

**UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ
CÂMPUS PATO BRANCO
CURSO SUPERIOR DE TECNOLOGIA EM ANÁLISE E DESENVOLVIMENTO DE
SISTEMAS**

EVALDO AUGUSTO BIANCHI

**SISTEMA PARA CONTROLE DE FROTAS DO TRANSPORTE COLETIVO COM
ACESSIBILIDADE PARA DEFICIENTES VISUAIS**

TRABALHO DE CONCLUSÃO DE CURSO

**PATO BRANCO
2014**

EVALDO AUGUSTO BIANCHI

**SISTEMA PARA CONTROLE DE FROTAS DO TRANSPORTE COLETIVO COM
ACESSIBILIDADE PARA DEFICIENTES VISUAIS**

Trabalho de Conclusão de Curso apresentado ao Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas da Universidade Tecnológica Federal do Paraná, Campus Pato Branco, como requisito parcial para obtenção do título de Tecnólogo.

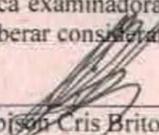
Orientador: prof. Robison Cris Brito

**PATO BRANCO
2014**

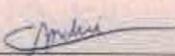
ATA Nº: 227

DEFESA PÚBLICA DO TRABALHO DE DIPLOMAÇÃO DO ALUNO EVALDO AUGUSTO BIANCHI.

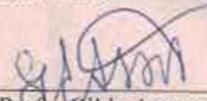
Às 16:05 hrs do dia 20 de fevereiro de 2014, Bloco V da UTFPR, Câmpus Pato Branco, reuniu-se a banca avaliadora composta pelos professores Robison Cris Brito (Orientador), Rúbia E. O. Schultz Ascari (Convidada) e Gilda Aparecida de Assis (Convidada), para avaliar o Trabalho de Diplomação do aluno Evaldo Augusto Bianchi, matrícula 1167995, sob o título **Sistema para Controle de Frotas de Transporte Coletivo com Acessibilidade para Deficientes Visuais**; como requisito final para a conclusão da disciplina Trabalho de Diplomação do Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas, COADS. Após a apresentação o candidato foi entrevistado pela banca examinadora, e a palavra foi aberta ao público. Em seguida, a banca reuniu-se para deliberar considerando o trabalho **APROVADO**. Às 16:36 hrs foi encerrada a sessão.



Prof. Robison Cris Brito, M.Sc.
Orientador



Prof. Andrei Carmiel, Esp.
Convidado



Prof. Gilda Aparecida de Assis, Dr.
Convidada

RESUMO

BIANCHI, Evaldo Augusto. Sistema para Controle de Frotas do Transporte Coletivo com Acessibilidade para Deficientes Visuais. 2014. 53 p. Trabalho de Conclusão de Curso - Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas, Universidade Tecnológica Federal do Paraná. Pato Branco, 2014.

Usuários do transporte coletivo que possuem alguma deficiência visual costumam encontrar dificuldades na simples tarefa de se locomover pela cidade onde vive, muitas vezes dependendo da ajuda de estranhos ou de um acompanhante. Buscando melhorar esta situação afim de permitir a autonomia destes usuários, o presente trabalho apresenta o desenvolvimento de um sistema para o controle do transporte coletivo com foco em acessibilidade, permitindo que o usuário possa solicitar qual o transporte mais próximo do seu local. Por ser de fácil acesso, possuir vários recursos e ferramentas que facilitam o desenvolvimento, o sistema Android foi escolhido para implementação do aplicativo.

Palavras chave: Transporte Coletivo. Deficiência Visual. Controle Transporte. Android.

ABSTRACT

BIANCHI, Evaldo Augusto. Sistema para Controle de Frotas do Transporte Coletivo com Acessibilidade para Deficientes Visuais. 2014. 53 p. Trabalho de Conclusão de Curso - Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas, Universidade Tecnológica Federal do Paraná. Pato Branco, 2014.

Transit riders who have a visual impairment , often encounter difficulties in the simple task of getting around the city where he lives , often depending on the help of strangers or a companion . Seeking to improve this situation , and allow the autonomy of users, this paper presents the development of a system for the control of public transport with a focus on accessibility, allowing the user to ask where the nearest transport of your site. Being easily accessible , have several features and tools that facilitate the development , the Android system was chosen for application deployment .

Keywords: Mass Transportation. Visual disability. Transport control. Android.

LISTA DE FIGURAS

Figura 1 - Demonstração do sistema BusAlert.	16
Figura 2 - Crescimento Estimado de Vendas de Aparelhos com Android.....	18
Figura 3 - Java SE.....	21
Figura 4 – AVD	22
Figura 5 – Eclipse	23
Figura 6 - Android SDK Manager	24
Figura 7 - NetBeans IDE	26
Figura 8 - Comparação entre código com e sem AndroidAnnotations.	27
Figura 9 - Visual Paradigm UML Community Edition.....	29
Figura 10 - Funcionamento do sistema	31
Figura 11 - Casos de uso	33
Figura 12 - Aplicativo Localizador	36
Figura 13 - Ativar GPS	37
Figura 14 - Configurações e listagem de locais.....	38
Figura 15 - Servidor.....	39
Figura 16 - Aplicativo Assistente de Transporte	40
Figura 17 - Configurações do assistente de transporte	40
Figura 18 - Visualização do mapa	41
Figura 19 - Resultado da busca	42

LISTA DE TABELAS

Tabela 1 - Materiais utilizados para o desenvolvimento do sistema	20
Tabela 2 - Versões do Android.....	25
Tabela 3 - Tabela de Localização	34
Tabela 4 - Tabela de Configurações	34
Tabela 5 - Tabela de localizações no servidor.....	35
Tabela 6 - Tabela de configurações no aplicativo cliente	35

LISTAGENS DE CÓDIGO

Listagem de Código 1 - Exemplo de GSON	28
Listagem de Código 2 - Classe Localizar	43
Listagem de Código 3 – Tratar a Localização	43
Listagem de Código 4 - Enviar para o servidor	44
Listagem de Código 5 - Gerar GSON	44
Listagem de Código 6 - Recebimento dos dados no servidor	45
Listagem de Código 7 - Cálculo da distância entre pontos geográficos.....	46
Listagem de Código 8 - Tratamento da conversão de texto em voz.....	47
Listagem de Código 9 - Verificação do GPS	47
Listagem de Código 10 - Tela do mapa.....	47
Listagem de Código 11 - Exibição do mapa na tela	48
Listagem de Código 12 - Adicionar marcador ao mapa.....	48

LISTA DE SIGLAS

ADT	Android Development Tools
API	Application Programming Interface
APP	Application
AVD	Android Virtual Device
GPS	Global Positioning System
IBGE	Instituto Brasileiro de Geografia e Estatística
ID	Identity
IDE	Integrated Development Environment
IP	Internet Protocol
JSON	JavaScript Object Notation
JVM	Java Virtual Machine
OMS	Organização Mundial da Saúde
SD	Secure Digital
SDK	Software Development Kit
SE	Standard Edition
SGDB	Sistema de Gerenciamento de Banco de Dados
SQL	Structured Query Language
SysML	Systems Modeling Language
UML	Unified Modeling Language
XML	Extensible Markup Language

SUMÁRIO

1 INTRODUÇÃO	11
1.1 CONSIDERAÇÕES INICIAIS	11
1.2 OBJETIVOS	12
1.2.1 Objetivo Geral	12
1.2.2 Objetivos Específicos	12
1.3 JUSTIFICATIVA	13
1.4 ESTRUTURA DO TRABALHO.....	14
2 REFERENCIAL TEÓRICO	15
2.1 O TRANSPORTE PÚBLICO E SUA UTILIZAÇÃO POR PORTADORES DE DEFICIÊNCIA VISUAL.....	15
2.2 TECNOLOGIA ANDROID E O DESENVOLVIMENTO PARA SMARTPHONES.....	17
3 MATERIAIS E MÉTODO	20
3.1 MATERIAIS.....	20
3.1.1 Java SE	20
3.1.2 ADT Bundle	21
3.1.3 SQLite.....	25
3.1.4 NetBeans IDE	26
3.1.5 Android Annotations	27
3.1.6 Google GSON	28
3.1.7 Visual Paradigm Community Edition	28
3.2 MÉTODO	29
4 SISTEMA DESENVOLVIDO.....	31
4.1 APRESENTAÇÃO DO SISTEMA	31
4.1.1 Modelagem do Sistema	32
4.1.2 Persistência dos dados	34
4.1.2.1 Aplicativo localizador	34
4.1.2.2 Servidor	35
4.1.2.3 Aplicativo Cliente.....	35

4.2 DESCRIÇÃO DO SISTEMA	36
4.2.1 APLICATIVO LOCALIZADOR	36
4.2.2 APLICATIVO SERVIDOR.....	38
4.2.3 APLICATIVO CLIENTE	39
4.3 IMPLEMENTAÇÃO DO SISTEMA	42
5 CONCLUSÃO.....	49
REFERÊNCIAS	50

1 INTRODUÇÃO

Neste capítulo serão apresentadas as considerações iniciais, com o contexto no qual se insere a proposta deste trabalho. Também são apresentados os objetivos, a justificativa e a estrutura do trabalho.

1.1 CONSIDERAÇÕES INICIAIS

Há muito tempo, um dos grandes desafios do avanço tecnológico é a capacidade de criar ferramentas que possam ser utilizadas sem dificuldade por qualquer pessoa, e que estas ferramentas sejam acessíveis, independente da condição do usuário.

No mundo moderno, onde a Internet está disponível para todos e os dispositivos móveis estão cada vez mais inseridos na sociedade, incluir pessoas com deficiências físicas, em especial, deficiências visuais é um grande desafio, já que estas tecnologias se baseiam cada vez mais no aspecto visual para trazer uma melhor experiência para o usuário, tornando a inclusão um desafio ainda maior.

Neste cenário, a acessibilidade pode ter um importante significado. De acordo com a OMS (Organização Mundial da Saúde), em 2002 havia cerca de 161 milhões de deficientes visuais em todo o mundo, o equivalente a 2,6% da população, sendo que no Brasil eram mais de dois milhões de pessoas com algum nível de inabilidade visual. Em 2010, segundo o censo realizado pelo IBGE (Instituto Brasileiro de Geografia e Estatística), no Brasil cerca de 37,5 milhões de pessoas, ou seja, 18,8% da população possui alguma dificuldade para enxergar. Atualmente boa parte dos equipamentos não são criados pensando nos deficientes visuais.

Os smartphones, um desses equipamentos, se tornaram verdadeiros computadores portáteis, contando com recursos como uma rede de comunicação de dados de alta velocidade, podendo ser acessada até mesmo em regiões rurais. É possível ainda executar músicas, vídeos, games e softwares. Têm-se também vários tipos de hardware integrados nos smartphones, como câmeras digitais, filmadoras, receptores de GPS (*Global Positioning System*), sensores de movimento, etc.

Com todos estes recursos, os smartphones podem executar diversos tipos de aplicativos, permitindo a inclusão digital, mesmo de pessoas que possuam baixa renda e ainda necessidades especiais. Estas pessoas podem utilizar os recursos dos aparelhos, como o acesso

a rede Internet, a utilização de GPS, reprodução de dados multimídia, utilizando de sensores, tudo isso para melhorar sua qualidade de vida e permitir acesso à informação que até então só era possível a usuários de computadores.

Apesar dos grandes avanços na área de acessibilidade para os deficientes visuais, assim como para outros tipos de deficiência, a simples tarefa de utilizar o transporte público pode se tornar um grande desafio. Em muitas situações, o passageiro é obrigado a perguntar para as demais pessoas no ponto de ônibus qual o próximo veículo que passa pela linha, e em situações piores, onde não há ninguém no ponto, o passageiro precisa parar o ônibus para conseguir identificar a linha (CARVALHO, 2013).

Assim, o presente trabalho visa apresentar a proposta de desenvolvimento de uma solução para auxiliar pessoas com deficiência visual a utilizar transportes coletivos, permitindo que estes usuários tenham autonomia para identificar qual transporte está próximo a ele e se vai ao destino que deseja.

1.2 OBJETIVOS

O objetivo geral se refere à finalidade principal de realização deste trabalho. Os objetivos específicos complementam o objetivo geral.

1.2.1 Objetivo Geral

Desenvolver um sistema móvel para localização de veículos do transporte público, que facilite a utilização deste por deficientes visuais.

1.2.2 Objetivos Específicos

Dentre os objetivos específicos do trabalho, destacam-se:

- Estudar e desenvolver um sistema de localização utilizando o GPS de smartphones e o sistema Android;
- Disponibilizar acesso aos dados de um servidor, no qual se encontram os dados referentes a localização do transporte coletivo, á dispositivos móveis.

- Estudar o funcionamento do transporte coletivo;
- Estudar e implementar a sintetização de voz para a apresentação dos dados aos usuários do sistema móvel;

1.3 JUSTIFICATIVA

O sistema a ser modelado e desenvolvido pretende auxiliar as pessoas que possuem deficiência visual, seja ela total ou parcial, para que possam utilizar com autonomia os serviços de transporte coletivo sem a necessidade da ajuda de terceiros.

Assim, pessoas com este tipo de deficiência poderão usufruir do serviço de maneira igual aos demais passageiros, tornando o sistema proposto uma ferramenta com grande responsabilidade social, colocando em prática o significado da palavra acessibilidade.

O software que auxiliará o usuário poderá ser executado em smartphones de baixo custo que utilizem a plataforma Android, necessitando apenas de conexão com a rede mundial de computadores – Internet e o recurso de localização global – GPS para identificar onde o dispositivo está.

Dentre as vantagens, pode-se destacar a inclusão da pessoa com deficiência visual, permitindo que esta tenha condição para ir e vir com segurança e autonomia, pois o aplicativo deverá fornecer as informações que precisa para realizar a tarefa de se locomover utilizando um serviço público, que todos os cidadãos tem direito a utilizar. Além, é claro, do crescimento e desenvolvimento tecnológico para a cidade e o enriquecimento do conhecimento.

Com o aplicativo proposto, a pessoa poderá receber, através dos recursos do dispositivo, como sons e vibrações, as informações sobre qual o próximo ônibus, ou a distância do veículo mais próximo.

1.4 ESTRUTURA DO TRABALHO

Este trabalho está organizado em capítulos, dos quais este é o primeiro e apresenta as considerações iniciais referentes ao trabalho, os objetivos e a justificativa.

O Capítulo 2 contém o referencial teórico do trabalho que está centrado no funcionamento do transporte público e na acessibilidade além do desenvolvimento de aplicações para dispositivos móveis com o sistema operacional Android.

No Capítulo 3 estão os materiais e o método utilizado no desenvolvimento deste trabalho.

O Capítulo 4 apresenta o sistema desenvolvido com a demonstração das telas, do funcionamento e alguns códigos importantes.

No Capítulo 5 está a conclusão com as considerações finais do trabalho.

2 REFERENCIAL TEÓRICO

Este capítulo apresenta o referencial teórico do trabalho, focando no funcionamento do transporte coletivo e na acessibilidade, em seguida, sobre a plataforma em que o software será desenvolvido, o Android.

2.1 O TRANSPORTE PÚBLICO E SUA UTILIZAÇÃO POR PORTADORES DE DEFICIÊNCIA VISUAL

O sistema de transporte coletivo no Brasil tem um funcionamento básico e muito semelhante por todo o país. As linhas de ônibus são definidas pelas prefeituras, mais especificamente pelos setores de transporte e infraestrutura, e em alguns casos, juntamente com as empresas que prestam o serviço de transporte.

As cidades possuem vários pontos de parada, espalhados em locais estratégicos, de forma a atender a maior parte da população, e dependendo do porte da cidade, conta também com terminais para a troca de ônibus, ou seja, o passageiro pode sair de uma linha e utilizar outra com a mesma passagem.

Também possuem um horário regular para passar em cada ponto de parada, e várias linhas podem passar pelo mesmo ponto, mas isso vai depender de cada cidade, da estrutura de funcionamento do trânsito e das vias, fatores geográficos, etc.

Por se tratar de um serviço público, ele deve estar disponível e acessível a todos os cidadãos. No quesito acessibilidade, alguns progressos foram alcançados, como por exemplo, desde outubro de 2008, de acordo com a lei 10.098/2000, regulamentada pelo Decreto 5.296/2004, dentre outras especificações, fica determinado que, no Brasil, somente podem ser fabricados ônibus que sejam acessíveis a pessoas com deficiência ou com outro problema de locomoção (fonte: PLANATO.GOV.BR, 2013).

Essa determinação, com certeza contribui com o avanço nas políticas de acessibilidade, permitindo que muitos usuários cadeirantes ou com dificuldades motoras possam utilizar o transporte público.

No caso das pessoas com deficiência visual não existem leis que determinem a implantação de sistemas de acessibilidade no transporte coletivo, porém, alguns projetos já estão sendo criados e testados em algumas cidades no Brasil.

Na cidade de São Carlos, em São Paulo, foi implantado um sistema chamado BusAlert, desenvolvido pela empresa Grupo Criar, de Ribeirão Preto (São Paulo). Este sistema possui um aplicativo para celulares que avisa por meio de mensagem de texto e de voz a hora que um ônibus está se aproximando do ponto de parada. Para isso, os pontos de ônibus precisam ser previamente cadastrados e então recebem um número de identificação, o usuário informa o número do ponto de parada e o aplicativo se baseia na posição do ônibus e do ponto para informar se o transporte está próximo (REI; MALTA, 2012).

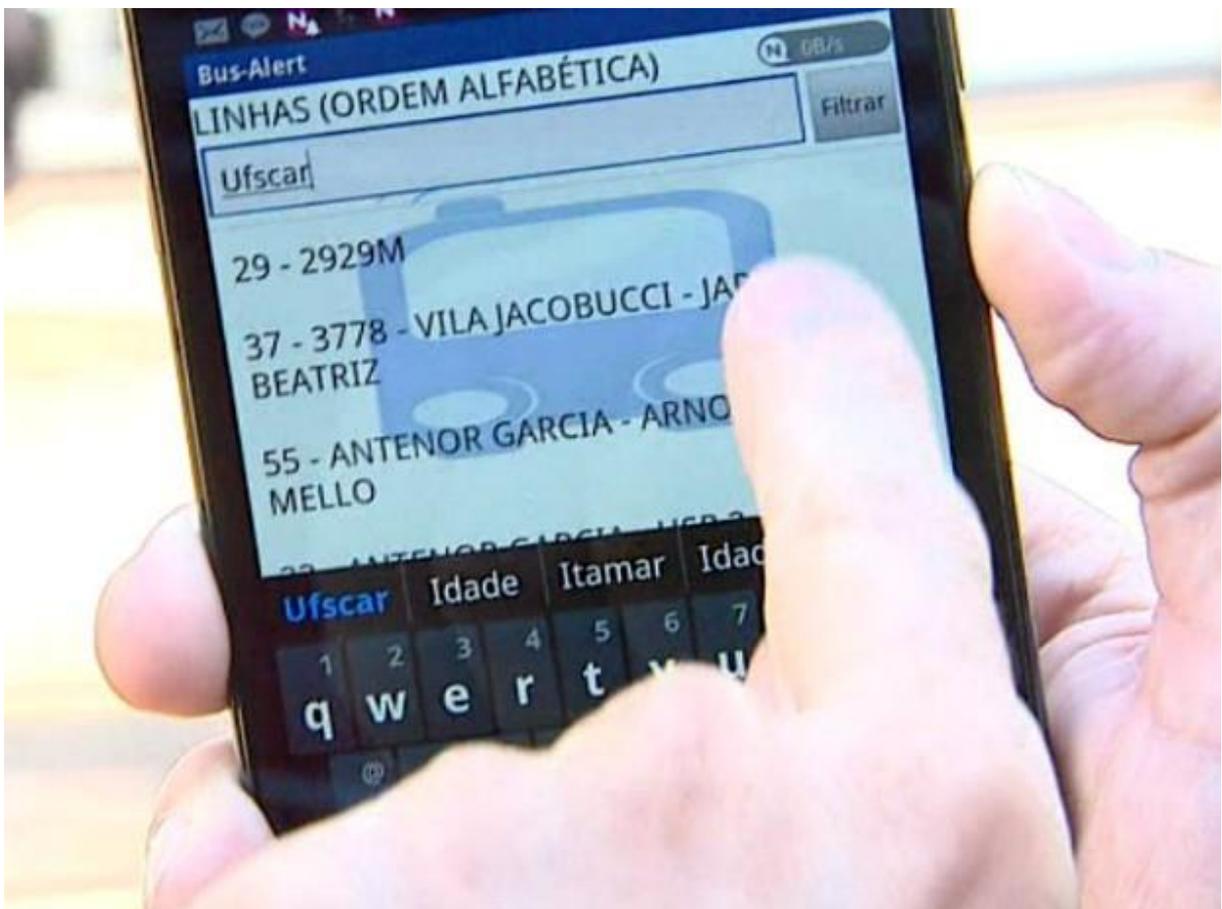


Figura 1 - Demonstração do sistema BusAlert.

Fonte: BUSALERT, 2013

Outro sistema, chamado DPS 2000, foi desenvolvido pela Universidade Federal de Minas Gerais e em 2009 a empresa Geraes Tecnologias Assistivas, licenciou a patente, que desde então, faz o desenvolvimento e comercialização do equipamento.

O sistema funciona com dois aparelhos: um transmissor, que fica com o usuário, e o receptor, instalado no ônibus. O aparelho que fica com o passageiro é do tamanho de um

celular, com uma tecla móvel, através dele, é possível cadastrar as linhas de ônibus do usuário através de menus que reproduzem áudio. Ao chegar ao ponto de embarque, o passageiro escolhe a linha no aparelho e este emite um sinal que alcança 100 metros de distância.

Quando o transporte coletivo entra na área de alcance, o sistema de áudio do receptor instalado no ônibus emite um bip avisando ao motorista que há um portador de deficiência que quer embarcar no próximo ponto. Na parada, o dispositivo sonoro, que fica instalado próximo à porta, informa repetidamente o número da linha permitindo que o deficiente visual se direcione para o local onde o veículo parou. O som só é desligado pelo passageiro, através do transmissor, após o embarque (D2, 2012).

2.2 TECNOLOGIA ANDROID E O DESENVOLVIMENTO PARA SMARTPHONES

O Android é um sistema operacional móvel que roda sobre o núcleo Linux. Sua história inicia em 2005, quando a Google adquiriu a empresa Android, Inc., que desenvolvia sistemas para celulares, três anos depois, no dia 22 de outubro de 2008, foi lançado o HTC Dream, o primeiro dispositivo a contar com este sistema operacional móvel. Desde então, o sistema não para de crescer, segundo dados do site TecMundo no ano seguinte, em 2009, foram vendidos quase 7 milhões de aparelhos, logo em seguida veio a consolidação, no ano de 2010, houve um crescimento de mais de 800% nas vendas de aparelhos de várias marcas que contavam com o Android. Desde então os números não pararam de subir (Figura 2 - Crescimento Estimado de Vendas de Aparelhos com Android.), assim como os aplicativos disponíveis para a plataforma, que no fim de 2012, de acordo com o site G1, somavam mais de 700 mil disponíveis na loja virtual *Google Play*.



Figura 2 - Crescimento Estimado de Vendas de Aparelhos com Android.

Fonte: RASMUSSEN, 2011

Uma pesquisa realizada pela empresa de consultoria Gartner, aponta que em janeiro de 2014 o sistema operacional Android no Brasil, representou 85,1% das vendas de aparelhos (GUIMARÃES, 2013).

Alguns dos motivos para sua popularização pode estar no fato de que o Android permite aos desenvolvedores escreverem software na linguagem de programação Java controlando o dispositivo via bibliotecas desenvolvidas pela Google, como também as APIs (*Application Programming Interface*) que são bastante fáceis, além da excelente documentação fornecidas pela Google.

A atualmente conhecida *Google Play*, que já se chamou *Android Market*, é a loja virtual de aplicativos para Android, onde qualquer desenvolvedor pode colocar seu APP (*Application*) para download, seja de forma gratuita ou paga, sendo necessário apenas pagar uma taxa a Google pela criação da conta e uma participação de 15% na venda dos seus softwares.

O *Google Play* tem uma política que facilita a distribuição de aplicativos pelo desenvolvedor, ao contrário da loja virtual do seu rival, a *AppleStore* com softwares para o sistema de dispositivos móvel IOS, onde os aplicativos necessitam passar pela aprovação da empresa para que possa ser disponibilizado em sua loja virtual.

A vantagem do *Google Play* é que se torna muito simples para o desenvolvedor distribuir seu software e ainda poder lucrar imediatamente com isso, porém o grande problema são os *malwares* e os sistemas de baixa qualidade, seja referente a interface visual ou de funcionalidade.

O Android hoje é um grande mercado de negócio e está em plena expansão, novas versões são lançadas constantemente, e grandes inovações e funcionalidades são incrementadas além da capacidade dos *smartphones*, cada vez com mais capacidade de processamento, armazenamento, conexões, etc., permitindo que os *softwares* possuam cada vez mais recursos e funcionalidades, que auxiliam e muito em nosso cotidiano.

3 MATERIAIS E MÉTODO

Este capítulo apresenta os materiais e o método utilizados. Os materiais se referem às tecnologias e ferramentas utilizadas para definir e implementar o sistema. O método reporta a sequência das principais atividades realizadas para desenvolver este trabalho.

3.1 MATERIAIS

Para o desenvolvimento deste projeto, foram utilizadas as ferramentas apresentadas na Tabela 1, estas detalhadas na sequência.

Tabela 1 - Materiais utilizados para o desenvolvimento do sistema

Materiais			
Software	Versão	Licença	Propriedade
Java SE	1.7.0_45-b18	Gratuito	Oracle
ADT Bundle	22.0.1-685705	Gratuito	Google
Android Plataform/ API Level	7	Gratuito	Google
SQLite	3.5.9	Gratuito	SQLite Consortium
NetBeans IDE	7.4	Gratuito	<u>Oracle</u>
Android Annotations	2.7.1	Gratuito	<u>GitHub</u>
Google GSON	2.2.4	Gratuito	Google
Visual Paradigm Community Edition	11.0	Gratuito	Visual Paradigm

3.1.1 Java SE

O Java SE, ou Java Standard Edition (ORACLE, 2013), é uma ferramenta de desenvolvimento para a plataforma Java. Contém todo o ambiente necessário para a criação e execução de aplicações Java, incluindo a máquina virtual Java ou JVM (*Java Virtual Machine*), o compilador e as APIs além de outras ferramentas utilitárias.

Está disponível para *download* no site da Oracle, empresa que gerencia a linguagem atualmente. A Figura 3 mostra a tela sobre a versão do Java.



Figura 3 - Java SE

Como o desenvolvimento para a plataforma Android exige uma IDE de desenvolvimento Java, assim como um ambiente de desenvolvimento, estes precisam da máquina virtual Java para executar, por esse motivo o Java SE deve ser instalado na máquina de desenvolvimento.

3.1.2 ADT Bundle

O ADT Bundle, ou *Android Development Tools* (DEVELOPER, 2013) foi desenvolvido pela Google para fornecer aos desenvolvedores um ambiente integrado para o desenvolvimento de aplicações para Android.

Está disponível de forma gratuita para *download* através do site para desenvolvedores (<http://developer.android.com>), já de forma integrada com a IDE (*Integrated*

Development Environment) Eclipse e com o Android SDK (*Software Development Kit*), mas também pode ser adquirido na forma de *plugin* para instalação na IDE Eclipse.

Um dos componentes principais do ADT é a máquina virtual Android, ou AVD (*Android Virtual Device*), este mostrado na Figura 4, que permite emular o sistema operacional em todas as versões disponíveis (Tabela 2 - Versões do Android), definindo inclusive modelos de aparelhos e configurações de hardware, como o tamanho da tela por exemplo. As AVDs são utilizadas para testar as aplicações desenvolvidas sem a necessidade de um aparelho real.



Figura 4 – AVD

O IDE Eclipse (2013), base para o ADT Bundle, é desenvolvido em Java, e segue o modelo *open source* de desenvolvimento. Foi iniciado pela empresa IBM que após desenvolver a primeira versão do produto, doou como *software* livre para a comunidade.

Este IDE pode ser adquirido de forma gratuita diretamente no site oficial do projeto (<http://www.eclipse.org>) neste trabalho foi utilizada na versão 4.2.2.

Atualmente, este é o IDE mais utilizado no mundo, possui amplo suporte e diversos *plug-ins*, como o ADT para o desenvolvimento Android, além de uma interface bastante funcional como demonstrado na Figura 5.

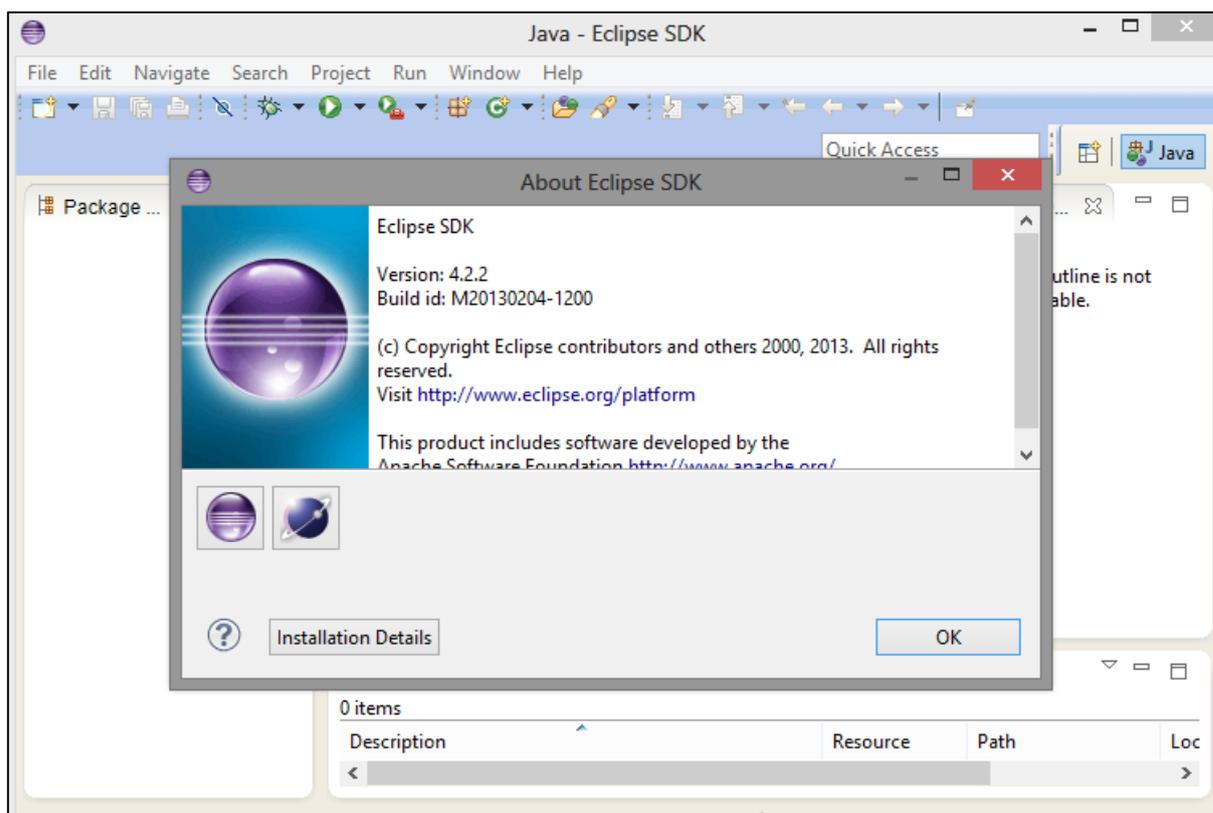


Figura 5 – Eclipse

O Android SDK, que também compõe o ADT Bundle, é desenvolvido pela Google. Este fornece as bibliotecas das APIs e ferramentas de desenvolvimento necessárias para desenvolver, testar e depurar aplicativos para o sistema Android.

Ele é composto por pacotes modulares, que contemplam as API, imagens do sistema para emular, documentação, etc. que podem ser baixados separadamente usando o Gerenciador de SDK Android. Quando as ferramentas do SDK são atualizadas ou uma nova versão da plataforma Android é lançada, é disponibilizado para baixá-los para o ambiente de desenvolvimento.

Existem vários pacotes diferentes disponíveis para o Android SDK como é mostrado na Figura 6.

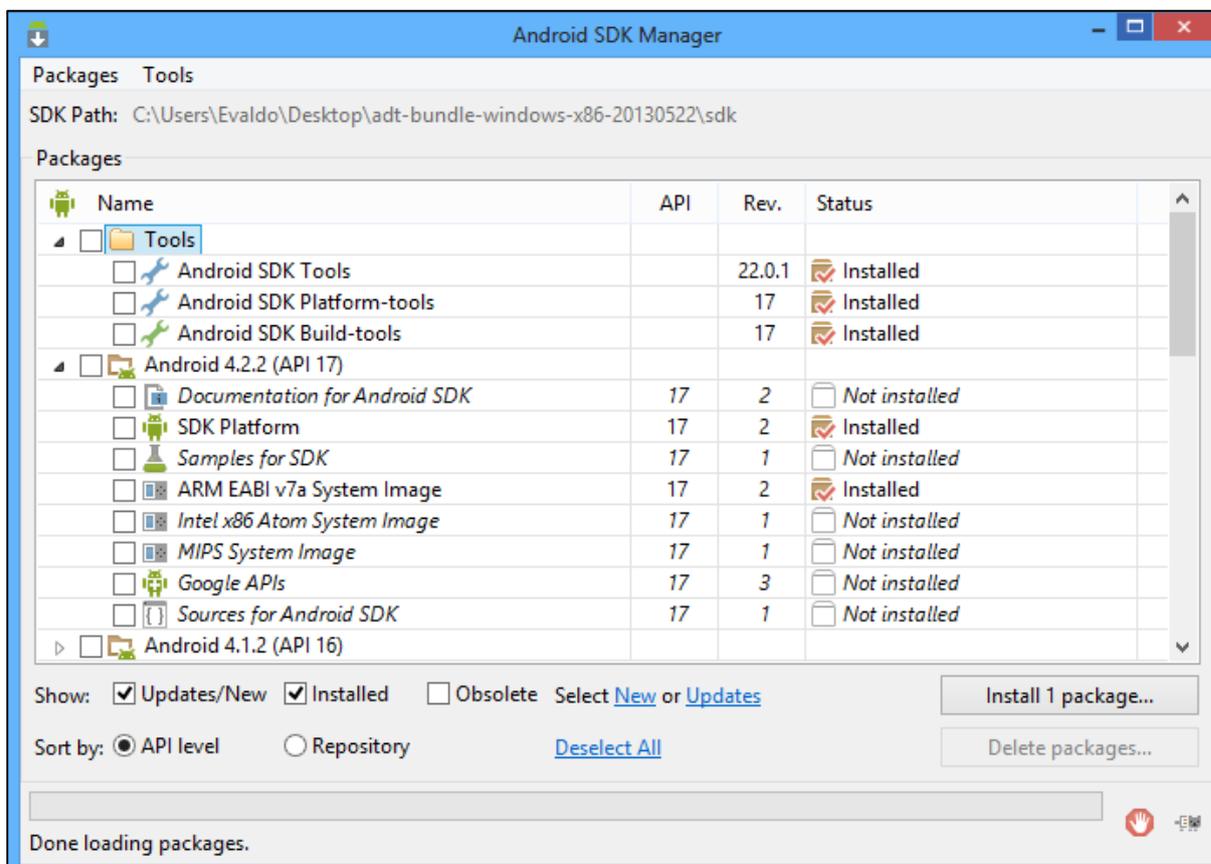


Figura 6 - Android SDK Manager

Para o desenvolvimento do aplicativo será utilizada a API ou *Plataform Level* versão 7 e também a versão 19. Cada plataforma Android tem um código para identificação, este código por sua vez, é chamado de API Level. Assim, quando são realizadas atualizações na plataforma, é gerado um novo API Level. De forma resumida, esta API são as versões do Android que existem no mercado, e quando se desenvolve, deve-se escolher a melhor opção para a versão da plataforma, para atender os dispositivos alvos.

É importante para os desenvolvedores entenderem isso, pois quando se cria um projeto é preciso definir qual a plataforma alvo. Um programa criado com a API Level 8 não deverá executar em dispositivos com APIs menores que essa.

Atualmente a última versão da API do Google é a 19, correspondente a versão 4.4 do sistema operacional Android, a versão utilizada para o desenvolvimento deste trabalho é a 7 correspondente ao Android 2.1, justamente por oferecer compatibilidade com as versões superiores garantindo que o software desenvolvido possa ser executado em praticamente todos os aparelhos com o sistema. Na Tabela 2 pode-se observar o histórico das versões do Android.

Tabela 2 - Versões do Android

Versões do Android e seus Codinomes			
Versão do Android	API level	Codinome	Ano
1.0	1	Sem codinome	2008
1.1	2	Sem codinome	2009
1.5	3	Cupcake	2009
1.6	4	Donut	2009
2.0	5	Eclair	2009
2.0.1	6	Eclair	2009
2.1	7	Eclair	2010
2.2	8	Froyo	2010
2.3	9	Gingerbread	2010
2.3.3	10	Gingerbread	2011
3.0	11	Honeycomb	2011
3.1	12	Honeycomb	2011
3.2	13	Honeycomb	2011
4.0	14	Ice Cream Sandwich	2011
4.0.3	15	Ice Cream Sandwich	2011
4.1	16	Jelly Bean	2012
4.2	17	Jelly Bean	2012
4.3	18	Jelly Bean	2013
4.4	19	KitKat	2013

3.1.3 SQLite

O sistema gerenciador de banco SQLite (2013), padrão da plataforma Android, é uma biblioteca em linguagem C que implementa um banco de dados SQL (*Structured Query Language*) embutido. Programas que usam a biblioteca SQLite podem ter acesso a banco de dados SQL sem executar um processo SGBD (Sistema de Gerenciamento de Banco de Dados) separado.

A biblioteca SQLite lê e escreve diretamente para e do arquivo do banco de dados no disco, ou no caso, em um cartão SD (*Secure Digital*). Seu uso é recomendado onde há

simplicidade da administração, implementação e manutenção são mais importantes que recursos avançados e complexos de SGBDs.

Mesmo todos os recursos ativados, o tamanho da biblioteca pode ser inferior a 500KB, dependendo da plataforma de destino e as configurações de otimização do compilador, por isso o SQLite é uma escolha de banco de dados popular na memória de dispositivos limitados, como celulares, PDAs e MP3 players. O desempenho é satisfatório mesmo em ambientes com pouca memória.

Atualmente é mantida por várias empresas parceiras, como Oracle, Adobe, Mozilla e Nokia entre outras, e disponibilizado de forma gratuita livre para qualquer fim, seja particular ou comercial.

3.1.4 NetBeans IDE

O IDE de desenvolvimento Netbeans, lançada no ano 2000 e criado pela empresa Sun Microsystems, é uma plataforma gratuita e de código aberto, que oferece aos desenvolvedores de diversas linguagens, os recursos e ferramentas necessárias para criar aplicativos profissionais de desktop, empresariais, Web e móveis multiplataformas (NETBEANS, 2013).

É possível desenvolver em linguagens como Java, C, C++, PHP, Groovy, Ruby, entre outras. Possui diversos recursos que auxiliam no desenvolvimento, como auto completar avançado, suporte a plug-ins, interface para design de aplicações desktop, suporte a banco de dados, servidor de aplicações, suporte ao Java Enterprise Edition, geração de Javadoc a partir dos comentários inseridos no código, etc. A Figura 7 demonstra a interface do NetBeans IDE.

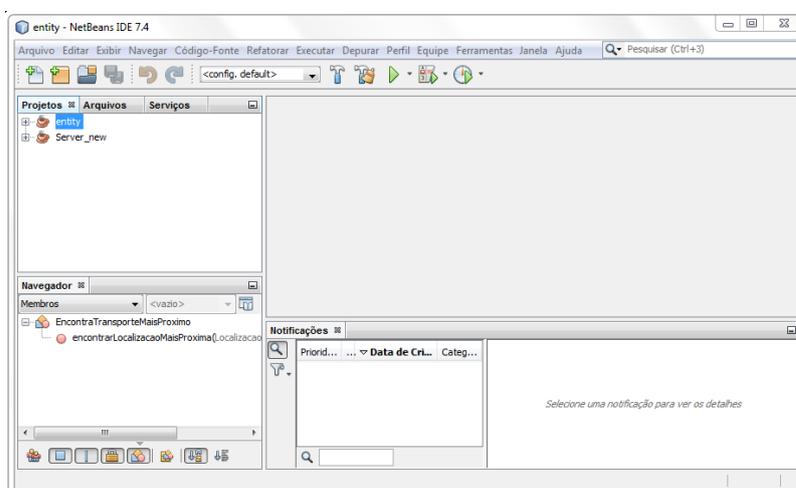


Figura 7 - NetBeans IDE

3.1.5 Android Annotations

Android annotations (2013) é um framework de código aberto e gratuito, que tem por objetivo acelerar o desenvolvimento de aplicações Android, simplificando o código e facilitando a manutenção.

Ao incluir essa biblioteca ao projeto Android, é possível utilizar anotações que serão compiladas em tempo de execução, gerando o código referente agilizando muito o desenvolvimento como demonstra a Figura 8.

Sem annotations	Com annotations
<pre>public class MyClass extends Activity { private EditText edit; private Button button; @Override protected void onCreate(Bundle savedInstanceState) { super.onCreate(savedInstanceState); setContentView(R.layout.meu_layout); edit = (EditText) findViewById(R.id.edit); button = (Button) findViewById(R.id.button); } }</pre>	<pre>@EActivity(R.layout.meu_layout) public class MyClass extends Activity { @ViewById private EditText edit; @ViewById private Button button; }</pre>

Figura 8 - Comparação entre código com e sem AndroidAnnotations. (Fonte: Github)

Uma das anotações mais utilizadas é `@Background`, que irá executar o método anotado em uma nova thread no sistema operacional, muito importante no Android para grandes processamentos de dados sem bloquear a interface gráfica.

Mas existem algumas outras, como `@AfterViews`, que executa o método seguinte na criação da activity, logo após o método `OnCreate`, padrão do Android, a `@ViewById` permite vincular um elemento do XML (*Extensible Markup Language*) da tela da activity a um objeto declarado na classe java, basta que o ID (*Identity*) seja igual ao nome do objeto ou caso sejam diferentes, adicionar entre parênteses o nome do ID do XML, por exemplo `@ViewById(R.id.IDdoComponente)`.

No site do projeto, androidannotations.org, é possível obter toda a documentação para a utilização dos recursos da biblioteca.

3.1.6 Google GSON

GSON é uma biblioteca criada e distribuída pela Google, que tem como função facilitar a conversão de objetos Java em JSON (*JavaScript Object Notation*) e vice-versa (GOOGLE-GSON, 2013).

O formato JSON, muito semelhante ao XML, é muito utilizado em sistemas que necessitam da troca de informações via Internet, como é o caso dos dispositivos móveis e do sistema desenvolvido neste trabalho.

A Listagem de Código 1 demonstra um exemplo de GSON, onde representa um objeto do tipo USER, com seus atributos e respectivos valores.

```
{
  user: {
    id: 123456,
    name: "nome exemplo",
    username: "userexemplo",
    email: "exemplo@email.com.br"
  }
}
```

Listagem de Código 1 - Exemplo de GSON

3.1.7 Visual Paradigm Community Edition

Visual Paradigm for UML (*Unified Modeling Language*) é uma ferramenta de modelagem para diagramas UML. Essa ferramenta fornece suporte para gerenciamento de casos de uso, classes, diagrama de requisitos SysML (*Systems Modeling Language*) e projeto de banco de dados com diagrama de entidades e relacionamentos organizando os fluxos de trabalho, além de gerar documentação (VISUAL PARADIGM, 2013).

Na Figura 9 pode ser observada a tela inicial do software, que para este trabalho foi utilizada a versão não comercial.

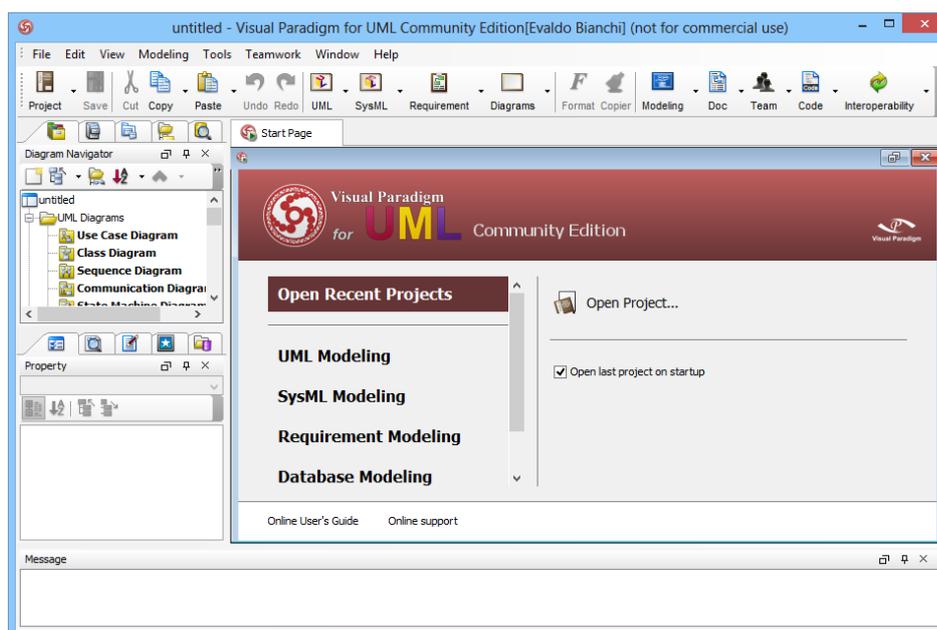


Figura 9 - Visual Paradigm UML Community Edition

3.2 MÉTODO

Para o desenvolvimento deste trabalho, alguns métodos foram utilizados a fim de concretizar o desenvolvimento do mesmo. A seguir serão descritas as etapas para o desenvolvimento juntamente com as principais atividades.

I) Definição inicial do escopo do sistema a ser desenvolvido

Após a análise dos softwares semelhantes existentes atualmente, foi definido o funcionamento do sistema, quais regras e métodos seriam seguidos.

II) Definição das tecnologias a serem utilizadas

Definido quais seriam os recursos e tecnologias necessárias para o desenvolvimento do software.

III) Definição do layout do sistema

Nesta etapa, por se tratar de um sistema mobile, onde o principal foco é a interface entre o aplicativo e o usuário, foram definidas como seriam as telas do sistema tanto em termos gráficos como funcionais.

IV) Análise e projeto

Definido o fluxo do sistema e a estrutura do banco de dados.

V) Codificação

Realização da codificação utilizando o ADT do Google e suas tecnologias nativas, juntamente com as demais tecnologias citadas.

VI) Testes

Após todas as implementações, realizados os testes em uma máquina virtual executando Android para identificar possíveis falhas no projeto, e também em aparelhos reais, simulando o funcionamento do transporte coletivo.

4 SISTEMA DESENVOLVIDO

Este capítulo apresenta o sistema desenvolvido como resultado deste trabalho. Referente a modelagem são apresentadas tabelas do banco de dados, referente a codificação, são apresentados parte dos códigos e as telas dos aplicativos.

4.1 APRESENTAÇÃO DO SISTEMA

O sistema desenvolvido como resultado deste estudo foi dividido em 3 aplicativos. O primeiro deles é o aplicativo localizador, e tem o objetivo de capturar a localização via GPS e enviar para o segundo aplicativo, o aplicativo servidor, desenvolvido para desktop.

Este servidor irá receber a posição enviada pelo localizador e armazenar em um banco de dados, isto para que o terceiro aplicativo, que será utilizado pelos usuários do transporte coletivo e que possui acessibilidade, solicite qual o transporte mais próximo ao seu local. A Figura 10 demonstra o funcionamento básico deste sistema.

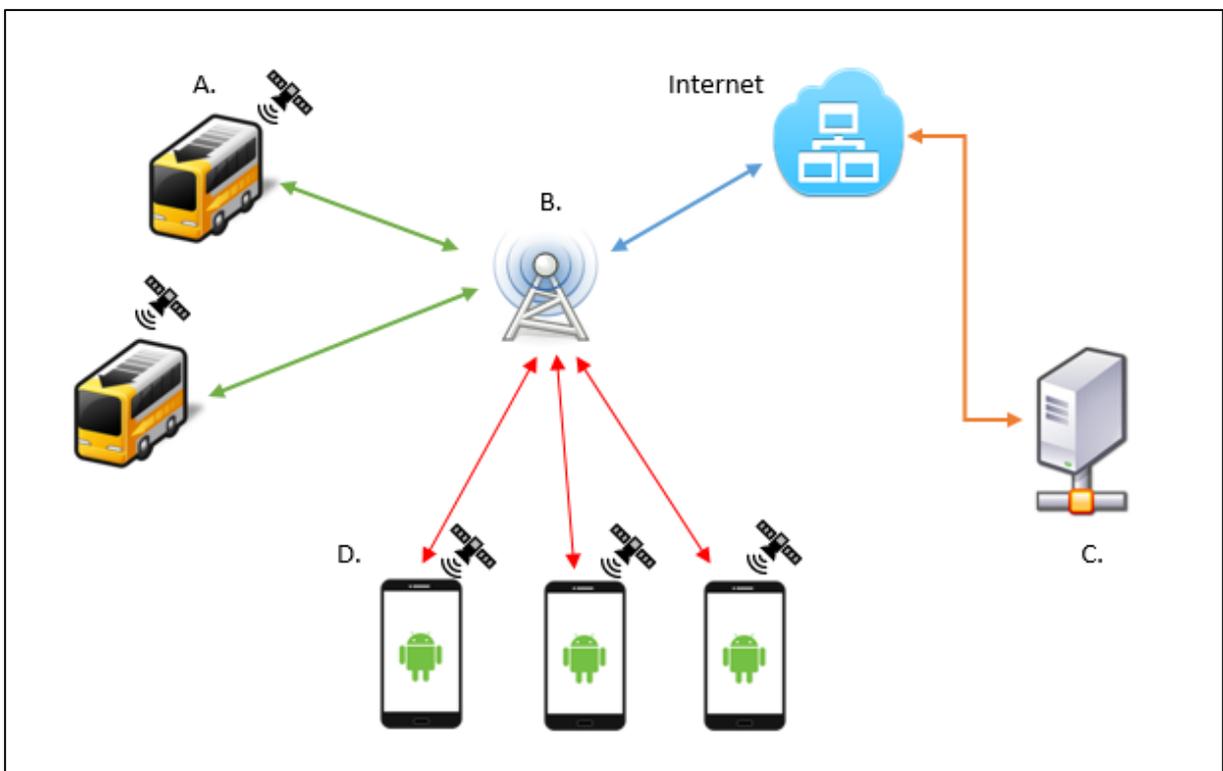


Figura 10 - Funcionamento do sistema

Na Figura, as letras representam:

- A. Representa o aplicativo localizador, ou seja, que captura a localização via GPS e envia para o servidor, através da infraestrutura de rede fornecida pelas operadoras de telefonia móvel.
- B. Demonstra a infraestrutura de dados das operadoras.
- C. O servidor que irá receber e armazenar as localizações dos veículos e responder as solicitações dos usuários.
- D. Os usuários do transporte, que irão solicitar qual o veículo mais próximo, com base na sua posição, também obtida por GPS.

4.1.1 Modelagem do Sistema

Para o desenvolvimento do sistema foram definidos alguns requisitos funcionais e não funcionais, estes listados nas tabelas 3 e 4, respectivamente.

Tabela 3 - Listagem dos requisitos funcionais

Identificação	Nome	Descrição
RF001	Cadastro de configurações	Cadastro das configurações de conexão com o servidor e preferências do usuário.
RF002	Porta de entrada de conexão	Permitir informar em qual porta o servidor irá receber os dados.
RF003	Obter localização	Obter a localização do dispositivo utilizando o GPS.
RF004	Armazenar local e datas	Armazenar em um banco de dados, todas as localizações identificadas pelo sistema de GPS, incluindo a data e hora em que foram obtidas.
RF005	Transmitir na forma de áudio as funções principais	O usuário deverá receber um retorno sonoro ao selecionar uma função.
RF006	Exibir localização em um mapa	Exibir a localização do usuário e a localização do transporte mais próximo em um mapa.

Tabela 4 - Listagem dos requisitos não funcionais

Identificação	Nome	Descrição
RNF001	Atualização da configuração	Os usuários do sistema tem permissão para alterar a configuração.
RNF002	Tipo de conexão	O método para a troca de informações será por meio de socket, com o mínimo de tráfego de dados possível.
RNF003	Acessos	O sistema irá aceitar acessos simultâneos.
RNF004	Banco de Dados	O banco de dados deverá ser criado de forma automática, na primeira execução dos aplicativos.

Com base nos requisitos funcionais e não funcionais, foram definidos os diagramas do sistema, entre eles, o diagrama de caso de uso do usuário com deficiência visual, o qual é apresentado na Figura 11.

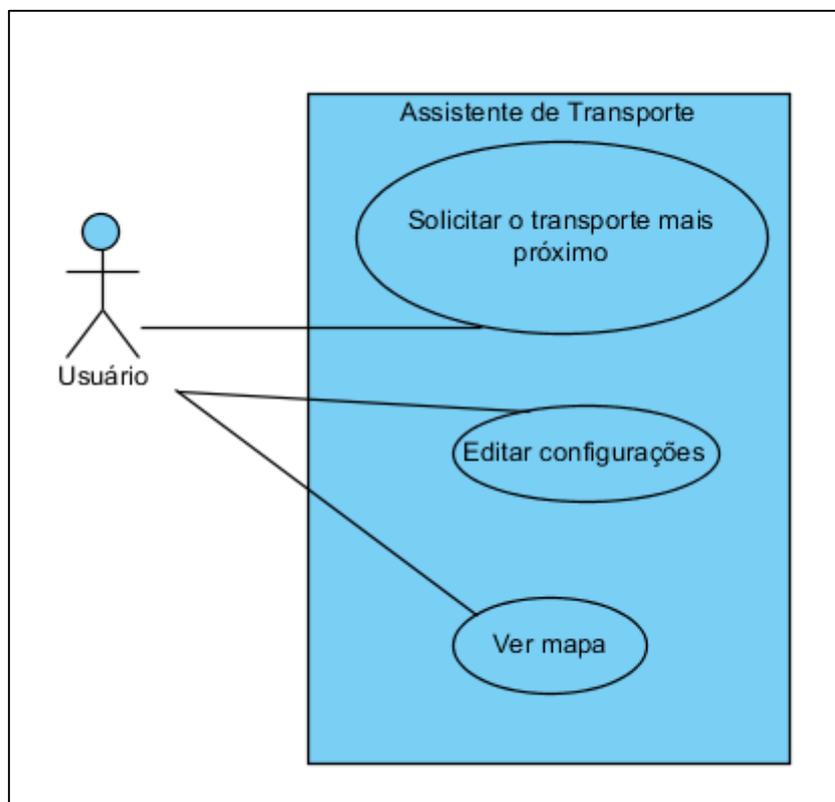


Figura 11 - Casos de uso

4.1.2 Persistência dos dados

Para o funcionamento do sistema, os três aplicativos possuem a estrutura de dados muito semelhante, apresentadas a seguir.

4.1.2.1 Aplicativo localizador

As localizações obtidas através do GPS, são armazenadas na tabela localização, descrita na Tabela 5, contendo um identificador, a posição (latitude e longitude), a data em que foi obtida e se já foi sincronizada com o servidor, esse campo receberá S para sim, e N para não, conforme Tabela 5.

Tabela 5 - Tabela de Localização

LOCALIZACAO	
Coluna	Tipo de dado
Id	Inteiro
Latitude	Texto
Longitude	Texto
Data	Texto
Sincronizado	Texto

As configurações do aplicativo, que irão armazenar o IP (*Internet Protocol*) e PORTA do servidor para o envio das localizações, e um apelido para identificar o veículo em que está instalado, serão salvas utilizando o recurso do Android, as SharedPreferences, uma técnica alternativa de persistência baseada em arquivo texto. A Tabela 6 demonstra como estes dados serão salvos.

Tabela 6 - Tabela de Configurações

CONFIGURACOES	
Campo	Tipo de dado
Apelido	Texto
Ip	Texto
Porta	Inteiro

4.1.2.2 Servidor

No aplicativo servidor, que irá comportar as localizações de todos os veículos da frota, possui a tabela Localização e armazena a posição, data e o apelido do veículo que enviou a informação como demonstra a Tabela 7.

Tabela 7 - Tabela de localizações no servidor

LOCALIZACAO	
Campo	Tipo de Dado
Id	Inteiro
Latitude	Texto
Longitude	Texto
Data	Texto
Apelido	Texto

4.1.2.3 Aplicativo Cliente

O aplicativo cliente não possui banco de dados, apenas armazena as configurações utilizando SharedPreferences. A estrutura do SharedPreferences é apresentada na Tabela 8. Os campos são o IP e porta do servidor, e se utiliza o recurso de sons para as funcionalidades, será representado por S (sim) ou N (não).

Tabela 8 - Tabela de configurações no aplicativo cliente

CONFIGURACOES	
Campo	Tipo de Dado
Ip	Texto
Porta	Inteiro
Utiliza_Sons	Texto

4.2 DESCRIÇÃO DO SISTEMA

O sistema de controle de transporte desenvolvido deve funcionar de forma simultânea e sincronizada, ou seja, os dados devem estar sempre atualizados entre os três aplicativos, com a maior frequência possível.

Como o sistema está dividido em 3 aplicativos principais, estes serão descritos na sequência.

4.2.1 Aplicativo localizador

O aplicativo localizador será um dispositivo Android presente nos veículos de transporte público. Ele é o responsável por obter a localização do veículo através do sistema de GPS do aparelho, armazenar em um banco de dados local e enviar constantemente esses dados para o aplicativo servidor.

Por se tratar apenas de um aplicativo que executará sem a interação humana (ficará fixado em um veículo), o aplicativo possui uma interface simples, e é possível executar em aparelhos com Android 2.1 ou superior, ou seja, aparelhos mais antigos e com baixo custo, bastando apenas estar disponível o recurso de localização.

A Figura 12 mostra como é a tela inicial do sistema. Ela contém um botão para iniciar a captura da localização, e um pequeno menu que direciona para a tela de Configurações, para a tela de Listar, que lista as localizações armazenadas e uma opção de envio, para enviar manualmente os dados.

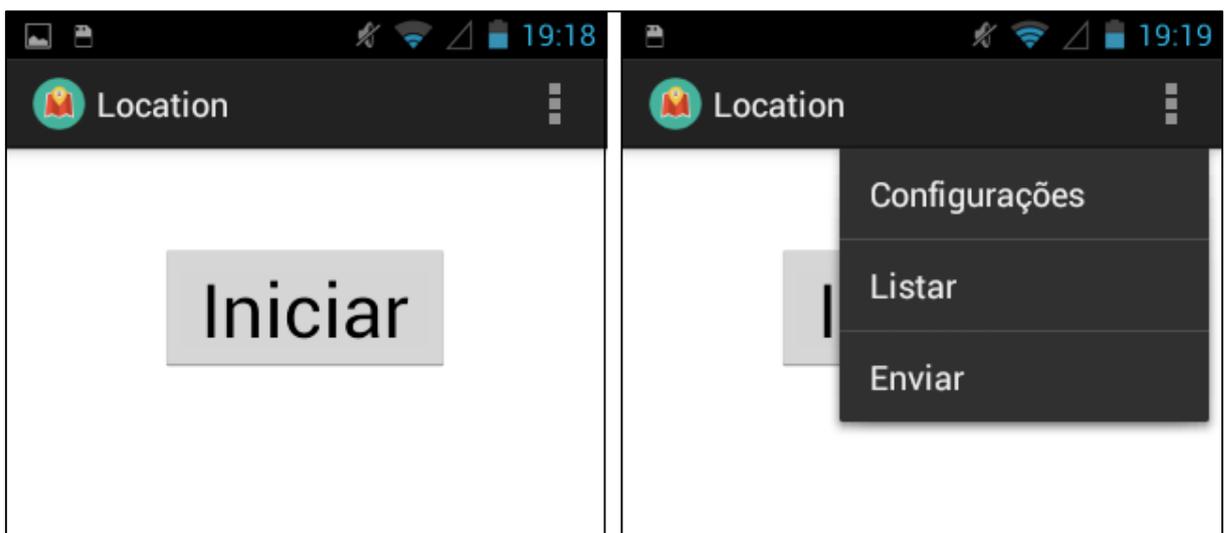


Figura 12 - Aplicativo Localizador

Ao selecionar a opção para iniciar a captura da localização, o usuário é direcionado para a tela mostrada na Figura 12, que também possui uma interface simples, e uma opção de parar o serviço.

Caso o GPS esteja desativado, uma mensagem é exibida para o usuário ativar, redirecionando o mesmo para a tela de configurações do Android como mostra a Figura 13.

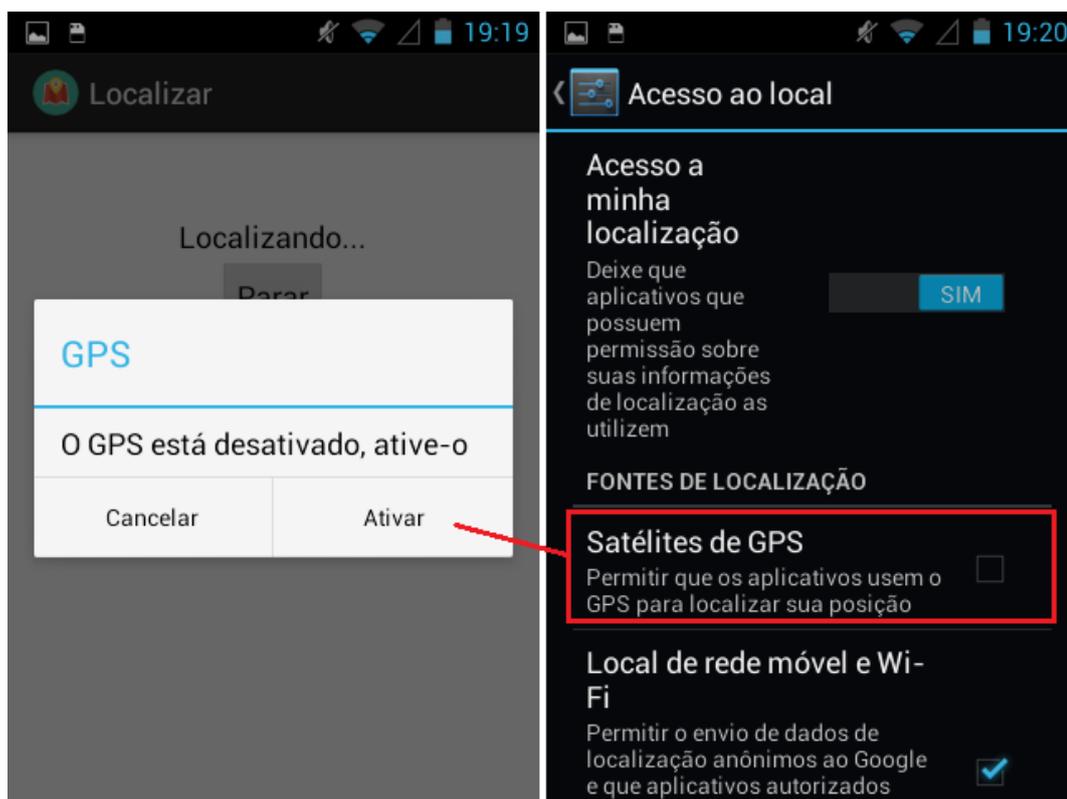


Figura 13 - Ativar GPS

Outras telas deste sistema são as de configuração e listar, mostradas na Figura 14. Estas possuem um layout simples. Nas configurações é possível informar o apelido do veículo, IP e porta do servidor, o sistema salva valores padrões para as configurações, caso o usuário não as altere, na tela da listagem, são exibidas as coordenadas de latitude e longitude capturadas.

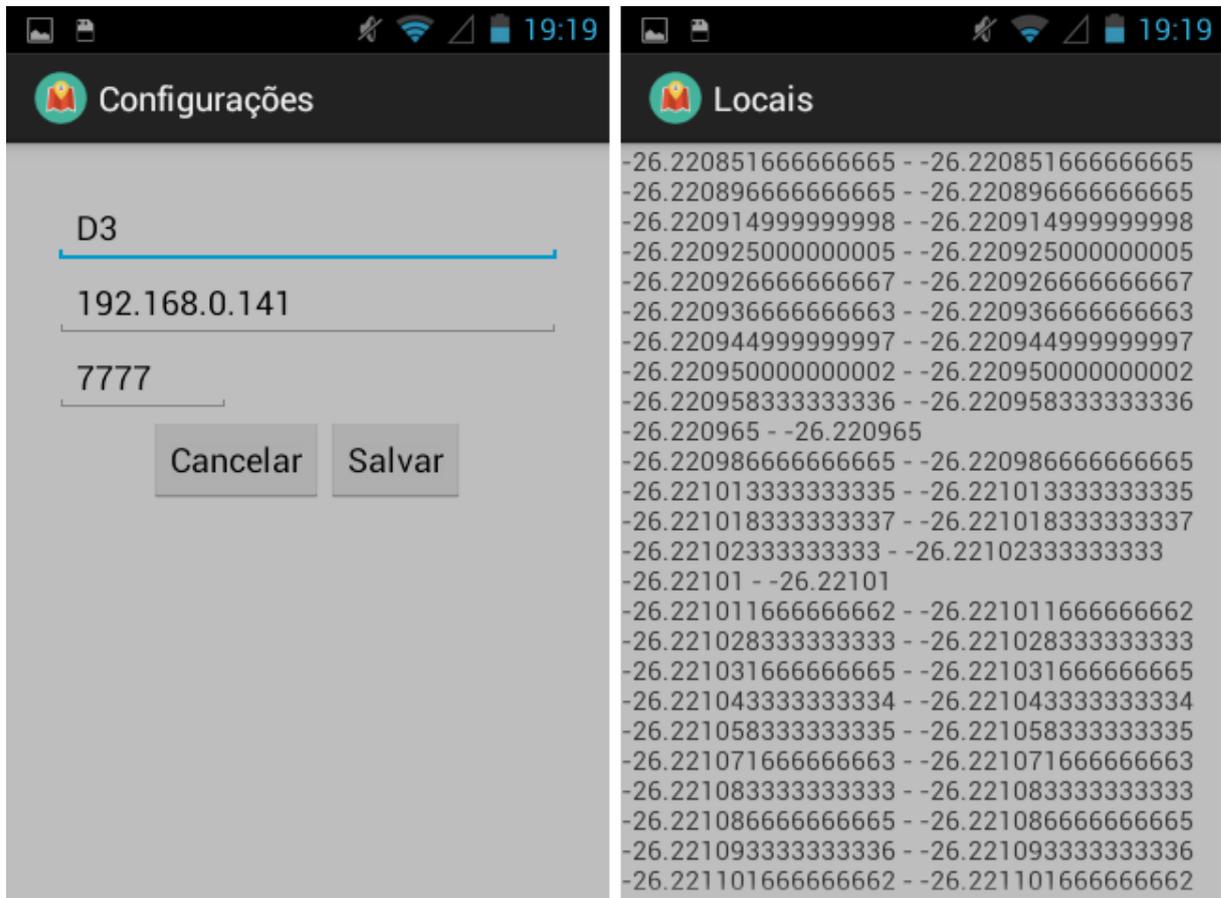


Figura 14 - Configurações e listagem de locais

O aplicativo localizador, armazena a posição do veículo a cada variação de 200 metros de posição, após armazenar no banco de dados, um evento é disparado e todas as localizações que não estiverem sincronizadas serão enviadas para o servidor. Desta forma, caso o local capturado no momento não seja sincronizado, na próxima localização, o sistema tentará sincronizar novamente.

4.2.2 Aplicativo Servidor

O aplicativo Servidor é quem irá receber os dados do aplicativo localizador, armazenando-os em um banco de dados e enviando-os para o aplicativo cliente quando solicitado. Ele foi desenvolvido para Desktop e possui uma única configuração, a porta que irá receber as conexões Socket, como mostra a Figura 15. O aplicativo também possui uma área onde são exibidos os logs dos dados recebidos.

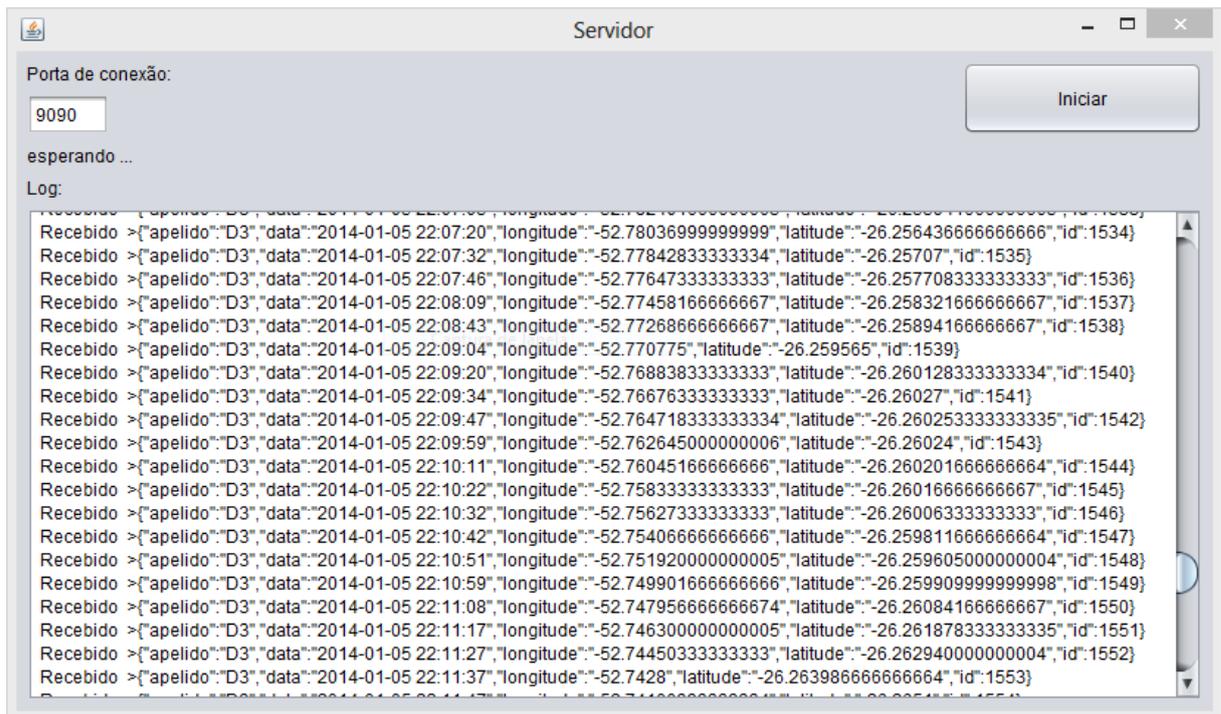


Figura 15 - Servidor

Como o objetivo principal do aplicativo é o aplicativo rastreador e o sistema cliente, este último utilizado por pessoas com deficiência visual, o aplicativo servidor se resume à receber dados do localizador e fornecer estes dados ao aplicativo cliente, recursos gerenciais e relatórios mais aprimorados, embora viáveis, não foram codificadas para o desenvolvimento deste relatório.

4.2.3 Aplicativo Cliente

O aplicativo cliente, chamado de assistente de transporte, será utilizado pelos usuários do sistema de transporte, ele deve fornecer, com base na sua posição e nos dados armazenados no servidor, qual o transporte mais próximo.

Com foco na acessibilidade, a interface do aplicativo contém um único botão, de um tamanho relativamente grande e em uma posição de fácil acesso na tela do aparelho, ao ser pressionado um retorno tátil é executado, ou seja, o aparelho executa uma pequena vibração.

Se estiver disponível e configurado para tal, a principal funcionalidade do assistente, é oferecer um retorno auditivo para a busca pelo transporte. Ao pressionar o botão “Encontrar o transporte mais próximo” esta mesma frase será pronunciada em áudio, assim que obtiver o retorno da solicitação, este também será retornado na forma auditiva. A Figura 16 mostra a

tela inicial do sistema e os possíveis avisos quanto a disponibilidade da conversão de texto em VOZ.

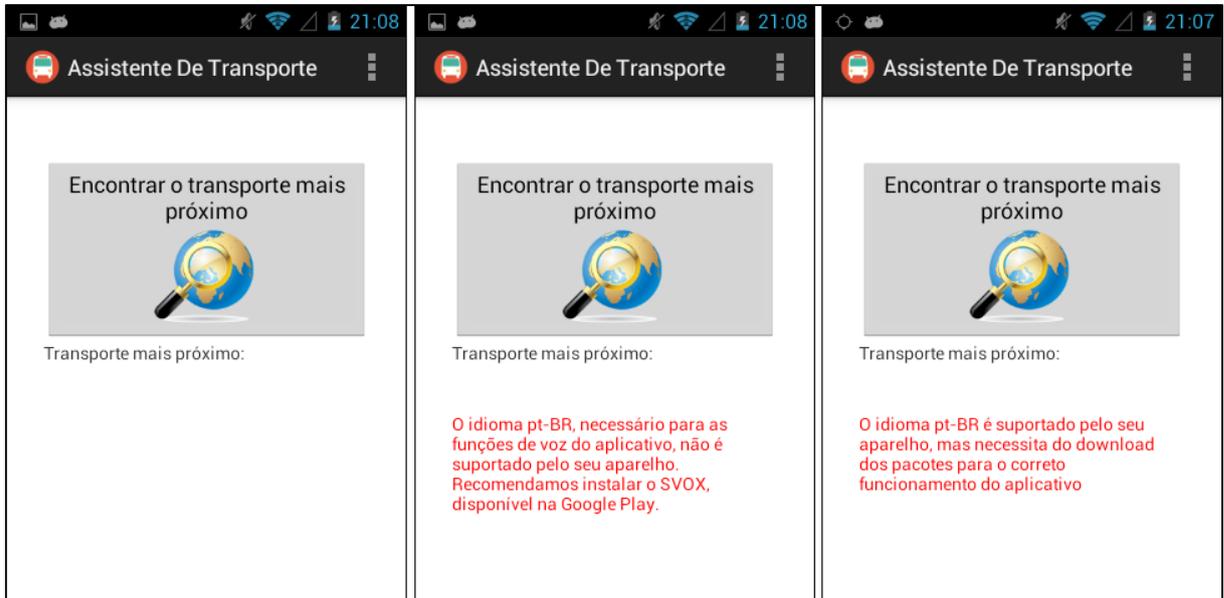


Figura 16 - Aplicativo Assistente de Transporte

Na tela de configurações, exibida na figura 17, é possível informar qual o IP e porta do servidor, e também se serão utilizados os recursos de áudio.



Figura 17 - Configurações do assistente de transporte

Ao selecionar a opção Mapa, no menu da tela principal, o usuário é direcionado para uma nova tela, onde um mapa é exibido com dois ícones, um representando a posição do

usuário, e outro a posição do transporte mais próximo, a Figura 18 mostra um exemplo desta funcionalidade. Como se trata de um retorno visual, esta tela será útil apenas para pessoas sem deficiências visuais, mas que venham a utilizar o aplicativo.

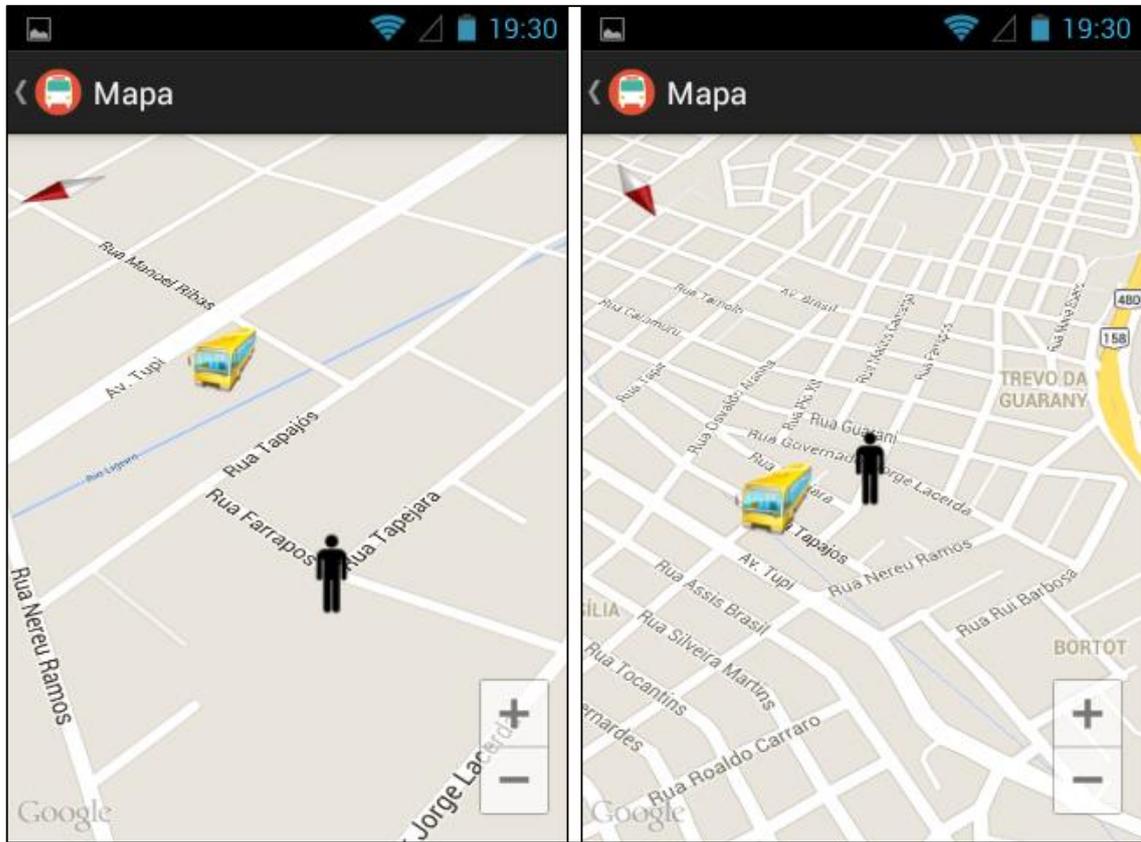


Figura 18 - Visualização do mapa

O resultado da busca pelo transporte mais próximo pode ser conferido na Figura 19.

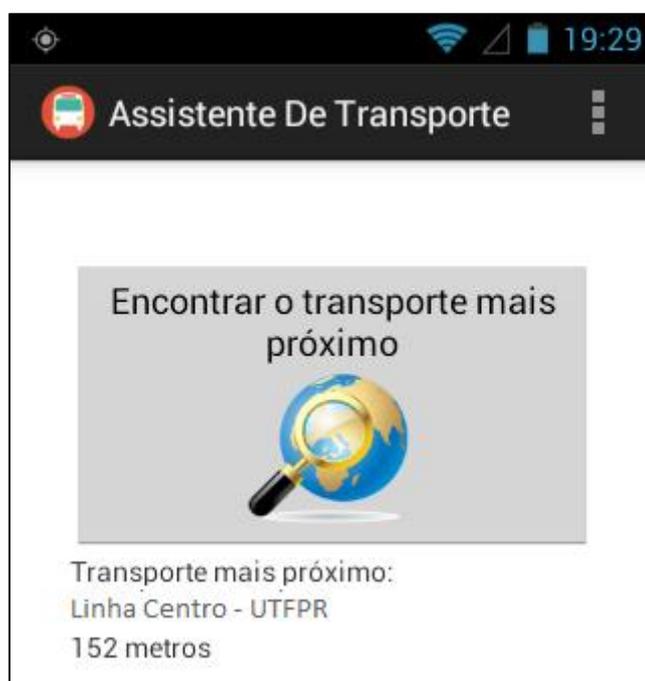


Figura 19 - Resultado da busca

4.3 IMPLEMENTAÇÃO DO SISTEMA

Algumas rotinas computacionais do sistema merecem destaque, como a sintetização de voz, utilização de mapas e GPS e comunicação via Socket.

No aplicativo localizador, a classe responsável por identificar a localização é apresentada na Listagem 2. A classe Localizar implementa a classe LocationListener, criada pela Google que possui quatro métodos de implementação obrigatória para a utilização do GPS, no código está sendo exibido o método onLocationChanged que será chamado toda vez que atender a condição definida no LocationManager, ou seja, a cada alteração de posição de 200 metros.

```

public class Localizar extends Activity implements LocationListener {
    private ServiceLocalizacao serviceLocalizacao = new
ServiceLocalizacao();
    @ViewById public TextView tvInfo;
    @AfterViews
    protected void init() {
        iniciarServico();
    }
    public void iniciarServico() {
        LocationManager locationManager = (LocationManager)
Localizar.this.getSystemService(Context.LOCATION_SERVICE);
        locationManager.requestLocationUpdates(LocationManager.GPS_PROVIDER
,0, 200, this);
    }
    @Override
    public void onLocationChanged(Location location) {
        tratarLocalizacao(location);
        tvInfo.setText("Último local "+ new Date());
    }
    @Background
    protected void tratarLocalizacao(Location location) {

        serviceLocalizacao.tratarLocalizacao(location,Localizar.this);
    }

    ...
}

```

Listagem de Código 2 - Classe Localizar

A anotação `@Background`, provida pelo `AndroidAnnotations`, executa o método de forma assíncrona, ou seja, em uma thread diferente da thread principal. No código, ele chama o serviço de localização, que será encarregado de armazenar a posição e enviar para o servidor. O código da classe `serviceLocalizacao` que irá tratar a localização é exibido na Listagem 3.

```

public void tratarLocalizacao(Location location, Context contexto) {
    Localizacao localizacao = new Localizacao();
    localizacao.setLatitude(Double.toString(location.getLatitude()));
    localizacao.setLongitude(Double.toString(location.getLongitude()));
;
    SimpleDateFormat simpleFormat = new SimpleDateFormat("yyyy-MM-dd
HH:mm:ss");
    String data = simpleFormat.format(new
Date(System.currentTimeMillis()));
    localizacao.setData(data);

    LocalizacaoDao dao = new LocalizacaoDao(contexto);
    if (dao.salva(localizacao))
        enviarLocalizacao(contexto);
    dao.close();
}

```

Listagem de Código 3 – Tratar a Localização

Se o registro for salvo com sucesso, será chamado o método para enviar a localização, demonstrado na Listagem 4.

```
public ServiceSocket(String ip, int porta) throws IOException {
    socket = new Socket(ip, porta);
    out = new DataOutputStream(socket.getOutputStream());
    in = new DataInputStream(socket.getInputStream());
}
public Integer enviar(String gson) throws IOException {
    try {
        out.writeUTF(gson);
        return in.readInt();
    } finally {
        close();
    }
}
public void close() throws IOException {
    out.close();
    in.close();
    socket.close();
}
```

Listagem de Código 4 - Enviar para o servidor

Os dados serão enviados no formato JSON, a biblioteca GSON permite converter facilmente objetos em JSON, como mostra a Listagem 5.

```
public String geraGson(Localizacao localizacao){
    Gson gson = new Gson();
    return gson.toJson(localizacao);
}
```

Listagem de Código 5 - Gerar GSON

No aplicativo servidor, código da Listagem 6, realiza o recebimento, tanto dos dados recebidos do localizador quando do cliente.

```

@Override
public void run() {
    try {
        DataOutputStream out = new DataOutputStream(socket.getOutputStream());
        DataInputStream in = new DataInputStream(socket.getInputStream());

        String recebido = in.readUTF();
        Gson gson = new Gson();
        Localizacao localizacao = gson.fromJson(recebido,
        Localizacao.class);
        if (localizacao.getId() == -1) {
            try {
                MaisProximo maisProximo = new EncontraTransporteMaisProximo().
                encontrarLocalizacaoMaisProxima(localizacao);
                out.writeUTF(new Gson().toJson(maisProximo));
            } catch (SQLException ex) {
                Logger.getLogger(ServidorSocketThread.class.getName()).log(Level.S
                EVERE, null, ex);
            } catch (ClassNotFoundException ex) {
                Logger.getLogger(ServidorSocketThread.class.getName()).log(Level.S
                EVERE, null, ex);
            }
            Principal.taLog.append("\nSolicitação do aplicativo cliente \n"
            + "sua posição = " + localizacao.getLatitude() + " - " +
            localizacao.getLongitude());
            } else {
                try {
                    new LocalizacaoDao().insert(localizacao);
                    Principal.taLog.append("Recebido >" + recebido + "\n");

                    out.writeInt(localizacao.getId());
                } catch (SQLException ex) {
                    Principal.taLog.append("Erro ao inserir o registro >" + ex.getMessage()
                    + "\n");
                } catch (ClassNotFoundException ex) {
                    Principal.taLog.append("Erro ao inserir o registro >" +
                    ex.getMessage() + "\n");
                }
            }

            out.close();
            in.close();
            socket.close();
        } catch (IOException e) {
            Principal.taLog.append("Erro >" + e.getMessage() + "\n");
        }
    }
}

```

Listagem de Código 6 - Recebimento dos dados no servidor

Caso o servidor receba uma solicitação do localizador, este irá armazenar os dados e retornar o id para que o localizador atualize como sincronizado. Já para o caso de uma solicitação do aplicativo cliente, será chamada a rotina que irá buscar do banco de dados os

dados mais recentes que possui, com base na data, e realizar o cálculo exibido na Listagem 7, para retornar qual o transporte mais rápido. Lembrando que a longitude e a latitude se comportam como pontos de um plano cartesiano, assim, pode-se aplicar fórmulas de geometria analítica para calcular a distância entre pontos (PORTA, 2013).

```
//Raio da Terra em Km
private static final Double fatorKM = 6371D;

public Double distanciaEntrePontos(Double latitudeA, Double longitudeA,
Double latitudeB, Double longitudeB ) {

return
fatorKM * Math.acos(Math.cos(Math.PI * (90 - latitudeB) / 180)
* Math.cos((90 - latitudeA) * Math.PI/180) +
Math.sin((90 - latitudeB) * Math.PI / 180) *
Math.sin((90 - latitudeA) * Math.PI / 180)
* Math.cos(( longitudeA - longitudeB) * Math.PI / 180));
}
```

Listagem de Código 7 - Cálculo da distância entre pontos geográficos

No aplicativo cliente, a Listagem 8, demonstra o funcionamento da conversão de texto em voz, verificando se está disponível no aparelho e configurado para utilizar o recurso.

O método `speech.isLanguageAvailable(new Locale("pt", "BR"))`, retorna um código referente a disponibilidade do idioma português, o retorno é tratado da seguinte forma:

- 0 - suportado mas não da forma como foi solicitado
- 1 - suportado exatamente da forma solicitada
- 2 - suportado exatamente da forma solicitada com o argumento variantes
- -1 - suportado mas necessita do download dos pacotes de voz
- -2 - não suportado

```

public void onInit(int status) {
    if (status == TextToSpeech.SUCCESS) {
        int idiomaDisp = speech.isLanguageAvailable(new Locale("pt", "BR"));
        if (idiomaDisp == 0 || idiomaDisp == 1 || idiomaDisp == 2)
            speech.setLanguage(new Locale("pt", "BR"));
        else if (idiomaDisp == -1)
            tvInfoIdioma.setText("O idioma pt-BR é suportado pelo seu
aparelho, mas necessita do download dos pacotes para o correto
funcionamento do aplicativo");
        else tvInfoIdioma.setText("O idioma pt-BR, necessário para as
funções de voz do aplicativo, não é suportado pelo seu aparelho.
Recomendamos instalar o SVOX, disponível na Google Play.");
    } else {
        Log.e("TTS", "Initalization Failed!");
    }
}
}

```

Listagem de Código 8 - Tratamento da conversão de texto em voz

Tanto no aplicativo localizador, quando no cliente, é verificado se o GPS está ativado e informado ao usuário no caso negativo, direcionando o mesmo para a tela de configurações do Android onde poderá ativa-lo. O código é exibido na Listagem 9.

```

public void onProviderDisabled(String provider) {
    new AlertDialog.Builder(MainActivity.this)
        .setMessage("O GPS está desativado, ative-o").setTitle("GPS")
        .setPositiveButton("Ativar", new DialogInterface.OnClickListener()
    {
        @Override
        public void onClick(DialogInterface dialog, int which) {
            Intent intent = new
Intent(Settings.ACTION_LOCATION_SOURCE_SETTINGS);
            startActivityForResult(intent, 1);
        }
    })
        .setNegativeButton("Cancelar", new DialogInterface.OnClickListener() {
            @Override
            public void onClick(DialogInterface dialog, int which) {
                finish();
            }
        }).show();
}

```

Listagem de Código 9 - Verificação do GPS

O código da XML tela responsável por exibir o mapa, mostrado na Listagem 10, utiliza-se de Fragment, incluído na versão 3.0 do Android, os fragments são como “buracos” na tela, que permitem inserir vários “pedaços de tela” na mesma activity (LECHETA, 2013).

```

<?xml version="1.0" encoding="utf-8"?>
<fragment xmlns:android="http://schemas.android.com/apk/res/android"
    android:id="@+id/map"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:name="com.google.android.gms.maps.MapFragment"/>

```

Listagem de Código 10 - Tela do mapa

O código da Listagem 11 adiciona o mapa na tela.

```
GoogleMap mapa;  
mapa = ((MapFragment)  
getFragmentManager().findFragmentById(R.id.map)).getMap();  
mapa.setMapType(GoogleMap.MAP_TYPE_NORMAL);
```

Listagem de Código 11 - Exibição do mapa na tela

A Listagem 12, exibe como são adicionados os marcadores no mapa.

```
private void adicionarMarcador(LatLng latLng, String titulo, String  
descricao, int iconeId) {  
    MarkerOptions markerOptions = new MarkerOptions();  
    markerOptions.position(latLng).title(titulo).snippet(descricao);  
    markerOptions.icon(BitmapDescriptorFactory.fromResource(iconeId));  
    mapa.addMarker(markerOptions);  
}
```

Listagem de Código 12 - Adicionar marcador ao mapa

4.3.1 Testes

Para a finalização do desenvolvimento, os testes foram realizados com 2 aparelhos reais, sendo um com Android 2.1, onde foi instalado o aplicativo localizador, e outro com Android 4.1.2 onde foi instalado o assistente de transporte e o aplicativo servidor para desktop foi instalado em um notebook.

Durante alguns dias, o aplicativo localizador ficou ativo capturando a localização do aparelho, dessa forma, foi possível obter uma base de dados com informações suficientes para a realização de testes com o aplicativo cliente.

Os dados do localizador foram sincronizados em uma rede local, onde o servidor estava executando, a partir disso, foram simuladas as localizações do usuário e o servidor retornou com êxito qual seria o transporte mais próximo e a distância deste com relação a posição do usuário.

Infelizmente, por questões de tempo e por não conhecer pessoas com deficiência visual, os testes reais não puderam ser realizadas, apenas simuladas.

5 CONCLUSÃO

Este trabalho teve como objetivo desenvolver um sistema móvel para o controle de transporte coletivo que viesse a auxiliar a utilização deste por pessoas com deficiência visual. Após todo o desenvolvimento do projeto, por meio do levantamento dos requisitos, das funcionalidades e das tecnologias utilizadas é possível concluir que o objetivo geral pode ser alcançado.

Foi possível entender, de uma forma geral, o funcionamento do sistema de transportes coletivo no Brasil, além de demonstrar a utilização dos recursos disponíveis na plataforma Android, como a utilização do GPS, conectividade entre os dispositivos e o servidor de dados e o funcionamento da sintetização de voz e mapas da plataforma.

No término do desenvolvimento, obteve-se um sistema funcional, com o aplicativo localizador, o servidor e o aplicativo cliente enviando e recebendo as informações e executando suas funcionalidades com eficiência, de forma que pode vir a auxiliar os usuários do transporte coletivo com ou sem dificuldades visuais.

Das dificuldades encontradas durante o desenvolvimento deste trabalho pode-se destacar, durante a pesquisa, a dificuldade em encontrar referências sobre o funcionamento do transporte coletivo e sobre a acessibilidade deste serviço para deficientes visuais. Durante a codificação, a sintetização de voz e a utilização do mapa foram os pontos mais complexos, e finalmente durante os testes, foram necessários dados de localização reais, para isso, foi preciso alguns dias de testes com um aparelho executando o localizador.

O sistema desenvolvido durante este trabalho, abre uma gama de possibilidades de ampliação e aperfeiçoamento para trabalhos futuros, como a implementação de um aviso quando o usuário estiver se aproximando do seu destino e também que o assistente de transporte guie o usuário até o ponto de parada mais próximo, além de aprimorar os recursos de acessibilidade implementado o reconhecimento de voz, para que o deficiente visual não necessite interagir com a tela.

Por fim, é possível concluir que o desenvolvimento deste sistema deve solucionar o problema proposto, facilitando a utilização do transporte coletivo por deficientes visuais fornecendo um pouco mais de autonomia para sua locomoção.

REFERÊNCIAS

ANDROIDANNOTATIONS. **AndroidAnnotations Wiki**. Disponível em: <<https://github.com/excilys/androidannotations/wiki>>. Acesso em 20 dez. 2013.

BUSALERT. **BusAlert**. Disponível em: <<http://www.busalert.com.br/>>. Acesso em 25 jan. 2014.

CARVALHO, Jary e Castro. **IR E VIR: Acessibilidade: compromisso de cada um**. Campo Grande MS: Editora GIBIM, 2013.

D2, Barcelos. **Sistema que auxilia deficientes visuais a usar transporte público pode ser implantado**. Disponível em: <http://www.em.com.br/app/noticia/gerais/2012/12/10/interna_gerais,335897/sistema-que-auxilia-deficientes-visuais-a-usar-transporte-publico-pode-ser-implantado.shtml>. Acesso em 29 dez. 2013.

DEVELOPER. **Android Developer**. Disponível em: <<http://developer.android.com/index.html>>. Acesso em: 1 dez. 2013.

ECLIPSE. **Eclipse**. Disponível em: <<https://www.eclipse.org/>>. Acesso em 5 dez. 2013.

G1. **Google Play tem 700 mil aplicativos, mesmo número da loja da Apple**. Disponível em: <<http://g1.globo.com/tecnologia/noticia/2012/10/google-play-tem-700-mil-aplicativos-mesmo-numero-da-apple-diz-agencia.html>>. Acesso em: 20 nov. 2013.

GARTNER. **Gartner**. Disponível em: <<http://www.gartner.com/technology/home.jsp>>. Acesso em 15 jan. 2014.

GOOGLE-GSON. **Google-GSON**. Disponível em: <<https://code.google.com/p/google-gson/>>. Acesso em 20 dez. 2013.

GUIMARÃES, Saulo Pereira. **Brasil é o quarto país do mundo em número de smartphones.** Disponível em: <<http://exame.abril.com.br/tecnologia/noticias/brasil-e-o-quarto-pais-do-mundo-em-numero-de-smartphones>>. Acesso em 29 nov. 2013.

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

NETBEANS. **NetBeans IDE.** Disponível em: <<https://netbeans.org/>>. Acesso em 1 jan. 2014.

OMS. **Organização Mundial da Saúde.** Disponível em: <<http://www.who.int/en/>>. Acesso em 10 dez. 2013.

ORACLE. **Java SE.** Disponível em: <<http://www.oracle.com/technetwork/pt/java/javase/downloads/index.html>>. Acesso em: 1 dez. 2013.

PLANALTO.GOV.BR. **LEI Nº 10.098, DE 19 DE DEZEMBRO DE 2000.** Disponível em: <http://www.planalto.gov.br/ccivil_03/leis/110098.htm>. Acesso em 15 dez. 2013.

PORTA, Giovani La. **Calculando a Distância entre 2 Pontos Latitude e Longitude em Java.** Disponível em: <<http://www.frameworksystem.com/blog/calculando-a-distancia-entre-2-pontos-latitude-e-longitude-em-java/>>. Acesso em 23 nov. 2013.

RASMUSSEN, Bruna. **Android: o sistema operacional móvel que conquistou o mundo.** Disponível em: <<http://www.tecmundo.com.br/infografico/9010-android-o-sistema-operacional-movel-que-conquistou-o-mundo.htm#ixzz2Y2OCqLHN>>. Acesso em: 17 nov. 2013.

REI, Gilson; MALTA, Jacqueline. **Cidadania e EMDEC avaliam sistema para cegos no transporte público.** Disponível em: <<http://www.emdec.com.br/eficiente/sites/portalemdec/pt-br/site.php?secao=noticiasgerais&pub=7216>>. Acesso em 25 jan. 2014.

SQLITE. **SQLite**. Disponível em: <<http://www.sqlite.org/index.html>>. Acesso em: 5 dez. 2013.

TECMUNDO. **TecMundo**. Disponível em: <<http://www.tecmundo.com.br/>>. Acesso em 1 dez. 2013.

VISUAL PARADIGM. **Visual Paradigm for UML 11.0 Community Edition**. Disponível em: <<http://www.visual-paradigm.com/product/vpuml/editions/community.jsp>>. Acesso em: 20 jan. 2014.