

Sistema de Controle Baseado em Telefonia Celular

Anita Maria da Rocha Fernandes
anita.fernandes@univali.br
UNIVALI

Rafael Daniel Laurindo
UNIVALI

Resumo: O sistema foi desenvolvido com base na telefonia celular é destinado à área de distribuição e entrega do setor logístico das empresas. Considerando a dificuldade da empresa em controlar suas entregas, tendo como trajeto a saída da mercadoria do Centro de Distribuição até a chegada em seus clientes, foram organizadas as informações de entrega e coleta para futuras análises. Considerando a popularização da telefonia celular, ganhando cada vez mais espaço no mercado corporativo, foi desenvolvido um sistema interligando o telefone móvel as atividades de distribuição da empresa, auxiliando na melhoria do controle de entrega, atingindo o objetivo principal deste trabalho. Este sistema pode ser incorporado a outras empresas, as quais realizem entregas de mercadorias (agente de entrega), fornecendo informações de distribuição e remessas para entrega no propósito de auxiliar a cumprir com mais eficiência sua atividade. Uma de suas características principais é proporcionar a empresa que utiliza o programa, o recebimento de informações de sua atividade interagindo com sua frota no exato momento em que os agentes forem entregando as mercadorias para os clientes, não tendo que esperar até que o agente de entregas retorne a empresa a fim de obter as informações da entrega, como acontece atualmente. Essa informação pode ser usada para ajudar a melhorar o serviço de distribuição e entrega ou mesmo para informar os clientes quanto à localização de suas mercadorias, aumentando assim a confiabilidade da empresa em relação às entregas. Com o crescimento e a facilidade de utilização das tecnologias oferecidas pelo Google, o sistema utilizou o sistema operacional Android para celulares, integrando com o servidor de aplicação Google App Engine e utilizando o kit de interfaces GWT (Google Web Toolkit).

Palavras Chave: Sistema informação - computação movel - sistema de entrega - Android -

1. INTRODUÇÃO

A popularização dos dispositivos de comunicações móveis no início deste século, em destaque o telefone celular, que cada vez mais vem ganhando espaço no mercado de comunicação por ser um dispositivo portátil e pessoal, tem despertado o interesse de muitas empresas em investir nessa tecnologia, com o intuito de beneficiar seus processos e serviços. Com o passar do tempo, o telefone celular deixou de ser apenas um simples dispositivo de originar e receber chamadas. Ele começou a incorporar funções que antes eram destinadas apenas a computadores, como, por exemplo, navegar na internet. Hoje os celulares são capazes de executar programas, possibilitando a sua utilização nas mais variadas aplicações empresariais. As empresas de transporte e logística de mercadorias tentam cada vez mais aperfeiçoar suas técnicas de armazenagem para que tenham maior agilidade nas entregas e nos custos relacionados à distribuição de mercadorias. Com isso, tende-se a automatizar cada vez mais esses processos para se obter um melhor desempenho nos resultados da empresa, reduzindo custos e riscos provenientes da má gestão da distribuição. É importante salientar que toda automatização da distribuição realizada pelo setor Logístico vêm ao encontro das necessidades do consumidor final, como salienta Novaes (2001).

Com as tecnologias de informação que vêm inovando o mercado, o trabalho de distribuição de mercadorias e afins vem se beneficiando com os novos recursos disponíveis para a otimização dessas funções. Sem essas tecnologias, ficaria complicado controlar a demanda das transportadoras com maior fluxo de entregas. Por ser uma tarefa complicada, ela requer muito tempo para que seja feito um planejamento adequado. Com a informática, esse tempo é reduzido, pois os sistemas logísticos fazem todo o processamento pesado. Unindo-se a tecnologia de informação ao sistema de distribuição das empresas especializadas em entregas e às novas tecnologias de desenvolvimento para dispositivos móveis, fica clara a oportunidade de desenvolvimento de um sistema de telefonia celular capaz de ajudar a controlar melhor a entrega de mercadorias. Com um sistema de controle de entregas para dispositivos móveis, a empresa encarregada pela distribuição e pela entrega de mercadorias pode ter informações precisas sobre a entrega em tempo real de todos os seus entregadores. Na maioria das transportadoras brasileiras existe uma grande dificuldade na organização das rotinas de armazenagem, distribuição e entrega das mercadorias transportadas. Integrar essas rotinas é uma atividade crucial para que o sistema logístico tenha um melhor funcionamento. Vale ressaltar que está sendo abordada a singularidade da “entrega” dentro de todo processo de “distribuição” de uma empresa.

Através de pesquisas bibliográficas, concebe-se que, logística é o processo de planejamento, implementação e controle do fluxo e do armazenamento eficiente e econômico de matérias-primas, materiais semi-acabados e produtos acabados, bem como das informações a eles relativas, desde o ponto de origem até o ponto de consumo, com o propósito de atender às exigências dos clientes (LAMBERT & STOCK, 1993). A armazenagem das encomendas é um ponto crítico neste processo, pois quando se tem um estoque muito alto de mercadorias dentro de seus armazéns, a transportadora terá altos custos com relação a isso. É por isso que a logística deve ser bem feita para que se tenha o menor armazenamento de encomendas possível para aperfeiçoar a utilização dos recursos da empresa. A distribuição e a entrega são os processos mais importantes para a transportadora, pois se não houver uma boa distribuição das encomendas, a empresa terá gastos muito altos com a entrega das mercadorias. É por essa razão que a distribuição e a entrega de suas encomendas devem ser muito bem planejadas, organizadas e sincronizadas com a chegada das encomendas no armazém. Segundo Nazário (1999), o impacto do transporte no serviço ao cliente é um dos mais significativos e uma das principais exigências do mercado. Geralmente esses serviços estão ligados à pontualidade (além do próprio tempo de viagem), à capacidade de prover um serviço porta a porta, à

flexibilidade, no que diz respeito ao manuseio de uma grande quantidade de produtos, a avarias e à capacidade de o transportador oferecer mais do que um serviço básico de transporte, tornando capaz de executar outras funções logísticas. Atualmente nas empresas de logística, o processo de entrega é um pouco deficiente de dados no que se refere às entregas realizadas. Geralmente as informações só chegam à empresa quando um agente de entregas retorna para a transportadora após cumprir o seu roteiro de entregas. Sendo assim, nunca se sabe onde o agente está, nem quais encomendas já foram entregues e quais não foram aceitas ou não foi encontrado o endereço. Antes da instalação e utilização do sistema na empresa, não se tinha o controle imediato das entregas. Com a implantação é possível saber com precisão a hora em que a mercadoria chegou ao cliente. Com o crescimento das tecnologias móveis, a comunicação interna da empresa fica mais rica em recursos, como, por exemplo, com um celular é possível se ter um sistema que interaja com praticamente toda a empresa. Assim, visando essas características e necessidades desta realidade de uma empresa que necessite de uma boa logística para suas tarefas, é importante a criação e a implantação de um sistema que integre essas áreas da logística e que possa auxiliar com informações operacionais e estratégicas a empresa, bem como auxiliar a coordenar melhor o processo logístico dessa organização.

Esse artigo apresenta um sistema que possibilita a melhoria na relação Empresa x Cliente, pois trata de um programa de acesso fácil e viável a todos, processando dados e transmitindo-os ao agente de entregas em tempo real, monitorando todo o funcionamento da logística dentro e fora da empresa. A rápida proliferação dos sistemas de processamento de dados capacita a empresa de distribuição ou de logística a empregar e controlar informações de tal forma que os métodos tradicionais de serviço ao cliente e oferta de produtos deverão mudar (LAMBERT & STOCK, 1993). Um sistema de informação logística deve ser capaz de movimentar a informação desde o ponto em que é obtida até o ponto em que é necessária, ou seja, deve transferir dados, armazená-los e transformá-los em informação. Existem três características básicas em um sistema de informação logística: o elemento de entrada, o elemento de processamento da base de dados e o elemento de saída (BALLOU, 1993). O sistema desenvolvido visa auxiliar a empresa com status das entregas, e o agente de entrega com informações de remessas e roteiro que ele deverá cumprir ao longo de um período, permitindo que esse agente se comunique com a empresa no propósito de trocar dados específicos das entregas realizadas ou em andamento. Com essa solução a organização ficará com um maior suporte de informações das entregas, não tendo que ficar mais esperando até o final do dia para que o agente passe as informações das entregas realizadas. A Figura 1 ilustra a solução.

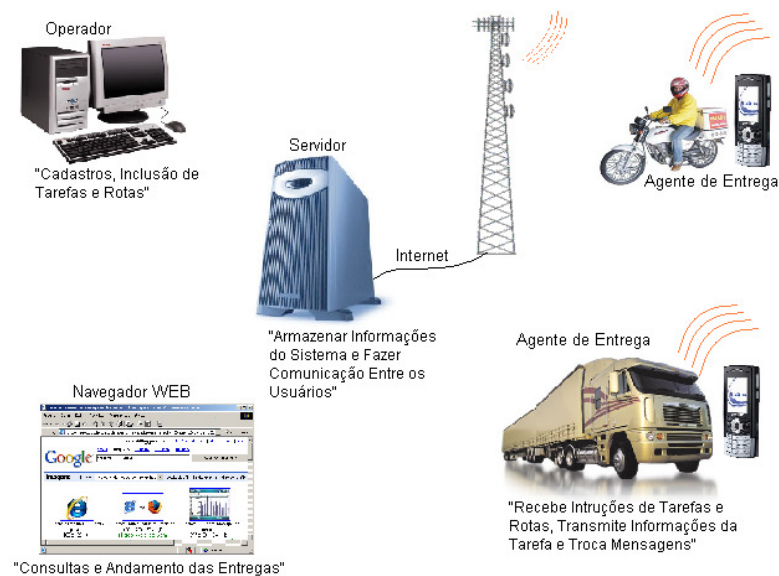


Figura 1. Esquematização da solução.

2. COMPUTAÇÃO MÓVEL

A computação móvel é caracterizada por um dispositivo móvel, com capacidade de processamento, em um ambiente sem fio. Ela vem surgindo como uma nova proposta de paradigma computacional advinda da tecnologia de rede sem fio e dos sistemas distribuídos. Nela o usuário, portando dispositivos móveis, como *palmtops*, telefones celulares ou *notebooks*, tem acesso a uma infra-estrutura compartilhada independentemente da sua localização física. Isto fornece uma comunicação flexível entre as pessoas e um acesso contínuo aos serviços de rede. As pessoas iniciaram uma busca insaciável pela mobilidade, impulsionando o crescimento de infra-estrutura móvel e a velocidade das redes móveis. Cada vez mais a tecnologia móvel vem mudando a vida do indivíduo; procurar informações, acessar a internet, verificar a sua conta no banco até se divertir já é possível com nessa tecnologia. E não pára por aí, a multimídia já é uma realidade nesses dispositivos, pode-se tirar fotos, gravar vídeos e ouvir músicas; isto tudo de forma cada vez mais veloz (ROMÁN; MENSONES & MARINAS, 2007). “Móvel subentende portabilidade; um aparelho como o PalmPilot, que viaja com o usuário. Móvel também é comumente usado para indicar que o aparelho tem uma conexão ‘sempre ligada’ com a Internet” (KALAKOTA, 2002).

O avanço tecnológico dos equipamentos portáteis e a inserção da possibilidade de comunicação sem fio permitem aos usuários móveis fazer uso de aplicações mais avançadas que as aplicações pessoais já utilizadas anteriormente (JING & HUFF, 1998). No entanto, a computação móvel, por meio da natureza de suas propriedades – portabilidade, mobilidade e conectividade, introduz restrições aos sistemas e às aplicações (AUGUSTIN, 2000). Apesar da evolução natural da tecnologia, a maioria dos pesquisadores considera que essas limitações permanecerão principalmente se comparadas ao ambiente de rede fixa (IMIELINSKI; VISWANATHAN & BADRINATH, 1997).

2.1. TELEFONIA CELULAR

No início dos anos 80, foram instaladas as primeiras redes de telefonia móvel na Europa. Naquela época o aparelho celular era um objeto de luxo tão diferente que “parecia ter saído de um filme de ficção científica”. Era utilizado somente por grandes empresários da elite européia. O telefone celular parecia algo muito distante do cotidiano humano. Passados alguns anos, esse dispositivo tornou-se um equipamento quase que indispensável na vida das

peessoas, criando um novo estilo de vida, “a vida móvel” (ROMÁN; MENSONES & MARINAS, 2007). Em 13 de outubro de 2008 fez 25 anos que o presidente da companhia americana Ameritech Mobile Communications hoje conhecida pela o nome de AT&T, fez a primeira ligação usando um telefone celular, a partir de um Motorola DynaTec 8000X para o neto de Alexandre Graham Bell, inventor da telefonia. Este aparelho pesava em torno de um quilo e tinha cerca de 30 centímetros de altura. No entanto, o maior problema de carregar um "tijolo" como este, não tinha a ver com peso ou volume, mas com os custos. O estreado DynaTAC custava US\$ 4 mil e, além disso, o portador de um aparelho tão exclusivo pagava US\$ 50 mensais pelo serviço, mais uma taxa de US\$ 0,24 a US\$ 0,40 por minuto em ligação (TERRA, 2007).

No âmbito computacional de tecnologia móvel hoje em dia, o telefone celular vem ganhando destaque pelo seu forte crescimento. Esses dispositivos inicialmente surgiram para a conversação de voz exclusivamente, mas, com o avanço da tecnologia e a evolução das gerações da telefonia celular, foram também tendo a capacidade de processamento e comunicação por meio da integração da rede de celular com a rede de dados, em especial a internet (FIGUEIREDO & NAKAMURA, 2003). Segundo Duarte (2005), a rede de telefonia móvel tem um grande potencial de exploração, seja para operadoras seja para empresas de desenvolvimento de software, oferecendo aplicações voltadas para dispositivos móveis. Com mais processamento e acesso à rede de dados, o telefone celular ganhou novas funcionalidades, como a capacidade de executar aplicações. Diversas empresas têm investido em desenvolvimento de plataformas para celulares. Uma delas é a Sun Microsystems, que desenvolveu a plataforma Java ME, a qual possibilita desenvolver aplicações em alto nível para dispositivos móveis; a Microsoft também investiu nessa área criando o Windows Mobile; e outras empresas também investiram nesse setor, tornando-se essa uma tecnologia promissora para o futuro computacional (FIGUEIREDO & NAKAMURA, 2003). Sendo assim, diversos sistemas móveis vêm surgindo nesse mercado, possibilitando inúmeras aplicações.

2.2. TECNOLOGIAS EXISTENTES

No início dos anos 80, foram instaladas as primeiras redes de telefonia móvel na Europa. Naquela época o aparelho celular era um objeto de luxo tão diferente que “parecia ter saído de um filme de ficção científica”. Era utilizado somente por grandes empresários da elite européia. O telefone celular parecia algo muito distante do cotidiano humano. Passados alguns anos, esse dispositivo tornou-se um equipamento quase que indispensável na vida das pessoas, criando um novo estilo de vida, “a vida móvel” (ROMÁN; MENSONES & MARINAS, 2007). Atualmente existem várias opções para o desenvolvimento de aplicações que são executadas em dispositivos móveis. Algumas dessas estão apresentadas a seguir. Novas linguagens vêm emergindo no mercado móvel, uma delas é a linguagem de programação Lua, lançada em 2006 e utilizada em vários projetos. Ela permite criar aplicações complexas e práticas para serem executadas em um sistema operacional Palm OS 4 e 5, e seus arquivos ocupam pouca memória do dispositivo (WEBMOBILE, 2008). A Microsoft também criou um sistema para atender às necessidades da computação móvel, o sistema chamado Windows Mobile da Microsoft, que, a grosso modo, poderia ser considerado o Windows para equipamentos móveis. Mas, apesar da aparência semelhante ao Windows para os PCs, os sistemas são totalmente diferentes, e os equipamentos móveis só executam programas criados para eles. Ele é mais pesado e por isso restrito aos aparelhos com mais poder de processamento, mas mesmo assim é usado em um número surpreendentemente grande de aparelhos e conta com uma boa coleção de softwares (MORIMOTO, 2008).

Há como opção também, o BREW, que é uma plataforma de desenvolvimento aberta, criada pela instituição Qualcomm com o objetivo final, permitir a criação e distribuição de aplicações para telefones celulares. (WEBMOBILE, 2008). Segundo Qualcomm (2008), o BREW foi projetado para permitir o desenvolvimento de aplicações a partir da linguagem que o desenvolvedor escolher para a codificação. Ele oferece suporte nativo à linguagem de programação C/C++, porém suporta integração com browsers e aplicações em outras linguagens, como Java e linguagens de marcação XML. O BREW permite que se desenvolva aplicações independentemente de que plataforma será executada. (WEBMOBILE, 2008). Nessas novas características de tecnologias móveis, criou-se o sistema operacional Android, que tem por objetivo padronizar os sistemas operacionais de telefones celulares, facilitando o desenvolvimento e a difusão de aplicativos para celulares. A plataforma é o resultado de uma parceria do Google com a Open Handset Alliance (OHA), entidade internacional formada por empresas que apóiam soluções de código fonte aberto, como Intel, Motorola, Qualcomm e Telefônica. Ela conta com o apoio de mais de 30 companhias, incluindo operadoras e fabricantes de aparelhos, entre elas China Mobile, eBay, HTC, LG, NTT DoCoMo, Samsung e Telecom Itália (O GLOBO, 2008).

Outro sistema operacional que por sua vez é muito difundido entre os telefones celulares, é o Symbian OS, que utiliza uma plataforma aberta, suporta aplicações em C++ e Java, facilitado pelo uso de Software Development Kits (SDK). Existem também bibliotecas para a gestão de dados, gráficos e textos. Pode-se também gerenciar banco de dados ClipBoard e acesso à impressora. Permite criar aplicações adicionais em agendas, gerenciador de contatos, planilhas, ferramentas de ajuda etc. (WEBMOBILE, 2008).

A empresa que vem se destacando nesse setor é a Sun Microsystems, que desde criação do Java ME (Micro Edition) vem ganhando cada vez mais espaço no mercado móvel, pois a vantagem de se desenvolver um aplicativo com a tecnologia Java é a de que as aplicações podem ser escritas apenas uma vez e rodar em uma variedade de terminais móveis, dentre eles celulares, PDAs (Personal Digital Assistant), não levando em consideração as diferenças entre seus sistemas operacionais e hardware (IMASTERS, 2007). Segundo Alçada (2004), o Bluetooth é um sistema de comunicação sem fios de curta distância com o alcance de aproximadamente 10 metros. Ele opera por padrão na faixa ISM (Industrial, Scientific and Medical) de 2,4 GHz e tem como princípio propor uma tecnologia de baixo custo para a comunicação sem fio. Para Sacks (2003), *Bluetooth* é uma especificação feita pelas indústrias de computação e de telecomunicações para descrever como diversos dispositivos móveis e/ ou fixos podem se interconectar facilmente através de uma conexão sem fio de curto alcance. O padrão de implementação do hardware é um único microchip de 9x9 milímetros. A seguir é apresentado o sistema operacional Android. Esta foi a plataforma eleita para o desenvolvimento pelas suas características e funcionalidades que serão descritas na próxima seção.

3. ANDROID

Nos dias atuais, os usuários de telefonia móvel vêm buscando cada vez mais dispositivos com mais recursos tecnológicos, para suprir suas necessidades diárias, como câmeras, Bluetooth, uma boa interface visual, jogos, GPS (Global Positioning System - Sistema de Posicionamento Global), acesso a internet e e-mails e até TV Digital. (LECHETA, 2009). No mercado corporativo o cenário não é diferente. As empresas estão procurando utilizar aplicações móveis para facilitar o seu trabalho cotidiano e obter melhores resultados, não só quanto à agilidade e qualidade de seus serviços, mais visando sempre seus lucros. Dessa maneira os celulares e *smartphones* podem ocupar um importante espaço onde a mobilidade está cada vez mais presente. (LECHETA, 2009). Os usuários comuns vêm

exigindo celulares com uma infinidade de tecnologias de ponta e as empresas juntamente com os desenvolvedores estão sempre buscando uma plataforma moderna e ágil para o desenvolvimento de aplicações corporativas para dar suporte aos seus negócios. Muitos fabricantes de celular vêm disputando a liderança desse mercado. Uma delas é a empresa de internet Google que percebeu que poderia atuar também nesse nicho de mercado. O Google se juntou com diversos fabricantes de celulares e criou um grupo chamado de Open Handset Alliance, a fim de criar um sistema operacional de código aberto e livre e padronizado para celulares. (LECHETA, 2009)

Google, vendo um grande crescimento da utilização da Internet e de pesquisa em dispositivos móveis, adquiriu o Android, Inc., em 2005, para concentrar o seu desenvolvimento em uma plataforma de dispositivo móvel. A Apple introduziu o iPhone, em 2007, com algumas idéias revolucionárias, incluindo *multitouch* e um mercado aberto para aplicações. Android foi rapidamente adaptado para incluir esses recursos e oferecer distinções definidas, como mais controle para os desenvolvedores e multitarefa. Além disso, o Android incorpora os requisitos corporativos, como o tais como suporte de câmbio, o apagamento remoto das informações e rede virtual privada (VPN - Virtual Private Network), para ir atrás do mercado corporativo de que a Research In Motion tem desenvolvido e realizado tão bem com os seus modelos de Blackberry. (STEELE;TO, 2011). A OpenHandset Alliance é um grupo formado pelas empresas líderes de tecnologia, fabricantes e operadoras de celulares do mundo todo, sob o comando do Google. Hoje o grupo tem 80 empresas participantes, entre elas estão importantes nomes como a HTC, LG, Motorola, Samsung, Sony Ericsson, Toshiba, Spring Nextel, China Mobile, T-Mobile, ASUS, Intel, Garmin e muitos outros. (LECHETA, 2009). O objetivo do grupo é manter uma plataforma única e aberta para dispositivos móveis e deixar os consumidores satisfeitos com o produto final. Outro objetivo dessa aliança é criar uma plataforma moderna e flexível para desenvolver aplicações corporativas (LECHETA, 2009). Assim nasceu o Android, uma plataforma móvel de código fonte aberto, baseado em sistema operacional Linux, de código aberto e livre onde cada fabricante pode utilizar e modificar esse sistema da maneira que quiser sem ter que pagar nada a ninguém.

O sistema operacional Android tem percorrido um longo caminho desde o anúncio da Open Handset Alliance no final de 2007. A idéia de um sistema operacional de código aberto para sistemas embarcados não era nova, mas o Google de forma agressiva apoiava seu desenvolvimento, isso definitivamente ajudou a alavancar o Android para frente em apenas alguns anos. (STEELE;TO, 2011). Com Android é possível desenvolver nessa plataforma aplicações móveis para dispositivos como *smartphones*, *tablets* e outros. Ele tem uma interface visual rica e moderna, conta também com suporte a GPSe diversas aplicações já instaladas. Uma grande vantagem do Android para os desenvolvedores, é que os aplicativos são feitos em Java. (LECHETA, 2009). Com essa união todos acabam ganhando. Os usuários têm um sistema repleto de recursos e uma interface amigável. Os fabricantes deixam de ter a necessidade de criar e manter um sistema operacional e usufruem de uma plataforma-padrão repleta de aplicativos e em constante desenvolvimento. (LECHETA, 2009). O Android é a plataforma pioneira completamente livre para celulares e de código aberto. Isso representa uma grande vantagem para sua evolução e popularização, sendo que desenvolvedores do mundo todo podem contribuir para o seu desenvolvimento e ajudar a melhorar cada vez mais a plataforma. (LECHETA, 2009). O Android está revolucionando o espaço móvel. É uma plataforma de código aberto que separa o hardware do software que roda nele. Isto permite um número muito maior de dispositivos para executar as mesmas aplicações e cria um ecossistema muito rico para os desenvolvedores e consumidores. (GARGENTA, 2011). É de grande vantagem também para os fabricantes de celulares, que se beneficiam de um sistema operacional atualizado e de alta qualidade, sem ter que pagar nada por isso. Além de poder alterar o código fonte do sistema da maneira que quiser e colocar em seus produtos sem ter

que compartilhar suas modificações com ninguém. (LECHETA, 2009). Muitas operadoras de telefonia móvel em vários países, através de diferentes protocolos de comunicação, têm um ou mais celulares com Android disponível. Outros dispositivos embarcados, como *tablets*, *netbooks*, televisores, e até mesmo automóveis, também adotaram o sistema operacional Android. (STEELE;TO, 2011)

3.1. FACILIDADES

A plataforma Android disponibiliza um conjunto de ferramentas para o desenvolvimento de aplicações. Este pacote de ferramentas é chamado de SDK (Software Development Kit) Android, que contém um emulador para simular a plataforma, ferramentas utilitárias e uma API (Application Programming Interface) completa para o desenvolvimento na linguagem Java. Embora o SDK tenha um emulador que pode ser executado com um aplicativo comum, existe também um plugin do Eclipse que integra o ambiente de desenvolvimento Java com as ferramentas do Android. (LECHETA, 2009). Embora a diversidade de dispositivos e adaptação rápida Android tenha ajudado a aumentar a sua base de usuários, ele apresenta potenciais desafios para os desenvolvedores. Os aplicativos precisam suportar vários tamanhos de tela, resoluções diferentes, teclados, sensores de hardware, versões do sistema operacional, as taxas de dados sem fio e configurações de sistema. Cada um pode levar a um comportamento diferente e imprevisível, e os testes de aplicações em todos os ambientes é uma tarefa impossível. (STEELE;TO, 2011)

O SDK do Android é composto da plataforma, ferramentas, amostras de código e documentação necessários para desenvolver aplicações Android. É construído como um *addon* para o Java Development Kit e está integrado um *plugin* para o Eclipse IDE (Integrated Development Environment). (GARGENTA, 2011). O *plugin* Android para Eclipse é o ADT (Android Development Tools), que integra todas as suas funcionalidades de desenvolvimento com o IDE Eclipse. Com ele é possível executar o emulador do Android podendo executar o código passo a passo, ou disparar uma mensagem SMS, simular uma chamada, enviar e visualizar arquivos, além de outras funcionalidades tudo isso a partir do IDE Eclipse. (ANDROID, 2011). A distribuição das aplicações da plataforma Android é feita através da compilação do projeto Java em um arquivo com a extensão APK (Android Package File). Este é um arquivo compactado com os binários e recursos do projeto. O bom disso é que ao compilar o projeto Java com o *plugin* ADT no Eclipse, já é gerado o arquivo **.apk**. E ao executar o projeto este arquivo é enviado ao emulador do Android e é aberta a aplicação dentro do emulador. (LECHETA, 2009). Com o *plugin* também é possível executar a aplicação diretamente num celular conectado em uma porta USB do computador. Isso facilita e muito os testes de aplicações em aparelhos reais, garantindo assim maior compatibilidade entre a aplicação e o dispositivo móvel. (LECHETA, 2009).

3.2. ARQUITETURA ANDROID

A arquitetura do sistema operacional Android é dividida em cinco distintos componentes. Sendo eles as seguintes componentes, Aplicações, Framework de Aplicação, Bibliotecas, Ambiente de Execução e o Kernel do Linux. A seguir serão brevemente apresentados todas as camadas, os principais componentes e suas características.

Applications (Aplicações): nativamente o Android já contém diversos aplicativos como cliente de e-mail, programa de envio de mensagem SMS (Short Message Service), calendário, mapas, navegador de internet, gerenciador de contatos e outros. Todos os aplicativos desenvolvidos para Android utilizam a linguagem de programação Java. (ANDROID, 2011). Aqui ficam os aplicativos que os desenvolvedores criam. Estas aplicações são o que os usuários finais encontram no Android. Eles podem vir pré-instalados no dispositivo ou podem

ser baixado de um dos muitos aplicativos disponíveis no Market do Android. (GARGENTA, 2011)

Application Framework (Framework de Aplicação): o *Framework* de Aplicação é um ambiente rico que oferece inúmeros recursos para ajudar o desenvolvedor do aplicativos, fazer o seu trabalho. Esta é a parte do sistema Android que está mais bem documentada e mais amplamente detalhada, porque é esta camada que possibilita aos desenvolvedores criativos, criarem aplicações inovadoras para o mercado. (GARGENTA, 2011). O *Framework* de Aplicação contém diversos recursos da plataforma Android. Isso possibilita a utilização destes recursos para desenvolver novos aplicativos reaproveitando as funcionalidades presente na plataforma. A arquitetura do aplicativo é projetada para simplificar a reutilização de componentes. Sendo que, qualquer aplicação pode criar recursos próprios e disponibilizar para qualquer outra aplicação fazer uso dessas novas funcionalidades. Este mesmo mecanismo permite que componentes padrões possam ser substituídos pelo usuário. (ANDROID, 2011)

Libraries (Bibliotecas): o sistema operacional Android conta com bibliotecas nativas C/C++. Muitas delas foram trazidas da comunidade de código livre. Estas bibliotecas facilitam o desenvolvimento de aplicações e permitem o acesso a funcionalidades de hardware. Entre elas estão: Webkit (um mecanismo de renderização web rápido usado pelo Safari, Chrome e outros navegadores), SQLite (um completo banco de dados SQL(Structured Query Language), Apache Harmony (uma implementação código aberto do Java), OpenGL (bibliotecas de gráficos 3D) e OpenSSL (a camada certificado de segurança). (GARGENTA, 2011). A plataforma Android conta com um banco de dados chamado SQLite, ele permite a utilização de banco de dados nos aplicativos desenvolvidos para Android. Cada aplicativo pode criar vários bancos de dados que somente estarão visíveis para a aplicação que o criou, garantindo assim a segurança dos dados do sistema. Com SQLite é possível executar comandos SQL padrão, o que torna mais fácil o desenvolvimento de aplicativos que necessitam de armazenamento de dados. (LECHETA, 2009). Existem também outras formas de armazenamento de dados no Android, uma delas é a persistência de arquivos no sistema de arquivos do Android. Isso pode ser feito de forma simples em poucas linhas de programação. Há também um método de guardar informações chamado de preferências, nele é necessário apenas passar uma chave e um valor para armazenar a informação desejada. (LECHETA, 2009)

Android Runtime(Ambiente de Execução Android): como todos os aplicativos do sistema operacional Android são escritos na linguagem Java, existe a necessidade de se ter uma máquina virtual Java para executar os aplicativos. Mas ficaria inviável colocar uma máquina virtual Java padrão, devido a escassez de recursos do dispositivo. Ao invés disso foi criado uma máquina virtual especial chamada Dalvik que é otimizada para executar aplicações Java em dispositivos móveis. (LECHETA, 2009). Em Java, escreve-se o código fonte Java, compila utilizando o compilador de Java, e em seguida, executa esse *bytecode* no Java VM (Virtual Machine – Máquina Virtual). No Android, as coisas são um pouco diferentes. Ainda escreve-se o arquivo fonte em Java, e continua a compilar utilizando o compilador de Java. Mas, nesse ponto, deve-se recompilar novamente usando o compilador Dalvik para gerar Dalvik *bytecode*. É este *bytecode* Dalvik que será executado na máquina virtual Dalvik (GARGENTA, 2011). Ao utilizar a linguagem Java para desenvolver aplicações móveis para Android, é gerado o arquivo o *bytecode*(.class) normalmente, depois ele é compilado para o formato DalvikExecutable (.dex) que é a aplicação Android compilada. Os arquivos compilados Dalvik juntamente com os outros recursos do programa como imagens e outros arquivos, são compactados em um único arquivo chamado Android Package File que é o resultado final da aplicação. (LECHETA, 2009)

Kernel Linux: o Android é um sistema operacional embarcado que invoca o kernel do Linux para executar os serviços solicitados pela aplicação, mas, não é um Linux embarcado. Por exemplo, utilitários normais do Linux, como o X-Windows e bibliotecas do GNU C não são suportados. Escrevendo aplicações para o Android utiliza-se o framework Java, mas não é um Java com bibliotecas padrão. Bibliotecas, como Timer não é utilizadas, pois eles foram substituídos por bibliotecas do próprio Android, que são otimizadas para uso em um ambiente de recursos limitados. (STEELE;TO, 2011). O sistema operacional Android foi desenvolvido com base do Kernel 2.6 do Linux, que é responsável por gerir todos os recursos primários do sistema. Como memória, os processos, threads, segurança do sistema de arquivos, redes e dos mais diversos drivers de hardware. (LECHETA, 2009). Os aplicativos são executados em uma máquina virtual chamada Dalvik e o sistema operacional é responsável por manter o gerenciamento dos recursos. Cada aplicação executada dispara um processo no sistema, e caso seja necessário o Android pode decidir encerrar algum processo, por falta de memória ou simplesmente porque o processo tornou-se inativo. Todo o controle de dos recursos ficam na mão do sistema operacional. (LECHETA, 2009)

4. GOOGLE APP ENGINE

Google AppEngine é um serviço de hospedagem de aplicativos web. As aplicações hospedadas neste serviço ficam disponíveis para serem acessadas em toda Web, geralmente por um navegador de internet. Aplicações como lojas virtuais, sites de redes sociais, jogos *multiplayers*, aplicações para celular, gerenciamento de projetos, publicidade, e todas as outras aplicações que a Web tem a oferecer. O AppEngine também serve com um servidor de sites tradicionais, compartilhando imagens, textos e documentos, porém este ambiente é especialmente desenvolvido para aplicações dinâmicas e em tempo-real. (SANDERSON, 2009). Este servidor é destinado a hospedar aplicações com muitos usuários simultâneos. Quando uma aplicação tem muitos usuários simultâneos consumindo recursos do servidor sem perder desempenho do sistema, então esta é uma aplicação escalar. Sistemas desenvolvidos para o GAE (Google App Engine) são automaticamente escalares. Quanto mais pessoas utilizam a aplicação, mais recursos o AppEngine aloca para a aplicação e gerencia o uso destes recursos. A aplicação não precisa conhecer nada sobre recursos utilizados. (SANDERSON, 2009). Diferentemente das hospedagens web tradicionais ou dos servidores alugados, com o Google AppEngine o usuário paga o que utiliza de recursos. Estes recursos são medidos pelo o usu da CPU (Central Process Unit), armazenamento de dados mensais, largura de banda utilizada pela entrada e saída de informações e uma série de recursos específicos para o serviço do AppEngine. O Google oferece um incentivo para quem está começando a desenvolver aplicações Web. Todo desenvolvedor ganha uma quantidade específica de recursos de graça para utilizar numa pequena aplicação com baixo trafego. (SANDERSON, 2009). O AppEngine é dividido em três partes: o ambiente de execução, o armazenamento de dados e os serviços escalares.

Uma aplicação no AppEngine responde a uma requisição web. Uma requisição web inicia quando um cliente, geralmente um usuário de navegador web, chama a aplicação através de uma requisição HTTP (Hypertext Transfer Protocol) passada por uma URL (Uniform Resource Locator). Quando o AppEngine recebe essa requisição, ele identifica qual é a aplicação a ser executada, pelo nome do domínio da URL ou por um subdomínio do domínio appspot.com (toda aplicação do GAE recebe um subdomínio do appspot.com). Então ele seleciona o servidor que vai executar a aplicação de acordo com seus critérios, sempre priorizando o servidor de melhor desempenho para a aplicação. Logo o servidor executa o aplicativo hospedado e devolve a resposta ao cliente que originou a chamada. (SANDERSON, 2009). A execução de uma aplicação no AppEngine tem algumas restrições. Ela pode ler seus arquivos, mas não tem acesso de escrita no sistema de arquivos, e não tem acesso de leitura ou

escrita a arquivos de outras aplicações. Uma aplicação pode enxergar variáveis do AppEngine, mas as alterações destas variáveis não serão mantidas entre as requisições. Ela também não tem acesso diretamente a rede ou o hardware do servidor. Todos os acessos são disponibilizados através de serviços do AppEngine. (SANDERSON, 2009). Resumindo, cada requisição vive na sua “sandbox” (Caixa de areia). Isso permite que o AppEngine escolha para cada requisição, um servidor que provê a resposta mais rápida para ela. Sendo assim, não é garantido que duas ou mais requisições da mesma aplicação serão executadas no mesmo hardware. (SANDERSON, 2009).

5. GOOGLE WEB TOOLKIT

GWT (Google Web Toolkit) é um conjunto de ferramentas de desenvolvimento voltadas para a construção de complexas aplicações WEB com base em AJAX utilizando a linguagem de programação Java, que é compilado para um código JavaScript altamente otimizado. O GWT permite que escreva aplicações em Java usufruindo de todas as vantagens dessa linguagem, e depois compile este fonte em JavaScript. Um aplicativo compilado pelo GWT consiste em arquivos JavaScript, HTML (HyperText Markup Language) e XML (Extensible Markup Language), e está aplicação que é executada igualmente nos principais navegadores de internet. (GUERMEUR; UNRUH, 2010). No GWT é possível depurar as aplicações web através de um plugin desenvolvido pelo Google para facilitar a implementação de aplicações GWT, como se fosse uma aplicação Java comum, interativamente, pode-se definir pontos de parada e analisar variáveis definidas na sua aplicação. Tudo isso em tempo de execução, com a aplicação rodando no navegador WEB. (GUERMEUR; UNRUH, 2010). Existem várias vantagens para este processo de compilação em termos de eficiência, segurança e compatibilidade de navegadores web. O JavaScript compilado é altamente eficiente e otimizado, e ajuda a proteger contra XSS (*Cross-site scripting*). Os navegadores web que o GWT tem suporte são os seguintes: Firefox 1.0, 1.5, 2.0, 3.0, e 3.5; Internet Explorer 6, 7, e 8; Safari 2, 3, e 4; Chromium Google Chrome; e Opera 9.0. Cada navegador tem suas particularidades. O GWT permite que trabalhe em um nível de abstração acima dessas diferenças. Quando o aplicativo Java é compilado para JavaScript, uma versão diferente e eficiente é gerada para cada tipo de navegador. No entanto, apenas o JavaScript exigido para aquele cliente é baixado quando uma requisição é feita para o aplicativo. (KEREKI, 2011).

O GWT é um conjunto padrão de *widgets* (como botões ou campos de texto de entrada) e painéis, que pode-se usar como componentes para a construção de interfaces para o usuário. Ele também permite que o usuário crie seus próprios *widgets*, partindo de um componente já existente ou criando um a partir do zero. Todos os estilos dos componentes podem ser definidos através de um arquivo de formatação de layouts CSS (Cascading Style Sheets). (GUERMEUR; UNRUH, 2010). *Widgets* são geralmente mapeados em objetos do navegador, para que esses componentes compartilhem o aspecto visual do navegador que o usuário utiliza. Existem também alguns objetos compostos como o DatePicker para a seleção de data, SuggestBox que é uma caixa de texto com sugestões baseadas no que o usuário digitou, RichTextArea para entrada de texto formatado e outros componentes que facilitam o desenvolvimento. Os painéis são contêineres para *widgets* ou outros painéis. O kit de desenvolvimento GWT inclui suporte para a construção de aplicações Web baseada em AJAX (Asynchronous Javascript and XML). Ele aborda o “problema” do botão voltar do navegador em aplicações Web, fornecendo suporte a gestão de histórico de navegação. Este problema decorre do fato que, em aplicações Web AJAX, o conteúdo de uma página pode mudar se alterar o endereço do URL. Nesses casos, o botão voltar não funciona como o usuário espera, nem necessariamente a URL devolve ao estado de interface que o usuário espera ver. A gestão de histórico de navegação do GWT fornece uma maneira de informar a URL sobre as

mudanças importantes de estado da aplicação, para que o botão voltar, históricos e favoritos funcione corretamente. (GUERMEUR; UNRUH, 2010)

O GWT tem uma API para chamada de procedimento remoto (RPC, Remote Procedure Call) e uma estrutura que suporta a comunicação entre cliente-servidor. Ele permite que transmita objetos entre cliente e servidor de forma simples. O pacote RPC é executado no App Engine (GAE), tornando mais simples a construção de interfaces GWT para aplicações que utilizam o Google App Engine. O pacote RPC também pode ser utilizado com outros containers Servlet, como por exemplo, o TomCat. O GWT também suporta outros meios de comunicação entre cliente-servidor, por exemplo o JSON (JavaScript Object Notation). (GUERMEUR; UNRUH, 2010). O GWT não precisa ser combinado com o GAE/Java como servidor de aplicação, porém ele é projetado para ser integrado, e trabalhar junto e de forma muito eficaz. Uma aplicação típica GWT+GAE geralmente incluem no seu pacote base, cliente, servidor e pacote compartilhados. Os pacotes compartilhados contém classes que devem ser acessíveis tanto pelo cliente como pelo servidor. Objetos de transferência de dados (DTO – Data Transfer Object) devem ficar neste pacote, para ficar disponível tanto para o cliente como para o servidor. (GUERMEUR; UNRUH, 2010). Como já citado anteriormente o Google desenvolveu um plugin de integração GWT e GAE para o IDE Eclipse, que provê uma série de recursos úteis. O plugin inclui o SDK para ambos, GWT e GAE. Ele permite o uso integrado do navegador do kit de ferramentas do GWT com o desenvolvedor, permitindo que possam ser utilizados pontos de parada em seu código Java, possibilitando a depuração e inspeção do programa fonte, de forma interativa ao acessar seu aplicativo no navegador. Com este plugin é possível implantar e carregar uma aplicação no servidor GAE, incluindo a compilação do código Java GWT e a cópia e empacotamento de todos os arquivos necessários para a implantação. (GUERMEUR; UNRUH, 2010). GWT suporta a maioria mas não todos os recursos básicos e semânticos da linguagem Java. Seu escopo é limitado pelo seu propósito. O código Java deve ser compilável em JavaScript. Como por exemplo Threading não é suportado pelo GWT. Em alguns casos, a emulação do GWT difere do ambiente de execução Java padrão, ou fornece um conjunto de funcionalidades do JRE (Java Runtime Environment). Em geral, a emulação Java do GWT suporta os seguintes pacotes: java.lang, java.lang.annotation, java.util, java.io, e java.sql. O uso de um IDE como o Eclipse é útil na indicação de quando um determinado método não é suportado. A documentação do GWT oferece mais detalhes das classes suportadas. (GUERMEUR; UNRUH, 2010)

6. O SISTEMA DESENVOLVIDO

O sistema desenvolvido esta dividido em dois conjuntos de interface: a interface Android e a interface web. O sistema todo pode ser controlado pelas telas. As telas do dispositivo móvel são de uso operacional e de fácil manuseio. Já as telas da interface Web são tanto de uso operacional quanto gerencial. A interface Android refere-se às telas que estabelecem a comunicação entre a empresa e o usuário de entrega, através de um dispositivo móvel com sistema operacional Android. As figuras de 2 a 9 apresentam esta interface.

A tela de autenticação do sistema (Figura 2) é a tela inicial do sistema na plataforma Android, e é responsável pela autenticação do usuário, liberando o uso do sistema ao colaborador da empresa. Nessa tela deve ser informado o código do usuário e sua senha, caso estas informações estejam corretas, o menu principal do sistema será exibido. O menu principal do sistema (Figura 3) possibilita a navegação do sistema. É ele que liga o sistema de uma ponta a outra, tendo como ações a abertura de telas ou o encerramento do sistema. Através dessa tela é possível abrir a lista de tarefas e de mensagens. Ela também informa o usuário que está utilizando o sistema. A tela “Tarefas” (Figura 4) exibe a lista de tarefas do usuário autenticado no sistema, esta lista contém informações pertinentes a tarefa bem como o

status da mesma. Ela permite a seleção de uma tarefa, a fim de exibir a tela de informações detalhadas da tarefa (Figura 5). A tela de informações detalhadas da tarefa (Figura 5) mostra todo o conteúdo de uma determinada tarefa. Dentre as informações estão o título da tarefa, o endereço em que a tarefa vai ser realizada, uma descrição completa da tarefa e seu status. A tela de finalização de tarefas (Figura 6) disponibiliza a funcionalidade de finalizar uma tarefa. Esta função encerra uma tarefa informando uma observação e o status (motivo) do encerramento.



Figura 2. Tela de autenticação



Figura 3. Menu principal



Figura 4. Tela de Menu de Tarefas

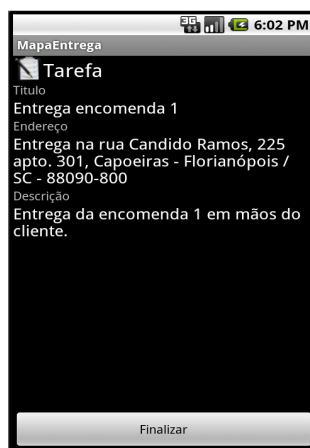


Figura 5. Tela de Tarefas

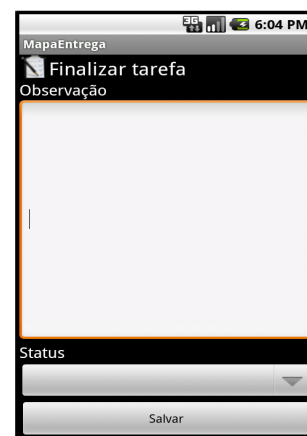


Figura 6. Finalização de tarefa

Após o encerramento o sistema já atualiza as informações com o sistema web. Permitindo assim maior controle no acompanhamento da execução das tarefas. Tela de mensagens recebidas e enviadas pelo usuário (Figura 7), exibe as mensagens contidas na caixa de entrada e de saída de um determinado usuário. Esta é uma tela muito importante do sistema, pois permite a troca de mensagens com os usuários mantendo-os atualizados com informações. Também é um redutor de custos para a empresa, utilizando ela a empresa pode deixar de utilizar outro tipo de comunicação tarifada, para dialogar com seus usuários. A tela de criação e envio de mensagem (Figura 8) permite que o usuário crie e envie mensagens a partir do sistema móvel. Nesta tela o colaborador de selecionar um usuário de destino, escrever a mensagem no campo destinado a essa informação e enviar. Instantaneamente o usuário de destino já recebe a mensagem do colaborador de origem. A tela de visualização de mensagem (Figura 9) permite que o usuário visualize o conteúdo de uma determinada mensagem. As informações contidas nas mensagens são, a data e hora do envio da mensagem, o usuário que a enviou e o texto da mensagem. Nesta tela é possível apagar, responder e encaminhar uma mensagem.

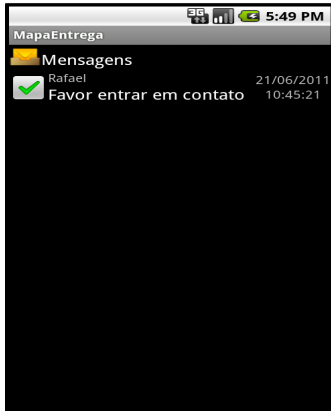


Figura 7. Tela de mensagem

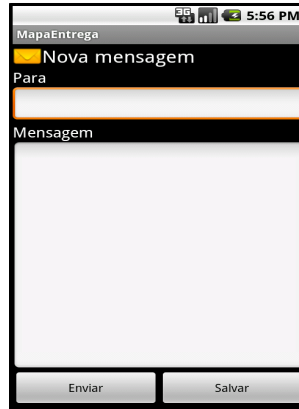


Figura 8. Nova mensagem

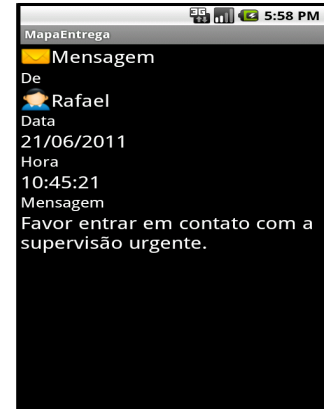


Figura 9. Ler mensagem

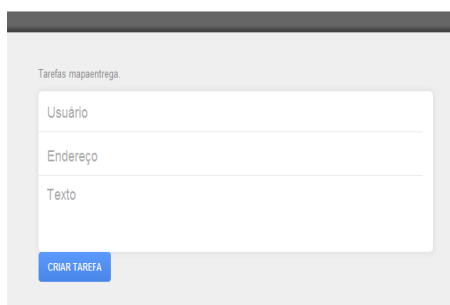


Figura 10. Tarefa

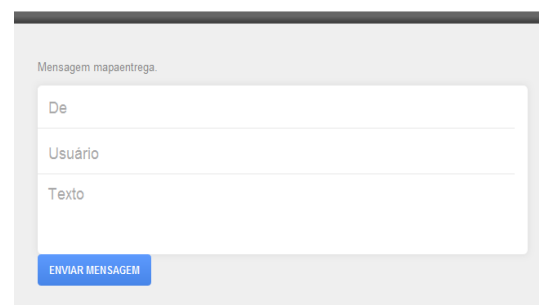


Figura 11. Mensagem

A interface web refere-se as telas responsáveis não só pela interface da parte operacional mas principalmente da parte administrativa e gerencial da aplicação. A tela de autenticação de usuário, assim como no sistema do dispositivo móvel, é a primeira interface que o usuário irá utilizar, garantindo a segurança na entrada do sistema. No cadastro de tarefas podem ser realizadas alterações, exclusões e inserções de novas tarefas (Figura 10). Na tela de mensagens (Figura 11), o responsável pode criar, alterar e excluir mensagens dos usuários do sistema.

7. TESTES

Para testar a aplicação desenvolvida, foi realizado um teste ergonômico e um teste de desempenho, executado por especialistas da área logística e analistas de sistemas. O instrumento dos testes encontrasse no Apêndice A seguindo a norma “QUALIDADE DE PRODUTOS DE SOFTWARE - ISO 9126”. Após a aplicação desse instrumento chegou-se a análise descrita a seguir.

No quesito funcionalidade o sistema apresentou uma ótima acurácia com 90% de aceitação dos usuários. Nos demais itens ficou assim: 80% de aceitação no item “Adequação”, 80% no item “Interoperabilidade”, 80% no item “Conformidade” e 70% no item “Segurança de acesso”. No quesito confiabilidade o sistema demonstrou ter maturidade média com algumas falhas na obtenção de dados da internet, quando a rede de dados não está disponível. Quando isso ocorre o sistema dá uma mensagem de erro e é encerrado. Quanto ao item “Recuperabilidade” o sistema teve 40% de aprovação. No quesito usabilidade foi demonstrado que o sistema tem excelência em todos os itens obtendo 100% de aprovação em “Inteligibilidade”, “Apreensibilidade” e “Operacionalidade”. Caracterizando ser um sistema de fácil utilização e aprendizado. No quesito eficiência, o sistema apresentou ser rápido, com média de 3 segundos no tempo de resposta dos dados obtidos do sistema web. Utiliza poucos recursos do celular, por muito pouco tempo. No quesito “manutenibilidade” notou-se que o sistema tem é de fácil modificação e adaptação com 100% de aceitação. No item

“Analisabilidade” obteve-se 90% de aceitação dos usuários, 80% de “Estabilidade” e 90% de “Testabilidade” do sistema. Já no quesito portabilidade, o sistema demonstrou ter “Conformidade” com 90% de aceitação dos usuários consultados. Nos demais itens obtiveram-se os seguintes resultados: 70% de “Adaptabilidade”, 50% de “Capacidade para ser instalado” em outros ambientes e 60% no item “Capacidade de substituição”.

8. CONCLUSÕES

O sistema de controle de entrega aqui apresentado pode ser utilizado em empresas logísticas que atuem na área de entregas e coletas de mercadorias, podendo ser necessários pequenos ajustes para o funcionamento deste sistema. O início do projeto começou com Java ME, mas com os avanços da tecnologia no período de desenvolvimento e análise do sistema, optou-se por utilizar a plataforma Android como padrão de desenvolvimento móvel.

A pesquisa desenvolvida ao longo deste trabalho buscou apresentar as principais características da plataforma Android, visando o desenvolvimento de aplicações para os dispositivos móveis. Também se analisou o *framework* Google Web Toolkit para o desenvolvimento de interface web e o servidor de aplicação Google App Engine. Conclui-se que a plataforma Android é um grande facilitador para o desenvolvimento de aplicações móveis, pois o sistema desenvolvido para o Android, pode ser utilizado em qualquer dispositivo móvel que possua o sistema operacional Android, independente do fabricante do celular. Outras vantagens também se encontram ao se desenvolver para esta plataforma, como um vasto conjunto de APIs que vão desde banco de dados a integração com GPS, aproveitando assim todo o potencial que o dispositivo pode lhe oferecer. Pode-se concluir também que ferramentas como o Android, eclipse IDE, Google Web Toolkit, Google App Engine e plugins desenvolvidos pelo Google, facilitam e muito o desenvolvimento, a implantação e a distribuição de aplicações Web e móveis. A implementação do sistema foi realizada de forma lenta, pois como se trata de tecnologias novas e inovadoras, teve-se o trabalho de aprender a utilizar as ferramentas de desenvolvimento e suas novas plataformas, onde foi perdida a maior parte do tempo. Outra dificuldade foi a comunicação entre o celular e o servidor. Após análises utilizou-se uma API de integração do Google chamada Android Cloud to Device Messaging Framework, Essa API foi liberada para uso no mês de maio de 2011. Essa API facilitou muito a comunicação entre o Android e App Engine.

Os testes foram realizados por 10 usuários, entre eles estão especialistas da área logística e analistas de sistema. O instrumento dos testes foi o formulário (Apendice A) de qualidade de software seguindo a norma ISO/IEC 9126. Os testes apresentaram um resultado satisfatório quanto a utilização do sistema. O tempo de acesso a dados é de em torno 3 segundos em redes móveis, podendo variar de acordo com o tipo de conexão. O desempenho do sistema depende de várias variáveis, como, velocidade do processador, memória e tipo de conexão de internet do *smartphone*. O sistema também apresentou ser confiável, de fácil utilização, eficiente no seu propósito, de fácil manutenção e portátil.

Como trabalhos futuros, pode-se trabalhar na adaptação e melhoria do sistema aqui apresentado, para que o mesmo funcione em novos dispositivos e sistemas operacionais que vem mudando os conceitos de telefone celular, como o iOS da Apple e Symbian Nokia. Implementações para utilização de GPS e integração com o Google Maps é muito interessante para esse tipo de sistema.

9. REFERÊNCIAS

ANDROID, Disponível em: < <http://developer.android.com/guide/basics/what-is-android.html> >. Acesso em: mar. 2011.

AUGUSTIN, I. Acesso aos dados no contexto da Computação Móvel. Porto Alegre: PPGC/UFRGS, 2000. (Exame de Qualificação).

- BALLOU, R. H.** Logística empresarial: administração de materiais e distribuição física. São Paulo: Atlas, 1993.
- DUARTE, R. L.** Utilização da tecnologia J2ME para acesso ao serviço de correio eletrônico. 2005. 146 f. Trabalho de Conclusão de Curso (Bacharelado em Ciência da Computação) – Universidade do Vale do Itajaí, São José, 2005.
- FIGUEIREDO, C.M. S.; NAKAMURA, E.** Computação móvel: novas oportunidades e desafios. Revista T&C Amazônia, [S.1], n. 2, jun. 2003. Disponível em: <https://portal.fucapi.br/tec/imagens/revistas/ed002_016_028.pdf>. Acesso em: 27 set. 2006.
- GARGENTA, Marko.** Learning Android. Sebastopol: O'Reilly Media, 2011.
- GOOGLE WEB TOOLKIT.** Disponível em: <<http://gwt.google.com/samples/Showcase/Showcase.html>>. Acesso em: 20 mai. 2011.
- GUERMEUR, D.; UNRUH, A.** Google App Engine Java and GWT Application Development. Birmingham: Packt Publishing, 2010.
- IMASTERS.** J2ME - Java para os portáteis. Disponível em: <<http://www.imasters.com.br/artigo.php?cn=1539&cc=19>> Acesso em: 10 jun. 2007.
- IMIELINSKI, T.; VISWANATHAN, S.; BADRINATH, B. R.** Data on Air: Organization and Access. IEEE Transactions on knowledge and Data Engineering, v.9, n. 3, 1997.
- JING, J.; HUFF, K.** Adaptation for Mobile Workflow Applications. In: WORKSHOP ON MODELING AND SIMULATION IN WIRELESS SYSTEMS, 1998, Montreal, Canada. Proceedings... Montreal, Canada, Jul. 1998.
- KALAKOTA, Ravi.** Adaptation for Mobile Workflow Applications. In: KALAKOTA, Ravi; ROBINSON, Marcia. M-business: tecnologia móvel e estratégia de negócios. Tradução de Maria Adelaide Carpigiani. Porto Alegre: Bookman, 2002.
- KEREKI, Frederico.** Essential GWT : building for the web with Google Web toolkit 2. Boston: Pearson Education, 2011.
- LAMBERT, D. M.; STOCK, J. R.** Strategic Logistics Management. 3. ed. Chicago: IRWIN, 1993.
- LECHETA, R. R.** Google Android: Aprenda a criar aplicações para dispositivos móveis com o Android SDK. São Paulo: Novatec, 2009.
- MORIMOTO, C.E.** Entendendo o Google Android. Disponível em: <<http://www.guiadohardware.net/artigos/google-android/>>. Acesso em: out. 2008.
- NAZARIO, P. R.** A importância de Sistemas de Informações para Competitividade Logística. Revista Tecnológica, jul. 1999.
- NOVAES, A. G.** Logística e gerenciamento da cadeia de produção. Estratégia, operação e avaliação. Rio de Janeiro: Campus, 2001.
- O GLOBO.** Google lança 'Android', sistema operacional aberto para celular. Disponível em: <<http://oglobo.globo.com/tecnologia/mat/2007/11/05/327030463.asp>>. Acesso em: out. 2008.
- QUALCOMM.** Developing with BREW: Easy and Smart. Disponível em: <<http://brew.qualcomm.com/brew/en/developer/overview.html>> Acesso em: out. 2008.
- ROMÁN, F.; MENSONES, F. G.; MARINAS, I.** Mobile Marketing: A Revolução Multimídia. São Paulo: Thomson, 2007.
- SACKS, A. G.** Sistema de gerenciamento de redes e processos através de computadores portáteis via Bluetooth. Rio de Janeiro, 2003.
- SANDERSON, Dan.** Programming Google App Engine. Sebastopol: O'Reilly, 2009.
- STEELE, J.; TO, N.** The Android Developer's Cookbook Building Applications with the Android SDK. Boston: Pearson Education, 2011.
- TERRA,** Redação. Do tijolo ao smartphone: 25 anos de celular. Disponível em: <<http://tecnologia.terra.com.br/interna/0,,OI3266753-EI4796,00.html>>. Acesso em: out. 2008.
- WEBMOBILE,** Artigo WebMobile 11: Introdução a Plua. Disponível em: <<http://www.devmedia.com.br/articles/viewcomp.asp?comp=4909>>. Acesso em: out. 2008.