertos e estabilizados nesta fase, assim como os principais riscos são mitigados. É nesta fase que alguns dos principais artefatos do PU (Modelo de Domínio, Modelo de Projeto, Modelo da Dados, Documento de Arquitetura de Software) são iniciados. As próximas seções apresentam alguns destes artefatos e suas respectivas definições no contexto de projeto da aplicação Bus Tracker.

4.5.1 Modelo de Domínio

O artefato do PU denominado de Modelo de Domínio (também chamado de Modelo Conceitual) tem como objetivo representar, de maneira visual, classes conceituais, ou

objetos do mundo real, em um domínio. Ele é considerado o modelo mais importante na análise orientada a objetos, além de servir com fonte de inspiração no projeto de alguns objetos de software, e de ponto de entrada para vários outros artefatos do PU (Fowler, 1997). O Modelo de Domínio pode ser ilustrado com um conjunto de diagramas de classe (UML) em que nenhuma operação é definida. Apenas objetos do domínio (classes conceituais), seus atributos e associações entre estes objetos devem aparecer no Modelo de Domínio. A seguir é apresentado um esboço inicial para o Modelo de Domínio da aplicação Bus Tracker (Figura 4.1).

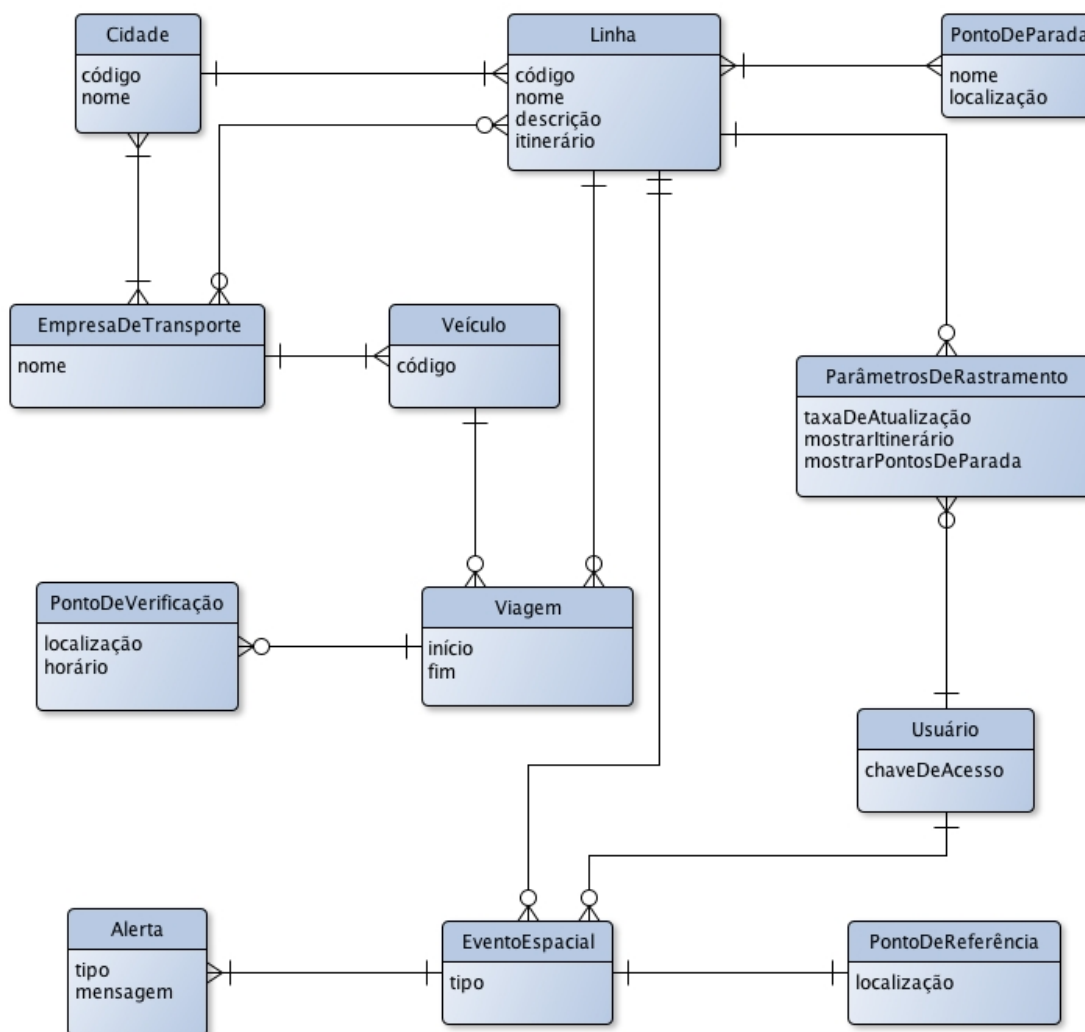


Figura 4.1: Modelo de Domínio.

4.5.2 Diagrama de Sequência do Sistema

Durante a interação entre um ator externo e sistema, eventos de sistema são gerados, geralmente solicitando alguma operação de sistema. Os casos de uso do sistema descrevem como estes atores externos interagem com o sistema por meio de solicitações, enquanto os Diagramas de Sequência do Sistema (DSS) explicitam estas solicitações. Um DSS mostra, para um cenário específico de um caso de uso, os eventos que os atores externos geram, sua ordem e os eventos entre sistemas. A ênfase do DSS está nos eventos que cruzam a fronteira do sistema. As figuras a seguir apresentam os DSSs da aplicação.

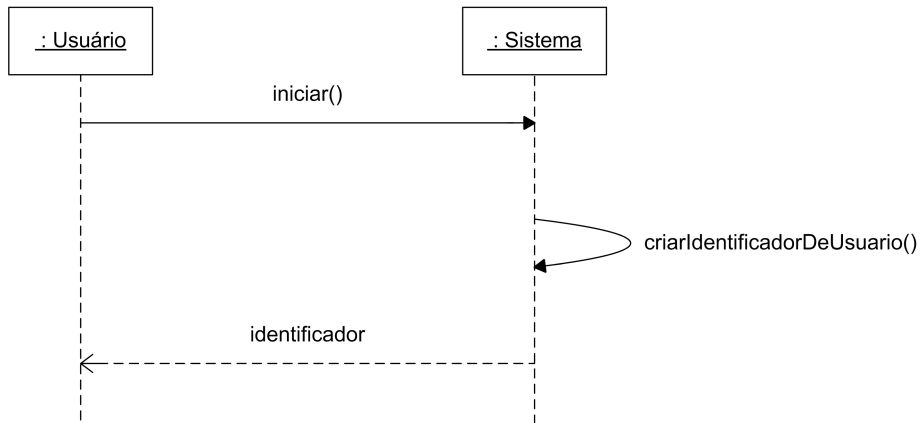


Figura 4.2: DSS para o caso de uso Criar Identificador de Usuário.

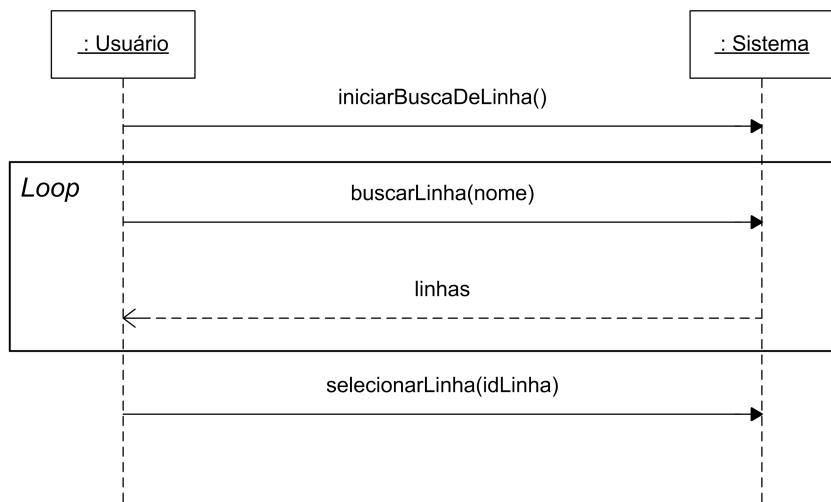


Figura 4.3: DSS para o cenário principal do caso de uso Selecionar Linha

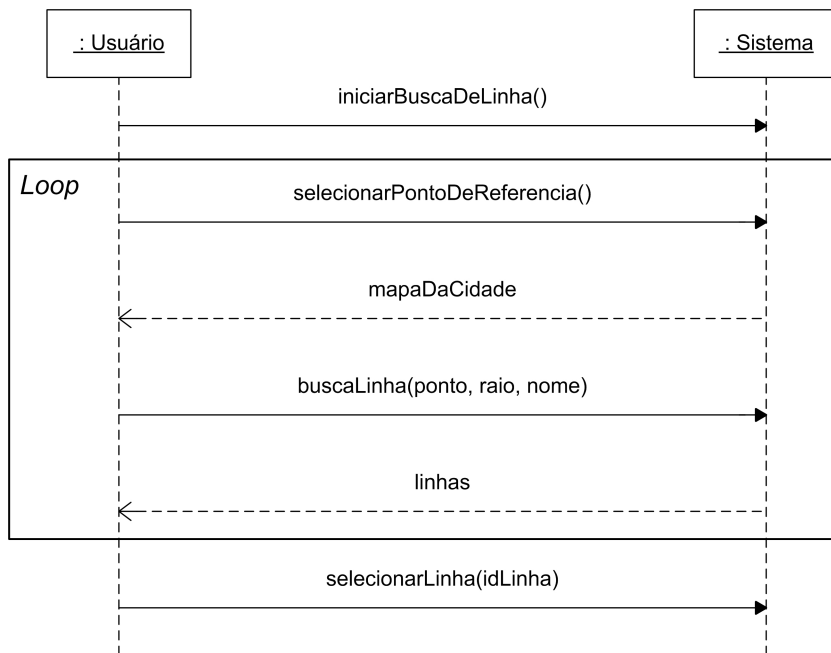


Figura 4.4: DSS para um cenário alternativo do caso de uso Selecionar Linha.

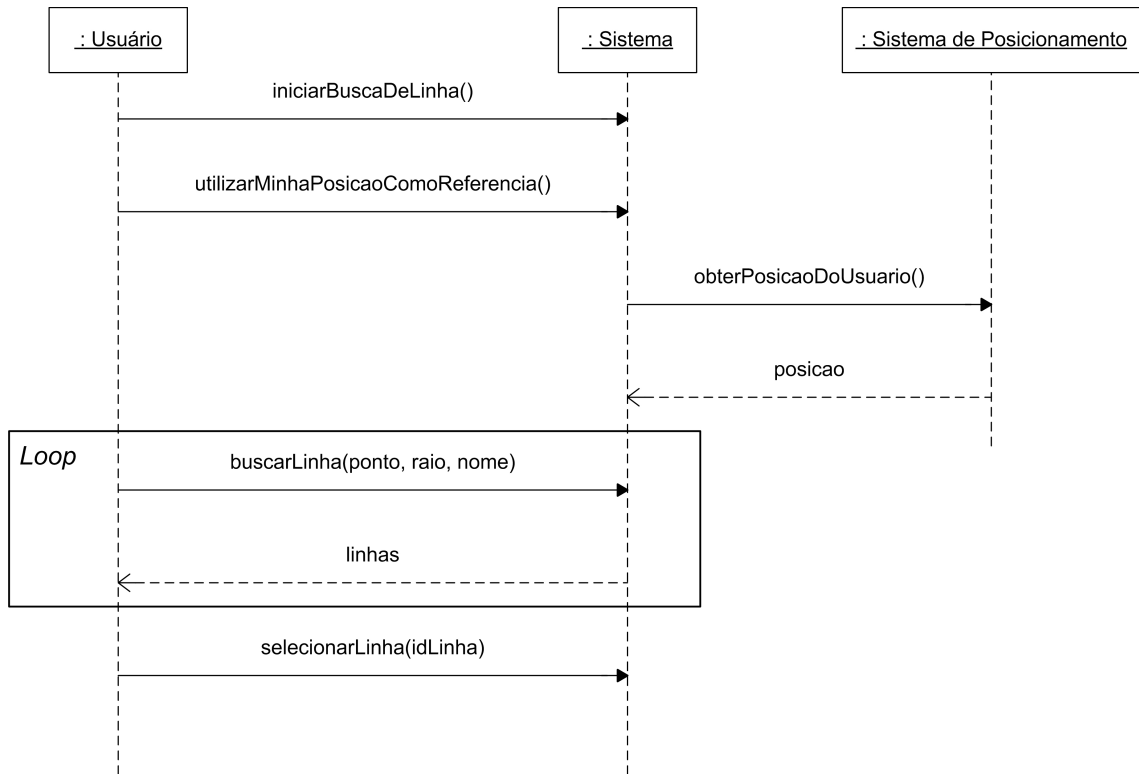
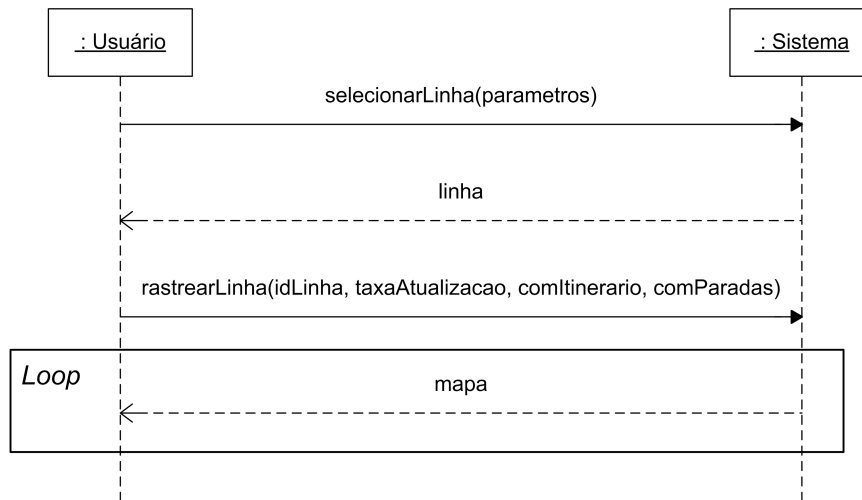


Figura 4.5: DSS para um cenário alternativo do caso de uso Selecionar Linha.



A operação de sistema rastrearLinha representa a interação ente Usuário e Sistema descrita pelo Caso de Uso Selecionar Linha.

Figura 4.6: DSS para o cenário principal do caso de uso Rastrear Veículo.

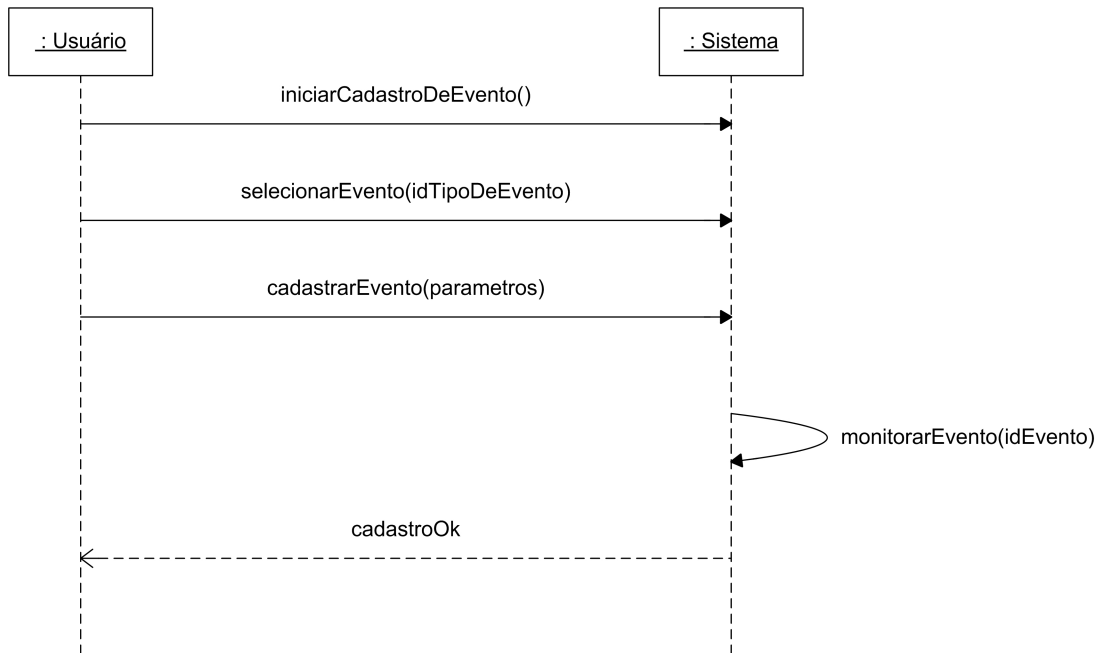


Figura 4.7: DSS para o cenário principal do caso de uso Cadastrar Evento Espacial.

4.5.3 Contrato de Operação

Contratos de Operação (CO) utilizam uma forma com seções de pós e pré-condições para descrever modificações em objetos do modelo de domínio, como resultado de uma operação do sistema. Desta forma os COs oferecem uma descrição mais detalhada e precisa do comportamento do sistema. A seção de pós-condições descreve modificações no estado dos objetos do Modelo de Domínio, tais como: a criação de instâncias, a modificação de atributos ou a formação de associações. As tabelas a seguir apresentam os Contratos de Operação para as operações do sistema que de alguma forma modificam o estado dos objetos do Modelo de Domínio da aplicação.

Tabela 4.7: Contrato de Operação: criarIdentificadorDeUsuario.

Operação: criarIdentificadorDeUsuario
Referências Cruzadas: Casos de Uso: Criar Identificador de Usuário
Pré-condições: O módulo Cliente instalado no dispositivo móvel do cliente nunca foi executado.
Pós-condições: - Foi criada uma instância de Usuário.

Tabela 4.8: Contrato de Operação: rastrearLinha.

Operação: rastrearLinha(idLinha, taxaAtualizacao, comItinerario, comParadas)
Referências Cruzadas: Casos de Uso: Rastrear Veículo
Pré-condições: O Usuário selecionou uma linha a partir da operação selecionarLinha.
Pós-condições: - Foi criada uma instância <i>params</i> de ParametrosDeRastreamento. - Os atributos de <i>params</i> foram iniciados: - <i>params.taxaAtualizacao</i> tornou-se taxaAtualizacao; - <i>params.mostrarItinerario</i> tornou-se comItinerario; - <i>params.mostrarParadas</i> tornou-se comParadas.

- *params* foi associado com o Usuário corrente.
- *params* foi associado com a Linha correspondente a idLinha

Tabela 4.9: Contrato de Operação: cadastrarEvento.

Operação: cadastrarEvento(params)
Referências Cruzadas: Casos de Uso: Cadastrar Evento Espacial
Precondições: O usuário selecionou o tipo de evento a ser cadastrado.
Pós-condições: <ul style="list-style-type: none"> - Foi criada uma instância <i>event</i> de EventoEspacial. - <i>event</i> foi associado com o tipo de evento selecionado pelo Usuário. - <i>event</i> foi associado com o Usuário corrente. - Foi criada um instância <i>alert</i> de Alerta. - <i>alert</i> foi associado com o TipoDeAlerta especificado em parâmetros. - <i>alert</i> foi associado a <i>event</i>. - Foram feitas todas as demais associações de <i>event</i> com outros objetos (Viagem, HorárioDeViagem, PontoDeReferencia) conforme o tipo do evento e os valores recebidos em parâmetros.

4.5.4 Projeto do Banco de Dados

A desenvolvimento de uma aplicação que utilize uma base de dados é uma tarefa complexa, que envolve, entre outras coisas, o projeto do esquema do banco de dados. As necessidades dos usuários da futura aplicação desempenham um papel fundamental no processo de projetar tal esquema. Nesta seção será apresentado o projeto do banco de dados da aplicação Bus Tracker, o qual será desenvolvido utilizando técnicas de projeto já bastante difundidas. Tais técnicas podem ser encontradas em (Silberschatz, Korth e Sudarshan, 2006) ou em (Heuser, 2009), além de uma infinidade de livros que abordam esse assunto.

O projeto de um banco de dados completo, que atenda as necessidades da aplicação que utilizará esse banco de dados, requer atenção à um amplo conjunto de aspectos que influenciam uma variedade de escolhas de projeto em diferentes níveis. Para tanto o projeto é dividido em diferentes fases:

- na fase inicial do projeto de banco de dados devem ser caracterizadas as necessidades de dados dos prováveis usuários do banco de dados. O projetista deve interagir com especialistas do domínio para obter tais informações. Esta primeira fase já foi aqui realizada nas seções anteriores deste capítulo e seus resultados foram apresentados na forma de artefatos da Análise de Projeto Orientado a Objetos. Artefatos como Visão, Modelo de Casos de Uso e Modelo de Domínio são excelentes fontes de informação para o projeto de banco de dados;
- na segunda fase o projetista escolhe um modelo de dados e, aplicando os conceitos desse modelo de dados, traduz as necessidades levantadas na primeira fase para um esquema conceitual de banco de dados. O modelo conceitual especifica as entidades que serão representadas no banco de dados, os atributos destas entidades, os relacionamentos entre elas e as restrições sobre as entidades. No projeto do banco de dados da aplicação Bus Tracker será utilizado o modelo de dados Entidade-Relacionamento, resultando em um diagrama de Entidade-Relacionamento (ER) que fornece uma representação gráfica do esquema.

Muitas vezes esse diagrama se assemelha muito com o Modelo de Domínio, uma vez que muitas das classes conceituais do Modelo de Domínio deverão ser representadas no banco de dados;

- em uma terceira fase, o esquema conceitual resultante da segunda fase, é revisado para garantir que ele atende as necessidades funcionais. Por exemplo, operações que modificam ou excluem dados, ou ainda operações de pesquisam e recuperam dados específicos;
- a quarta fase é conhecida como fase de projeto lógico, e representa o início da transição de um modelo de dados abstrato (conceitual) para o modelo de dados de implementação. O modelo de implementação normalmente é um modelo de dados Relacional, logo, nesta fase é feito um mapeamento do esquema conceitual (ER) para um esquema relacional;
- por fim, na fase de projeto físico, os recursos físicos do banco de dados (tabelas, colunas, índices, etc) são especificados.

Seguindo as fases de projeto descritas acima, as próximas seções apresentam os modelos conceitual e lógico do banco de dados da aplicação Bus Tracker.

4.5.5 Modelo Conceitual do Banco de Dados

Nesta seção é apresentado o diagrama Entidade-Relacionamento obtido após o projeto conceitual do banco de dados da aplicação Bus Tracker. Nesta fase, os artefatos resultantes da análise orientada a objetos (Visão, Modelo de Casos de Uso e Modelo de Domínio) serviram de fonte para que as necessidades do sistema, quanto ao uso do banco de dados, fossem identificadas e modeladas corretamente.

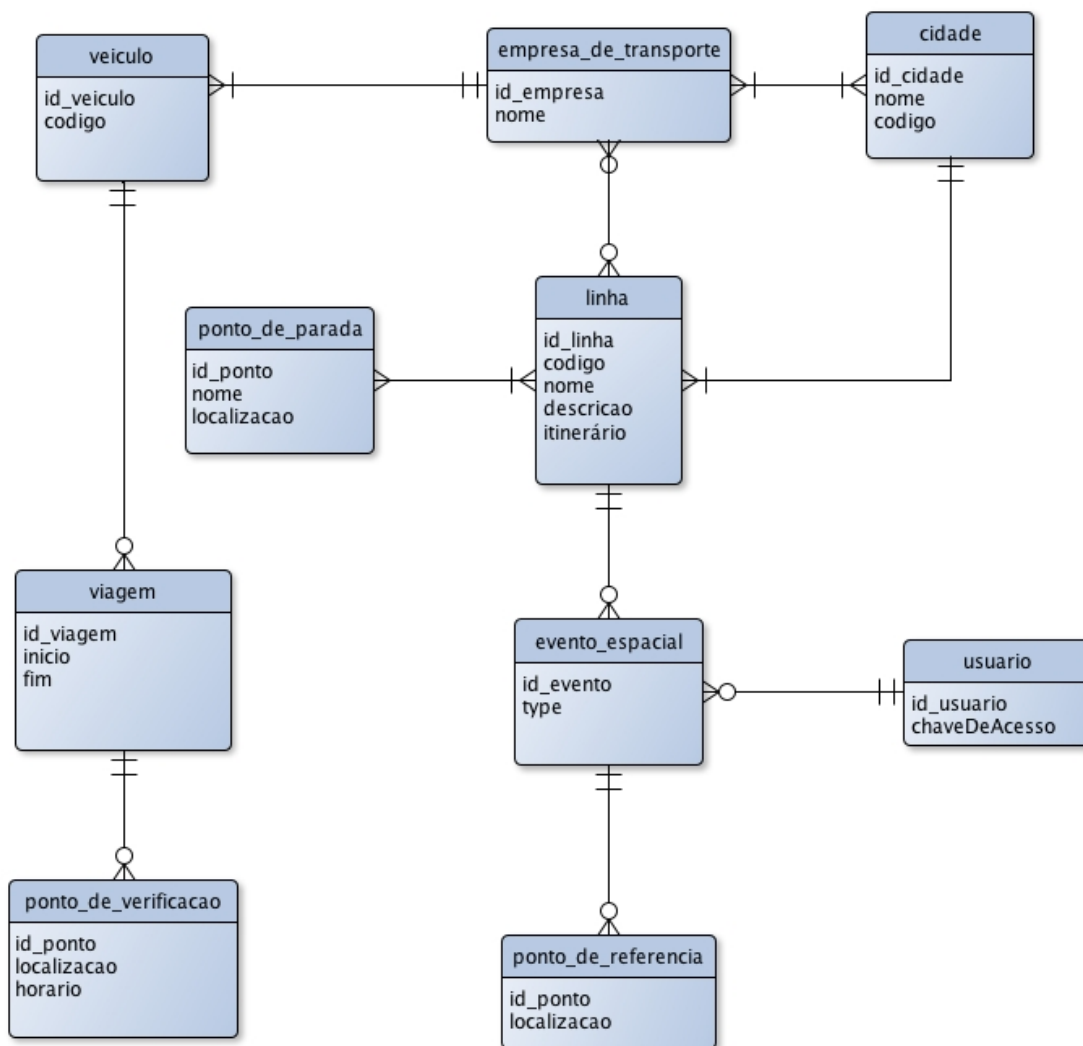


Figura 4.8: Diagrama Entidade-Relacionamento da aplicação Bus Tracker.

4.5.6 Modelo Lógico do Banco de Dados

O modelo lógico de um banco de dados pode ser obtido a partir de um modelo conceitual. O modelo de dados relacional, o qual será utilizado aqui para representar o modelo de dados lógico da aplicação Bus Tracker, trás os detalhes de implementação do banco de dados levando em consideração um determinado Sistema de Gerenciamento de Banco de Dados (SGBD). Antes de dar início a transformação do modelo conceitual definido na seção 4.5.5 para o modelo lógico é necessário a escolha de um SGBD adequado as necessidades da aplicação.

4.5.6.1 Escolhendo um Sistema de Gerenciamento de Banco de Dados

A escolha de um SGBD adequado a determinada aplicação pode ser tanto uma tarefa simples quanto complicada. Os critérios adotados na escolha e os requisitos a serem atendidos pelo aplicação, no que diz respeito ao armazenamento e acesso de dados, servem como base para a escolha de um SGBD. No caso da aplicação Bus Tracker, a escolha teve como critério inicial a utilização de um SGBD gratuito. É grande o número de bons SGBDs disponíveis no mercado, alguns deles são pagos e bastante difundidos como é o caso do produto desenvolvido pela empresa Oracle, enquanto que o

PostgreSQL e MySQL estão entre os SGBDs gratuitos mais utilizados em aplicações comerciais.

A aplicação Bus Tracker tem um diferencial com relação aos dados que devem ser armazenados e a forma de tratamento destes dados. É exatamente esse diferencial que vai guiar a escolha de um SGBD adequado as necessidades da aplicação. Grande parte dos dados gerados e consumidos pela aplicação pertencem ao domínio geométrico, ou seja, são pontos no espaço (paradas de ônibus, localização do usuário), retas, polígonos (itinerários), os quais devem ser acessados e manipulados de forma eficiente para fornecer uma melhor experiência ao usuário. Muitos desses dados devem sofrer algum tipo de manipulação especial, totalmente relacionada ao domínio desses dados, como por exemplo: verificar quais linhas de ônibus tem algum ponto de parada próximo a localização do usuário. Utilizando este importante requisito e a preferência por um SGBD gratuito a escolha tornasse simples, pois o PostgreSQL traz a implementação nativa de inúmeros tipos de dados geométricos (ponto, reta, polígono, círculo, retângulo, etc), assim como uma gama de funções e operadores para manipulação desses dados. Mais informações sobre a implementação de dados e operadores geométricos presente no PostgreSQL podem ser encontradas em sua documentação (PostgreSQL).

4.5.6.2 Projeto Lógico

Tendo em mãos o SGBD que será utilizado na implementação do banco de dados da aplicação podemos dar continuidade ao projeto lógico. Como foi mencionado, o modelo lógico será obtido a partir do modelo conceitual (diagrama ER) seguindo algumas regras de transformação que levam em consideração objetivos como boa performance no acesso aos dados, simplicidade no desenvolvimento da aplicação que irá utilizar os dados, entre outros. Uma definição destas regras pode ser encontradas em (Heuser 2009).

Com a finalidade de simplificar a codificação da aplicação e de padronizar a nomenclatura a ser utilizada, todas as entidades e atributos apresentados no diagrama ER tiveram seus nomes adaptados para a língua inglesa à medida em que o Diagrama Relacional foi construído. A figura 4.9 apresenta a primeira versão do modelo lógico obtido a partir do diagrama ER da figura 4.8. Essa primeira versão pode sofrer pequenas adaptações conforme os módulos servidor e cliente forem desenvolvidos, o que não significa que o modelo foi concebido de maneira errônea. Pequenas alterações e ajustes são inerentes ao desenvolvimento iterativo, pois, a cada iteração, pequenos módulos da aplicação são desenvolvidos e testados, dando um retorno imediato de quais decisões foram acertadas e de quais devem ser revistas.

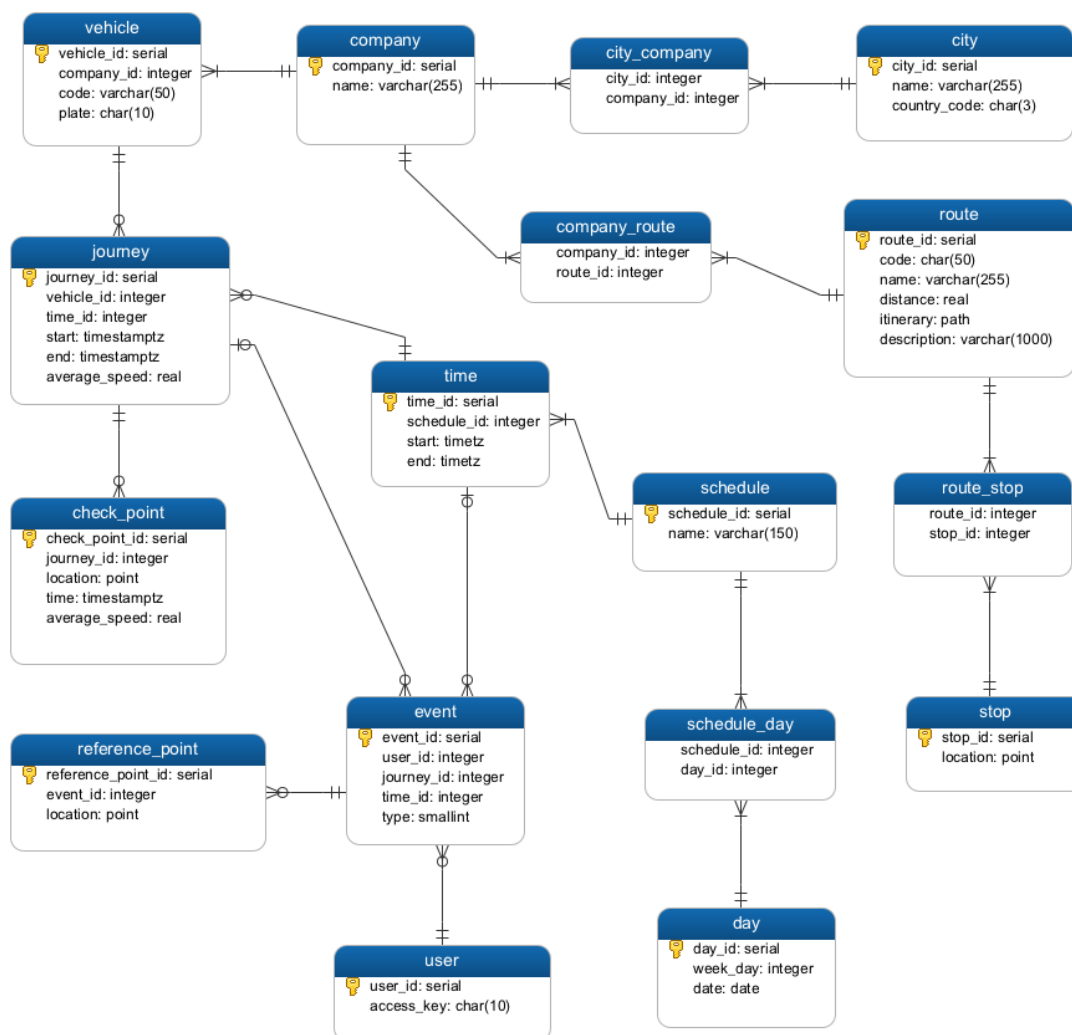


Figura 4.9: Modelo de Dados Relacional da aplicação Bus Tracker.

4.6 Considerações finais

Este capítulo chega ao fim, tendo apresentado a definição, a análise e o projeto da aplicação Bus Tracker. Cada uma destas etapas faz parte daquilo que definimos no início deste mesmo capítulo como Engenharia de Software. Os artefatos resultantes deste processo servirão agora como uma receita (um guia a ser seguido) para que a aplicação Bus Tracker possa ser implementada e avaliada segundo seus requisitos.

O capítulo a seguir, Prototipação, apresenta uma implementação parcial do projeto aqui definido, com o intuito de permitir que as principais funções da aplicação Bus Tracker sejam testadas e avaliadas quanto à sua viabilidade de implementação em condições reais de uso.

5 PROTOTIPAÇÃO

Este capítulo apresenta a prototipação da aplicação Bus Tracker, as tecnologias escolhidas para desenvolvimento do protótipo, o porque de tais escolhas, assim como a validação do protótipo.

5.1 Visão Geral

O objeto da prototipação aqui apresentada não é implementar todos os requisitos definidos, analisados e projetados no capítulo anterior, mas sim, implementar uma aplicação que permita avaliar as principais funcionalidades da aplicação Bus Tracker. Os casos de uso, apresentados durante o projeto, não serão completamente implementados, mas servirão como base para garantir que as funcionalidades principais da aplicação sejam implementadas. Ao final, será possível avaliar os resultados obtidos e identificar possíveis obstáculos para uma implementação real da aplicação Bus Tracker.

Antes de dar início a prototipação é necessário definir quais das funcionalidades da aplicação Bus Tracker serão implementadas. Ou seja, é necessário estabelecer quais são as principais funcionalidades da aplicação e como estas funcionalidades serão prototipadas. O artefato Visão (apresentado durante a análise) nos traz, em linhas gerais, o propósito funcional da aplicação. Tal propósito é aqui reproduzido novamente para que tenhamos uma diretriz a ser seguida durante o desenvolvimento do protótipo. Segundo o artefato Visão, a aplicação Bus Tracker é: uma aplicação para dispositivos móveis que permite aos usuários do transporte público, não apenas, acompanhar em tempo real o deslocamento dos veículos de transporte, como também obter serviços de alerta (alarmes, mensagens) à medida que determinados eventos (como a aproximação de determinado veículo de um ponto específico) são detectados pela aplicação. Considerando que um protótipo deve permitir avaliar algumas das principais funcionalidades de uma aplicação, definimos que o protótipo a ser implementado deve permitir ao usuário buscar uma linha dentre várias linhas existentes e a partir daí acompanhar a localização, em tempo real, dos veículos que realizam viagens para esta linha. Além disso, o usuário deve ser capaz de informar a aplicação que deseja ser notificado da aproximação do veículo de um ponto em questão. Em contrapartida a aplicação deve alertá-lo quando tal aproximação ocorrer.

Uma última questão, que necessita ser abordada antes que a prototipação seja apresentada, diz respeito a origem dos dados referentes ao domínio da aplicação, como linhas de ônibus, pontos de parada, itinerários, etc. Os dados aqui utilizados, são dados reais de linhas e de pontos de parada da cidade de Porto Alegre. Estes

dados foram obtidos junto ao site PoaTransporte (<http://www.poatransporte.com.br/>), o qual disponibiliza tais informações através da Internet. Este site é uma iniciativa da prefeitura de Porto Alegre, e sua implementação foi feita pela empresa pública Procempa (PROCEMPA) em parceria com a Empresa Pública de Transporte e Circulação (EPTC).

Tendo em mãos uma ideia geral de como o protótipo deve se comportar, de quais tarefas ele deve realizar e de que dados ele irá utilizar, podemos dar início a sua implementação.

5.2 Escolhendo uma linha de ônibus

Esta seção apresenta a prototipação da funcionalidade que permite ao usuário selecionar uma linha de ônibus e visualizar seu itinerário no mapa. Para tanto, o módulo cliente servirá como ponto de partida, pois ele é a interface direta com os usuários. É a partir deste módulo que o usuário poderá consultar linhas, acompanhar veículos em movimento e receber notificações sobre eventos relacionados com o deslocamento dos veículos. Antes de dar continuidade ao desenvolvimento, é necessário definir quais tecnologias serão utilizadas na implementação tanto do módulo cliente quanto do módulo servidor. Como podemos ver no capítulo 2, duas plataformas (Android e iOS) dominam grande parte do mercado de smartphones e tablets. Ambas as plataformas apresentam recursos suficientes (principalmente no que diz respeito a aplicações com base em localização) para a implementação das funcionalidades definidas para este protótipo. Como não existem restrições tecnológicas que impeçam a escolha de uma delas ou ainda das duas, a plataforma iOS foi a escolhida para implementação do módulo cliente. Já para o módulo servidor a tecnologia a ser utilizada é a linguagem de script PHP em conjunto com um dos seus mais famosos frameworks, o Yii Framework (Yii). Aqui a escolha também não segue restrições tecnológicas, pois tanto o PHP quanto qualquer uma das outras tecnologias citadas na seção Servidor do capítulo 2 (Ruby, Java, Python, etc) servem ao propósito desta aplicação. Já para a base de dados, seguiremos a escolha feita no final do capítulo 4, onde o PostgreSQL foi eleito (neste caso por questões tecnológicas referentes ao armazenamento e manipulação de dados espaciais) como o SGBD gratuito mais adequado para a implementação da aplicação Bus Tracker. Por questões de simplicidade o mesmo modelo relacional apresentado durante o projeto será implementado como modelo de dados do protótipo, mesmo que parte dele não venha a ser utilizado pela aplicação.

Após a definição das tecnologias a serem empregadas no desenvolvimento, é necessário definir o comportamento inicial da aplicação. Ou seja, como a aplicação irá se apresentar ao usuário quando for iniciada. Ao ser iniciado, o protótipo do módulo cliente apresenta um mapa com a localização atual do usuário (figura 5.1). A localização do usuário será considerada como um ponto de referência durante o funcionamento do protótipo, com a finalidade de simplificar suas funcionalidades. Como pode-se ver na figura, a aplicação traz ainda uma barra na parte inferior com 3 botões: Stops, Track e Alert. É por meio destes botões que o usuário irá interagir com a aplicação para que esta realize suas principais funcionalidades.

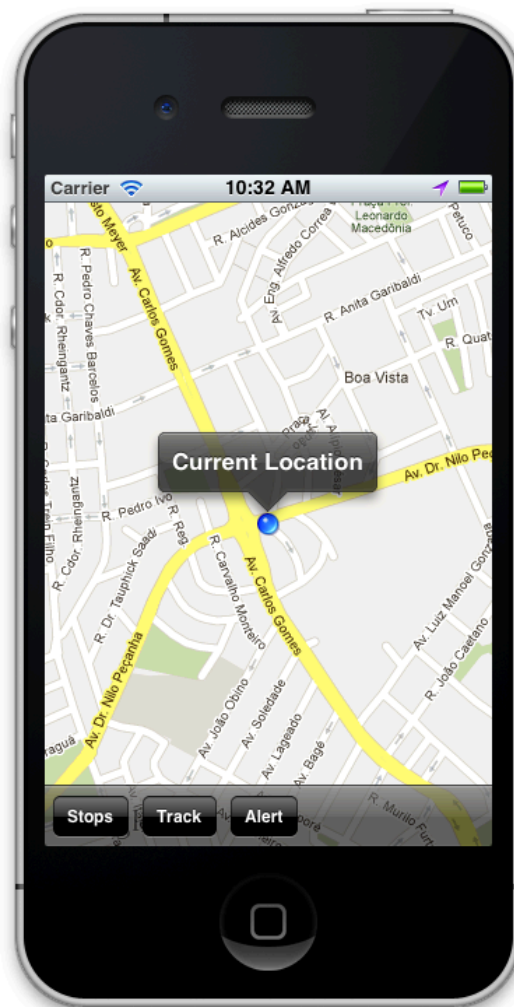


Figura 5.1: Módulo Cliente – Tela inicial.

O foco desta seção é a escolha de um linha de ônibus e a visualização de seu itinerário no mapa. E o caminho para se atingir tal objetivo passa pelos pontos de parada. Ou seja, quando o usuário clicar no botão “Stops”, a aplicação solicitará ao módulo servidor uma lista de pontos de parada com as respectivas linhas que passam nestes pontos. Para que o resultado desta consulta não sejam todos os pontos de parada presentes em Porto Alegre, a localização do usuário será enviada ao servidor, o qual irá utilizar tal localização na busca pelo pontos. O resultado de tal busca é uma lista com as paradas mais próximas desta localização. Sendo que a distância máxima entre a localização do usuário e uma parada deverá ser de aproximadamente 500 metros.

No momento em que o usuário pressiona o botão “Stops”, uma requisição é feita ao módulo servidor, enviando a localização do usuário. A requisição é tratada pelo servidor (PHP), o qual consulta na base de dados por pontos de parada próximos a localização recebida. O tratamento desta requisição é extremamente simples, devido a escolha de um SGBD que, além de armazenar dados espaciais, provê inúmeras funções e operadores para lidar com estes dados. Isso significa que a tarefa de localizar pontos de ônibus próximos de uma localização pode ser realizada com uma simples consulta na base de dados. A consulta SQL a seguir representa uma possível

busca por pontos de parada próximos da localização (-30.029383,-51.180153), a qual serve como ponto central de um círculo, dentro do qual a localização da parada deve estar.

```
SELECT * FROM "stop" "t" WHERE location <@ circle '(-30.029383,-51.180153),0.003000>'
```

Não apenas os pontos de paradas devem ser retornados ao módulo cliente, mas as linhas que utilizam tais pontos também. Permitindo assim, que o usuário visualize tais linhas ao selecionar uma parada. O resultado da requisição feita pelo módulo cliente será um JavaScript Object Notation (JSON) representando uma lista de paradas com suas respectivas linhas, conforme exemplo a seguir.

```
[
  {
    "code": "2924",
    "name": null,
    "location": {
      "lat": -30.028737,
      "long": -51.178135
    },
    "routes": [
      {
        "code": "4302-2",
        "name": "BELA VISTA/ANITA VIA CARAZINHO/IJUÍ"
      },
      {
        "code": "T7-2",
        "name": "NILO/PRAIA DE BELAS"
      }
    ]
  },
  {
    "code": "2917",
    "name": null,
    "location": {
      "lat": -30.029481,
      "long": -51.180779
    },
    "routes": [
      {
        "code": "430-2",
        "name": "BELA VISTA / ANITA"
      },
      {
        "code": "429-2",
        "name": "PROTASIO / IGUATEMI"
      },
      {
        "code": "4291-2",
        "name": "PROTASIO/IGUATEMI/CANANEIA"
      }
    ]
  }
]
```

Ao receber o resultado de sua requisição o módulo cliente irá apresentar na tela os pontos onde estão as paradas mais próximas do usuário. A figura a seguir apresenta o resultado desta requisição para a posição do usuário demonstrada na figura 5.1. As

paradas são representadas por pinos verdes, que ao serem tocados apresentam uma opção para visualização das linhas que utilizam aquela ponto de parada.

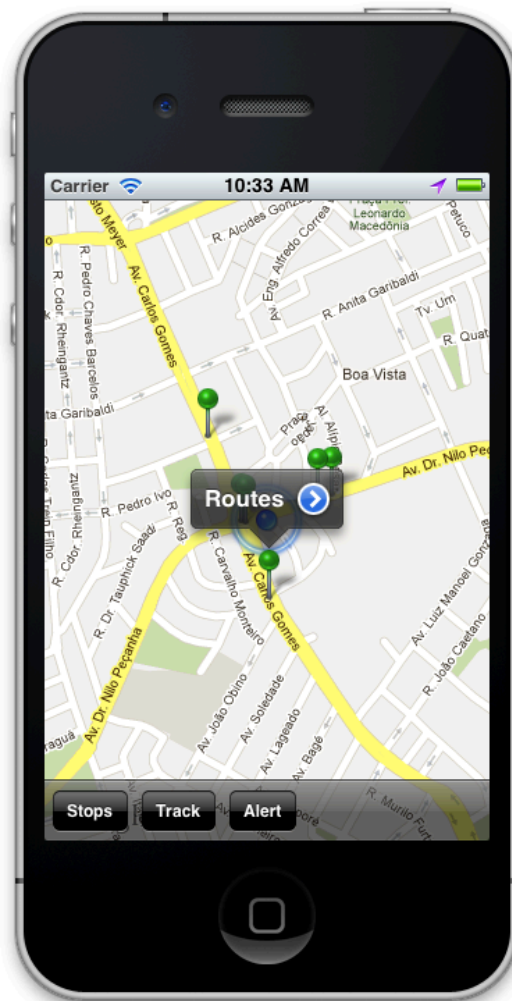


Figura 5.2: Módulo Cliente – Pontos de parada próximos ao usuário.

No momento em que o usuário solicita visualizar as linhas a partir de uma parada, o módulo cliente apresenta uma nova tela onde as linhas que utilizam o ponto de parada em questão são apresentadas (figura 5.3).

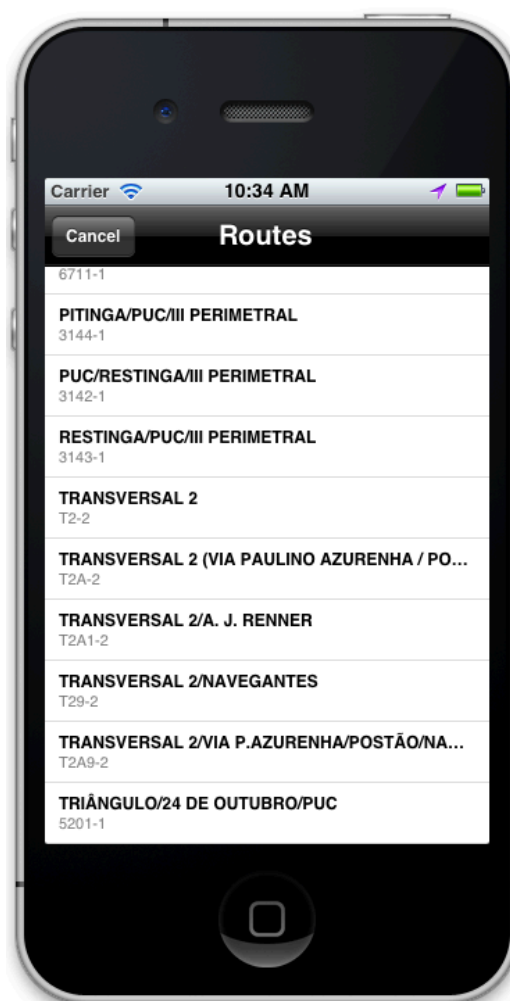


Figura 5.3: Módulo Cliente – Lista de linhas de ônibus que utilizam o ponto de parada selecionado pelo usuário.

A partir da lista de linhas apresentada na figura anterior, duas ações podem ser tomadas pelo usuário. Uma é retornar ao mapa pressionando o botão “Cancel” na barra de navegação localizada no topo da tela. Esta ação não tem nenhum resultado sobre as informações apresentadas no mapa, ou seja, a aplicação retorna ao mesmo contexto em que se encontrava antes do usuário solicitar a visualização das linhas de um ponto de parada. A segunda ação permitida ao usuário é selecionar uma das linhas da lista. Ao fazer isso o usuário atinge o objetivo desta seção que era a seleção de uma linha de ônibus para visualização de seu itinerário. A figura a seguir apresenta o itinerário da linha “5201-1 TRIÂNGULO/24 DE OUTUBRO/PUC”, a qual aparece como última na lista de linhas da figura anterior. A porção do mapa visível ao usuário foi alterada para que todo o itinerário da linha ficasse visível.

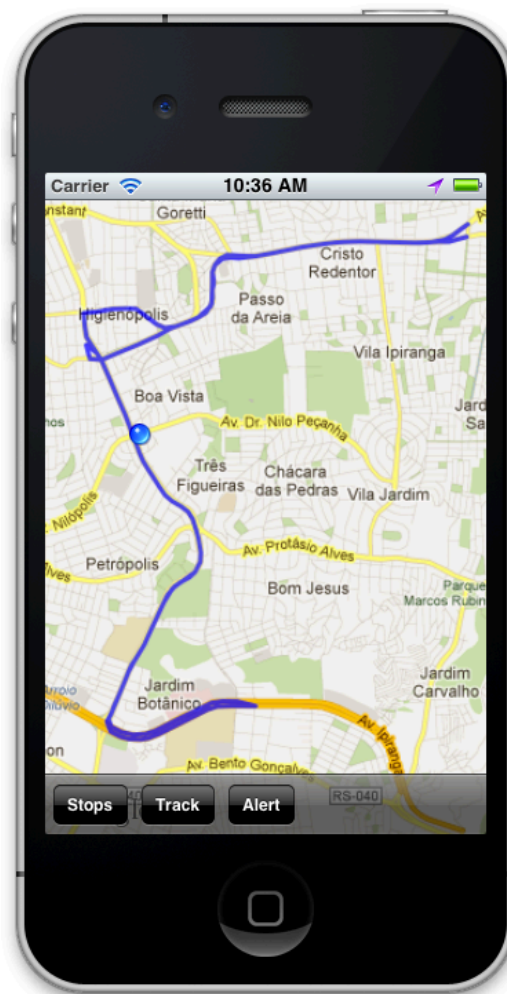


Figura 5.4: Módulo Cliente – Itinerário da linha “5201-1 TRIÂNGULO/24 DE OUTUBOR/PUC”.

5.3 Acompanhando veículos em tempo real

A segunda funcionalidade a ser prototipada permite que o usuário visualize no mapa a localização dos veículos que realizam viagens para uma determinada linha. A grande desafio aqui, é a obtenção de tais informações. Uma vez que não é objetivo da aplicação Bus Tracker definir como deve ser a infraestrutura de rastreamento, nem como as informações de rastreamento chegam até a base de dados, em tempo real, poderemos utilizar alguns artifícios para simular tais informações. Num primeiro momento estas informações serão geradas por um simples script executado junto ao módulo servidor que, em intervalos de tempo constante produz, dados de rastreamento (supostas posições de veículos) para uma linha de ônibus específica. Numa segunda fase de avaliação, um pequeno módulo rastreador será instalado em outro dispositivo móvel. Esse módulo rastreador irá realizar a simples tarefa de a cada 10 segundos recuperar a localização do dispositivo e enviar ao módulo servidor para que seja salva. Com isso, será possível simular de forma mais realista a viagem de um ônibus e assim acompanhar em um outro dispositivo toda a movimentação dele sobre o mapa.

A visualização de veículos de uma determinada linha tem como ponto de partida o resultado apresentado na figura 5.4. Ou seja, o usuário já selecionou um linha e está visualizando no mapa seu itinerário. A partir deste cenário basta ao usuário pressionar o botão “Track” na barra inferior para que o módulo cliente inicie a tarefa de constantemente requisitar ao módulo servidor informações de localização de veículos desta linha. Quando o módulo servidor retornar alguma informação de rastreamento, ícones representando os veículos serão apresentados no mapa sobre o itinerário da linha, sendo a posição destes ícones constantemente atualizada à medida que novas informações são recebidas do servidor. Ao solicitar informações de rastreamento ao módulo servidor, o módulo cliente envia sempre o código da linha para a qual deseja tais informações. A convenção aqui adotada é que a linha selecionada pelo usuário, a qual tem seu itinerário desenhado sobre o mapa, será a linha de referência para requisitar e apresentar informações de rastreamento. Conforme podemos ver na figura 5.5, dois veículos realizam suas respectivas viagens para a linha selecionada anteriormente.

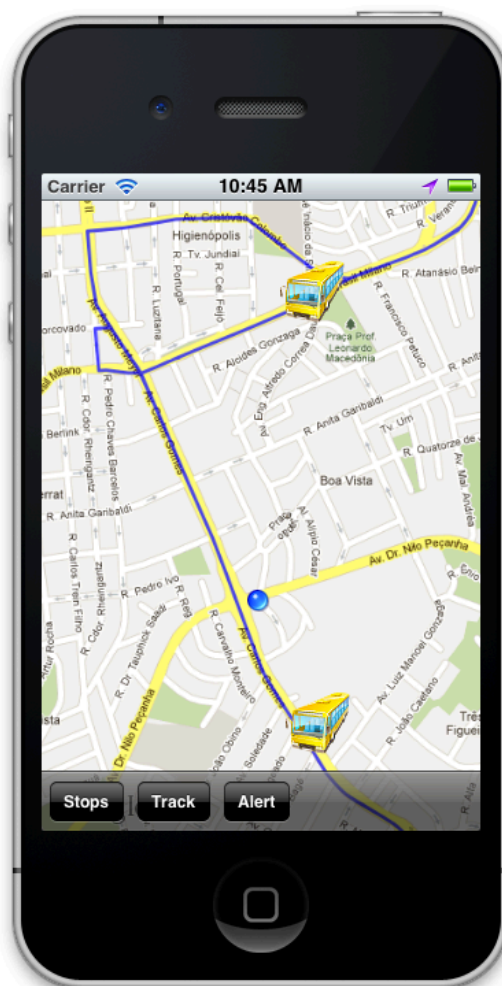


Figura 5.5: Módulo Cliente – Rastreamento veículos da linha (Posição 1).

5.4 Recebendo alertas de aproximação de veículo

A última funcionalidade a ser prototipada é talvez a mais interessante. A ideia aqui é permitir que o usuário seja alertado quando um ônibus da linha que ele selecionou se aproximar da sua localização. O propósito desta funcionalidade é validar umas das principais funcionalidades da aplicação Bus Tracker, onde, segundo o projeto, o usuário poderia cadastrar algum evento espacial pelo qual desejasse ser notificado quando da sua ocorrência. No protótipo, por simplicidade, o usuário não precisará cadastrar nenhum evento, pois o sistema assume a aproximação de um ônibus da localização atual do usuário como o evento a ser notificado. Partindo do cenário apresentado na seção anterior, onde o usuário solicitou acompanhar os veículos de uma linha em particular, basta que o usuário pressione o botão “Alert” na barra inferior da tela para que o sistema passe a monitorar a aproximação de algum ônibus da posição em que ele se encontra. À medida em que recebe do módulo servidor novas informações de rastreamento da linha que está sendo monitorada, o módulo cliente avalia se algum dos veículos que realizam viagem naquela linha se encontra a uma determinada distância (aproximadamente 1000 metros) ou ainda mais próximo do usuário. As figuras 5.6, 5.7 e 5.8 demonstram a aproximação de um veículo do usuário e o momento em que o usuário é alertado desta aproximação.

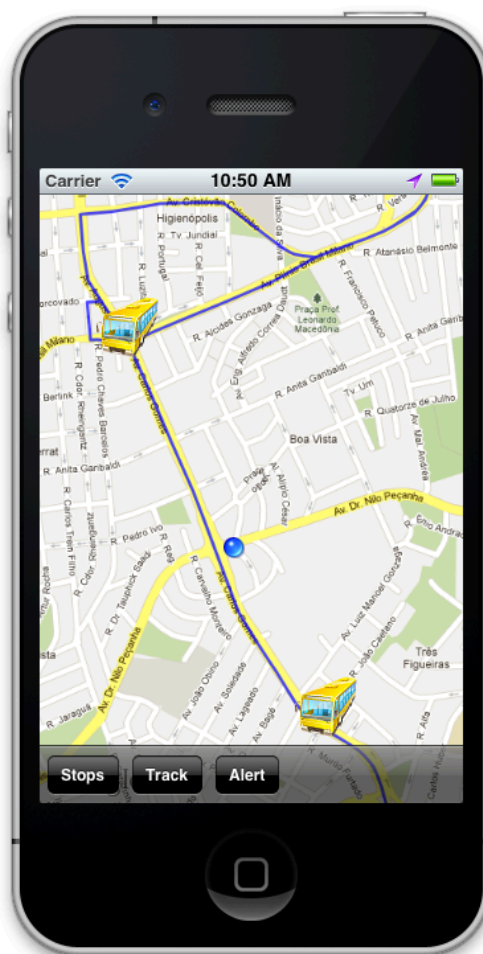


Figura 5.6: Módulo Cliente – Rastreamento veículos da linha (Posição 2).

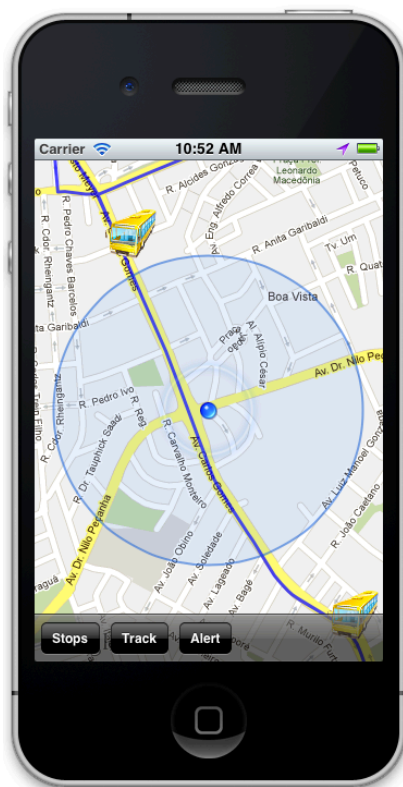


Figura 5.7: Módulo Cliente – Rastreamento veículos da linha (Posição 3).

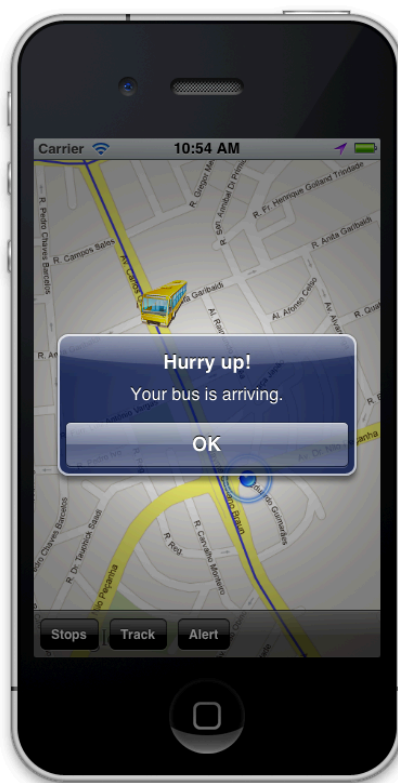


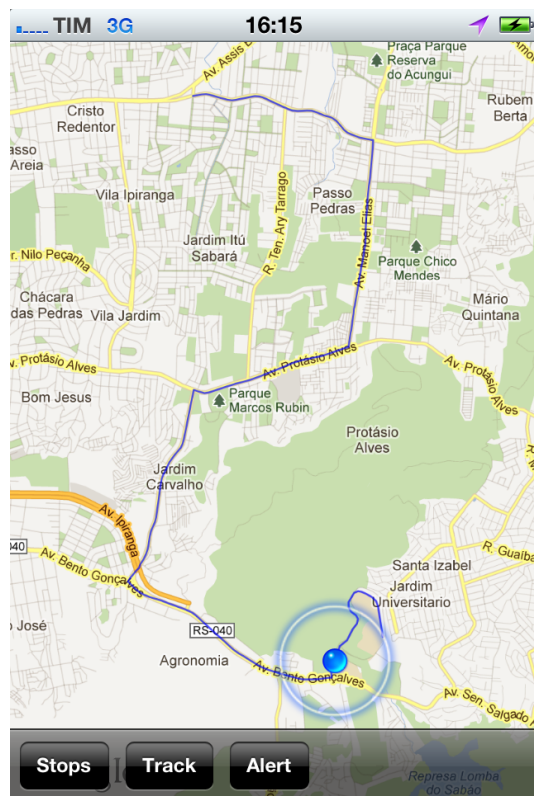
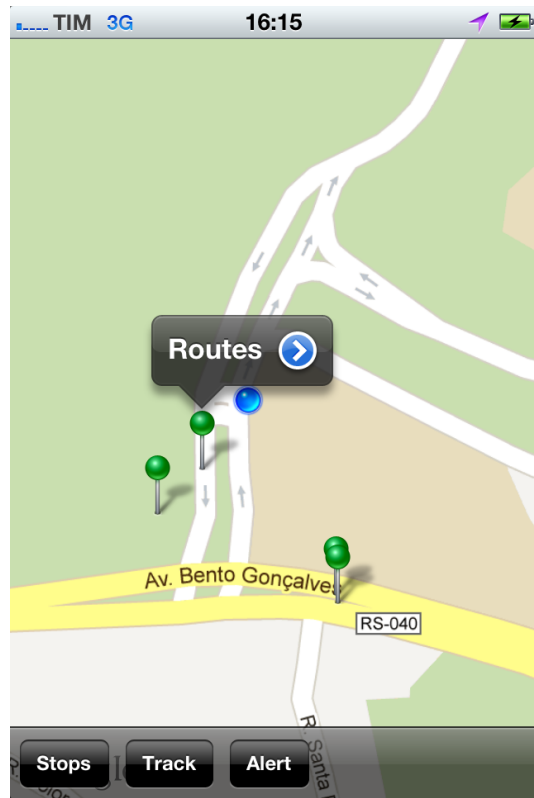
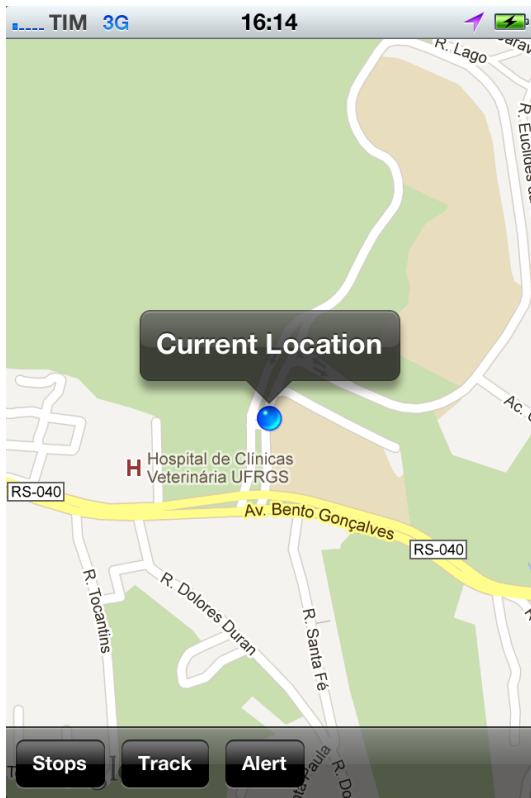
Figura 5.8: Módulo Cliente – Alertando ao usuário a aproximação de um ônibus.

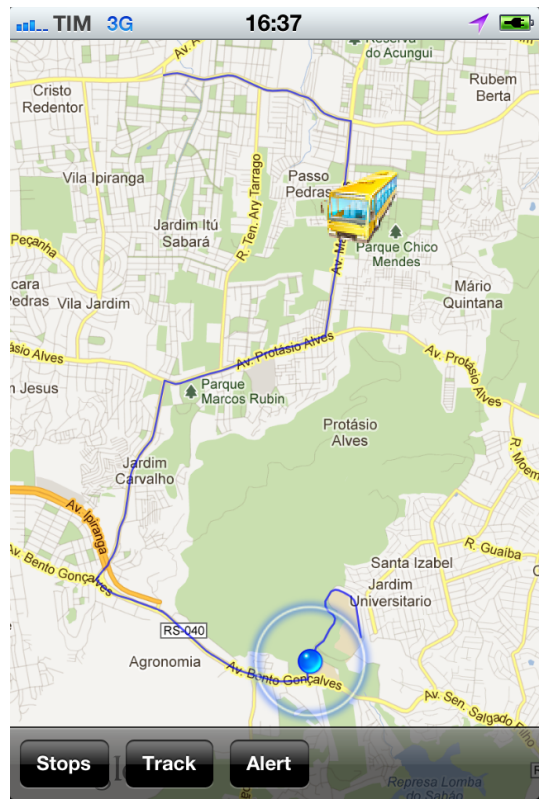
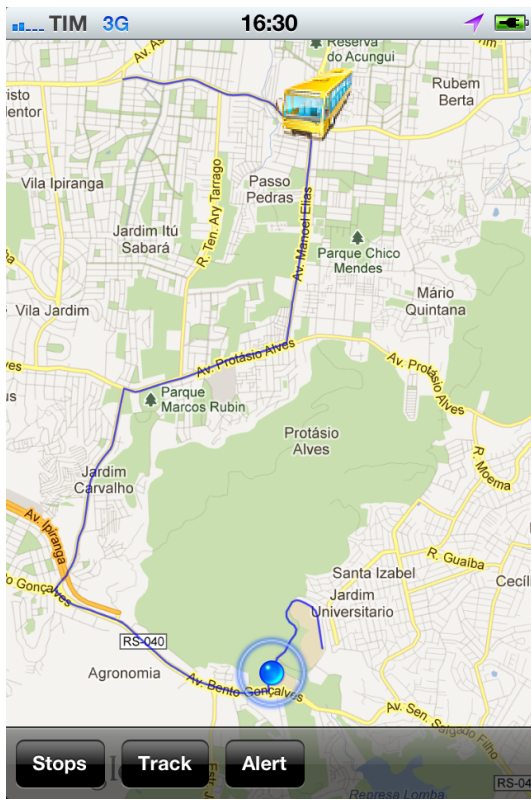
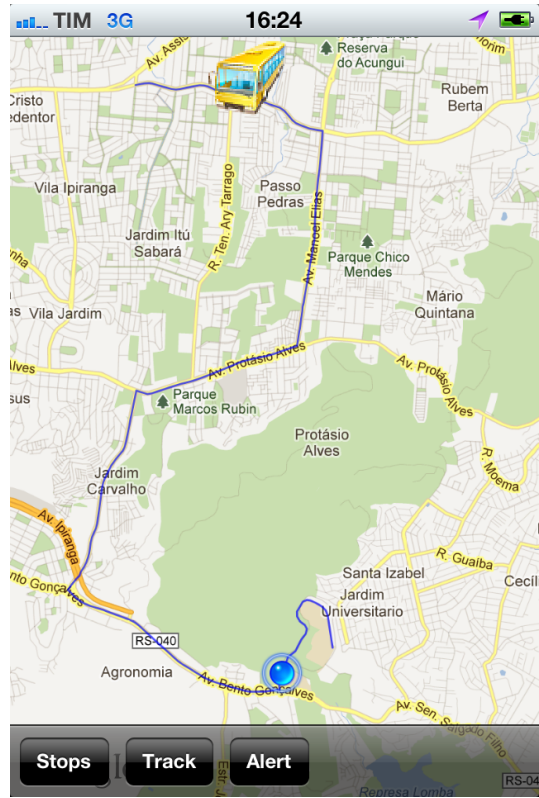
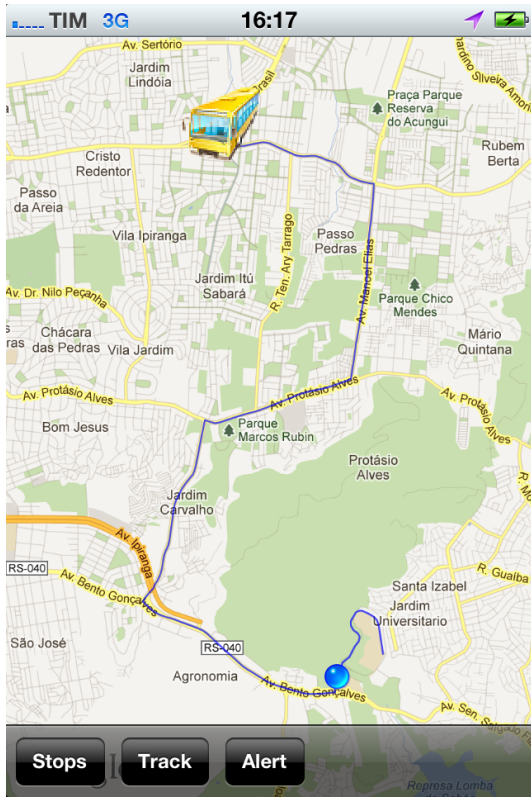
5.5 Validação da implementação

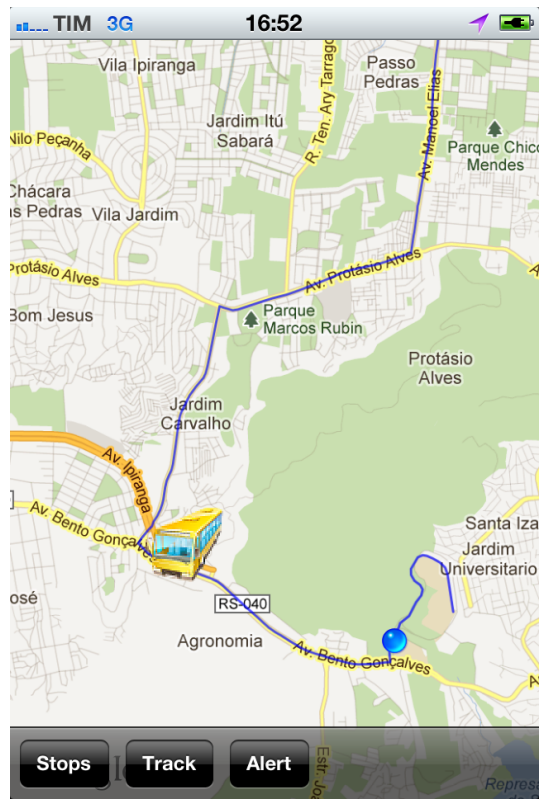
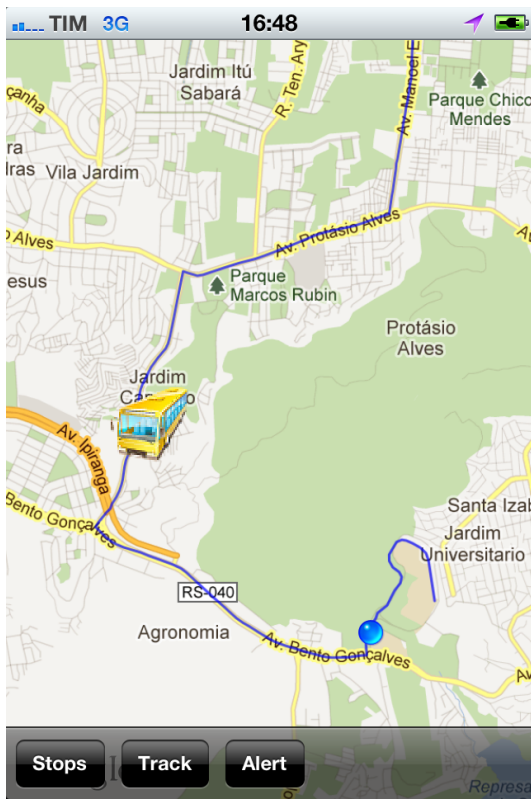
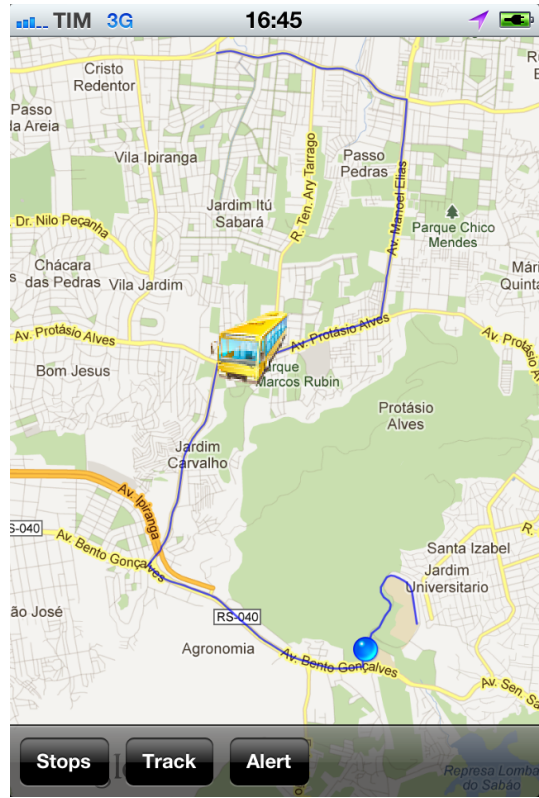
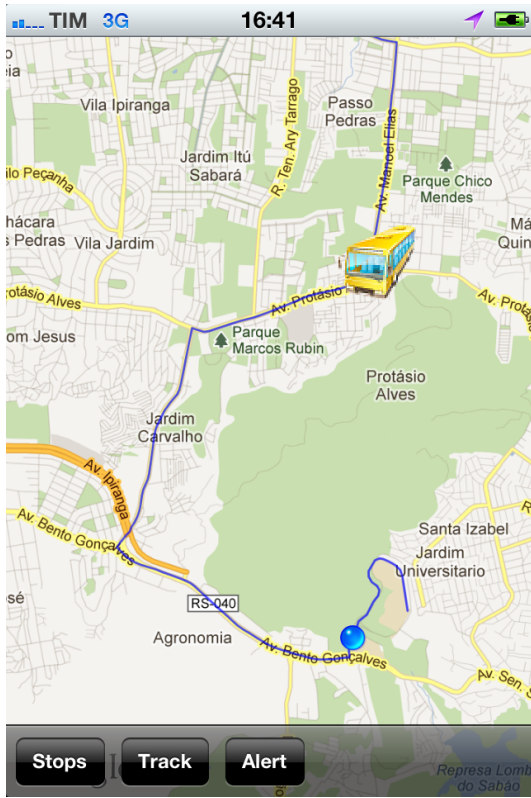
Todas as figuras apresentadas durante a prototipação, são imagens da aplicação sendo executada no simulador do SDK da plataforma iOS. Durante os testes no simulador a aplicação fez requisições a um servidor Apache local onde o módulo servidor respondia a tais requisições. Os resultados obtidos de uma avaliação realizada em um simulador pode não ter grande valor em se tratando de aplicações móveis, uma vez que o ambiente não reflete a infraestrutura de comunicação encontrada em condições reais. Com isso, muitos dos problemas que poderiam ser enfrentados em situações reais, como possíveis falhas de comunicação entre os módulos cliente e servidor ou atrasos na obtenção de informações de rastreamento, acabam sendo ignorados.

Devido ao fato da cidade de Porto Alegre não ter uma infraestrutura de rastreamento aplicado a sua frota de transporte urbano, a avaliação do protótipo da aplicação Bus Tracker em condições reais é uma tarefa um tanto complicada. Nem por isso, este trabalho irá se abster de uma avaliação mais realista. Para tanto, um pequeno módulo rastreador foi implementado. Como já foi mencionado na seção 5.2, o propósito deste módulo é simular um veículo sendo rastreado enquanto realiza sua viagem por uma linha específica. O funcionamento do módulo rastreador é bastante simples. Ele apenas recupera sua posição atual a cada 10 segundos e envia para o módulo servidor, informando que aquela posição pertence a um veículo que está realizando uma viagem para uma linha predefinida. O módulo servidor apenas armazena esta informação que posteriormente será consumida pelo módulo cliente.

Para realizar testes mais realistas, o módulo rastreador foi instalado em um dispositivo iPhone o qual foi utilizado em um carro para simular uma viagem de ônibus. A linha escolhida para realizar a simulação foi a linha “T102-1 TRIANGULO/ANTONIO DE CARVALHO”. Além disso, o módulo servidor teve que ser disponibilizado na Web de forma que ficasse acessível pela Internet, tanto para o módulo rastreador quanto para o módulo cliente. O carro partiu da zona norte de Porto Alegre, onde fica o terminal da linha escolhida, em direção à UFRGS (Campus do Vale) onde fica o outro terminal da linha. Um usuário ficou, em um ponto de ônibus próximo ao final da linha, acompanhando o veículo pelo módulo cliente, aguardando ser notificado quando o veículo se aproximasse dele. A figura a seguir apresenta a tela do módulo cliente em diferentes momentos da simulação.







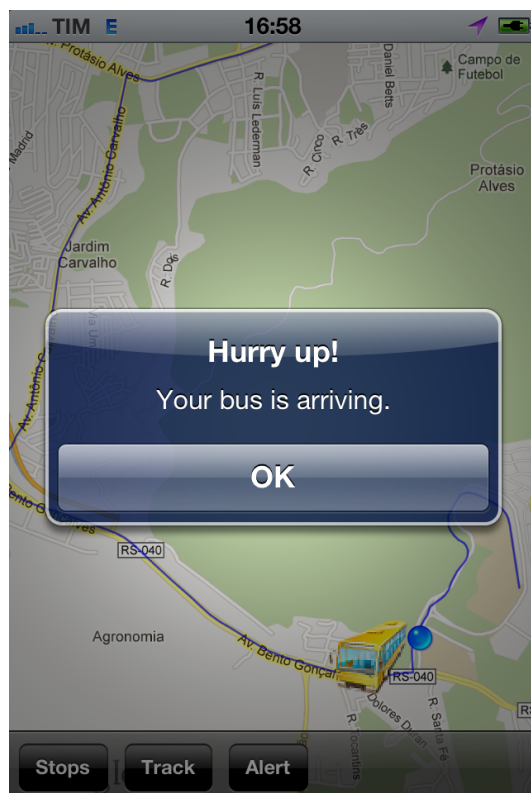
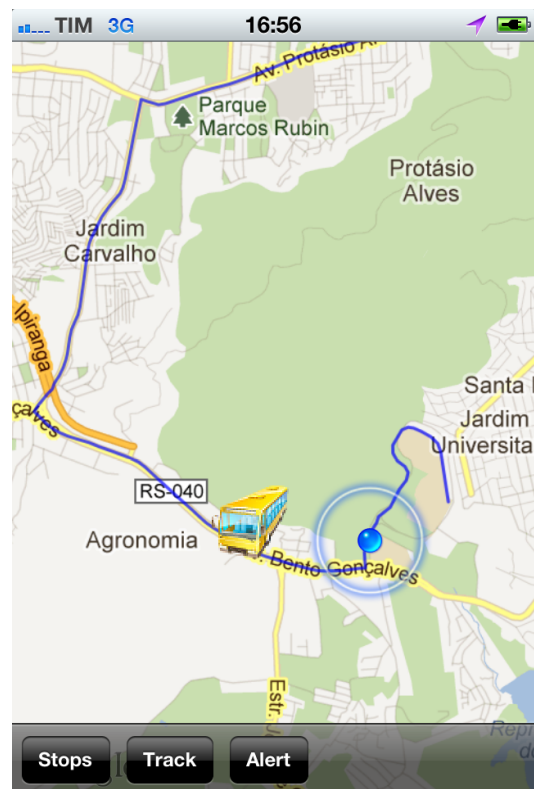
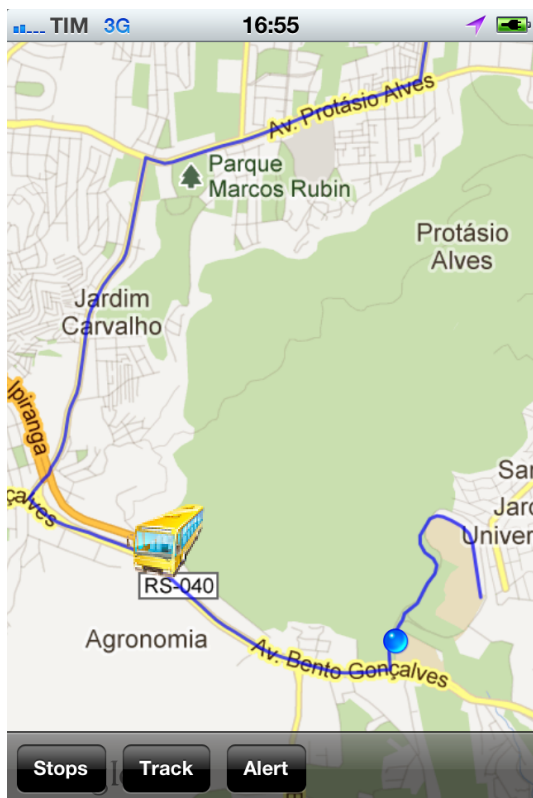


Figura 5.9: Módulo Cliente – Sequência de telas obtidas durante avaliação com módulo rastreador.

Apesar da realização de uma avaliação em condições mais realistas, muitas questões ainda ficaram em aberto. Por exemplo, como garantir a disponibilidade e a qualidade dos serviços da aplicação em lugares onde as redes de celulares não disponibilizam um

serviço de qualidade. No Brasil, a qualidade e a cobertura das redes 3G ainda é muito aquém do que se encontra em países mais desenvolvidos. Este é certamente um obstáculo a ser transposto por aplicações móveis que utilizam alguma informação de tempo real, como no caso da aplicação Bus Tracker. Além disso, o número de cidades brasileiras que investiram em uma infraestrutura de rastreamento para o transporte urbano ainda é muito pequeno. Porto Alegre, por exemplo, tem um sistema que apenas monitora a passagem de veículos por determinados pontos da cidade (SOMA). Muito provavelmente os dados gerados por este sistema não permitem o desenvolvimento de uma aplicação como a que foi proposta neste trabalho. Para a implementação de um serviço destes em Porto Alegre, seria necessário que cada veículo fosse monitorado individualmente, e que seus dados de posicionamento estivessem disponíveis para utilização em tempo real. A questão é que, o custo para se implementar e manter uma infraestrutura de rastreamento ainda é muito alto, apesar das tecnologias relacionadas terem apresentado um barateamento considerável nos últimos anos. Esse custo inibe as empresas de transporte a investirem no rastreamento por acreditarem ser um investimento sem muito retorno, ficando muitas vezes a cargo do Estado esse tipo de investimento.

6 CONCLUSÃO

6.1 Análise geral do trabalho

Para que a realização deste trabalho fosse possível, algumas importantes premissas tiveram que ser assumidas. Questões relacionadas a obtenção de dados de rastreamento ficaram fora do escopo do trabalho, cabendo apenas um estudo sobre tecnologias utilizadas para tal fim. Sem a preocupação com a obtenção dos dados de rastreamento, o foco do trabalho tornou-se o módulo cliente, onde o usuário teria a capacidade de acompanhar em tempo real os veículos do transporte assim como receber notificações sobre eventos relacionados a estes veículos. A prototipação buscou validar o fluxo principal de serviços oferecidos pela aplicação, optando por simplificar a interação com o usuário sem abandonar os requisitos que guiaram o projeto. O resultado é um protótipo que permite ao usuário avaliar as principais funcionalidades aqui projetadas.

Apesar da impossibilidade de uma avaliação utilizando dados gerados a partir de uma infraestrutura de rastreamento, os resultados obtidos com o protótipo são interessantes, pois confirmam a viabilidade de aplicações móveis que de alguma forma beneficiam usuários do transporte urbano com informações sobre a localização de veículos obtidas em tempo real. Neste sentido, o propósito almejado com a criação de um protótipo foi alcançado, pois foi feita uma prova de conceito que demonstrou ser viável a implementação de uma aplicação nos moldes da aplicação Bus Tracker.

O trabalho deixa em aberto uma série de questões relacionadas a desempenho, disponibilidade, dentre outros critérios de qualidade relacionados aos sistemas de informação, devido as dificuldades de se executar uma avaliação em condições reais de uso. Nem por isso, as ideias, a análise e o projeto da aplicação Bus Tracker devam ser desconsiderados. Pelo contrário, o projeto serve de base para futuros trabalhos que queiram solucionar ou propor alternativas para questões aqui não abordadas, ou até mesmo expandir e aprimorar as ideias apresentadas.

6.2 Possibilidades futuras

Aplicações de tempo real apresentam uma complexidade maior quando comparadas com aplicações onde o tempo não é relevante. Ainda mais, quando as informações, cujo tempo é relevante, devem ser transmitidas através de redes como no caso da aplicação Bus Tracker. Nessas condições os pontos de falha podem ser muitos, difíceis de identificar e até mesmo impossíveis de serem contornados pela aplicação, pois podem estar fora do seu controle. Aplicações móveis baseadas em localização são, na grande maioria, dependentes de redes de celular, as quais nem sempre estão disponíveis ou oferecem os serviços mínimos para o bom funcionamento da aplicação.

As redes 3G oferecem serviços como o GPRS, que barateiam consideravelmente os custos de comunicação utilizando redes de celular. O problema é que, no Brasil, ainda são poucas as cidades estão cobertas por redes 3G, sendo que nessas poucas cidades a cobertura não é total e a qualidade dos serviços ainda deixa a desejar. Este é sem dúvida uma grande obstáculo a ser vencido para que então um número maior de aplicações com base em localização possam trazer reais benefícios a seus usuários.

No caso de cidades onde não existe uma infraestrutura de rastreamento, uma aplicação que permita aos próprios usuários do transporte informar de tempos em tempos a localização do veículos em que estão pode ser uma alternativa para a obtenção de informações de rastreamento. Esta é, sem dúvida, uma alternativa ousada, e talvez um pouco difícil de ser aceita, pois envolve questões de privacidade, além é claro, de exigir um sistema com inteligência para tratar informações provenientes de diferentes fontes, validar essas informações quanto a sua veracidade. Hoje existem diferentes aplicações que estimulam os usuários a divulgar em redes sociais sua localização. Path, por exemplo, é uma rede social que estimula seus usuários a compartilharem sua rotina com seus amigos, principalmente questões de localização, como: almoçando em tal lugar, passeando em tal parque, etc (Path). Porque não pensar em uma rede social para rastreamento do transporte urbano. As pessoas geram a informação enquanto estão nos veículos e consomem a informação quando estiverem necessitando de um veículo.

Este trabalho não tem como objetivo querer estabelecer conceitos ou ideias absolutas sobre os assuntos aqui abordados. Ele apenas ressalta a infinidade de possibilidades que estão surgindo com a popularização de tecnologias relacionada a mobilidade. Sem dúvida, a mobilidade ainda está dando seus primeiros passos. Há muito para ser feito. Há muitas barreiras para serem transportas. Desafios gigantescos para serem enfrentados. Uma nova área da computação se apresenta para aqueles que movem o mundo com suas ideias inovadoras. Certamente, se espera que outros trabalhos venham para complementar, aprimorar e evoluir as ideias e os conceitos aqui apresentados.

REFERÊNCIAS

3GPP, 3rd Generation Partnership Project. Disponível em: <<http://www.3gpp.org/>>. Acesso em: maio 2011.

Alfacomp Automação Industrial. Disponível em: <<http://www.alfacomp.ind.br/site/default.asp>>. Acesso em: abril 2011.

Android Developer. Disponível em: <<http://developer.android.com/index.html>>. Acesso em: maio 2011.

Apple Developer. Disponível em: <<https://developer.apple.com/>>. Acesso em: junho 2011.

Auckland Transport, Real time passenger information. Disponível em: <<http://www.aucklandtransport.govt.nz/Pages/default.aspx>>. Acesso em: março 2011.

CITFOR (2004), Autarquia Municipal de Trânsito, Serviços Públicos e Cidadania de Fortaleza. Disponível em: <www.amc.fortaleza.ce.gov.br>. Acesso em: março 2011.

Cittati Tecnologia. Disponível em: <<http://www.cittati.com.br>>. Acesso em: março 2011.

Cockburn, A. (2001), Writing Effective Use Cases. Addison-Wesley.

Dey, A. K., G. D. Abowd. (1999), Towards a better Understanding of Context and Context-Awareness, Technical Report GIT-GVU-99-22, Georgia Institute of Technology.

EPTC, Empresa Pública de Transporte e Circulação. Disponível em: <www.eptc.com.br>. Acesso em: abril 2011.

Fowler, Martin (1997), Analysis Patterns: Reusable Object Models, Addison-Wesley.

G3VB, G3VB Sistemas e Tecnologia. Disponível em: <<http://www.g3vb.com.br>>. Acesso em: março 2011.

GeoSIT, Informações do projeto Geosit. Disponível em: <www.geosit.com.br>. Acesso em: abril 2011.

GPS, Global Positioning System. Disponível em: <<http://www.gps.gov/>>. Acesso em: abril 2011.

HeLMi, Project HeLMi. Disponível em: <http://www.hel2.fi/liikenteenohjaus/eng/pt_telematics.asp>. Acesso em: março 2011.

Heuser, Carlos Alberto (2009), Projeto de Banco de Dados, Bookman, 6a edição.

Hightower, J., G. Borriello. (2001a), Location Systems for Ubiquitous Computing. IEEE Computer, Vol. 34, No. 8. August (2001), 57-66.

- Hightower, J., G. Borriello. (2001b). Location Sensing Techniques. Companion Report to the IEEE Computer article on Location Systems for Ubiquitous Computing. University of Washington, Computer Science and Engineering.
- Kupper, Axel (2005), Location-based Services: Fundamentals and Operation, John Wiley & Sons Ltd.
- Kruchten, P. (2000), The Rational Unified Process – An Introduction, Addison-Wesley, 2nd edition.
- Larman, Craig (2004), Applying UML and Patterns: An Introduction to Object-Oriented Analysis and Design and Iterative Development, Prentice Hall, 3rd edition.
- Magalhães, Caroline Tristão de Alencar (2008), Avaliação de Tecnologias de Rastreamento por GPS para Monitoramento do Transporte Público por Ônibus, Dissertação - Universidade Federal do Rio de Janeiro, COPPE.
- Magela, Rogerio (2006), Engenharia de Software Aplicada: Princípios (volume 1), Alta Books.
- Maxtrack Industrial Ltda. Disponível em: <<http://www.maxtrack.com.br/site/home>>. Acesso em: março 2011.
- NextBus, NextBus Inc. Disponível em: <<http://www.nextbus.com/homepage>>. Acesso em: março 2011.
- NSW – TransportInfo. Disponível em: <<http://www.131500.com.au>>. Acesso em: maio 2011.
- OGC (2001), OpenGIS Implementation Specification: Coordinate Transformation Services. Disponível em: <<http://www.opengis.org/>>. Acesso em: junho 2011.
- OGC (2004), OpenGIS Location Services (OpenLS): Core Services. Disponível em: <<http://www.opengis.org/>>. Acesso em: junho 2011.
- Ônibus Legal, Projeto Ônibus Legal. Disponível em: <<http://www.onibuslegal.com>>. Acesso em: março 2011.
- Open Geospatial Consortium. Disponível em: <<http://www.opengeospatial.org/>>. Acesso em: maio 2011.
- Open Mobile Alliance. Disponível em: <<http://www.openmobilealliance.org/>>. Acesso em: maio 2011.
- Path. Disponível em: <<https://path.com/>>. Acesso em: janeiro 2012.
- PostgreSQL. Disponível em: <<http://www.postgresql.org/>>. Acesso em: agosto 2011.
- Python. Disponível em: <<http://www.python.org/>>. Acesso em: julho 2011.
- Rigaux, P., M. Scholl, A. Voisard. (2002). Spatial Databases with Application to GIS. Morgan Kaufmann Publishers.
- RTD, Regional Transportation District. Disponível em: <<http://www.rtd-denver.com/>>. Acesso em: maio 2011.
- Silberschatz, Abraham; Korth, Henry F.; Sudarshan, S. (2006), Database System Concepts, McGraw-Hill Companies, Inc.
- SIT, Sistema Integrado de Transporte. Disponível em: <www.uberlandia.mg.gov.br>. Acesso em: abril 2011.
- SPTrans. Disponível em: <<http://www.sptrans.com.br/>>. Acesso em: junho de 2011.

TCRP, Transit Cooperative Research Program. Disponível em:
<<http://www.tcrponline.org>>. Acesso em: abril 2011

UTCC, Urban Traffic Control Center. Disponível em:
<<http://www.hel2.fi/liikenteenohjaus/eng/index.asp>>. Acesso em: abril 2011.

Wplex, Wplex Software Ltda. Disponível em: <<http://www.wplex.com.br>>. Acesso em:
Março 2011.

ANEXO <MONOGRAFIA DO TRABALHO DE GRADUAÇÃO I>

Aplicação Bus Tracker – Oferecendo uma melhor experiência aos usuários do transporte urbano, a partir da utilização de informações de rastreamento veicular

Autor: Luciano Goularte Siqueira

Orientador: Cláudio Fernando Resin Geyer

Instituto de Informática – Universidade Federal do Rio Grande do Sul (UFRGS)
Porto Alegre – RS – Brasil

lgoularte@gmail.com, geyer@inf.ufrgs.br

Resumo. Ao olhar para o mercado de dispositivos móveis (celulares, smartphones, tablets) pode-se identificar um universo de oportunidades que cresce a cada dia. Cientes da importância deste mercado e do crescimento por ele apresentado nos últimos anos, empresas de tecnologia investem pesado para desenvolver novas tecnologias, novos serviços, mais comodidade e agilidade aos usuários. Na última década, foi possível acompanhar a evolução das tecnologias relacionadas à mobilidade, e como essas tecnologias permitiram o surgimento de inúmeras aplicações que agregam valor e facilidade à vida do usuário. Entre os principais serviços que se tornaram populares com a evolução dos dispositivos móveis estão os serviços baseados em localização, ou seja, serviços que tiram proveito da capacidade de obter de alguma forma a localização do dispositivo em relação a algum ponto ou sistema de referência. Considerando tal contexto, o propósito deste trabalho é projetar e desenvolver um protótipo de uma aplicação para dispositivos móveis, que permita aos usuários do transporte público acompanhar em tempo real o deslocamento dos veículos de transporte (ônibus, trem, lotação, etc.), assim como configurar eventos pelos quais desejam ser avisados (por exemplo, receber um alerta quando determinado ônibus de determinada linha se aproximar de um ponto de ônibus específico).

1. Introdução

1.1. Motivação

Com o surgimento dos primeiros computadores pessoais surgiu também um novo mercado com infinitas possibilidades. Com o tempo os computadores evoluíram, se tornaram portáteis, foram integrados a outros dispositivos, como no caso dos telefones celulares, criando novas possibilidades de evolução para os sistemas de informação. Toda essa explosão tecnológica deu origem a novos mercados, novas possibilidades de serviços que até então não eram possíveis devido a não existência de tecnologias necessárias. Um destes novos mercados diz respeito aos dispositivos móveis (como telefones celulares, PADs, smartphones, e mais recentemente os tablets) e ao novo universo de possibilidades que esses dispositivos apresentam tanto aos desenvolvedores de aplicações quanto aos usuários.

À medida que as tecnologias relacionadas à mobilidade evoluem, também evoluem as aplicações relacionadas a ela. Aplicações conscientes do contexto passam a ser comuns [Dey e Abowd 1999]; o usuário passa a ser reconhecido como indivíduo; os serviços passam a ter consciência do contexto relacionado a este indivíduo e a fazer uso deste contexto para agregar valor ao que é ofertado ao usuário. Os serviços com base em localização (LBS – Location Based Services) fazem parte deste paradigma, onde o referido contexto é aqui representado pela localização [Kupper 2005]. Os LBSs tem a capacidade de conhecer a localização, ou do usuário ou de algum alvo específico, e a partir desta informação direcionar seus serviços para que levem em consideração este contexto e propiciem uma melhor experiência para o usuário.

1.2. Objetivo

É no contexto de dispositivos móveis e tecnologias de posicionamento que este trabalho fundamenta o projeto e a prototipagem de um LBS que ofereça, aos usuários do transporte público das grandes cidades, serviços que permitem o monitoramento em tempo real dos veículos de transporte e que façam uso destas informações de rastreamento para gerar avisos e mensagem de alerta ao usuário à medida que eventos pré configurados por eles (como no caso de determinado veículo se aproximar-se do ponto de embarque do usuário) sejam identificados.

Os LBSs começaram a ganhar força com o popularização e barateamento das tecnologias de posicionamento, como o Sistema Global de Posicionamento (GPS - Global Positioning System [GPS]) e dos serviços de comunicação prestados pelas redes de telefonia celular GSM, como é o caso do Serviço de Rádio de Pacote Geral (GPRS – General Package Radio Service) [3GPP].

Levando em consideração que o número de pessoas que tem algum tipo de dispositivo móvel com tais tecnologias é cada vez maior, e que há uma necessidade de melhoria constante por parte dos serviços de transporte público, pode-se imaginar que serviços, como o que será proposto neste trabalho, tragam um benefício considerável aos usuários do transporte.

Não é objetivo deste trabalho definir uma infraestrutura de rastreamento veicular para o transporte público. Ele apenas apresenta um panorama das tecnologias utilizadas para tal fim, e define um modelo de base de dados para armazenar as informações de posicionamento dos veículos geradas, em tempo real, por uma possível infraestrutura de

monitoramento, assim como informações referentes a itinerários, pontos de parada e identificação dos veículos. Tendo como ponto de partida uma base de dados com as informações acima referidas, é então definida uma aplicação (aqui denominada de *Bus Tracker*), no modelo cliente-servidor, que permite ter acesso, via dispositivos móveis, as informações de posicionamento dos veículos em tempo real.

1.3. Organização do texto

Este trabalho está organizado em uma sequência lógica, partindo da motivação e do objetivo do trabalho, passando pelas tecnologias relacionadas a este, até apresentar a análise de requisitos e parte inicial do projeto de uma aplicação Bus Tracker.

O Capítulo 2, Rastreamento veicular aplicado ao transporte público, traz algumas experiências, com rastreamento veicular, voltadas transporte urbano, além de demonstrar como a aplicação destas tecnologias melhoram o controle da qualidade do transporte bem como a experiência dos usuários em relação ao serviço de transporte. Além disso, são apresentadas algumas empresas que vendem serviços ou até mesmo soluções completas para o rastreamento e monitoramento do transporte urbano.

O Capítulo 3, Serviços com base em localização, traz uma definição do que são Serviços Conscientes do Contexto e Serviços com Base em Localização, e qual a relação entre eles. Apresenta ainda, conceitos relacionados com as principais tecnologias que dão suporte aos LBSs.

O Capítulo 4, Projeto, apresenta inicialmente uma breve definição do que é a Engenharia de Software. O restante do capítulo traz a análise de requisitos e o início do projeto da aplicação Bus Tracker, sendo que este processo segue as diretrizes do Processo Unificado - metodologia de Análise de Projeto Orientado a Objetos, a qual também é apresentada neste capítulo à medida que o projeto evolui.

O Capítulo 5, Conclusão, apresenta, de uma perspectiva geral, os resultados deste primeiro trabalho e como eles influenciam as próximas tarefas. Além disso é apresentado um plano de atividades a ser seguido durante o desenvolvimento do Trabalho de Graduação 2, o qual é uma continuação do que será exposto neste primeiro trabalho.

2. Rastreamento veicular aplicado ao transporte público

Antes mesmo da popularização de tecnologias como o GPS, algumas cidades já experimentavam o rastreamento veicular como ferramenta de controle e melhoria dos serviços de transporte público. Em meados da década de 90, Londres (Inglaterra) deu início à implantação do seu sistema de rastreamento. O primeiro sistema implantado na cidade, denominado de *Countdown*, utilizava transponders, permitindo assim a detecção dos veículos ao passarem por pontos de leitura. O sistema começou a ser testado em 1992 nos ônibus da rota 18. Os testes realizados com os passageiros sobre o projeto foram muito positivos. Entre 1993 e 1994 o projeto foi expandido e amplamente testado em outros corredores de ônibus. A partir de 1996 teve início a instalação de sistemas automáticos de localização de veículos como tecnologia de rastreamento. Em março de 2002, 1473 painéis tinham sido instalados e estavam operacionais. O plano era ter um total de 2400 instalados em março de 2003 e 4000 no ano de 2005. Esse número de 4000 painéis instalados chegaria a cobrir 25% de todos os pontos de parada de Londres e beneficiaria 60% de todas as viagens de passageiros [Magalhães 2008].

Tanto o poder público quando as empresas de transporte estão cada vez mais conscientes da necessidade de aprimoramento deste serviço tão essencial que é o transporte urbano. Na iniciativa privada não é diferente, pois já é grande o número de empresas que de alguma forma explora este mercado tão promissor. A empresa Maxtrack, por exemplo, oferece um grande número de equipamentos e soluções de rastreamento, tanto para o transporte público, como para outras áreas de transporte. Algumas empresas como a G3VB Sistemas e Tecnologia [G3VB], Cittati e Wplex oferecem soluções completas (hardware, software, treinamento, etc.) voltadas para o transporte público, com serviços próprios para este tipo aplicação. A Wplex, por exemplo, desenvolveu o sistema WplexCO, o qual permite o monitoramento da frota de ônibus em tempo real assim como a comunicação direta com os motoristas em trânsito [Wplex]. A instalação do sistema demanda pouco investimento em infraestrutura, pois ele foi desenvolvido segundo o conceito de Software como um Serviço (SaaS). Ou seja, todas as informações de gerenciamento e todo o monitoramento é feito via *Web*. A comunicação de dados entre veículos e empresa ocorre no seguinte sentido: junto ao veículo é instalado um rastreador GPS/GPRS embarcado e um terminal com teclado; o rastreador envia informações de posicionamento, assim como outras informações de controle para o centro de dados da empresa, onde as informações são processadas e armazenadas em uma base de dados. A empresa de transporte tem acesso, via *Web*, a serviços que consultam tal base de dados, permitindo assim, o rastreamento e controle da frota por parte da empresa de transporte. O WplexINFO é parte complementar do WplexCO, disponibilizando aos usuários informações e serviços que facilitam a utilização do transporte público, como itinerários, pontos de parada, previsão de chegada, dentre outras.

O número de cidades que tem algum tipo de infraestrutura de rastreamento aplicada ao monitoramento do transporte público é cada vez maior. Alguns bons exemplos são Auckland (Nova Zelândia) [Auckland Transport], Denver (Colorado – USA), Sydney (Austrália), Uberlândia, que foi a primeira cidade brasileira a ter toda a frota de transporte urbano monitorada [GeoSIT], e Helsinque (Finlândia), que integrou o sistema de rastreamento com semáforos permitindo que veículos que estejam atrasados tenham prioridade de passagem [HelMi].

Como se pode ver, a tendência tecnológica no rastreamento de veículos aplicado ao transporte público é a utilização de módulos rastreadores, com tecnologias GPS e GPRS, instalados nos veículos. Estes módulos enviam suas informações para um centro de dados que as processa e armazena em uma base de dados. Além disso, a utilização de sistemas web (software como um serviço), para disponibilizar serviços tanto para a empresa de transporte quanto para os usuários, é comum em quase todos os sistemas.

3. Serviços com base em localização

Neste capítulo serão abordados os serviços com base na localização e as principais tecnologias que auxiliam no desenvolvimento destes serviços. É apresentada uma definição para serviços conscientes do contexto, e como os serviços com base em localização (LBS – Location Based Services) pertencem ao conjunto destes serviços. Logo em seguida são apresentadas algumas das tecnologias que auxiliam no desenvolvimento de um LBS.

3.1. LBSs e serviços conscientes do contexto

Serviços Conscientes do Contexto (CAS - Context-aware Services) são definidos como serviços que automaticamente adaptam seu comportamento de acordo com um ou mais parâmetros que refletem o contexto de um determinado alvo [Dey e Abowd 1999]. Estes parâmetros são chamados de informações de contexto, e podem pertencer a diferentes categorias de informações, como por exemplo, contexto pessoal, contexto espacial, contexto social, contexto físico, etc. Os LBSs são considerados um subconjunto dos serviços conscientes do contexto, onde o contexto é a localização espacial.

A forma como informações de contexto são utilizadas em um serviço com base em localização fica clara nesta definição apresentada em [Kupper 2005]: LBSs são serviços de fornecem informações que foram criadas, compiladas, selecionadas, ou filtradas levando em consideração a localização atual dos usuários ou de outras pessoas ou objetos móveis.

Um LBS não é responsável pela geração das informações de localização. Ele apenas faz uso destas informações, por ele obtidas junto a um subsistema chamado de Serviço de Localização (LS - Location Service). Este sim é responsável pela geração e distribuição das informações de localização. Essencialmente, os Serviços de Localização contribuem para o funcionamento dos LBSs, sendo aqueles um importante subsistema destes.

Os LBSs podem ser classificados como reativos ou proativos. Um LBS reativo é sempre ativado pelo usuário enquanto um LBS proativo responde automaticamente a algum evento de localização predefinido. Ao contrário de LBSs reativos onde, normalmente, o usuário precisa ter sua localização identificada apenas uma vez, os LBSs proativos precisam rastrear o usuário para detectar algum evento de localização.

A tecnologia chave dos LBSs é chamada de Posicionamento (Positioning). É ela que torna tão interessante os LBSs, pois permite obter a informações de posicionamento sem que os usuários necessitem entrar com estas informações manualmente. Existem inúmeras técnicas para implementação desta tecnologia, sendo que elas diferem na qualidade dos parâmetros de posicionamento resultantes.

Um exemplo simples de um LBS é aquele que provê ao usuário pontos de interesse (como restaurantes, farmácias, etc.) próximos a sua localização. Esse tipo de sistema, quando solicitado, automaticamente identifica a localização do usuário e a utiliza em sua pesquisa como uma forma de filtro para as informações a serem apresentadas. Normalmente, a pesquisa é encaminhada a um servidor que devolve uma lista de pontos que foram compilados utilizando as informações de localização do usuário e uma distância limite entre o usuário e o ponto em questão. Este tipo de serviço é uma espécie de páginas amarelas que utilizam o contexto de localização para refinar os resultados da busca. Os resultados de uma busca podem ser combinados com algum serviço de navegação, agregando ainda mais valor ao LBS.

3.2. Fundamentos de localização

No contexto dos LBSs, a palavra localização denota o lugar onde determinado objeto se encontra no mundo real. Segundo este conceito, as localizações podem ser classificadas de três formas:

1. Localização Descritiva – Representa a geografia natural, como montanhas, rios, ou ainda objetos geográficos feitos pelo homem, como fronteiras, rodovias,

prédios, etc. Estas localizações são de fundamental importância, pois construímos nossos sistemas de localização e referência baseados nelas.

2. **Localização Espacial (Posição)** – Representa, simplesmente, um ponto no espaço Euclidiano. Normalmente é representada por coordenadas em duas ou três dimensões. Este tipo de localização serve de base para o levantamento e mapeamento de localizações descritivas.
3. **Localização na Rede** – Refere-se a forma como redes de telecomunicações identificam nodos dentro da sua rede ou outras redes.

Uma importante função de um LBS é realizar o mapeamento entre as diferentes categorias de localização, a fim de cumprir com sua tarefa. Por exemplo, uma localização descritiva precisa ser convertida em uma localização espacial para que sua distância em relação a uma outra localização possa ser calculada. Ou ainda, uma localização espacial pode ser convertida em uma localização de rede para que algum tipo de roteamento baseado no contexto seja realizado corretamente. Todos esses mapeamentos podem ser atingidos com a utilização de bases de dados espaciais e Sistemas de Informação Geográfica (GIS - Geographic Information Systems) [Kupper 2005].

3.2.1. Localização espacial

A localização espacial faz uso de sistemas de referência que dividem uma área geográfica, normalmente a terra inteira, em unidades de formato e tamanhos fáceis de serem tratados. Um sistema de referência espacial é composto por:

- um sistema de coordenadas, por exemplo, Sistema de coordenadas cartesianas;
- um ponto de partida, que refere-se a uma definição da forma e do tamanho da terra;
- e uma forma de projeção em mapa [OGC 2001].

3.2.2. Projeções em mapa

Projeções em mapas são usadas para representar a superfície tridimensional da terra em uma superfície bidimensional, o que é mais conveniente para ser apresentado em monitores e telas de dispositivos móveis.

3.2.3. Bases de dados espaciais e Sistemas de Informação Geográfica (GIS)

A maioria dos métodos de posicionamento usados pelos LBSs resulta na informação de uma localização espacial. O problema é que nem sempre esse tipo de informação é adequado para todos os propósitos do LBS. Por exemplo, no caso de um LBS que apresente em um mapa os possíveis caminhos entre dois pontos de uma cidade utilizando o transporte público, ele deve, de alguma maneira, ser capaz de fazer conversões entre localizações espaciais e localizações descritivas.

As Bases de dados espaciais e os GISs representam a tecnologia base para realização desta tarefa. Uma base de dados espacial é uma base de dados otimizada para as necessidades de armazenamento e consultas relacionada ao domínio da localização espacial. Um GIS pode ser caracterizado como um sistema que permite a captura, gerência, integração, análise, manipulação e apresentação de dados espaciais

relacionados a Terra. Desta forma, pode-se dizer que uma base de dados espacial é parte integrante de um GIS, sendo responsável pelo armazenamento de dados espaciais [Rigaux, Scholl e Voisard 2002].

3.3. Posicionamento

O Posicionamento é o processo de obter a posição espacial de algum objeto. Existem vários métodos para realizar esta tarefa, sendo que eles diferem em questões como qualidade do resultado, quantidade de processamento realizado para obter o resultado, dentre outros. Posicionamento é, normalmente, determinado pelos seguintes elementos:

- um ou mais parâmetros a serem observados pelos métodos de medição;
- um método de posicionamento para calcular a posição a partir dos parâmetros medidos;
- um sistema de referência descritivo ou espacial;
- uma infraestrutura (rede telefônica, rede de satélites);
- protocolos para coordenar o processo de Posicionamento.

O ponto-chave do Posicionamento é a medição dos parâmetros observados. Estes parâmetros refletem uma relação espacial entre o ponto a ser medido e um ou mais pontos fixos próximos a este ponto, onde estes pontos fixos são pontos conhecidos [Hightower e Borriello 2001a].

4. Projeto

4.1. Engenharia de Software

A Engenharia de software é uma área do conhecimento voltada para a especificação, desenvolvimento e manutenção de sistemas de *software*, aplicando tecnologias e práticas que englobam linguagens de programação, banco de dados, ferramentas, plataformas, bibliotecas, padrões e processos dentre outras. Os fundamentos científicos da engenharia de software envolvem o uso de modelos abstratos que permitem especificar, projetar, implementar e manter sistemas de software, avaliando e garantindo suas qualidades. Além disso, a engenharia de software oferece mecanismos para planejar e gerenciar o processo de desenvolvimento [Magela 2006].

Assim como toda engenharia, a engenharia de software tem seus processos. Um processo de desenvolvimento de software é uma sequência coerente de práticas que objetiva o desenvolvimento ou evolução de sistemas de *software*. Estas práticas englobam as atividades de especificação, projeto, implementação, testes; e caracterizam-se pela interação de ferramentas, pessoas e métodos.

À medida que as tecnologias relacionadas ao desenvolvimento de software evoluem as metodologias, ou seja, os processos de engenharia de software também evoluem. Um bom exemplo foi a mudança de paradigma proposta pelo Processo Unificado em relação ao Modelo em Cascata ou Sequencial, como também é chamado. Em um Modelo em Cascata há uma tentativa de definir em detalhes todos ou a maioria dos requisitos, ou ainda, criar um projeto abrangente antes da programação. Há também a tentativa de definir um plano ou cronograma confiável logo no início do processo. A falsa suposição de que as especificações são previsíveis e estáveis, e podem ser corretamente definidas no início, com baixas taxas de modificação, faz com que o

modelo em cascata não seja realmente adequado ao desenvolvimento de software, levando assim a uma taxa grande de falhas, quando da sua utilização.

4.2. Processo unificado

Neste trabalho será utilizado como metodologia de desenvolvimento uma abordagem ágil do Processo Unificado (PU) [Larman 2004]. A ideia central nesta abordagem é a de usar iterações curtas, com duração fixa, em um processo de desenvolvimento iterativo, evolutivo e adaptativo. Algumas das melhores práticas desta abordagem incluem:

- enfrentar os problemas que envolvem altos riscos e alto valor já nas iterações iniciais;
- construir uma arquitetura central coesa nas primeiras iterações;
- verificar continuamente a qualidade; fazer testes logo de início, com frequência e em situações o mais realísticas possíveis;
- aplicar casos de uso quando adequado;
- modelar visualmente o software (UML);
- gerenciar requisitos cuidadosamente.

Um projeto que utilize o PU deve organizar o trabalho e as iterações de acordo com as seguintes fases:

1. **Concepção** – A concepção é um passo inicial curto, para estabelecer uma visão comum e o escopo básico do projeto. Inclui a análise dos requisitos funcionais críticos, e a preparação do ambiente de desenvolvimento para que a programação possa começar na fase seguinte.
2. **Elaboração** – A elaboração é a fase inicial de iterações durante a qual a arquitetura central e de alto risco do software é programada e testada, a maioria dos requisitos é descoberta e estabilizada e os principais riscos são mitigados e retirados.
3. **Construção** – A construção representa a segunda fase de iterações onde são implementados os elementos restantes de menor risco e mais fáceis, e onde é feita a preparação para a implantação.
4. **Transição** – A transição é a fase de testes beta e implantação.

Cada uma das fases descritas acima produz seus próprios artefatos (artefato é o termo usado para qualquer produto de trabalho, como código, gráficos, esquemas de banco de dados, documentos de texto, modelos, dentre outros). Entretanto, um aspecto importante do PU é que quase todas as atividades e os artefatos são opcionais, ficando a critério da equipe do projeto definir quais práticas adotar e quais artefatos gerar.

4.2.1. Análise de Requisitos

Requisitos são capacidades e condições às quais o sistema deve atender. O PU promove um conjunto de melhores práticas, dentre as quais *gerenciar requisitos* tem destaque, pois ela estimula o uso de uma abordagem sistemática para encontrar, documentar, organizar e rastrear as mudanças de requisitos de um sistema [Kruchten 2000]. Para tanto, o PU oferece vários artefatos de requisitos. Os principais são:

- **Modelo de Casos de Uso** – Conjunto de cenários típicos de uso do sistema.

- Especificação Suplementar – Tudo o que não está nos casos de uso. Este artefato se aplica principalmente a requisitos como desempenho, autenticação, etc.
- Glossário – Define termos importantes do domínio da aplicação.
- Visão – Resume requisitos de alto nível que são detalhados no modelo de casos de uso e especificação suplementar. É um documento curto contendo uma visão geral para entendimento rápido das grandes ideias do projeto.

4.3. Concepção – 1ª Fase do PU

A concepção é a fase do PU em que o escopo do produto é concebido. Esta fase tem como alguns de seus artefatos: a Visão, o Modelo de Casos de Uso, Especificações Suplementares, Lista de Riscos, Protótipos e Provas de Conceito. As subseções a seguir apresentam os artefatos escolhidos para este trabalho (Visão e Modelo de Casos de Uso) na fase de concepção.

4.3.1. Visão

Como foi mencionado no capítulo introdutório, o propósito deste trabalho é o projeto e prototipagem de uma aplicação para dispositivos móveis que permita aos usuários do transporte público, não apenas, acompanhar em tempo real o deslocamento dos veículos de transporte, como também obter serviços de alerta (alarmes, mensagens) à medida que determinados eventos (como a aproximação de determinado veículo de um ponto específico) são detectados pela aplicação.

Em um nível mais elevado de abstração, a aplicação pode ser descrita como a interação entre os seguintes componentes:

- uma Base de Dados que armazena informações referentes ao domínio da aplicação (que neste caso é o setor de transporte coletivo de uma cidade) e informações de tempo real pertinentes ao rastreamento dos veículos de transporte;
- um módulo Servidor que interage com a base de dados para obter as informações necessárias ao processamento das requisições recebidas de clientes;
- um módulo Cliente, a ser instalado em dispositivos móveis, que permite que aos usuários do transporte obtenham benefícios das informações de rastreamento por meio de requisições ao módulo Servidor.

Nas seções a seguir é dada uma definição mais detalhada dos três componentes e de suas interações, tendo como ponto de partida a interface com o usuário final, ou seja, o módulo cliente.

4.3.1.1. Módulo Cliente

Sendo uma das três partes fundamentais da aplicação, o módulo Cliente representa a interface com os usuários. É através do módulo Cliente, o qual é uma aplicação para dispositivos móveis, que o usuário poderá acompanhar, em tempo real, os veículos do transporte público.

A aplicação Cliente deve permitir ao usuário, como serviços básicos, localizar determinada linha de ônibus através de um sistema de busca e permitir que ele visualize, em um mapa, a posição dos veículos desta linha em tempo real. A busca por uma linha pode utilizar os seguintes critérios:

- o usuário simplesmente busca uma linha pelo nome ou parte do nome que identifica a linha;
- o usuário pode marcar um determinado ponto no mapa e buscar por linhas que passem próximas a ele;
- caso o dispositivo móvel utilizado tenha recursos tecnológicos que permitam obter a posição do usuário (como o GPS), ele pode refinar os métodos de busca levando em consideração sua posição atual, como no caso de buscar por linhas próximas a ele.

A partir de uma linha selecionada, a aplicação deve permitir a visualização do itinerário, dos pontos de parada, assim como a posição atual dos veículos (ônibus, lotação, etc.) que pertencem a esta linha, sendo esta posição atualizada a uma taxa constante, a qual poderá ser configurada pelo usuário.

Até aqui foram definidos diferentes meios pelos quais o usuário pode localizar uma determinada linha e visualizar a posição de seus veículos em tempo real. Este serviço, por si só, já se apresenta como algo bastante útil. Mas o propósito desta aplicação é ir um pouco além, permitindo que o usuário possa cadastrar eventos espaciais pelos quais deseja ser avisado. No contexto de eventos espaciais relacionados ao domínio da aplicação, a aproximação de um veículo de um ponto específico dentro do seu itinerário, é sem dúvida um excelente ponto de partida para definir quais eventos devem ser reconhecidos pela aplicação. Sendo assim, a aplicação deve permitir que os usuários configurem eventos desta natureza, pelos quais desejam ser avisados, e a forma de aviso que desejam receber (alerta sonoro, mensagem visual, etc.). A próxima seção apresenta a definição do módulo Servidor, o qual recebe as requisições dos clientes.

4.3.1.2. Módulo Servidor

O módulo Servidor é a parte da aplicação responsável por processar e responder as requisições dos clientes. O processamento de uma requisição passa pelo acesso a informações armazenadas na base de dados e posterior processamento destas informações, para então entregar ao cliente aquilo que foi solicitado em sua requisição (por exemplo, o resultado da busca por uma linha). O módulo Servidor deve ser capaz de responder a todas as requisições necessárias para que todos os requisitos definidos na seção Módulo Cliente sejam atendidos.

4.3.1.3. A Base de Dados

Não sendo finalidade deste trabalho definir como os dados de rastreamento são obtidos, é então, assumido como verdadeira a existência de uma infraestrutura de rastreamento veicular para o transporte público e que, de alguma forma, as informações obtidas por esta infraestrutura são armazenadas na base de dados da aplicação em tempo real. Partindo dessa premissa, a base de dados da aplicação deve ser capaz de armazenar as informações referentes ao domínio da aplicação (linhas, veículos, itinerários, etc.), assim como armazenar as informações de rastreamento que chegam em tempo real. Além disso, a base de dados deve ser estruturada de maneira a responder de forma eficiente as consultas, principalmente as consultas referentes ao monitoramento de veículos, permitindo que o usuário tenha a impressão de precisão por parte do sistema.

4.3.2. Modelo de Casos de Uso

O Modelo de Casos de Uso são documentos de texto que descrevem os requisitos funcionais (comportamentais) através de um conjunto de cenários típicos do uso de um sistema [Cockburn 2001]. As tabelas a seguir apresentam o Modelo de Casos de Uso da aplicação Bus Tracker.

Tabela 1. Caso de Uso: Criar Identificador de Usuário

Escopo: aplicação Bus Traker
Nível: objetivo do Sistema
Ator Principal: Sistema
Interessados e Interesses: - Sistema: identificar de maneira única o usuário. - Usuário: ser identificado pelo sistema.
Precondições: o Usuário instalou o módulo Cliente da aplicação em seu dispositivo móvel, mas ainda não o executou.
Garantia de Sucesso: o Sistema criou um identificador único para o usuário. O usuário recebeu seu identificador.
Cenário de Sucesso Principal: 1. Usuário inicia o módulo Cliente (Sistema) instalado em seu dispositivo móvel. 2. Sistema cria um identificador único para o usuário. 3. Sistema informa ao Usuário que ele foi identificado e o apresenta o identificador.
Extensões: 2a. O Sistema falha ao criar o identificador do Usuário 1. Sistema informa a ocorrência de uma falha. 2. Usuário finaliza a operação.

Tabela 2. Caso de Uso: Selecionar Linha

Escopo: aplicação Bus Traker
Nível: objetivo do Usuário
Ator Principal: Usuário
Interessados e Interesses: - Usuário: selecionar uma linha de transporte.
Precondições: o Sistema tem conhecimento das linhas de transporte, com seus itinerários e pontos de parada.
Garantia de Sucesso: o Usuário encontra e seleciona uma linha.
Cenário de Sucesso Principal: 1. Usuário informa ao Sistema que deseja buscar uma linha. 2. Sistema solicita ao Usuário que indique os critérios de busca. 3. Usuário inseri o nome ou parte do nome da linha desejada. 4. Sistema busca por linhas que se enquadram no critério de busca do Usuário (nome ou parte do nome da linha). 5. Sistema apresenta o resultado. 6. Usuário repete os passos 2-4 até encontrar a linha desejada. 7. Usuário seleciona uma linha.
Extensões: *a. A qualquer momento, Usuário cancela a operação. 2a. Usuário seleciona um ponto no mapa para ser usado como critério na busca por linha.

1. Usuário informa ao Sistema que deseja selecionar um ponto no mapa da cidade.
2. Sistema apresenta o mapa da cidade.
3. Usuário seleciona um ponto no mapa.
4. Sistema questiona a distância máxima em relação ao ponto selecionado, para ser considerada na busca.
5. Usuário define o raio de distância máximo do ponto selecionado.
6. Usuário continua a busca (podendo inserir parte do nome da linha).
- 2b. Usuário utiliza a própria posição como ponto de referência para a busca.
 1. Usuário informa ao Sistema que deseja usar sua posição como critério na busca.
 2. Sistema tenta obter a posição do usuário, solicitando ao Sistema de Posicionamento.
 - 2a. Sistema obtém a posição do Usuário.
 1. Sistema recebe informação de posicionamento.
 2. Sistema informa ao Usuário que obteve a posição dele, e questiona a distância máxima em relação a esta posição, para ser considerada na busca.
 3. Usuário define o raio de distância máximo da sua posição.
 4. Usuário continua a busca (podendo inserir parte do nome da linha).
 - 2b. Sistema NÃO obtém a posição do Usuário.
 1. Sistema é avisado, pelo Sistema de Posicionamento, que houve falha na obtenção da posição do Usuário.
 2. Sistema informa ao Usuário o problema.
 3. Usuário finaliza a operação.
 - 4a. Sistema falha na busca por linhas.
 1. Sistema informa ao Usuário a falha na busca por linhas.
 - 1a. Sistema tenta novamente a busca.
 1. Usuário solicita ao Sistema que ele tente novamente.
 2. Sistema tenta executar novamente a busca.
 - 2a. Usuário interrompe a busca.
 1. Usuário finaliza a operação.
 - 6a. Usuário não encontra uma linha que satisfaça seus critérios.
 1. Usuário finaliza a operação.

Tabela 3. Caso de Uso: Rastrear Veículo

Escopo: aplicação Bus Traker
Nível: objetivo do usuário
Ator Principal: Usuário
Interessados e Interesses: - Usuário: visualizar em tempo real a posição de um veículo de uma determinada linha.
Precondições: o Sistema tem conhecimento das informações de posicionamento dos veículos de transporte da linha selecionada.
Garantia de Sucesso: o Usuário visualiza em um mapa a posição, em tempo real, do veículo de uma linha por ele escolhida.
Cenário de Sucesso Principal: 1. Usuário invoca <u>Selecionar Linha</u> a fim de encontrar a linha desejada. 2. Sistema e Usuário interagem até uma linha ser selecionada.

<ol style="list-style-type: none"> 3. Usuário informa ao Sistema que deseja rastrear os veículos desta linha. 4. Sistema solicita que o Usuário informe algumas configurações necessárias ao rastreamento como: o período de atualização da posição dos veículos no mapa, se deve ser apresentado o itinerário e/ou os pontos de parada da linha. 5. Sistema busca informações de rastreamento. 6. Sistema apresenta um mapa com a posição real do ou dos veículos daquela linha, conforme configurações escolhidas pelo Usuário. 7. Sistema repete os passos 5 e 6, atualizando a posição dos veículos conforme a taxa de atualização, até que o Usuário cancele a operação.
<p>Extensões:</p> <p>*a. A qualquer momento, Usuário cancela a operação.</p> <p>2a. Usuário não encontra a linha desejada.</p> <ol style="list-style-type: none"> 1. Usuário finaliza a operação. <p>5a. Sistema falha na busca por informações de rastreamento.</p> <ol style="list-style-type: none"> 1. Sistema tenta novamente buscar informações de rastreamento. 2. Sistema repete o passo 1 em caso de falhas. <p>2a. Sistema obtém sucesso na busca por informações de rastreamento.</p> <ol style="list-style-type: none"> 1. Sistema prossegue o rastreamento buscando e apresentando as informações ao Usuário. <p>2b. Sistema falha em 3 tentativas consecutivas.</p> <ol style="list-style-type: none"> 1. Sistema informa ao Usuário falha na busca de informações de rastreamento. 2. Usuário finaliza operação.

Tabela 4. Caso de Uso: Cadastrar Evento Espacial

Escopo: aplicação Bus Traker
Nível: objetivo do usuário
Ator Principal: Usuário
<p>Interessados e Interesses:</p> <ul style="list-style-type: none"> - Sistema: armazenar os eventos cadastrados pelo Usuário de forma que eles possam ser monitorados e facilmente identificados quando da sua ocorrência. - Usuário: cadastrar um evento espacial pelo qual deseja ser avisado pelo Sistema.
<p>Garantia de Sucesso: o Usuário cadastra um evento espacial pelo qual deseja ser avisado quando da sua ocorrência. O Sistema armazena o evento cadastrado pelo Usuário e monitora a ocorrência deste evento.</p>
<p>Cenário de Sucesso Principal:</p> <ol style="list-style-type: none"> 1. Usuário informa ao Sistema que deseja cadastrar um evento. 2. Sistema apresenta ao Usuário os possíveis eventos espaciais a serem cadastrados. 3. Usuário seleciona o tipo de evento que deseja cadastrar. 4. Sistema apresenta ao Usuário opções de configuração conforme o tipo de evento selecionado. 5. Usuário entra com as configurações necessárias para o cadastro do evento. 6. Sistema armazena as informações de configuração do evento. 7. Sistema invoca <u>Monitorar Evento</u>. 8. Sistema informa ao Usuário que o evento foi cadastrado. 9. Usuário finaliza a operação.
<p>Extensões:</p> <p>*a. A qualquer momento, Usuário cancela a operação.</p>

- 6a. Sistema falha ao armazenar as informações de configuração do evento.
1. Sistema informa ao Usuário que houve uma falha e que o evento não pode ser cadastrado.
 2. Usuário finaliza a operação.

Tabela 5. Caso de Uso: Monitorar Evento

Escopo: aplicação Bus Traker
Nível: objetivo do Sistema
Ator Principal: Sistema
Interessados e Interesses: - Sistema: monitorar e identificar a ocorrência de eventos espaciais cadastrados.
Precondições: existe pelo menos um evento cadastrado pelo Usuário.
Garantia de Sucesso: o Sistema identifica a ocorrência de um evento espacial cadastrado pelo Usuário e o avisa da ocorrência.
Cenário de Sucesso Principal: <ol style="list-style-type: none"> 1. Sistema busca informações de eventos cadastrados pelo usuário. 2. Sistema verifica se os requisitos que definem a ocorrência de um evento foram atingidos. 3. Sistema repete os passos 1 e 2 até que a ocorrência de um evento seja identificada. 3. Sistema identifica a ocorrência de um evento. 4. Sistema invoca <u>Alertar Ocorrência de Evento</u>. 5. Sistema repete passos 2-4 até que não existam mais eventos a serem monitorados.
Extensões: <ol style="list-style-type: none"> 1a. A busca por informações de eventos cadastrados falha. <ol style="list-style-type: none"> 1. Sistema informa a ocorrência de uma falha. 2. Sistema finaliza monitoração de eventos.

Tabela 6. Caso de Uso: Alertar Ocorrência de Evento

Escopo: aplicação Bus Traker
Nível: objetivo do Sistema
Ator Principal: Sistema
Interessados e Interesses: - Sistema: alertar ao Usuário a ocorrência de um evento espacial por ele cadastrado. - Usuário: ser alertado pelo Sistema sobre a ocorrência de um evento esperado.
Precondições: o Sistema identificou a ocorrência de um evento espacial cadastrado. O Usuário pode, de alguma forma, ser alertado pelo Sistema.
Garantia de Sucesso: o Sistema informa ao Usuário a ocorrência de um evento espacial pré cadastrado.
Cenário de Sucesso Principal: <ol style="list-style-type: none"> 1. Sistema busca informações de como o Usuário deseja ser informado sobre a ocorrência do evento. 2. Sistema informa o Usuário da ocorrência do evento. 3. Usuário finaliza a operação.

4.4. Elaboração – 2ª Fase do PU

A elaboração é a fase inicial de iterações onde, em um projeto, a arquitetura central e de alto risco do software é definida e implementada. A maioria dos requisitos são

descobertos e estabilizados nesta fase, assim como, os principais riscos são mitigados. É nesta fase que alguns dos principais artefatos do PU (Modelo de Domínio, Modelo de Projeto, Modelo de Dados, Documento de Arquitetura de Software) são iniciados. As próximas seções apresentam alguns destes artefatos e suas respectivas definições no contexto de projeto da aplicação Bus Tracker.

4.4.1. Modelo de Domínio

O artefato do PU denominado de Modelo de Domínio (também chamado de Modelo Conceitual) tem como objetivo representar, de maneira visual, classes conceituais, ou objetos do mundo real, em um domínio. Ele é considerado o modelo mais importante na análise orientada a objetos, além de servir com fonte de inspiração no projeto de alguns objetos de software, e de ponto de entrada para vários outros artefatos do PU [Fowler 1997]. O Modelo de Domínio pode ser ilustrado com um conjunto de diagramas de classe (UML) em que nenhuma operação é definida. Apenas objetos do domínio (classes conceituais), seus atributos e associações entre estes objetos devem aparecer no Modelo de Domínio. A seguir é apresentado um esboço inicial para o Modelo de Domínio da aplicação Bus Tracker (Figura 1).

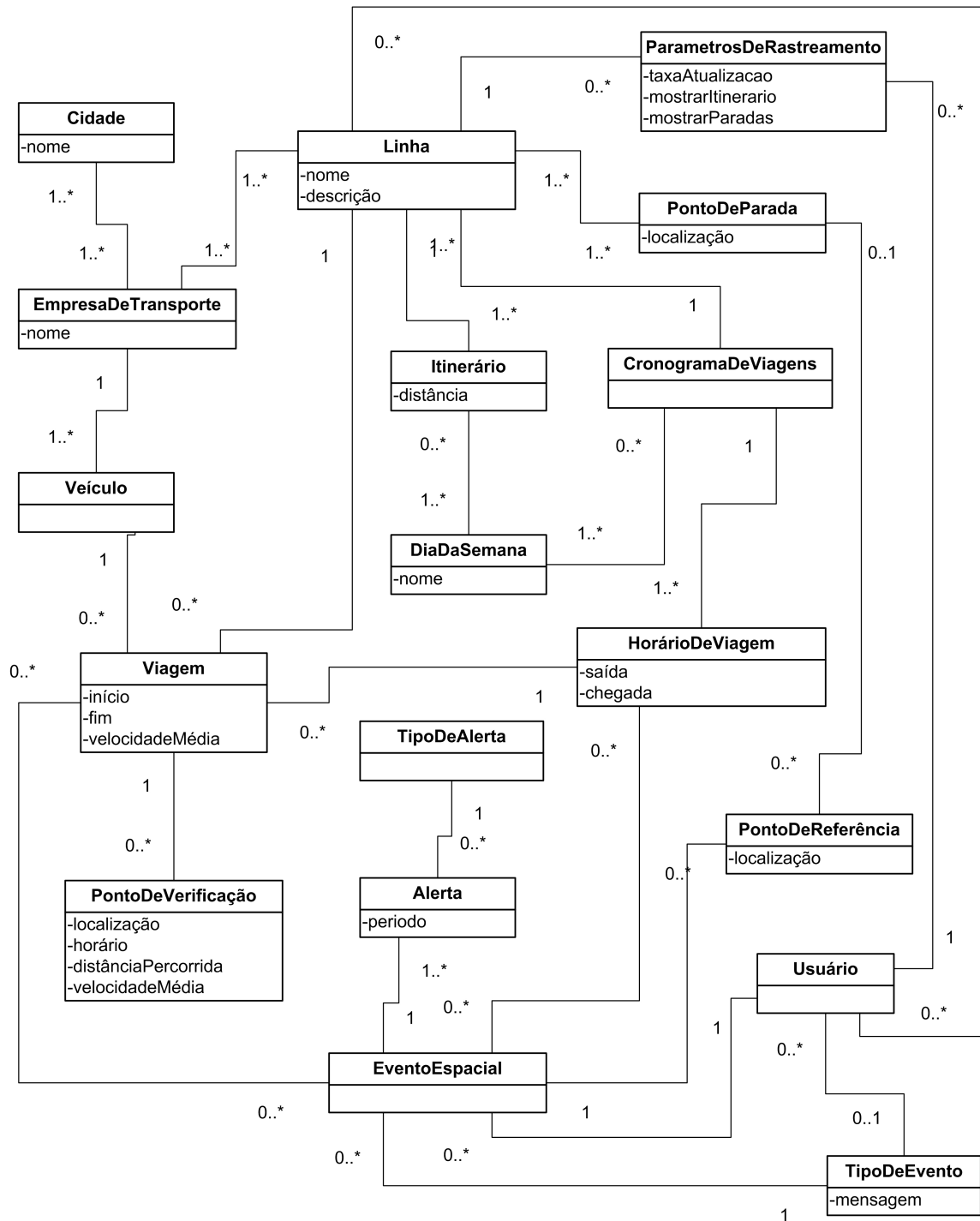


Figura 1. Modelo de Domínio

4.4.2. Diagrama de Sequência do Sistema

Durante a interação entre um ator externo e sistema, eventos de sistema são gerados, geralmente solicitando alguma operação de sistema. Os casos de uso do sistema descrevem como estes atores externos interagem com o sistema por meio de solicitações, enquanto os Diagramas de Sequência do Sistema (DSS) explicitam estas solicitações. Um DSS mostra, para um cenário específico de um caso de uso, os eventos que os atores externos geram, sua ordem e os eventos entre sistemas. A ênfase do DSS

está nos eventos que cruzam a fronteira do sistema. As figuras a seguir apresentam os DSSs da aplicação.

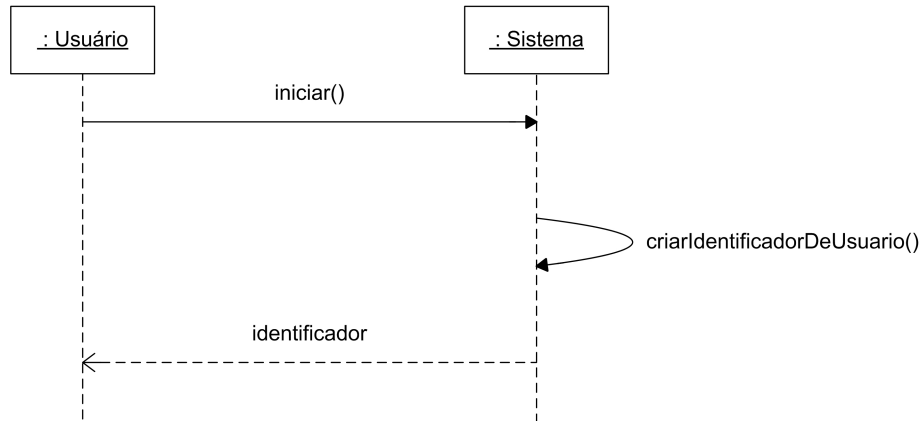


Figura 2. Digrama de Sequência do Sistema para o caso de uso Criar Identificador de Usuário

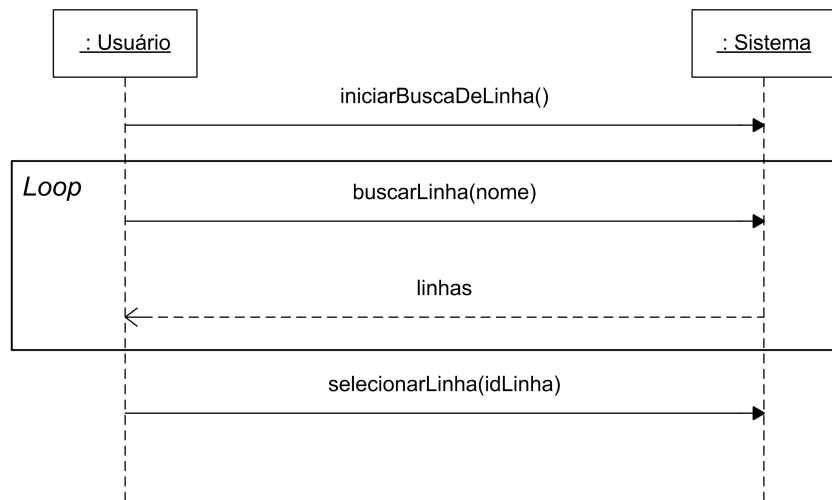


Figura 3. Digrama de Sequência do Sistema para o cenário principal do caso de uso Selecionar Linha

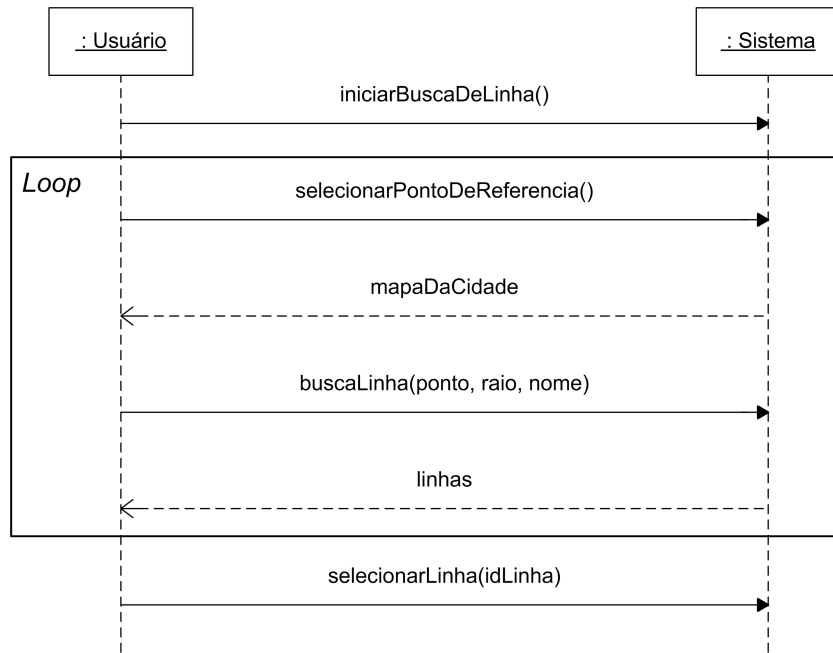


Figura 4. Digrama de Sequência do Sistema para um cenário alternativo do caso de uso Selecionar Linha

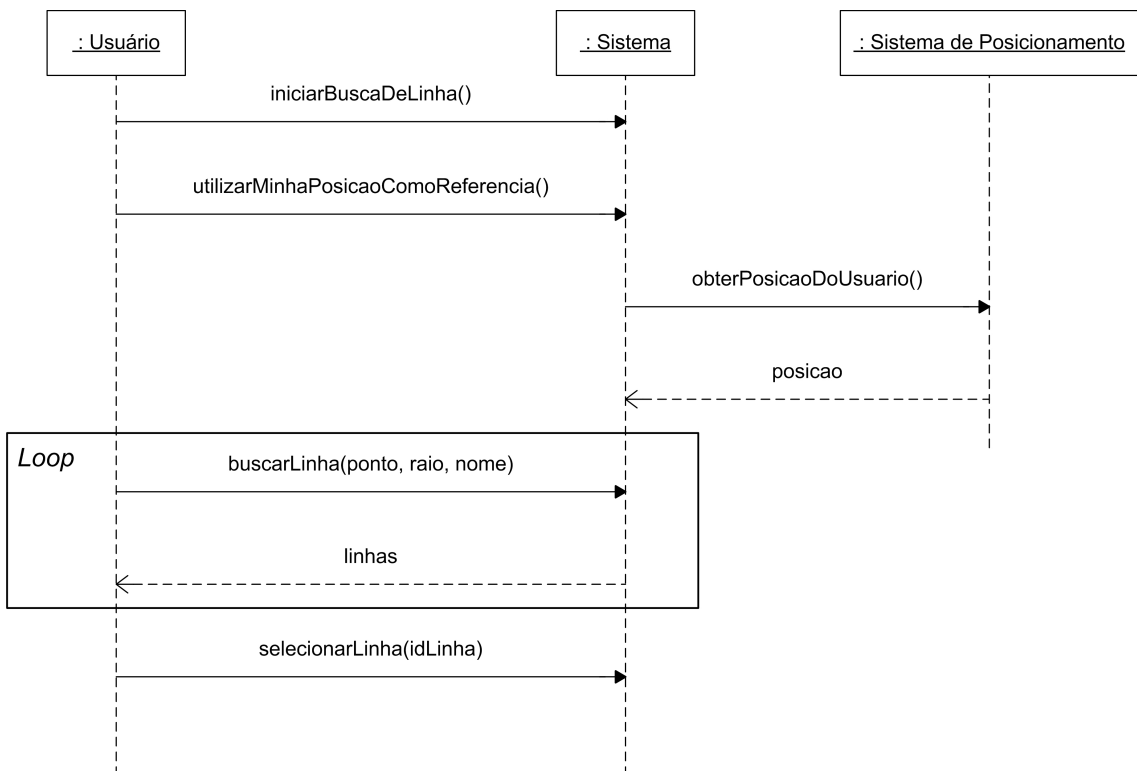


Figura 5. Digrama de Sequência do Sistema para um cenário alternativo do caso de uso Selecionar Linha

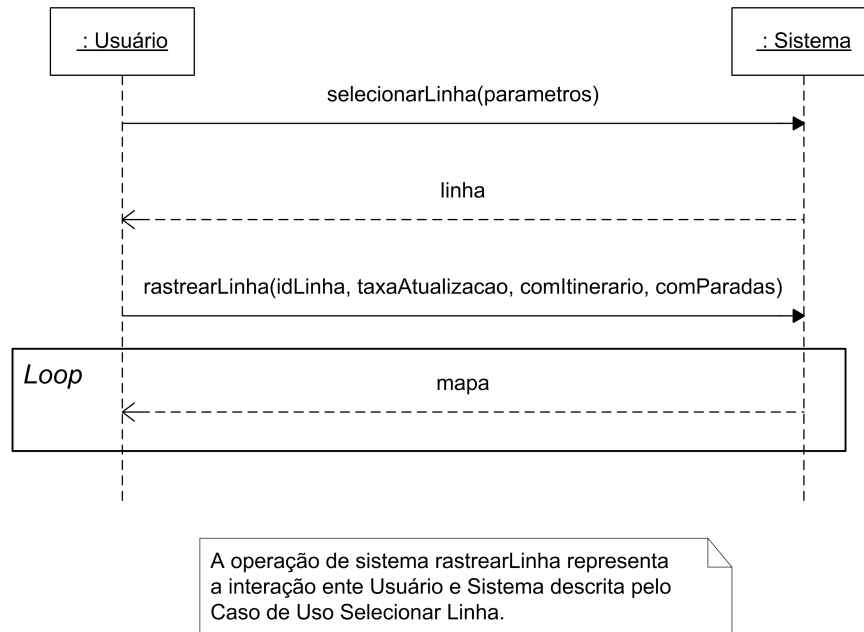


Figura 6. Digrama de Sequência do Sistema para o cenário principal do caso de uso Rastrear Veículo

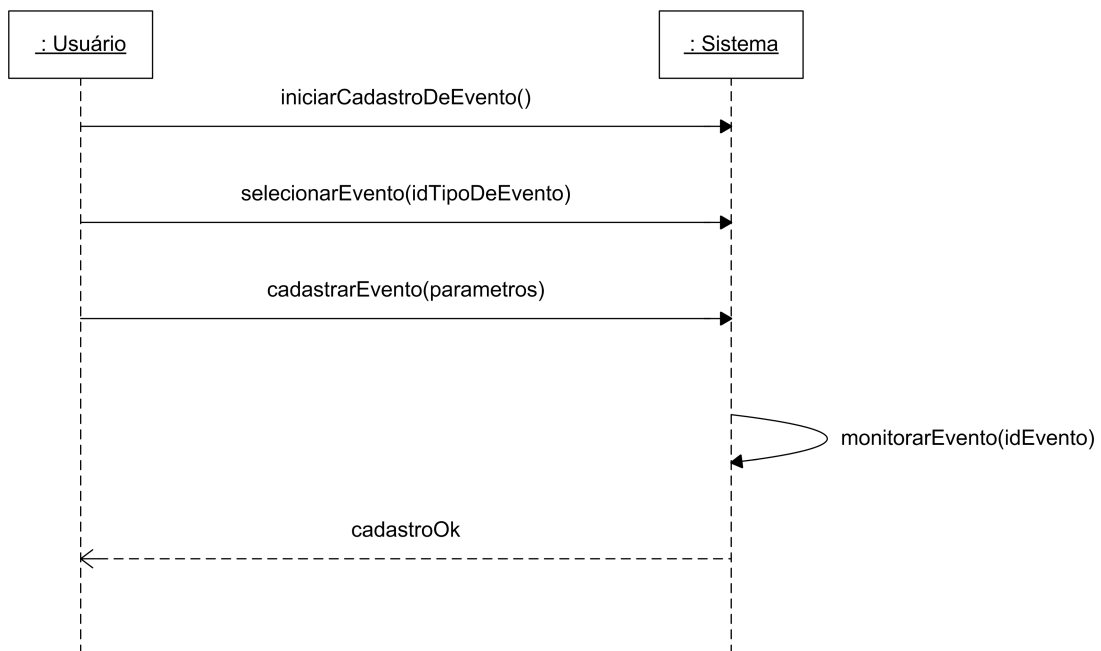


Figura 7. Digrama de Sequência do Sistema para o cenário principal do caso de uso Cadastrar Evento Espacial

4.4.3. Contrato de Operação

Contratos de Operação (CO) utilizam uma forma com seções de pós e pré-condições para descrever modificações em objetos do modelo de domínio, como resultado de uma operação do sistema. Desta forma os COs oferecem uma descrição mais detalhada e precisa do comportamento do sistema. A seção de pós-condições descreve modificações no estado dos objetos do Modelo de Domínio, tais como: a criação de instâncias, a modificação de atributos ou a formação de associações. As tabelas a seguir apresentam os Contratos de Operação para as operações do sistema que de alguma forma modificam o estado dos objetos do Modelo de Domínio da aplicação.

Tabela 7. Contrato de Operação: criarIdentificadorDeUsuario

Operação: criarIdentificadorDeUsuario
Referências Cruzadas: Casos de Uso: Criar Identificador de Usuário
Precondições: O módulo Cliente instalado no dispositivo móvel do cliente nunca foi executado.
Pós-condições: - Foi criada uma instância de Usuário.

Tabela 8. Contrato de Operação: rastrearLinha

Operação: rastrearLinha(idLinha, taxaAtualizacao, comItinerario, comParadas)
Referências Cruzadas: Casos de Uso: Rastrear Veículo
Precondições: O Usuário selecionou uma linha a partir da operação selecionarLinha.
Pós-condições: - Foi criada uma instância <i>params</i> de ParametrosDeRastreamento. - Os atributos de <i>params</i> foram iniciados: <ul style="list-style-type: none">- <i>params.taxaAtualizacao</i> tornou-se taxaAtualizacao;- <i>params.mostrarItinerario</i> tornou-se comItinerario;- <i>params.mostrarParadas</i> tornou-se comParadas.
- <i>params</i> foi associado com o Usuário corrente. - <i>params</i> foi associado com a Linha correspondente a idLinha

Tabela 9. Contrato de Operação: cadastrarEvento

Operação: cadastrarEvento(params)
Referências Cruzadas: Casos de Uso: Cadastrar Evento Espacial
Precondições: O usuário selecionou o tipo de evento a ser cadastrado.
Pós-condições: - Foi criada uma instância <i>event</i> de EventoEspacial. - <i>event</i> foi associado com o tipo de evento selecionado pelo Usuário. - <i>event</i> foi associado com o Usuário corrente. - Foi criada um instância <i>alert</i> de Alerta. - <i>alert</i> foi associado com o TipoDeAlerta especificado em parametros. - <i>alert</i> foi associado a <i>event</i> . - Foram feitas todas as demais associações de <i>event</i> com outros objetos (Viagem, HorárioDeViagem, PontoDeReferencia) conforme o tipo do evento e os valores recebidos em parametros.

5. Conclusão

Como pode ser visto ao longo deste trabalho, os LBSs vem ganhando destaque entre os serviços conscientes de contexto, pois eles tornam-se cada vez mais populares à medida que as tecnologias necessárias para a sua criação ficam mais baratas. É neste contexto, e no contexto de mobilidade que este trabalho buscou apresentar a definição e o projeto parcial de um LBS que ofereça uma melhor experiência aos usuários do transporte público. Os resultados parciais deste projeto serão aperfeiçoados e incrementados, resultando em um projeto completo e na implementação de um protótipo para a aplicação Bus Tracker. As atividades relativas a continuidade deste trabalho pertencem ao escopo do Trabalho de Graduação 2 (TG2), e serão brevemente

descritas em um plano de atividades. As atividades serão divididas em iterações conforme definido pelo PU, sendo que, ao final de cada iteração, uma parcela da aplicação estará finalizada, testada e pronta para ser utilizada. Desde as primeiras iterações será possível utilizar a aplicação, pois um ou mais casos de uso estarão prontos ao final de cada iteração. O tempo total de duração de todas as iterações não deverá ultrapassar 90 dias; tempo aproximado para realização do TG2. A tabela a seguir apresenta o plano de atividades para cada iteração e o tempo de duração destas.

Tabela 10. Plano de Atividades

<p>Iteração: 1 Atividades: - Escolha de um Sistema de Gerenciamento de Base de Dados (SGBD) adequado, levando em consideração a necessidade de armazenamento de dados relativos ao domínio espacial, assim como a realização de operações sobre dados deste domínio. - Projeto físico da base de dados. - Implementação e testes da base de dados. Tempo de duração: 7 dias.</p>
<p>Iteração: 2 Atividades: - Escolha das tecnologias a serem utilizadas no módulo Servidor (PHP, Java, etc.) e no módulo Cliente (Android, iOS, etc.). - Definição de como se dará a comunicação entre Cliente e Servidor (tecnologias, protocolos). Tempo de duração: 5 dias.</p>
<p>Iteração: 3 Atividades: - Implementação da arquitetura central e estabelecimento comunicação entre os módulos Cliente e Servidor. Tempo de duração: 10 dias.</p>
<p>Iteração: 4 Atividades: - Projeto de objetos, implementação e testes do caso de uso <i>Criar Identificador de Usuário</i>. Tempo de duração: 7 dias.</p>
<p>Iteração: 5 Atividades: - Projeto de objetos, implementação e testes do caso de uso <i>Selecionar Linha</i>. Tempo de duração: 10 dias.</p>
<p>Iteração: 6 Atividades: - Projeto de objetos, implementação e testes do caso de uso <i>Rastrear Veículo</i>. Tempo de duração: 10 dias.</p>
<p>Iteração: 7 Atividades: - Projeto de objetos, implementação e testes do caso de uso <i>Cadastrar Evento Espacial</i>. Tempo de duração: 10 dias.</p>
<p>Iteração: 8</p>

Atividades:

- Projeto de objetos, implementação e testes do caso de uso *Monitorar Evento*.

Tempo de duração: 10 dias.

Iteração: 9**Atividades:**

- Projeto de objetos, implementação e testes do caso de uso *Alertar Ocorrência de Evento*.

Tempo de duração: 10 dias.

Referência Bibliográfica

- 3GPP, 3rd Generation Partnership Project, <http://www.3gpp.org/>, Acesso em: Maio 2011.
- Alfacomp Automação Industrial, <http://www.alfacomp.ind.br/site/default.asp>, Acesso em: Abril 2011.
- Auckland Transport, Real time passenger information, <http://www.aucklandtransport.govt.nz/Pages/default.aspx>, Acesso em: Março 2011.
- CITFOR (2004), Autarquia Municipal de Trânsito, Serviços Públicos e Cidadania de Fortaleza, www.amc.fortaleza.ce.gov.br, Acesso em: Março 2011.
- Cittati Tecnologia, <http://www.cittati.com.br>, Acesso em: Março 2011.
- Cockburn, A. (2001), Writing Effective Use Cases. Addison-Wesley.
- Dey, A. K., G. D. Abowd. (1999), Towards a better Understanding of Context and Context-Awareness, Technical Report GIT-GVU-99-22, Georgia Institute of Technology.
- EPTC, Empresa Pública de Transporte e Circulação, www.eptc.com.br, Acesso em: Abril 2011.
- Fowler, Martin (1997), Analysis Patterns: Reusable Object Models, Addison-Wesley.
- G3VB, G3VB Sistemas e Tecnologia, <http://www.g3vb.com.br>, Acesso em: Março 2011.
- GeoSIT, Informações do projeto Geosit, www.geosit.com.br, Acesso em: Abril 2011.
- GPS, Global Positioning System, <http://www.gps.gov/>, Acesso em: Abril 2011.
- HeLMi, Project HeLMi, http://www.hel2.fi/liikenteenohjaus/eng/pt_telematics.asp, Acesso em: Março 2011.
- Heuser, Carlos Alberto (2009), Projeto de Banco de Dados, Bookman, 6a edição.
- Hightower, J., G. Borriello. (2001a), Location Systems for Ubiquitous Computing. IEEE Computer, Vol. 34, No. 8. August (2001), 57–66.
- Hightower, J., G. Borriello. (2001b). Location Sensing Techniques. Companion Report to the IEEE Computer article on Location Systems for Ubiquitous Computing. University of Washington, Computer Science and Engineering.
- Kupper, Axel (2005), Location-based Services: Fundamentals and Operation, John Wiley & Sons Ltd.
- Kruchten, P. (2000), The Rational Unified Process – An Introduction, Addison-Wesley, 2nd edition.
- Larman, Craig (2004), Applying UML and Patterns: An Introduction to Object-Oriented Analysis and Design and Iterative Development, Prentice Hall, 3rd edition.
- Magalhães, Caroline Tristão de Alencar (2008), Avaliação de Tecnologias de Rastreamento por GPS para Monitoramento do Transporte Público por Ônibus, Dissertação - Universidade Federal do Rio de Janeiro, COPPE.

- Magela, Rogerio (2006), Engenharia de Software Aplicada: Princípios (volume 1), Alta Books.
- Maxtrack Industrial Ltda, <http://www.maxtrack.com.br/site/home>, Acesso em: Março 2011.
- NextBus, NextBus Inc, <http://www.nextbus.com/homepage>, Acesso em: Março 2011.
- OGC (2001), OpenGIS Implementation Specification: Coordinate Transformation Services, <http://www.opengis.org/>.
- OGC (2004), OpenGIS Location Services (OpenLS): Core Services. <http://www.opengis.org/>
- Ônibus Legal, Projeto Ônibus Legal, <http://www.onibuslegal.com>, Acesso em: Março 2011.
- Open Geospatial Consortium, <http://www.opengeospatial.org/>,
- Open Mobile Alliance, <http://www.openmobilealliance.org/>, Acesso em: Maio 2011.
- Rigaux, P., M. Scholl, A. Voisard. (2002). Spatial Databases with Application to GIS. Morgan Kaufmann Publishers.
- SIT, Sistema Integrado de Transporte, www.uberlandia.mg.gov.br, Acesso em: Abril 2011.
- Wplex, Wplex Software Ltda, <http://www.wplex.com.br>, Acesso em: Março 2011.