

PONTIFÍCIA UNIVERSIDADE CATÓLICA DE MINAS GERAIS

Curso de Sistemas de Informação

Paulo Vítor Alves da Silva

**DESENVOLVIMENTO DE APLICAÇÕES MÓVEIS UTILIZANDO O
POSICIONAMENTO GEOGRÁFICO EM PLATAFORMA GOOGLE ANDROID**

Belo Horizonte

2012

Paulo Vítor Alves da Silva

**SISTEMA DE ORIENTAÇÃO DE TRANSPORTE COLETIVO ATRAVÉS DE
POSICIONAMENTO GEOGRÁFICO EM PLATAFORMA GOOGLE ANDROID**

Monografia apresentada ao curso de
Sistemas de Informação da Pontifícia
Universidade Católica de Minas Gerais,
como requisito parcial para obtenção do
título de Bacharel em Sistemas de
Informação.

Orientador: Paulo César do Amaral Pereira

Belo Horizonte

2012

Paulo Vítor Alves da Silva

SISTEMA DE ORIENTAÇÃO DE TRANSPORTE COLETIVO ATRAVÉS DE POSICIONAMENTO GEOGRÁFICO EM PLATAFORMA GOOGLE ANDROID

Monografia apresentada ao curso de Sistemas de Informação da Pontifícia Universidade Católica de Minas Gerais, como requisito parcial para obtenção do título de Bacharel em Sistemas de Informação.

Paulo César do Amaral Pereira (Orientador) – PUC Minas

Belo Horizonte, 13 de junho de 2012

AGRADECIMENTOS

Agradeço primeiramente a Deus por tudo que possuo hoje, conhecimento, saúde e vontade de prosseguir. A meus pais e familiares, pelo apoio e entendimento em todos os momentos. Aos amigos e colegas, sendo de faculdade, colégio, profissão, pois com eles pude aprender muito do que sei.

RESUMO

O objetivo deste trabalho é, além de desenvolver um programa, explorar alguns recursos disponíveis hoje em dia, na computação móvel. O programa foi focado na necessidade de informações de usuários do transporte coletivo, Com a utilização do programa, o usuário terá uma noção mais exata do horário aproximado em que o ônibus passará pelo local informado.

Palavras-chave: Desenvolvimento Android; GPS; Transporte Coletivo; Ônibus

LISTA DE FIGURAS

Figura 1 - <i>TOP Smartphone Platforms</i>	19
Figura 2 - Camadas de Abstração do iOS.....	20
Figura 3 - Satélite Navstar. Utilizado no sistema GPS.	23
Figura 4 - Lista das empresas que pertencem a Open Handset Alliance.....	27
Figura 5 - Ranking Linguagens de Programação.....	28
Figura 6 - Diagrama Arquitetura Android.....	30
Figura 7 - Descobrindo a debug.keystore	37
Figura 8 - Inscrição para uma Chave da API Google Maps	38
Figura 9 - Inclusão da classe MapView.....	44
Figura 10 - Localização do dispositivo móvel	47
Figura 11 - Ponto de ônibus selecionado pelo usuário.....	48
Figura 12 - Próximo horário do ônibus selecionado	49

LISTA DE SIGLAS

3G – 3ª Geração de Padrões de Tecnologias Móveis

ADT – *Android Development Tools*

API – *Application Programming Interface*

BSD - *Berkeley Software Distribution* ou Distribuição do Sistema de Berkeley

CDMA – *Code Division Multiple Access* ou Acesso Múltiplo por Divisão de Código

DOD – *Department of Defense*

eBook – *Eletronic Book* ou Livro Eletrônico

FDMA – *Frequency Division Multiple Access* ou Múltiplo Acesso por Divisão de Frequência

GPS – *Global Positioning System*

JDK – *Java SE Development Kit* ou Kit de Desenvolvimento Java

MIT - *Massachusetts Institute of Technology* ou Instituto de Tecnologia de Massachusetts

MVD – *Dalvik Virtual Machine* ou Máquina Virtual Dalvik

OHA – *Open Handset Alliance*

PC – *Personal Computer* ou Computador Pessoal

PDA's – *Personal Digital Assistants* ou Assistente Pessoal Digital

PPS – *Precise Positioning Service* ou Serviço de Posicionamento Preciso

SDK – *Software Development Kit* ou Kit de Desenvolvimento de Software

SMS – *Short Message Service*

SO – Sistema Operacional

SPS – *Standard Positioning Service* ou Serviço de Posicionamento Padrão

TDMA – *Time Division Multiple Access* ou Acesso Múltiplo por Divisão de Tempo

VM – *Virtual Machine* ou Máquina Virtual

WEB – *Word Wide Web* ou Rede de alcance mundial

SUMÁRIO

1	INTRODUÇÃO	12
1.1	O PROBLEMA	13
1.2	OBJETIVO	14
1.3	ORGANIZAÇÃO DO TRABALHO	14
2	REVISÃO BIBLIOGRÁFICA.....	16
2.1	DISPOSITIVOS MÓVEIS	16
2.1.1	<i>A evolução dos dispositivos móveis</i>	<i>16</i>
2.1.2	<i>Os principais tipos de dispositivos móveis</i>	<i>18</i>
2.1.3	<i>Sistemas operacionais móveis</i>	<i>19</i>
2.2	GPS	22
2.3	APLICAÇÕES MÓVEIS.....	23
3	PLATAFORMA DE DESENVOLVIMENTO	25
3.1	ANDROID	25
3.1.1	<i>Arquitetura</i>	<i>29</i>
3.1.2	<i>Bibliotecas (Libraries)</i>	<i>31</i>
3.1.3	<i>Aplicação Android (Android Runtime)</i>	<i>33</i>
3.1.4	<i>Núcleo do Linux (Linux Kernel).....</i>	<i>34</i>
3.1.5	<i>Desenvolvimento para Android</i>	<i>34</i>
3.2	GOOGLE MAPS	35
3.3	CÓDIGO DE CERTIFICAÇÃO DIGITAL	36
3.4	CÓDIGO ACESSO AO GOOGLE MAPS	38
3.5	IDE ECLIPSE	39
4	ESTUDO DE CASO	41
4.1	RESUMO	41
4.2	DESCRIÇÃO DO PROBLEMA	41
4.3	AMBIENTE DE DESENVOLVIMENTO	41
4.4	O PROJETO ANDROID	43
4.4.1	<i>A classe GeoPoint</i>	<i>44</i>
4.4.2	<i>Overlays e Mapa.....</i>	<i>44</i>
4.4.3	<i>LocationManager</i>	<i>45</i>
4.4.4	<i>Classes auxiliares</i>	<i>46</i>

5	APLICATIVO	47
6	CONCLUSÃO	50
	BIBLIOGRAFIA	52
	APÊNDICE A – CLASSE COORDENADA.JAVA	57
	APÊNDICE B – CLASSE DADOSONIBUS.JAVA	59
	APÊNDICE C – CLASSE HORARIOONIBUS.JAVA	60
	APÊNDICE D – DIAGRAMA DE CLASSES.....	61

1 INTRODUÇÃO

A era da informação está sendo marcada como um período onde a resposta ao usuário precisa de agilidade e mobilidade para que se ganhe confiança e destaque em um ambiente competitivo. A computação ubíqua se destaca neste ambiente permitindo que os usuários tenham acesso a informações a qualquer hora e em qualquer lugar.

A computação móvel está cada dia mais presente e indispensável na vida das pessoas, e com ela vêm diversos recursos interessantes que podem ser utilizados para facilitar o dia a dia, tais como: o GPS (*Global Position System* ou Sistema de Posicionamento Global), acesso à Internet e diversas aplicações existentes.

Um sistema móvel é basicamente um sistema embarcado utilizado em aparelhos móveis, em outras palavras, é um sistema voltado para celulares, smartphones e derivados (Embarcado Básico, 2012). Os SO's para celulares se tornaram um tipo de sistema embarcado à parte, pois cada vez mais apresentam novos recursos, funções e interatividade com o usuário.

Conforme o mercado de telefonia celular se desenvolve, cresce também a possibilidade de se difundir aplicações que necessitam de conexão direta, que fazem uso de algum serviço disponível na WEB, uma vez que os dispositivos móveis estão cada vez mais capazes de executar tarefas que, outrora, somente computadores com processadores mais potentes executavam. Desta forma, as empresas estão voltando seu olhar para o mercado móvel, disponibilizando serviços on-line que podem ser acessados por celulares, smartphones e através de uma conexão com a internet.

O sistema operacional Google Android surgiu com o intuito de padronizar os SO's para dispositivos móveis, manter uma plataforma livre, desta forma, diversas aplicações podem ser desenvolvidas e utilizadas em qualquer celular que tiver o sistema operacional Google Android, obedecendo apenas às restrições de cada versão.

O GPS (*Global Positioning System* ou Geo-posicionamento por Satélite, ou Sistema de Posicionamento Global) é um sistema de navegação por satélite que fornece a posição de um aparelho receptor móvel, assim como informação horária, sob quaisquer condições atmosféricas, a qualquer momento e em qualquer lugar na terra, desde que o receptor encontre no campo de visão de quatro satélites GPS (Sistema de Posicionamento Global, 2004). Atualmente existem vários tipos de receptores, Geodésicos, Topográficos e de Navegação. A maior diferença entre eles é o preço e a precisão, podendo ser na casa de centímetros. A categoria de maior uso é a de Navegação, mais barata e de menor precisão, que hoje em dia é contemplada em diversos dispositivos móveis e é a categoria utilizada para desenvolver este trabalho.

1.1 O PROBLEMA

Em razão do grande crescimento da população brasileira, as cidades estão mais cheias, as pessoas estão adquirindo veículos e em consequência disto as vias estão ficando congestionadas, tornando o trânsito caótico. Uma solução para este problema seria retirar de circulação a maior quantidade possível de veículos particulares das vias públicas, fornecendo um transporte público com qualidade e eficiência, proporcionando ao usuário conforto e ganho de tempo no trajeto que será percorrido.

Atualmente, um dos maiores problemas enfrentados pelo usuário do transporte público rodoviário é saber quando o ônibus passará pelo ponto de parada desejado, causando um desconforto diário na espera do mesmo.

Se o usuário soubesse o momento aproximado que o ônibus passa pelo ponto de parada mais próximo, ele poderia se programar e se deslocar para ele no horário próximo à chegada do ônibus.

O mercado de celulares está crescendo cada vez mais. Estudos mostram que hoje em dia mais de três bilhões de pessoas possuem um aparelho celular, e isso corresponde a mais ou menos metade da população mundial. (LECHETA 2009, 2010). Aproveitando este fato, grande parte da população poderia utilizar o programa para se orientar no caso citado anteriormente.

1.2 OBJETIVO

Desenvolver uma ferramenta para dispositivos móveis capaz de oferecer aos usuários de transportes coletivos algumas informações referentes ao itinerário e previsão de horários das linhas que eles utilizam diariamente através de um aparelho de telefonia móvel.

Utilizar tecnologias móveis e ferramentas de desenvolvimento, tais como GPS, Android, Java e Eclipse, para desenvolver a aplicação proposta.

Auxiliar nas disciplinas de Sistemas Distribuídos, Aplicações Móveis, Sistemas Operacionais, Tecnologias WEB, entre outras, como referencial teórico e prático.

1.3 ORGANIZAÇÃO DO TRABALHO

O capítulo 2 trata sobre sistemas móveis, GPS e aplicações móveis, apresentando informações básicas sobre os mesmos, bem como a evolução, o surgimento e dados importantes para desenvolver a aplicação móvel proposta neste trabalho.

O capítulo 3 contém informações sobre a plataforma de desenvolvimento utilizada, o Android, sua arquitetura, bibliotecas, aplicações, o núcleo do Linux

próprio para ele e o SDK, informações sobre a IDE de desenvolvimento Eclipse, sua história, versões mais recentes e características, obtenção do código de certificação digital e acesso ao Google Maps.

O capítulo 4 contém o estudo de caso, contendo um resumo do trabalho, a descrição do problema e as etapas de desenvolvimento de uma aplicação Android em IDE Eclipse com *plugin* ADT, utilizando o SDK Android e API de Mapas do Google.

2 REVISÃO BIBLIOGRÁFICA

Este capítulo apresenta as características básicas sobre aplicações e sistemas e dispositivos móveis, bem como os sistemas operacionais móveis mais utilizados nos dias de hoje e GPS, apresentando a aplicabilidade dentro do tema proposto.

2.1 DISPOSITIVOS MÓVEIS

A chegada dos dispositivos móveis causou uma grande mudança no dia a dia de diferentes tipos de pessoas e organizações. A crescente inovação da tecnologia móvel torna produtos e equipamentos rapidamente obsoletos, acarretando uma necessidade ágil de adaptação por parte das operadoras e fabricantes de telefones móveis.

A era da mobilidade cada vez mais se instala no cotidiano das pessoas e o telefone móvel passa a ser o instrumento essencial de sobrevivência no mundo. Esse cenário é cada dia mais visível principalmente pelos inúmeros serviços e aplicações multimídia que estão surgindo no mercado, somando *streaming* de áudio e vídeo, acesso à Internet em banda larga, serviços de localização, jogos 3D multiusuários e *mashups* das mais diversas aplicações, além do uso para se comunicar via voz com outras pessoas.

2.1.1 A evolução dos dispositivos móveis

A transmissão de dados via celular não é tão recente. Há 15 anos tentativas de se construir acesso à internet pelo celular foram realizados, porém a velocidade de transmissão, o tamanho dos visores e o próprio conteúdo disponível na Web limitavam o uso pelo celular.

As necessidades dos usuários levaram as empresas fabricantes de dispositivos móveis a criarem dispositivos cada vez mais sofisticados em termos de tecnologia. A tecnologia dos aparelhos móveis evoluiu de forma mais rápida após o lançamento do iPhone pela Apple que combinava telefonia móvel convencional, gestão e armazenamento de músicas além de acesso à internet. Além de possuir uma interface amigável e tela sensível ao toque, este dispositivo era completamente reconfigurável através de aplicações (*apps*) com funções específicas. A partir daí, o celular virou um software, reprogramável.

Juntamente com os dispositivos móveis, foram evoluindo também as redes móveis, responsáveis por permitir os dispositivos trafegarem além de voz, dados, possibilitando a navegação na internet, compartilhamento de arquivos com outras pessoas, download de conteúdos multimídias, entre outros. Os tipos de rede mais novos e importantes hoje em dia são as redes 3G e 4G.

A terceira geração (3G) traz melhorias em diversos níveis como qualidade de voz, internet de alta velocidade, serviços multimídia, etc. Foi definida pelos seguintes requisitos: 1. Taxas mínimas de 144Kbps em espaços abertos ou 2Mbps dentro de infraestruturas; 2. Transferência de dados simétricos e assimétricos; 3. Transferência adicional de comutação por pacotes obrigatória; 4. Suporte a serviços IP; 5. Alta qualidade de voz; 6. Alta eficácia espectral. (MARTINHO, 2012).

A quarta geração (4G) chega como uma revolução computacional desde o surgimento da internet. Totalmente baseada por IP, alcançando convergência entre redes por cabo e sem fios com velocidades muito elevadas, mantendo assim a qualidade de serviço ponto a ponto. Camada de rede e inferiores em uma só camada e uma arquitetura de longo alcance sem fio. (MARTINHO, 2012).

2.1.2 Os principais tipos de dispositivos móveis

Um dispositivo móvel originalmente chamado de *Handheld* é um computador de bolso geralmente equipado com uma tela (dispositivo de saída) e um teclado (dispositivo de entrada). No caso dos *PDA's* entrada e saída combinam-se numa tela sensível ao toque.

Entre os dispositivos móveis mais comuns hoje em dia estão os Smartphones, PDA's, Celulares, Video Games portáteis, entre outros.

Com o passar dos tempos, os fabricantes foram integrando mais funcionalidades aos dispositivos móveis, principalmente aos telefones celulares. Juntamente com as funcionalidades, os aparelhos foram recebendo processadores mais poderosos, memórias maiores e mais rápidas e espaço amplo para armazenamento de dados. Com isto, os dispositivos foram se tornando verdadeiros computadores portáteis e por possuírem grande poder de processamento são conhecidos hoje como *Smartphones*: aparelhos celulares que incluem diversas funcionalidades além de realizar ligações telefônicas.

Evoluíram juntamente com os dispositivos móveis os sistemas operacionais dos mesmos, uma vez que devem controlar todas as funcionalidades incluídas nos aparelhos, como GPS, rede sem fio, processador e memórias (armazenamento e volátil), entre outras.

O mercado de hoje oferece uma gama de sistemas operacionais voltados para dispositivos móveis, alguns exemplos são: Android, BlackBerry, iOs, Symbian, Windows CE, Windows Phone 7, Windows XP Embedded e Windows Mobile.

2.1.3 Sistemas operacionais móveis

Até o mês de março de 2012, mais de 234 milhões de pessoas utilizaram smartphones só nos EUA, onde 51% dos aparelhos utilizavam o Sistema Operacional Android (Google), 30,7% o iOS (Apple), 12,3% SO da RIM (utilizado nos aparelhos da BlackBerry), 3,9% SO da Microsoft e 1,4% utilizavam o SO Symbian. (Figura 1).

Top Smartphone Platforms 3 Month Avg. Ending Mar. 2012 vs. 3 Month Avg. Ending Dec. 2011 Total U.S. Smartphone Subscribers Ages 13+ Source: comScore MobiLens			
	Share (%) of Smartphone Subscribers		
	Dec-11	Mar-12	Point Change
<i>Total Smartphone Subscribers</i>	100.0%	100.0%	N/A
Google	47.3%	51.0%	3.7
Apple	29.6%	30.7%	1.1
RIM	16.0%	12.3%	-3.7
Microsoft	4.7%	3.9%	-0.8
Symbian	1.4%	1.4%	0.0

Figura 1 - TOP Smartphone Platforms. Fonte: comScore, 2012

Os sistemas operacionais móveis mais utilizados hoje em dia com foco no mercado consumidor são: Google Android, Apple iOS e Windows Phone 7.

2.1.3.1 Apple iOS

O iOS é o sistema operacional móvel voltado para dispositivos móveis da Apple, a princípio para o iPhone mas foi aplicado também em iPod Touch, iPad e Apple TV. A interface com o usuário do iOS é fundamentada no conceito de manipulação direta através de multi-toques. (iOS, 2011).

A interação com o sistema operacional da Apple se dá através de toques na tela, deslizar os dedos sobre a mesma e toques em forma de “pinça” para ampliar ou reduzir uma imagem, por exemplo. Além disto, o iOS conta com a percepção de movimentação dos dispositivos móveis, através de acelerômetros contidos nos mesmos, respondendo à eles com ações específicas, como desfazer uma digitação ao agitar o aparelho e girar a tela de uma aplicação em três dimensões ao mudar a posição do dispositivo do modo retrato para o modo paisagem, e vice versa.

O iOS consiste em quatro camadas de abstração (Figura 2), sendo elas: camada *Core OS*, *Core Service*, *Mídia* e *Cocoa Touch*. (Apple, 2012).



Figura 2 - Camadas de Abstração do iOS. Fonte: Apple, 2012

O SO da Apple utiliza aproximadamente 600 megabytes de armazenamento no dispositivo instalado, podendo variar de acordo com o modelo.

A Cisco já havia utilizado o nome IOS no seu sistema de infraestrutura de redes há uma década antes do lançamento do iOS pela Apple, com isso, esta licenciou o nome “iOS” como marca registrada da Cisco. (PaidContent, 2010).

2.1.3.2 Microsoft Windows Phone 7

O Windows Phone 7 é um SO móvel desenvolvido pela Microsoft para suceder a plataforma Windows Mobile, (KOH, 2010) embora não seja compatível com plataforma anterior (ZIEGLER, 2010).

O Windows Phone 7 é novo em tudo em relação às plataformas Windows Mobile, inclusive a interface gráfica. Apresentando o *Metro*, a nova interface da Microsoft para o SO não permite interfaces personalizáveis, como acontecia nas versões anteriores, controlando rigorosamente todo o *hardware* do dispositivo móvel. (BUCHANAN, 2010).

Ao contrário do Windows Mobile, a nova plataforma é focada no mercado consumidor, enquanto a outra era focada no mercado empresarial. Desta forma, a Microsoft entra para disputar a grande fatia do mercado com outros grandes como a Google e a Apple. (MICROSOFT, 2011).

2.1.3.3 Google Android

O Google Android é um sistema operacional móvel que roda sobre o núcleo Linux, embora sua versão do núcleo não seja a oficial (STEVEN J, 2010). O Android permite o desenvolvimento de aplicações na linguagem de programação Java, através das bibliotecas de desenvolvimento distribuídas pela Google, o SDK Android. (SHANKLAND, 2007).

A proposta deste estudo é utilizar o sistema operacional Android. Esta escolha se deu devido ao fato de que este SO está em constante crescimento e já ocupa a maioria dos smartphones utilizados no mercado americano (comScore, 2012) e também nos demais países.

2.2 GPS

O GPS (Global Positioning System – Geo-posicionamento por Satélite, ou Sistema de Posicionamento Global) é um sistema de navegação por satélite que permite ao aparelho móvel, receber a posição em tempo real, sendo bastante útil em situações onde o usuário necessita saber sua localização.

O GPS surgiu para fornecer um novo sistema de navegação, capaz de suprir todas as limitações existentes nos sistemas mais antigos. Foi desenvolvido pelo Departamento de Defesa dos Estados Unidos da América – DoD (Department of Defense), com o intuito de ser o principal sistema de navegação das forças armadas americanas (MONICO, 2000).

É um sistema de abrangência global, funcionando em quaisquer condições climáticas, bastando que o usuário esteja em qualquer local da superfície terrestre.

Segundo Monico (2000), a informação da localização do usuário é exata, pois é calculada por, no mínimo, quatro satélites (Figura 3) disponíveis em órbita. Estes quatro satélites foram determinados para efetuar o cálculo das coordenadas da antena do aparelho móvel em relação a eles, podendo então definir o ponto exato. Do ponto de vista geométrico, apenas três distâncias, desde que não pertencentes ao mesmo plano, seriam suficientes.

Nesse caso, o problema se reduziria à solução de um sistema de três equações, a três incógnitas. A quarta medida é necessária em razão do não-sincronismo entre os relógios dos satélites e o do usuário, adicionando uma incógnita ao problema (MONICO, 2000).

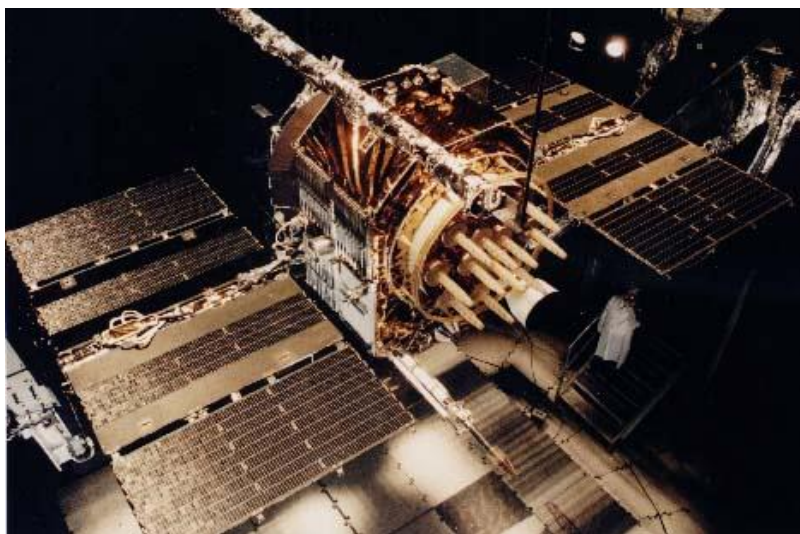


Figura 3 - Satélite Navstar. Utilizado no sistema GPS. Fonte: Navstar

Até meados do ano 2000, existiam dois tipos de serviços no sistema GPS, conhecidos como SPS (Standard Positioning Service ou Serviço de Posicionamento Padrão) e PPS (Precise Positioning Service ou Serviço de Posicionamento Preciso).

O SPS era o serviço disponível para todos os usuários, em qualquer parte do mundo, porém, possuía uma menor precisão, comparando ao PPS. Esta diferença era proposital, pois o DoD temia que o sistema fosse utilizado para oferecer risco à população. Hoje em dia a precisão do sistema GPS disponível para a população civil aumentou, sendo na casa de centímetros, para receptores Geodésicos e topográficos.

Os receptores da categoria navegação possuem precisão na casa de metros, porém, têm grandes vantagens em relação aos demais, como preço, tamanho, além da possibilidade de livre escolha do software que fará a interpretação das coordenadas e poderá exibir graficamente na tela do dispositivo o mapa do local, bem como a localização do usuário.

2.3 APLICAÇÕES MÓVEIS

Uma aplicação móvel nada mais é do que um programa feito especialmente para dispositivos móveis, respeitando seus limites de processamento e memória, tamanho de tela, etc.

Aplicações móveis apresentam várias características que agregam funcionalidades a seus usuários. A primeira delas é a mobilidade, a capacidade de manter voz constante e comunicação de dados enquanto se movimenta. (DIAS e FONTES, 2003).

O imediatismo é outro fator relevante para o uso das aplicações móveis, pois permite ao usuário obter conectividade independente de onde estiver através das redes móveis. Por fim, a localização provida por dispositivos GPS acoplados nos dispositivos permite ao usuário obter informações precisas e em tempo real sobre o seu posicionamento.

A combinação destas e outras características fornece uma grande faixa de possíveis aplicações a serem oferecidas aos usuários. Podemos citar algumas categorias em que as aplicações móveis se enquadram:

- *Comunicações* – E-mail, fax, mensagem unificada e acesso à internet;
- *Serviço de valor agregado* – (VAS – *Value added services*) – serviços de informação e jogos;
- *M-commerce* – venda a varejo, compra de bilhetes, serviços bancários e comércio financeiro;
- *Aplicação baseada em localização* – Navegação, condições de tráfego de veículos, localização de pessoas e hospitais, etc;
- *Aplicações verticais* – Gerenciamento de frota, alocação de recursos, etc;
- *Publicidade*.

3 PLATAFORMA DE DESENVOLVIMENTO

A principal IDE de desenvolvimento utilizada para o desenvolvimento Android é o Eclipse que, com o download de um único e simples plugin (o ADT) é possível o desenvolvimento completo para a plataforma, como a criação de projetos e telas através de editores gráficos, modo debug (inclusive direto do aparelho) e geração de pacotes para a distribuição das aplicações criadas. (GLOBALCODERS, 2010).

Além da IDE configurada é necessário que o SDK do Android também esteja instalado no computador. A primeira instalação do SDK é somente um gerenciador de plataformas e não conterá nenhuma versão de sistema operacional.

Antes de começar o desenvolvimento, foi feito *download* de todas as versões disponíveis do Android através do SDK Manager.

3.1 ANDROID

O Android é uma nova plataforma de desenvolvimento para aplicativos móveis como smartphones e contém um sistema operacional baseado em Linux, uma interface visual rica, GPS, diversas aplicações já instaladas e ainda um ambiente de desenvolvimento bastante poderoso, o Android SDK, inovador e flexível, que disponibiliza ferramentas e API's necessárias para a plataforma. (LECHETA 2009, 2010).

Segundo Tanenbaum (2003), um sistema operacional deve ser capaz de gerenciar processador, memória e outros dispositivos de entrada e saída, além de fornecer, aos programas de usuário, uma interface mais simplificada com o hardware.

Nos dispositivos móveis é necessário que o sistema operacional seja capaz de fazer um bom gerenciamento de processador e memória, pois comparado à outros computadores, eles possuem menor capacidade e necessitam de maior eficiência.

As empresas e os desenvolvedores buscam uma plataforma moderna e ágil para o desenvolvimento de aplicações corporativas para auxiliar em seus negócios e lucros. Já os usuários comuns buscam um celular com um visual elegante e moderno, de fácil navegação e uma infinidade de recursos. (LECHETA 2009, 2010).

Foi com base nas observações citadas acima que um grupo de empresas do mercado de telefonia resolveu fundar uma aliança, denominada Open Handset Alliance, ou simplesmente OHA. Composta por fabricantes como Acer, ASUS, Dell, Motorola, Samsung, entre várias outras (Figura 4), visava criar uma nova plataforma única para aparelhos móveis, celulares, smartphones, tablet's, onde aplicações poderiam ser desenvolvidas por qualquer desenvolvedor, instaladas em qualquer aparelho que possuísse a plataforma, de forma flexível e moderna, não limitando as aplicações à apenas uma parcela de usuários do mercado. (OHA, 2011)

<i>Operadoras Móveis</i>	<i>Empresas de Software</i>	<i>Empresas de Comercialização</i>	<i>Indústria de Semicondutores</i>	<i>Fabricantes de celulares</i>
<ul style="list-style-type: none"> • China Mobile • KDDI Corporation • NTT DoCoMo • Sprint Nextel • T-Mobile • Telecom Italia • Telefónica 	<ul style="list-style-type: none"> • Ascender Corporation • eBay • Esmertec • Google • LivingImage • NMS Communications • Nuance Communications • PacketVideo • SkyPop • SONiVOX 	<ul style="list-style-type: none"> • Aplix • Noser Engineering • The Astonishing Tribe • Wind River Systems 	<ul style="list-style-type: none"> • Audience • Broadcom Corporation • Intel Corporation • Marvell Technology Group • NVIDIA Corporation • Qualcomm • SiRF Technology Holdings • Synaptics • Texas Instruments 	<ul style="list-style-type: none"> • HTC • LG • Motorola • Samsung Electronics

Figura 4 - Lista das empresas que pertencem a Open Handset Alliance. Fonte: Devmedia – Java Magazine – Android, a nova plataforma móvel – Parte 1

Outra facilidade do Android é poder desenvolver suas aplicações utilizando a linguagem de desenvolvimento Java, uma das mais utilizada no mundo. (Figura 5) (TIOBE Software, 2012).

Position Mar 2012	Position Mar 2011	Delta in Position	Programming Language	Ratings Mar 2012	Delta Mar 2011	Status
1	1	=	Java	17.110%	-2.60%	A
2	2	=	C	17.087%	+1.82%	A
3	4	↑	C#	8.244%	+1.03%	A
4	3	↓	C++	8.047%	-0.71%	A
5	8	↑↑↑	Objective-C	7.737%	+4.22%	A
6	5	↓	PHP	5.555%	-1.01%	A
7	7	=	(Visual) Basic	4.369%	-0.34%	A
8	10	↑↑	JavaScript	3.386%	+1.52%	A
9	6	↓↓↓	Python	3.291%	-2.45%	A
10	9	↓	Perl	2.703%	+0.73%	A
11	13	↑↑	Delphi/Object Pascal	1.727%	+0.73%	A
12	30	↑↑↑↑↑↑↑↑	PL/SQL	1.418%	+1.01%	A
13	11	↓↓	Ruby	1.413%	-0.09%	A
14	23	↑↑↑↑↑↑↑↑	Transact-SQL	0.925%	+0.38%	A
15	15	=	Lisp	0.922%	-0.01%	A
16	22	↑↑↑↑↑↑	Visual Basic .NET	0.784%	+0.22%	A-
17	18	↑	Pascal	0.771%	+0.07%	A
18	32	↑↑↑↑↑↑↑↑	Logo	0.717%	+0.31%	A--
19	17	↓↓	Ada	0.633%	-0.09%	B
20	19	↓	NXT-G	0.604%	-0.04%	B

Figura 5 - Ranking Linguagens de Programação. Fonte: TIOBE Software, 2012

Desta forma, as empresas, usuários e desenvolvedores são beneficiados, pois as empresas podem disponibilizar serviços para usuários móveis, aumentando a sua área de atuação. Os usuários têm ao alcance das mãos um aparelho com interface gráfica bonita e de fácil utilização, diversas aplicações nativas, podendo ainda instalar aplicações criadas posteriormente à aquisição do aparelho, recursos desejados nos dispositivos móveis, como tocador de músicas digitais, leitor de livros

(eBook), scanner de código de barras, navegador de internet, GPS, câmera digital, etc.

Os desenvolvedores podem utilizar a API de desenvolvimento para Android disponibilizada gratuitamente na web para criar aplicações baseadas na linguagem Java.

3.1.1 Arquitetura

Sendo baseado no núcleo 2.6 do Linux, este é totalmente responsável pela gerência de memória, processos, threads, driver, entre outros. Desta forma, o Android não se preocupa com as tarefas básicas de um sistema operacional, criando apenas novos processos e repassando ao núcleo do Linux, o encarregando de gerenciar da melhor maneira os recursos do hardware, assim, as aplicações podem executar paralelamente a outras, não sendo necessário aguardar até o término da execução de outra tarefa.

A figura 6 demonstra a arquitetura do Android, bem como suas camadas, sendo elas: Aplicações (Applications), Ferramentas de Aplicação (Applications Framework), Bibliotecas (Libraries), Aplicação Android (Android Runtime) e o Núcleo Linux (Linux Kernel) utilizado.

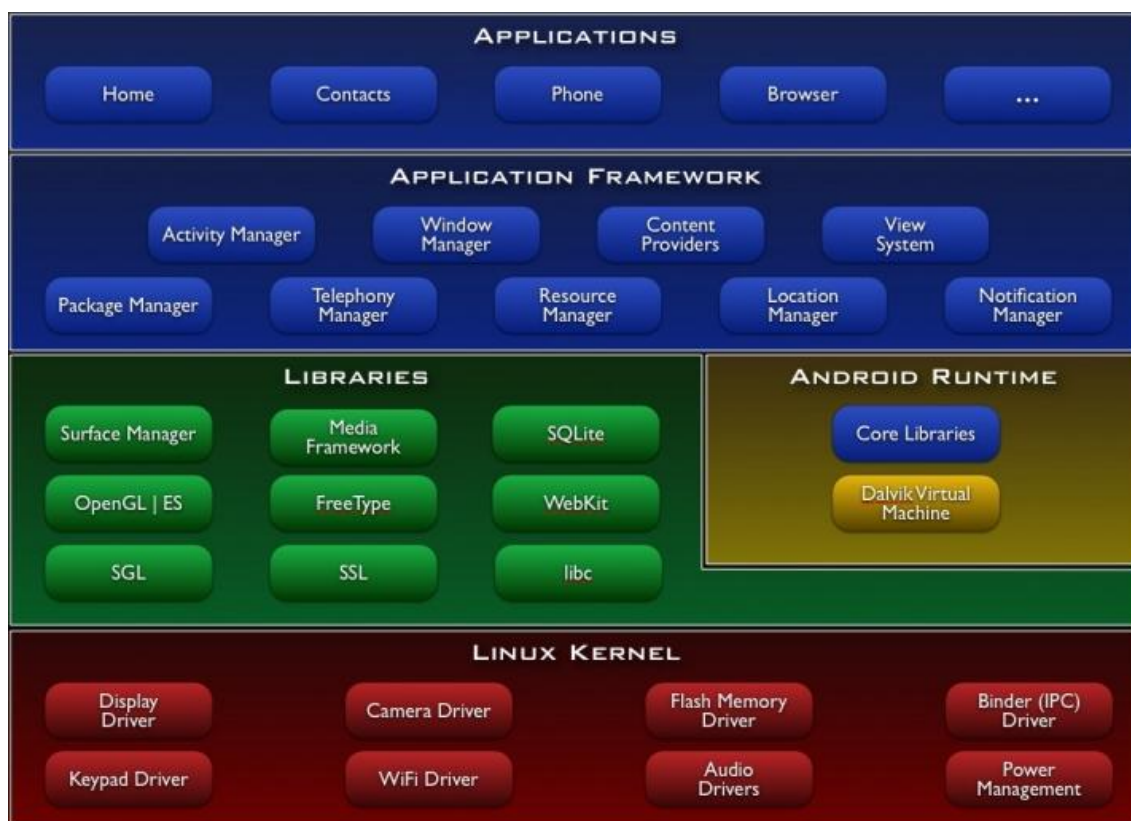


Figura 6 - Diagrama Arquitetura Android. Fonte: Project, 2011

3.1.1.1 Aplicações (Applications)

O Android acompanha um conjunto de aplicações, incluindo um cliente de e-mail, mensagem de texto, calendário, mapas, navegador web, agenda de contatos entre outros. Todas as aplicações são desenvolvidas na linguagem de programação Java. (Project, 2011).

Estas aplicações nativas acompanham qualquer dispositivo que possua o sistema Android instalado, independente da versão ou modelo do dispositivo móvel.

3.1.1.2 Ferramentas de aplicação (Application Framework)

As ferramentas de aplicação, ou o framework como é mais conhecido, são disponibilizadas para os usuários, de modo que, podem acessar e utilizar os mesmos recursos que as aplicações nativas utilizam. Desta forma, reutilizar funcionalidades já criadas e personalizar os componentes existentes é muito simples, evitando problemas de compatibilidade entre versões do Android e modelos de aparelhos móveis.

Dentro desta categoria, existem as seguintes categorias de serviços e sistemas:

- Views – É um conjunto rico e extensível de telas que pode ser utilizado para criar aplicações, incluindo listas, tabelas, caixas de texto, botões, além de um navegador baseado no já existente e contemplado pelo sistema;
- Content Providers – É responsável por permitir que as aplicações acessem dados de outros aplicativos, além de permitir que a sua aplicação compartilhe os dados com os demais. Pode-se citar como exemplos, os contatos, as imagens contidas no cartão de memória, entre outros;
- Resource Manager – É responsável por fornecer acesso à recursos não codificados, como String, gráficos e arquivos de layout;
- Notification Manager – Permite que as aplicações mostrem suas mensagens personalizadas na barra de status;
- Activity Manager – É responsável pelo ciclo de vida da aplicação, além de prover uma navegação entre os estágios da mesma.

3.1.2 Bibliotecas (Libraries)

Os componentes do sistema Android possuem diversas bibliotecas C/C++, que também estão disponíveis para os desenvolvedores através do framework de aplicação. Pode-se listar as principais bibliotecas abaixo:

- System C library – é uma distribuição de código fonte aberto proveniente da Universidade da Califórnia, Berkeley (BSD) (LEHEY, 1999) derivada da biblioteca padrão C, ou libc, adaptado para dispositivos que utilizam núcleo do Linux. (Biblioteca Sistema C, 2011);
- Media Libraries – Bibliotecas que suportam gravação e execução dos formatos mais populares de áudio e vídeo, além de imagens estáticas. Baseado no PacketVideo's OpenCore contempla formatos como AAC, AMR, JPG, MP3, MPEG4, PNG, entre outros (SALES e BARROS, 2008);
- Surface Manager – Gerencia o acesso à visualização de subsistemas e componentes sensíveis a camadas gráficas bi e tri-dimensionais (2D e 3D) para múltiplas aplicações;
- LibWebCore – É um moderno motor (engine) de navegação que dá suporte tanto ao navegador nativo do Android, quanto às visões internas da web;
- SGL – É um motor de gráficos 2D subjacentes;
- 3D Libraries – Baseado na API OpenGL ES 1.0, esta aplicação provê o uso de aceleração de hardware 3D, quando disponível, otimizando a execução de aplicativos que fazem seu uso;
- FreeType – Realiza a renderização de fontes vector e bitmap;
- SQLite – É um sistema de banco de dados relacional, leve e poderoso, disponível para todas as aplicações do sistema.

3.1.3 Aplicação Android (Android Runtime)

O Android inclui um conjunto de bibliotecas que fornece a maioria das funcionalidades disponíveis na biblioteca do Java.

Uma aplicação Android é interpretada de maneira análoga ao *byte-code* Java, com a diferença que o Android gera códigos Dalvik Executáveis (.dex) e não os *byte-code* (.class) do Java, tais códigos são interpretados pela Máquina Virtual Dalvik (MVD) (Project, 2011).

Criada por Dan Bornstein, a MVD é uma alteração da Máquina Virtual Java (JVM), a qual é otimizada para os objetivos que o Android visa suprir (BORNSTEIN, 2008).

Conforme citado anteriormente, toda aplicação Android roda em seu processo único, sendo uma única instância da MVD. O Dalvik foi inscrito de modo a proporcionar que o dispositivo possa executar várias VM's (Virtual Machine ou máquinas virtuais) de forma eficiente. A MVD executa os arquivos executáveis que possuem a extensão .dex (Dalvik Executable ou Executável Dalvik). Este formato é otimizado para ocupar a menor quantidade de memória possível. Esta máquina virtual é baseada em registradores.

Após a compilação, todos os arquivos .dex e outros recursos utilizados pela aplicação, imagens e sons por exemplo, são compactados em um arquivo do tipo .apk (Android Package File), sendo este arquivo a aplicação finalizada e pronta para ser distribuída e instalada em qualquer dispositivo com Android (LECHETA. 2009).

A MVD também é incluída no SDK do Android, onde transformam os códigos das classes Java (.class) em códigos Dalvik executáveis (.dex), que posteriormente serão executados pelo emulador Android (BORNSTEIN, 2008).

A MVD é altamente dependente do núcleo do Linux, pois necessita executar algumas funcionalidades de baixo nível, como por exemplo, o gerenciamento de memória.

3.1.4 Núcleo do Linux (Linux Kernel)

Conforme citado anteriormente, o Android utiliza a versão 2.6 do núcleo do Linux para serviços essenciais como segurança, gerenciamento de memória e processos, rede e modelo de driver. O núcleo atua também como uma camada de abstração entre hardware e software, isentando assim qualquer tratamento direto entre as aplicações e os recursos de memória, por exemplo.

Além das bibliotecas, os desenvolvedores têm a sua disposição o framework de aplicativo, que fornece componentes que auxiliam o desenvolvimento dos programas, permitindo a criação de listas, grades, caixas de texto, botões, etc. Alguns componentes do framework que pode-se citar: O gerenciador de atividades, que gerencia o ciclo de vida das aplicações e permite sua execução em segundo plano (GOOGLE INC., 2011a).

3.1.5 Desenvolvimento para Android

A linguagem de programação utilizada para se desenvolver para a plataforma Android é o Java, desta forma, existem algumas IDE's mais difundidas no mercado, sendo muito utilizadas, como o Eclipse e NetBeans, porém é possível desenvolver para Android com qualquer IDE, desde exista o plugin de desenvolvimento para a mesma.

A Google criou sua própria IDE de desenvolvimento para Android, a Google App Inventor, que possibilita os usuários criarem suas aplicações sem ter conhecimento em linguagem de desenvolvimento Java e lógica de programação, assim, eles podem criar aplicações arrastando os componentes existentes na API Android, criando telas com botões, imagens e funcionalidades.

Neste trabalho foi utilizada a IDE Eclipse por ser uma das mais difundidas no mundo. Ela é patrocinada pela IBM, é livre para o uso comercial e possui código aberto.

3.2 GOOGLE MAPS

O Google Maps é um projeto de grande sucesso da Google, com grande aceitação comercial e acadêmica é uma biblioteca para a criação e visualização de mapas, permitindo uma interação com o usuário e possibilitando uma utilização da localização rica.

Para integrar o Google Maps nos projetos Android a classe `MapView`, disponível da biblioteca `com.google.android.maps` deve ser utilizada. Como internamente essa classe necessita de internet, é necessário declarar a permissão de `INTERNET` no arquivo `AndroidManifest.xml`. (LECHETA, 2009).

Como a plataforma Android está completamente interligada aos serviços do Google e este é responsável pela elaboração da plataforma, ainda não existe forma de se obter esta biblioteca senão com um projeto Android.

A principal classe da biblioteca de mapas é o `MapView`, que é um objeto gráfico responsável pela visualização de mapas providos através da localização atual. É possível se alternar esta visualização entre três tipos diferentes de mapas (mapas de imagens de satélite, mapas topográficos e mapas de ruas). É possível

desenvolver aplicações robustas que compartilhem de todos recursos do Android utilizando essa biblioteca juntamente com outras da plataforma.

A classe MapController é responsável pelo controle do mapa, é capaz de alterar o nível de zoom, mudar a localização exibida, entre outras funções. Também é encontrado a classe OverLay que é geralmente utilizada para exibir um desenho especial, que pode ser desenhado a fim de exibir uma imagem que traga sentido a aplicação. (LECHETA, 2009).

Esta biblioteca possibilita a criação de diversas aplicações contemplando inúmeros recursos, desde a inclusão de imagens personalizadas e desenhos no mapa até a exibição do posicionamento exato do dispositivo móvel, através do GPS, como foi desenvolvido para este estudo de caso.

3.3 CÓDIGO DE CERTIFICAÇÃO DIGITAL

A plataforma Android só permite que aplicações assinadas digitalmente por um certificado digital sejam instaladas, sendo assim, é preciso assinar digitalmente a aplicação deste trabalho.

Utilizando a IDE Eclipse com o plugin ADT o certificado é gerado automaticamente, baseado em uma chave gerada no computador onde o ambiente de desenvolvimento está configurado, chamada *debug.keystore*, e o local pode ser verificado no Eclipse de acordo com a figura 7. (LECHETA. 2009)

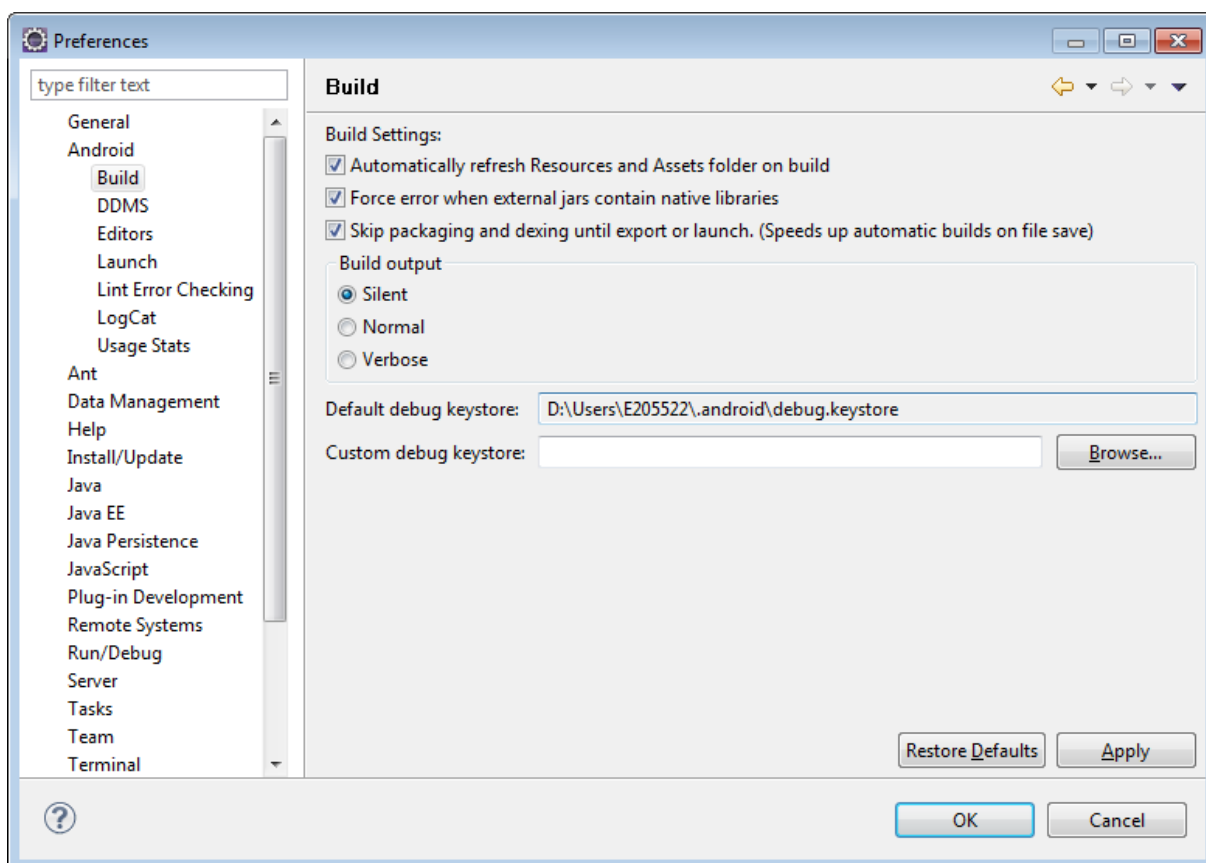


Figura 7 - Descobrimos a debug.keystore

Após descobrir a localização do arquivo *debug.keystore*, é preciso abrir o prompt e utilizar o comando *keytool* disponibilizado pelo JDK instalado na máquina, da seguinte forma:

```
"keytool -list -alias androiddebugkey -keystore <Caminho da debug.key> -
storepass android -keypass android"
```

3.4 CÓDIGO ACESSO AO GOOGLE MAPS

Para desenvolver aplicações que utilizam a API de mapas da Google, tanto para web, quanto para plataformas móveis, no caso Android, é necessária uma chave para utilização do serviço, sendo possível obter tal chave no endereço abaixo:

<<http://code.google.com/android/add-ons/google-apis/maps-api-signup.html>>

No endereço acima, o usuário deverá vincular uma conta válida da Google, além de informar o certificado digital da aplicação, de acordo com o item Código De Certificação Digital (item 3.3 Código de Certificação Digital).

Após digitar o certificado digital da aplicação no site informado anteriormente, é exibida uma tela contendo a chave para utilização do serviço (Figura 8).



Google Maps BETA

API do Google Maps
[Página inicial do Google Code](#) > [API do Google Maps](#) > Inscrição na API do Google Maps

Obrigado por se inscrever para uma chave de API do Android Maps!

Sua chave é:

```
0yFrKue7wTckuxCtwAWaHoV7AxwF9_M6ejJGYTA
```

Esta chave é válida para todos os aplicativos assinados com seu certificado cuja digital é:

```
1A:BC:0A:F9:89:C8:95:5C:A0:DE:E8:13:08:71:ED:D9
```

Aqui está um exemplo de layout xml para ajudá-lo a começar a usar os recursos de mapeamento:

```
<com.google.android.maps.MapView
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:apiKey="0yFrKue7wTckuxCtwAWaHoV7AxwF9_M6ejJGYTA"
/>
```

Confira a [Documentação da API](#) para obter mais informações.

©2008 Google - [Política de Privacidade](#) - [Empregos](#) - [Termos e Condições](#) - [Sobre o Google](#)

Figura 8 - Inscrição para uma Chave da API Google Maps

3.5 IDE ECLIPSE

Antigamente, basicamente antes do ano de 2000, havia vários ambientes de desenvolvimento para Java que tinham como foco a própria linguagem e não lidavam bem com a evolução dos servlets e paginas HTML.

A IBM, então, lançou no mercado inicialmente o WebSphere, em 1998, que funcionava basicamente como um repositório de servlets. Ainda não satisfeita e com a necessidade de um ambiente que complementasse suas ferramentas de desenvolvimento em Java que já existiam, a Empresa lançou o WebSphere Studio que possuía a possibilidade de edição de código HTML. (Yaw Tecnologia, 2012).

Em 2001, o WebSphere Studio evoluiu para o Eclipse. Ele continha uma gama de componentes gráficos que usavam componentes nativos do sistema operacional para construir a interface com o usuário.

O Eclipse é considerado hoje uma das principais IDEs Java e pode ser visto como um caso de sucesso amplo no desenvolvimento de softwares. Este modelo open source se destaca ganhando força e influência nas comunidades a favor de modelos de código aberto. (LUCKOW e MELO, 2010).

Possui como característica a forte orientação ao desenvolvimento baseado em *plugin-ins* e um amplo suporte aos desenvolvedores com centenas de *plugin-ins* que procuram atender as diversas necessidades dos programadores que utilizam esta ferramenta. (Eclipse, 2012).

O Eclipse IDE possui 11 releases principais, desde sua primeira versão até os dias de hoje. Cada release, a partir de 2006, foi lançado utilizando um codinome. Os mais recentes são:

- Eclipse Helios (versão 3.6) em 2010.
- Eclipse Indigo (versão 3.7) em 2011.

- Eclipse Juno (versão 4.2) em 2012.

Para 2013, o nome provisório do novo release seria Eclipse Kepler.

4 ESTUDO DE CASO

4.1 RESUMO

Este estudo de caso foi focado no desenvolvimento de uma aplicação que utiliza o GPS, recurso presente em grande parte dos smartphones que possuem o sistema operacional Android, bem como utilizar o SDK homônimo e a IDE de desenvolvimento de software bem difundida no mercado, o Eclipse.

4.2 DESCRIÇÃO DO PROBLEMA

Considere o seguinte cenário: Um usuário de transporte público está prestes a sair de casa em direção ao ponto de parada de ônibus, para esperar a condução desejada. Acontece que ele não possui o quadro de horários dos ônibus que passam pelo ponto de parada próximo, nem todos os ônibus que por ali passam.

O usuário deseja saber qual o ônibus está mais próximo do ponto, bem como este ônibus servirá para o seu deslocamento. Este programa será capaz de fornecer estas informações para o usuário, uma vez que ele se orientará pelo GPS do aparelho celular, informando qual ponto de ônibus é mais próximo do usuário, bem como os números de ônibus que passam por este ponto, inclusive o horário aproximado de chegada.

4.3 AMBIENTE DE DESENVOLVIMENTO

O protótipo desenvolvido neste trabalho é para ser utilizado em aparelhos móveis que possuem a plataforma Android. Para desenvolver aplicações para esta plataforma, é necessário, além da instalação do Android SDK, que é o kit de desenvolvimento disponibilizado pela Google, a IDE Eclipse (a utilizada é a versão Helios Service Release 2) com o plugin ADT (*Android Development Tools*).

Como existem diversos celulares Android no mercado, é possível que cada um deles tenha uma versão diferente. No Android, uma versão do sistema operacional é conhecida como plataforma, pode-se dizer que existem diversas plataformas diferentes do Android e cada plataforma possui um código identificador, chamado de API Level. (LECHETA, 2009).

A plataforma Android tem várias versões, desde a 1.0 até a 4.0. Até a versão 2.3.4, as versões eram voltadas para smartphones e as 3.x para tablets. A versão mais nova, 4.0 conhecida também como *Ice Cream Sandwich*, não tem distinção entre dispositivos, podendo funcionar tanto em *smartphones* quanto em *tablets*. (Android Ice Cream Sandwich, 2012). O programa desenvolvido utiliza a plataforma 2.1 (Éclair), API level 7 e com a biblioteca do Google APIS.

Como foi utilizada a plataforma Android, o ambiente de desenvolvimento para a mesma será composto pela IDE Eclipse juntamente com o *plugin* ADT, além do Android SDK que é o kit de desenvolvimento disponibilizado pelo Google. Além disto, é necessário a instalação do JDK versão 6 ou posterior para utilização do Eclipse.

O primeiro passo para começar o desenvolvimento é instalar as ferramentas necessárias. O download do eclipse está disponível no site <http://www.eclipse.org/>.

Depois de baixar o Eclipse, também é necessário instalar o Android SDK, disponível em <http://developer.android.com/sdk/index.html>.

Finalmente é necessário instalar o Android Development Tools plugin (ADT plugin) no eclipse. O plugin está disponível em

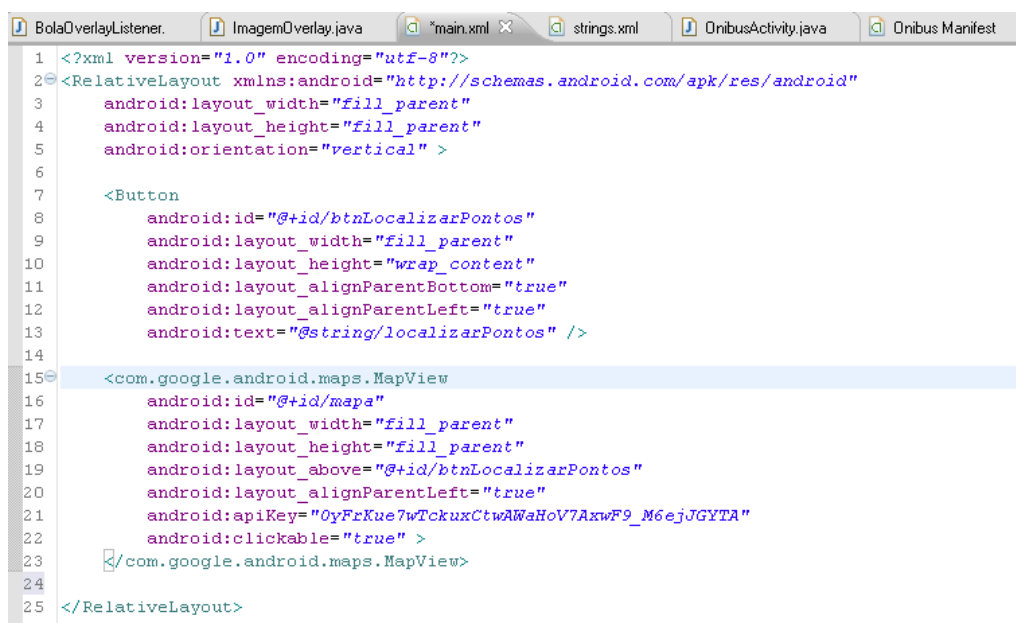
<http://developer.android.com/sdk/eclipse-adt.html>, juntamente com as instruções de instalação.

4.4 O PROJETO ANDROID

Para desenvolver a aplicação, um novo projeto Android foi criado no Eclipse, utilizando a API *Google Maps* 7 (plataforma Android 2.1). O projeto recebeu o nome de “Ônibus” e seu layout contempla um mapa que exibe os pontos de ônibus próximos ao dispositivo móvel quando selecionado o botão “Localizar Pontos Próximos”.

No Android existem diversas formas de se criar um *layout* através de gerenciadores. Esses gerenciadores utilizam o componente *View* do Android. Existem dois tipos de componentes: os chamados *Widgets* e os gerenciadores de *layout*. Um *widget* é um componente que é herdeiro diretamente da classe *View*. Já os gerenciadores de *layouts* são os *layouts* em si. (LECHETA, 2009).

O botão “Localizar Pontos Próximos” é um *Widget Button* e o mapa é um *MapView* disponibilizado pela API *Google Maps*. Para se incluir o *MapView* na aplicação, deve-se utilizar esta classe no arquivo XML de *layout* da aplicação, juntamente com a chave do *Google Maps* obtida anteriormente, conforme a figura 9.



```

1 <?xml version="1.0" encoding="utf-8"?>
2 <RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
3     android:layout_width="fill_parent"
4     android:layout_height="fill_parent"
5     android:orientation="vertical" >
6
7     <Button
8         android:id="@+id/btnLocalizarPontos"
9         android:layout_width="fill_parent"
10        android:layout_height="wrap_content"
11        android:layout_alignParentBottom="true"
12        android:layout_alignParentLeft="true"
13        android:text="@string/localizarPontos" />
14
15    <com.google.android.maps.MapView
16        android:id="@+id/mapa"
17        android:layout_width="fill_parent"
18        android:layout_height="fill_parent"
19        android:layout_above="@+id/btnLocalizarPontos"
20        android:layout_alignParentLeft="true"
21        android:apiKey="0yFrKue7wTckuxCtwAWaHoV7AxwF9_M6ejJGYTA"
22        android:clickable="true" >
23    </com.google.android.maps.MapView>
24
25 </RelativeLayout>

```

Figura 9 - Inclusão da classe MapView

4.4.1 A classe GeoPoint

Esta classe é disponibilizada pela API do *Google Maps* e é utilizada para representar uma localização usando as coordenadas latitude e longitude em microdegrees. Esta notação se refere a um grau multiplicado por $1E6^{11}$, sendo necessária para que seja criado um objeto *GeoPoint* e será através deste que as imagens do mapa e o mapa serão centralizados.

Para este trabalho foi criada uma classe auxiliar que estende a classe *GeoPoint* chamada de *Coordenada* que pode ser visualizada no apêndice A. Esta classe poderá receber as coordenadas em formato latitude e longitude além de microdegrees e um objeto *Location*.

4.4.2 Overlays e Mapa

Um *Overlay* é qualquer imagem exibida sobre o mapa. Para controlar os overlays e a exibição do mapa no Android é utilizada a classe `MapController`. Outras funcionalidades do mapa também são controladas por esta classe, como o *zoom* por exemplo.

Um overlay é simplesmente um *widget* da classe `View`, ele é utilizado para exibir um desenho especial, que pode ser desenhado e adicionado ao `MapView` para dar uma aparência personalizada e profissional a aplicação. (LECHETA, 2009).

Neste trabalho foram criados dois *overlays*, um desenho de um círculo vermelho que é feito pela classe `CirculoOverlayListener` e uma imagem de um ponto de ônibus que representa os pontos existentes no mapa.

A classe `CirculoOverlayListener` estende a classe `CirculoOverlay` que faz o desenho de um círculo no mapa representando a localização do dispositivo móvel no momento e implementa a classe `LocationListener`, que através do método `onLocationChanged`, que é chamado a cada mudança de localização, o círculo é redesenhado no último local.

O outro overlay utilizado que representa os pontos de ônibus no mapa é criado através da classe `ImagemOverlay`. Esta classe recebe um objeto do tipo `Coordenada` e insere no mapa a imagem na posição latitude e longitude informada quando o botão “Localizar Pontos Próximos” é selecionado.

4.4.3 LocationManager

Esta classe é responsável por receber as atualizações dos provedores de localização, o GPS no caso deste trabalho, e sempre que a aplicação é inicializada, o método `requestLocationUpdates` é chamado passando o provedor de localização, tempo de atualização (em milissegundos), a distância mínima para atualização e a classe que implementa os métodos exigidos pela interface `LocationListener`: `onLocationChanged`, `onProviderDisabled`, `onProviderEnabled`, `onStatusChanged`.

No método `onLocationChanged` é chamado quando a localização do dispositivo móvel é alterada, nele é criada uma nova instância da classe `Coordenada`, com a coordenada atual que a `LocationManager` forneceu, alterada a posição do `Overlay` que representa a localização atual, e direcionado para essa coordenada, o `MapController` centraliza o mapa a essa localização e objeto `MapView` é invalidado com para que o mapa seja redesenhado com a posição atualizada.

4.4.4 Classes auxiliares

Foram criadas duas classes auxiliares que contém dados para o programa, são elas `DadosOnibus` e `HorarioOnibus`. Elas contém informações sobre os ônibus que passam pelos pontos exibidos no mapa.

A classe `DadosOnibus` pode ser visualizada no apêndice B. Ela possui as seguintes informações: Número do ônibus, nome do ônibus e os horários (compostos por uma coleção da classe `HorarioOnibus`). A classe `HorarioOnibus` por sua vez possui a hora e o minuto em que o ônibus passará pelo ponto, podendo ser visualizada no apêndice C.

O diagrama de classes do projeto também pode ser visualizado no apêndice D, contendo as classes utilizadas bem como seus relacionamentos.

5 APLICATIVO

O trabalho desenvolvido é uma aplicação que mostra a localização atual do dispositivo móvel em um mapa através da API do *Google Maps* utilizando as coordenadas informadas pelo GPS. A localização atual é demonstrada por um círculo vermelho (Figura 10). Possui também um botão “Localizar Pontos Próximos” que ao ser pressionado exibe no mapa os pontos de ônibus próximos ao dispositivo móvel.

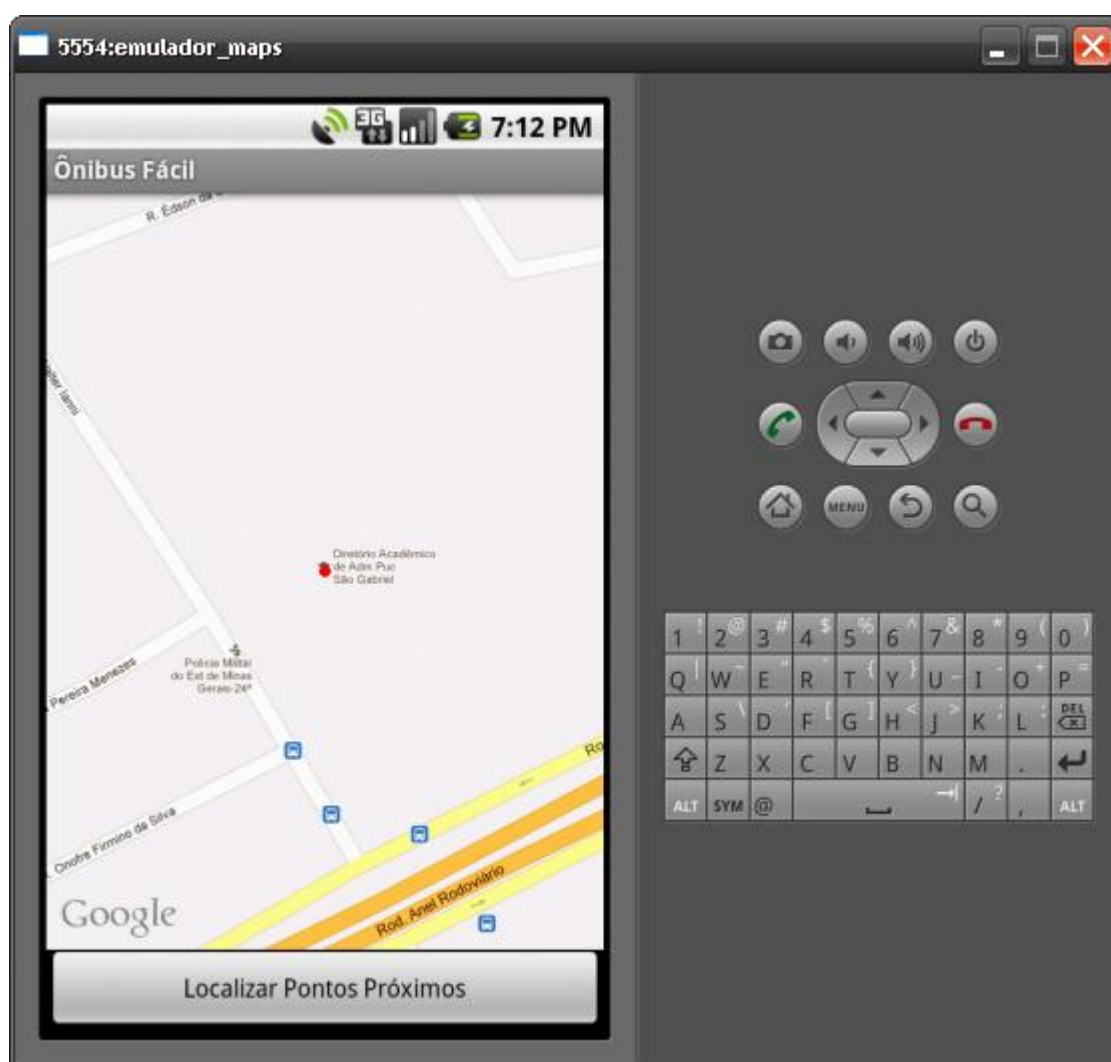


Figura 10 - Localização do dispositivo móvel

Ao pressionar o botão “Localizar Pontos Próximos” o aplicativo carrega no mapa os pontos de ônibus próximos, possibilitando ao usuário selecionar um ponto para que seja exibida uma lista contendo os ônibus que passam por ele. (Figura 11).



Figura 11 - Ponto de ônibus selecionado pelo usuário

Ao selecionar um ônibus contido na lista, o sistema informa ao usuário o próximo horário em que ele passará pelo ponto bem como o tempo restante para tal. (Figura 12).



Figura 12 - Próximo horário do ônibus selecionado

6 Conclusão

Com o crescimento contínuo de informações disponíveis cabe às pessoas saber utilizá-las, no caso dos ônibus os horários em que os mesmos passarão por um dado ponto de parada são muito importantes para se programar, evitando atrasos e perda de tempo parado aguardando o mesmo. Aplicações que tratam tais informações devem ser desenvolvidas oferecendo bem estar e conforto ao usuário, ainda mais em épocas onde o uso de *smartphones* e aparelhos celulares é praticamente indispensável e encontra-se numa contínua crescente.

Este trabalho apresentou o desenvolvimento de uma aplicação, com intuito social de compartilhar informações sobre a localização de pontos de ônibus bem como os ônibus que passam por eles e os horários dos mesmos, para dispositivos móveis que possuem o sistema operacional móvel Android.

A API de desenvolvimento para o Android, disponibilizada pelo Google, é facilmente integrada com a IDE Eclipse, facilitando o desenvolvimento de aplicações por desenvolvedores que já utilizavam tal ambiente. A linguagem Java também é outro facilitador, uma vez que é utilizada para desenvolver aplicações Android.

Embora existam facilitadores para o desenvolvimento de aplicações para este sistema operacional móvel, ele possui peculiaridades que podem demandar certo tempo de aprendizado, até o desenvolvedor abstrair a forma de desenvolvimento, como de *layouts* das telas, a classe “R” que contém os itens e componentes existentes na aplicação, como imagens e componentes de *layout* inseridos nos arquivos xml, por exemplo.

O desenvolvimento deste trabalho fez uso de diversas funcionalidades disponibilizadas pela plataforma Android e os dispositivos móveis, dentre elas podemos citar os *widgets* de *layout*, botões, a API de Mapas do Google, manipulação de GPS, leitura de xml, etc.

O aplicativo pode possuir novas funcionalidades incorporadas em trabalhos futuros, como:

- Interligação com WebServices providos pelas empresas de transporte público informando a posição real dos ônibus através de GPS e informando a mesma ao usuário;
- Informar ao usuário também os horários e localização dos trens urbanos (metrô).

BIBLIOGRAFIA

Android Ice Cream Sandwich. (2012). Disponível em: <<http://www.android.com/about/ice-cream-sandwich/>> Acesso em: 09/04/2012

Android Tools. (2011). Disponível em: <<http://www.andriodtools.com/>> Acesso em: 30/05/2011

Apple. Survey the Major Frameworks. (2012). Disponível em: <<https://developer.apple.com/library/ios/#referencelibrary/GettingStarted/RoadMapiOS/Frameworks/SurveytheMajorFrameworks/SurveytheMajorFrameworks/SurveytheMajorFrameworks.html>>. Acesso em 31/05/2012

Biblioteca Sistema C. (2011). Disponível em: <<http://www.netmite.com/android/mydroid/1.5/bionic/libc/docs/OVERVIEW.TXT>> Acesso em: 30/05/2011

BORNSTEIN, D. Dalvik Virtual Machine. (2008). Disponível em: <<http://www.dalvikvm.com/>> Acesso em: 26/05/2012

BUCHANAN, Matt. Windows Phone 7 Series: Everything Is Different Now. (2010). Disponível em: <<http://gizmodo.com/5471805/windows-phone-7-series-everything-is-different-now>>. Acesso em: 30/05/2012

DIAS, Klessis Lopes; FONTES, Wescley Pimentel. (2003). Desenvolvimento de Aplicações para Dispositivos Móveis utilizando a Plataforma J2ME. Belém

Eclipse. About the Eclipse Foundation. (2012). Disponível em: <<http://www.eclipse.org/org/>>. Acesso em: 05/05/2012

Embarcado Básico. (2012). Disponível em: <embarcadosbasico.wordpress.com> Acesso em: 02/04/2012

iOS. (2011). Disponível em: <http://pt.wikipedia.org/wiki/IOS#cite_note-camadas-0>. Acesso em 28/05/2012

KOH, Damian. Microsoft sobre o Windows Phone 7. CNET Asia. (2010). Disponível em: <<http://asia.cnet.com/qanda-microsoft-on-windows-phone-7-series-62061278.htm>>. Acesso em: 30/05/2012.

LEHEY, Greg. Explicando o BSD. (2012). Disponível em: <http://doc.fug.com.br/doc/pt_BR.ISO8859-1/articles/explaining-bsd/>. Acesso em: 03/05/2012

MARIN, Neto. (2011). Devmedia – Java Magazine – Android, a nova plataforma móvel – Parte 1. Disponível em: <<http://www.devmedia.com.br/android-a-nova-plataforma-movel-parte-i/8431>>. Acesso em: 05/06/2011

GLOBALCODERS, Começando o Desenvolvimento com Android. (2010). Disponível em: <<http://blog.globalcode.com.br/2010/04/comecando-o-desenvolvimento-com-android.html>>. Acesso em: 08/06/2012

Google Inc. (2011a). Android - an open handset alliance project. Disponível em: <<http://code.google.com/android/toolbox/custom-components.html>>. Acesso em: 03/05/2012

GPS, Wikipedia (2011). Disponível em: <http://pt.wikipedia.org/wiki/Sistema_de_posicionamento_global> Acesso em: 02/06/2011

Latitude e Longitude, Instrumentos de Medição, Ciência Viva (2012). Disponível em:<<http://www.cienciaviva.pt/latlong/anterior/gps.asp>>. Acesso em 26/05/2012

LECHETA, R. R. Google Android: Aprenda a criar aplicações para dispositivos móveis com o Android SDK. Novatec, 1st edition, 2009.

LEMONS, André: Cidade e mobilidade. Telefones celulares, funções pós-massivas e territórios informacionais. MATRIZES, N 1, Outubro de 2007.

LUCKOW, Décio Heinzelmann; MELO, Alexandre Altair de. Programação Java para a Web: Aprenda a desenvolver uma aplicação financeira pessoal com as ferramentas mais modernas da plataforma Java. 1 ed. São Paulo: Novatec, 2010.

MARTINHO, D.S. Evolução das redes móveis e evolução paralela dos conceitos de qualidade. Disponível em: <<http://pt.calameo.com/read/0005356552002434cdde4>>. Acesso em: 26/05/2012

MICROSOFT. Nokia and Microsoft Announce Plans for a Broad Strategic Partnership to Build a New Global Mobile Ecosystem. (2011). Disponível em: <<http://www.microsoft.com/en-us/news/press/2011/feb11/02-11partnership.aspx>>. Acesso em: 30/05/2012

MONICO, J F G. Posicionamento pelo NAVSTAR-GPS. 2000. Disponível em: <http://books.google.com.br/books?hl=pt-BR&lr=lang_pt&id=n3q4ypMODpEC&oi=fnd&pg=PA13&dq=GPS&ots=6VKmC4KzBO&sig=R23EcnivajAGPP_SZlqR8HT-Ws0#v=onepage&q&f=false> Acesso em: 02/06/2011

Navstar. (2011). Disponível em: <<http://pt.wikipedia.org/wiki/Ficheiro:Navstar.jpg>> Acesso em: 02/06/2011

OHA. (2011). Disponível em: <http://www.openhandsetalliance.com/oha_members.html>. Acesso em: 31/05/2011

PaidContent. Apple Avoids iPhone - Like Trademark Battle Thanks To Cisco, FaceTime Deals. (2010). Disponível em: <<http://paidcontent.org/2010/06/07/419-deja-vu-apples-new-ios-brand-is-already-used-by-cisco/>>. Acesso em: 31/05/2012

comScore. Plataformas Utilizadas nos Smartphones (EUA). (2012). Disponível em: <http://www.comscore.com/Press_Events/Press_Releases/2012/5/comScore_Reports_March_2012_U.S._Mobile_Subscriber_Market_Share>. Acesso em: 01/06/2012

Project, A. O. S. O que é o Android. (2011). Disponível em: <<http://developer.android.com/guide/basics/what-is-android.html>> Acesso em: 30/05/2011

SALES, Fábio de Oliveira; BARROS, Kecyo Sacramento. (2008). Solução de Integração entre JAVA, Web Services e Android, utilizando a Arquitetura SOA. Salvador.

SHANKLAND, Stephen. "Google's Android parts ways with Java industry group", CNET News. (2007). Disponível em: <http://news.cnet.com/8301-13580_3-9815495-39.html>. Acesso em 28/05/2012.

Sistema de Posicionamento Global, Departamento de Geodésia. IG/UFRGS. (2004) Disponível em: <http://www.ufrgs.br/museudetopografia/Artigos/Sistema_de_%20Posicionamento_Global.pdf>. Acesso em: 26/05/2012

STIVANIN, Taíssa. (2010). Mundo já tem 5 bilhões de telefones celulares. RFI Português. Disponível em: <<http://www.portugues.rfi.fr/mundo/20100716-mundo-ja-tem-5-bilhoes-de-telefones-celulares>>. Acesso em: 26/05/2012

STEVEN J. Vaughan-Nichols. (2010) .Android/Linux kernel fight continues. Disponível em: <http://blogs.computerworld.com/16900/android_linux_kernel_fight_continues>. Acesso em: 30 de Maio de 2012.

TANENBAUM, A. S. (2003). Sistemas Operacionais Modernos. Editora Pearson, 2 Edição. São Paulo.

TEIXEIRA, M. (2010) Evolução de redes de telefonia móvel. MBB Engagement Consultant.

TIOBE Software. (2012). Disponível em: <<http://www.tiobe.com/index.php/content/paperinfo/tpci/index.html>> Acesso em: 20/03/2012

Yaw Tecnologia. Eclipse, o ecossistema por trás do IDE. (2012). Disponível em: <<http://www.yaw.com.br/artigos/view/eclipse-o-ecossistema-por-tras-do-ide/>>. Acesso em: 05/05/2012

ZIEGLER, Chris. Microsoft sobre desenvolvimento para Windows Phone 7 à frente da GDC: Silverlight, XNA, e sem compatibilidade com versões anteriores. (2010) Disponível em: <<http://www.engadget.com/2010/03/04/microsoft-talks-windows-phone-7-series-development-ahead-of-gdc/>>. Acesso em: 30/05/2012

APÊNDICE A – CLASSE COORDENADA.JAVA

```

package com.onibus;

import android.location.Location;

import com.google.android.maps.Geopoint;

/**
 * Classe auxiliar Coordenada
 * Permite criar um objeto Geopoint através de coordenadas latitude e longitude
 * @author Paulo Vítor
 */
public class Coordenada extends Geopoint {

    /**
     * Construtor que recebe valores em graus * 1E6 (microdegrees)
     * @param latitudeE6
     * @param longitudeE6
     */
    public Coordenada(int latitudeE6, int longitudeE6) {
        super(latitudeE6, longitudeE6);
    }

    /**
     * Construtor que recebe a latitude e longitude e converte para "graus * 1E6"
     * @param latitude
     * @param longitude
     */
    public Coordenada(double latitude, double longitude) {
        this((int) (latitude * 1E6), (int) (longitude * 1E6));
    }

    /**
     * Construtor que recebe um objeto 'Location' diretamente recebido do GPS
     * @param location
     */
    public Coordenada(Location location) {
        this(location.getLatitude(), location.getLongitude());
    }

    @Override
    public int getLatitudeE6() {
        return super.getLatitudeE6();
    }

    @Override

```



```
public int getLongitudeE6() {  
    return super.getLongitudeE6();  
}  
  
public double getLatitude() {  
    return super.getLatitudeE6() / 1E6;  
}  
  
public double getLongitude() {  
    return super.getLongitudeE6() / 1E6;  
}  
}
```

APÊNDICE B – CLASSE DADOSONIBUS.JAVA

```
package com.onibus;

import java.util.List;

/**
 * Classe que contém as informações de um ônibus
 *
 * @author Paulo Vitor
 */
public class DadosOnibus {

    private String numero;
    private String nome;
    private List<HorarioOnibus> horarios;

    public DadosOnibus(String numero, String nome, List<HorarioOnibus>
horarios) {
        this.numero = numero;
        this.nome = nome;
        this.horarios = horarios;
    }

    public String getNumero() {
        return numero;
    }

    public void setNumero(String numero) {
        this.numero = numero;
    }

    public String getNome() {
        return nome;
    }

    public void setNome(String nome) {
        this.nome = nome;
    }

    public List<HorarioOnibus> getHorarios() {
        return horarios;
    }

    public void setHorarios(List<HorarioOnibus> horarios) {
        this.horarios = horarios;
    }
}
```

APÊNDICE C – CLASSE HORARIOONIBUS.JAVA

```
package com.onibus;

/**
 * Classe que possui um horário de um ônibus
 * @author Paulo Vitor
 */
public class HorarioOnibus {

    private int hora;
    private int minuto;

    public HorarioOnibus (int hora, int minuto){
        this.hora = hora;
        this.minuto = minuto;
    }

    public int getHora() {
        return hora;
    }
    public void setHora(int hora) {
        this.hora = hora;
    }
    public int getMinuto() {
        return minuto;
    }
    public void setMinuto(int minuto) {
        this.minuto = minuto;
    }
}
```

APÊNDICE D – DIAGRAMA DE CLASSES

