

UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ  
DEPARTAMENTO ACADÊMICO DE INFORMÁTICA - DAINF  
TECNOLOGIA EM SISTEMAS PARA INTERNET

LUÍS HENRIQUE TIBURCIO FERRACINI

PLATAFORMA ANDROID™ EM AMBIENTE CORPORATIVO

TRABALHO DE CONCLUSÃO DE CURSO



**UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ  
DEPARTAMENTO ACADÊMICO DE INFORMÁTICA - DAINF  
TECNOLOGIA EM SISTEMAS PARA INTERNET**

**LUÍS HENRIQUE TIBURCIO FERRACINI**

**PLATAFORMA ANDROID™ EM AMBIENTE CORPORATIVO.**

**TRABALHO DE CONCLUSÃO DE CURSO**

**CURITIBA**

**2012**

**UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ  
DEPARTAMENTO ACADÊMICO DE INFORMÁTICA – DAINF  
TECNOLOGIA EM SISTEMAS PARA INTERNET**

**LUÍS HENRIQUE TIBURCIO FERRACINI**

**PLATAFORMA ANDROID™ EM AMBIENTE CORPORATIVO.**

Trabalho de Conclusão de Curso  
apresentado à UTFPR como requisito  
parcial para obtenção do título de  
Tecnólogo em Sistemas para Internet.

Orientador: Prof. Leandro Batista de  
Almeida

**CURITIBA**

**2012**

Ferracini, Luís Henrique Tiburcio

Plataforma Android™ em ambiente corporativo.

40 p.

Trabalho de Diplomação – Universidade Tecnológica Federal do Paraná. Curso de Tecnologia em Sistemas para Internet.

## SUMÁRIO

<b>LISTA DE ABREVIATURAS E SIGLAS .....</b>	<b>5</b>
<b>LISTA DE FIGURAS .....</b>	<b>5</b>
<b>LISTA DE TABELAS .....</b>	<b>5</b>
<b>1. INTRODUÇÃO .....</b>	<b>8</b>
1.1. apresentação.....	8
1.2. justificativa.....	11
<b>2. OBJETIVO .....</b>	<b>12</b>
2.1. Objetivo geral.....	12
3.1. Objetivo específico.....	12
<b>3. REFERENCIAL TEÓRICO.....</b>	<b>13</b>
3.1 OPEN HANDSET ALLIANCE .....	13
<b>4. SISTEMA ANDROID™ .....</b>	<b>15</b>
4.1. ANDROID™SDK.....	16
4.2. ANDROID™SDK.....	19
4.3. SERVICE .....	21
4.4. INTENT .....	21
<b>5. WEB SERVICE .....</b>	<b>23</b>
<b>6. FERRAMENTAS DE DESENVOLVIMENTO .....</b>	<b>24</b>
6.1 ECLIPSE@.....	24
6.2 Netbeans@.....	24
<b>7. ESTUDO DE CASO STATIONMOB .....</b>	<b>26</b>
7.1 MATERIAIS E MÉTODOS .....	26
<b>8. CONCLUSÃO .....</b>	<b>36</b>
<b>REFERÊNCIA .....</b>	<b>37</b>
<b>ANEXO 1 - DOCUMENTAÇÃO DO WEBSERVICE .....</b>	<b>39</b>
<b>ANEXO 2 - DOCUMENTAÇÃO DO SISTEMA STATIONMOB .....</b>	<b>40</b>

## LISTA DE ABREVIATURAS E SIGLAS

**IDE:** Integrated Development Environment

**JVM:** Java Virtual Machine

**API:** Application Programming Interface

**APK:** Android Application Package

**SDK:** Software Development Kit

**ADT:** Android Development Tools

## LISTA DE FIGURAS

Figura 1 - Arquitetura Android™ .....	17
Figura 2 - Ciclo de vida Android™ .....	20
Figura 3 - Arquitetura StationMob.....	27
Figura 4 - Estrutura do projeto.....	29
Figura 5 - Menu principal – StationMob .....	30
Figura 6 - Lista de postos – StationMob .....	32
Figura 7 - Mapa de postos – StationMob.....	34
Figura 8 - Diagrama de casos de uso - WebService .....	39
Figura 9 - Diagrama de classes - StationMob.....	40

## LISTA DE TABELAS

Tabela 1 - Uso do celular corporativo .....	10
Tabela 2 - Tipo de dispositivo móvel mais utilizado.....	10

## RESUMO

A comodidade oferecida pelos aparelhos móveis gera um aumento crescente, e esperado, na população de usuários de smartphones. Estes equipamentos funcionam por meio de sistemas operacionais, dentre eles o Android™, que disponibilizam ao usuário ferramentas básicas tais como: agenda, calculadora e clientes de e-mail. Baseando-se na premissa anterior, o presente estudo tem como objetivo central demonstrar como a plataforma Android™ em dispositivos móveis faz com que as tarefas sejam automatizadas, facilitando, através de integrações entre os sistemas, as atividades nos mais diversos ambientes, com destaque para o meio corporativo. Visando demonstrar a facilidade gerada pela automatização dos sistemas, o estudo em questão explora as particularidades da integração de uma aplicação móvel para monitoração de postos de gasolina, através da comunicação de um servidor de informações.

**Palavras-chave:** Sistemas operacionais, Android™, WebServices, integração, smartphone.

## ABSTRACT

The convenience offered by mobile devices generates a steady and expected increase on the smartphone users' population. These devices work by means of operating systems, including Android™, which provide basic tools to users such as: calendar, calculator and e-mail clients. Based on the herein premise, the present study aims to demonstrating how the Android platform for mobile devices automate the tasks, facilitating the activities, through systems integration, into different environments, with emphasis on the corporative field. In order to demonstrate that automated systems can make thing easier, the present study explores the specific integration of a mobile application for monitoring gas stations, through the communication with an information server.

**Keywords:** Operating systems, Android, Web Services, integration, *smartphone*.



## 1. INTRODUÇÃO

### 1.1. APRESENTAÇÃO

O cenário atual da telefonia móvel, com seu elevado número de vendas de aparelhos móveis e a tendência mundial de praticidade, nos leva cada vez mais a crer na união das funcionalidades, tais como, câmeras fotográficas, calculadoras, agendas, tocadores de músicas, jogos, Internet, entre outras funcionalidades, antes usadas separadamente em outros tipos de aparelhos. Os celulares, ao longo dos últimos anos, vêm unificando tais funcionalidades que anteriormente eram comercializadas em tecnologias diferentes e dissociadas. (TELECO, 2010)

A escolha de uma plataforma ideal para o desenvolvimento de um projeto significa optar por uma solução que forneça os melhores benefícios, em termos de custos, eficiência e tempo de desenvolvimento esperados para a finalização do projeto, pontos estes cruciais no desenvolvimento de aplicações com escopo corporativo. Sendo assim, o presente documento tem como objetivo fazer uma explanação e demonstração sobre a importância e viabilidade do uso da plataforma Android em ambientes corporativos.

Uma vez que o Android apareceu com a primeira versão de seu sistema operacional no mercado, existia apenas um pequeno número de aparelhos celulares disponíveis para apoiá-lo. Como acontece com qualquer lançamento antecipado, houve uma probabilidade de haver mudanças regulares, melhorias no *software* e nas bibliotecas de desenvolvimento (MEIER, 2010).



Gráfico 1 - Celulares no mundo

Fonte: TELECO

Com vendas que ultrapassam 5,3 bilhões de celulares no mundo, em que as funcionalidades mais procuradas pelos usuários, além das mencionadas no parágrafo anteriormente, são os recursos como transferência de dados sem fio, ótima interface visual, sistemas de posicionamento global (*GPS*), acesso à Internet, televisão digital, despertam maior atenção dos fabricantes de celulares nesses tipos de tecnologias dentro dos aparelhos. Anualmente são lançados novos modelos de aparelhos, cada vez mais específicos para determinados perfis de usuários e mais adequados para as suas necessidades e interesses (TELECO, 2010).

A crescente demanda vinda do mercado consumidor fazem os fabricantes acreditarem que o investimento na área seja promissor, resultando na certeza de que inovações tendem a seguir o mesmo ritmo (PEREIRA et al., 2009).

A atual fase revela um momento de oportunidades para os desenvolvedores de aplicativos móveis. Telefones celulares nunca foram tão populares, e *smartphones* poderosos são uma escolha para os consumidores regulares. Telefones elegantes e versáteis com recursos de *hardware*, como GPS, acelerômetros, e telas sensíveis ao toque fazem parte de uma plataforma sobre a qual são criadas aplicações móveis inovadoras (MEIER, 2010).

O mercado corporativo nacional também cresce, e conseqüentemente a concorrência mais acirrada entre fabricantes leva a maiores investimentos em qualidade, para que haja diferencial competitivo. Diversas empresas buscam incorporar aplicações móveis, visando eficiência em suas funcionalidades, em busca de agilidade em negócios empresariais e comunicações com sistemas internos a fim de obter

maiores lucros. Por outro lado, encontramos usuários que buscam uma interface visual mais elegante, comum à usabilidade cada vez maior e com uma infinidade de recursos (LECHETA, 2009).

**Tabela 1 - Uso do celular corporativo**

Quantidade de funcionários	% de empresas que utilizam celular corporativo*	
	2009	2010
10 a 49	61%	61%
50 a 249	80%	82%
250 ou mais	90%	93%
<b>Total</b>	<b>65%</b>	<b>65%</b>

Fonte: Teleco

**Tabela 2 - Tipo de dispositivo móvel mais utilizado**

Dispositivo	2007
<b>Notebook</b>	66,3%
<b>Celular</b>	50,0%
<b>Smartphone</b>	49,0%
<b>PDA</b>	43,3%
<b>Tablet PC</b>	1,9%
<b>Outros</b>	3,8%

Fonte: TELECO

Com o objetivo de suprir as necessidades funcionais do crescente número de usuários da mobilidade, foi desenvolvido um sistema operacional denominado Android™ pela empresa Google. Atualmente sustentado por uma união de mais de 70 empresas, este grupo é conhecido pelo nome de Open Handset Alliance (OHA), cuja criação foi intencionada na padronização do sistema para celulares, visando atender às

demandas e expectativas do mercado (LECHETA, 2009) e (OPEN HANDSET ALLIANCE, 2007).

O Android™ foi construído, desde a sua base, para permitir que os desenvolvedores criem aplicativos atraentes e funcionais com a finalidade de explorar ao máximo de recursos que cada aparelho possui. O Android™ é um sistema de código aberto e livre código sob a licença Apache, tanto em seu aspecto financeiro quanto no aspecto funcional (LECHETA, 2009).

## 1.2 JUSTIFICATIVA

Esse projeto se justifica pela importância de demonstrar o uso da plataforma Android no ambiente corporativo e a importância na demonstração no estudo de caso envolvendo as funcionalidades disponíveis dentro da plataforma Android se integrando com outros sistemas e aplicativos. A integração entre sistemas são hoje pontos chave quando se falam em sistemas corporativos, pois representam integridade de dados, comunicação e mobilidade.



## 2. OBJETIVO

### 2.1. OBJETIVO GERAL

Este trabalho visa uma pesquisa sobre o funcionamento geral da plataforma Android™ e de sua real efetividade em comunicação com outros sistemas, em especial os sistemas corporativos, dando foco à comunicação com o sistema da rede de postos de gasolina SIGHA PAF®, como estudo de caso. Neste trabalho, devem ser demonstradas as várias funcionalidades que o sistema pode oferecer, para o consumidor final e para os desenvolvedores de aplicativos da plataforma.

O presente trabalho tem o objetivo de demonstrar as usabilidades do sistema Android™ em um ambiente corporativo. Para tal demonstração, além da explanação do funcionamento do sistema, é realizado um estudo de caso em uma rede de postos de gasolina.

### 3.1. OBJETIVO ESPECÍFICO

Este estudo procurará esclarecer o funcionamento do sistema Android™ e a sua aplicação para a gerência do ambiente corporativo de uma rede de postos de gasolina. Com a definição do funcionamento do sistema e o estudo do ambiente corporativo escolhido, será definida uma estratégia para a aplicação do sistema, com a criação de um protótipo.

### 3. REFERENCIAL TEÓRICO

#### 3.1 OPEN HANDSET ALLIANCE

A *Open Handset Alliance* nasceu da união entre grandes empresas de telefonia móvel, chefiadas pelo Google, com a finalidade de continuar o projeto Android™. Criado pelo Google, o projeto é um sistema em que o objetivo inicial foi lançar várias funcionalidades de qualidade, competitividade, com baixo grau de complexidade e custos, a fim de garantir uma boa experiência ao cliente final (OPEN HANDSET ALLIANCE, 2007).

O benefício da união entre as empresas e criação de um sistema com preocupação na experiência do usuário é mútuo, para o cliente e para as empresas, são separadas dependendo de suas áreas de atuação, dependendo de cada tipo de empresa que está associada à OHA (LECHETA, 2009). A relação abaixo lista as áreas de atuação das empresas dentro da aliança, juntamente com as empresas que fazem parte da área:

As companhias operadoras de telefonia móvel são responsáveis pelo serviço para o usuário final, sendo elas as seguintes: Bouygues Telecom, China Mobile Communications Corporation, China Telecommunications Corporation, China United Network Communications, KDDI CORPORATION, NTT DOCOMO, INC.SOFTBANK MOBILE Corp., Sprint Nextel, T-Mobile, Telecom Italia, Telefónica, TELUS e Vodafone (OPEN HANDSET ALLIANCE, 2007).

As companhias fabricantes de aparelhos fazem parte da criação do *hardware*, sendo as seguintes: Acer Inc., Alcatel mobile phones, ASUSTeK Computer Inc., CCI, Dell. Foxconn International Holdings Limited, Garmin International, Inc., Haier Telecom (Qingdao) Co., Ltd., HTC Corporation, Huawei Technologies, Kyocera, Lenovo Mobile Communication Technology Ltd., LG Electronics, Inc., Motorola, Inc., NEC Corporation, Samsung Electronics, Sharp Corporation, Sony Ericsson, Toshiba Corporation e ZTE Corporation (OPEN HANDSET ALLIANCE, 2007).

As empresas fabricantes de componentes semicondutores são responsáveis pela confecção dos chips que compõem os aparelhos celulares: *AKM Semiconductor Inc, Audience, ARM, Atheros Communications, Broadcom Corporation, CSR Plc., Cypress Semiconductor Corporation, Freescale Semiconductor, Gemalto, Intel Corporation, Marvell Semiconductor, Inc., MediaTek, Inc., MIPS Technologies, Inc., NVIDIA Corporation, Qualcomm Inc., Renesas Electronics Corporation, ST-Ericsson, Synaptics, Inc, Texas Instruments Incorporated e Via Telecom* (OPEN HANDSET ALLIANCE, 2007).

As companhias de desenvolvimento de *software* são responsáveis pela criação e manutenção dos programas que devem ser executados na plataforma Android™, listadas a seguir: *ACCESS CO., LTD., Ascender Corp., Cooliris, Inc., eBay Inc., Google Inc., LivingImage LTD., Myriad, MOTOYA Co., Ltd., Nuance Communications, Inc., NXP Software, OMRON SOFTWARE Co, Ltd., PacketVideo (PV), SkyPop, SONiVOX, SVOX e VisualOn Inc* (OPEN HANDSET ALLIANCE, 2007).

Outras empresas também são responsáveis pelo marketing, divulgação e comercialização dos produtos para o usuário final, listadas a seguir: *Aplix Corporation, Borqs, L&T Infotech, Noser Engineering Inc., Sasken Communication Technologies Limited, SQLStar International Inc., TAT - The Astonishing Tribe AB, Teleca AB, Wind River e Wipro Technologies* (OPEN HANDSET ALLIANCE, 2007).

#### 4. SISTEMA ANDROID™

O Android™ é consiste em uma nova plataforma de desenvolvimento para aplicativos móveis, baseada em um sistema operacional Linux, com diversas aplicações já instaladas e, ainda, um ambiente de desenvolvimento muito poderoso e flexível. É o primeiro projeto em código aberto (*open source*) de uma plataforma móvel, e envolve um pacote com programas para celulares, *middleware*, aplicativos e interface do usuário. (LECHETA, 2009).

Devido ao fato de ser um projeto *open source*, permite que os desenvolvedores adaptem e evoluam cada vez mais estas funcionalidades (PEREIRA et al., 2009).

O Android™ SDK é o kit de desenvolvimento que disponibiliza as ferramentas e API's necessárias para desenvolver aplicações para a plataforma, utilizando a linguagem Java (PEREIRA et al., 2009).

Entretanto, no sistema operacional não existe uma máquina virtual Java (JVM). A máquina virtual do Android™ é chamada *Dalvik*, e é otimizada para execução em dispositivos móveis. Ao desenvolver aplicações, todos os seus recursos são utilizados normalmente da linguagem, porém, depois que o *bytecode* (.class) é compilado, ele é convertido para o formato .dex (Dalvik Executable), e em seguida os arquivos .dex e outros recursos como imagens são compactados em um único arquivo com extensão .apx (Android™ Package File), que representa a aplicação final, pronta para ser distribuída e instalada. (LECHETA, 2009).

A *Dalvik* é projetada para executar várias máquinas virtuais paralelamente. A máquina virtual funciona em sistemas com baixa frequência de CPU, pouca memória de RAM disponível e sistema operacional sem espaço de *Swap*. Além disso, é otimizada para consumo mínimo de memória, bateria e CPU (PEREIRA et al., 2009). Este sistema operacional é baseado na linguagem Java e roda o *Kernel 2.6* do Linux. É um sistema leve e cheio de recursos (DIMARZIO, 2008).

Embora este sistema tenha sido construído com base no Linux, não se trata especificamente de um sistema Linux, pois não possui *windowing system* nativo (componente de GUI), não suporta *glibc* e não possui alguns dos conjuntos de padrões apresentados em algumas distribuições Linux (PEREIRA et al., 2009).

Ainda assim, o Android™ possui alta capacidade de segurança, pois cada processo da aplicação é considerado uma *sandbox*, isso significa que toda vez que um



aplicativo for instalado em uma estação Android™ é criado um novo usuário Linux para aquele programa, com diretórios que serão usados pelo aplicativo, mas somente para aquele usuário Linux, deixando acessos a recursos internos do aplicativo configuráveis. Os aplicativos ficam completamente isolados uns dos outros, e qualquer tentativa de acessar informações de outro aplicativo precisa ser explicitamente autorizada pelo usuário, podendo ser negada a instalação do aplicativo, ou autorizada a instalação, porém, com controle das permissões que este aplicativo poderá ter através de um mecanismo de permissão específico de cada aplicação (PEREIRA et al., 2009).

#### 4.1. ANDROID™ SDK

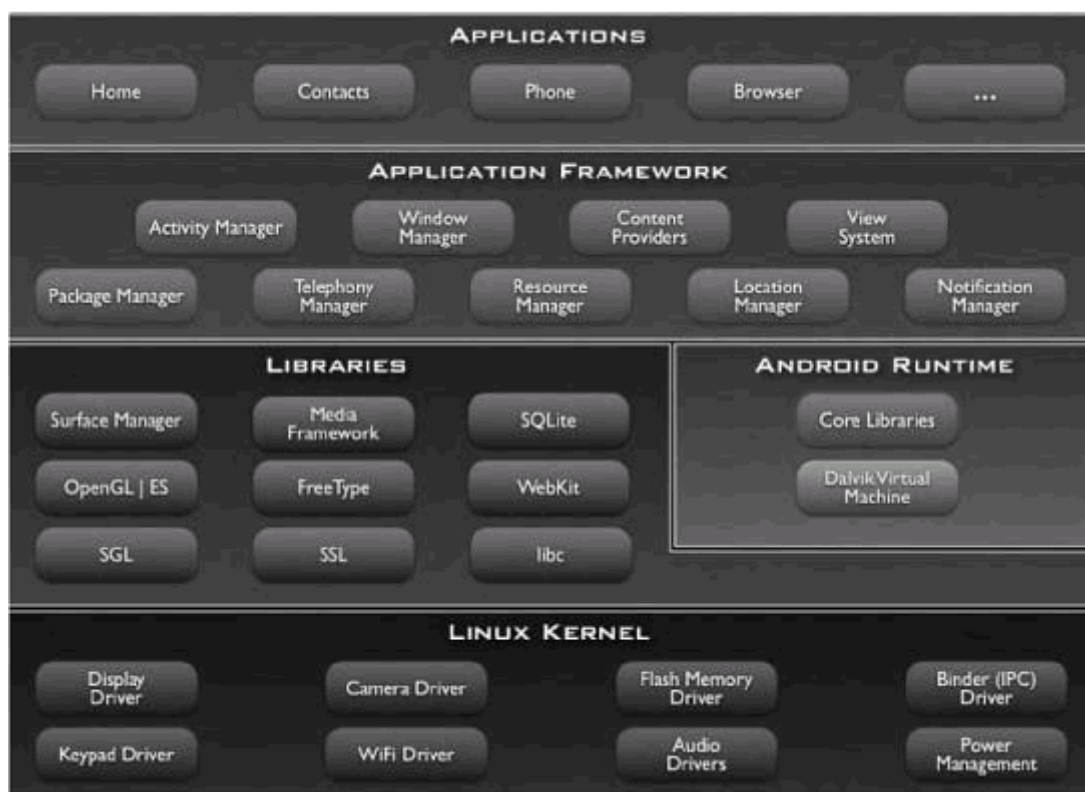
O Android SDK é o software utilizado para desenvolver aplicações no Android™, que tem um emulador para simular o celular, ferramentas utilitárias e uma API completa para a linguagem Java, com todas as classes necessárias para desenvolver as aplicações. (LECHETA, 2009).

O download do Android™ SDK incluem:

- As API's do Android™: o núcleo do SDK são bibliotecas de API que dão acesso à pilha de *software* do Android™. Estas são as mesmas bibliotecas utilizadas no Google para criar aplicações nativas do Android™.
- Ferramentas de desenvolvimento: É possível transformar o código fonte do Android™ em aplicações executáveis para o mesmo.
- Gerenciador de dispositivos virtuais Android™ e emulador: O emulador Android™ é totalmente interativo e apresenta várias alternativas *skins*. O emulador é executado dentro do dispositivo virtual do Android™ que simula o dispositivo de configuração do *hardware*. Usando o emulador é possível determinar como o aplicativo irá se comportar no real dispositivo do Android™.
- Documentação completa: O SDK inclui um extenso nível de código com informações de referência detalhando exatamente o que está incluído em cada pacote e classe, além de como usá-los.

- Código de exemplo: O SDK possui uma seleção de aplicativos de exemplo para demonstrar algumas das possibilidades do Android™, bem como programas simples que mostram como usar os recursos de cada API. (MEIER, 2010).

Na figura 1 é possível visualizar como é estruturada a arquitetura da informação dentro do sistema Android™.



**Figura 1 - Arquitetura Android™**

**Fonte: Pereira et al.**

No topo de todas as camadas está a de aplicativos, onde se encontram os escritos em Java, cliente de e-mail, mapas, navegadores, calendários, programas de SMS, gerenciador de contatos, agendas, entre outros que serão desenvolvidos pela comunidade. Isso significa que para desenvolver programas para a plataforma Android™ é necessário criar os aplicativos em Java para serem executados na máquina virtual Dalvik (PEREIRA et al., 2009).

Na camada do framework encontram-se todas as API's e os recursos utilizados pelos aplicativos. Dentro dessa camada estão:

- *Activity Manager*, que gerencia o ciclo de vida de todas as *activities*, possibilitando o deslocamento de uma *activity* para outra, e assim por diante;
- *Package Manager*, que é utilizado pelo *activity manager* para ler as informações dos APK's (Pacotes de arquivos do Android™). O *package manager* se comunica com o resto do sistema e diz quais os pacotes estão sendo utilizados no dispositivo e quais são as capacidades destes pacotes;
- *Window Manager*, que é responsável por gerenciar as apresentações de janelas;
- *Content Providers*, que possibilita o compartilhamento de dados entre os aparelhos, possibilitando a troca de informações entre os aplicativos,
- *View System*, que disponibiliza todo o tratamento gráfico para a aplicação, como botões, *layouts* e *frames* (PEREIRA et al., 2009).

Na camada framework ainda são encontrados outros elementos, como *Location Service*, *Bluetooth Service*, *Wi-Fi Service*, *USB Service* e *Sensor Service*. Cada elemento citado possui pleno acesso às API's e são utilizadas pelo núcleo da plataforma (PEREIRA et al., 2009).

O Android™ possui bibliotecas desenvolvidas em C/C++, que poder ser acessadas por frameworks disponibilizados para os desenvolvedores. Entre as principais bibliotecas estão:

- *Freetype* para renderização de fontes e *bitmaps*
- *System C library* é derivado da biblioteca C padrão do BSD, sintonizada para dispositivos rodando Linux,
- *WebKit* é baseado no código aberto do *browser webkit*, que é um renderizador de páginas para navegadores, com suporte para CSS, *javascript*, DOOM e AJAX,
- *SQLite* um banco de dados relacional, implementada em C, com suporte à base de dados acima de 2 *terabytes*. Implementa a maioria dos SQL92,
- *SGL* responsável pelos gráficos 2D subjacentes,
- *Surface Manager* fornece o acesso aos subsistemas de exibição, como as camadas de aplicações 2D e 3D,
- *Media Libraries* baseado no *Packet Videos's OpenCORE*. As bibliotecas suportam os mais populares formatos de áudio e vídeo, e também *hardware* e *software* de *plugins* de codec,

- *LibWebCore* um web browser *engine* utilizado tanto para Android™ quanto para exibições web,
- 3D *libraries* uma implementação baseada nas API's do *OpenGL ES 1.0's*; as bibliotecas utilizam aceleração 3D via *hardware* ou *software* de renderização 3D altamente otimizado incluído no Android™ (PEREIRA et al., 2009).

A perspectiva comercial Android™ não exige certificação para se tornar um desenvolvedor. O mercado Android™ prevê distribuição e monetização de seus aplicativos, não possuindo processo de aprovação para a distribuição de aplicativos e dando total controle sobre sua marca e a tela inicial do usuário (MEIER, 2010).

Toda estrutura de aplicativo possui alguns componentes chave que os desenvolvedores precisam entender antes que eles possam começar a escrever aplicações baseados no framework Android. Por exemplo, do mesmo jeito que é necessário entender *JavaServerPages (JSP)* e *servlets* para desenvolvimento *Web Java*, é necessário entender sobre as *activities*, *views*, *intents*, *content providers*, *services* e sobre o arquivo *AndroidManifest.xml* para que se comece a se desenvolver aplicações para o Android. Abaixo serão mencionados alguns tipos fundamentais dos componentes citados (KOMATINENI et al., 2011).

## 4.2 ANDROID™ SDK

De todos os componentes do sistema Android™, a *Activity* é a mais utilizada. Geralmente este componente é representado por uma tela de aplicação (PEREIRA et al., 2009), responsável por controlar os eventos da tela e definir qual visualização será responsável por desenhar a interface gráfica do usuário (LECHETA, 2009).

Uma *Activity* normalmente representa uma única tela em sua aplicação. A *Activity* é muito mais do que pode também ser a visualização de dados, criação de dados, ou edição de dados. A maioria dos aplicativos Android™ têm diversas *activities* dentro delas. (KOMATINENI et al., 2011)

Há métodos da classe *activity* que podem ser utilizados para controlar o estado da aplicação, são também conhecidos como *Eventos*.



É necessário a implementação do evento *onCreate*, método da *activity* pelo qual é usado para inicialização necessária para executar a aplicação (LECHETA, 2009). Os eventos compõem o ciclo de vida de uma *activity*, isto é, os possíveis estados em que ela se encontra são listados abaixo:

- Executando,
- Temporariamente interrompida em segundo plano
- Completamente destruída (LECHETA, 2009).

O ciclo de vida da *Activity* está esquematizado no desenho abaixo.

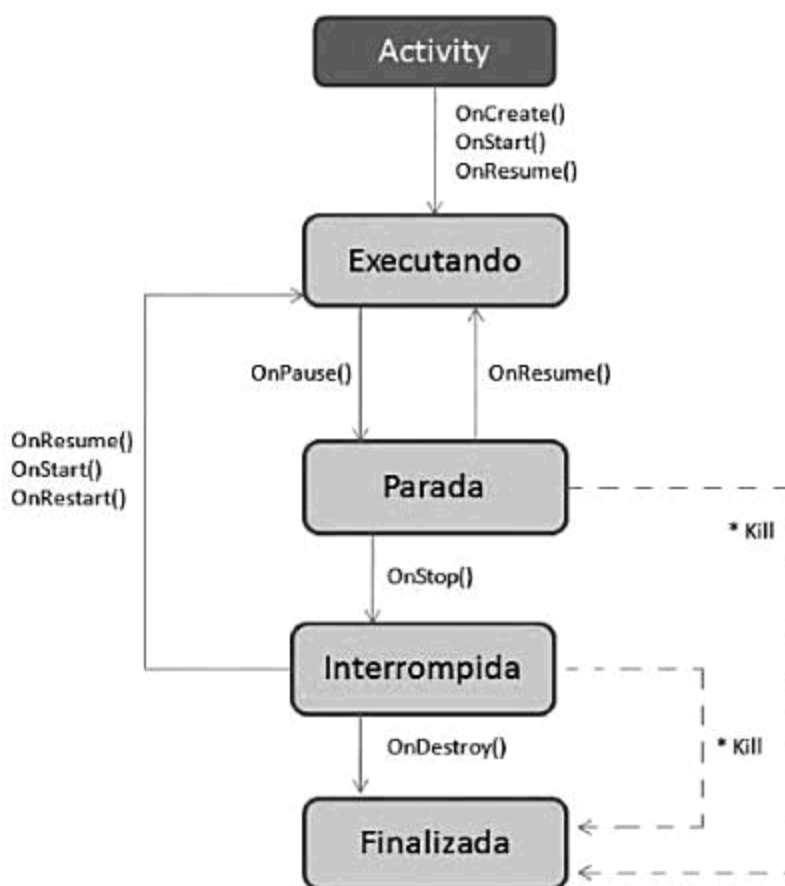


Figura 2 - Ciclo de vida Android™

Fonte: Pereira et al.

- *onCreate*: Método chamado quando a atividade é inicialmente criada,
- *onStart*: Chamado quando a atividade torna-se visível para o usuário,
- *onResume*: É o topo da pilha de atividade, chamado quando vai iniciar a interação com o usuário,

- *onRestart*: Chamado quando sua atividade estiver interrompida e prestes a ser acionada pelo usuário,
- *onPause*: Chamado quando o sistema está perto de começar uma próxima atividade. Geralmente é utilizado para gravar as informações que ainda não foram salvas,
- *onStop*: É chamado quando a atividade não estiver mais sendo utilizada pelo usuário e perdeu o foco para outra atividade,
- *onDestroy*: Pode ser chamado quando a atividade terminou, ou quando o sistema precisa finalizar a atividade para a liberação de recursos,
- *onFreeze*: Possibilita salvar o estado atual da atividade. É quando a atividade está prestes a ser suspensa (PEREIRA et al., 2009).

### 4.3 SERVICE

O componente *Service* é utilizado dentro do sistema Android™ para executar um processamento em segundo plano, geralmente vinculado a algum processo, por tempo indeterminado até que tal serviço tenha recebido uma nova ordem. Ele faz um alto consumo de recursos, memória e CPU, não precisa interagir com o usuário e, conseqüentemente, não havendo necessidade de interface gráfica (LECHETA, 2009).

Cada classe *service* deve ter uma declaração `<service>` correspondente no arquivo `AndroidManifest.xml` seu pacote. Os serviços podem ser iniciados com `Context.startService()` e `Context.bindService()`. (Android Developers, 2012).

A classe *service* pode ser utilizada para criar um player MP3, fazer download de alguma informação da internet e outros processamentos demorados sem que o usuário perceba. O serviço é sempre executado em segundo plano, e o próprio sistema operacional cuida de seu processo e do gerenciamento de memória. (LECHETA, 2009)

### 4.4 INTENT

*Activities* e *services* são ativados por mensagens assíncronas, denominadas de *Intents*, diferentemente dos *Content Providers*, que são ativados através de pedidos de

um *Content Resolver* (PEREIRA et al., 2009). As *Intents* representam um importante papel na arquitetura *Android*<sup>TM</sup> para integrar diferentes aplicações (LECHETA, 2009).

Embora uma *intent* seja facilmente entendida como um mecanismo para invocar um componente, o *Android*<sup>TM</sup> mostra muitas ideias sobre seu conceito. É possível a utilização do *Intent* para chamadas de aplicações externas a partir de sua aplicação, também podendo ser utilizado para chamadas de componentes internos ou externos de sua aplicação (KOMATINENI et al., 2011).

O *intent* é um objeto que detém o conteúdo de uma mensagem. Por sua vez o *intent filter* serve para descrever quais *intents* uma *activity* é capaz de tratar. Uma *intent* explícita é aquela que vai direto ao destino, sem consultar a *Intent Filter*, mas uma *intent* implícita é entregue ao destino apenas e passa pelos filtros dos componentes. Estas são registradas no arquivo *AndroidManifest.xml*, uma vez que a aplicação deve conhecer a capacidade de cada componente (PEREIRA et al., 2009).

O *AndroidManifest.xml* é essencial para toda a aplicação, é um arquivo de configuração que descreve os elementos da aplicação, as classes de cada componente a ser utilizado, qual o tipo de dado que ele pode tratar, quando pode ser ativado, ou seja, seu objetivo é definir os dados de cada elemento. As permissões também podem ser definidas dentro deste arquivo (LECHETA, 2009).

## 5. WEB SERVICE

Um *Web Service* é um aplicativo servidor que disponibiliza um ou mais serviços para seus clientes, de maneira fracamente acoplada. Geralmente, o protocolo de troca de mensagens utilizado é o SOAP (CLEUTON, 2006).

Um *Web Service* expõe sua interface para usuários usando um documento XML, conhecido como *Web Services Description Language* ou WSDL. Usando o WSDL, é possível descobrir quais são os tipos de dados, formatos de mensagens e serviços disponibilizados por um *Web Service*. O padrão WSDL define um *Web Service* como uma coleção de *endpoints* de rede mais conhecidos como *port*. Uma *port* ou *endpoint* permite algumas *operations* e cada *operation* implica na troca de algumas mensagens (ou *messages*), que são formadas por tipos de dados definidos em um esquema XML (CLEUTON, 2006).

- *Types*: definição de dados usados nas mensagens (*messages*), usando algum sistema de definição de dados, como um esquema XML,
- *Message*: a definição abstrata dos dados sendo trocados entre um *Web Service* e um *consumer*,
- *Operation*: a definição abstrata de uma ação suportada pelo *Web Service*,
- *Port type*: um conjunto abstrato de operações suportado por uma ou mais portas (ou *endpoints*),
- *Binding*: uma especificação concreta de protocolo e formato de dados para um *port type*,
- *Port*: um único *endpoint* formado pela combinação de um *binding* e um endereço de rede,
- *Service*: uma coleção de *endpoints* (ou *ports*) relacionados (CLEUTON, 2006).

## 6. FERRAMENTAS DE DESENVOLVIMENTO

### 6.1 ECLIPSE©

O Eclipse© é um sistema controlado por uma organização sem fins lucrativos independente, chamada *Eclipse Foundation*. O Eclipse© é um IDE (*Integrated Development Environment*), que pode ser utilizado para desenvolver *software* em qualquer linguagem e não apenas em Java. Este sistema funciona em vários sistemas operacionais, entre eles os mais populares, *Windows*, *Linux* e *Mac* (BURNETTE, 2006).

A ferramenta Eclipse© concedeu o suporte necessário ao desenvolvimento da aplicação StationMob. O suporte ao desenvolvimento de aplicativos na plataforma Android a partir da ferramenta Eclipse parte desde um *plugin* para a ferramenta (ADT) até um suporte ao simulador de um sistema operacional Android dentro do ambiente de desenvolvimento.

O *Android Development Tools* (ADT) é um *plug-in* para o Eclipse IDE que é projetado para dar um ambiente integrado de desenvolvimento de aplicativos Android. O ADT amplia os recursos do Eclipse para que rapidamente se possam criar novos projetos Android, criar uma interface de aplicativo, adicionar componentes baseados no Android Framework API, depurar aplicativos usando as ferramentas do Android SDK, e até mesmo exportar pacotes de instalações Android a fim de distribuir a sua aplicação (Android Developers, 2012).

### 6.2 NETBEANS©

O NetBeans© é um Ambiente de Desenvolvimento Integrado (IDE) *open-source* para desenvolvedores de software. Fornece todas as ferramentas necessárias para criar ambiente de trabalho profissional, empresarial, web e aplicações móveis com a plataforma Java, bem como com *C/C++*, *PHP*, *Javascript* e *Groovy*. (NetBeans, 2012).

O NetBeans© foi utilizado para criação do *WebService* que fica alocado no servidor. A ferramenta NetBeans© foi crucial na rápida criação do *WebService* disponível para fornecimento dos dados do servidor (informações sobre os postos de gasolina).

## 7. ESTUDO DE CASO STATIONMOB

### 7.1 MATERIAIS E MÉTODOS

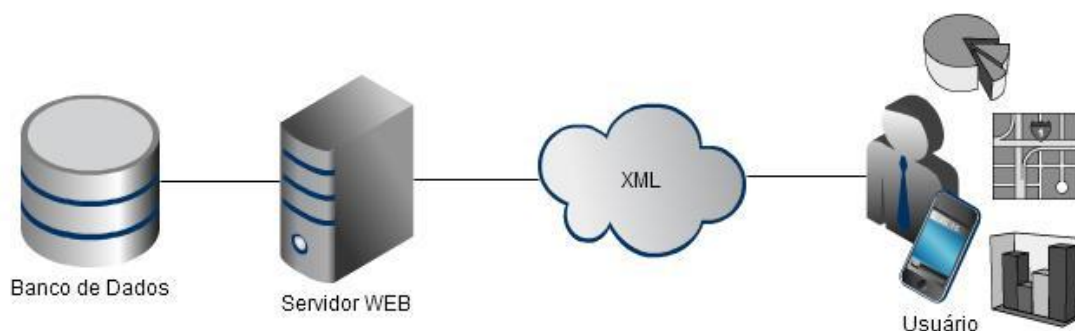
Antes de criar o sistema para Android™ foi necessário criar uma base de dados que contenham os dados referentes a uma rede de postos, tais como: franquias, localizações, produtos e vendas. Os dados e a estrutura dos dados foram alocadas em um servidor PostgreSQL 8.3.

A estrutura e os dados do banco estavam pré-definidas de acordo com o trabalho realizado pelo Sidnei Fernando de Castilho em sua monografia apresentada em 2011 (*Sistemas Assíncronos utilizando a tecnologia JEE.*), onde foi demonstrado o funcionamento de mensagens assíncronas independente de conexão remota utilizando a tecnologia JEE (Java Enterprise Edition) com estudo de caso foi desenvolvido um sistema PAF-ECF para automação de Postos de Combustível chamado de Sistema Autoflex.

E com base no Sistema Autoflex foram fornecidos dados relevantes para que um servidor de serviços Webservice fosse implementado a partir desse projeto.

Para que haja o fornecimento destes dados para o Android™ (sem que seja necessário o armazenamento local dos mesmos) foi usada a solução *Webservice* (como citado no parágrafo anterior) para integração dos dados de um servidor (sistema onde são armazenados os dados – Sistema Autoflex) com o sistema Android™ (sistema que irá transformar os dados em informações na tela do celular) desenvolvido neste trabalho como estudo de caso (StationMob).





**Figura 3 - Arquitetura StationMob**

As tentativas de criação do WebService foram inúmeras, devido ao grau de dificuldade de encapsulamento de objetos dentro de serviços web, que resultou no uso da IDE NetBeans para criação do serviço *web* através do projeto JAX-WS, rodando em um servidor aplicações Web GlassFish 3.

O desenvolvimento do WebService resultou em um serviço onde as principais premissas de retorno seriam os postos de gasolina que estavam cadastrados na base de dados e as informações destes postos, como mostra o diagrama do anexo 1 deste documento.

Para demonstrar algumas das funcionalidades do sistema operacional Android™ e representar seus componentes principais (Activities e Intents) foi desenvolvido um protótipo de sistema, denominado *StationMob*, para monitoramento móvel da rede de postos. Utilizando-se da IDE Eclipse®, foi possível integrar um ambiente para desenvolvimento para Android™, a partir da instalação do plug-in e o Android™ SDK e integração com IDE Eclipse®.

Todo passo a passo para instalação e configuração do ambiente de desenvolvimento para Android no IDE Eclipse® pode ser encontrado na página <http://developer.android.com/sdk/installing.html>. A partir deste site podemos encontrar o download do SDK, o *plugin* ADT do Eclipse®, o simulador AVD (*Android Virtual Device*) usado para integrar com o Eclipse® a fim de que seja possível emular um dispositivo android dentro do ambiente de desenvolvimento, executar a aplicação e testar a execução do sistema que será desenvolvido.

A partir do plug-in é possível instalar e rodar o emulador do Android™ dentro do Eclipse, facilitando no desenvolvimento com o debug tal como se fosse qualquer outra aplicação Java.

Ao criar o projeto StationMob foi solicitado no *Wizzard* da criação o nome do projeto (StationMob), nome do pacote raiz onde conterà a Activity principal (com.br.stationmob), nome da *activity* principal (ActStationMob) e o nome da aplicação (StationMob) como mostrado no diagrama de classes incluso no Anexo 2 deste documento.

Logo após foram estruturadas a hierarquia de pacotes dentro da raiz do projeto (br.com.stationmob) para organização do projeto na seguinte definição:

- *datasources*: Classes usadas para fornecimento de dados para amostragem no sistema.
- *objects*: Java Beans usados para abstrair os objetos.
- *source*: Implementação de métodos de regras de negócio
- *útil*: Classes que são usadas dentro do sistema a fim de padronizar e deixar transparentes algumas regras.
- A estrutura de pastas que compõem um projeto Android™ são compostas por:
- *src*: Pasta do projeto que contém as classes Java, inclusive uma classe gerada automaticamente pelo plug-in chamada R, que é usada para acesso a recursos que estão mapeados nas outras pastas listadas abaixo;
- *assets*: Contém arquivos opcionais ao projeto, como por exemplo, uma fonte customizada;
- *res*: Pasta de contém os recursos da aplicação, como imagens, *layout* de telas, e arquivos de internacionalização. Dentro da pasta *Res* são encontradas mais três subpastas: *drawable*, *layout* e *values*;
- *drawable*: Pasta com as imagens da aplicação;
- *layout*: Contém arquivos XML de layouts para construir as telas da aplicação;
- *values*: Contém os arquivos XML utilizados para a internacionalização da aplicação e outras configurações. O XML é composto de um layout simples com chave=valor.

Após a estruturação de pastas foram criadas as Activitys referentes às telas do sistema: ActPostos para listagem dos postos em modo lista e a ActMapa para listagem dos postos no modo mapa.

Na Activity principal (ActStationMob) foi criado um layout específico (main.xml) para especificação de como se comportará o menu no sistema e passado tal arquivo de layout para mostra de uma lista de *strings* usadas para o menu. Sendo assim, foi criado um *listener* para o menu no método *onListItemClick* para tratamento de seleções no menu, encaminhando cada evento para suas respectivas Activities.

Abaixo segue como ficou estruturado o projeto dentro do ambiente de desenvolvimento:

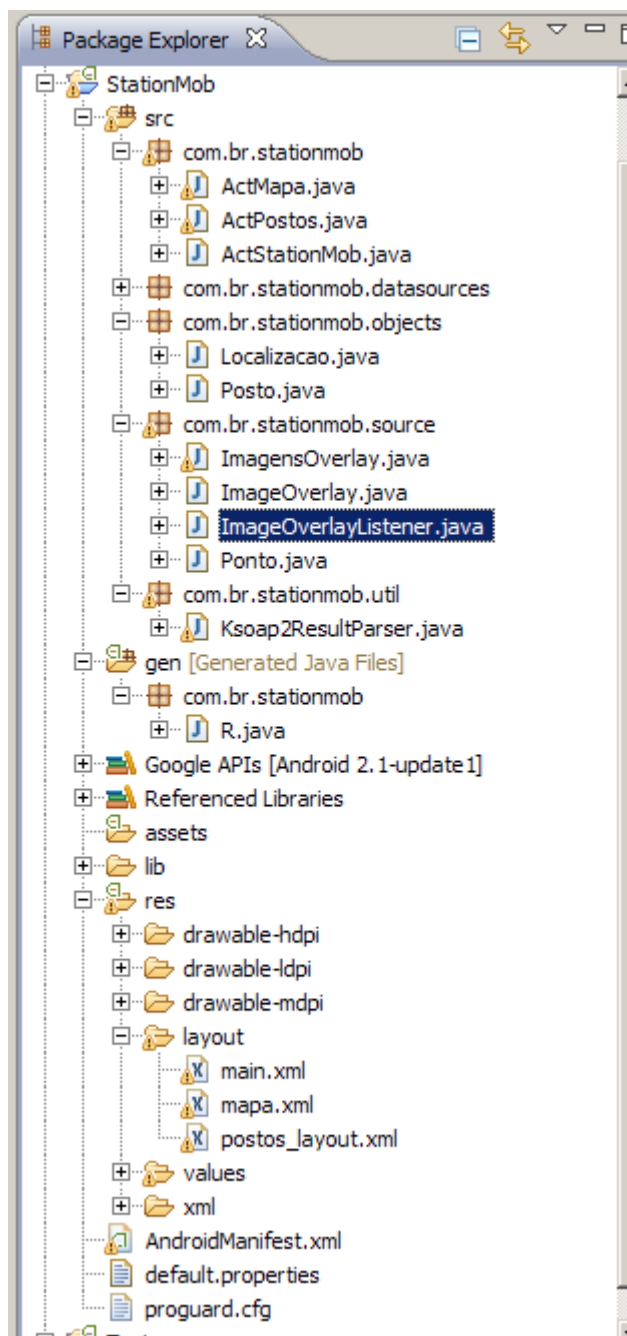


Figura 4 - Estrutura do projeto



Figura 5 - Menu principal – StationMob

Fonte própria

Sempre é necessário mapear as Activitys dentro de um arquivo dentro do projeto chamado "*AndroidManifest.xml*" passando caminhos de execução de classes e propriedades das telas configuradas.

Neste mapeamento também foram inclusas as telas que mostram os postos a partir de um Webservice.

A primeira listagem mostra uma simples lista com layout especificado em um arquivo separado. Os postos da lista sempre são recuperados a partir de um Webservice que disponibiliza informações do servidor.



Figura 6 - Lista de postos – StationMob

Fonte própria

A partir de cada item é possível desenvolver um tratamento de Click.

Também foi desenvolvida como protótipo uma tela de mapas onde são integrados os postos de acordo com suas localizações.

A partir dos postos listados no mapa é possível integrar outros sistemas como o de navegação e de procura de endereço.





Figura 7 - Mapa de postos – StationMob

Fonte própria



## 8. CONCLUSÃO

A plataforma Android™ concede uma grande gama de facilidades e de benefícios para os seus desenvolvedores. Uma das facilidades que pode ser numerada é o acesso facilitado a dados de aplicativos externos e servidores contendo dados, com o uso do tráfego de dados 3G ou através de conexão com a Internet via WiFi. Outra facilidade elencada é o acesso a informações internas que podem ser alocadas dentro do próprio aparelho celular, em base de dados ou com o direcionamento de outros aplicativos já instalados no aparelho.

A facilidade de desenvolvimento para a plataforma, assim como a independência de outras plataformas que sustentem o desenvolvimento é um fator que contribui para o crescimento de desenvolvedores para Android™.

A tendência atual da mobilidade entre os usuários, com a simplificação do acesso a aparelhos multifuncionais e com acesso à Internet e o crescimento da independência de computadores de mesa, notebooks e outros dispositivos favorece o crescimento do uso de plataformas sustentadas por dispositivos móveis. Os aparelhos multifuncionais ganham cada vez mais funcionalidades e os usuários criam maior vínculo com este tipo de produto.

Tais ferramentas tendem a criar também um vínculo maior em aplicativos corporativos, como apresentações, editores de textos, agendas, calendários, clientes de e-mails, entre outros. A variedade de aplicativos com funcionalidades corporativas cria uma maior fidelização do público consumidor, tornando cada vez mais viável a produção de aplicativos voltados para este setor.

Para a aplicação dentro de uma rede de postos de combustível, o desenvolvimento de um protótipo em sistema Android™, detalhado neste trabalho, se mostrou viável e rentável. As facilidades de customização no desenvolvimento aliadas à variedade de dispositivos que suportam o sistema facilitam o desenvolvimento e a aplicabilidade prática do uso.

## REFERÊNCIA

BURNETTE, Ed. **Eclipse IDE: Guia de bolso**. Porto alegre: Bookman, 2006. Pg 11 e 12.

CLEUTON, Sampaio. **SOA e WebServices em Java**. Rio de Janeiro : Brasport, 2006. Pg 38, 39.

KOMATINENI, Satya; MACLEAN, Dave; HASHIMI, Sayed Y. **Pro Android™ 3**. New York: Apress, 2011. Pg 125.

LECHETA, Ricardo R. **Google Android™.1. Ed**. São Paulo: Novatec, 2009. Página 19, 20, 24.

OPEN HANDSET ALLIANCE. **Alliance Members | Open Handset Alliance**. Estados Unidos, 2007. Disponível em < [http://www.openhandsetalliance.com/oha\\_members.html](http://www.openhandsetalliance.com/oha_members.html) > Acesso em 01 Fev. 2011.

PEREIRA, Lucio Camilo Oliveira; SILVA, Michel Lourenço da. **Android™ para desenvolvedores**. Rio de Janeiro: Brasport, 2009. Página 2 – 13.

TELECO. **Estatísticas de Celulares no Brasil**. Disponível em < [www.teleco.com.br](http://www.teleco.com.br) > Acesso em 31 Jan. 2011.

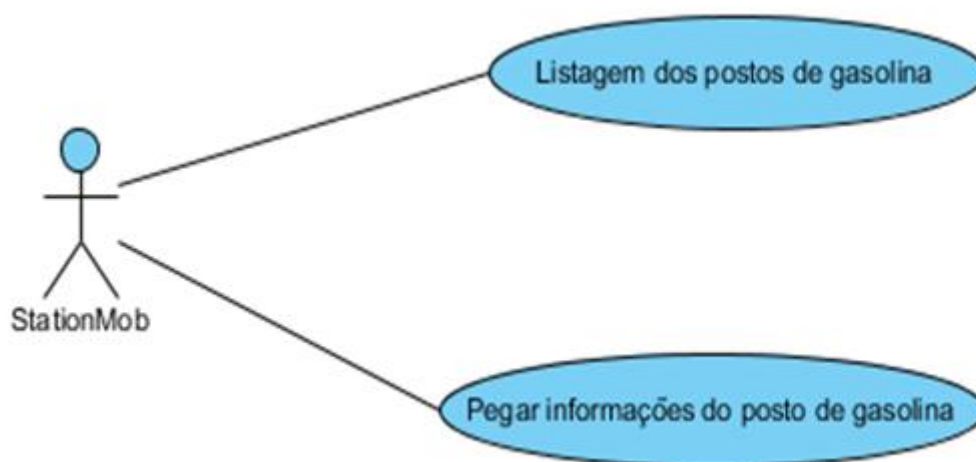
DIMARZIO, Jerome J. F.. **Android™: a programmer's guide**. Ed.Mc Graw Hill, 2008. Página 6 e 7.

MEIER, Reto. **Professional Android™ 2 Application Development**.1 ed. Indianapolis: Wiley Publishing, 2010. Página 10,12,13.

NetBeans IDE. **Features**. Disponível em <<http://netbeans.org/features/index.html>> Acesso em 01 de Mar. 2012.

Android Developers, **ADT Plugin for Eclipse**  
<<http://developer.android.com/sdk/eclipse-adt.html>> Acesso em 01 de Mar. 2012.

Android Developers, **Service | Android Developers**  
<<http://developer.android.com/reference/android/app/Service.html>> Acesso em 09 de Mar. 2012.

**ANEXO 1 - DOCUMENTAÇÃO DO WEBSERVICE****Figura 8 - Diagrama de casos de uso - WebService**

## ANEXO 2 - DOCUMENTAÇÃO DO SISTEMA STATIONMOB

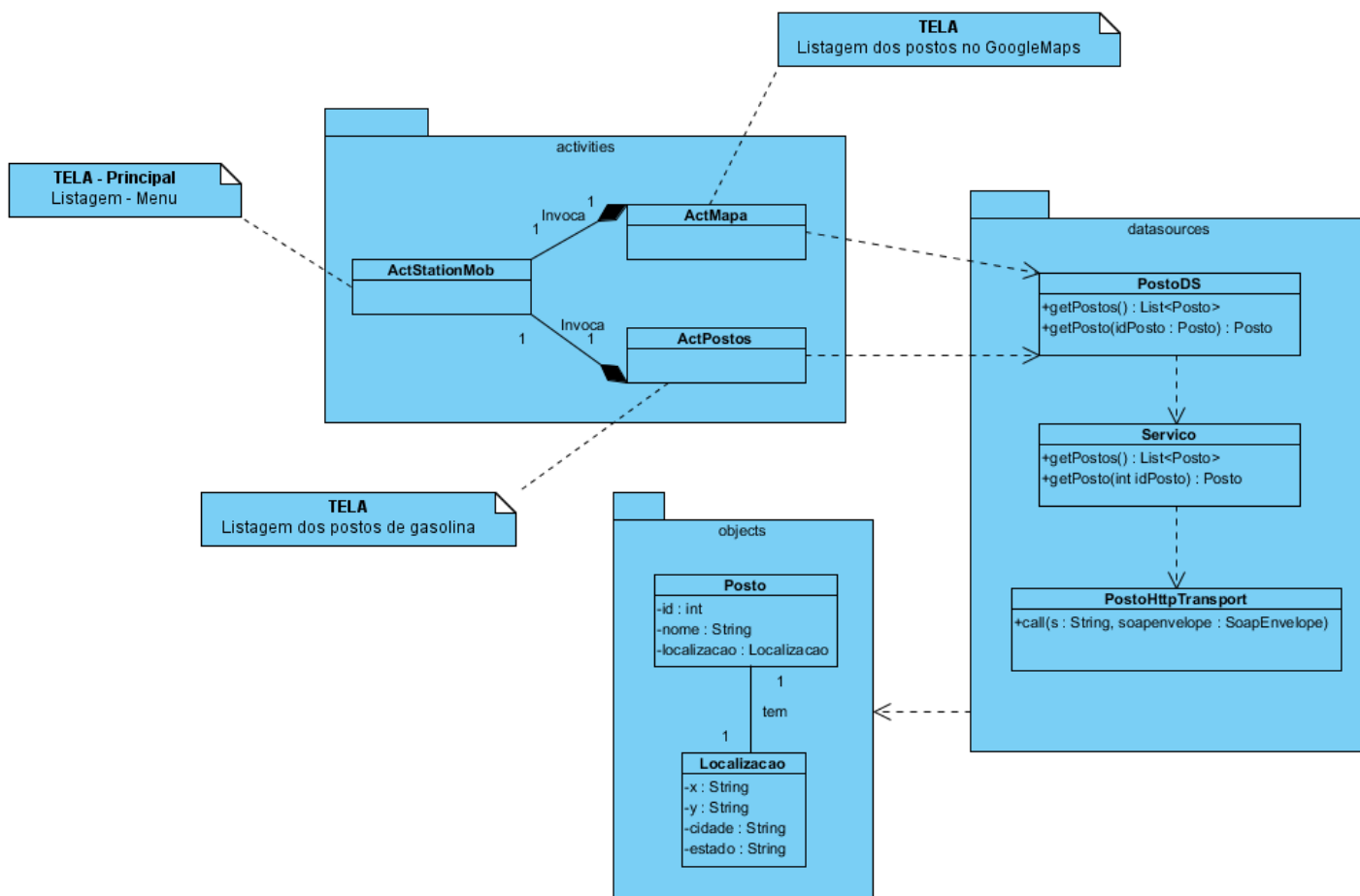


Figura 9 - Diagrama de classes - StationMob